An Intelligent Content Filter based framework for Mobile Web Services

J RONALD MARTIN, S ALBERT RABARA
Department of Computer Science
St. Joseph’s College, Bharathidasan University
Tiruchirappalli, Tamil Nadu
INDIA
Martinronald.007@mail.sjctni.edu, rabara@mail.sjctni.edu

Abstract: - Since mobile and internet penetration is highly increasing all over the world and these technological advancements of mobile devices are changing every time, most of people are using their mobile devices for their day to day transactions, business etc., and access services from the internet. Today, lot of search engines are available, these search engines will dump information abundantly. People are spending most of time for choosing the required service from the internet. In this context the research on intelligent web services are inevitable. Hence, in this paper we have proposed a novel framework for mobile web services with intelligent content filtering.

Key-Words: - Content Filter, Mobile Web service, mobile agents, intelligence

1 Introduction
In current scenario, web services has become one of the standard technologies for sharing information, in recent days number of web services accessible on the internet is constantly increasing. According to recent statistics, there are 28,606 web services available on the Web, provided by 7,739 different providers [1].Subsequently mobile and internet penetration is very high and also increasing everyday all over the world and the technological advancements of mobile devices are changing every time, now-a-days people are using their mobile devices for their day to day transactions, business and access the services from the internet.

The mobile interoperable framework is important in system design, because it has a direct impact on the service availability, usability and performance. Currently, the capabilities of mobile devices have been significantly enhanced by faster processors, larger installed memory and enhanced user display. Due to these advantages of mobile devices, people can search huge amount of information through their mobile devices, but people wants to spending time in choosing the required web services from the result. In this context the research on intelligent content filter for web services is inevitable.

Presently, anyone can host the content on the web; some contents are harmful to children and innocent adults. The aim of the Content filtering mechanism is to filter unwanted information before the content reaching the users. Currently some software filters are available like WebChaperone [2], the software content filters are blocking the access of URLs only. So, some kind of intelligent mechanism is need. This paper proposes a novel framework for mobile web services with intelligent content filter in secure manner. This framework enables service integration, mobile agent, and content filter and supports access to the services from any mobile devices with security.

The rest of the paper is organized as follows: Section 2 represents the related work on service integration, mobile gents and content filtering. The intelligent, integrated architecture is proposed for securing mobile web services that includes intelligent service categorization, mobile agents and content filtering are presented in section 3. The security algorithms are presented in section 4. The section 5 presents the testing scenario and the experimental results. Section 6 concludes this paper.

2 Related Work
Jinka et al [3] have developed an intelligent framework for social network data analysis namely (PERISIKAN). This proposed framework provides a way for analyzing and comparing formal and informal information flows in an organization, as well as comparing information flows with officially defined work process. This proposed framework consists of major high level components such as Social Computing Unit (SCU): SCU implies the social networking organizations including FaceBook, LinkedIn, MySpace and the likes. Data Extraction and Filtering Engine: Data extraction component includes the required APIs and technologies that could periodically extract the required data from SCUs. Filtering engine is a combination of set of parsers and stemmers that could allow the require data to be sent to the data store. Modeling and Indexing Engine: modeling and indexing on data store periodically. Pattern Search and Visualization Engine: this is a user interface component that allows user to search for required information from the data store and presents the results in a user friendly manner. The author also presented the case study on FaceBook wall posts emotion analysis.
using the framework and the steps i) Search for groups, ii) Analyze messages for emotion posted on to the FaceBook groups, iii) capture the other details of the wall posts, iv) Identify location clusters of customers, and v) Build and Visualize social graph. This framework consists of six steps such as Grouping Parsing, Group Indexing, Emotion interpretation, Bias interpretation, Graph visualization, and Location visualization. This proposed framework aims to provide solution only for social network analysis based on the concept of view and not intended for providing any kind of security services in order to handle the social network services in a secure manner.

Ye Zhong et al [4] have developed mobile agent based framework for web service integration. This framework uses the agent model in the normative JADE (Java Agent Development Environment) which is compatible with FIPA (The Foundation for Intelligent Physical Agents). Framework includes two types of agents such as Mobile Agent and Non-mobile Agent. The mobile agent consists of two agents namely Guard Agent and the Procedure Agent. The service Agent in the MAWSI framework represents a web service cluster. This framework provides three combinations mode for web service integration such as Sequence, Exclusive and Parallel. This framework supports off-line operations which can overcome hidden perils in network, and provide reliable services in an unreliable network. The author verifies the validity of the framework in two different network connections, Web services integration time calculated in two different manners, first one consider with Mobile Agent technology, second one consider without agent technology. Finally the result has little different between the mobile agent based web service and non -based one. The time spent on the mobile agent based web services integration is less than the spent on the non-based one. However, this framework does not support any kind of service matching mechanism; it does not provide any security supports to service handling through mobile agents.

Elton et al [5] developed intelligent agents for e-health. This architecture proposed a multi-agent architecture able to organize the complex work rescue operations, and to support the human staff to take the appropriate decisions. The architecture defines various agents such as Activator agents, Device service agent, Client service agent, Mobile Operative agent and Stationary Operative Agent. The Service Agent (SA) represents a mobile device such as notebook or PDA that a user’s uses to insert or read data. The corresponding agent divided into two categories such as Device Service Agent (DSA) is used to represents the medical device. Client Service Agent (CSA) is the agent represent the client interface device of an operator such as notebook, PDA, smart-phone. This agent retrieves the information from the client device and collects information from other DSAs to be visualized. Operative Agent (OA) is the agent represents the operative unit like ambulance, medical care, hospital etc. it split in to two categories: Mobile Operative Agent (MOA) this agent communicates with service agent to obtain the necessary information. Stationary Operative Agent (SOA) is the agent that represents a stationary unit like hospital or temporary first aid camp. The activator is special agent, it activating the other agents for ready to be used and deactivating the operative units when no more available. It receives the requests from the operative center for new operations and dispatches them to the MOA. However, the secure communication between agents is another concern and there is no technical information about them which leads to shortcoming of this architecture.

Zhang et al [6] have developed a framework for Mobile Web Service. This framework each device publishes a WSDL description on a Service Broker. The Web Service Description Language (WSDL) describes the public interface to the web service. The framework consists of five separate layers: an Input / Output core, a Procedure Management tier, a Data Access tier, a Data Storage tier, and a Mobile soap Engine. The Input / Output tier aggregates methods by which users interact with application. The Core Management tier encapsulates the core business model which provides the basis for all business transactions of the application. The Data Access tier provides Data Access Interface (DAI), and a Data Center (DC). The DAI provides a command interface to the database and isolates database operations from data queries. The mBOSS SOAP engine consists of eight parts. The function of the Message Handler is to transfer request messages to the SOAP receiver and transfer SOAP messages from the SOAP sender. The HTTP is used to transfer messages. The framework request messages are transferred in an HTTP POST request. The response messages are transferred in HTTP Response message. The message type is "text/xml" which is used for the messages containing the SOAP envelope. This framework test conducted the average size of Simple, Medium and Complex messages. The Simple messages include the text only, the Medium and Complex messages include multimedia data, and the sizes dramatically increase from 20 to 80 times. The framework failed to adopt security mechanism between the service requester and service provider.

Evan et al [7] have proposed an intelligent framework for mobile devices. The author described usage of mobile devices. The author classified the mobile hardware and software issues. The hardware issues include screen size, Insufficient Memory, Low Battery Power. The author proposed a component-based framework for mobile applications. This framework addressed the issue of bloated XML files used in a SOA environment. The XML files are used to storing information for use within framework. The framework supports both data and media files in the form of XML and images, respectively.
allowing an intelligent compression process. This framework uses two types of compression process such as Lossless and Loss. The lossless compression process is a condensed form of file for transmission or storage. This file can be reconstructed when required. The loss compression process eliminates unnecessary bits of information, trimming the file so that is smaller. XML compression reduces the quantity of data transferred to and stored on the mobile device. XML compression decreases the necessary bandwidth, storage, and processing on mobile device. The Loss compression process used to compress images. The author tested the effectiveness and efficiency of proposed framework. The test based on the following scenario i) Different mobile screen sizes, ii) Image files of different files and types and iii) XML files of different sizes and different data stored. The framework tested both data and media files. The author presented the results for response time, initial size vs. compressed size transfer rate and wireless transfer rates. The framework transferred XML and image files without security. However, this framework does not ensure the significance of security service.

Jangwon et al [8] have developed multi-layer based access control model for GIS mobile web services. The model describes how to provide a GIS Mobile Web services to each user. This model first gather distributed resources from other GIS Web services such as WMS, WFS, and WCS. This model offer an optimal service, it needs to merge various kinds of GIS Web services which have different their own rights or access control authority. The GIS web services issues a secure problem, therefore this model provide secure web services using access control. The model creates various policies to be assigned to users. The author used a Google mobile phone in client side. The author developed GIS web services in mobile computing environment and demonstrate with GIS services of OGC standard such as WMS, WFS, and GML. The multi-layer model support secure access control policy, but it does not address the remaining security mechanism such as authentication, integrity and confidentiality.

Having studied the literature, there is a growing need for the development of mobile web services with content filtering. Even though there has been considerable amount of research work carried out in the area of mobile web services, but there is no concrete proposal offered the so far to build an intelligent content filter based mobile web services across different information service providers with security. Hence, a novel framework has been proposed for the various service requesters with the aim of providing integrated and intelligent secure web services through mobile devices.

3 The Proposed framework

The Proposed framework defines the basic functions, their relationships and guidelines for interactions of the architecture. It allows the system developers to adapt specific functions according to their demands. The framework identifies repeatable elements, consistent interfaces and then take into consideration the trade-offs. The modern framework design encourages the concept of separation in business functionality from the infrastructure that supports increased understandability of the system for both implementation and deployment. It also enforces increased scalability and reusability of components in the system. The security services are realized in the same manner as the business services and the same technologies and patterns can also be used to combine them.
The proposed framework Intelligent, Integrated Secure Mobile Web Services (IISMWS) is designed mainly to provide secured and filtered web services and well-suited to handle public related services anywhere and anytime through their mobile handheld devices. The framework consists of five major components such as Mobile Client Manager, Intelligent Integrated Manager, Mobile Agent Manager, Content Filter Manager and Corporate Service Manager. The framework is designed as per the standards with built-in services for service requisition, service validation, service invocation, service configuration, composition of services, and service execution with multi-level security. The proposed framework presented in Figure 1. The functionalities of the proposed framework are briefly presented.

**Functionality of the proposed framework**

The mobile user handles a verity of mobile devices which have different technical and physical characteristics in terms of performance and its functionality. The recent mobile phones like smart phones having same capability like the desktop computer, which makes them a potentially reliable access of mobile services. There are several components are available for the mobile client to access the mobile web services.

### 3.1 Mobile Application

The mobile application software consists with several components. The responsibility of User Interface Manager is to acquire and validate data entered by the mobile users as per business and security logic. Android is the preferred development platform for mobile application due to the portability of Java code, and ability to establish a different security at the client device.

The Request/Response Handler (RRH) handles the user service requests and service responses raised by the clients. The Communication Manager (CM) establishes and message connectivity with the AAA server. It allows the exchange of service request in a secured way between the communication entities. The CM keeps the details about the source and
destination of communication, information about communication list, the communication enabled time, etc. the CM also considered communication issues such as performance, security and privacy, litigation, etc. the mobile user device executes the user’s request including service request in multitasking environment.

The Synchronization Manager’s main responsibilities are sending and receiving the data between the users through the AAA and the ISC server. During the data synchronization, The Security Manager encrypts and decrypts the users’ request. The Security Manager is also responsible for device, user authentication. The Certificate Manager validates the server certificates with Certificate Authority (CA), signing the request, certificate management, secret key management and distribution etc. the device information manager keeps the information about the mobile application Business Logic, validations and alerts, and authentication information. The Backup Manager picked up the failed transactions from the restore point and to resume the data, instead of restart again. The Log Manager maintains the log for each service request with Log ID, time at which the log is created, and so on.

The mobile users start to communicate with the IISMWS through activating the mobile application icon from their mobile device. Once activating the mobile application, the activated client device authenticate by the AAA server. After the successful device authentication, the mobile application interface enables the prompts for e-mail ID and password for user authentication. Once the client makes entries, those parameters are encrypted using SecCode and sent to the AAA server in secured way.

3.2 AAA server

The AAA server is designed with the following manager i) CommunicationManager, ii) SecurityManager, iii) ServiceManager, iv) DatabaseManager. The main aim of this CommunicationManager is to establish and authenticate communication between the mobile user and Intelligent Service Composition (ISC) server The AAA server is to establish the communication with the mobile user and between the ISC server. The SecurityManager encrypts and decrypts the user request information that comes from client mobile device to AAA server and vice versa by using asymmetric key scheme. The AAA module contains APIs for to validate user and users’ device. Once the initial authentication is completed by the SecurityManager, the service requests are sent to the ServiceManager. The ServiceManager receives all the service requests and creates a session for each service request. Then, the Service Manager communicates with the DatabaseManager keeps the records of all users’ profile including their secret information in an encrypted format.

Once the connection is established with user device, the server is authenticated to the user and then the user parameters e-mail ID and password are decrypted using the SecurityManager, otherwise, the AAA server will send an error message to the client device. The AAA modules validate the users with the help of DatabaseManager through ServiceManager. After completing the authentication process by the AAA server, it establishes the communication with the ISC server to upload the registered service list by the corporate service providers to mobile user device. The user can chose their service from their service list and make the service request to ISC server through AAA server. The AAA server encrypts the service request with the help of SecurityManager and encrypted service request sends to the ISC server to configure web services.

3.3 Intelligent Service Composition (ISC) Server

Once the ISC server receives the encrypted service request of authenticated users’ from AAA server through the CommunicationManager of ISC server. Once the communication is established between the AAA server and the ISC, both of them authenticates each other using challenge-response mechanism with the help of SecurityManager. After the successful authentication, the ISC server decrypts the service request with the help of SecurityManager. The ServiceRequestQueueManager is responsible for defining a queue for each service request as well as for service responses. The ServiceManager configuring service based on the service queue and creates a session for each service. During the service configuration the ServiceManager obtain the service information such as service ID, URL of the service and WSDL from the UDDI server. After the service configuration the ISC server establishes the communication with the Intelligent Mobile Agent.
Server (IMAS) through the CommunicationManager to generate agents for each service. The ServiceIntegrationManager obtain the service responses from the mobile agents and integrate the service responses. Finally, the integrated service responses send to the user through AAA server in secure manner.

The ServiceCategorizationManager categorizes the registered web services into three categories such as Simple Web Service (S-WS), Transaction Information Web Service (I-WS), and Transaction Financial Web Service (F-WS). The ServiceCategorizationManager applies an Automatic Web Service Categorization Algorithm to categorize the web services.

3.4 Intelligent Mobile Agent Server

The IMA server authenticates the ICS server with the help of SecurityManager. Once the ISC server authenticated, the IMA server obtain the web service information from the ISC server then IMA server generate agents for each service to obtain the service responses from corporate service providers. The MobileAgentManager responsible for the agent creation, agent behavior, and agent life cycle states. The generated agents are compatible with the Foundation for Intelligent Physical Agent (FIPA). The generated agents send the service request to the corresponding corporate server as an encrypted format. Mobile agent and web service request is validated by the corporate, if the mobile agent and service request is valid one; the web service will process the service request and provide the service response. The mobile agent obtains the service responses from the corporate and forwards the service responses to the Intelligent Content filter Server to filter the unwanted information from the received service responses.

3.5 Intelligent Content Filter Server

The intelligent content filter (ICF) server is multi-agent server. The ICF server consists of Core Engine, Load Balancing Manager, Computing Manager, Content Storage Manager and Client Manager. The Core Engine is a mobile agent server, it is consists of Blacklist Agent, Header Check Agent, Classification Agent. Initially the core engine obtains the service responses from the mobile agents and creates log for the each service responses. This log file contains the mobile agent ID, mobile use ID, and which time the log was created etc. The Blacklist Agent extracts the address from the each service response and check with the blacklist, if the address is in the blacklist; the Blacklist Agent immediately refuses the particular service response. The Header Check agent analyses the domain names of the service response, if the header check agent refuses the domain names are the illegal after the domain name analyses the header check agent extracts the abstract of the each service response content for feature extraction. The classification agent uses the Naive Bayesian filter method to classify the service responses according to the features. The non-spam contents are marked as Ns and stored in the Non-spam set of content storage manager, the spam contents are marked by Sp and stored in the spam set of content storage manager, and then the unclassified contents are stored in unconfirmed set in the content storage manager. The load balancing manager is non-mobile agent, it contains load balancing agent it executes all the process by computing agents and also monitoring their computing process. The client agents are get the non-spam contents from the non-spam set of the content storage manager and forward through the mobile agents to the mobile user. The overall functionalities of the proposed framework presented in Figure 3.
3.6 Naive Bayesian Classification

The working principle of the Naive Bayesian Filter is described as following. Initially, the filter studies from the spam set and the non-spam set to start the feature vectors of spam and non-spam. When content is received from the agents, then the filter extracts the features of the contents and establishes the vector space of the contents. Then the filter works out the probability of belonging to spam (SP1) and the probability of belonging to non-spam (NP2). If SP1>NP2, then the content is a spam, otherwise the content is non-spam.

Suppose the content L has n features (w1, w2…wn), and the sample space has two sorts: spam C1 and non-spam C2. Assume N1 is the amount of the content belonging to C1, and N2 is the amount of the contents belonging to C2. The probability SP1 and NP2 are:

\[ SP_m = SP(C_m | L) = SP(C_m) \times SP(L | C_m) \quad m=1,2 \]

Where SP (C1) and SP (C2) are:

\[ P(C_m) = \frac{1+N_m}{2+N_1+N_2} \quad m=1,2 \]

The probabilities SP (L|C1) and SP (L|C2) are:

\[ P(L|C_m) = P(w_1, w_2, \ldots, w_n | C_m) = \prod_{j=1}^{n} P(w_j | C_m) \quad m=1,2 \]

SP (w1|C1) is the probability of w1 belonging to C1, and SP (w1|C2) is the probability of w1 belonging to C2. They can be computed as follows:

\[ P(w_i | C_m) = \frac{1+TF(w_i, C_m)}{|V| + \sum_{j=1}^{n} TF(w_j, C_m)} \quad m=1,2 \]

where |V| is the amount of features, and TF (w1, Cm) is the sum of times of w1 appearing in Cm. The weight is applied to increase the precision of the classification.

4 Various Security Levels of the proposed framework

The proposed framework is to enhance the security at various levels such as Device level, Client level, Service level, and Agent level.

4.1 Device Level: Initially, the mobile device is authenticating through the certificate manager. The certificate manager verifies the device certificate with trusted Certificate Authority using Online Certificate Status Protocol (OCSP) mechanism. In this proposed model, X.509 user certificate is applied to authenticate the mobile device.

4.2 Client Level: the mobile application software request for e-mail ID and password for the user authentication and the entered e-mail ID and password is send to the AAA server as an encrypted format using Public Key Infrastructure (PKI). The AAA server decrypts the e-mail ID and password and its compares with database. If both them match, the mobile user is authenticated by the AAA server. Otherwise the access denied message send to mobile user.
4.3 Service Level: The proposed framework incorporates the Web Services security, “WS-Security” from Organization for the Advancement of Structured Information Standards (OASIS). The proposed framework also supports XML Encryption to encryption of XML documents, and XML Signature to message digest, message authentication code or digital signature to authenticate XML documents.

4.4 Agent Level: The X.509 certificates used to validate the mobile agents. The agent X.509 certificate exchange with public key between Agent server and corporate service providers once the certificates is validated, the mobile agents are communicate with the corporate service providers.

5. Sequence diagram of the proposed framework

![Sequence diagram of the proposed framework](image)

Figure 4 Sequence diagram of the proposed framework

6 Experimental Setup & Performance Results

6.1 Test Bed

The test bed is established based on a 3-tiered architecture where the mobile user-side functional components are implemented using Java 2 Enterprise Edition (J2EE) and Android 4 development kit. The major advantages of J2EE over the other competing technologies are better portability and the simplicity for the implementation. Moreover, J2EE is a platform independent technology. The functional components are developed using Java servlets according to the standard used in Tomcat server 7.1 on Red Hat Enterprise 6.0. The AAA server has the major responsibilities such as authenticating the mobile users, processing their profiles and forwarding the user request to intelligent Service composition.
server (ISCS). The proposed framework provides API specifications and the interface description for authenticating and authorizing the registered mobile users, validating the web services and preparing the service request. The Certificate Authority server handles the certificates validation functionality. The test bed of the proposed framework is presents in Figure 5.

![Figure 5: Test bed of the proposed framework](image)

The mobile user interface has been implemented using the Sun Java Developer Toolkit with secure transactions by enabling Hypertext Transfer Protocol over SSL (HTTPS) connections. The functionality of the mobile user interface has been developed using Android Development kit 4 it supports elegant Graphical User Interfaces (GUI), business logic for the service requester and the ability to support secure communications with the server. Figure 6 represents the software setup of the proposed framework.

![Figure 6: Software setup of the IISMWS](image)

### 6.2 System Implementation and Testing

The mobile applications have been developed using Android Development kit 4. This platform is organized in the large-scale, federated security architecture. The internal components of the architecture are different types of servers and workstations. The internal structure of the proposed framework consists of the following modules:

- **GUI module**: It handles the creation of all the user interfaces, related objects such as forms, text editors, choice boxes etc.
- **Communication module**: It includes the creation of communication interfaces for all supported protocols between mobile users and server.
- **Business logic module**: It has the responsibility for creating web service request messages and processing the web service responses messages.
- **Security module**: It implements the security elements to provide security features such as authentication, authorization, confidentiality, integrity and non-repudiation for the web service transactions and web service-oriented operations.

### 5.3 Performance Comparison

The main objective of this task is to investigate the comparative study of the processing time for the execution the results of web services with content filter and without content filter. Our proposed framework incorporates WebChaperone, which is one of the primary content filter software. In order to achieve this, test cases with varying web service results ranging from 10 to 200 has been considered for studying the response time which is the total time spent for filtering the web service and responding to the mobile users through the mobile agents.

The response time of the web service with content filter is denoted as $RT_{cf}$ and the response time of the web service without content filter as $RT_{nocf}$. Table 1 shows the response time for the web services with and without content filter.

<table>
<thead>
<tr>
<th>No. of Service request</th>
<th>$RT_{nocf}$ (ms)</th>
<th>$RT_{cf}$ (ms)</th>
<th>Increase</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>12</td>
<td>15</td>
<td>3</td>
</tr>
<tr>
<td>20</td>
<td>22</td>
<td>31</td>
<td>9</td>
</tr>
<tr>
<td>40</td>
<td>31</td>
<td>43</td>
<td>12</td>
</tr>
</tbody>
</table>
Table 1: response time for the web services with and without content filter.

The response time for the web service access with content filter and without content filter has been graphically illustrated in Figure 7.

Figure 7: graphical represent of the response time for the web services with and without content filter.

6 Conclusion

We propose an intelligent and secure framework to access web services from mobile devices with content filter. This proposed framework explained with their components, functionalities. The proposed architecture tested with test bed and the software setup of the proposed framework also presented. The response time of the proposed framework measured with and without content filter. The results also presented graphically. As a future work, we are planning to implement the intelligent content filtering mechanism with real time public related web services to ensure the accuracy percentage of the content filter with mobile web services.

References:

[1.] Lina Yao and Quan Z. Sheng, Recommending Web Services via Combining Collaborative Filtering with Content-based Features, 20th International Conference on Web Services, IEEE, 2013.


