Bachelor's thesis

Information Technology

2017

Tingting Liu

RESEARCH AND IMPLEMENTATION OF THE SOA PROGRAMMING MODEL

TURKU AMK
TURKU UNIVERSITY OF APPLIED SCIENCES

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Information Technology

2017 | 56

Tingting Liu

RESEARCH AND IMPLEMENTATION OF THE SOA PROGRAMMING MODEL

The application systems of an enterprises are increasing gradually with the development of the IT industry. There are several heterogeneous systems left after the software development, some of them are programmed with different languages, and some are running on different system platforms. Thus, aggregation and integration of legacy system is a great problem for the modern IT companies. That is the reason why Service-Oriented Architecture (SOA) has been used by some IT elites. SOA has the characteristics of loose-coupling, regardless of technology restriction, integrated, it is manageable and maintainable, and has became one of the most popular software architecture in recent years. Although there is a variety of technologies which has been brought forward to this area, but the most attention-grabbing one among these technologies is SCA (Service Component Architecture). SCA is a specification which is used to describe in a constructs application and system model with SOA. Systems based on SCA have more advantages than SOA in some specific area.

This thesis has researched the critical technique of making use of SOA to construct application and system, a design and implement the integration of the legacy system, it also provides a new method for a small and medium-sized enterprise which needs to integrates legacy information system. Besides, analyze the SOA Architecture theory system and Web Service security protocol, focusing on an analysis of the SCA specifications and the SCA Security policy of the SCA policy framework are also included. Finally, this thesis also has made use of the design ideas of SOA and SCA programming model, with Apache Tuscany and PHP SCA implementation technique, to achieve the integration among the legacy MIS system for the company.

KEYWORDS:

SOA, SCA, SCA Component, Web Service, programming model

CONTENTS

FIGURES

LIST OF ABREVIATIONS (OR) SYMBOLS

SOA	Service-Oriented Architecture
SCA	Service-Component Architecture
C/S	Client/Server
B/S	Browser/Server
Loosecoupled	one in which each of its components has, or makes use of, little or no knowledge of the definitions of other separate components
SOAP	Simple Object Access Protocol
WSDL	Web Services Description Language
UDDI	Universal Description, Discovery, and Integration
WS-Security	Web-Service Security
QoS	Quality of Services
IoC	Inversion of Control
WSIF	Web Service Invocation Framework
Heterogeneous	Legacy IT systems

1 INTRODUCTION

With the rapid development of IT industry nowadays, many companies already have constructed their own IT systems for different customer demands. Although these IT systems could already be a complete business strategy. In the ever-changing business environment, customer's demands is constantly changing, too. So many software products' lifetime duration has become shorter and shorter. A new business operation mode is proposed. However, this new operating mode needs to be supported by business processing systems, and these new procedures may need more than one department's inner cooperation, sometimes it probably needs cross-department or even cross-enterprise cooperation. So a business procedure that can not only satisfy the customer's demand but can also flexibly responds to the constantly changes may become the key point of the enterprises competitiveness.

On the other hand, IT system techniques and IT architecture have experienced a changing process, from the traditional C/S (Client/Server) architecture, to the rise of B/S (Browser/Server) architecture in the 90s and its wide use in nowadays, and finally to the basic service architecture that people can gradually accept today. Also, the programming technique is constantly developing, too from basic languages such as machine languages, programming languages, to the Procedure-Oriented programming languages such as C, and Object-Oriented Languages such as Java, and finally developing into modular distributed enterprise level programming architecture such as J2EE, .Net. We can observe that the programming granularity is becoming larger, the IT system's layering concept is more generally accepted. Therefore, the developers only need to focus on the logic development and this saves much developing time and it is becoming a trend right now [1].

The changes of business mode and the development trends of IT technology need to be supported by a particular IT strategy architecture. SOA is proposed exactly under these backgrounds and commands. It's rapidly developed for having characteristics of re-usable, measurable and exploitable.

As the name implies, SOA is Service-Oriented structured, or rather saying it's an IT system architecture, constructed based on services. Its core of SOA architecture is service, and modeling in business based on the coarse-grained service and it makes the whole business and system structure become more concise. A service based

implementation can make IT system have more flexible respond way and more easily to be reused [2].

SOA is a loose-coupling software architecture. It also includes methodology related to running environment, programming model and architecture style, and it covers the entire life circle of service. SOA can give priority to IT business requirement for having the features of platform-regardless, protocol-regardless and specific-programming-technique-regardless. These techniques can fully responding to business requirements in order to reflect the value of this technique [3].

2 SOA BASICS

This chapter introduces the SOA concept, features, protocols, Web Service security protocols and security model.

2.1 SOA Outline

2.1.1 Definition of SOA

What is SOA? As the name “Service Oriented Architecture” implied, it’s a service constructed based on enterprise IT architecture. SOA is first proposed by the famous IT analyze organization Gartner Group in 1996, later in 2006, Gartner Group put forward the idea that SOA is an important research project in modern application development. Gartner first describes SOA as, a design method of Client/Server software. An application consists of both software service and software service user. Meanwhile, software includes software service and software service consumers. The differences between SOA and most universal Client/Server model is that SOA emphasis on the loose-coupling feature of software component, and it’s with independent standard interface. However, because of the limitation of technology environment at that time, SOA system architecture technology has existed for over 20 years but never been widely used. But with the acceptance of the appearance of Web service technique, SOA time has finally arrived [4].

At present, the definition of SOA has various versions, service-architecture.com has given it a description that it’s essentially an assembly of services. Communication between services could be simple data transmission, or it could be two or more services joined in some coordinate activities. Services needs some methods to be connected to each other. And the services itself means precise definition, perfect packaging and independent from other function of environment and state [5].

Looselycoupled.com has defined SOA as: System for connection resources as needed. In SOA rule, resources can be an independent services access through the standard way and being provided with other members in the network. Compared to the traditional system structure, SOA has set a more flexible loose-coupling relationship between resources [6].

IBM has defined SOA as: An IT system architecture style which supports convert business into a mutually connected services or a reusable business mission and these services and missions can be access through network when needed. SOA system architecture allows reconstruction for reuse and working with loose-coupling mode. SOA is system architecture that truly supports reusable component or assembly business procedure, these components or services can independent from application or their running environment [7].

SOA has many definitions, but basically it can be divided into two categories: One of them says that SOA is mainly an architectural style; other one says that SOA is a set of new distributed software system architectural method and environment which includes running environment, programming model, architectural style and related methodology. It covers the entire lifetime of service: modeling, developing, deploying, integrating, running and managing. The latter covers a bigger range, and is more suitable for future development. So this thesis is more inclines to the definition of the latter.

In this thesis, SOA is described as a software architecture based on a distributive system structure. This architecture has used the ideas of service oriented design, it divides application systems according to business demands to provide services. Different component is connected by defining interfaces and protocols. Interfaces us used to describe the content that service provides. It can be defined by a neutral manner. The implementation of service and the exchange between services doesn't rely on any specific hardware platform, operating system or communication protocol. Different service component has the feature of loose-coupling and reusable. From the aspect of software system architecture, SOA's idea covers the entire lifetime of software engineering: Modeling, designing, programming, integrating and deploying. It provides the guiding ideology and a complete set of methodology and system theory.

One of the important point of view that SOA architecture has brought is business generate IT, which means that IT and business is aligns more closely. Modeling up the business based on the coarse-grained business services and produces more concise business and system view, using services to make the IT system more flexible, more focus on reuse ability and faster and better reaction in facing changes. It also provides the business mode and IT correlation implementation a better trace ability through the explicit definition, description, implementation and coarse-granularity in business management level. In this way, the distances between them is reduced, and business changes is easier to transfer to IT.

In recent years, some IT organizations has successfully established and implemented SOA application. Companies such as IBM, BEA, Oracle has saw the inside value of this and all published their own SOA solution one after another. SOA is surly becoming a practical method of software architecture and engineering with an absolute advantage.

2.1.2 The Service Feature

SOA is a Service-Oriented Architecture, service is one of the component that constructs SOA, and the feature of services in SOA are as following [8-10]:

(1) Service can be operated independently. Service can provide external operations to realize its function. These operation is published and encapsulated through standard methods. The implementation of these function can independent form other services or component and can be easily invoked independently.

(2) Service using the standard description format to define the message format and available operations, it is self-descriptive, so the service publisher and users don't have to pay extra attention to some less important information such as implementing technique and addresses.

(3) Service has the characteristics of heterogeneous and loose-coupling. Its publisher and user can using the distributed deployment method to implement making services running on different platforms through different technique.

(4) Service can be used in combo. Some corresponding service combine techniques can be used to implement this. For example, service scheduling and process engine technology at present can combine a set of simple services into a complex services, that is to say service has very high flexibility.

(5) Service is dynamic. A published service can be dynamical discovered and bound.

(6) Service is standard and open. This promises different departments and different enterprises can call and dynamical combine this services together in using of business procedure.

(7) Service can guarantee the legacy system function, and it makes the service area more extensive. By reusing the enterprises' legacy IT resources, the investment in redeveloping the application would be saved a lot.

(8) Service has quality assurance. It's the property of service QoS.

Web service is a branch of service. SOA is an option form one of the implementing technique. Web service is an already existing mature service technology standard. In next section, the basic specification of web service will be introduced.

2.2 Web Service Basic Specification

Web Service has three basic specification: SOAP, WSDL and UDDI. These three specification has supported Web Service development. Figure 1. Is the Web Service Architecture.

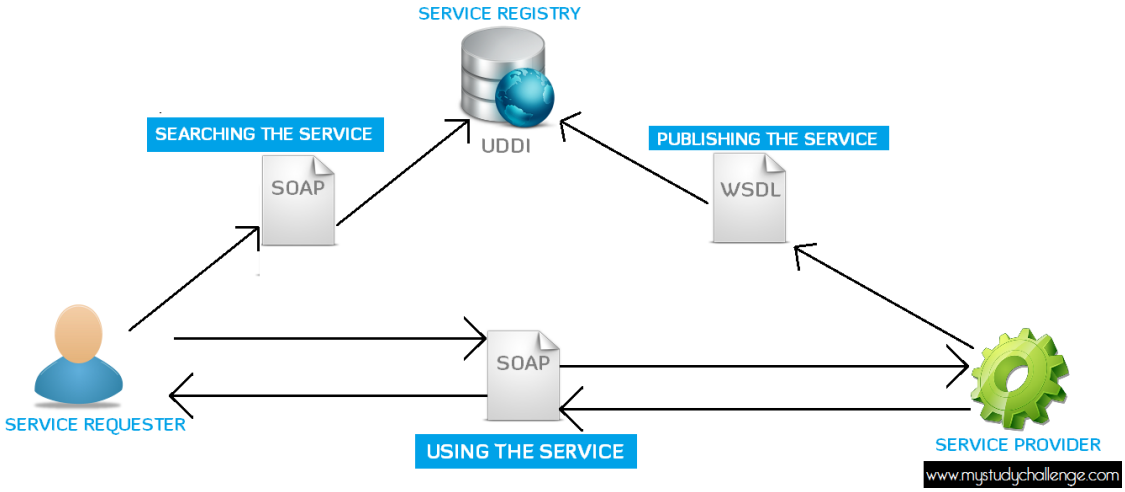

Figure 1. Web Service Architecture(reference from www.mystudychallenge.com)

Web Service system architecture is consists of service requester, service provider and service registry center. Between them exists three operation: publish, search and binding.

(1) Service Requester: When service requester needs to require a service, it will launch service inquiries to the registration center.

(2) Service Provider: Service usually have well defined interfaces (bot internal and external), and the definition of interfaces is usually platform and language neutral. The

descriptive information is usually published to Service Content (also called Service Registry), in order to be dynamical discovered and invoked.

(3) Service Registry Center: Service Registry Center provides service content function. It's the bridge of communication service require person and provider. It contains a service-available storage place, and allows interested service user to lookup provider's service interfaces.

Meanwhile, three kinds of operation is defined for between this three:

(1) Publishing: In order to make the service accessible, service description is required to make inquirer be able to looking up them.

(2) Searching: In searching operation, service requester directly retrieval search the service description or to search the required service type in the service registry center.

(3) Binding: In the binding operation, the service requester analysis the service binding information from registered server, for example, service access path, parameters of service call, return result, transport protocol, safety requirement and so on. By configuring the system to make remote calling services from the service provider.

Next, there will be three separate introduction about basic specification of the Web Service: SOAP, WSDL and UDDI.

2.2.1 SOAP protocol

SOAP (Simple Object Access Protocol) [11] is a lightweight simple communicating protocol based on XML. It defines a message format for transmit XML messages through network. This kind of message is consists of a SOAP envelope element, an optional SOAP header element surrounded by an envelope element and a SOAP body element. SOAP is born to solving the internet interoperability issue. Among these, HTTP is used for SOAP message transmitting and XML is an encoding mode of SOAP. Figure 2. is the composition diagram of collaboration between each component.

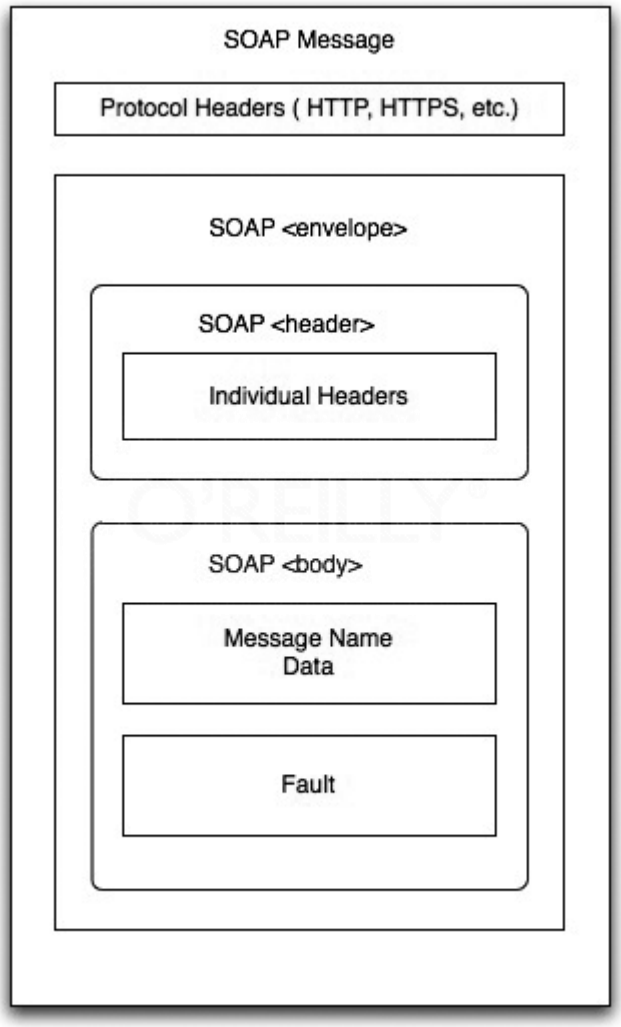

Figure 2. Message Composition Structure Chart

SOAP has defined a communicating protocol between requester and provider. In this way, under the object-oriented environment, the requester object executes a remote calling method for provider object's use. The advantages of SOAP is that it is manufacture-regardless, it is implemented independently from platform, operating system, target model and programming language.

SOAP has provided a lightweight mechanism for structured type information exchanging with using XML in an incompact and distributed environment. SOA specification is consists of four parts.

(1) SOAP encapsulation: Encapsulation has defined the content of a descriptive information, who send the messages and who needs to do the processing and acceptance work, along with how to deal with their framework.

(2) SOAP Coding rule: Defines the mechanism of exchanging application and data type example.

(3) SOAP RPC representation: Defines the agreement of representing remote procedure calls and responses.

(4) SOAP Binding: Defines a convention of using bottom transport protocol to implement exchanging SOAP envelopes in the node.

2.2.2 WSDL Protocol

The full name of WSDL is Web Services Descript ion Language [12]. It's a standard language used for describe Web Service based on XML. The goal is for the provider of Web Services to use all their services content, for example, the transmission mode, service mode interface, interface parameter and service path, to generate corresponding documents and send them to user. User may create corresponding request messages to service provider through this WSDL document. After the service is completed, service requester will analysis the returned service result messages of WSDL document and turn it into content that they are familiar with. WSDL1.1 got certificated in 15th, Mar, 2001. And officially become a standard. WSDL document is consists of the following parts:

(1) Type: Type definition. Defines all the data type set that is used by the Web Services and can be quoted by the Message Components.

(2) Message: Message Definition. Defines all the Web Services request messages, responding messages.

(3) Port Type: Interface Definition. This part describes the Web Services' interface definition. It can be regard as abstract interface similar with JAVA language.

(4) Biding: Contains of how to convert abstract interface into specific details that is the combination of specific data format and protocol: the combination of interfaces and related service access point.

2.2.3 UDDI Protocol

WSDL has described related information of visiting specific Web Services. But in the internet or between different departments of enterprise, how do us able to discovery the Web Service that is needed. People needs a simple and fast way to search all the possible trading partner and build convenient contact and system docking. But those Web Service provider needs a way to publishing the Web Service developed by their own, and spread it. So Universal Description, Discovery and Integration has born. UDDI is a cross-industry, cross-platform open architecture. It helps Web Service provider to publish service information on the internet.

The UDDI concept is first put forwarded by companies such as IBM, Microsoft, Ariba and so on, and is submitted to OASIS organization to get standardized. From 2000 until now, it has developed into the third version. In the beginning, UDDI Standard has got great support. IBM, Microsoft, SAP, NTT and many other companies has built a public oriented registry center. But UDDI development didn't becoming like people's expectation as becoming a basic core technique under the Internet free developing trading environment. At present, UDDI is still an important standard in Service Discovery Area. The research of UDDI is very meaningful for understanding new technologies [13].

2.3 Web Service Security

2.3.1 Introduction of Web Security Protocol

For the Web Service application, safety is a very important part. OASIS organization has defined the Web Service Security (WSS) protocol since 2002. The latest version of WSS protocol is 1.1.

WSS cannot solve all the Web Service Security problems. It must working with other security protocol to make sure the effectiveness of safe system. For example, WSS and SSL can achieve end to end transmission, and it's safe in both SOAP message layer and network connection layer. Implementation of WSS and PKI can make sure the third party certification is effective. WSS and Kerberos can make the single-point Web Service cross-area single sign on (SSO) became possible. Besides, in WS-I basic profile, exists some other protocol, for example, WS-Policy, WS-Address, an assembly of them can providing a broader security mechanism. Within the industry, IBM,

VeriSign and Microsoft have provide a solution model of Web Service Security. Described as below:

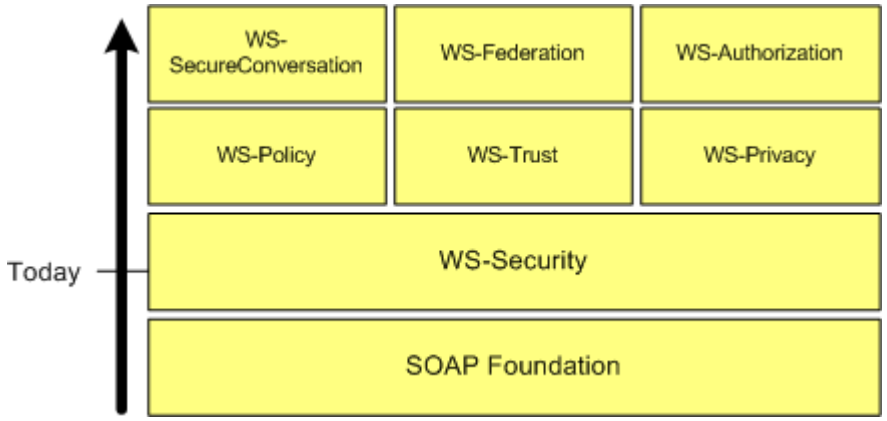

Figure 3. Web Service Security Mode Protocol Composition Chart

The entire security model is based on WSS and SOAP, down below is the introduction of Web Service protocol set's Service Security model:

(1) WS-Policy: Defines the Grammar and Semantics for service requester and service provider, in order tho describe their demand, first option and performance. Grammar expresses in the form of strategy has provided a flexible and simple method for demands in every field.

(2) WS-Trust: Being used for managing trust and build a trust relationship between different Web Service participator. It provides some expansion for WSS specification, and specially process the publication and verification of related security tokens. Assured all the operation between different providers are under a safety and trusted safe data exchanging environment [14].

(3) WS-Privacy: Described the message security model between service requester and provider.

(4) WS-Secure Conversation: Providing identify and security context switching between different participants.

(5) WS-Federation: Providing distribute and manage "trust" model to multiple heterogeneous federated system.

(6) WS-Authorization: Providing management authorization information and strategy.

These protocols includes WSS protocols that can be used in combination, in order to adapt to particular demand scenario. Some common scene includes as follow [15-16]:

(1) Different Web Service message provider needs to be isolated, to make them able to consume the same encrypted information.

(2) A scenario needs message intermediary to process.

(3) The platform for transmit SOAP messages is not HTTP.

(4) Multiple user login information is needed to attach to message.

2.3.2 Secure SOAP message based on WS-Security

In the aspect of Web Service Security, a common way is using SOAP message to deliver user authentication information, message integrity information and confidential information. Figure 4. Has describe how to implement SOAP security messages using WS-Security.

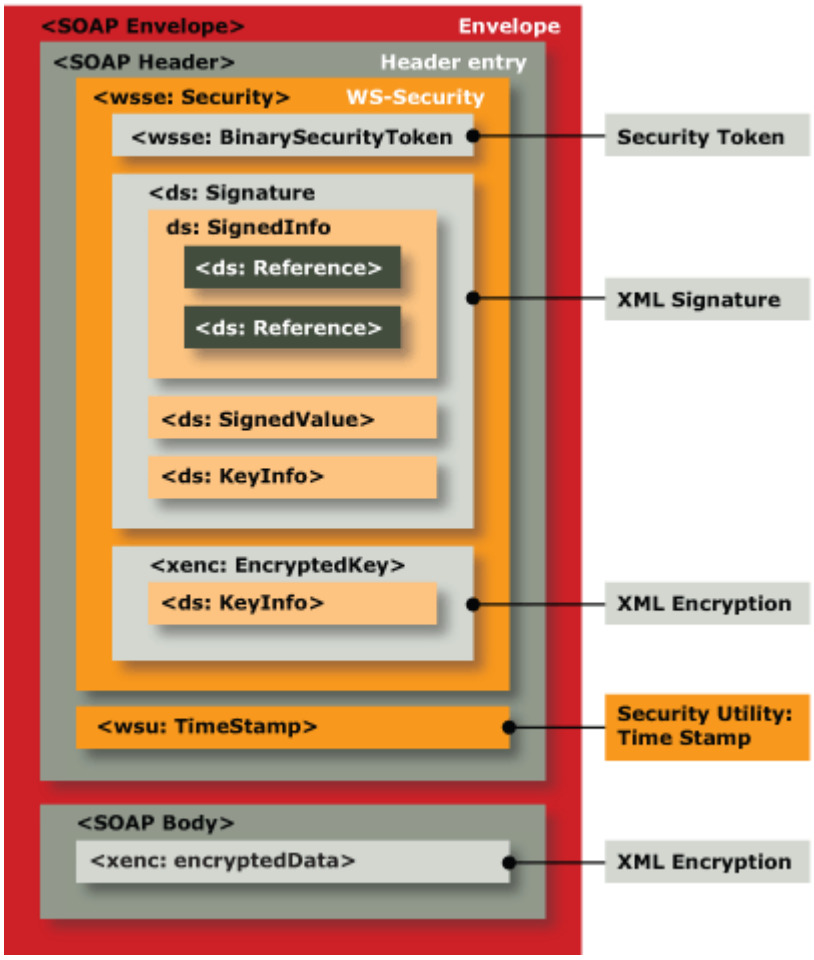

Figure 4. Implementation of SOAP Security Message based on WS-Security

For the safety SOAP message, it consists of two parts: the security related part in SOAP headers and the encrypted information in SOAP message body. The SOAP header's security related part, security token, time stamp, signature and cryptographic key can be stored.

Security authentication control: Able to adding WSSE label in SOAP header messages. The UsernameToken label under WSSE label contains username and password. After server receiving SOAP messages, messages will then be processed. The authentication method of username and password is a basic method of passing user authentication through WS-Security. There are some other authentication methods, for example, digital signature, LTPA token and user expanding token [17-19].

Integrity Control: Integrity means information is safe, no modification and loss under security control. In reality, security digest and signature can be used to assure messages' integrity. In the WSSE label of SOAP header, there are user authentication

certificate, signature information and algorithm result. Message receiver confirm the signature information through user certificate and algorithm after received the messages, legal message body will then be processed.

Confidentiality: SOAP messages security mechanism is implemented by encrypt SOAP message body. Only those message receiver who has the corresponding decryption key can see the decrypted information.

2.3.3 Axis2 Web Service Security

Axis2 is a Web Service engine in the Apache open source organization, it is implemented based on the latest SOAP specification and SOAP with Attachment specification of Java language. Many popular developing tool also using Axis to implement the Web Service function, including the SCA open source optimism Tuscany. Axis2 has features of modular and extensible: From one aspect, Axis2 can do the Web Service security management through the handler class programmed by the user them-self. On the other hand, Axis2 has introduced Apache Tuscany module, in order to satisfy the web service security requirement.

Apache Rampart is a module implemented by WS-Security based on Apache Wss4j. Rampart using standard WS-Security Policy to declare. Or they can using their own declaration method. Rampart provided user identification, time-stamp, system encryption and signature to ensure the Web Service security guarantee such as integrity and authenticity of information.

Axis2 is consists of Out-Flow and In-Flow. When Axis2 receiving messages, In-Flow will be invoked in use. When Axis2 sending messages, Out-Flow will be invoked in use. Rampart engine processing In-Flow and Out-Flow by two class. When processing In-Flow, Rampart using two processor to process received SOAP messages, one of them is Security Header Processor, for process the header part of received SOAP messages; the other one is TokenProcessor, used for process binary security token; For the Out-Flow, Rampart engine using two builders to construct sent SOAP messages, one of them is Security Header Builder, for construct sent SOAP messages' header; The other one is Token Builder, for construct binary security token. These Builders, Processors and component such as Encryption, Signature, and Username Token and so on, all have SOAP message security processing [20-22]. Figure 5 is the Entity and Function

chart of Rampart:

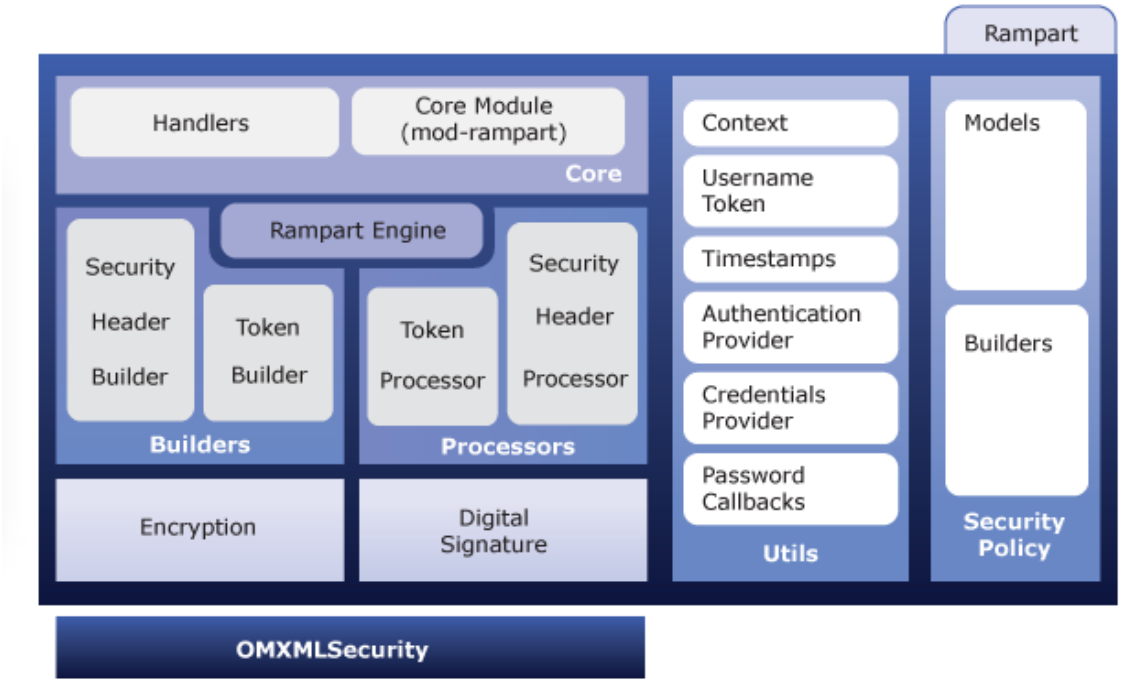

Figure 5. Rampart utilities and function Chart

3 SOA PROGRAMMING MODEL SCA

This chapter will introduce the origin of SOA programming model SCA, the assembly of SCA and some other technique and SCA strategy framework, especially content related to security strategy, and have introduced the Apache Tuscany, the Java open source implement of SCA.

3.1 The Origin of SCA

As a service integrated technique, SCA's performance is not abrupt; from the rise of Web Service, to the short-lived Web Service Invocation Framework (WSIF), and finally to the publishing of SCA, can help us clearly see the developing outline of service integrated technique: emphasizing coarse particle size combination and loose-coupling.

Web Service has used WSDL to unify the service interface descriptions, but Web Service lacking of a uniform invocation model. Uniform invocation includes uniformly

invoke the Web Service and other kinds of services. For example, Bean and JMS service, IBM has proposed an invocation framework WSIF for Web Service based on this requirement. The purpose of putting forward this Web Service Invocation Framework is to solve the invoking problem of Web Service. Web Service invocation framework thinks that the description of service interfaces (service operating, input, output, and bug) belongs to the category of business logic, which should be cared by the service user. But Web Service location and specific transport protocol belongs to the category of technique. WSIF always supports the thoughts of separate the logic and technique from the beginning. And with the business and technique becoming more complex, WSIF has faced the reality and separate them only from the framework layer. Business staff can only cares about the business task, technical person can emphasis on the realization of technology. Web Service Framework provider has abstract the location and transport protocol of the service as provider. Technical person responsible for the implementation of the provider, but business person only needs to use the service interface description. After Web Service Invocation Framework has implemented the Web Service provider, it also implemented provider with the local protocol such as Java, EJB, JMS and JCA. In this way, local service such as Java, EJB, JMS and JCA or any other kinds of services can have a uniform invoking call with the Web Service invocation framework. This unified invocation method is called Web Service Invoking [23].

Web Service Invocation Framework was first proposed and implemented in IBM. IBM has donated WSIF to Apache open source organization. Right now, WSIF is listed in the Apache Web Services' sub-project. But due to many reasons, Web Services didn't really get popular among the industry, but there are many concept in this Web Service Framework is worth advocating [24]. It implemented separate business logic with technique, unified different service invoke for Web Service' use, and it achieved reused the current services and provided convenience for the development and tests for the Java local provider. These are great improvement. But WSIF didn't provide a decent data model, nor a service construct supports. On the basis of these past experiences, SCA was finally proposed. From the technical aspects, SCA is the continuing and expanding of Web Service Invocation Framework.

On March of 2007, Open SOA Corporation Organization has published the standard version 1.0. And has submitted OASIS for authentication.

3.2 SCA Brief Introduction

Service Component Architecture, SCA, is a component programming architecture used for service invoking, construct and implementing-language-regardless. It provides a programming model for construct application and solution based SOA. It based on the idea of change the business function into a set of services for providing and combine these service sets to make a solution in order to meet some specific business demands. These combined application includes some specific services created for application use and also includes the reuse for existed application system function [25].

SCA provides a uniformed invoking methods of uniformly invoking different kinds of services for example POJO, EJB, BPEL, JMS and WEB Services. This idea is from WSIF.

SCA also provides a construction model based on component, in order to uniformly construct some other different kinds of services, for example, EJB, JMS and Web Service. It has compensate the defects of WSIF and Web Service.

SCA doesn't have its own data model, but its together-born partner Service Date Objects has provided a data model used for services.

From all these above, SCA has provided some enterprise business budget oriented Quality of Service, QoS, and ability.

All these features of SCA has make the enterprises obtain a better layering architecture and can separate business logic and technical logic pretty well, in order to make application more easier to construct, deploy and adjust. These features are exactly oriented to service architecture demand. In below, there will be some introduction of main part of SCA specification [26].

1. Service Component

Service component is a business logical unit, it is also a very basic element and structural component unit of SCA application. Component is the service provider, and could also be the service consumer. In below is the Component Structure Chart.

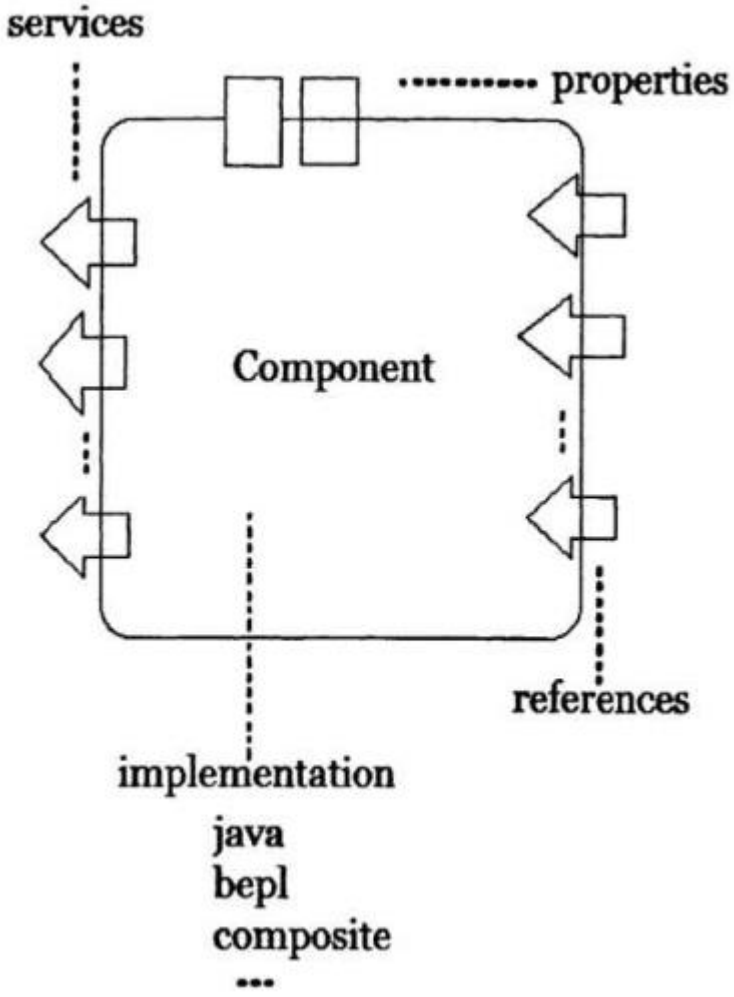

Figure 6. Component Structure Chart

A SCA Service Component contains four parts:

(1) Service, represents the business function provided from local component to other component. A service component is able to provide various kinds of service function to the external through interfaces. Right now, the interface format of that SCA has defined is Java interface and WSDL port type. And a Service accessing methods is described in binding, for example, Web Service binding can be used to make Web Service invocation of specific service.

(2) Properties, this is a data value that can effects the business function. Within one component, its property value can be defined, one components can have many different property value. Property type can be various, it can be both simple and complicated at the same time.

(3) References, the implement of component rely on the service provided form other components, and is marking with quotes. One components can quotes many other services that components provided.

(4) Implementation, implementation is the main body of realize the business function. SCA allows several of implementing methods, such as some traditional programming languages, Java, C++, BPEL, as well as some scripting language PHP, JavaScript, and some dynamic language such as Groovy, Ruby, beside that is some declarative language such as XQuery, SQL. Also, it can be some specific framework or some runtime environment such as Spring, EJB. Each specific type represents specific implementing methods. Component Service is programmed based on service interface. In this way, if we have implement the interfaces' business methods, then some more extra amount of programming language and environment can be put in use in this case.

2. Composite

SCA defines the content and connection in assembling application as Composite. It described the content within one SCA and connection between different SCA. Or as, how to assembly component and how to combine different component and makes it into a complete solution.

One composite is consists of the many different kinds of components. The main purpose of composite is to grouping the components through different kinds of logical methods and connect these components inside the group with wires, even though they are implemented through different technique. We can regard composite as one of the practical methods of implementing components, in order to construct more complex composites. Composite promotes internal components' service, quote and property to implement its own service, quote and property. Composite is also provided with service, the interfaces that is currently supported is Java and WSDL interfaces type. The way that composite is provide to the external is binding. Binding one service means from external that one specific composites can be invoked through the binding. The binding methods that is currently supported is JMS, JCA and SCA protocols [27].

3. Binding

Same with the interface, binding is also a very important concept that is used in components' service and quoting. Binding is a methods of accessing services. Binding in services represents the accessing mechanism when service client invoking other services, and the references in services represents the accessing mechanism when service client invoking other references.

SCA supports different kinds of binding methods: Such as SCA Service, Web Service, EIS Service, and Stateless Conversation Bean. More binding type can be supported by the extending mechanism of SCA, such as RMI and CORBA. Binding is defined through services or binding element in references. Figure 7. shows the Component Binding Schematic Diagram:

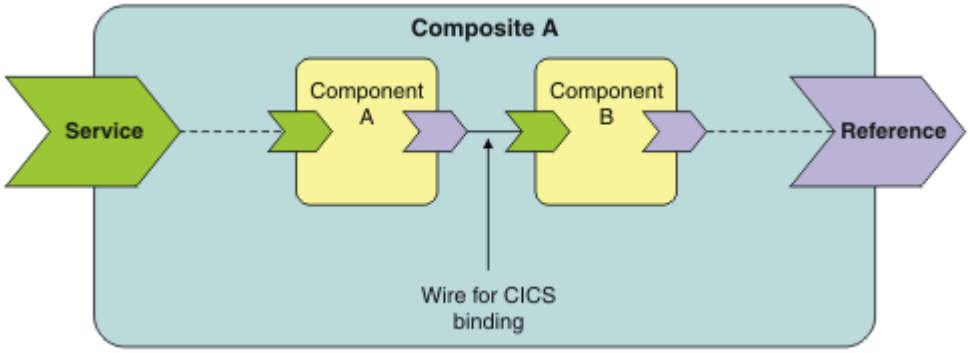

Figure 7. Component Binding Schematic Diagram

4. Domain

SCA domain is series of services, see Figure 8, that provides a business function of a specific organization control. Every domain in SCA is divided for one business function. Every domain is mapping to a business range, this range can be a sub system or a module. Each domain is marked with a URI. The Component A is implemented by the Composite A, meanwhile, Component A is one of the part that consists Composite Y. In this way, an Embedded Layering Structure of "Composite within Composite" is formed. That's exactly what a normal structure of business procedure is, a procedure usually combined of various service oriented atom services, and every atom services is usually constructed of a series of basic function services.

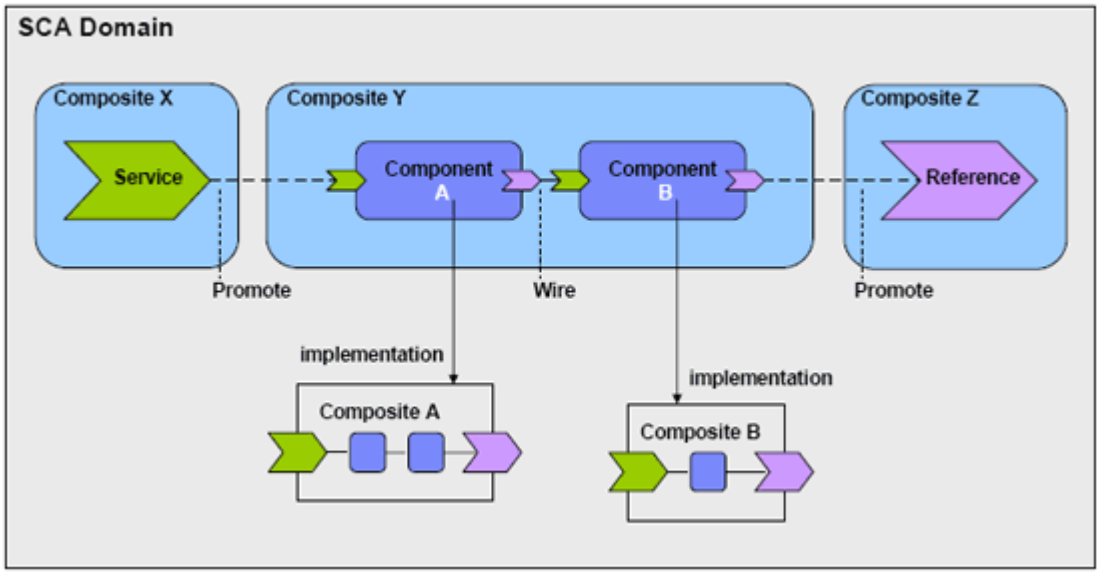

Figure 8. The SCA Domain

3.3 The Feature of SCA

According to the description of SCA to SCA's model and main component, we can see that SCA has the following features [28-31]:

(1) Loose-Coupling

The Definition, Implementation and Using in SCA is separated form each other. Components and composites provide services through interfaces. In this way, the service caller only needs to know the specific details of interfaces and do not have to care about the specific implementing details of interfaces' component or composite. The caller can choose suitable methods in component and composite implementation. And the adjustment to future implementing technique has no effect to component and composite caller. Besides, the services that component and composite provided can have different binding methods, or as providing different connecting methods to client. This kind of Loos-Coupling structure can make SCA is more flexibly used in extension, and can adapt to the ever-changing realism, to implement the demanding adjusting and extending of application component.

(2) Heterogeneous

SCA supports widely use of implement and connecting methods in services. In the

aspect of SCA implementation, SCA supports basic popular programming languages, and also supports the framework and environment for these programming languages. In service connecting part, SCA component running accepts various widely used communicating and service accessing technique, for example, Web Service, Messaging System, Remote Invoking Procedure RPC.

(3) Reuse

As the construction units of SCA application, the using method of SCA component and composites can be regard as building blocks, component a composite can be regards as “wooden blocks”, they can be used in different building methods in order to meet different application demands. Of Course, as a non-SCA system, it can also joining in the SCA system through using a implementing method of being a SCA component services and becoming a “wooden blocks” like SCA component and composites, or can be connected to SCA systems and becoming a service provider or consumer.

SCA is a simplified new technique in implementing SOA application program development. It’s not an alternative technique for any other programs, it’s born for simplified the operation between techniques and make new system easier to be created or to make the current IT property published as reusable service. All of these are with the same purpose that SOA has proposed.

3.4 SCA and Spring

Spring is an open-source framework, created by Rod Johnson. It was born for simplified enterprise’s application. Spring using the basic JavaBean to accomplished functions that only EJB used to have. Spring can not only be used in Server Development, it also makes any Java application benefits from its Simple, Testable and Loose-Coupling features.

To put it in a simple way, Spring is a container based on IoC (Inversion of Control) and AOP (Aspect Oriented Programming) and a lightweight framework. Spring has promote the loose-coupling through IoC technique. An object that is relied on other object can be passive transferred in, instead of creating an object or searching for the relying object by the original object itself, cause this searching method can be regarded as opposite to JNDI. Spring has provided rich technical support of AOP for separate business logic and other service of system-level in enterprises’ development. That is,

the application implementing object can only care about completing the business object and not responsible for other points in system such as log and business support.

Spring is Interface-Oriented-Programmed, so do SCA. But to put it in a more specific way, SCA is more likely to be Service-Oriented-Programmed. Interface can be used to describe the function method, parameter of method, return value and abnormal info that service can provide, but it can't describe the providing way. In SCA technique specifications, the providing way of component service is binding, if there's no binding method assigned, then the default binding method of SCA is SCA Binding. From that we can know, between SCA components are references of Service Oriented and only when using the default binding method, SCA binding, it can be similar to Spring interface oriented programming method. From this aspects, interface oriented method is the simplest form of service-oriented-programming.

Spring use dependency injection method to uniformly use currently existing technologies.

While the method of using other techniques, SCA and Spring are quite similar. SCA open-source implementation Tuscany also used dependency injection, in order to provide an architecture, and using other techniques in SCA methods. Compared to Spring, SCA also using service-bound method in services to construct other techniques, in order to make SCA component services available for other existing techniques. From all these above, SCA technical specification can be regards as more mild, and can be more harmonious in using other techniques.

One of the most important way that SCA using other techniques is implementing. Other techniques can become one of the SCA component by implementation. When SCA meets Spring, one of the effective way is using Spring to construct coarse-grain service component to implement, and should be referenced in SCA in order to open service, relate service component and processing heterogeneous and distributed system. Then the Spring container can be used to implement business services and log and business dealing operation. SCA can add some useful function in application implemented by Spring. For example, extending supports to remote component and multiple-component. It supports using various kinds of programming languages which is not supported by JVM to programming some component, it also supports asynchronous programming mode and strategies designed for the activities and securities by WS-

Policy. The second method of combining Spring and SCA is using SCA in Spring, in order to solve this problem, SCA has provides Spring extended marks. There are three elements can be used in Spring [32-33]:

(1) <sca:service>: Which provides a way of controlling those Spring bean to provide service to SCA. SCA is responsible for create suitable server binding when ts running, and applying these required strategy to services according to SCDL settings.

(2) <sca:reference>: Provided a declare methods of dependent relationship between Spring application context and compounded set of other SCA components.

(3)<sca:property>: Declared the dependent relationship between the Spring application context and settable properties provided by SCA implementation. The core container of Tuscany is also a sample of IoC design. The Spring implementing supports is also provided in Tuscany.

In the implementation of Apache Tuscany SCA, SCA used Spring as implementing techniques in SCA compound sets, and is able to use <implementation.spring location/>label to implement.

Meanwhile, there are three ways of presenting designated URI through location property:

(1) Designate the Spring context files, and can be used in system implementation;

(2) Designate content, which contains all Spring-related context files.

(3) Designate archived file, these files in jar format contains all the context that is Spring related.

3.5 SCA Strategy Framework

3.5.1 The brief Introduction of SCA Framework

In the declaration cycle of component and composite, the acquirement of non-functional requirement is an important part in service component definition. From the design to the specific deployment of component, SCA has provided a framework to support the restrain, function, QoS expectation specification and WS-Policy Attachment and some other strategy language to designate strategy and strategy theme that is

relate to SCA component. These strategy can be used in SCA binding and implementation. SCA strategy framework model is consists of policy intent and policy sets.

Strategy intends to describe a service quality requirement from entity, and has provided a method for state the service quality in a senior abstract way from developer and integrated personnel, to choose those binding and specific strategy set to meet these requirements, the binding can be used while developing and assembling or designate later-stage binding method when deploying. SCA prefer this kind of later-stage binding method for it can promotes the possibility of the reuse of component. SCA component in different applying methods may needs different binding and specific strategy. Bindings and strategies decided in the early-stage may limits the reuse ability and using component in new applying environment. For example, when a component published services has an “authenticated” strategy intend, client needs to provide authentication information when trying to reference this component service, but it didn’t provide how to get the authentication. In that way, when the component is deployed, the biding is already settled. For example, when binding SOAP, HTTP, the deploy person can freely choose WS-Security and single-to-multiple authentication technology.

Strategy Intend can be used to represent two requirements: interaction policies and implementation policies. Interaction Policies usually means the specifications used in services and references. It’s used in the communication between service consumer and service provider. The ”authentication” mentioned above is an example of representing interaction policies. It can also be used in representing the implementing strategy in SCA component’s implement. These interaction policies has proposed when SCA container is running component, they must provide QoS related requirements, for example, “Component can only be running in an object.” [34-35].

Strategy Set is consists of policy. A policySet element is used to define a specific strategy set, these strategy is applied in some binding type or implementing type. The policySet element are as below:

(1) @name attributes has declared the name of policySet.

(2) @appliesTo attributes is used to determine the SCA structure which can be set by this strategy set. The attribute content must match the XPath 1.0 expression.

(3) @provides property, its value is an intent name list, and designate policySet to provide intend. The list member is separate by blank space.

It contains one or more elements in below:

(1) intendMap element

(2) policySetReference element

(3) wsp: policyAttachment element

(4) wsp: policy element

(5) wsp: policyReference element

(6) xs: any extensibility element

Any amount of element type in above may contains the sub element which is regarded as policySet element, as well as extending elements. This feature is quite similar to specific binding technique and different strategy language of domain. In order to make one policySet contains any strategy languages, extending elements may from any namespace and can be mixed. After then, SCA strategy expects when expressing interaction strategy, WS-policy could be a public strategy language, especially when Web Service is bounded.

3.5.2 SCA Security Strategy

SCA security model has provided flexible mechanism for developers in order to define the required security protecting level in order to meet the business demands. Meanwhile, developers do not need to be familiar with specific implementation of security mechanism. In security area, interaction strategy is used to solve the confidential requirement between service consumer and service provider. Using implementing strategy is a safety constraint in managing service implementation, usually used in component.

SCA security interacting strategy can be designate by using strategy intend and strategy sets. Strategy intend chart represents an abstract and safety service quality demand, and is specific-security-protocol-regardless. However, strategy set has defined a specific strategy, which is always related to specific security protocol. SCA

security specification has defined the following strategy intend, and is used in describing interact strategy's authentication, confidentiality and Integrity.

(1) Authentication: Validation intend is used specify the use of a SCA service. Client must be used to identify itself. Usually, client basic security structure is responsible for server authentication, in order to avoid the attachment in the middle.

(2) Confidentiality: Confidential strategy is used for specify that the message content can only be read by the authenticated service consumer and service provider. A common way is to encrypt the message, or using other available methods.

(3) Integrity: Assembling intend is used to specify when message content won't be alternated when it is transmitted between sender and receiver. A common way is using digital signature in message.

All these three strategy intend above is also the standard core strategy intend of SCA definition, all of the SCA implementation supports these strategy intends. For example, the Java implement of SCA, Apache Tuscany has achieved these three strategy intend. Any one of these security intend can be used in advance for more specific business requirement. SCA security specification has defined two identifier: transport and message, both of them can be used in any of the intend above.

(1) Transport: the intend used in transport identifier's definition can be identified in transport layer of communication protocol.

(2) Message: the intend used in transport identifier's definition can be identified in message layer of communication protocol.

3.6 The Introduction of Apache Tuscany

Tuscany is the name of an Italian province in geography. The way of naming has also make accordance with how the Apache open-source organization name its software. The name is quite specific but it has no relationship with the software itself from its surface. Tuscany is a recommending open-source implementation of SCA specification. The first Java implementing version of Tuscany is completed in WPS of IBM and after its success. Many big brand such as IBM and BEA has proposed the SCA specification together. And has donated the original code to the open-source organization Apache.

At present, Tuscany has already done its job successfully under the environment of Apache, and has become a top-level project with the promoting of companies like IBM and Puyuan. Tuscany has provided two technical language implementation: Java and C++. It's divided into three parts:

(1) SCA open-source implementation, it achieves the service integration.

(2) SDO open-source implementation, it achieves the data integration.

(3) DAS, data accessing service, providing interface service from SDO to relational database.

Within this, the latest version of Tuscany Java implementation is Tuscany 1.6 and 2.0m4 version, and Tuscany 1.6 version will be used in this thesis. As a lightweight SCA framework, Tuscany has these following features:

(1) Tuscany is platform-regardless embedded framework, it can run on any PC platform, such as Tomcat, JBoss, WAS and Web Container, or can run under the J2SE environment.

(2) The core module of Tuscany has provided the SCA specification, API implementation. The SPI interface of Tuscany system, some basic system implementation (such as even, factory, storage), and a complete set of extension mechanism, these extension mechanism has provides the basis for Tuscany to integrate the services in different platform.

(3) The extension of Tuscany is completely Loos-Coupling. The framework itself has provided a big amount of extension implementation, the user can also extend the implement of Tuscany in their own system, and as long as they obey the Tuscany extension specification and API interface.

(4) Combined Apache Rampart to implement the SCA security strategy framework.

Tuscany is implementation of open-source SCA. It accomplish the SCA specification definition, and has some function extension in other aspects. The SCA specification and extension are as below:

(1) Implementation: SCA component implementing method. One SCA component can

be implemented by different kinds of languages or technique platform, for example: POJO, EJB, Spring Bean, bpel Procedure and many other scripting languages.

(2) Binding: It's the implementation of SCA Binding specification, SCA Service and Reference binding method and a SCA service can be bind and released as Web Service, Java RMI invoking, http resource and JMS message. One SCA reference can also call remote service through Web Service, RMI Invoke, http invoke and JMS invoke.

(3) Databinding: Data binding method, this is a concept proposed by Tuscany, it's always used in defining transmitting format of parameters in Binding. For example: Web Service Binding usually use XML format, SCA binding usually use SDO format, Jsonrpc Binding usually use Json format.

(4) Interface: It's the implementation of SCA Interface Specification, the interface exposing method of SCA Service and Reference usually are Java and WSDL type.

Axis2 is the Web Services' deploying framework in Tuscany. It implement the WS-Security specification by reference Rampart security module. And it also takes the processing class Handler to handle the Web Service Security. Tuscany implementation's security framework is a SOAP message level security module implemented by Axis2 and Rampart and has combined the SCA security definition and SCDL to implement a flexible security strategy framework.

4 THE IMPLEMENTATION OF HOTEL RESERVATION FUNCTION

At present, SCA has been widely recognize in IT industry. It has provides the technique of construct application system model for enterprise architecture SOA. In this chapter, the implementation of SCA in a travel advisory company will be mainly discussed as an application case.

4.1 System Requirement

The application case focuses on the integration and expanding of a travel company. The information systems of this company mainly include VIP customer information management system, hotel information system, travel advisory websites, etc. At present, all these systems has stable operation.

VIP Customer Information Management System is a web application created for VIP customer personal information, credits information management and SP text sending. Figure 9. is an example of presenting these basic system function. VIP customer management system gives out the VIP card to customer and confirm their identity by the correct password. VIP customer will have privileges and credits refund by making consumption in member companies related to catering, culture and entertainment industry. These member company also have rights to check the log-in system and sales info, as well as searching VIP customer's information and adding VIP card credits. The system can also sending sales info of Member Company to VIP customers by cooperate with a third party text messages service provider.

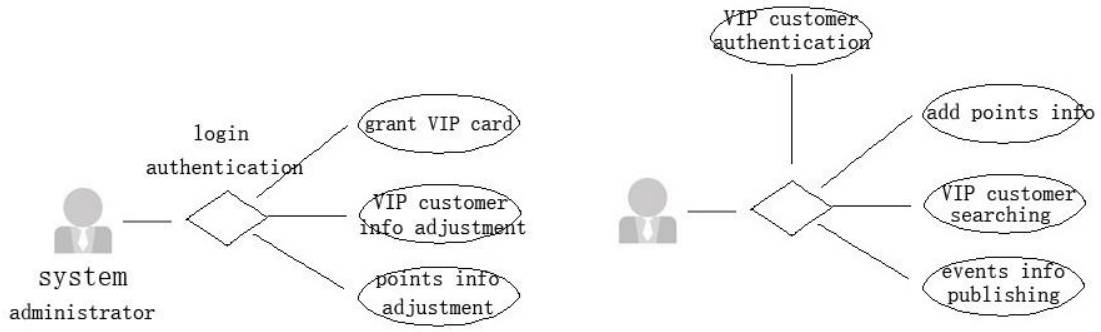

Figure 9. The Example of VIP Client Management System Basic Function.

The hotel information management system mostly provides hotel info around the world. For example, basic hotel room details, including room type, price etc. Transportation information such as the city bus, railway and highway are also included.

Under the fiercely competitive IT environment, the website needs to make improvement. For more efficient use of services, these kind of website can have a further expansion. These are the new added functions.

1) Providing hotel reservation function to VIP customer.

2) Improving the privilege of member hotel, to provide member hotel with user login interface to let them manage their reservation systems themselves. Inform member hotle the reservation result with text messages. VIP customer will receive VIP card points after they have successfully check in. The member hotel function are in the Figure 10 below:

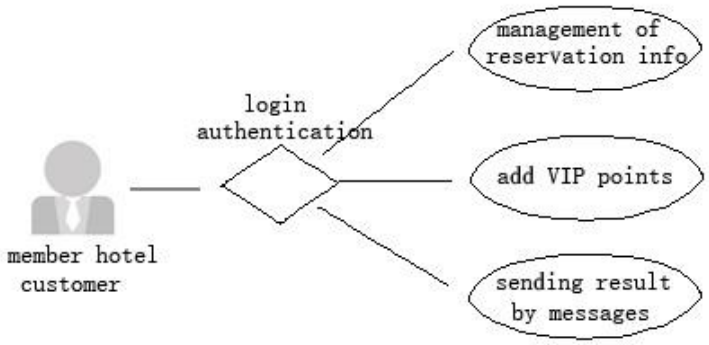

Figure 10. Hotel Membership Function Use Case Diagram

4.2 System Design

The travel advisory website had used the J2EE open source Struts, Spring and Hibernate as developing technique. According to implement this function, it needs the following function support:

1) VIP customer authentication and credits management function

2) Text messages sending function

3) Hotel room information advisory function

4) Hotel room reservation function

5) Reservation info management function

6) Hotel user management function

For the VIP customer authentication, credits info management function and text sending function will be used in implementing VIP customer information management system. However, during the development, there will be iterative development exists and these two different system will not have an excellent integration. The implementing technique of VIP customer management system are JSP, JavaBean. The main implementing methods are JSP script and JAVA function class, without the support of MVC layered. The implementing methods of two system are different and they couldn't

be combined as one. In this way, a distribution technique of making use in other system are also required. At the same time, the functions in VIP customer information management system needs to be expand based on SCA specification:

1) Assembly text messages sending platform to text messages sending component. Then binding and publishing them as web services and RMI mode. In order to using this component services in different technique situations.

2) Assembly VIP customer information management function as customer management component and then binding and publishing it as web services.

After implement these functions, travel advisory website can use these components in its platform, sending text messages and manage VIP customers' info.

The hotel information management system was developed in early days. PHP scripting language has been used. Since it is impossible to combine travel advisory website and hotel information management system as one, It is necessary using PHP SCA open source to binding these components and publish them as web services. In this way, travel advisory websites can put the hotel room search function component into use.

The travel advisory implementing methods are Struts, Spring and Hibernate in MVC structure. The system implements expanding of travel advisory platform in SOA structure based on SCA. It adds service layer to the original business layer and performance layer. Figure 12 below is the structure after expanding of travel advisory platform. In this way, the system functions components can be assembled and published for other system to use. Figure 11 is the room reserve component:

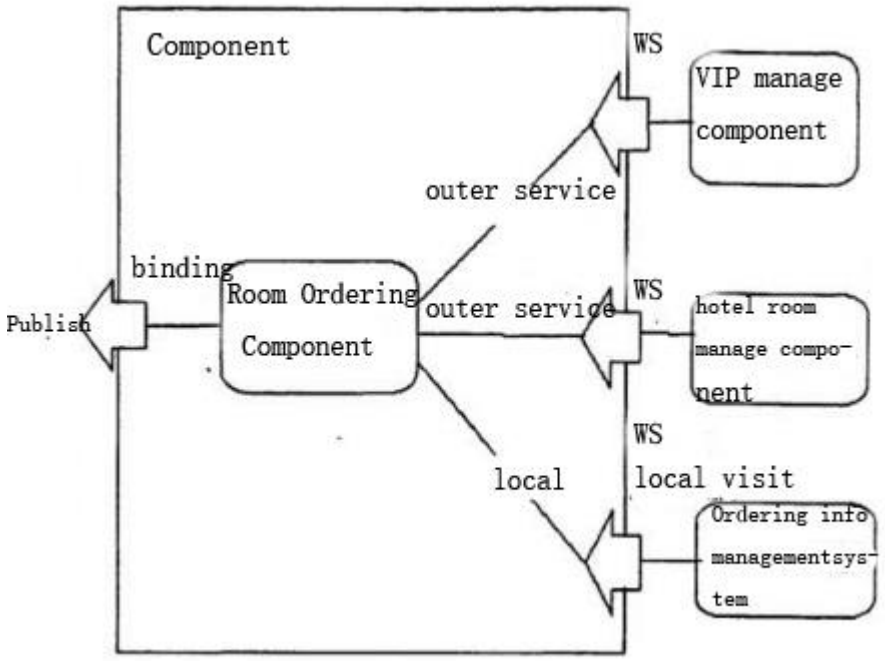

Figure 11. Room Schematic Component Diagram

4.3 System Implementation

The system implementing technique of SCA container had use J2EE technique while applied with Apache Tuscany 1.6. As for the implement of hotel management system on SCA container, PHP SCA implementation will be used. The travel advisory website will still using open source Struts, Spring and hibernate as new developing function model. Apache Tuscany 1.6 will be used in SCA container development, the developing tool will be Eclipse 3.5, and the web container will be Tomcat 6.0.

For the implementing of systems, the assembly of original components are necessary in order to make new functional components. The implementation of Apache Tuscany based on Java newest version 1.6 and PHP open source based on SCA are being applied.

4.3.1 SCA Component Developing Environment Settings based on Tuscany

The running of Tuscany will using the Jar package core. They will be assigned under the application's CLASSPATH. Tuscany can run independently because it contains a

web container called Jetty, and its web service engine is Axis2. All the existing systems which needs to be integrated belong to the web container. For running in other web container, VIP customer information management system and other travel advisory website needs to use Tomcat as their specific web container. So these two following steps with SCA support are needed:

1)To define contribution attributes using sca-contribution.xml file in web application META-INF:

```
<contribution xmlns=”http://www.osoa.org/xmlns/sca/1.0”
          xmlns;sample=”http://sample”>
       <deployable composite=”sample;”Define name attribution in
   SCDL””/>
<contribution>
```

2)Adding filter setting in web application’s configuration file web.xml:

```
<filter>
 <filter-name>tuscany</filter-name>
 <filter-class>
  org.apache.tuscany.sca.host.webapp.TuscanyServletFilter
</filter-class>
</filter>
<filter-mapping>
    <filter-name>tuscany</filter-name>
    <url-patter>/*</url-patter>
</filter-mapping>
```

That’s how to add the Tuscany support in web application developing environment. At most situation, the development of SCA components based on Tuscany has these four steps down below:

1) Define the interface of service components

2) Define the implement class of service components

3) The assembly of service components

4) The application of service components for client

4.3.2 Client Management Component and Implementation of SP Message Sending Component

The technique applied in VIP customer information system are JSP and JavaBean. JavaBean is used for implement the model layer. It belongs to simple POJO class, and used as a VO value object. The function of business processing layer had used JSP as page display technique. It's the simplest model in J2EE development. The assembly procedure are down below:

1) Define service component interface and implementation class

The interface and implement class for VIP customer management function model are UserService and UserServiceImpl. The value object for moder is Class User, as described in Figure 12:

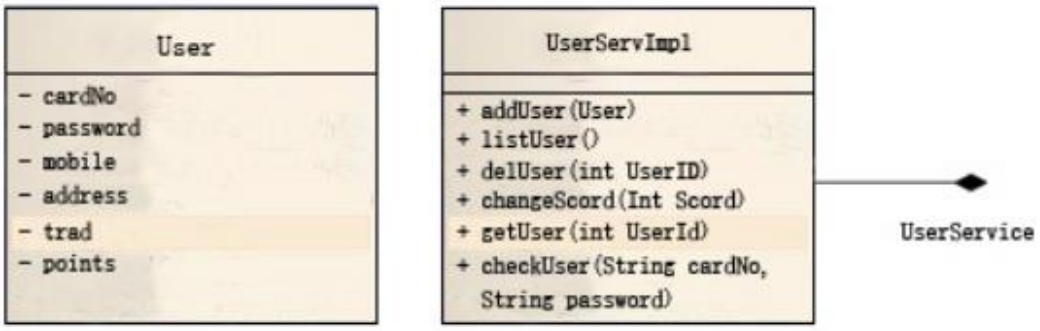

Figure 12. Class Diagram in Implementation of VIP Client Management Component.

Class **UserServiceImpl** implement function includes adding user function **addUser**, method **listUser** and deleting user method **delUser.** Method of change user's credits called **changeScord.** Method of checking user's identity called **checkUser**. VIP customers' User authentication confirmed by card number and correct password.

Sending SP text messages will be implemented by SMS gateway using Java. It will be sealed inside the business processing class code. The corresponding interface and implement class are SendMsgService and SendMsgServiceImpl.

Because the VIP customer information management component and text-message-sending component will be published as web services, the interface **UserService** and **SendMsgService** will be add to the interface,too. Description **@Remotable** will also be added as a note, in order to claim that this is a remote interface. The specific code are shown below:

```
@Remotable
public interface UserService {
//interface method statement
}
@Remotable
public interface sendMsgService{
//interface method statement
}
```

2) Service component assembly

After the function component interface and implementing class has been authenticated, SCA assembly specification will be applied in different components. They are defined with SDCL language and needs to be completed inside SCDL configuration file. The content of SCDL configuration file are down below:

```
  <!- Customer management component configuration file SCDL->
<service name="userService" promote="userServiceComponent">
       <interface.java interface="com.hotel.authority.UserService"/>
```

```
        <binding.ws uri="http://www.xxx.com/user/userserviceWS"/>
</service>
<component name="userServiceComponent">
        <implementation.java
    class="com.hotel.authority.Impl.UserServiceImpl"/>
</component>
        <!-Text message sending configuring file SCDL->
<service name="sendMsgService" promote="sendMsgServiceComponent">
        <interface.java.interface="http://www.xxx.com/spService/sendMsg
    WS"/>
        <binding.ws uri="http://www.xxx.com/spService/sendMsgWS"/>
              <tuscany;binding.rmi host="www.xxx.com" port="8099"
serviceName="sendMsgRMI"/>
</service>
<component name="sendMsgServiceComponent">
        <implementation.java.class="com.hotel.spservice.Impl.SendMsgSer
    viceImpl"/>
</component>
```

On these describing files above, we can see that with the use of lable **<implementation.java/>**, all the implementing method of **userServiceComponent**, **sendMsgServiceComponent** are in Java Class. It is the most frequently seen implementing method in SCA. Component **userService** can be used to bind and publish web services by improving it with **promote**.

(3) The reference of the service component

Once VIP customer information management components and SMS send components have been published and provides services, the travel advisory website will using this function. SCA provides a unified component invoke. This invoke will be implemented by the SCA specification which is service component reference **@Reference** to achieve. It will be discussed in detail in the following paragraphs.

4.3.3 The Implementation of Hotel Room Management Component

At present, hotel information system implementation technique is the PHP scripting language. The system uses the open source PHP SCA implementation to achieve. The hotel information list use PHP script to assemble the component into the hotel management system. The room information component is bundled and published as a Web service. PHP scripting language doesn't have interface and implementation class provided by Java [36]. The only method of assemble the functionality of PHP script into assemble and distributive components is used in here. The following code is the hotel room information component:

```
<?php
  Include"SCA/SCA.php"
  /**
   *implementation for a remote hotelQuery WebService
   *@service
   *@binding.soap
   */
   *class hotelQuery
     /**
       *to search hotel information by city
      */
     Function getHotel($city, $price){
       //Hotel information will be sealed as XML format, and return to
    the XML file content
       }
}
```

4.3.4 The Implementation of Reservation Information Component and Hotel Customer Management Component

Room reservation function is completed in the travel advisory website. Room reservation component requires providing customer management component and hotel room management component. Reservation management function and hotel customer management function are also needed. User log-in interface is required and each

customer info needs to be added according to different hotel number.

1) Define the service component interface, implement class

There's subscription information service interface **RoombookMsgService**, the realization of class **roombookMsgServiceImpl** and booking information **RoombookMsg** class diagram.

The class interface of hotel user management function **hotelUserService**, the implementation of class **hotelUserServiceImpl** and user information **HotelUser** class map are also included.

Class **RoomsbookMagServiceImpl** has implemented adding the reservation information method **addRoombookMng**, delete the booking method **delRoombookMsg**, listing all the information according to hotel number method **listMsgByHotel.** For VIP customers, setting the reservation information method **listMsgByCardNo**, the method **changeState** for changing the status of the reservation info according to the reservation number being used. There are four status which indicate reservation information, they are: -1, 1, 2, 3 respectively correspond to the booking information that "has been canceled", "not confirmed", "confirmed", "completed" Status. This class will implement all the functionality of the system for booking information management.

Class **HotelUserServiceImpl** is to achieve the method of adding hotel users **addHotelUser**, delete the store user method **delHotelUser**, check the hotel user login verification method **checkLogin**, etc., this class will achieve the system of all hotel customers management functions.

The design of travel advisory website uses an open source Spring framework to manage the objects of the system. The operation will be supported by Bean as a Spring container management system that has joined SCA's open source implementation environment Tuscany, according to the third chapter of the system. SCA and Spring can be seamlessly integrated.

The system uses Spring and Tuscany as two framework to achieve the realization of the service layer in the system implementation. SCA name space element <sca:service/> and Spring beans will be declared as SCA services [37]. It defines the

application of the acquired SCA servicess and attributes through the SCA component specifications. The following is the Spring configuration file with the business object:

<- Spring configuration file in bean-rbMsg.xml configuration letter ->

```
<sca: service

  name =”roombookMsgServiceBean"

      Type=”com.hotel.roombook.RoombookMsgService"
target="rbMsgService"/>

      <bean id="rbMsgService"

           Class=" com.hotel.roombook.RoombookMsgServicelmpl “>

           <property name="roombookMsgDao" ref =”roombookMsgDao"


   <!-Spring configuration file in bean-hotlUser.xml confifuration
letter->

<sca:service

     name=”hoteluserserviceBean"
type="com.hotel.hoteluser.HotelUserservice”
           target="hotelUserservice"/>

    <bean id="hotelUserService"/>

      Class=”com.hotel.hoteluser.HotelUserServicelmpl”>

     <property name="hotelUserDao" ref"hotelUserMsgDao"/}

    </bean>
```

3)Assembly of service components

SCA: service exposes beans in the Spring container to SCA services, and the SCA runtime is responsible for creating the appropriate server bindings. The implementation of the service component will be different from that of the VIP client component. The contents of the SCDL assembly file are as follows:

```
<service name"rbMsgService" promote=”rbMsgComponent”>

   <interface.java interface-
=”com.hotel.roombook.RoombookMsgService”/>

   <binding.ws uri=”http://www.xxx.com/rb/rbMsgWS”/>

</service>
```

```
<component name=”rbMsgComponent">
  </implementation.spring location"META-INF/spring bean-rbMsg.xml”/>
  </component>
<service name=”hotelUserservice" promote"hotelUserComponent”>
<interface.java.interface="com.hotel.hoteluser.HotelUserService”/
 <binding.ws uri=”http://lwww.xxx.com/hotelUser/hotelUserWS”/>
 </service>
 <component name=”hotelUserComponent">
    <implementation.spring location=”META-INF/spring/bean-hotelUser.xml
“/>
</component>
```

The value of the spring location attributein element <implementation.spring> is the path to the Spring configuration file. Because the external binding is published for Web services, you also need to add @**Remotable** definition to interfaces in **RoombookMsgService** and **HotelUserService** interfaces to binding and publishing which declares that they are remote interfaces. The realization of room booking function is also achieved in the travel advisory website. Room reservation information management component invoke is a local invoke. Invoke as web services is not necessary, however, the correct way to making local invoke is to add definition @**AllowsPassByReference** to interfaces. This is for improving the performance between different component in one process.

4.3.5 The Implementation of Hotel Room Reservation Component

Previous sections of this chapter describe the assembly and distribution process of the four main components in customer information management. They are customer info management component, hotel room information component, room reservation information component and the SMS sending component. It focuses on describing the implementation of publishing service components. What we need to notice is the implementation of the room booking component requires the support of the other three service components. This section will present how to implement room reservation function in travel advisory platform through component invoke. The other four

components are needed, down below is the configuration of the hotel reservation component **RoombookComponent**.

```
<component name=”RoombookComponent”>

 <implementation.java

     class =”com.hotel.service.component.RoombookComponent”/>

     <reference name=”userService">

      <interface,java interface="com.hotel.service.Userservice”>

      <binding.ws url="http://www.xxx.com/user/userserviceWS"

      wsdlElement=”http://service.authority.hotel.com/
#wsdl.port(UserService

    eService/UserservicePort)"/>

    </reference>

     <reference name"hotelRoomService">

      <interface.wsdl interface="http://xxx/hotel/getHotel.php?wsdl"/>

<binding.ws url=”http://www.xxx.com/user/luserserviceWS"

           wsdlElement="http://service.room.hotel.com/#wsdl.port(Hotel
    RoomServi ceservicel/HotelRoomServicePort)"/>

</reference>

<reference name=”sendMsgService">

<interface.java

Interface="com.hotel.spservice.service.SendMsgService”/

<binding.ws url="http://www.xxx.com/spService/spServiceWS"
wsdlElement=”http://service.spservice.hotel.com/#wsdl.port(SendMsgSe
rviceService/SendMsgServicePort)"/>

</reference>

 <reference name=”roombookMsgService” target=”rbMsgComponent”>

</component>
```

The component of customer management and SP message sending are provides as services. The room reservation component also needs to be advertised as a web

service. Therefore, the component's **roombookComponent** must be added with the **@Remotable** annotation on the interface declaration. This service interface is defined as the remote interface. For the hotel room information management component, it's implemented by PHP scripting language, which was provided as web services after PHP SCA assembly. The system generates a corresponding Java interface **HotelRoormService**, for web service provided by the WSDL definition. It will invoke the web service to return to the hotel room information. Then it will be stored in XML file. Finally, room info will be sealed by using object **HotelRoom** for convenient.

For the Corresponding components of the assembly file,these are the original code of implementation class:

```
public class RoombookComponent {

    private Userservice userservices;

        private HotelRoomService hotelRoomService;

    private RoombookMsgService roombookMsgService

    private SendMsgService sendMsgService,

    @Reference

     public void setUserservice(Userservice userservice) {

         this,userservice=userService;

        }

        @Reference

        public void setHotelRoomService(HotelRoomService
    hotelRoomService) {

        this.hotelRoomService=hotelRoomService;

        @Reference

         public void set RoombookMsgService(RoombookMsgService
            roombookMsgService) {

             this.roombookMsgService=roombookMsgService;

            }

        @Reference

        public void setsendMsgService(SendMsgService sendMsgService)  {

            this.sendMsgService=sendMsgService;
```

```
}
    /**
    * ordering room
    *@param cardNo    Customer's VIP card number
    *@param password Customer's password
    *@param fromDate  The check-in date
    *@param toDate     The check-out date
    *@param hotelNo    The hotel number
    *@param roomNo    The room number
    *@return            The ordering number
    */
    public String bookRoom (String cardNo, String password, Date
fromDate, Date toDate, String hotelNo, String roomNum)
    }
    ...
```

Hotel room reservation function can be achieved by method **bookRoom** in getting components in SCA container. Other component by invoking **Roombook** Component can also be obtained, which will make the system development more convenient, Figure 16 is a room booking procedure, it describes specific booking methods.

4.4 Implementation Effect

At present, new function hotel reservation system in travel advisory website can provides good services. Customers can easily select the city, searching hotel name, hotel address, contact information, room type, and room prices. As well as VIP room preferential prices and the number of rooms with similar type and other information. Room type includes deluxe rooms, standard rooms, double rooms, etc., After all entry there's a reservation button. Click on the booking button, then a booking room interface will appear. You will then asked to enter the card number, password, the number of rooms required for specific type and the time of booking, departure time and contact information. If you have already filled the form once and don't want to re-fill, it will be based on previous VIP customer information for booking. If the booking is successful, it

will return "Your reservation has been successfully issued, please wait for us to process with your booking, your booking number xxxxxx!" while the VIP customers will be sent mobile phone SMS tips. If the subscription fails, etc., it will return and sending the booking failed information.

The travel advisory website will provide the hotel log-in interface . The hotel user will log in to the system using the username and password assigned by the hotel customer. The reservation information includes the VIP customer card number, the contact information of the reservation type of the room, the room price after booking, the time of booking, the time of booking, the departure time and so on. Each entry will be followed by confirmation of the reservation, booking the rejection function button, the information will be required to book customer room number information. Customer will also be informed with full room status, they can also cancel the reservation by reply the text massage on phone.

5 CONCLUSION

This thesis focuses on the architecture of service-oriented architecture SOA and SOA-based programming model SCA architecture. The use of SCA open source Apache Tuscany to integration of legacy systems are being discussed. As well as the combination of SCA strategy framework to support the release of component services.

The main work of this thesis are as follows:

Basic theory, related features, security model and Apache Axis based on open source are studied. The implementation Specification for Web Service Security with Apache. The characteristics and components of the SCA programming model are studied.

The use of SCA with its service invoke and construct feature to achieve hotel room booking function. As well as provides the method of construct SCA programming model with open source technique.

SOA technology system is enormous, however, with my own ability and with the limit of time, there are many technologies which cannot be carry out in-depth research in this thesis, such as the throat triangle “SCA, SDO and BPEL technology”. As well as one of the most important branch in SOA technique system (ESB). There are still systems need to be improve for example, it is possible to handle the hotel room information with the SDO unified data model. What’s else, since obtain the hotel room list by invoke web services take a period of time, a better user experience with Ajax technology is possible to achieve.

In conclusion, SCA-ESB and BPEL in SOA technology system is still the most widely recognized in the IT industry. With the SCA programming model standards, many IT vendors and organizations are busy with its promoting in market. I believe that the SCA Development will become more mature and then become the most preferred programming model for SOA applications.

6 REFERENCES

[1] Wang, Z. (2008). SOA core technology and application. Beijing: Publishing House of Electronic Industry.

[2] Mao, X. (2007). SOA Principle, Methodology and Practices. Beijing: Publishing House of Electronic Industry.

[3] China Electronic Standardization Institution (2017). SOA User Guide. Beijing: Publishing House of Electronic Industry.

[4] META Group, Practical Approaches to Service-Oriented Architecture, Meeting the Demand Today and Tomorrow. (2003). [online] Available at: http://www.ebizq.net [Accessed 22 Jun. 2017].

[5] Service-Architecture. Service-Oriented Architecture (SOA) definition. (2006). [online] Available at: Http://www.service-architecture.com/web-services/articles service - structure architecture soa definition.Html.2006 [Accessed 22 Jun. 2017].

[6] Loose coupled definition. (2017). [online] Available at: http://www.loosecoupled.com [Accessed 22 Jun. 2017].

[7] Service-Oriented Architecture. (2017). Available at: http://www.ibm.com/developerworks/cn/workplacesoa /index.html, 2006, 2 [Accessed 22 Jun. 2017].

[8] Next-generation software architecture SOA. (2004). [online] Available at: Http://dev2dev.bea.com.cn/techdoc./200404186.html, 2004 [Accessed 22 Jun. 2017].

[9] Web Services org. (2017). SOA Governance Balancing Flexibility and Control within an SOA. [online] Available at: http://www.webservices.org [Accessed 22 Jun. 2017].

[10] Erl, T. (2016). Service-oriented architecture. Prentice Hall, pp.48-87.

[11] W3C.org. (2007). SOAP Version 1.2. [online] Available at: http://www.w3.org/TR/soapl2-partO [Accessed 22 Jun. 2017].

[12] Chai, X. (2003). Web Services Technology, Architecture and Application. Beijing: Publishing House of Electronic Industry.

[13] W3C.org. (2001). WSDL 1.1. [online] Available at: http://www.w3.org/TR/sdl [Accessed 22 Jun. 2017].

[14] Universal Description, Discovery and Integration (UDDI) technical reference information. (2017). [online] Available at: http://uddi.xml.org/ [Accessed 22 Jun. 2017].

[15] Ibm.com. (2017). Web Services Security specification 9.0.0. [online] Available at: https://www.ibm.com/support/knowledgecenter/beta/SSEQTJ_9.0.0/com.ibm.websphere.nd.multiplatform.doc/ae/cwbs_wssv6chron.html [Accessed 22 Jun. 2017].

[16] Gordon, A. and Bhargavan, K. (2006). Verified Reference implementations of WS-Security Protocols. Heidelberg, pp.88-106.

[17] Shi, W. (2003). Web service security specification based on the SOAP protocol [J]. Application Research of Computers. pp.100-102.

[18] Wang, F., Li, Y. Lang, B. and Li, C. (2004). Securing SOAP Message Exchange with WS-Security. JOURNAL OF COMPUTER APPLICATIONS, 24(4), pp.121-126.

[19] Liu, C., Yuan, J. and Han, M. (2007). Data security exchange with WS-Security. Microcomputer Information, 23(10), pp.91-93.

[20] Wang, C. (2007). Implementation of SOAP security in AxisWeb. Journal of Taiyuan University of Technology, 38(04).

[21] Wso2.com. (2007). Apache Axis2/C – Web Services Engine. [online] Available at: http://wso2.Org/library/2406 [Accessed 22 Jun. 2017].

[22] Wso2.com. (2007). The Insider's Guide to Apache Rampart/C and OMXMLSecurity. [online] Available at: http://wso2.com/library/1815/ [Accessed 22 Jun. 2017].

[23] Shan, J. and Lu, Z. (2008). SOA Integration Solution. Beijing: Publishing House of Electronic Industry., p.10.

[24] IBM Knowledge Center. (2017). Service Component Architecture. Version 5.1.0 [online] Available at: https://www.ibm.com/support/knowledgecenter/en/SSGMCP_5.1.0/com.ibm.cics.ts.applicationprogramming.doc/bundleinterface/sca.html [Accessed 22 Jun. 2017].

[25] Ibm.com. (2006). Open SOA (OSOA.org) Service Component Architecture Specifications. [online] Available at: https://www.ibm.com/developerworks/community/blogs/KRL/entry/open_soa_osoa_org?lang=en [Accessed 22 Jun. 2017].

[26] Tang, Y. (2008). Research of SCA application (Master's Thesis), Guangzhou: Sun Yat-sen University

[27] Jing G., Yao H., Zhao Y. and Tan J. (2006). SOA Quick Guide Part 2: Service Modeling. [online] Available at: http://www.ibm.com/developerworks/cn/webservices/061O jingge/ index2.html

[28] Herrington, J. (2003). Code generation in action. Greenwich, CT: Manning.

[29] Graham B. (2003). Service Component Architecture Specifications [EB/OL]

[32] Tuscany.apache.org. (2016). Apache Tuscany: Java SCA Documentation. [online] Available at: http://tuscany.apache.org/java-sca-documentation-menu.html [Accessed 22 Jun. 2017]. 2008

[33] SCA Spring Component Implementation Version 1.1. (2011). [online] Available at: https://www.oasis-open.org/committees/download.php/42052/sca-springci-1.1-spec-WD08.pdf [Accessed 22 Jun. 2017].

[34] TechTarget SOA. (2010). Tuscany and SCA Strategy. [online] Available at: http://www.searchsoa.com.cn/showcontent_33146.html [Accessed 22 Jun. 2017].

[35] Docs.oasis-open.org. (2011). SCA Policy Framework Version 1.1. [online] Available at: http://docs.oasis-open.org/opencsa/sca-policy/sca-policy-1.1.html [Accessed 22 Jun. 2017].

[36] Wso2.com. (2008). PHP Web Services: Getting Started. [online] Available at:

http://wso2.com/library/3032/ [Accessed 22 Jun. 2017].

[37] Design and development of SCA component using Spring Framework. (2009). [online] Ibm.com. Available at: https://www.ibm.com/developerworks/cn/opensource/os-springsca1/ [Accessed 22 Jun. 2017].