Reference Architecture for Personalized E-learning Systems using Proxy and Caching (RAPESPAC)

K. Palanivel
Computer Centre
Pondicherry University
Pondicherry – 605014, India

S. Kuppuswami
Department of Computer Science
Pondicherry University
Pondicherry – 605014, India

ABSTRACT

Service-Oriented approach is the most appropriate architectural approach that is widely used for building highly large-scale enterprise systems. SORAPES is a service-oriented reference architecture designed for enterprise e-learning systems. This architecture is designed by re-using web services and learning objects for distributed environment. It is notified that the system generate more traffic during authentication/ authorization process and retrieval of information from the database. The problem here is to minimize the traffic during authentication and retrieval ode learning objects process. This traffic will be minimized by centrally managing caching scheme and secured web service requests. This improves the performance and scalability of SORAPES in distributed environment, and hence it is proposed to design reference architecture using proxy and caching to authentication. The objective of this research paper is to design a reference architecture using proxy and caching called RAPESPAC. The RAPESPAC reduces the load on large-scale e-learning applications in distributed environment by centrally managing and caching the web service requests. The important features of RAPESPAC are load balancing, proxy, caching and policy. This architecture could be deployed with confidence in high transaction volume service environments providing high availability of data and reliable security mechanism.

General Terms

Service-Oriented Computing, E-learning Systems, Distributed Computing.

Keywords

Reference Architecture, Service-Oriented, Personalization, Web Service, Security, Proxy, Caching.

1. INTRODUCTION

Web services are platform-independent, loosely coupled Web-enabled applications to complete a concrete task or conduct a business transaction [1]. Web services use well-defined interfaces and message-based interaction to expose business services and applications over the existing internet infrastructure, hence enable robust and cost-efficient business interactions among individuals and organizations.

Service-Oriented Architecture (SOA) allows many applications to be loosely joined and yet perform different functions efficiently. SOA was used for loosely-coupled systems that need to interconnect with other systems in the distributed environment. SOA is required to implement a general purpose service that renders in different formats [2]. This removes the need to keep building multiple copies of services that would be needed by e-learning applications in distributed environment. In the future if the developer of e-learning applications decides to render content in a completely new format, the same general purpose service could be called to do the rendering. In designing service-oriented architecture, it is necessary to connect interface elements to back-end services.

Reference architecture [3] provides guidance for the development of new versions of architectures to e-learning systems. They address the issue of design by reusing learning objects and contents from different domains to design the views of the reference architecture. Design is an important step in creating reference architecture to learning systems.

SORAPES [4] is a highly scalable and reference architecture designed using Web services and SOA. This architecture is designed to supports more scalability and more reusability for large scale e-learning systems in distributed environment. SORAPES It provides managers with the capability to monitor the progress of learners and assess course effectiveness.

During the resign of e-learning architecture, it is notified that the system generates more traffic during authentication, authorization and content retrieval process and this leads to decrease the performance and scalability in distributed environment. Hence it is proposed to design standard e-learning architecture with less traffic during login and content retrieval process. This can be done by centrally managing caching, proxy and secured web service requests. This improves performance and scalability of the proposed architecture using proxy and caching to the users.

The objective of this paper is to investigate and provide reference architecture for personalized e-learning systems using proxy and cache (RAPESPAC) web services. RAPESPAC provides optimized access to the distributed services, without influencing the scalability of the e-learning applications running in such distributed environments. The proxy and cache service reduces the load on the enterprise's security infrastructure. Instead of generating lot of message traffic across the network, it invokes web services such as proxy and caching. Th proxy reduces the traffic by centrally managing and caching the authentication and authorization of web service requests.

1.1 Motivation Example

The motivation behind to design RAPESPAC is presented in Fig 1. The users of e-learning systems are connected to the E-learning portal via Internet. The users are entered their user-id and password. The authentication or LDAP Server offers Single Sign-on (SSO) service that provides access control of multiple related, but independent software systems. A user logs in once and gains access to all other modules without being prompted to log in again at each of them. The Policy server responds to and locates policy events to relevant policy rules assigned to different users. It performs validity check for state and resources loaded by the teachers. It converts policy rules stored in the repository for enforcement. The *E-learning Application Server* is a public domain standard, which describes a standard for processes within e-learning application management.

The *E-learning Portal Server* communicates with other web servers / system in the network. It provides information about remote role assignment and remote links. The portal communicates remote role assignment information to the remote client/server. The *Learning Object Server* shares service which gives a single interface to upload, store, manage and share stuff like photos, videos, files, bookmarks almost any digital learning content. The learners and teachers are connected to the E-learning portal server and availed various services offered by the portal in large-scale distributed environment.

1.2 Problem Statement

In the above example, a lot of message is generated across the network to authenticate and authorized each user / learner every time. Also, many users retrieve the same learning contents or objects from the database. This may lead to decrease the performance and scalability of the E-learning applications/architecture. We can improve the performance of the E-learning applications by using caching and proxy, in distributed environment. The proxy reduces the load on large-scale distributed e-learning applications [24] by centrally managing and caching the authentication and authorization of web service requests. The statement of the problem is to improve the performance and scalability during the design of reference architecture under distributed environment.

1.3 Solution to the Problem

The solution to the above problem is to investigate if a reference architecture for personalized E-learning systems using proxy and caching services i.e. RASPESP. It is a scalable and layered architecture that consists of presentation layer, service registry layer, application layer, caching layer and storage layer. The *presentation layer* provides web interface between learners/users and the systems or service providers. This *service layer* consists of all the services identified and defined within the SOA in the form of Web Services Definition Language (WSDL) description. This information is not necessarily written using WSDL. The E-learning *Application layer* serves as supporting components of the learning services and mediator between presentation and storage layers. It receives requests from the presentation layer, performs the functions and need to be able to respond to those requests. The responsibility of *storage layer* is storing persistent data.

Various services have been identified in RAPESPAC and they are security, caching, proxy, notification, etc. These services are registered in service registry. RAPESPAC allow the content creators/providers to easily share their archives through the Web portal or with e-learning application. It also allows the administrator to define the way data should be rendered to the learners.

The research paper is structured as follows. *Section 2* describes the theoretical background such as security services, proxy services, caching services and some related works Web Service Notifications, etc. Literature review was done in *section 3.* Various architectures using security were identified and studied. *Section 4* describes the architecture of RASPESP. The description includes the requirements and architecture. Finally, section 5 concludes the paper and provides recommendations for future works.

2. BACKGROUND DETAILS

2.1 SORAPES

SORAPES is a service-oriented reference architecture designed for personalized e-learning systems. The design uses layered web application architecture. Fig.2 shows that the architecture of SORAPES. It consists of presentation layer, service layer, application layer, semantic layer, cache and data layer.

The *presentation layer* handles the interaction between learners/users and application layer. This layer consists of an HTML based browser user interface and displays information to the user. This layer interprets the commands from the user into actions upon the application layer components. The *service layer* consists of all the services defined within the SOA. The specification provides the users or consumers with sufficient detail to invoke the applications exposed by a provider of the service. The E-learning *Application layer* serves as supporting components of the learning services and do not communicate directly with the presentation layer. It accepts requests from the presentation layer and need to be able to respond to those requests. The responsibility of *database layer* is storing/ retrieving persistent data. The communication should be where a application component sends SQL requests to data sources and data sources sends responses back to application components.

2.2 Web Service

A Web Service [5] is a web application designed to support interoperable over a network. They expose only their interfaces using WSDL to the public. The Web Service provides, messages and data types that are used to construct those messages. The implementation detail of Web Services is hidden behind the interface. The users or consumer is aware that certain functionality is provided not how this is done are abstracted behind the Web Service interface. Web services, which have three main goals: virtualization, interoperability and reusability. Various components of Web Service are SOAP, WSDL and UDDI and they described below:

SOAP [6] is an application-level protocol used for data exchange and remote procedure call in distributed applications. It is usually used for accessing Web Services. SOAP is platform and programming language independent due to XML. SOAP messages are transmitted embedded into or on top of other application-level protocols such as HTTP, SMTP or JMS.

The Web Service Description Language (WSDL) describes a Web Service interface through an XML document that conforms to an XML Schema as defined by the WSDL specification [6] of the W3C. WSDL provides machine-processable information on how to interact with a given Web Service to a Web Service consumer application. The Web Service is fully described by the WSDL document.

The Universal Description Discovery and Integration (UDDI) [8] defines a Web Service registry that allows possible Web Service consumers to dynamically discover the Web Services. The UDDI registry is actually a Web Service itself and makes use of WSDL to describe its interface. The purpose of the UDDI registry is to allow client applications to dynamically discover Web Services that provide a required service.

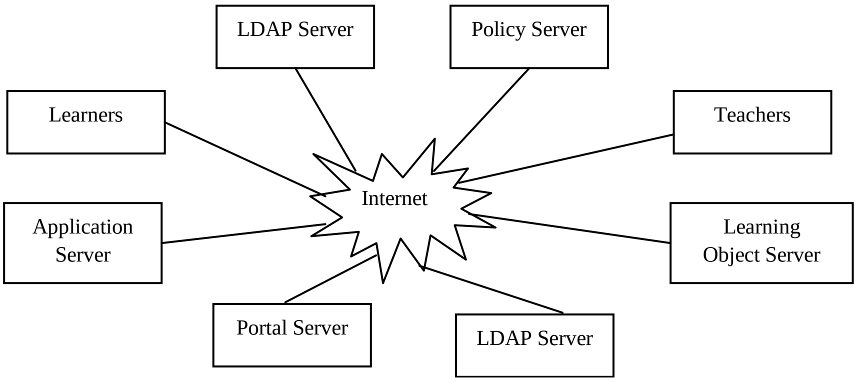

Fig1: Motivation Example for RAPESPAC

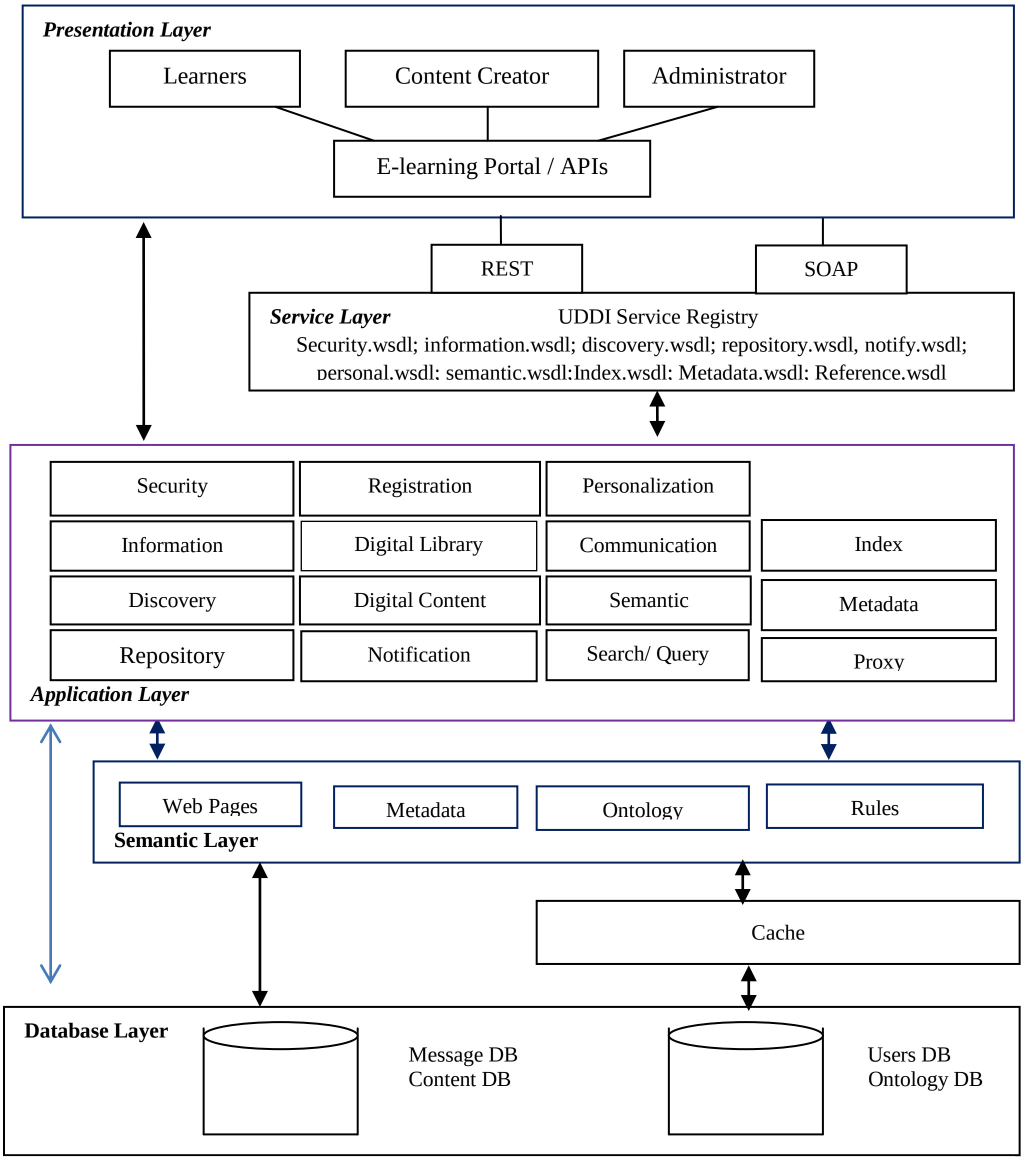

Fig 2: Architecture of SORAPES

The use web services also bring about the important benefit of collaboration. The use of service-oriented architecture and specifically the use of web services could increase the level of collaborations and cooperation and also remove all bottlenecks that impede cooperation. In addition to that, it could make the digital library agile enough to meet increasing demands on it.

2.3 Web Service Notifications

Web Service Notifications (WSN) [9] allows Web Services to notify the interested entities of events that have occurred inside the Web Service. A Web Service publishes a topic of events to which other interested entities may subscribe using publish-subscribe pattern [10]. Occurrence of an event inside the publishing Web Service then triggers the notification of subscribed entities. WSN may also involve an intermediary Web Service and enhanced scalability to the Web Service Notification architecture.

2.4 Web Service Security

WS-Security [11] is a good choice for interoperability with external partner but might have a negative effect on performance when used internally. The security services view therefore needs to provide specifications of the available security services both for internal use as well as for external use. They provide a large degree of flexibility. In terms of virtualization, it needs to be able to authenticate users in one place and authorize them in another through well defined security model [26] is defined by the security constructs in the document.

2.5 Security Services

The security service [14] is concerned with the provided security services of security architecture. Security aims to centralize most of the security related functionality. The security goals such as confidentiality, integrity, and availability (CIA) are delivered through such mechanisms as authentication, authorization, and auditing, but the challenge is deploying these mechanisms as widely and flexibly as possible through services. C

Confidentiality shares the information among authorized users, entities and processes at authorized times and in an authorized manner. In E-learning systems, the importance of confidentiality is examination.

Integrity gives piece of information is timely, accurate, authentic and complete. It refers to the validity of data, user account controls should not be inappropriately modified because even a momentary change can lead to service interruptions and result in breach of confidentiality. Integrity can be ensured by making use of digital signatures which are equivalent to traditional handwritten signatures allowing the sender to verify his identity.

Availability supports systems being usable and accessible on a timely basis. It requires the systems responsible for delivering, storing and processing information to be accessible when needed by other systems or stakeholders which need them. Availability is also an assurance that relevant information should be provided to the client of the service, devoid of the annoyance of information overload. There are systems today with computing resources whose architectures are specifically designed towards improving availability. An example, the e-learning systems need to be available online on a 24/7 basis.

2.6 Aspects of Security

SOA dictates additional security requisites that have to be satisfied in order to have a secure environment and they are authorization, authentication, access control, identity and policy. Security can be viewed as a three part process which involves protection, detection and response.

Consider an e-learning application that combines the capabilities of multiple services. As an action the e-learning application may consist of multiple actions in constituent services, the composite application should ideally check the access control rules of all constituent services before initiating an action. But this is only possible if the access control rules of each constituent service are also available to the composite application.

Authorization is allowed access to resources to only those who are permitted to use them. i.e. if a person has the rights to access the resource, or has a particular level of security clearance. Authentication is the process of establishing or confirming the identity of a user. This usually involves a username-password combination, but can also include other methods of demonstrating identity such as a smart card or use of biometrics techniques like fingerprint reading, retina scan etc. Identity is an essential attribute required to authenticate a user. The identity has to be managed for an entire session such that the user should have to login only once. This is usually accomplished by making use of Single Sign-on (SSO) solutions.

Auditing is a requirement which helps us to detect security vulnerabilities and respond with suitable measures. E-learning system should be configured to track messages between services and generate usage logs during specific periods of time. This audit serves as an important to investigate problems and diagnose potential security weaknesses. Compliance is essential to protect the right data longer, retrieve it as needed, and also discard it when it’s obsolete.

Security rules, policies, procedures and guidelines help to maintain and address all aspects of information security. It also encompasses all the SOA participants i.e. clients, services, processes and infrastructure.

2.7 Proxy

Proxy [12] is a service which translates service calls between two different client-service protocols. For example, there is a a service that is only accessible via RMI. It is needed to access this service via SOAP interface. To solve this problem a service proxy which receives SOAP calls and translates them into their corresponding RMI calls to the Java RMI service. Fig. 3 shows a client calling a service proxy via SOAP. The service proxy then calls the Java RMI service. The response from the Java RMI service is translated into SOAP by the service proxy, and returned to the client.

2.8 Caching

Caching [13] improves the speed to access both service state and application data, while enabling to scale the services out at the same time. Caching achieves this by minimizing the amount of traffic and latency between the services which use the cache and underlying data providers. To reduce the response latency of the entire e-learning solution, it is proposed to utilize a high-performance caching system to use in tandem with the data services or in the data base layer.

3. LITERATURE REVIEW

Several research papers have been published on E-learning systems recently using Web services and SOA. Some of the research papers that are reviewed and discussed are given below:

Maria Grazia Fugini et.al [15] defined a service-oriented model and architecture for security in multichannel information systems, where services can be delivered on various channels adaptively, including mobile systems channels. The security model considers security as a *Quality of Service (QoS)* parameter, according to the definitions given in the MAIS Project. A security architecture is presented, based on the defined model, based on the definition of *service delivery channel* given in MAIS and presented how the security architecture is framed.

Xufei Zheng et.al. [16] discussed how cloud based cloud security model can effectively coexist with traditional scanning technologies, and what are the advantages and limitations of this new approach. This model combines local-host based detector in host agent with multiple detection engines in the cloud. This model enabled detection of malware by multiple detection engines in the cloud in parallel. To explore and validate the idea, constructed a prototype which included a lightweight host agent, multiple detection engines in the network, and an AIS-based detection engine. Finally, they evaluated the performance and efficacy of the system using a dataset of 1500 malware samples through Arbor Malware Library (AML) covering a one year period.

Ian Zhang et.al. [17] designed a Web Services-based security model for digital watermarking; the watermark embedding and watermark detection technology components will be the part of the Web Services of the site. The entire system architecture is based on Web Services via SOAP service requester and the exchange of information between service providers, and uses digital certificates, XML encryption and digital signatures and other security technology to ensure information exchange security. The model proposed for digital watermarking and Web services combined with a certain reference value, it also works in the marketing of multimedia network environment effectively protect digital products.

Lingling Han et.al. [18] consisted of four lays: infrastructure layer, data layer, management layer, and service layer. The structure and function of each layer are described in great detail. The new digital library platform can be used to solve the problem of library resource storing and sharing effectively, and provide fast, safe, convenient and efficient services to users.

Marieta [19] proposed architecture is based on a set of replicated services running in a fault-tolerant container and a proxy service able to mask possible faults, completely transparent for a client. The solution not only masks possible faults but also optimizes the access to the distributed services and their replicas using a load-balancing strategy, whilst ensuring a high degree of scalability. The advantages of the proposed architecture were evaluated using a pilot implementation. The obtained results prove that the solution ensures a high degree of availability and reliability for a wide range of service-based distributed systems.

Palanivel [20] design service-oriented reference architecture for personalized e-learning systems (SORAPES) and validate the architecture. Some of the existing e-learning architectures that serve as the domain model for personalized e-learning system were considered, discussed in details and finally proposed a new architecture which is called SORAPES. The SORAPES was designed by re-using web services and learning objects. It is a layered architecture and highly-scalable for personalized e-learning system.

4. ARCHITECTURE: RAPESPAC

This section presents the architecture of RAPESPAC using Web Services and SOA. RAPESPAC allows content creators/ providers to easily share their archives through the secured Web interface or with application. It also the administrator to define the way data should be rendered to the end users. User interfaces to SOA-based systems need to submit requests (typically in XML over the Web) and parse and reformat or transform responses (typically also in XML) in order to generate portions of the Web interface. Fig 3 shows the overall architecture of RAPESPAC.

Web Service requests coming from outside into your network are secured by SOA Security. Alternatively, a user may also access the E-learning Portal Server, which in turn makes a Web Service request to a Web Service. Web Services deployed within an enterprise can also make requests to each other as part of a particular business process. Common central Policy Server secures both Web Service traffic and Website traffic used together.

4.1 Requirements

The requirements of RAPESPAC are functionality, performance, availability, scalability and maintainability [25] and they are discussed below.

- *Improve Throughput*: The functionality is improved by maintaining a caching service. The Caching service manages the instances of learning objects and improves the performance of the server. Maintaining a copy of an learning object in memory and/or on local disk can improve response time by avoiding the cost of finding or creating the object on each request. It improves throughput by making more efficient use of resources and improve scalability by making more copies of the learning objects available thus avoiding centralized accesses.
- *Increase Performance*: The performance is increased by many times faster than retrieving objects from the database. The performance will vary significantly depending on the cost of creating or writing the object. The most common uses will likely be managing learning objects created from data in the database. The goal is to have the retrieval from the cache to be many times faster than retrieving from the database even in the simplest case.
- *Improve Availability*: To improve availability, we can maintain local copies of learning objects using caching server. The cache service itself should not degrade the availability of the server using it.
- *Scalability:* The scalability of a server can be improved by using the cache to manage multiple copies of an object across multiple processes and nodes.
- *Maintainability:* The logging facilities help to trace and identify the problems that should occur in the cache or. The logging can be integrated with logging functional/service provided by the server. This logging facility is intended for use within the caching service not as a general logging service for application use.

4.2 Web Services Identified

The services identified in RAPESPAC are security, policy, notification, proxy and caching, etc., are presented below. All the identified services are registered in the UDDI registry. The service registry provides a central place where all service definitions are stored. A service registry also provides means to publish services for other developers to discover and reuse. This improves service reuse.

- *Security Service*: The Security service (Security.WSDL;) is a necessary for privacy. To enhance customer data privacy and to fulfill requirements of various stakeholders in service compositions, privacy is established as a trusted third party in the interaction between a learners and composite services [21]. Also dependability is one of the requirements for services running in large scale distributed systems.
- *Policy Service:* The Policy service (Policy.WSDL;) discovers the existing resources and their dependencies by using existing repositories in an distribute environment. This includes any metadata that is associated with the resource. Policies can be defined based on identity, service or resource, environment, and business.
- *Proxy Service:* The Proxy service (Proxy.WSDL;) route messages to application services and exposed to service consumers. It is configured by specifying its interface, type of transport, and its associated message processing logic. It routes the inbound and outbound messages through proxy server. It can perform distinct operations such as transformation, enrichment and logging etc.
- *Personalization Service:* The Personalization services (Personal.WSDL;) [24] help the users to find information resources available in a digitally chaotic world. There is a demand for learners or students instruction on the use of learning systems. They should help users to formulate disciplinary search strategies and provide assistance in developing new digital information resources.

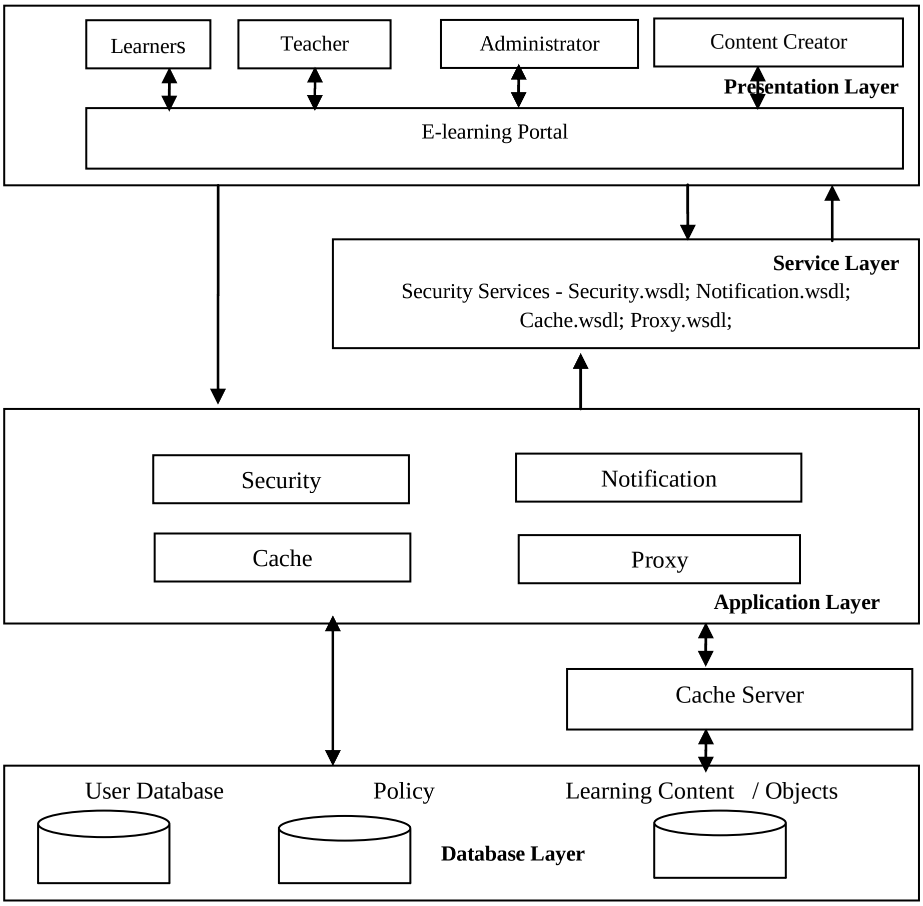

Fig 3: Overall Architecture of RAPESPAC

- *Registration Service:* The registration service (Register.WSDL;) registers or unregisters learners and instructors for a virtual learning. The service also updates the existing registration information.
- *Reporting Service:* The Reporting service (Report.WSDL;) monitors and report on SOA using logging function.

The above identified services are registered in the services registry. The registry manager can either start/restart or stop the services.

4.3 Security Module

Security component offers authentication, authentication, access control and confidentiality [25] and structure is shown in Fig 4.

- The *authentication* handles the authentication requests sent to the E-learning portal. It is passed log-on credentials for a user and returns an authentication certificate.
- The *Authorization* follows authentication, once a user or system has be authenticated. Authorization means making a decision about whether an authenticated or even an authenticated identity is allowed to access a resource.
- The *confidentiality* is for ensuring non-disclosure of sensitive information traveling through untrusted communication networks or at restInformation at rest includes security, user, and application information. It is commonly rely on ton cryptographic techniques such as encryption.

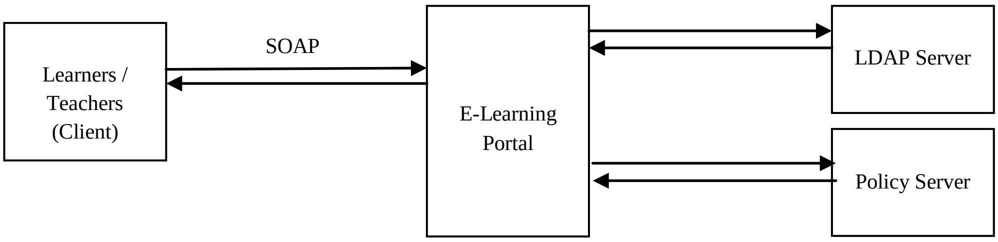

Fig 4: SSO on E-learning Portal to Internal/External Web services

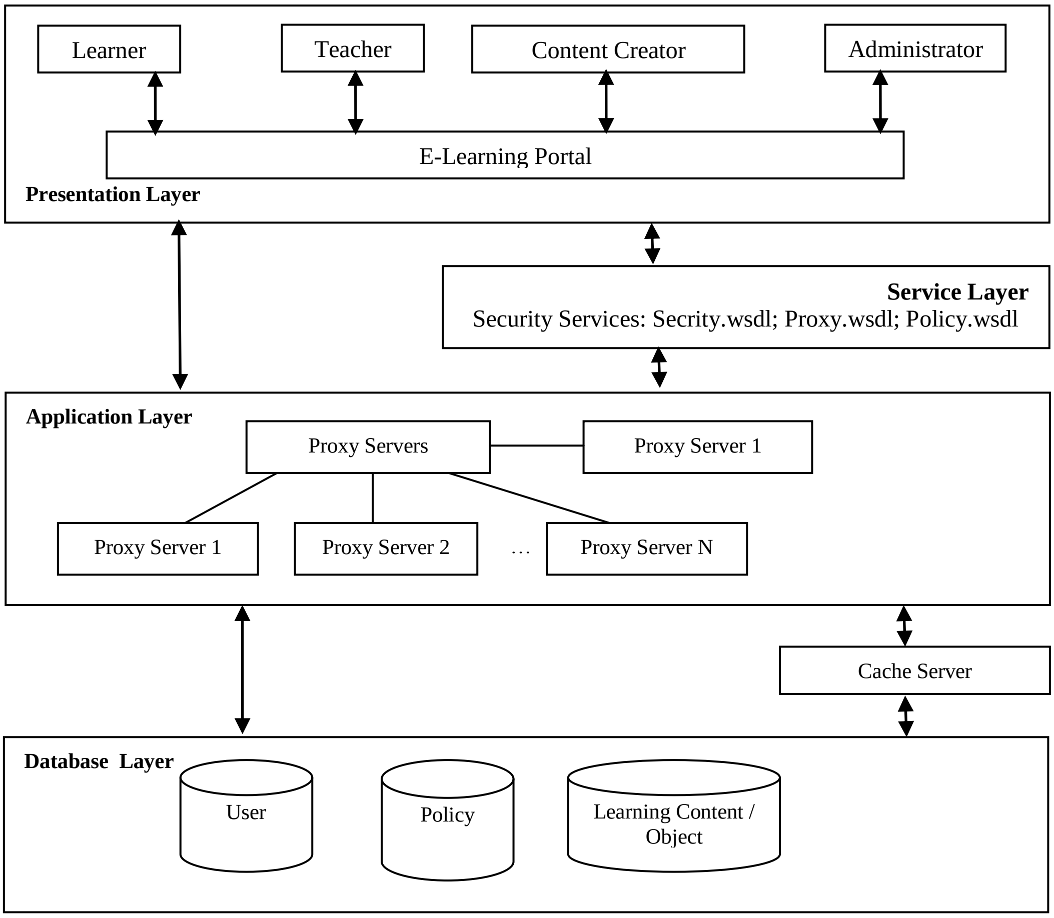

Fig 5: Structure of Proxy Module of RAPESPAC

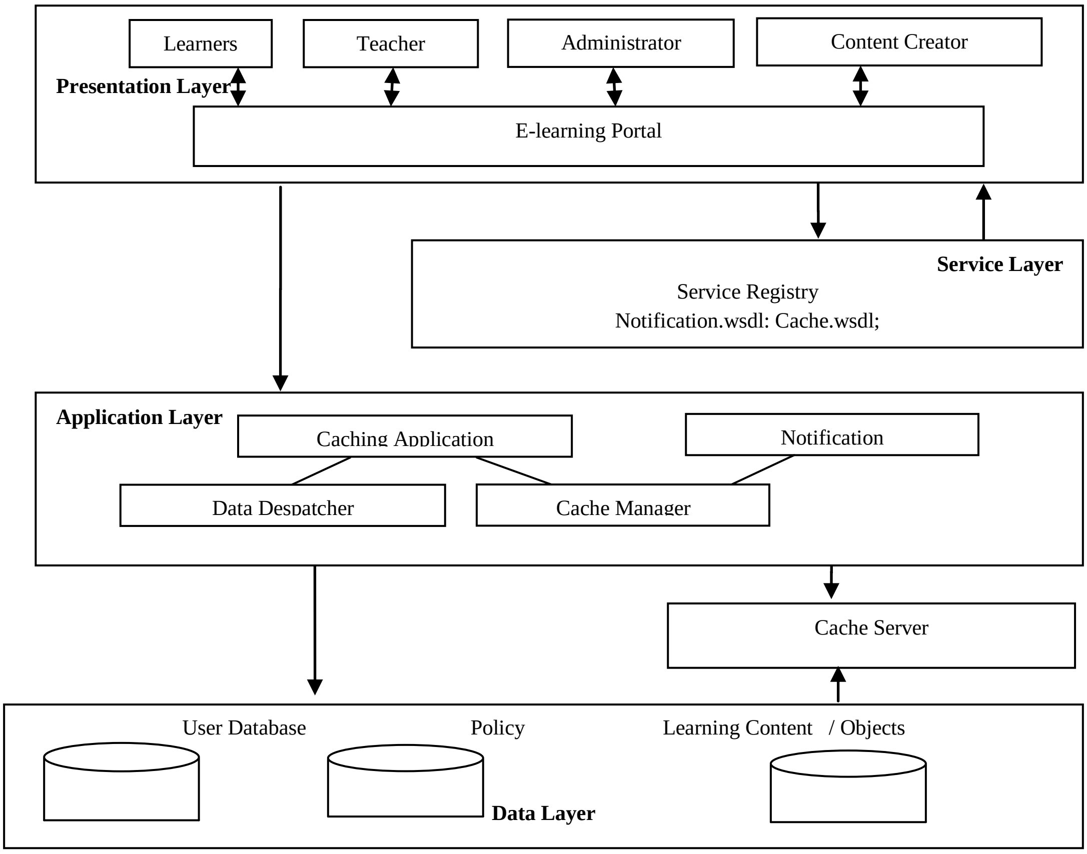

Fig 6: Structure of Caching Module of RAPESPAC

4.4 Single Sign-On Module

The Single sign-on (SSO) can permit a user to access all computers and systems where he has access permission, without the need to enter multiple passwords. Single sign-on reduces human error, a major component of systems failure and is therefore highly desirable but difficult to implement. Fig. 4 shows the SSO implementation from E-learning Portal to Internal/External Web services. The SSO manages access to RAPESPAC with a single log-on per user. The procedure of SSO is presented as below:

a. The user logs into the e-learning portal secured with user id and password to view the learning objects/ contents or upload the document.
b. The E-learning Portal-based application makes a SOAP call to the security service using the user's security context.
c. The user's session gets validated and authorized by the LDAP/Policy Server that is protecting security service.
d. The Security Manager then generates a WS-Security/SAML token and adds it to SOAP Header of request for the next step in the Web Service, which in this example is to the credit check Web Service.
e. The Security service sends SOAP request with SAML token to the other service provided by the provider.

The Security service authenticates the requester using WS-Security standard and provides response or denial to the user.

4.5 Proxy Module

Web Service requests coming from outside into your network are secured by SOA Security. Alternatively, a user may also access the E-learning Portal Server, which in turn makes a Web Service request to a Web Service. The effective way to protect the security of core systems is to avoid letting anyone reach the service hosting platform. This can be done by deploying a proxy for the web services within SOA. A proxy can receive and respond to all web service requests on behalf of the actual web services, and is therefore protected from malicious intent. The proxy also inserts the authentication and authorization SAML assertions into the message, thereby eliminating the need for the actual web service instance to query the security system directly. The structure of Proxy module in RAPESPAC is presented in Fig 5.

The proxy service can enable authentication and authorization when accessing a service. The proxy service between the service consumer and the original service gains the transparency. This can monitor and supervise the service and control the inbound and outbound messages. This becomes important when changes happen. When a service interface or the payload changes, the proxy service can mask the changes to all service consumers that have not yet been upgraded to use the new version.

The proxy service contains the message processing logic and will be used to decouple service clients from the service provider. The proxy service validates the request against the corresponding XML schema. It performs error handling and alert the service administrator of any problems with the service execution.

4.6 Caching Module

The caching module is used to cache only a subset of the data that is in the database based on the services. A cache server can give an SOA application a significant scalability boost It will performance and scalability of the service layer dramatically as it shall save a large number of time consuming database trips. The need for database synchronization arises because the database is really being shared across multiple applications, and not all of those applications have access to your cache. If E-learning service application is the only one updating the database and it can also easily update the cache. Fig 6 shows the structure of caching module in RAPESPAC.

The basic logic to implement caching, before going to the database is first checks to see if the cache already has the retrieved data. If it does, take it from the cache. Otherwise, go to the database to fetch the data Put it in the cache for next time.

There are significant advantages to using caching to improve the network performance. It reduces the bandwidth consumption because fewer requests and responses go over the network . It reduces E-learning portal server load because there are fewer requests for e-learning servers to handle. It also reduces latency because responses for cached requests are available from a local instead of from across the Internet.

4.7 Implementation and Evaluation

For pages other than the first page, start at the top of the page, and continue in double-column format. The two columns on the last page should be as close to equal length as possible

The system is focused to create security model using web service. Here, the main role or service is of central LDAP server. E-learning Portal server provides single window access to the user. User interface is based on the keywords related to meta-data. All the users login into the library portal. The security checks the user whether authorized or not. The authorized user sends a query in any language to the central server. The Portal server translates the query to the requested languages. Portal server passes query to the searching service. The Searching service checked with indexing services. Searching service do the ranking on the outcome of the indexing service. Searching services passes the response to the central server. The Portal server reformulates the ranked response according to the format expected by primary servers. The Application servers execute query provided by the central server and return results. The Portal server returned results to user after translation, summarization, filtering etc. of the content as required or the link to the content.

The performance is an issue in terms of computation and communication overhead. The proxy will introduce additional complexity and latency in Web service transactions. However, if proxy is arguably a service that should not be given for free, then the overhead is acceptable as long as it is minimized.

5. CONCLUSIONS

This paper discussed the reference architecture for e-learning systems using proxy and caching web services. RAPESPAC uses proxy and cache provides visibility and control of e-learning application interaction. Adding policies to RAPESPAC makes more consumable and accelerating the adoption of SOA solutions. The proxy reduces the load on the enterprise's security infrastructure. It reduces the traffic by centrally managing and caching of web service requests.

Nowadays storage plays an increasing role to address storage challenges efficiently and with flexibility. The storage offers a managed service which has several benefits. Optimized storage can scale effectively, coupled with security and management tools. Hence the future work seeks to develop reference architecture using storage as service for the service provider to enable it to deploy the offering for its customers.

6. REFERENCES

[1] M. Papazoglou, (2008), Web Services: Principles and Technology, Pearson, Prentice Hall.

[2] William Osei-Poku, (2009), Service Oriented Architecture, http://works.bepress.com/raspino/4.

[3] Reference Architecture for Service Oriented Architecture Version 1.0, (April 2008) Public Review Draft.1.

[4] K. Palanivel, S. Kuppuswami, (June 2011), Service-Oriented Reference Architecture for Personalized E-learning Systems (SORAPES). International Journal of Computer Applications, 24(5):35–44.

[5] Service-Oriented Architecture and Web Services: Concepts, Technologies, and Tools, Sun Developer Network, April 2005,

[6] W3C. SOAP Version1.2. Part-1: Messaging Framework, Second Edition. http://www.w3.org/ TR/ soap12-part1/

[7] W3C. SOAP Version1.2. Part-1: Messaging Framework, Second Edition. http://www.w3.org/ TR/ soap12-part1/

[8] W3C, Web Services Description Language (WSDL) Version 2.0 Part-1. http://w3.org/ TR/ EC-wsdl2020070626/

[9] W3COASIS UDDI Version3.0.2 UDDI Spec Technical Committee Draft. http://oasis-open.org/ pubs/ uddiv3.html.

[10] Steve Vinoski, (2004), Web Services Notifications, IEEE Internet Computing, Vol.892, 86-90.

[11] Ian Thurlow, Alistair Duke, John Davies, Applying Semantic Web Technology in a Digital Library, Demos and Posters of the 3rd European Semantic Web Conference (ESWC 2006), Budva, Montenegro, 2006.

[12] Web Services Security, (Apr 2002), IBM Developer Network.

[13] Zhendong Ma, J¨urgen Manglery, Christian Wagner, Thomas Bleier, (Aug. 2011), Enhance Data Privacy In Service Compositions Through A Privacy Proxy, Sixth International Conference on Availability, Reliability and Security (ARES 2011), 615 - 620 .

[14] Jerry Bortvedt, (2011), Functional Specification for Object Caching Service for Java (OCS4J), 2.0.

[15] Understanding SOA Security Design and Implementation (2007), IBM Redbook.

[16] Maria Grazia Fugini, (2005), A Security Model and Architecture for Multichannel systems, Dipartimento di Elettronica e Informazione, Politecnico di MilanoPiazza L. da Vinci, 332 I-20133, Italy,.

[17] Xufei Zheng, Yonghui Fang, (Aug. 2010), An Ais-Based Cloud Security Model, International Conference On Intelligent Control And Information Processing (Icicip), 153 – 158.

[18] Zhang, Jidong, Yin, Qun, (2010), Research Of Digital Library Architecture Based On Semantic Grid, International Symposium On Information Engineering and Electronic Commerce (IEEC), 1-4.

[19] Lingling Han and Lijie Wang, (2011), Research on Digital Library Platform Based on Cloud Computing, Communications in Computer and Information Science, Vol.214, 176-180, 2011.

[20] Zheng Qiaoying, Chen Zhaoneng,Bai Xuesong, (2003), Research on the Application of Notification Service for Service-oriented Digital Library, Shanghai Jiao Tong University Library, Shanghai, China.

[21] K. Palanivel, U. Nagalingam, (Nov. 2011), Service-Oriented Reference Model to Security for Digital Library (SORMSDL), edited by B. Ramesh Babu and P. Nageswara Rao, “Information Security in the Digital Era”, Society for Electronic Transactions and Security, 230-243.

[22] George Buchanan, Annika Hinze, (2005), A Generic Alerting Service for Digital Libraries, 7–11.

[23] Yin Zhang, Jiangqin Wu, Yueting Zhuang, (2004), The Personalized Services in CADAL Digital Library, the College of Computer Science ZheJiang University,HangZhou, China.

[24] K. Palanivel, V. Amouda, S. Kuppuswami, (2009), A Personalized E-Learning Using Web Services and Semantic Web, Int. Journal of Computer Engineering and Information Technology, Winter Edition 21(5), 27-37.

[25] Marieta Nastase, Ciprian Dobre, Florin Pop, Valentin Cristea, (2009), Fault Tolerance using a Front-End Service for Large Scale Distributed Systems, Inter. Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 229 – 236.

[26] Vivek Jonnaganti, 2009, An Integrated Security Model for the Management of SOA, Improving the attractiveness of SOA Environments through a strong Architectural Integrity, Master Thesis, Department of Applied Information Technology, University of Gothenburg, Sweden.