



Open Source ESB

Johannes Carlén, Annica Rydin
Callista Enterprise AB



Agenda

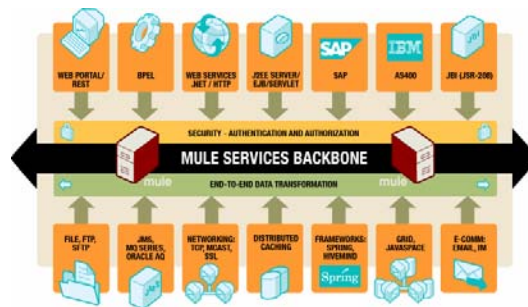
- Definition of Enterprise Service Bus
- Products
- Summary

CADEC 2007, ESB, Slide 2
Copyright 2007, Callista Enterprise AB



What is an Enterprise Service Bus?

- The definition of ESB varies
 - Standard, framework, product?
 - Enterprise Integration Pattern?
 - Architectural style?



CADEC 2007, ESB, Slide 3
Copyright 2007, Callista Enterprise AB

CALLISTA

Svårt att hitta en exakt definition av vad en ESB är

För en person är det ett ramverk som används som grundsten i en tjänstebaserad arkitektur

För en annan är det en produkt som inbegriper en massa verktygsstöd och flödesmodellering

Det kan vara ett designmönster eller en väg att gå i designen av en integrationsarkitektur

använda alla dessa begrepp för att försöka ge en förhoppningsvis tydligare bild av vad det handlar om

What is not an ESB?

- Not about process orchestration (with BPEL)
- Not only web services
- Does not automatically lead to SOA

CADEC 2007, ESB, Slide 4
Copyright 2007, Callista Enterprise AB



ESB handlar om orkestrering av meddelanden och tjänster och inte om affärsflöden och -processer. I flera produkter läggs t ex en processorkestreringsmotor ovanpå en ESB. Den kan vara BPEL-baserad och hantera t ex statefulprocesser samt uppdelning och sammanslagning av affärsflöden.

En ESB handlar inte bara om Web Services utan om en större mångfald av protokolltillgänglighet

Man kan inte enkelt säga att man använder sig av en tjänsteorienterad arkitektur bara för att man börjar använda sig av en ESB som bas i sin integrationsarkitektur. SOA handlar om mer än så.

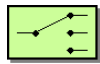
..autonoma, löst kopplade tjänster med väldefinierade interface som kan accessa genom en stödjande infrastruktur. En huvudtanke med lösa kopplingar är att man enkelt ska kunna byta ut komponenter utan att det stör övrig trafik, en annan att man vill separera applikationslogik från externa interface och därmed gömma implementationen av en språk- eller miljöspecifik tjänst.

Ett mönster hos en ESB är dock att de ger ett stöd för att implementera en SOA-arkitektur. En ESB kan alltså vara *en* grundkomponent av flera i en större tjänstebaserad integrationsarkitektur.

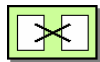
SOA är ett koncept. ESB är en del av infrastrukturen

Core features

- Mediation



Routing



Message transformation

– Protocol transformation

CADEC 2007, ESB, Slide 5
Copyright 2007, Callista Enterprise AB



Om vi ser på vilken kärnfunktionalitet en ESB har så handlar det i stort om ett centralt begrepp – Mediation

Meddelandetransformering

Protokolltransformering

Routing

På en högre abstraktionsnivå:

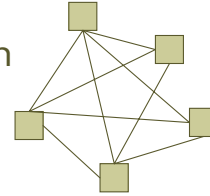
Message enhancement – lägga till eller ändra informationen enligt krav från en specifik service provider

Service mapping – översätta en business service till motsvarande implementation och ha ansvar för konfiguration av bindings och lokalisering och som definieras i termer av begreppen ovan - meddelandetransformering och routing

Why use an ESB?

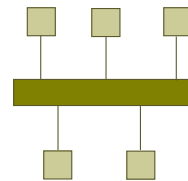
– Simple point to point communication

- Harder to
 - maintain and make modifications
 - controlling non-functional requirements
- Spaghetti...



Message bus

- Traceability
- Performance
- Loosely coupled services



CADEC 2007, ESB, Slide 6
Copyright 2007, Callista Enterprise AB



Varför inte bara använda sig av en point2point-kommunikation?

Det blir en underhållsmässig mardröm ocdet blir mer sårbart vid förändringar

Pastakastrull

Med en meddelandebuss kan vi få en renare, tydligare integrationsarkitektur med bättre spårbarhet och vi kan lättare planera strategier för att minimera risker. Kontroll över prestanda i en miljö där prestanda är absolut kritiskt får vi större kontroll alltså.

Vi kan bygga upp en mer tjänsteorienterad arkitektur med löst kopplade tjänster där vi har möjlighet att förändra delar av en systemportfölj utan att behöva förändra hela systemet.

Mediation – Message Routing



Message Router Patterns

- Static or deterministic
- Content based
- Rules based
- Itinerary based

CADEC 2007, ESB, Slide 7
Copyright 2007, Callista Enterprise AB



Fortsätta med att gå in lite närmare kärnfunktionaliteten i en ESB. Detta tänkte jag göra genom att gå igenom några stycken välkända integrationsmönster.

Det kan vara en form av statisk routing där vi i förhand konfigurerar upp var ett specifikt meddelande skall ta vägen.

Vi kan också ha en form av innehållsbaserad routing – alltså beroende på innehållet i ett meddelande så kanske vi bestämmer vilken väg eller till vilken tjänst meddelandet ska gå.

Exempel: ordersystem; databöcker eller skönlitteratur. Order kommer in på EIP, i vilken ordersystem skall vi ta vägen?

Innehållsbaserad routing kan vi dra ytterligare ett steg och använda ett mer komplicerat regelverk med regler för hur meddelanden skall routas vidare

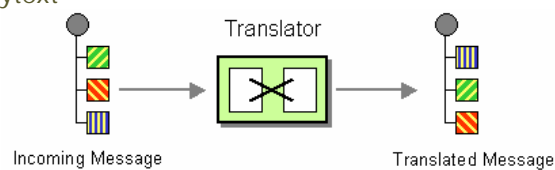
Mediation - Message transformation

“Converting the structure and format of the incoming service request to the structure and format expected by the service provider.”



Message translation pattern

- Converting message structures and formats
- Examples to and from...
 - XML
 - Cobol copytext
 - Objects



CADEC 2007, ESB, Slide 8
Copyright 2007, Callista Enterprise AB



Vi får in ett meddelande i ett copytext-format men vi vet att tjänsten vi skall skicka det vidare hemska gärna tar emot meddelanden i ett snyggt xml-format.

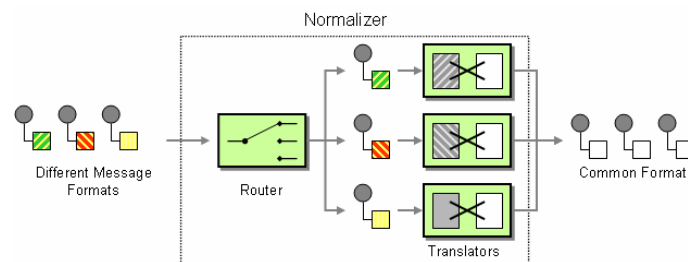
Då kan vi dra det genom en översättare som skapar den förväntade xml-strukturen

Mediation - Message transformation



Normalizer pattern

- Route incoming messages to a translator
- Translator transforms message into a common format



CADEC 2007, ESB, Slide 9
Copyright 2007, Callista Enterprise AB



I en större miljö kan vi ha ett incitament att vilja representera informationen internt på ett standardiserat format.

Då kan vi beroende på vad det är för struktur på meddelandet ruta det vidare till rätt översättare som fixar till den interna representation vi vill ha

Mediation – Protocol transformation

- Protocol transformation
 - Receive messages from one type of protocol and forward to a service provider through a different protocol
 - Example protocols
 - SOAP/WS => JMS/MQ
 - HTTP => XMPP
 - POP3 => CICS/MQ

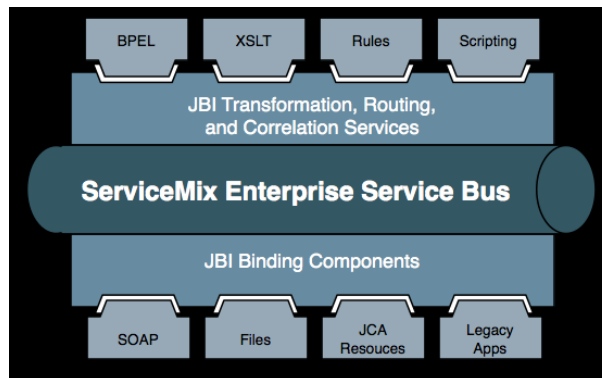
CADEC 2007, ESB, Slide 10
Copyright 2007, Callista Enterprise AB



Får in ett telegram över VCOM och drar det vidare på en kö genom MQ

The JBI specification

- JBI
 - Standards-based architecture for integrating middleware components to perform ESB capabilities.



CADEC 2007, ESB, Slide 11
Copyright 2007, Callista Enterprise AB



JBI är en standardbaserad arkitektur för integrationslösningar som ger möjlighet att plugga in tredjepartskomponenter i en infrastruktur på ett standardiserat sätt. JBI är inte intresserad av hur externa konsumenter eller tjänsteproviders interagerar utan snarare om hur dessa interagerar *internt*.

Syftet är att uppnå ett oberoende gentemot produkter/komponenter på samma sätt som inom J2EE och applikationsservermarknaden så man ska kunna mixa både open source och kommersiella produkter.

JBI är dock ifrågsatt av främst IBM och BEA men stöds mycket starkt av Sun. I open source-världen pekar det mot att alla produkterna har eller får någon form av stöd för JBI, antingen genom en direkt implementation eller genom pluggbarhet.

Non-functional requirements

- Reliability
- Availability
- Scalability
- Performance

CADEC 2007, ESB, Slide 12
Copyright 2007, Callista Enterprise AB



Ickefunktionella egenskaper/ krav

Det handlar om tillförlitlighet och tillgänglighet– fungerar allt även när det inte fungerar? Kan vi garantera att meddelanden kommer fram dit de skall? Och inom utlovad tid? Hur är det med klustring. Har vi någon failoverlösning då – om någon process går ner, har vi då andra som kan ta över?

Apropå klustring - kan vi skala upp lösningen med fler processer om lasten ökar och verkligen öka throughput?

Inte minst viktigt , kan vi få igenom tillräckligt många meddelanden per sekund som verksamheten kräver?

Open source ESB

- Emerging market
 - Venture capital
- Open source ESB companies
 - Business built upon open source products
 - Support
- User communities

CADEC 2007, ESB, Slide 13
Copyright 2007, Callista Enterprise AB



OS ESB – helt klart en växande marknad. Poppat upp flera produkter det senaste.

Det har vuxit fram ett stort intresse för integrationsarkitektur. I miljöer där man förr använde sig av point2point-kommunikation vill man idag reda upp härvan och lära sig av sina misstag och man har en tradition av att skraddarsy även samspelet mellan sina system.

I kölvattnet av det här intresset har det också uppstått ett överskott på kapital att investera i branchen?

Precis som i applikationsservermarknaden byggs det upp företag runt dessa open sourceprodukter och erbjuder support och bygger på med egna verktyg för konfiguration och övervakning

Genomgående inom open source-produkterna kan man se att det finns en bra grundsten, komponentbaserad vilket gör att det är enkelt att skapa och plugga in egna adapters och även ny funktionalitet.

JBI-specifikationen är viktig för alla produkterna i någon form vilket kan betyda att det finns en framtid för att bygga vidare och utveckla denna också framöver.

Användarbasen för flera av produkterna är redan hyfsat stora och det finns många ivriga utvecklare som svarar på frågor i forum och mailinlistor

Open source products

- Open ESB (Sun)
- Apache ServiceMix (LogicBlaze - Fuse)
- Mule (MuleSource)
- Celtix (Iona – Celtix Enterprise)
- JBoss ESB (JBoss)
- ChainBuilder ESB (Bostech Corporation)

CADEC 2007, ESB, Slide 14
Copyright 2007, Callista Enterprise AB



Tänker inte positionera de olika utan bara nämna något kort om varje.

ServiceMix, Open ESB bygger från början på JBI och WS men även har adapterar för en ansenlig mängd andra kommunikationsvägar och miljöer. Ett subset av Open ESB kommer med i Suns JEE 5 SDK. Open ESB är en av dem som har störst sortiment av adapters för legacysystem

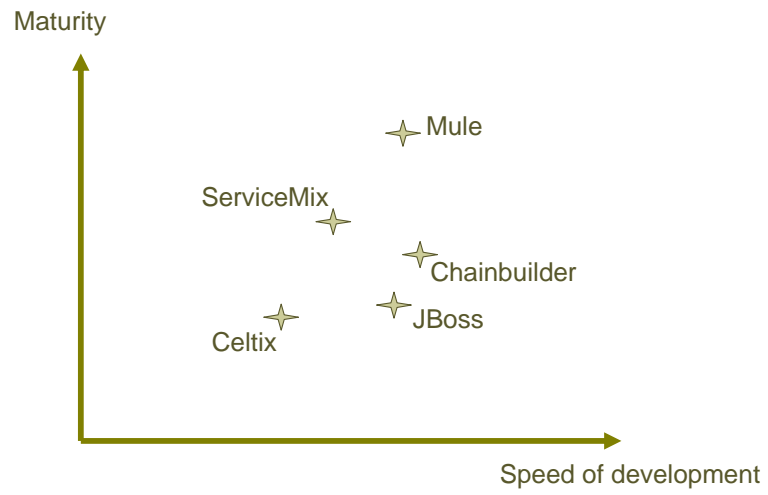
Mule, som sedan tidigare har ett stort momentum bland open source-produkterna, mycket på sin enkelhet, har en egen intern objektrepresentation men också stöd för JBI – WS en av flera adapters. Kan användas tillsammans med alla JBI-containers.

Celtix JBI kompliant, implementerar JAX-WS specification och kräver ingen extern implementation som t ex Axis för att skicka soap-meddelanden.

ChainBuilder ESB, den senaste i raden där man satsat stort på verktygen runtomkring för konfiguration. Också JBI-compliant

Apache Synapse – mediationramverk för web services

Momentum



CADEC 2007, ESB, Slide 15
Copyright 2007, Callista Enterprise AB



Commercial ESB's



CADEC 2007, ESB, Slide 16
Copyright 2007, Callista Enterprise AB



Commercial vs Open Source ESB

- Licensing cost
- Non-functional requirements
 - reliability, availability, scalability, performance
- Adapters
- Value added functionality
 - Process Coreography (BPEL)
 - Tooling / Graphic modelling
 - Monitoring / Administration
- JBI

CADEC 2007, ESB, Slide 17
Copyright 2007, Callista Enterprise AB



Vad får vi då för fördelar med en kommersiell produkt

Har ofta lösningar för att kunna uppfylla de ickefunktionella kraven som jag nämnt innan. som tillförlitlighet/skalbarhet/tillgänglighet etc

Support, färdigt stöd / färdiga adaptrar för många olika plattformar och legacysystem

Tilläggfunktionalitet som ger ett extra värde.

Ofta lägger tillverkarna ett extra skal över ESBn och ger möjligheter att t ex orkestrera affärsprocesser och flöden genom flera tjänster m h a BPEL.

Man ger ofta ett hyfsat verktygsstöd för konfiguration och även möjligheter att grafiskt modellera hur tjänster anropas mm

Det finns ofta verktyg för att monitorera meddelandeflödet och även verktyg för administration

Saknas:

Adapters till enterprisemiljöer,

Värdehöjande tilläggfunktionalitet som processkoreografi, övervakning,

Summary

- Comparisons to J2EE in the beginning
 - Emerging market
- Open source integration architecture
- Core component maturity
- Adapters for many protocols and platforms
- Commercial products has more added value functionality
- Licensing cost

CADEC 2007, ESB, Slide 18
Copyright 2007, Callista Enterprise AB



Varför välja open source

Mycket har hänt det senaste

Stort driv in om ESB – forum, mailinglistor och även support från open source-företagen

Flera nya produkter

Stort intresse för lösningar för integrationsplattformar och även inom skräddarsydd utveckling

Bubblar lite som J2EE – som håller på att mogna och konsolideras, nya konstellationer uppstår

Mule kanske ”störst” just nu – men det händer grejer...

Saknas viss funktionalitet

Verktögsintegration. Övervakning

Använda när?

Fixar funktionella/ickefunktionella krav? Rapa upp!

724, failover, adapters(ims cics stordator)

If you remember just one thing...

- ESB from Fullers Brewing
- <http://www.fullers.co.uk>



CADEC 2007, ESB, Slide 19
Copyright 2007, Callista Enterprise AB



Q&A



CADEC 2007, ESB, Slide 20
Copyright 2007, Callista Enterprise AB



Commercial ESB's - products

- IBM WebSphere ESB / Message Broker
- BEA AquaLogic Service Bus
- SAP XI
- Oracle ESB
- TIBCO BusinessWorks
- Sonic ESB
- Fiorano ESB
- Iona Artix 4

CADEC 2007, ESB, Slide 21
Copyright 2007, Callista Enterprise AB



Det finns ett helt gäng av aktörer på marknaden och här är en del av de starkast lysande stjärnorna på ESB-himlen.

IBM med sin message broker och numera även med en mer lättviktig koncentrerad ESB

BEA med sitt AquaLogic-koncept

Där finns Sonic som ansågs vara först ut på ESB-kartan med sin produkt

Samt ett helt gäng andra mer eller mindre kända och jag har säkert missat några här.