ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 8, August 2014

Service Oriented Architecture using ISO RM-ODP with Respect to Computational Viewpoint

C. Madana Kumar Reddy[1], Dr. A.Rama Mohan Reddy[2], Dr. P.Chittibabu[3]

Research Scholar, Ph.D.Computer Science, Associate Professor of MCA, Annamacharya PG College of Computer Studies, Rajampet, Kadapa(dt),Andhra Pradesh,India[1]

Professor, Dept of C.S.E, S.V.U.College of Engineering, S.V.University, Tirupati, India[2]

Principal, Annamacharya PG College of Computer Studies, Rajampet, Kadapa(dt), India[3]

ABSTRACT: The Service Oriented Architecture using ISO Reference Model for Open Distributed Processing is a high performance technique for providing effective services to the users. This architecture consists of five different viewpoints namely Enterprise viewpoint, Computational viewpoint, Information viewpoint, Engineering viewpoint and Technology viewpoint. This paper discusses the details about the Computational viewpoint. Computational viewpoint defines the functional decomposition of the system into a set of services that interact at interfaces. These services can be found in a security system and present as classification based on the functionality of services. Interaction between the service requester and the security system, and the communication between the security system and the service provider is carried out through external messages. The security system receives an input message then processes it and produces an output message. The semantics of the message is the issue of information view point. If the security check fails or if the processing logic can't be correctly applied, security services can raise exceptions signalling that the errors encountered. Exception message contain annotations and other meta-information.

KEYWORDS: Services, performance, Computational viewpoint, security, process.

I. INTRODUCTION

This viewpoint describes the main concerns. According to RM-ODP [5], the Computational Viewpoint defines the functional decomposition of the system into a set of services that interact at interfaces. It performs the internal actions of these services, interactions that occur between services and contracts between the services and their environment, without regard to distribution. It decompose the security system into functional components i.e., security services. These services can be found in a security system and present a classification based on the functionality of services.

II. RELATED WORK

Service Oriented Architecture using ISO RM-ODP (Reference Model for Open Distributed Processing) main focus is to provide security in effective form, as the part of providing security it requires five different viewpoints. These are Enterprise viewpoint, Computational viewpoint, Information viewpoint, Engineering viewpoint and Technology viewpoint. Enterprise viewpoint describes the scope, role, policies and activities of the system. Computational viewpoint describes the description of the system as a set of interfaces, without regard to distribution. Information viewpoint describes the information that needs to be stored and processed by the system. Engineering viewpointdescribes the system distribution and the infrastructure required to support it.Technology viewpoint describes the choice of technology chosen to implement the system.

DOI: 10.15680/IJIRSET.2014.0308055

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 8, August 2014

III. COMPUTATIONAL OBJECT

The ISO RM-ODP [6] describes objects and computational objects as follows:

1. **Object**: An object is a model of an entity. An object is characterized by its behaviour and, duality, by its state. An object is distinct from any other object. An object is encapsulated, i.e. any change in its state can only occur as a result of an internal action or as a result of an interaction with its environment.
2. **Computational Object**: A computational object is an object seen in the computational viewpoint. It represents functional decomposition and interacts with other computational objects. Since it is an object, it has state and behaviour, and its interactions are achieved through interfaces.

IV. SECURITY SERVICES

For making security systems simple, it split into smaller parts or subsystems. From a computational viewpoint, each subsystem should be regarded as a computational object: it has state and behaviour and it interacts with the other subsystems in order to fulfil the scope of the security system. Here, these subsystems are called security services. In Figure 1, a security system has composed from several security services and each security service is composed from service part and system part.

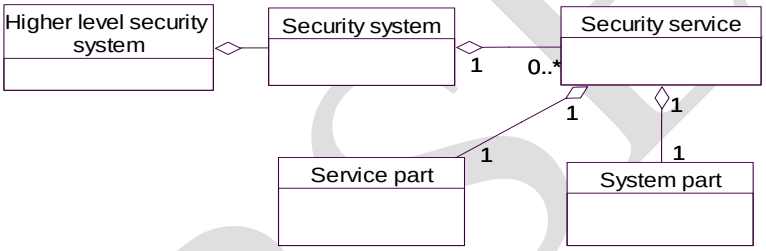

Fig.1Security Services (UML class diagram)

Service Part: Service part is specific to the security service. It implements the functionality of the security service.
System Part: System part describes that how services link to the system. It is concerned with how the security services interact in order to fulfil the security of system. Normally, services run on top of some sort of a middleware system which implements the system part.

V. INTERACTION INTERFACES

If there is no security system, the service requester can directly communicate with the service provider using the external messages. When a security system is placed in between the service requester and the service provider, the communication between the service requester and the security system and the communication between the security system and the service provider is carried out through external messages.

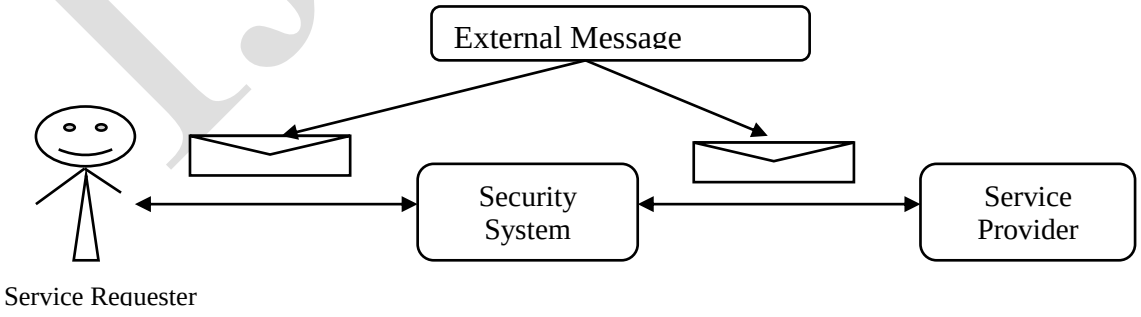

Fig. 2.Interactions through external messages

DOI: 10.15680/IJIRSET.2014.0308055

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 8, August 2014

Figure 2 shows how the security system is interacting with the service requester and service provider. This makes the security system easy to implement. Internally, the security system is structured into many services. Strictly how these services are organized and distributed is the issue of engineering viewpoint. The security system receives an input message then processes it and produces an output message. The semantics of the message is the issue of information viewpoint.

VI. SERVICE STRUCTURE FLOW

General principles of Service structure flow are as follows:

- The purpose of the security system is to communicate the communication between clients and the protected services. Thus the main task of all security services is to process messages in order to apply security functionality.
- For achieving the system's scope, security services must work together. i.e., services communicate with one another by means of annotations.

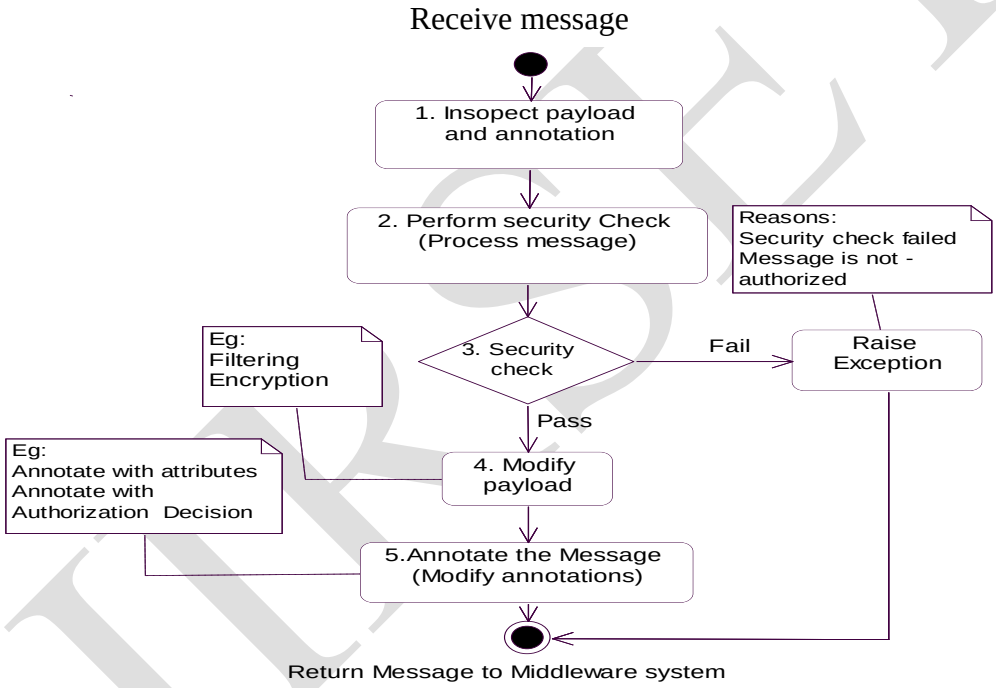

Fig. 3Internal structure flow for security services

The general internal structure flow for security services is shown in Figure 3 is as follows:

1. After receiving a message, a security service will first analyse the message and inspect the payload of the message and the annotations created by the services that previously processed this message.
2. Based on both the payload and the annotations, the processing logic will be applied. For examples verify the identity of a user by checking against username in directory or making an authorization decision based on the information in the payload and the annotations that were prepared by services.
3. If the security checks failed, or processing logic could not be performed correctly, an exception is raised. The messaging middleware will dispatch the exception.
4. In the second step, the payload will be modified i.e., parts of it can be filtered out or it can become greater or transformations can be applied (E.g. encryption).
5. If the service needs to share information with the other services, it will introduce additional annotations or modify the existing annotations.
6. At the end, the message is returned to the middleware and hands out the message to other services.

DOI: 10.15680/IJIRSET.2014.0308055

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 8, August 2014

VII. EXCEPTIONS

If the security check fails or if the processing logic cannot be correctly applied, security services can raise exceptions signalling that the errors encountered. Here, a model similar to the SOAP (Service Oriented Approach) model [7] is used for exceptions: an exception is a message with a distinct payload containing details or reasons about the fault. In addition to this, the exception messages can also contain annotations and other meta-information. Exception messages must be distinctly marked. Because exceptions are treated like regular messages, they can be processed in a similar way with request and response messages. It is up to the messaging middleware to properly route the exception messages to appropriate services that handle them.

VIII. SERVICE TAXONOMY

It describes the possible types of services in a security system. For each of the services we should specify how messages are processed, how the payload and annotations are altered and what decisions are taken. The description of the security services is done from a computational point of view. Thus, focus on the functionality of each service and on the interactions between the different services. For each of them specify the prerequisites - what the service expects from the other services that have processed the message at a previous point in time; and the responsibilities - the functional description of the service, including the communication between this service and other services.

VIII.1 Gateway Service

On its way from the service requester to the protected service a message will pass by means of different transport protocols. All transport protocols are able to convey some kind of security-relevant information. However, when switching from one protocol to another, it is not always possible to pass this information forward - some protocols are higher in security information than others. The only way to relay such information is to extract it from the transport protocol and append it to the message.

One place where this usually represents an issue is at the network layer, where the message coming from the service requester enters the security system. At this point, a gateway service can be introduced which extracts security relevant information from the transport protocol and appends it to the message in the form of annotations. Because of the gateway service is located at the network layer, it convert messages from external messages (without annotations) to its equivalent internal messages (with annotations). The Envelope Wrapper pattern is applied here i.e., the original message is wrapped around an enveloping message which is semantically richer.

Prerequisites: None.

Responsibilities:

1. Wraps the external message in an internal message;
2. Extracts security relevant information from the transport protocol layer (i.e. IP address, HTTP Authentication Information) and appends it to the message in the form of annotations.

VIII.2 Authentication

Authentication is a major security issue. An authentication service is a security service that verifies an identity claimed by an entity [8]. Authentication consists of two different activities: identification and verification service.

Identification Service:

Identification refers to the process of distinguishing the requester from other entities. This is done by means of attributes. Some identification information is included by the requester himself in the message. The task of an identification service is: first, verify the identification information sent by the requester. Then, it must extend with sufficient additional attributes for the other services to be able to perform their tasks.

Prerequisites: The service expects some information related to the identity of the requester to be present in the message.

Responsibilities: An identification service will typically have the following behaviour:

1. Verify the supplied identification information is acceptable, sufficient and can be understood.
2. Lookup the user in a database and retrieve additional attributes about the user;

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 8, August 2014

3. Annotate the message with the attributes related to the user and the authentication method, so that other services can utilize this information.

Verification Service:

Service requesters normally attach some information about their identity to service requests. The task of a verification service is to check that the authentication information accompanying the message is valid.

Prerequisites: The service expects some information about the identity of the requester to be present in the message.

Responsibilities: Verify that the information about the identity of the requester is correct. This might include checking the validity of digital signatures and certificates, verifying passwords and other authentication methods.

VIII.3 Authorization

Authorization is the process of determining whether the requester may use the protected service or not. This is perhaps the most obvious security function of all.

OASIS's XACML 2.0 standard [4] defines the following components:

Policy Administration Point (PAP) The system entity that creates a policy or a set of policies.

Policy Information Point (PIP) The system entity that acts as a source of attribute values.

Policy Decision Point (PDP) The system entity that evaluates applicable policies and renders an authorization decision.

Policy Enforcement Point (PEP) The system entity that performs access control, by making decision requests and enforcing authorization decisions.

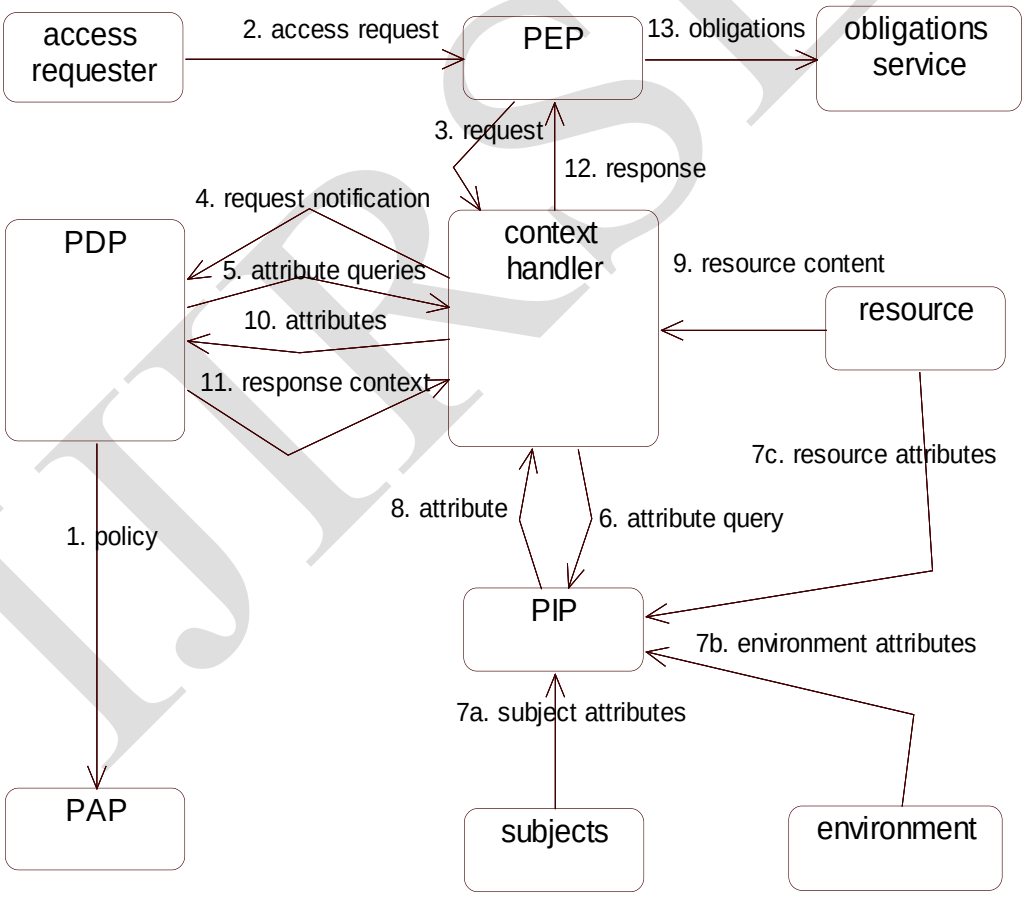

Fig. 4 Data flow and component interconnections in XACML

IETF's AAA Authorization Framework defines the following components:

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 8, August 2014

Policy Retrieval Point (PRP) An entity responsible for retrieving the authorization policies from a policy repository on behalf of the organization that requires it.

Policy Information Point (PIP) An entity that provides information against which policy conditions are evaluated (such as resource status, session state, or time of day).

Policy Decision Point (PDP) A logical entity that makes policy decisions for itself or for other network elements that request such decisions [9].

Policy Enforcement Point (PEP) A logical entity that enforces policy decisions [9].

Both specifications propose similar components. The PIP, PDP and PEP have the same name, perform similar tasks and are interconnected in similar ways in both models (Figure 4shows the data flow between the different components in XACML).

PIP, PDP These two services can be realized as both mediation services as well as RPC. XACML [4] proposes an RPC definition for these services.

PEP If realized as a service separate from the rest of the security system, it only makes sense to realize this service as an intermediate service.

PAP / PRP: The administration of policies and the policy repository, if exposed as a service, it only makes sense to be realized as an RPC service.

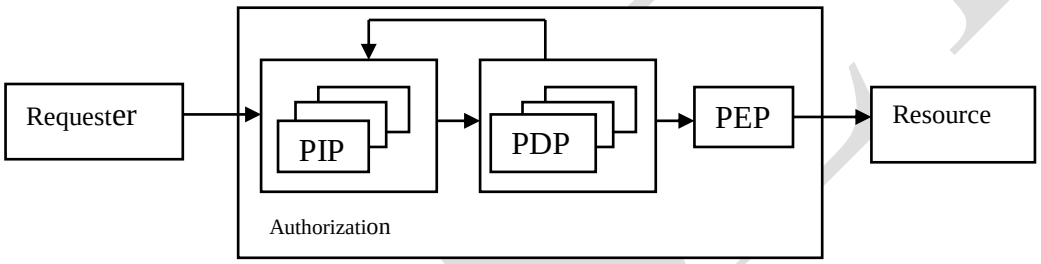

Fig. 5 Data flow and component interconnection for authorization in SOSA

Figure 5 illustrates how the authorization services can be chained together in an authorization process. The engineering viewpoint describes in more detail regarding how different security services can be combined and several patterns are proposed.

VIII.4 Policy Information Point - PIP

The role of the PIP is to provide information against which the policy conditions are evaluated. It must provide the PDP with enough information for this one to be able to make the access control decision. In XACML [4], the PIP provides information in the form of attributes. Three types of attributes are defined:

1. S*ubject attributes* (refer to the service requester),
2. *Resource attributes* (refer to the resource being accessed) and
3. *Environment attributes* (refer to the context / environment).

Prerequisites: None

Responsibilities:The service inspects the messages it receives and annotates them with attributes. These attributes will later be used by PDP services when evaluating the policies.

VIII.5 Policy Decision Point - PDP

The PDP is to make authorization decisions based on the applicable policies. A PDP service first needs to determine the policies which are applicable for the given context (message, requester, and environment). To do this, the PDP may query one or more PAP services which will provide the policies to the PDP. The mechanism for this, together with a description of how different policies can be combined and how conflicting rules are resolved is described in the XACML specification [4].

In the policy evaluation process, the PDP may require additional information about the requester, the environment or the resource being accessed. In XACML, a pull model is used (see Figure 4) - the PDP requests attributes from the context handler, a logical component which then connects to different PIP services. This model assumes PIP services expose an RPC-like interface. In this thesis assume that security services are realized as mediation services. In this case,

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 8, August 2014

a push model is necessary: the PIP services annotate messages with attributes, in this way pushing the information to the PDP service.

Prerequisites: The PDP requires enough information in order to be able to evaluate the applicable policy. Most access control decisions are based on the identity of the requester; it is the task of identification services to provide identity attributes.

Responsibilities:

1. Determine the applicable authorization policies. For this the PDP may query PAP services.
2. Based on the authorization policies, the payload of the request and other relevant information (identity attributes, other attributes provided by PIP services, authorization decisions from other PDP services) make an authorization decision.
3. Annotate the message with the authorization decision so that the other services (PEPs, Audit, other PDPs, etc.) can use this information.

VIII.6 Policy Enforcement Point - PEP

The PEP is to enforce the authorization decisions taken by one or more PDP services.

Prerequisites: The PEP expects one or more authorization decisions to be attached to the message.

Responsibilities:

1. Retrieve the authorization decision and the obligations from the message.
2. Enforce the authorization decision and discharge the obligations. The enforcement is done by denying the requests for which the authorization decision is not authorizedor for which the obligations cannot be discharged.
3. As an option, annotate the message with the reason for rejecting / filtering the message.

VIII.7 Encryption

An encryption security service encrypts parts of the message. It ensures the confidentiality of the data being encrypted. Only the entities possessing the decryption key are able to read the data. The data is protected while in transit against eavesdroppers that tap the communication channels and against intermediaries that further process the message. The encryption algorithm and keys can be either statically configured or dynamically assigned. A typical usage scenario for an encryption service is where a client invokes the protected service and mandates that the response message be encrypted with his public key.

Prerequisites: None.

Responsibilities:

1. Determine what parts of the message shall be encrypted, what algorithms and keys shall be used;
2. Apply the encryption algorithm.

VIII.8 Digital Signature

A digital signature security service digitally signs parts of the message it processes. As such, it ensures the integrity of these parts (i.e. other entities processing the message can be sure that these parts were not altered) and non-repudiation (i.e. the receiving party has a proof of origin). Here a client invokes the protected service and mandates that the response message be digitally signed by the protected service.

Prerequisites: None.

Responsibilities:

1. Determine what parts of the message shall be digitally signed, what algorithms and keys shall be used;
2. Apply the digital signatures.

VIII.9 Digital Rights Management

Digital Rights Management is a collection of technologies that enable technically enforced licensing of digital information [10]. In most of the Digital Rights Management systems the digital contents has an associate digital license which states the conditions under which the content might be used. In order to enforce the license, the content is usually packaged together with its license and then encrypted. Only authorized devices (that know the semantics of the package and have knowledge of the decryption key) are able to decrypt the package and make use of the content. A Digital Rights Management security service is a service that assigns a license to the content, bundles the license

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 8, August 2014

together with the contents and finally encrypts the bundle so that only authorized devices are able to make use of the contents.Examples for this are music, video, books, etc.

Prerequisites: None.

Responsibilities:

1. Associate a license with the contents;
2. Package the license together with the contents;
3. Encrypt the package using the appropriate algorithm and keys.

IX. GRANULARITY AND REUSABILITY

The Granularity of the services is also an important issue. While having fine grained defined services brings a good separation of duties which in turn makes the services reusable, this also has its negative impact because it makes the choreography of the services complex, and may render performance problems. When making a choice regarding the granularity of the services, it is important to consider all the requirements of both the service requesters and the protected service. Another issue related to granularity is reusability. There is always the choice between making a specialized service and a more general, configurable and reusable service.

X. SIMULATION & RESULTS

The Sensor Grid implementation has been used in several projects for archival and real-time data access. Real-time data sources such as sensors are generally used by small number of experts and specific applications.The first test is to run the system for 24 hours and record the timings. At the end of the test we first measure the average message delivery times, and then by dividing the timings into segments, figure out if continuous operation degrades the system performance. We also want to see is the message delivered in the incoming order.

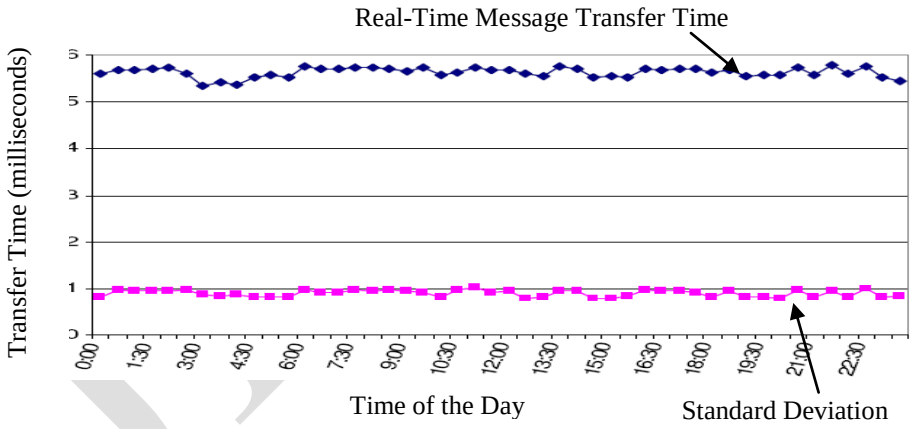

Fig. 6 System Stability Test Results for 24-hour operation of sample test setup.

XI. CONCLUSION

A security system has composed from several security services and each security service is composed from service part and system part. For providing the security between service requester and service provider, the communication should be carried out with the external messages. Suitable exceptions will rise corresponding to the errors. SOAP (Service Oriented Approach) model is used for handling exceptions. Exception messages must be distinctly marked.

REFERENCES

[1] Bishop, M.,"A model of security monitoring", Fifth Annual Computer Security Applications Conference, 1989.
[2] David A. Chappell," Enterprise Service Bus", O'Reilly, pp.145-156, 2004.
[3] David Booth, Hugo Haas, Fracis McCabe, Eric Newcomer, Michael Champion, Chris Ferris, and David Orchard, "Web Services Architecture". World Wide Web Consortium (W3C), http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/, 2004.

DOI: 10.15680/IJIRSET.2014.0308055

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 8, August 2014

[4] ISO/IEC 10746-1, Information technology - Open Distributed Processing - Reference Model: Overview. International Standards Organization (ISO), "http:// isotc.iso.org", 1998.
[5] ISO/IEC 2382-01,"Information Technology – Open Distributed Processing - Use of UML for ODP system specifications", International Standards Organization (ISO), committee draft v02.00 edition,2005.
[6] Jiawei Han and Michelilne Kamber,"Data Mining:Concepts and Techniques",Morgan Kaufmann publishers Newdelhi,pp.192-203,2008.
[7] Madana Kumar Reddy.C., "Operating Systems Made easy" ,University Science Press, New Delhi, pp.23-49,2009.
[8] Mita Devi,"Information security Global perspectives",The ICFAI University press, Hyderabad,pp.24-49,2005.
[9] RFC2828:Internet Security Glossary. Internet Engineering Task Force(IETF),Category: Informational, "http://tools.ietf.org/html/rfc282",2000.
[10] Rob H. Koenen, Jack Lacy, Michael Mackay, and Steve Mitchell, The long march to interoperable digital rights management, "http://ieeexplore.ieee.org /iel5/5/28864/01299164.pdf.", pp.883 – 897,2004.
[11] Wolfgang Eibach and Dietmar Kuebler,Metering and accounting for web services. IBM developerWorks," http://www-128.ibm.com/ developerworks/ webservices/ library/ws-maws", 2001.