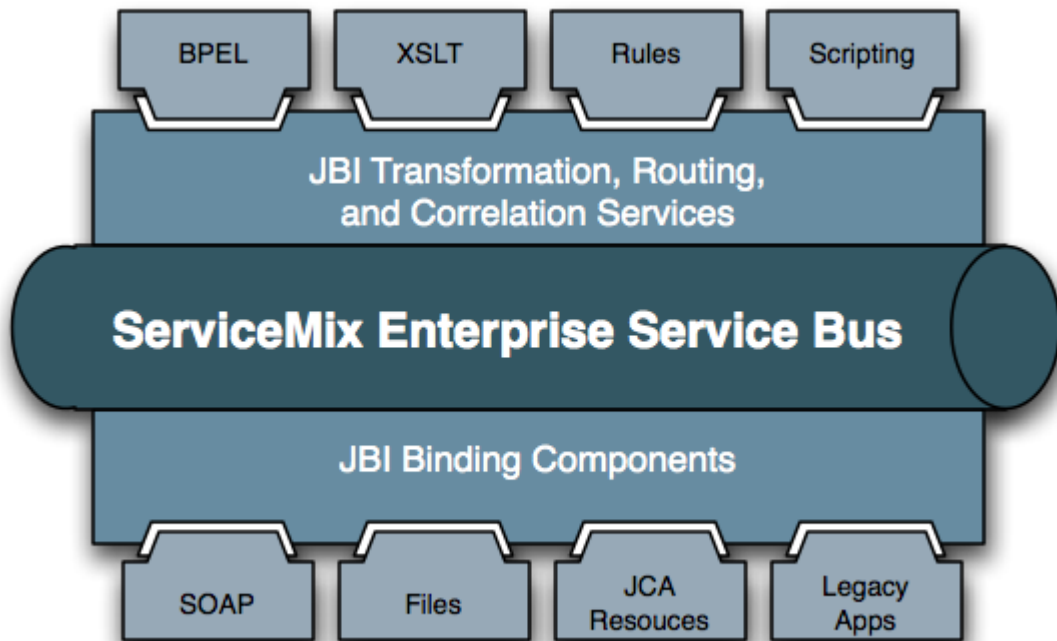


Q. Can you explain the architecture of Apache ServiceMix and its main components?

Apache Service mix EBS Basic understanding...

Apache Service mix is an open source (ESB) Enterprise Service Bus that combines functionality of service-oriented architecture (SOA) and an event driven architecture (EDA) to create an Agile enterprise EBS.

Apache Service mix is an open source distributed (ESB) Enterprise Service Bus built from ground up on the Java business integration (JBI) specification JSR208 and released under the Apache license. The goal of JBI is to allow components and services to be integrated in a vendor independent way, allowing users and vendors to plug and play.



#### JBI Container

Service Mix includes JBI container supporting all parts of JBI application including:

- Normalized message service and router
- JBI management MBeans
- Ant Tasks for Management and installation of components
- Full support for the JBI deployment units with bot deployment of JBI components

In addition, service mix provides an implementation of WS Notification

#### JBI Components

- SOAP
- Files
- JCA resources
- Legacy Apps
- BPEL
- XLST
- Rules

## Features

Service mix is a light weight and easily embeddable has integrated spring support and can be run at edge of the Network (inside a client or server) as a standalone ESB provider or as a service within another ESB. You can use Service mix in Java SE or a Java EE application server.

Service Mix uses active MQ to provide remoting, clustering, reliability and distributed failover. Service mix is completely integrated in Apache Geronimo.

## Q. How to evaluate an ESB (Enterprise Service Bus)

-> When faced the Need for integration middleware, it's natural progression to consider adoption of ESB. In doing any software evaluation. it's important to approach things in a Manner that utilizes requirements as central instrument in the process. There are commonly two levels of evaluation that need to take place when considering the adoption of any software including your business requirement and the feature set of software. Your business requirement and ESB feature set. There are many topics in these two levels of consideration, and they will be reviewed here.

### Your Business Requirements.

It's important to begin with your business requirements. Without a solid understanding your business requirement for a given application, you're spinning the wheels, and the effectiveness diminishes immediately. You don't necessarily need all the requirements upfront, but you need a starting point even if you're using agile methods, to develop your software This is where a knowledge of your business process is important even for the software developers.

### Your business purpose.

Don't say that your work is SOA. That is the *how* and not the *What*. SOA is nothing but the way of thinking to design software. Your business process is the goal you're trying to achieve for the business through the use of software, you are developing.

Notice that: Business process focuses on the goal of the software from business point of view.

Understanding your business process puts you in better control of driving the functional requirements to meet the goal.

### Your Functional Requirements

Once the business goal of the software project is understood, then move on to the functional requirement.

The functional requirement improve dramatically when the business purpose is identified and clearly communicated to all stakeholders in the software development projects. Such requirements are going to drive the architectural decisions in the systems and the software.

### Your architectural designs

Before you speak to any vendors it's also important to establish some architectural design about the software being developed. These are not written in stone but are definitely subject to change. Architectural decisions include what will be used to achieve the goals and requirements of the project. This includes designing the system in a service-oriented manner Use of specific packages and maybe even hardware and software necessary for developing and testing the software project. You may have even made some of these decisions implicitly throughout the earlier exercises. If so, it's important to identify and communicate these explicitly to the necessary stakeholders. So that there is an appropriate level of clarity amongst the group.

By working through these exercises you should now have not only a set of artefacts to share with all stakeholders via your wiki, but you will also have criteria for evaluating your ESBs

Your criteria for evaluating ESBS.

Any appropriate software evaluation requires knowledge of items discussed above. Without a good grasp of these topics, it's difficult to equate use of software with meeting specific goals.

It's also important that a software evaluation is not based on checkbox marketing (i.e., comparing feature sets side by side) This style of evaluation doesn't provide picture in actual use of software. Feature based comparison are meaningless without hands on experience.

Example.

What if software package claims to provide secure access to resource x but is bear to use API?

What if software package advertises features 1, 2 and 3 but it's performance is awful.

Hands-on experience with a given software package in your environment is a high priority. When you're talking about ESBS this is even more critical because ESBS handle system integration and no two system integration scenarios are ever same. Using ESB in your system to solve one or two of your problems via a proof-of-concept will provide you with a more informed and realistic set of expectations than will any Feature comparison

ESB oriented architecture: The wrong approach to adopting SOA

When considering the data you already have created through the steps above for evaluating and ESB, you should also be aware that an ESB oriented architecture is wrong approach to adopting SOA. In this article Bobby Woolf makes a case that's not a good idea to adopt an ESB without a plan to identify and construct the services that are care of an SOA. This is an ESB Specific instance of a general IT Problem about which much has been written. You should not adopt any technology without first considering the value that technology will produce.

*Q. Which version of Apache Service mix we will be using?*

-> Service mix 7.0

*Q. Can you enlist the modules we will upgrade as part JMMB ESB Upgrade project?*

Sprint No.	Duration (DD/MMM – DD/MMM)	Module
1		ESB-SMSBundle
2		ESB-T24DataXFerBundle
2		ESB-AFPXFerBundle
3		ESB-T24CasewareBundle
3		ESB-PM8Bundle
4		ESB-T24InvestmentRelayBundle
4		ESB-PortfolioManagementBundle
5		ESB-InvestmentGatewayBundle
5		ESB-GlobalAccountsTool
5		ESB-UnitTrust
6		ESB-BillPaymentBundle
6		ESB-RateServiceBundle
7		ESB-ATMDataXFerBundle
7		ESB-UniversalCustomerBundle
8		ESB-MoneylineCoreBundle

-----

Resource	Questions	Date	Answered (Yes/No)
Sahil	Post Upgrade and integration can we expect proof of concept? A demo along with QA Deliverables?	2 <sup>nd</sup> March	No
Sahil	Will evaluation of ESB documented? If yes, which deliverable will it be part of? Migration Document?	2 <sup>nd</sup> March	No
Sahil	Hands on experience for QAs and developers is necessary, once the application is accessible, are we aware that they will need hands on experience?	2 <sup>nd</sup> March	No
Sahil	What value will upgrading software produce? Will it improve the factor of cost/ time/ will it have access to more features and components of ESB?	2 <sup>nd</sup> March	No
Sahil	Need advice on which are necessary in Architecture Design document/Diagram to assist architecture decisions	2 <sup>nd</sup> March	No
Sahil	Can Developers assist QA for Hardware and Software requirements for ESB Testing?	2 <sup>nd</sup> March	No
Sahil	Are business goals and functional requirements of project documented explicitly? (sahil)	2 <sup>nd</sup> March	No
Sahil	How do we analyze release notes? Since this will help us with, new features, bug fixes, and backward compatibility issues	3 <sup>rd</sup> March	No
Sahil	Where do we store backup of current (4.7) Version of service mix environment? The backup should include customization, configurations and data	3 <sup>rd</sup> March	No
Sahil	Newer versions might need	3 <sup>rd</sup> March	No

	hardware, software pre-requisites, could we have these details documented, for representatives to review?		
Sahil	Migration document to include migration details, configuration adjustments and integration steps, right?	3 <sup>rd</sup> March	No
Sahil	Are there any Customizations or special configurations of current setup that are compatible with further updates?	3 <sup>rd</sup> March	No
Sahil	Should we List down functionalities critical for business operations?	3 <sup>rd</sup> March	No
Sahil	Should we have a Checklist of current setup in testing environment before deploying it into production environment?	3 <sup>rd</sup> March	No
Sahil	Will Documentation made during upgrade will be part of operational document?	3 <sup>rd</sup> March	No