

Exposé

zur

Masterarbeit

zur Erlangung des akademischen Grades
Master of Science in Engineering

Eingereicht von

Christoph Ruhsam, BSc

Betreuer: DI (FH) Thomas Reidinger, 3 Banken IT GmbH
Begutachter: FH-Prof. DI Dr. Herwig Mayr

Juli, 2020

Inhaltsverzeichnis

Erklärung	iii
Abstract	v
1 Introduction	1
1.1 Motivation	1
1.2 Objectives	2
Quellenverzeichnis	3
Literatur	3
Online-Quellen	3

Abstract

An Enterprise Service Bus (ESB) is a crucial part of an enterprise, which connects the enterprise to its partners, customers, and other branches. The appearance of containerization, cloud services, and the microservice architecture have provided new possibilities for implementing and running an ESB. But, an ESB is commonly used by large conservative enterprises, which don't adapt new technologies fast, and wait until a new technology has proven itself. Especially the cloud is something the industry denied to use for a long time, because of the fact, that the infrastructure and data are managed and maintained by external service providers.

These days, we live in the so called cloud age, whereby global enterprises like Red Hat or Amazon provide cloud services such as Platform as a Service (PaaS), which can scale with the business. Enterprises start to consider to move their ESB installations to the cloud to profit from the cloud service provided features. Moving an ESB to the cloud will be a long term process for an enterprise, because the established processes for development, running, and managing the ESB will have to change.

This thesis has the goal to give the reader an overview of the cloud related concepts and technologies such as, Infrastructure as Code (IaC) and Docker, which are the base for cloud services. The implemented ESB prototype, is available at <https://github.com/cchet-thesis-msc/prototype>, and shows how an ESB could be implemented on a PaaS platform.

Kapitel 1

Introduction

1.1 Motivation

Large enterprises work with several independent applications, whereby each application covers an aspect of their business. In general, these applications are from different vendors, implemented in different programming languages, and with their own life-cycle management. To provide a business value to the enterprise, these applications are connected via a network and they contribute to a business workflow. The applications have to exchange data, which is commonly represented in different formats and versions. This leads to a highly heterogeneous network of applications, which is hard to maintain.

The major challenge of an IT department is the integration of independent applications into the enterprise application environment. The concept of Enterprise Application Integration (EAI) provides patterns, which help to define a process for the integration of applications into a heterogeneous enterprise application environment. One of these patterns is the Enterprise Service Bus (ESB), which is widely used in the industry[HW08].

Often, the term ESB application is used to refer to an ESB, which integrates internal and external hosted applications. But an ESB is a software architectural model rather than an application. The term could have been established by the usage of middleware such as JBoss Fuse, which provides tooling to integrate applications into an ESB. Red Hat JBoss Fuse is based on the JBoss Enterprise-Application-Platform (JBoss EAP), where all integration services run in the same runtime environment[Red18a].

With the appearance of cloud services such as Platform as a Service (PaaS), it is now possible to move an ESB from a dedicated environment to a cloud environment, whereby each integration service runs in its own runtime environment, rather than joining an existing runtime environment. The concept of Integration Platform as a Service (IPaaS) is built on top of PaaS, and enhances a common PaaS service with the integration features required by EAI[DG15; Liu+15].

Thus, enterprises can reduce the effort in implementing and maintaining an ESB, integrating applications via the ESB, and reducing the costs of an ESB, by using a consumption based pricing model. The cost are reduced due to the fact, that the ESB can scale down when its load decreases, whereby less resources are consumed, which have to be paid for.

1.2 Objectives

This thesis aims to implement an ESB on Openshift PaaS. An ESB is different to a Service Hub, because the ESB is a distributed system by design, whereby for instance transformation is not centralized, as it is with a Service Hub. A Service Hub has a central component, which performs transformation, routing and orchestration, whereby the scaling is limited by the hardware capabilities of the central hub. Additionally, the hub component of a Service Hub represents a single point of failure[HW08].

Commonly, an ESB is implemented with the help of middleware such as JBoss Fuse, which is based on JBoss EAP. The concepts of PaaS and IPaaS are in general new to the industry, which commonly hosts their integration services in their own data centers, due to the lack of trust for cloud services and knowledge about the new approaches such as the microservice architecture[Red18b].

Before implementing an ESB, which runs on a PaaS platform such as Openshift, it is necessary to understand the new concepts such as Infrastructure as Code (IaC) or containerization with Docker, which are covered in the following chapters. The microservice architecture and cloud services such as PaaS are becoming more important for the software industry, because of the features they provide. For instance, Red Hat is currently moving its ESB middleware JBoss Fuse to the cloud, whereby JBoss Fuse will fully rely on Openshift, and the integration services will have to be implemented as microservices. This has a huge impact on Red Hats customers, who are used to JBoss Fuse on top of JBoss EAP.

This thesis was commissioned by the company Gepardec IT Services GmbH, a company, which is working in the area of Java Enterprise and Cloud Development. The migration from a monolithic ESB to a microservice structured ESB, which is hosted in a PaaS environment is a major concern for them. The migration from a monolithic ESB to a microservice structured ESB will be a major challenge for their customers, because the microservice architecture and cloud services are mostly new to them.

Over the past years a huge technology dept has been produced by the industry, due to the monolithic architecture of their applications and little refactoring work on their application sources. It will be hard for them to reduce the accumulated technology dept, which they will have to, to keep competitive. Gepardec sees a lot of potential for their business and their customers in this new approach of implementing and hosting an ESB.

The term *monolithic ESB* refers to an ESB implementation, whereby all integration services are part of a single application, and therefore, are part of the application life-cycle, instead of having an own life-cycle per integration service. An integration service is a service, which integrates two or more other service with each other, by handling aspects like data transformation or security between the integrated service and its consumers.

Quellenverzeichnis

Literatur

- [DG15] T. Devi und R. Ganesan. *Platform-as-a-Service (PaaS): Model and Security Issues*. School of Computing Science und Engineering, VIT University, 2015 (siehe S. 1).
- [HW08] Gregor Hohpe und Bobby Woolf. *Enterprise Integration Patterns*. 11. Aufl. Boston, MA: Pearson Education, Inc., 2008 (siehe S. 1, 2).
- [Liu+15] Lawrence Liu u. a. *Integration Platform as a Service: The next generation of ESB, Part 1*. IBM, 2015 (siehe S. 1).

Online-Quellen

- [Red18a] Red Hat Inc. *Red Hat JBoss Fuse*. 2018. URL: <https://developers.redhat.com/products/fuse/overview/> (besucht am 22.02.2018) (siehe S. 1).
- [Red18b] Red Hat Inc. *Red Hat OpenShift Container Platform*. 2018. URL: <https://www.openshift.com/container-platform/features.html> (besucht am 22.02.2018) (siehe S. 2).