

Enterprise Service BUS – A Snapshot

Applies to:

SAP NetWeaver, SAP Exchange Infrastructure, enterprise SOA

Summary

Enterprise Service BUS (ESB) enables the implementation of enterprise SOA within an IT landscape. This article provides a brief overview of ESB and its relationship with traditional EAI.

Author: Hemant Negi

Company: HCL Technologies.

Created on: 9 May 2007

Author Bio

Hemant Negi currently works with HCL Technologies in SAP NetWeaver Practices.

Table of Contents

TApplies to:.....	1
Summary.....	1
Author Bio.....	1
EAI Defined.....	2
ESB Defined.....	2
Architecture of ESB.....	3
Why ESB.....	3
Related Content.....	Fehler! Textmarke nicht definiert.
Disclaimer and Liability Notice.....	4

Enterprise Application Integration Defined

Enterprise Application Integration (EAI) is defined as the unrestricted sharing of data and business processes throughout the networked applications or data sources in an organization. It is the process of integrating multiple applications that were independently developed. These applications generally include Data warehousing, Data linking, Application linking, and so on.

EAI may involve translation, transformation, routing, message delivery, message acceptance, and business process. It can be implemented through two basic types of integration architectures: Point to Point (BUS) and Message-based Integration (Hub and Spoke).

The EAI solution may span across multiple languages and platforms, including legacy systems, which makes the capturing and comparison of test results more difficult. Due to these factors, several barriers for EAI still exist, including incompatible application interfaces, limitations of current middleware technology, and the expense of adapting entire enterprise to EAI.

EAI message queues and brokers are centrally built around huge systems that are tightly integrated with each other. Having an ASYNC BUS only guarantees asynchronous delivery of messages for applications. The Process Manger normally requires human intervention or a tool to orchestrate the inter-application messages.

ESB Defined

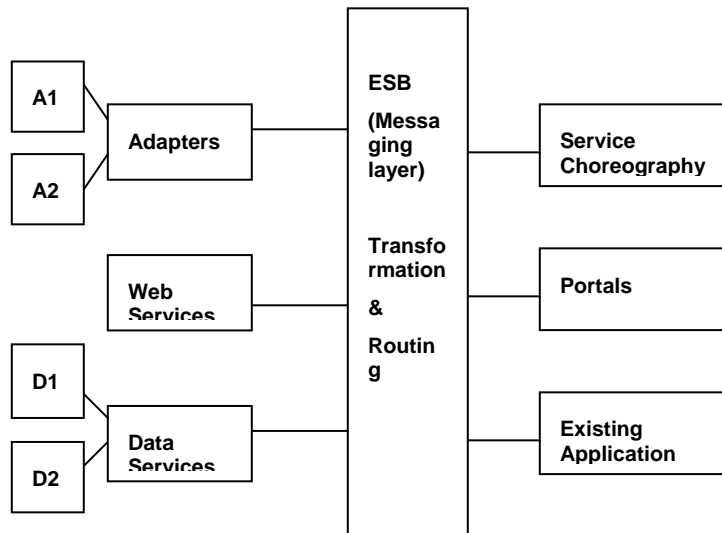
Enterprise Service BUS (ESB) can be defined as middleware that enables the Web service and message queuing to integrate with the existing infrastructure and scenarios. It is a medium upon which enterprise Service-Oriented Architecture (enterprise SOA) can be implemented – in other words, ESB acts a backbone for enterprise SOA implementation. It is related to building an organization that creates connection-oriented architecture to meet the IT and Business goals.

ESB is an open standard based on distributed environment. It supports content-based routing, multi-protocol collaboration, and the transformation and validation of messages.

Architecture of ESB

ESB provides common integration and communication services. The epicenter of ESB is a BUS that provides message delivery using SOAP, JMS and HTTP standards. It enables both asynchronous and synchronous messaging protocols to perform secure transformation and routing of service requests.

The following figure demonstrates the architecture of a typical ESB Scenario:



The above figure illustrates that adapters can integrate with different enterprise applications (A1, A2). Service choreography can be done with the help of Business Process Execution Language (BPEL).

Web Services are used for connectivity of legacy & integration technologies.

Data service provide real time data from different data sources (D1, D2) to existing applications generally running on Java/.NET Platforms.

The ESB supports message, service & event triggered interaction in heterogeneous environment. It requires service routing directory in order to route service request.

Why ESB?

ESB is an extended part of Enterprise Application Integration (EAI). It facilitates high level of distributed environment with loosely coupled systems.

Some of the key features exhibited by an ESB are listed below:

1. **Communication** - ESB should support addressing & routing for at least one messaging style.
2. **Service interaction** - Allows messaging model & interface definition format due to the widespread adoption of Web services standards (SOAP& WSDL).
3. **Integration** - It supports the provision of different adapters and service based integration capabilities.
4. **Service Choreography engine (Orchestration)** – It supports both stateful & stateless processes in order to orchestrate several business services.

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.