International Journal of Cloud Computing and Services Science (IJ-CLOSER)
Vol.x, No.4, October 2012, pp. 177~182
ISSN: 2089-3337

Enterprise Architecture Frameworks and Services for Cloud Computing

P.Deivendran*, Dr.E.R.Naganathan**
* Departement of Information technology, Velammal Institute of Technology, Chennai, India.
** Departement of Computer science and Engineering, Hindustan University, Chennai, India

Article Info

Article history:

Received July 20th, 2012
Accepted Sept 09th, 2012

Keyword:

Enterprise Architecture
SOA
Cloud Services
Framework
Registry

ABSTRACT

Commercial Services for provisioning software components and virtual infrastructure in the cloud are emerging; this creates a multitude of possibilities for outsourcing part of the Service to third parties in order to run their applications. These possibilities are associated with different running services so cloud customers have to determine the services. In this paper, we present to create new services. We assume that applications are described as templates, fixing the deployment service and hardware components. Different parts of the application may be outsourced to different providers and several levels of outsourcing can be considered. Our services are decomposes the application in order to discover all suitable Cloud services from a registry. We present architecture for achieving SOA level, which enables user to run the Services and other SOA framework as well as provides an environment to build multiple applications.

Corresponding Author:

P.Deivendran,
Departement of Information Technology,
Velammal Institute of Technology,
Chennai - Kolkatta High Road, Panchetti, Thiruvallur District. Pincode - 601 204, India
Email: deivendran77p@yahoo.com

1. INTRODUCTION

Software as Service (Saas)[2]. Has been proposed as a solution to this problem, where users can outsource their complete application to a third party to manage and maintain. For example, with Google Apps, document handling and internal web sites to Google, providing SaaS for applications such as Email has been realized by now to a great extent, but same is not true of outsourcing Service oriented Architecture(SOA)[3]. Service providers of infrastructure to run customer artifacts are commonly referred to as Platform as a Service (PaaS) provides, PaaS is about offering a platform that customers can deploy their components that in turn either offer services to others or used to build SaaS applications. When a PaaS provider offers a platform to customers, they need to ensure complete isolation between the customer and provider Service Level Agreement(SLA).

This is critical as often by providers simply offer a Physical machine with the appropriate software installed per the customer's needs. This does not provide architecture for scaling to deal with burst loads not offer any approach for sharing. The Second generation of providers uses operating system level virtualization to create virtual machine per customer that shares a single server or a cluster of real servers. For example Amazon Relational Data Service, a machine running as instance of MySQL specifically for the customer. Motivated by the success of SaaS have been many efforts to build multiple frameworks.

This paper is organized as follows: section II discusses related work in the field of Services in general and SOA in particular. Section III defines goals of a SOA platform. Then section IV presents our SOA platform and its implementation. Section V presents some performance evaluations of the proposed platform. Section VI reviews the security aspects of the proposed platform, Section VII delas with Service deployment, Section VIII described the .NET service operation, Section IX introduces the Token generation

and scalable for hosting the services and Section X wraps up the paper and identifies services of further problems that are yet to be extented.

2. RELATED WORK

Initial work was done in the context of Application Service Providers (ASPs). The ASP context had goal very similar to the goals of the SaaS model and motivated. There are four main approaches for SOA deployment, software program, APIs, Runtime, Operating system. Among these the first approach software programs we need a development environment with which to program and assemble the Software program [5]. It provides development for tools. The second approach, we need API's that expose features and functions offered by the runtime that we can build our software program to interact the function. The third approach, we need a runtime for which we will be designing our software [2]. Among these, the final approach scale our best we need on operating system on which to deploy the runtime, API and the software program. The operating system interface with underlying hardware. For instance level-1 provides an instance per tenant, Level 2 provides a configurable instance per tenant, level 3 runs a single instance that serves all customers, and finally, level4 enables to scale up by running multiple instances and load balancing to scale it up.

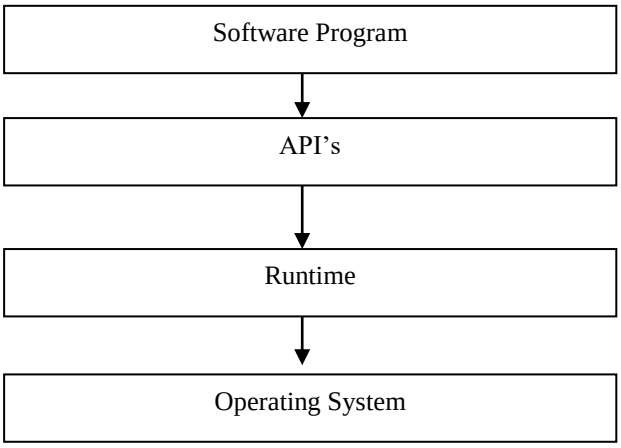

Figure 1. Fundamental software Architecture layers

3. MULTIPLE MESSAGE PROCESSING AND BUSINESS LOGIC IN SOA

To understand the SOA applications include message processing logic is performed by a Combination of runtime services Execution, service agents, security services, service logic released to the processing of Web Service Description Language (WSDL[7]). The back end part of a web service that performs in response to the receipt of message business logic application specific and can change in scope depending on the functionality by the WSDL. We group above services under Execution invoke developing and running involves deploying service which are often implemented as Web Services and composing those services together to create Business process. Security Service defines the ownership and authorization of both data as well as executions in the framework [3, 4]. An organization that registered as a tenant should be able to manage/administer its own users and services. Only the administrations of the infrastructure will be able to access these functions in order to help maintain & operate the running SOA framework.

4. PROPOSED ARCHITECTURE

The session describes the proposal architecture to implement Enterprise Architecture for SOA [7]. In larger IT environment, the need to control and direct IT infrastructure is critical. When numerous, disparate application architecture co-exist and sometimes even integrate, the demands on underlying hosting platforms can be complex [6]. Continuing our previous analogy, an enterprise architecture specification is to an organization what an urban plan is to a city.

Therefore, the relationship between an urban plan and the blueprint of a building are comparable to that of enterprise and applicati0on architecture specifications, Typically, changes to enterprise architecture directly affect application architecture [4], which is why architecture specifications often are maintained by the same group of individuals.Enterprise architecture often contains a long-term vision of how the

organization plans to evolve its technology and environment. Finally the technology and policies behind enterprise wide security measures [6]. However, these often are isolated into a separate security architecture specifications.

The Request-Response is the most popular MEP in use of among distributed application environments and the one pattern that defines synchronous communication [8]. The request-response MEP establishes a simple exchange in which a message is first transmitted from a source to a destination upon receiving the message the destination the responds with a message back to the source.

When client sends a message addressed to a particular service, the request must indicate the tenant in some manner. The default approach adds the tenant name in the URL as follows.

```
http://www.Xmltc.com\tls\service\service-name
```

Once the tenant is identified that message to the relevant tenant's-specific handlers and modules will be executed.

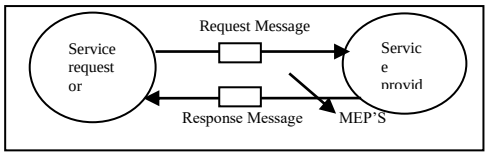

Figure 2. Request-Response using MEPS's

5. SERVICE DEPLOYMENT

Currently al service hosting components of JAX-WS and Apache services and BPEL processes have underlying execution engine [9]. The Business process Execution Language to demonstrate how process logic can be described as part of a concrete definition that can be implemented and executed via a complaint orchestration engine [6].

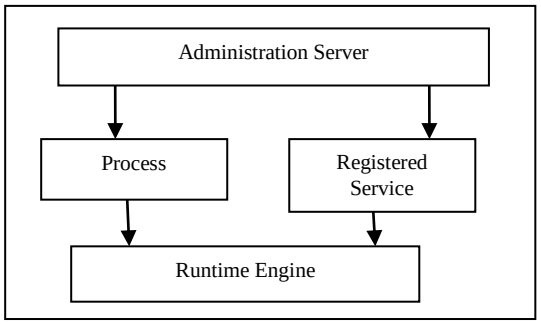

Figure 3. Service deployment Structure

This module handles HTTP(S) **Proxy**, This module handles HTTP(S) protocols and provides a standard web proxy interface for applications in the software cloud [3], such as using default proxy port 8000; it receives the outbound requests from SaaS application and hands them over to the rating engine.

5.1 Routing Engine:

This deployment receipt of server request, the engine will look up by routing table by the service URL, and find the next hop address, for no matched request the routing engine will reject it immediately [6].

5.2 Registration Server:

The registration server provides two instances. One is a secured web interface through the administrators can change the service, The other interface is different from a general web interface in that it requires clients certificate by authenticated[8]. The registered service and agents will be stored in a database, a run time copy is pushed to the routing engine for performance enhancement.

6. WEB AND WEBSERVICE SUPPORT

As web service is the most adopted interface for SaaS integration and HTTP(s) are commonly used for web service invocation, the support for HTTP(s) becomes a must [5]. The SaaS is capable of fully supporting both protocols of HTTP and HTTPS; proxy-based and decoupled architecture allows extension for new protocol.

7. BASIC BINDING SERVICE

```
<CONFIGURATION>
<System.servicemodel>
<behavior>
<Endpointbehavior>
<String> s1 assign <source>
<String> s2 assign <destination>
<%-- start web service invocation--%><hr/>
<%-- end web service invocation-- %><hr/>
</endpointbehavior>
</service>
<system.servicemodel>
</configuration>
```

I can write a client application to send messages through the .NET Service Bus on the same rendezvous address. The client configuration file will need to look just like the host configuration file. Only the endpoint definition will need to appear within the <client Section>. Other than that, it should look about the same, including the <transportClientEndpointBehavior> configured on the client endpoint. I can run multiple instances of the host application followed by a single instance of the client application.

8. CLOUD WORKFLOW SEQUENTIAL

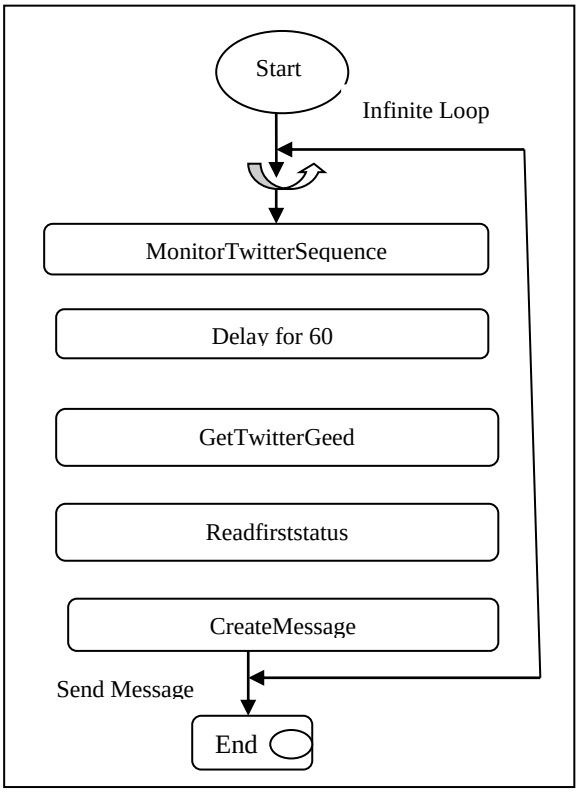

Figure 4. Cloud Workflow Design

The .NET Workflow service operates, let's build a workflow that consumes the public Twitter feed and sends the latest entry to any subscribed TweetNotifierApp instances through the .NET Service Bus. The first thing I need to do is create a new CloudSequential Workflow project in Visual Studio 2008. At this point you should see the traditional WF designer. If you expand the activity toolbox, you will see the restricted set

of activities that you are allowed to use. The public Twitter feed every 60 seconds. So first I drag the while activity onto the workflow design surface, and then specify that the condition with be a Declarative Rule Condition.

<DisplayTweet><text> status goes here </text></DisplayTweet>

I want to delay for 60 seconds, so I drag a CloudDelay activity into the sequence and specify 00:01:00 for the Timeout value. Then I need to retrieve the public Twitter feed using an HTTP GET request, so I place a CloudHttpSend activity right below the CloudDelay activity and configure.

9. AZURE SERVICES PLATFORM

Windows Azure and .NET Services require different invitation tokens, as you move towards cloud computing, workflow provides a simplified approach for coordinating complex service interactions in the composite cloud solutions you build. The .NET Workflow Service provides a scalable hosting environment for running and managing WF workflows in the cloud. Because the hosting environment is built on Window Azure, it's capable of scaling on demand, and because the WF runtime is being used, workflow instances are not pinned to any particular server-they are free to move from one server to another for each episode of execution. The .NET Workflow Service relies on a WF persistence service that leverages Microsoft SQL Services to save the state of running workflows and to ensure recovery capabilities, you build workflows for the cloud using Visual Studio and the same workflow designer that you've always use. You ultimately create XAML workflows and rules file. These XML files are then deployed to the .NET Workflow Service where they can be sued to create workflow instances.

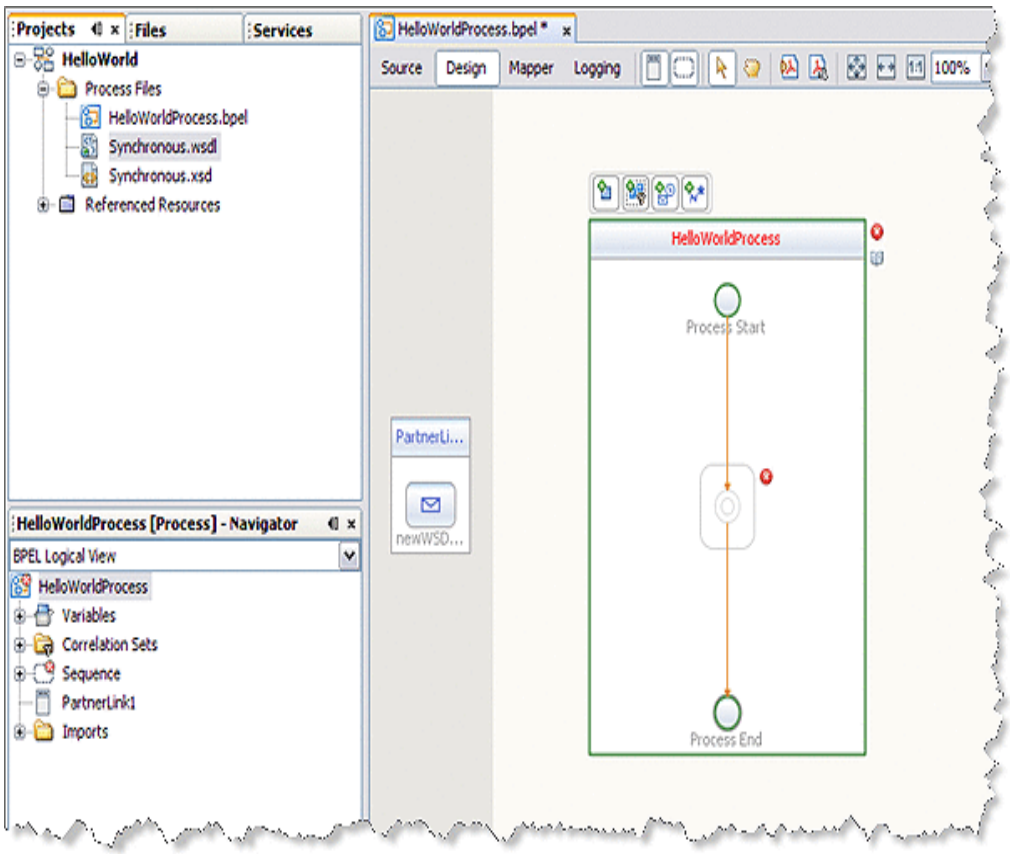

Figure 5. Sending Message

10. CONCLUSION

With the wider adoption of cloud computing efficient management of large number, diverse cloud resource relies on a flexible, scalable and robust information service. To enable the users in describing application oriented resource view, we present a resource view description language, simply request the service or update we want and it gets done for us transparently. Cloud computing can help us or at least

manage our infrastructure costs better and allow availability of our application and data over the Internet. One must be able to ensure that data is available anytime one needs it and having good software and a seamless connectivity to the service is essential in this regard.I believe proposed multi-tenancy platform will be a useful addition to architectures.

REFERENCES

[1] Immanuel Trumer, Frank Leymann, Ralph Meitner,. "Cost optimal Outsourcing of Applications into the Clouds". *2nd IEEE International Conference of Cloud Computing Technology and Science*, pp. 135-142, 2010.

[2] Dehui Zhou, Liang Zhong, Tianyu. "CloudView: Describe and Maintain Resource View in Cloud". *2nd IEEE International Conference on Cloud Computing Technology and Science*, pp. 151-158, 2010.

[3] Manuel Rodriguez-Martinez, Jaime Seguel, Melvin Greer Lockheed Martin, "Open Source Cloud computing Tools: A Case study with a Weather Application", *3nd IEEE International Conference on Cloud computing*", pp.443-449, 2010.

[4] Stefano Ferretti, Vittorio Ghini, Fabio Panzieri, "QoS-aware Clouds", "3rd *IEEE International conference on cloud Computing"*pp.321-327,2010.

[5] Balachandra Reddy Kandukur, Ramakrishna Paturi V, Dr. Atanu Rakshit, "Cloud Security Issues", *IEEE International Conference on Services Computing*", pp. 517-520, 2010. Vol. 265 No. 3, 1991, pp. 94–104.

[6] Daniel Ebneter, Stella Gatziu Grivas, Tripathi Uttam Kumar, Holger Wache,"Enterprise Architecture Frameworks For Enabling Cloud Computing", "*IEEE 3rd International Conference on Cloud Computing*".pp.542-543, 2010.

[7] Rui Zhang, Ling Liu. "Security Models and Requirements for Healthcare Application Clouds". "*IEEE 3rd International Conference on Cloud Computing*", pp. 268-275, 2010.

[8] Afkham Azeez, Srinath Perea, Dimuthu Gamage, Ruwan Linton, "Multi-Tenant SOA Middleware for Cloud Computing", "IEEE *International Conference on Cloud computing*", pp. 458-465, 2010.

[9] T. GU, H. Pung, and D. Zhang. "A service-oriented middleware for building context-aware services". *Journal of Network and Computer Applications*, Vol. 28, No. 1, 2005, pp. 1–18.

[10] Irena Bojanova , Augustine Samba," Analysis of Cloud Computing Delivery Architecture Models", '*Workshops of International Conference on Advanced Information Networking and Applications*" pp.453,458, 2011.

[11] Mohamed Wahib and Asim Munawar, Masaharu Munetomo and Kiyoshi Akama," A Framework for Cloud Embedded Web Services Utilized by Cloud Applications', "*IEEE World Congress on Services'*, pp.265-271, 2011.

[12] Tao Chen, Rami Bahsoon,"Scalable Service oriented Replication in the cloud", "*IEEE 4th International Conference on Cloud Computing*", pp.766-767, 2011.

[13] Sang-Ho Na, Jun-Young Park, Eui-Nam Huh, "Personal Cloud Computing Security Framework", IEEE Asia Pacific Services Computing Conference, pp. 671-675, 2011

[14] Jian Wu, Qianhui Liang, Elisa Bertino, "Improving Scalability of Software Cloud for Composite Web Services", "*IEEE International Conference on Cloud Computing*", pp.143-146, 2009.

[15] Keith R. Jackson, Krishna Muriki, Shane Canon, Shreyas Cholia, John Shalf,"Performance Analysis of High Performance Computing Applications on the Amazon Web Services Cloud", "*2nd IEEE International Conference on Cloud Computing Technology and Science*", pp.159-168, 2011.

BIOGRAPHY OF AUTHORS

Mr.P.Deivendran is doing Ph.D in Manonmaniam Sundaranar University; He received UG degree in Mathematics from Madurai Kamaraj University, PG degree in Information Technology from Sathyabama University. Now He is working as an Assistant Professor in the Department of Information Technology, Velammal Institute of Technology, and Chennai, India. He has 13 years experience in the field of Teaching and Research. He has published 10 National, International and Journals/Conferences.He has published one book. His broad interests include cloud computing and Service Oriented Architecture.

Dr. E.R.Naganathan is a Professor in Computer Science and Engineering, Hindustan University, Chennai, India. He received M.Sc. (Applied Mathematics) from Thiagarajar College of Engineering Madurai, India in 1985, M.Tech, from Manonmaniam Sundaranar University in 2008 and PhD from Alagappa University in Jan 2000. He has more than 25 years of teaching and Research experience in the field of Computing Sciences at different Institution in India. He visited Jordon and served as an Assistant Professor for two years in the Department of Computer Science, Jerash University. He has published 56 research papers in National and International Journals/Conferences. His area of interest is Information Security, Data Mining and Algorithms.