International Journal of Engineering Works
Kambohwell Publisher Enterprises
ISSN: 2409-2770 Vol. 2, Issue 4, PP. 54-57, April, 2015
www.kwpublisher.com

# End to End Monitoring in Oracle Fusion Middleware for Data Verification

Kashif Ali, Usman Javaid

***Abstract*—** In large enterprises multiple departments use different sort of information systems and databases according to their needs. These systems are independent and heterogeneous in nature and sharing information or data between these systems is not an easy task. The usage of middleware technologies have made data sharing between systems very easy. However monitoring the exchange of data or information for verification purposes between target and source systems is often complex or impossible for maintenance department due to security or access privileges on target and source systems. In this paper, we are intended to present our experience of an end to end data monitoring approach at middleware level implemented in Oracle BPEL for data verification without any help of monitoring tool.

Keywords— BPEL, Oracle Fusion Middle ware, SOA, SLA, QoS.

## I. Introduction

Large scale organizations and enterprises are often divided in many kinds of different units & departments' i.e. human resource management, finance, marketing, customer relationship management, IT operations, and sales & distributions. Due to the diverse nature of work and size of these departments, they need different types of information systems and databases. The selection of such systems is based on the requirements and needs of each department which leads the organization's management to adopt or purchase different kind of solutions (information systems & databases) from different vendors. These solutions are often implemented in different technologies/languages like Java, .Net, Python etc.

and use different versions of databases & data warehousing tools i.e. Oracle, SAP, Microsoft SQL Server, Tara Data etc. Once these heterogeneous solutions are adopted by different departments in an organization these departments are required to exchange information in order to fulfill business needs.

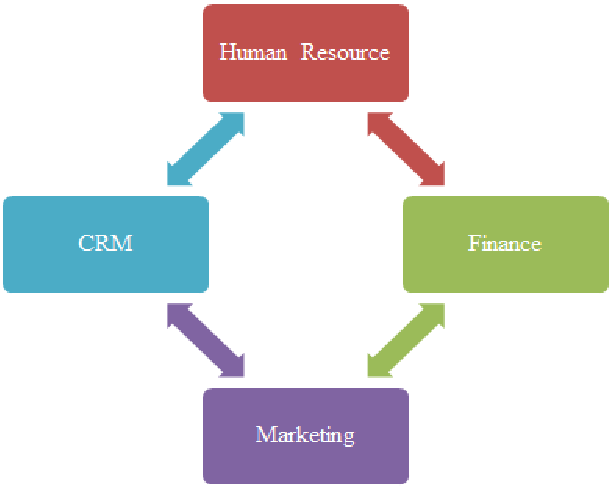

Figure 1: Departmental Info Sharing

The exchange of information between these departments is very complex task due to lack of interoperability issues between different systems used by these departments. However the usage of middleware technologies has made the exchange of data/information between different systems easy. Middleware is simply a software layer between that connects different software components and applications [1]. There are different sort of middleware technologies available in market like Oracle Fusion middleware, IBM Web Sphere and Microsoft's MOM (Message Oriented Middleware). Many of middleware technologies provide the framework for Service Oriented Architecture (SOA). SOA can be implemented with the help of web services but one can implement SOA within the mainstream distributed environments such as J2EE or .NET [2]. However web services are ideal for usage when applications are operating over internet where reliability and bandwidth cannot be guaranteed and mix of platforms & vendor products are used.

Although the data exchange from source system to target system can be done with the help of middleware. Monitoring of data at source or target system level is not an easy task. Different departments have sensitive data or information which is not available publically and required to be protected from other organizational departments and users. So verification of

Kashif Ali: Telenor Pakistan, syed.ali2@telenor.com.pk
Syedkashif1990@gmail.com
Usman Javaid: Telenor Paksistan, usman.javaid1@telenor.com.pk

exchanged data at target or source system level is limited or in some cases impossible. But the data being exchanged can be monitored at middleware level in order to verify that the correct data is transferred from source system to target system. There are many tools and technologies available for specific monitoring purposes like BAM, IBM etc. but implementing them is costly and required implementing overhead. Thus this paper is explaining an event based monitoring approach to verify data coming from source & destination systems. The paper is divided in to three further sections. Section 2 briefly explains the background of SOA and monitoring & middleware technologies. Section 3 illustrates the practical implementation of solution of E2E monitoring in oracle fusion middleware by using BPEL processes and finally section 4 provides the results and conclusion of our practical approach.

## II. Back Ground

SOA is basically a set of components that can be invoked over network/internet; the interfaces of these components are published and can be discovered. In SOA a component is defined as software or more precisely a service. A service is the building block of SOA, which provides a coherent functionality. A service can be composed of many other components or services. Some of the SOA advantages [1] are

**Reduction in development cost & time** Service are reusable and easily assembled in order to provide new composite services.
**Low maintenance costs** Reusable services reduce the number & complexity of services.
**High quality service** Increase in reusability improves services quality through multiple test cycles.
**Lower integration costs** Standardized services know how to work with other services, easy to connect.
**Risk Reduction** Reusability provides more control over governance policies which limits the risk.

Before a provider provides services to its consumer both parties need to be agreed on a SLA. A SLA defines Quality of Service (QoS) attributes (functional & non-functional) for a service, to be provided to its consumer. It also defines the conditions under the service will be provided to its consumers. In order to ensure satisfaction of the Service Level Agreement, monitoring is required at service provider end or at the service consumer end [3]. Basically, there are three main types of web service monitoring in SOA: Provider side monitoring (centralized or distributed), customer or end user side monitoring and third party or middleware monitoring. The drawback of provider-side and consumer-side monitoring is that in the case of problems neither side will trust the others [4]. The third party or middleware monitoring has bottleneck performance issues when the messaging between customers and providers is increased. But when the message frequency is limited, the middleware monitoring is a reliable monitoring approach than other two monitoring approaches.

As IBM [5] describes middleware as a software glue which helps different programs and databases running on different machines work together. The formal definition of middleware is “Runtime system software that directly enables application-level interactions among programs in a distributed computing environment.” There are some basic functions of a middleware

- Hide the distributed nature of the application & show system as one coherent system
- Hide the heterogeneity of the enterprise applications including operating systems & communication protocols etc.
- Provide high level interfaces and standards to application developers so that end to end solutions can be made easily and highly reusable

Oracle fusion middleware is one of the leading middleware technologies and widely used in industry. Oracle fusion middleware is basically set of products which include web servers, application servers, content management systems and different tools that used in application development, deployment and delivery of solutions. BEPL Process Manager (BPM) is one of the products in the oracle fusion middleware family. Business Process Execution Language (BPEL) is an XML based language it is used to explain business process from a high level point of view. Particularly speaking, BPEL describes all the features and the services provided by every web service that is a partner in the business process (executable process) [6]. BPEL enables multiple tasks sharing in a grid computing or distributed environment and it is a standard for assembling a set of discrete services in to an end to end process flow [7]. The service created in BPM are easily orchestrated, executed and optimized.

## III. Implementation

At Telenor Global Shared Services Pakistan, integration development team was assigned with a task of transferring employee’s data from Human Resource Information Management System (HRIMS) to a payroll system SAP for a specific business unit on monthly basis. A certain condition was imposed as the part of requirement; that business (finance) & system (I.T) analysts must be able to cross verify data from both source and target system. According to organizational policies direct access to the information in multiple systems is not allowed and in some special cases limited access for limited time is given to specific users. However with the imposed analysis condition, architects came up with a solution and decided to save specific data of HRIMS into analysis table as well as data sent to SAP. Fig 2 depicts the high level diagram of solution.

ISSN: 2409-2770

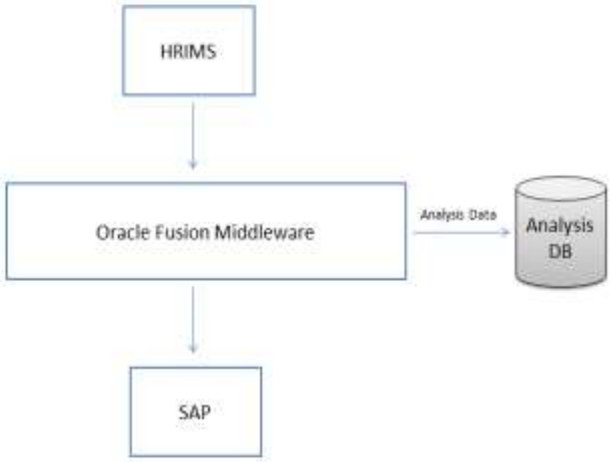

Figure 2: High Level Solution Diagram

The HRIMS provides a web service which sends data to oracle fusion middleware. Oracle fusion middleware receives the data, apply business logic and some simple transformation and send the data to SAP meanwhile middleware saves the received data from HRIMS and data sent to SAP in another database used for analysis purposes. As discussed in previous section, fusion middleware is a set of products; we have selected BPEL processes to implement this solution. Fig 3 illustrates the BPEL composite.

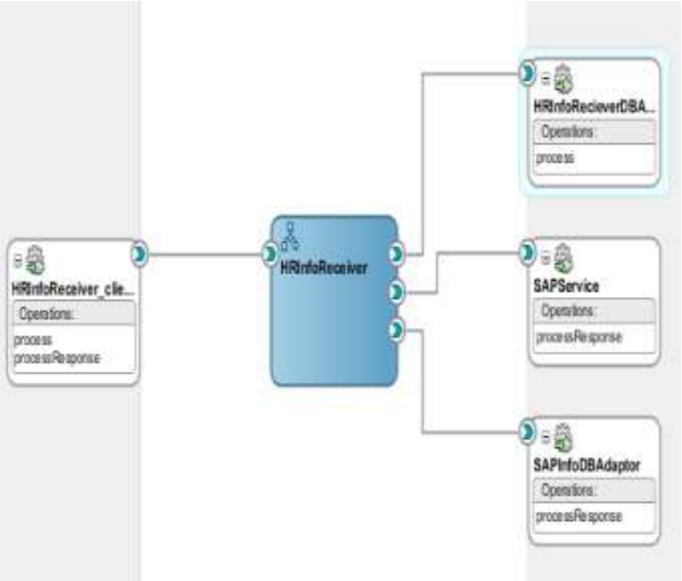

Figure 3: BPEL Main Composite

A BPEL process HRInfoReceiver exposed as web service HRInfoReciever_Client_ep, HRIMS uses this web service to send data. HRInfoReciever receives data from exposed interface and calls three web services.

- **HRInfoRecieverDBAdaptor**
- **SAPService**
- **SAPInfoDBAdaptor**

Fig 4 describes the process flow and the detail functionality of process flow steps are following.

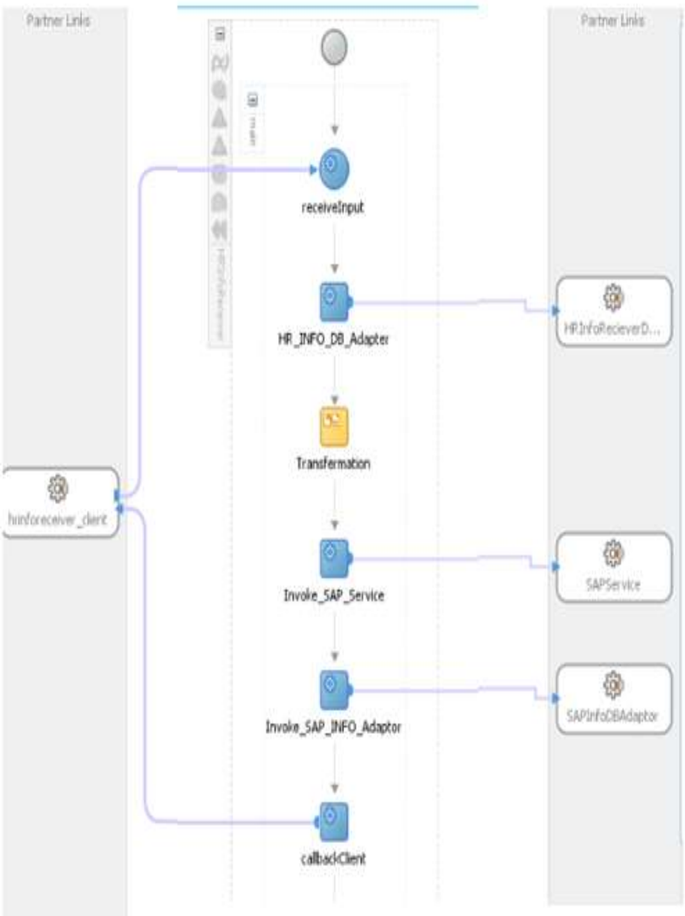

Figure 4: HR Info Receiver Activities

After HRInfoReceiver process receives data from HRIMS and invokes HRInfoRecieverDBAdaptor web service that saves data into the analysis database and then passes received data to transformation activity. The transformation activity transforms and maps data into SAP parameters.

SAPService gets the transformed data and pushes data into SAP system through invoke_SAP_Service. SAPInfoDBAdaptor SAPService sends back confirmation message and the same transformed data is sent to invoke activity Invoke_SAP_INFO_Adapter that calls the SAPInfoDBAdaptor which saves the data sent to SAP in analysis database and sends back a confirmation message.

## IV. Conclusion

In this paper we presented our encounter with an industrial problem where different departments share limited or no data at user level due to security or organizational policy reasons.

To overcome with such issue, an end to end monitoring approach implemented in oracle fusion middleware is presented for cross data verification without using any data monitoring tools. The basic purpose of our approach is to save two types of data at middleware level, data coming from source system and data going to target system after transformation or applying business logic.

## References

[1] http://docs.oracle.com/cd/E14571_01/core.1111/e10103/intro.htm#BABEICDD

[2] [Eugen Vasilescu, Seong K. Mun, "Service Oriented Architecture (SOA) Implications for Large Scale Distributed Health Care Enterprises," Proceedings of the 1st Distributed Diagnosis and Home Healthcare (D2H2) Conference Arlington, Virginia, USA, April 2-4, 2006.

[3] Philipp Leitner, "Monitoring, Prediction and Prevention of SLA Violations in Composite Services," in IEEE International Conference on Web Services, 2010.

[4] Zain Balfagih, "Agent based Monitoring Framework for SOA Applications Quality," in IEEE, 2010.

[5] http://www01.ibm.com/software/info/websphere/partners4/articles/gartner/garwho.html

[6] Marina Mongiello, Daniela Castelluccia, "Modeling and verification of BPEL business processes," in Proceedings of the Fourth Workshop on Model-Based Development of Computer-Based Systems, 2006.

[7] http://www.oracle.com/technetwork/middleware/bpel/overview/index.html