Volume 4, No. 4, March-April 2013

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

REVIEW ARTICLE

Available Online at www.ijarcs.info

SOA Appliances: An Embedded System Perspective

Deepali Tripathi*
Associate Professor,
MIMT, Kota, India
deepalidt@gmail.com

Yogita Punjabi
Asst. Professor
JECRC, Jaipur, India
yogita_gandhi2k3@yahoo.com

Abstract: Service Oriented Architecture, SOA, is a way of doing things by sharing resources. It is a concept not new to the enterprise software world, that can do big things in the embedded systems world also. This was previously considered too expensive or too proprietary. In this paper, we will show that in the coming years there will be ever increasing demand for Embedded SOA Appliances, that will perform the difficult tasks of enterprise integration, inter-operation and intercommunication. SOA Appliance can bridge the gap between complex SOA implementation and deployment and the effort to quickly move to SOA. The discussed appliances do a difficult job while hiding the complexities of SOA service design and maintenance. SOA Adapters will play an important role in such device and equipment integration efforts.

Keywords: SOA, Embedded systems, SOA Appliances, Integration Network, Multilayer Integration.

I. INTRODUCTION

Service Oriented Architecture (SOA) is a design paradigm for implementing distributed systems. It is primarily used to design reliable, distributed systems with inherent advantages of reusability, interoperability, scalability, flexibility, return of investment (ROI) etc. It comprises of a set of components as services that can be invoked and whose interface descriptions are published and discovered. Service is a unit of solution logic specially designed to attain the inter operability, federation, agility, loose coupling and other strategic goals of SOC [1]. It is a self contained stateless business function that accepts multiple requests and returns multiple responses through well defined standard interface. They are hosted by the service provider and are initially discovered and further invoked and interacted by the service consumer through UDDI (universal description, discovery and integration) service registry following the service contract as shown in figure 1[2]. Service orientation, services, service compositions, service inventory and SOA are essential elements for realizing SOC.

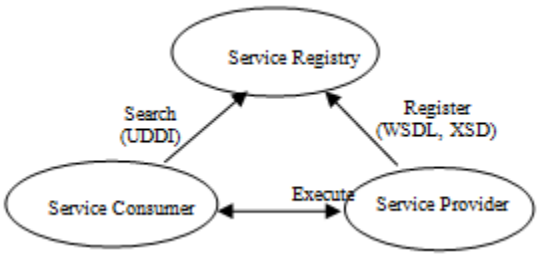

Figure 1: Service Oriented Architecture

As the name suggests, SOA, or Service Oriented Architecture is a technology architecture which focuses on building systems based on services. The term "service" is very broad in SOA. It does not necessarily mean software. In the definition itself, SOA does not limit the architecture to software system design only. In fact, it is now being used in hardware and embedded appliances design as well. All definitions of SOA leads to the same common conclusion as 'SOA is a design pattern which is composed of reusable and inter-operable platform agnostic services in which each of these services follow a well defined standard. Each of these services can be bound or unbound at any time and as needed' [2, 3].

Traditionally, SOA is attributed to large software systems like financial accounting, banking, health-care, insurance and so on, but there is no limit to what this architecture can do. SOA can bridge the gap between the very proprietary world of embedded systems and the world of enterprise software. As more companies look towards building integrated systems, SOA will begin to play one of the most important role in integration, inter-operation, interaction and intercommunication [3].

SOA itself is an architecture, not an implementation of the architecture. The implementation of this architecture is the popular term called Services or Web Services. Web Services technology approaches the design pattern of SOA by using standards based SOAP (Simple Object Access Protocol) XML protocol to communicate between services and uses WSDL (Web Services Description Language) for describing exposed web services. There is also another popular implementation called REST. While a few other implementation also exists, the basic goal of SOA remains the same.

II. TRUE SOA

The term 'A True SOA' is to build applications such that they are all inter-operable in any environment. The word 'environment' includes all of the following: operating systems, application servers, client execution environment, programming languages, mobile devices, embedded

systems, mainframes, security environment, databases and standards compliance [2,4].

a. ***Operating Systems*****:** Operating systems range from large mainframe operating systems like z/OS or z/Linux to tiny embedded operating systems like Embedded Linux. A true SOA implementation must support all of the operating systems.

b. ***Application Servers*****:** Application servers are servers that act as the host for Internet applications. Apache web server, Tomcat, WebSphere, WebLogic, and so on, are all application servers because they host Internet applications. SOA architecture is more appropriate for the Internet as it opens the world for integration and inter-operation. A true SOA implementation will not set a limit on where these applications may run.

c. ***Client Execution Environment*****:** A client of a system is the consumer of a SOA service. A consumer (or client) may be a mobile phone or a mainframe. In a true SOA implementation, there should be no limit to what kind of client can consume the service.

d. ***Programming Languages*****:** True SOA should not be a limitation to any particular programming languages. Any application written with the choice of a programming language should be able to consume a SOA service.

e. ***Mobile Devices*****:** In a broad term, a mobile device is any device that is mobile or portable. This ranges from mobile phone to toys. Mobile devices tend to be small and battery operated. True SOA implementations should not limit the consumption of services by these mobile devices.

f. ***Embedded Systems*****:** In the vast world of embedded systems, we have seen very large and powerful embedded systems implementation like those in aircrafts, to very tiny and inexpensive implementations in toys. In a true SOA implementation, SOA should scale to the size of the embedded systems in use and will not limit consumption of services based on the size of implementation.

g. ***Mainframes:*** Regardless of the size of mainframes, or when they built, a true SOA implementation will work on any of them. They can both host a SOA service as well as be consumed by these systems.

h. ***Security Environment:*** Security is one of the important topics in SOA as it is widely exposed as a service. True SOA must give equal consideration to all SOA consumers and must provide assured security when consumed. From embedded systems to mainframes, true SOA must deliver the same level of security across them all.

i. ***Databases*****:** There are as many databases as there are programming languages. Usage of databases are basically a choice of business decision makers as well as enterprise architects. An application must not be designed to only focus on a limited number of databases. Instead, it must be designed using a uniform database architecture. In true SOA, all databases must be equally visible through the same service.

j. ***Standards Compliance*****:** SOA must be standards based, as it is now, otherwise interoperability could not be guaranteed. Presently, many standards are defined around Web Services, which is basically an implementation of SOA in terms of the Internet using standards based XML protocol. When standards are created or modified, the implementation of these standards should not be limited to any of the other variables of the operating environment.

III. WHAT IS SOA APPLIANCE

A SOA Appliance is a self contained unit of software or hardware (or both) to perform a SOA based solution. One of the popular SOA Appliances example is IBM's DataPower, a hardware XML accelerator, converter and security gateway [5,6]. It has hardware XML acceleration, which allows XML transformation, messaging, to be done very quickly. The future of SOA Appliances will go much further than this. One of the basic reasons for growth of SOA Appliances is because of SOA itself. The following points illustrate some of the benefits of having a SOA Appliance [7]:

a. SOA is based on web services and usually, web services development requires thoughtful design and extensive testing. This leads to thousands of hours of development time. When the same is offered in a hardware box as an already completed appliance with the needed software ready to be deployed - significant time and effort have been saved.

b. SOA implementation requires other supporting software. For example, if the SOA is implemented using Java, then the minimum additional required software will be JVM and Java Application Server. There is a wide choice of Java Application Server and the vendor usually has to purchase it separately obtain a free one. Application server management is also an additional resource requirement. Having this all available as a SOA appliance eliminates this additional software.

c. Most SOA works over HTTP and HTTPS protocols. This leads to separate server and port management as well as network security management. There is also a need for certificate generation, signing and deployment. All these tasks require the involvement of a Network Applications Engineer. When all these features are bundled in a SOA Appliance, minimal human effort is needed to maintain the protocols and certificates.

d. Some SOA applications need to be portable and low powered for use in remote areas [8,9]. Examples of this would be, RFID using SOA technology, Outdoor security camera, and other types of monitoring and surveillance devices. A SOA appliance is definitely the right way to approach it.

e. If SOA is used in industrial or factory environments, the server is usually located in another location and PC's are usually needed to communicate with those servers[10]. By using a SOA Appliance you eliminate

the need for those PCs as the job can be done by this SOA device.

f. As an actual hardware device the SOA Appliance will execute much faster than if the same service is done on a server. This hardware device is built to consume very little power and designed to switch to power saving mode when idle making it a "green" option for future projects.

g. The SOA client consumes the exposed SOA services is usually small, but, mostly they come as a set of libraries or tools. As such, a developer will use these tools and libraries in their own applications. If the libraries are for multiple platforms and programming languages, there will be duplicated efforts for each of the supporting platforms. Having the SOA client libraries in hardware form requires no extra development time at all.

SOA Appliances will become widespread as SOA based software is adopted by industry.

IV. SOA INTEGRATION APPLIANCE

SOA is usually seen as one of the technologies that can integrated a diverse range of operating environments. Therefore, there will be a need for a SOA Appliance called SOA Integration Appliance [11,12]. A typical SOA Integration Appliance will have the following features:

a. It will be able to connect to any other known SOA appliance.
b. Any other SOA appliance can connect to it.
c. It is operating environment agnostic.
d. Any application written in any programming languages can use SOA Integration Appliance.
e. SOA Integration Appliance can act as a gateway to connect to another foreign system.

A. Single Layer Integration:

This is shown in Figure 2, a SOA Integration Appliance sites in the middle and any other SOA compatible device, appliance, applications, embedded systems, can use services from the appliance.

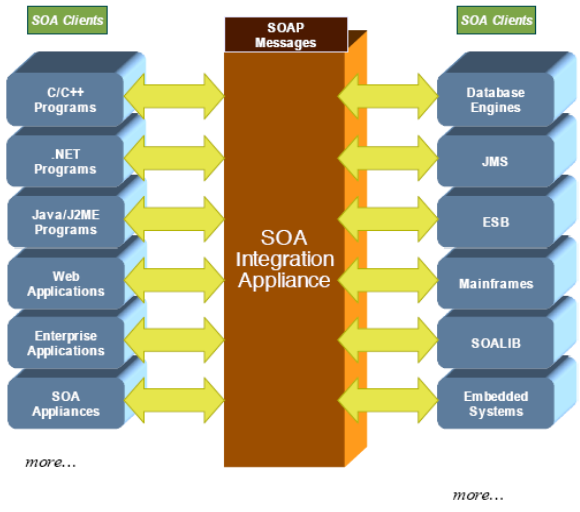

Figure 2: SOA Integration Appliance

Also, the SOA Integration Appliance acts as a link for all of the possible links that can be created, it can also serve applications.. This is an example of single layer integration in which all individual entities integrate through a single SOA Appliance. A single SOA integration layer is useful for a less distributed environment. However, when the units are geographically distributed and integration is required to isolate from the outside world through a firewall, multi-layer integration will be required.

B. Multi-Layer Integration:

In a multi-layer integration (Figure 3), portions of the integration are done inside a firewall or more generally, inside a black box. A black box may be a very large system, for example, it may be an entire mobile phone network. On the other hand, a black box may also be represented by another proprietary network, system or applications. The SOA Integration Appliance connects the two black boxes together using another SOA Integration Appliance. This make all SOA clients in each black box integrated. Each SOA Integration Appliance work through SOA so one integration appliance appears as a SOA client to the other. However, when integrating multiple SOA Integration Appliances to build an integration network, special SOA services will be needed for integration purposes as well.

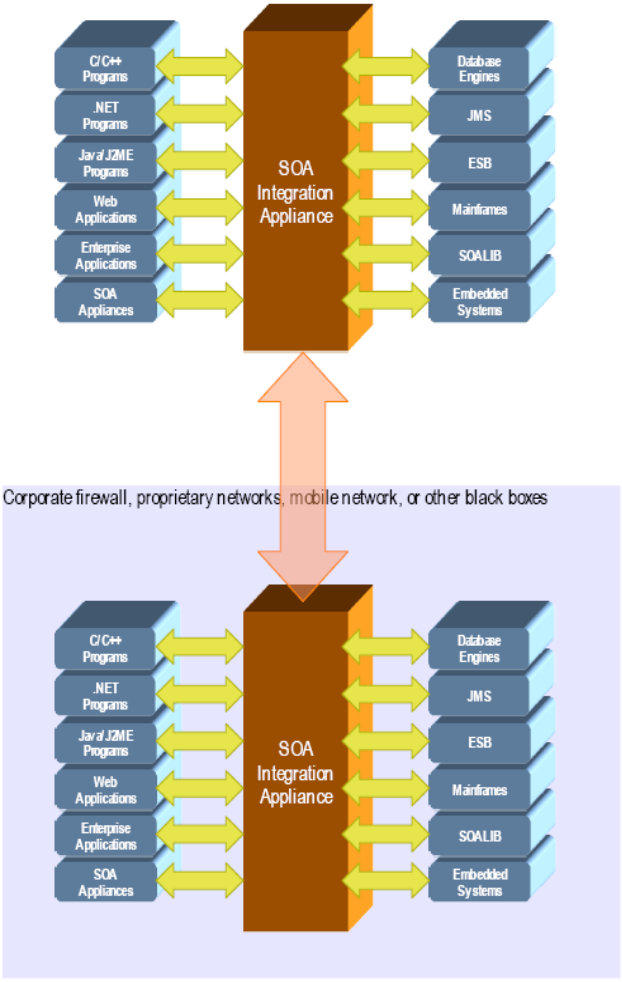

Figure 3: Multilayer Integration

V. SOA INTEGRATION NETWORK

By using multiple SOA Integration networks (Figure 4), everything may be fully integrated by using SOA.

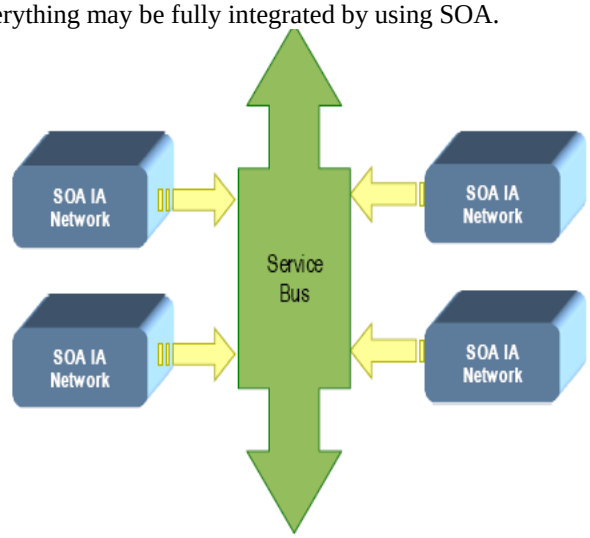

Figure 4: SOA Integration Network

In a SOA Integration network, all SOA Integration Appliances are inter-connected with their black box. Using SOA Integration Network mobile and corporate network, embedded systems devices, machines and equipment, application software, all become connected. Integrated applications reduce cost and increase efficiency.

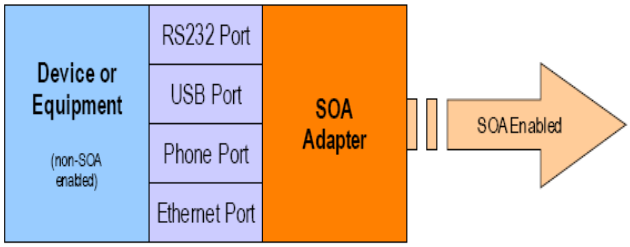

Fig5: SOA Adapters

To complete the integration network, a small hardware device called a SOA Adapter could be used. These SOA Adapters connect to one or more of the I/O ports that come with these devices to control and receive data from these devices. The SOA adapter (Figure 5) converts the information as SOAP packets and sends them to the SOA Integration Appliances.

VI. CONCLUSION

Service oriented computing is an emerging distributed computing technology that is set to replace the existing ways of building software. SOA Appliance can bridge the gap between complex SOA implementation and deployment and the effort to quickly move to SOA. These appliances do a difficult job while hiding the complexities of SOA service design and maintenance. SOA Adapters will play an important role in such device and equipment integration efforts. Multiple SOA client adapters are also available written in different programming languages. The adapters are primarily software based, but hardware SOA Adapters are due to be announced.

VII. REFERENCES

[1]. D. Tripathi, “Development Trends and Evolution of SOA”, *ETEMC*, April, 2010

[2]. D. Jana, “Service Oriented Architecture-A new Paradigm”, CSI Journal, March 2006, pp 12-15

[3]. IBM’s Vision For The New Enterprise Data Center, March 2008 at www.ibm.com/systems/ optimizeit/ datacenter/ pdf/NEDC_POV_MAR_2008_-_02.pdf

[4]. T. Erl, “SOA: Design Patterns”, 1st Ed. Prentice Hall, 2009.

[5]. Willy Chiu, vice president, high-performance Computing, IBM Software Group & IDG. http://www.cio.com/article/438371/Cloud_Computing_Hype_Versus_Reality?page=2&taxonomyId=1448.

[6]. Ioannidis J., D. Duchamp and G. Q. Maguire.”Ip-Based protocols for mobile internetworking.” Proceedings of ACM SIGCOMM Symposium on Communication, Architectures and Protocols.

[7]. Ejaj Jamil, “Embedded SOA Appliances”, January 2010, Oracle White Paper in Enterprise Architecture.

[8]. Chrysanthis, P.K. , “Transaction processing in mobile computing environment”, Proceedings of the IEEE Workshop on Advances in Parallel and Distributed Systems, 2003. .

[9]. Sumari, P. Darus, R.M. Kamarulhaili, H. “Data organization for broadcasting in mobile computing”, International Conference on Geometric Modeling and Graphics, 2003. Proceedings.

[10]. S. Moosavi, M. Seyyadi, “A method for Service Oriented Design”, 6th International Conference on IT, New Generations, 2009.

[11]. J.B. Hurson, A.R. Lim, “Transaction processing in mobile, heterogeneous database systems, Knowledge and Data Engineering”, IEEE Transactions ,Vol: 14 Issue: 6, Nov/Dec 2002.

[12]. Moss, 3. E. B. Nested Transactions: “An approach to reliable distributed computing”. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, April 2001.