Towards Developing Failure Tolerant Communication Framework for GUISET Services

T. Shezi¹, E. Jembere², and M. Adigun³
Department of Computer Science
University of Zululand, Private Bag X1001, KwaDlangezwa, Empangeni 3886
Tel: +27 35 902 6393, Fax: +27 35 902 6569
Email: 1 mrtshezi@gmail.com, 2 ejembere@gmail.com, and 3 madigun@pan.uz.ac.za

Abstract- Enterprise Service Bus (ESB) is now a popular term in today’s information systems integration reflecting an implementation of Service Oriented Architecture (SOA). ESB act as an integration and communication platform for connecting heterogeneous services. There are many ESB solutions that have been implemented and available as both commercial and open source software. Within these solutions heterogeneous services exchange messages regardless of their nature, leading to great flexibility of information sharing amongst these services. However, with all messages going through the bus - the ESB can quickly become a single point of failure for an organization. To ensure that communication between services continuously exist in case of ESB failure, this work proposes failure tolerant framework that is based on federation of ESBs. We intend to adopt distributed service registries between federation of ESBs to assist in publishing, finding and binding of heterogeneous services, and content based intelligent routing to facilitate communication in the ESB federation.

Index Terms- Enterprise Service Bus, Federation, communication platform and distributed service registry.

I. INTRODUCTION

Enterprise Service Bus (ESB) was introduced as an evolution of Enterprise Application Integration (EAI) a couple of years ago [1]. An ESB seeks to address difficulties in EAI by creating a service based architecture that exploits web services, messaging middleware, intelligent routing and data transformation in a highly distributed environment [2]. An ESB is responsible for, among many other functions, routing service request and response between heterogeneous services.

One of the highly distributed environments proposed in [3] is Grid-based Utility Infrastructure for SMMEs Enabling Technology (GUISET). GUISET allow SMMEs to subscribe to use infrastructure as a platform for deploying, finding and binding to a pool of services. GUISET aims to provide an e-infrastructure, where SMMEs come together with the aim of resource sharing and collaboration among themselves and their external partners.

Environments like GUISET where services exchange messages in a flexible manner; minimizing failure on communication platform becomes very important because it may lead to loss of income and customer dissatisfaction [4]. When ESB is deployed, services rely on it to facilitate communication amongst themselves, meaning that when an ESB fails no communication exist between services. This problem is known as a single point of failure.

As an attempt to solve this problem, this work seeks to deploy ESBs on different distributed server nodes connected in a form of federation. When one ESB fails the traffic intended for it is automatically re-routed to distributed service registry on another ESB in a federation using content based intelligent routing.

II. RELATED WORK

There are several attempts towards minimizing failure in an enterprise service bus, with each ESB vendor implementing its own different failure tolerance mechanism. Authors of [5] proposed two approaches for fault tolerant in Fuse ESB. 1) The use of Fuse Message Broker allows multiple service instances to run concurrently on different servers to guarantee service availability when a server fails. This approach also uses active/passive configuration and its disadvantage is that a failed active broker cannot be reintroduced. To reintroduce the old active broker, the new active broker has to be shutdown; therefore in case of service broker failure it means an outage of service [6]. 2) Another approach is to use the Fuse Service framework to support pure transparent client. Client catches the failure of a connected server in an interceptor and handles reconnection there. This configuration has an advantage that client reconnection happens at a client interceptor [6]. Authors of [7][8] show that Mule ESB relies on Java Message Service (JMS) and a number of open source JMS broker including Fuse Message Broker and JBossMQ for ensuring high availability. When a JBossMQ is employed it uses active/passive clustering and Mule has to connect to a new
active JMS broker instance after a failure of an old active instance. However, Kruessmann et al. (2009) prove that Mule instances are bound to a specific JMS broker address and cannot recognize new instance of JMS broker. The disadvantage of this approach is that manual configuration is required, which introduce delay.

The above approaches to failure tolerant system relies on JMS, which does not support automatic reconnection after a connection to JMS server has been lost resulting in service failure. Therefore it’s necessary to restart the service manually, which introduce delay [6].

Because multiple ESB deployments can be federated with one another, it’s now possible to provide a distributed integration platform that promotes the service reuse and scalable infrastructure within an enterprise [9].

Baude et al. (2010) proposed a method for enabling federation of ESBs as a means to realize their service cloud. The method advances a service bus by scaling the registries and message routers to the level of federations via hierarchical approach, and by incorporating the communication facilities offered by semantic space. Callaway et al. (2008), proposed the use of distributed service registries as an enabler for appropriately disseminating policies to federate in the federation of Enterprise Service Buses. Authors introduce two new protocols (Interior Federation Protocol and External Federation Protocol) that maintain the state of the distributed registry within and between autonomous federations. These solutions can be related to our proposal, but our work seeks to combine distributed service registries and content based intelligent routing in federation of ESBs for our failure tolerant system.

III. PROPOSED RESEARCH

The main goal of this research is to develop failure tolerant framework that is based on federation of ESBs as an integration and communication platform for GUISET services (namely Portal Interface Services, Identity Management Services and Application Services). The specific objectives for this research project are as follows:

(i) To investigate which integration capabilities GUISET services will need through creation of GUISET usage scenarios.
(ii) To determine which ESBs can best support GUISET integration needs through empirical comparative analysis of the existing ESB implementations.
(iii) To formulate failure tolerant framework, based on the results of objective (ii), as a federation of ESBs using distributed service registries to ensure that a replica of a service exist and is accessible on distributed server in the federation in case of an ESB failure, and using content based intelligent routing to facilitate communication by re-routing messages to the replica of service available in another available ESB.
(iv) To prototype the framework proposed in objective (iii) and performs the experiments to evaluate it using GUISET services to show improved ESB availability and failure tolerance.

IV. CONCLUSION

Today’s enterprise systems depend on an enterprise service bus as the standard way of communication due to its superiority which includes intelligent routing and data transformation. However, with all messages going through the bus, the single point of failure problem arises. Based on the existing deployments of ESBs, this work proposes failure tolerant framework based on federation of ESBs as a communication platform for GUISET services.

The proposed framework is expected to offer improved ESB availability, single point of failure tolerant system, and scalable deployment of ESBs.

REFERENCES


BIOGRAPHY

Themba Shezi received his undergraduate degree at the University of Zululand in 2010 and is currently studying towards his Masters Science degree with interest in service integration on distributed environments.