



Questions from the Webinar

Webinar Title: Building your Next Generation IT Infrastructure for SOA

Hosted by: Dave Chappelle, BEA Worldwide Enterprise Architecture Team

Webinar took place: September 5, 2007

What infrastructures mentioned in this presentation is supported in current BEA product? For example, if I want to use ESB and security framework with support message lever security, what product I need to use?

The security framework, and support for message level security, mentioned in the presentation is already built into most of the BEA product line. So if you want an ESB with this capability you would only need to use ALSB.

How do you compare with BEA ESB product and XML appliance?

BEA's ESB offers the ability to configure proxy services through a simple GUI, and optionally extend the capabilities with custom Java coding. This is a powerful combination as it is easy to use and yet supports unique complex needs. XML appliances are generally much less flexible. I would suggest contacting your local Sales Engineer for a product presentation, demo, and possibly a direct comparison with a specific appliance.

Is it possible to engage BEA in a free/low cost project that pilots key pieces of the SOA IT Infrastructure?

The local sales team may be able to arrange free demonstrations and low cost proof of concept engagements.

How do you handle cross-domain interaction when other domains are on a different technology stack (i.e. BEA ESB on one side with IBM/SAP product in the other domain)

The key is to find common ground between the domains and allow the mediators to mediate to and from that common format. For example, you might decide that both stacks support XML+SOAP+basic auth and use that as the protocol between stacks.

Can you talk a bit about the Business Process Modeling (BPM) and the ESB ie the extent of integration between the two?

BPM is a mechanism to automate business processes. Business processes are made up of business activities that may be manual or automated. Automated business activities may be developed as shared services. The ESB acts as a proxy between consumers of services (the BPM product in this case) and the producers of services. So the integration pattern would be: BPM product executes business processes, and as a result interacts with the ESB in order to execute shared services. The BPM process (managed by the business analyst) represents the orchestration of business activities, and therefore the orchestration of services. The ESB (managed by IT) represents the mediator between consumers and producers of services, and therefore handles transport, message, security, and version mediation for each service invocation.

When we talk about SOA being technology agnostic, how do we justify that argument considering the fact that it seems to be technology driven in the real world?

SOA is technology agnostic from the standpoint that it does not rely on any particular technology. It doesn't require a specific programming language, interface type, protocol stack, etc. It is up to IT to define these



constraints. We like to think of SOA as an IT strategy - the idea of breaking down large projects and applications into smaller reusable services. Greater agility, less risk, and faster time to market can be gained. But at the end of the day IT must deploy something. So we build from the best technologies and frameworks we have today and apply them to meet the new strategy. New infrastructure is delivered to meet the needs of that strategy. New standards and best practices emerge. Technology is the medium that IT works with, and SOA is a strategy to guide IT towards a better way to build systems.

How much of a rework to introduce products like enterprise repository and ESB at a later stages compared to initial stages of SOA?

Repository is a collection of service artifacts, so the rework effort would depend on how much effort has already gone into organizing these artifacts in another way. Usually they aren't organized anywhere other than in the heads of people across IT. So its more of an exercise in getting things organized to begin with rather than rework. I'd suggest a product demo by the local Sales Engineer as a way to see how easy/hard it is to use the BEA repository. As for ESB, that would depend on what you are using today. If no mediator is used, then each service client would need to be re-targeted to the ESB proxy service instead of the service itself. Proxies would need to be configured for each service. Proxy configuration could be very simple pass-thru flows at first, which would easy to do. BEA has a service offering designed to help customers with these types of endeavors. Our SOA Infrastructure service includes analysis and planning phases to evaluate the effort and produce a deployment plan.

How do you use presentation functionality as a service?

WSRP provides a way to reuse portlets the same way you reuse other types of services.

Is the role of a common enterprise portal key in successfully deploying SOA projects enterprise wide?

I'm not sure what role you intend the common enterprise portal to play. If it's a way to consume services, then I'd say its one of many initiatives that can benefit from SOA. Others include BPM, data management, B2B, and so forth. If it's a way to communicate the SOA strategy, reference architecture, and service catalog, then yes, it would help, but communication can be through other means (e.g., wiki's, web sites, and repository).

Is the BEA ESB supported in a OEM mode?

So far we do not have a customer who has used BEA AquaLogic Service Bus as an OEM product but we will be open if the right opportunity arises.

Can the authentication/authorization be plugged into a Single Sign-on Infrastructure?

Single Sign-on infrastructure generally has an authentication mechanism already. The answer would depend on which product you use and if it can call out to 3rd party authenticators. It's job is to authenticate and produce a token that can be passed to other applications to propagate identity and provide proof of authentication. Authorization is generally not done by a single sign-on solution. Authorization can be broken down (using the XACML model) loosely into enforcement, decision making, and information gathering activities. The enforcement should always be local to the service/function/code that is being executed. The authorization decision should also be local, whenever possible, in order to avoid imposing a performance penalty. That is why authorization decisions are generally done by a security framework that is deployed with the application code. The Entitlements Engine I mentioned provides a way to push information such as access policies from a central location down to the services themselves so that local enforcement and decisions can be made but authorization policy can be managed globally.



How mature are the SOA IT Infrastructure Technologies?

It depends on which technologies you are referring to. Web services have been around for quite some time, so SOAP, WSDL, and UDDI are getting to be quite mature. There are many specifications and standards that lag behind these key technologies, but WSS (WS-Security), SAML, XML Encryption, and XML Signature have pretty much reached a maturation point. Generally you can tell by what the vendors are including in their products. Once multiple vendors "settle in" on a particular technology and version, then that technology can more easily be included in deployments, and therefore proliferate.

Can you summarize what's the ESB's real value?

The ESB enables loose coupling between service consumers and service providers, thus eliminating point-to-point connections between each provider and consumer. Consumers are no longer tightly coupled to the service endpoint, transport protocol, message format, security format, or interface technology. The service can change and evolve much more easily using a mediator than if it had to coordinate directly with every consumer - which improves agility and time to market.

Do you have developers license for ESB?

We do not offer developers only (non-deployment) license. However, the developer has a choice of 30 day free trial license.