



# Open ESB v2, Open ESB.next and Project Fuji

Andreas Egloff  
Lead Architect SOA / Business Integration  
Sun Microsystems

# Introduction

# Project Open ESB

My pages **Projects** Communities java.net

Projects > java-enterprise > open-esb

**Open ESB Project**

- [Home](#)
- [All Downloads](#)
- [Architecture](#)
- [Components](#)
- [Documentation](#)
- [FAQs](#)
- [Licensing](#)
- [Get Help](#)

**Project Tools**

- [Announcements](#)
- [Build Instructions](#)
- [Developer Tools](#)
- [Source Code \(CVS\)](#)
- [Fisheye of Source](#)
- [Issue Tracker](#)

Go to: Issue#

**Open ESB Community**

- [Community Home](#)
- [Developer Wiki](#)
- [News](#)
- [Events](#)
- [Blogs](#)
- [Forums](#)
- [Mailing Lists](#)
- [Partners](#)
- [Developers](#)
- [Get Involved](#)

**About The Website**

- [Site Map](#)
- [Contact Us](#)
- [Feedback](#)
- [Ideas](#)



Over **600** members and **600,000** downloads

**CDDL license** – Free to download and free to deploy

Close to 40 components

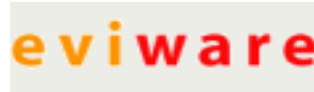
<http://open-esb.org>

**About Open ESB**  
 Project Open ESB implements an Enterprise Service Bus (ESB) runtime using Java Business Integration as the foundation. This allows easy integration of web services to create loosely coupled enterprise class composite applications.  
[Learn More](#) [Architecture](#) [Components](#) [Documentation](#)

**Downloads**  
 Looking for executables? Check our downloads page to find what you are looking for.  
[Latest Builds](#) [Extras and Tools](#) [All versions](#)

**Components**  
 Want to find out more about a specific Open ESB component? No problem! Check our components page to find everything you need.  
[Components](#)

**Join Our Community**  
 Would you like to help expand and improve Open ESB? Got tips and tricks to share? Sign-up here and join us!  
[Community](#) [News](#) [Events](#) [Blogs](#) [Forums](#) [Mailing Lists](#)  
[Partners](#) [Developers](#)



# Project Open ESB

- > Project Incubated @ JavaOne 2005
- > New Component Project, “Open JBI Components” – September 2006
- > Vibrant Developer Community
  - More than 600 Registered Members
- > Vibrant Partner Community
  - Gestalt, Imola, Bostech, Eviware, ServiceSpan, Xcalia, Danet, Persistent Systems, RemedyMD, ...
  - Several Individual Contributors
  - Over 10 active partners and over a 100 active contributors

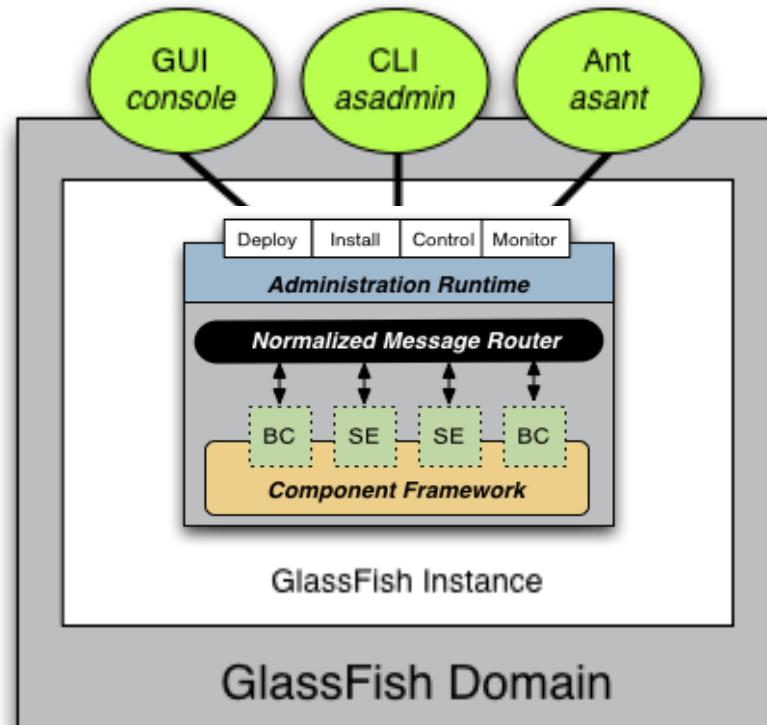
# What's What in Open ESB

- > Open ESB v2 is the current production version
  - Included and supported in the Java CAPS R6 suite, released in June 2008
  - Many more components and enhancements being worked on and planned for this and next year
  - Is the basis of GlassFish ESB v2 later this year, a supported distribution on GlassFish v2
- > Project Fuji, Open ESB.next
  - Research into the next generation integration stack, some enhancements will flow earlier into v2
  - Shared “early” to develop fully in the open in the community

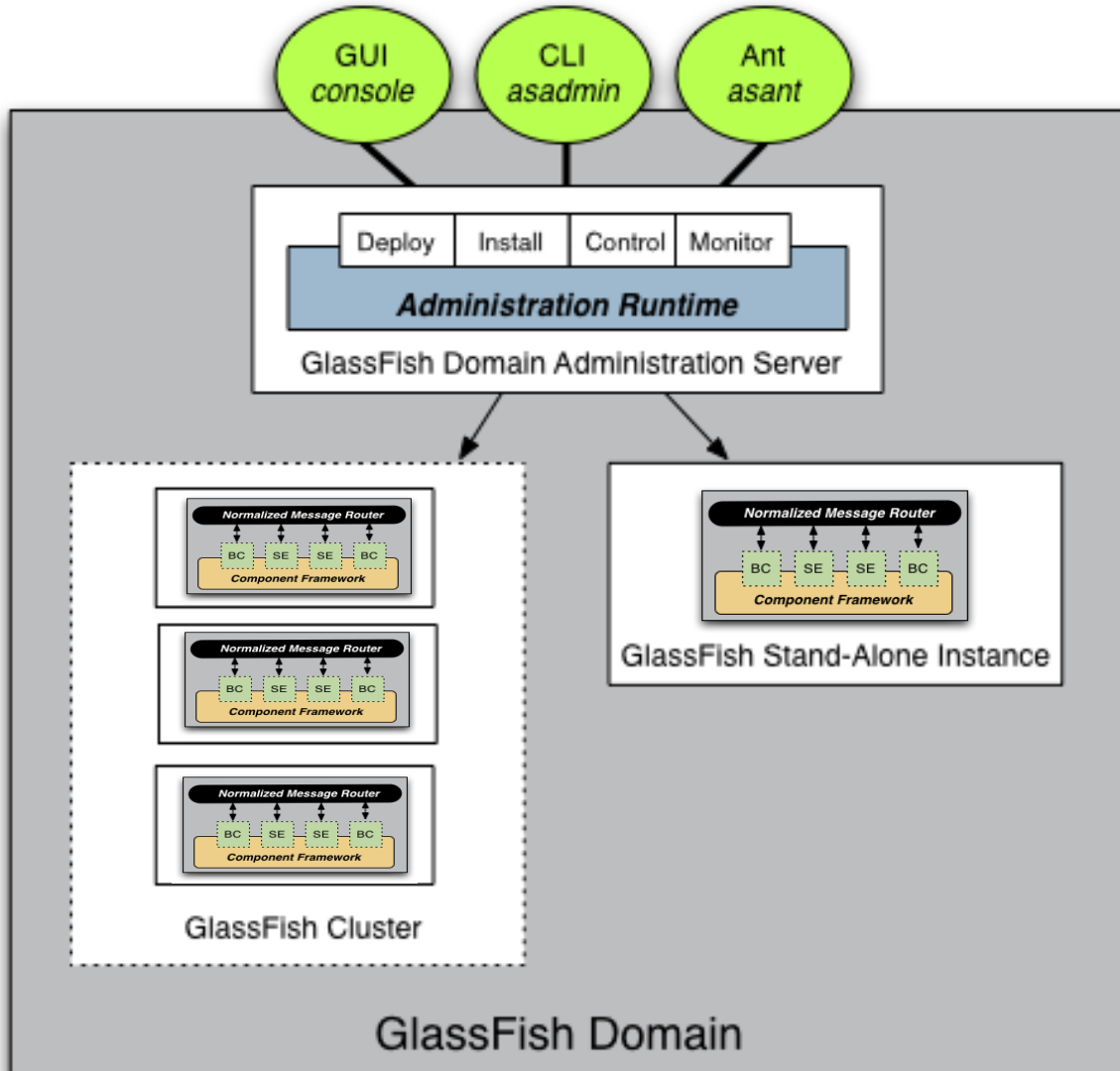
# Fully integrates into GlassFish v2

- **Integrated Runtime**
  - > JBI runtime pre-installed with GlassFish
  - > Enabled by default in every GlassFish instance
  - > Pre-packaged components: HTTP BC and Java EE SE
- **Integrated Administration**
  - > Admin GUI
  - > CLI (asadmin)
  - > Ant (asant)
- **Integrated Architecture**
  - > Open ESB fits seamlessly into GlassFish domain administration model
  - > Clustered Open ESB

# GlassFish - Developer Profile



# GlassFish - Cluster Profile





# Administration & Monitoring

- Extensive runtime and tooling support for management and monitoring
  - > Installation/Deployment
  - > Life Cycle Management
  - > Monitoring
  - > Configuration
- Completely integrated with Glassfish administration interfaces for seamless look and feel
- Open contracts allow third-party components to plug-in to existing administration framework

# Administration Example

Home Version [Logout](#) [Help](#)

User: admin Domain: domain1 Server: localhost  
**Sun Java™ System Application Server Admin Console**

- Application Server
- Applications
  - Enterprise Applications
  - Web Applications
  - EJB Modules
  - Connector Modules
  - Lifecycle Modules
  - Application Client Modules
- Web Services
- JBI
  - Service Assemblies
  - Components
    - bostech-tcpip-binding
    - gestalt-rss-binding
    - gestalt-sip-binding
    - gestalt-uddi-binding
    - gestalt-xmpp-binding
    - jbi4cics
    - jbi4corba
    - jbi4ejb
    - sun-dcom-binding
    - sun-email-binding
    - sun-exec-binding
    - sun-file-binding
    - sun-ftp-binding
    - sun-hl7-binding
    - sun-http-binding
    - sun-ims-binding

JBI > Components > sun-bpel-engine
 

General	Configuration	Application	Descriptor	Loggers	Monitoring	Libraries
---------	---------------	-------------	------------	---------	------------	-----------

### sun-bpel-engine - Monitoring Statistics for this component

View statistics for this component.

- ⌵ Summary Statistics
- ⌵ Endpoint Statistics

- ⌵ Message Exchange Statistics
- ⌵ Component Provided Statistics

---

#### Summary Statistics

<b>Component Name</b>	sun-bpel-engine
<b>Instance Name</b>	server
<b>Up Time</b>	0 days, 0 hours, 15 minutes and 39.386 seconds
<b>Activated Endpoints</b>	3
<b>Completed Exchanges</b>	3
<b>Active Exchanges</b>	0
<b>Error Exchanges</b>	0

[⏪ Back to top](#)

---

#### Message Exchange Statistics

<b>Response Time</b>	137598333 nanoseconds <small>Avg. response time for message exchange</small>
<b>In Component</b>	137160333 nanoseconds <small>Avg. time taken in component by message exchange</small>
<b>In Delivery Channel</b>	91000 nanoseconds <small>Avg. time taken in delivery channel by message exchange</small>
<b>In Message Service</b>	344666 nanoseconds <small>Avg. time taken in message service by message exchange</small>

[⏪ Back to top](#)

Done

10

# Adapters

- Communication Adapters
  - > FTP BC
  - > Email BC
  - > CICS BC
  - > CORBA BC
  - > DCOM BC
  - > File BC
  - > HTTP BC
  - > JMS BC
  - > LDAP BC
  - > RSS BC
  - > SIP BC
  - > SNMP BC
  - > TCP/IP BC
  - > UDDI BC
  - > XMPP BC
  - > Asterisk BC
  - > MQ Series BC
  - > MSMQ BC

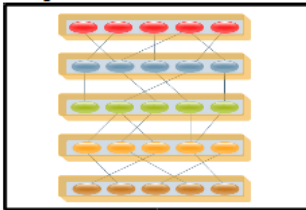
# Adapters and Service Containers

- Database Adapters
  - > JDBC BC
  - > SQL SE
- Application Adapters
  - > SAP BC
  - > IMS BC
  - > HL7 BC
  - > SWIFT BC
- Service Containers
  - > XSLT SE
  - > Data Mashup SE
- Service Containers continued
  - > Java EE SE
  - > BPEL SE
  - > ETL SE
  - > JRuby SE
  - > Groovy SE
  - > Scripting SE
  - > Aspect SE
  - > IEP SE
  - > WorkListManager SE
  - > **And more...**

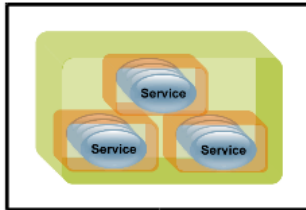
# Architectural Properties Crash-course

# SOA Big Rules

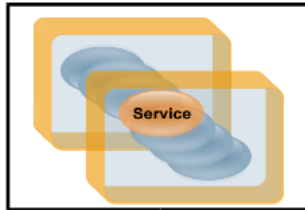
**Layered**



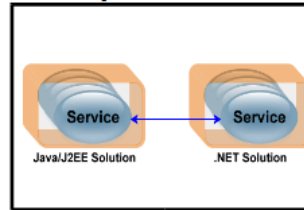
**Modular & Autonomous**



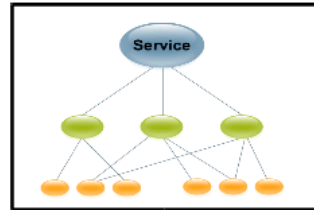
**Service Reuse**



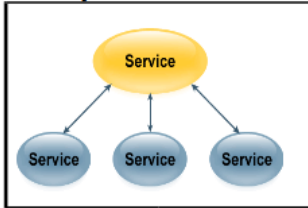
**Interoperable**



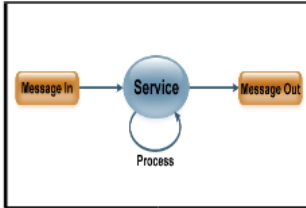
**Coarse-Grained Business Services**



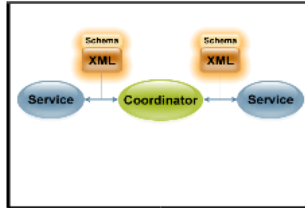
**Composable**



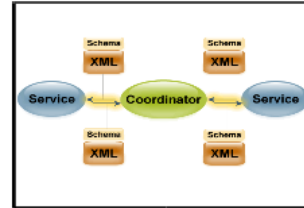
**Stateless**



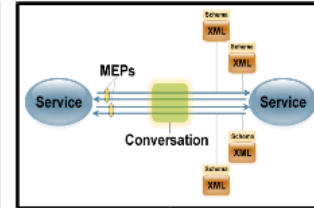
**XML Doc-Based**



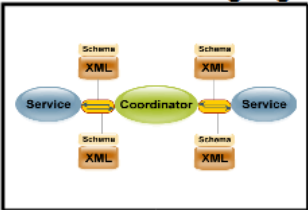
**Asynchronous**



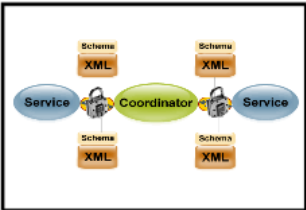
**Conversational**



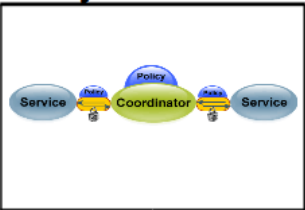
**Reliable Messaging**



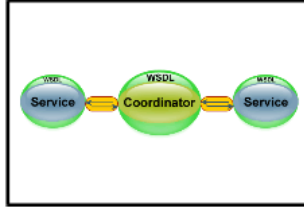
**Secure**



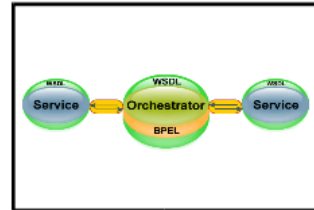
**Policy-Driven**



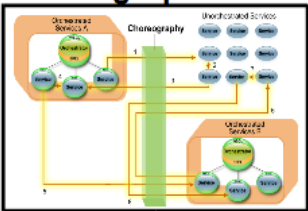
**Described**



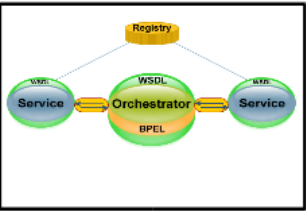
**Orchestrated**



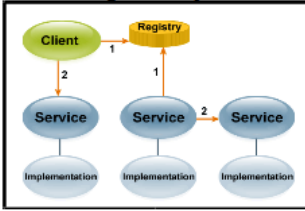
**Choreographed**



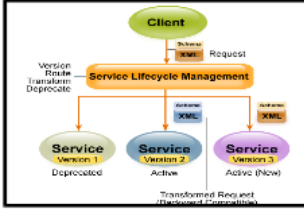
**Registered & Discovered**



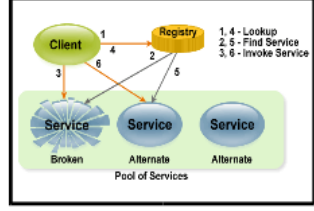
**Loosely Coupled**



**Versioned**



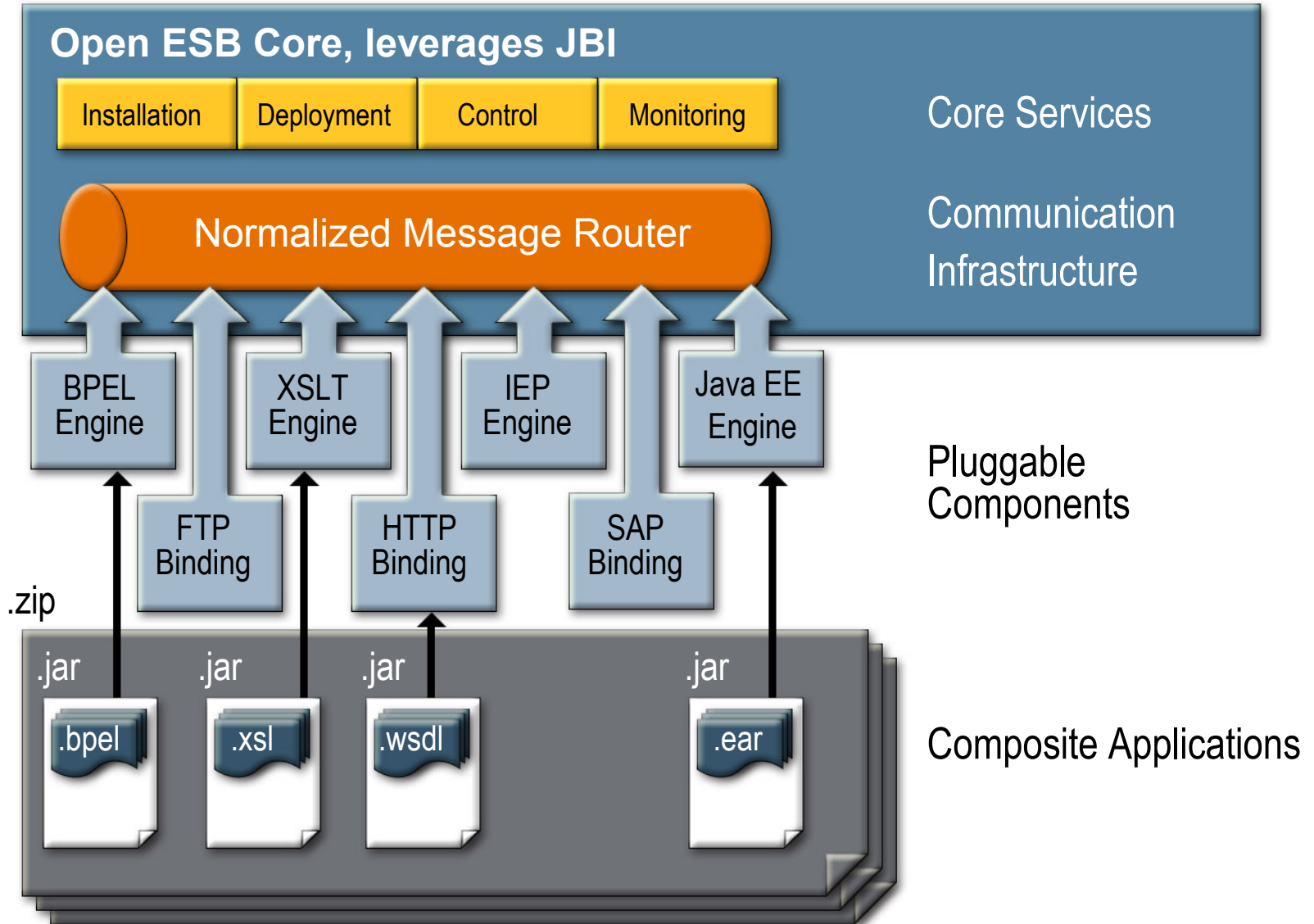
**Self Healing**



# Service Oriented Architecture

- Navigating a terminology minefield
  - > Service. Service Consumer. Service Provider
  - > Service Description
  - > Service Bus
- Key aspects of a SOA runtime
  - > Message-based
  - > Asynchronous
  - > Mediation
- Service-Oriented Integration
  - > Integration viewed as a conversation between services
  - > More than just web services

# Pluggable Architecture





# Pluggable + Many Components = Needs Systemic Qualities

- Identifying system-wide features and quality concerns and addressing them in a unified manner
- A few examples
  - > Application Configuration
  - > Component Configuration
  - > Component Upgrade
  - > Password Handling
  - > Logging
  - > Monitoring
  - > Wire Qualities

# Some Architecture Principles

- Adds a standard communications contract
  - > Promotes interoperability between service containers
  - > Allows for easy service mediation
- Asynchronous service invocation
  - > Allows individual containers (“components”) to scale
- Includes in-vm message exchanges
  - > Allows effective (distributed) transaction propagation
  - > Streaming, Passing by reference
- Message based integration
  - > Everything is represented as services, including legacy systems

# Inherent Scalability w/o Coding

- Several properties of Open ESB allow for effective scaling without actually explicitly coding
- Vertical Scalability
  - > Leverage SEDA (staged, event driven architecture) principles with asynchronous message exchange approach of JBI
  - > Asynchronous exchanges means that no threads have to be blocked, a thread is not tied to an invocation or message – saving threads, memory and resources
- Horizontal
  - > Loose coupling and mediation capabilities make it easier to distribute the system

# Less Boiler Plate Code

- Programming model should not change
  - > For example, develop annotated POJOs for business logic “as usual”
    - Behind the scenes deploys to Java EE service engine
  - > The service containers and service bus transparently add the additional qualities and features
  - > Application developers are expected to use existing components (service containers), typically not to write them
- External connectivity generally done via configuration rather than coding against APIs
  - > Service based interaction with the external

# Additional flexibility of in-VM SOA

- Transaction scopes can span as many services as desired
  - > Not forced to add a transaction boundary for each stage as a pure JMS based architecture would
- Queuing and persistence points chosen by user where needed and desirable
- Light weight contract, less overhead
  - > No serialization required
  - > Efficient streaming, pass by reference capabilities
- Easily interacts with other SOA based systems
  - > Low impedance mismatch

# Interoperability, Loose Coupling

- Components are not directly aware of each other
  - > Only interact through standard service bus (normalized message router), message exchange contract and message format
  - > Eliminates “stove-pipe” integration issue between components



# Pluggable, Component Ecosystem

- Pluggable architecture + component interoperability
  - > Mix and match components from different contributors and partners
  - > Components from different vendors (using the JBI standard) can be used with no or small changes
- Vibrant Component Ecosystem
  - > Over 30 components in open-jbi-components project today for use with Open ESB and others
  - > <https://open-jbi-components.dev.java.net/>
- Reduces the danger of vendor lock-in

**Where are we going?  
Project Fuji,  
Open ESB.next**



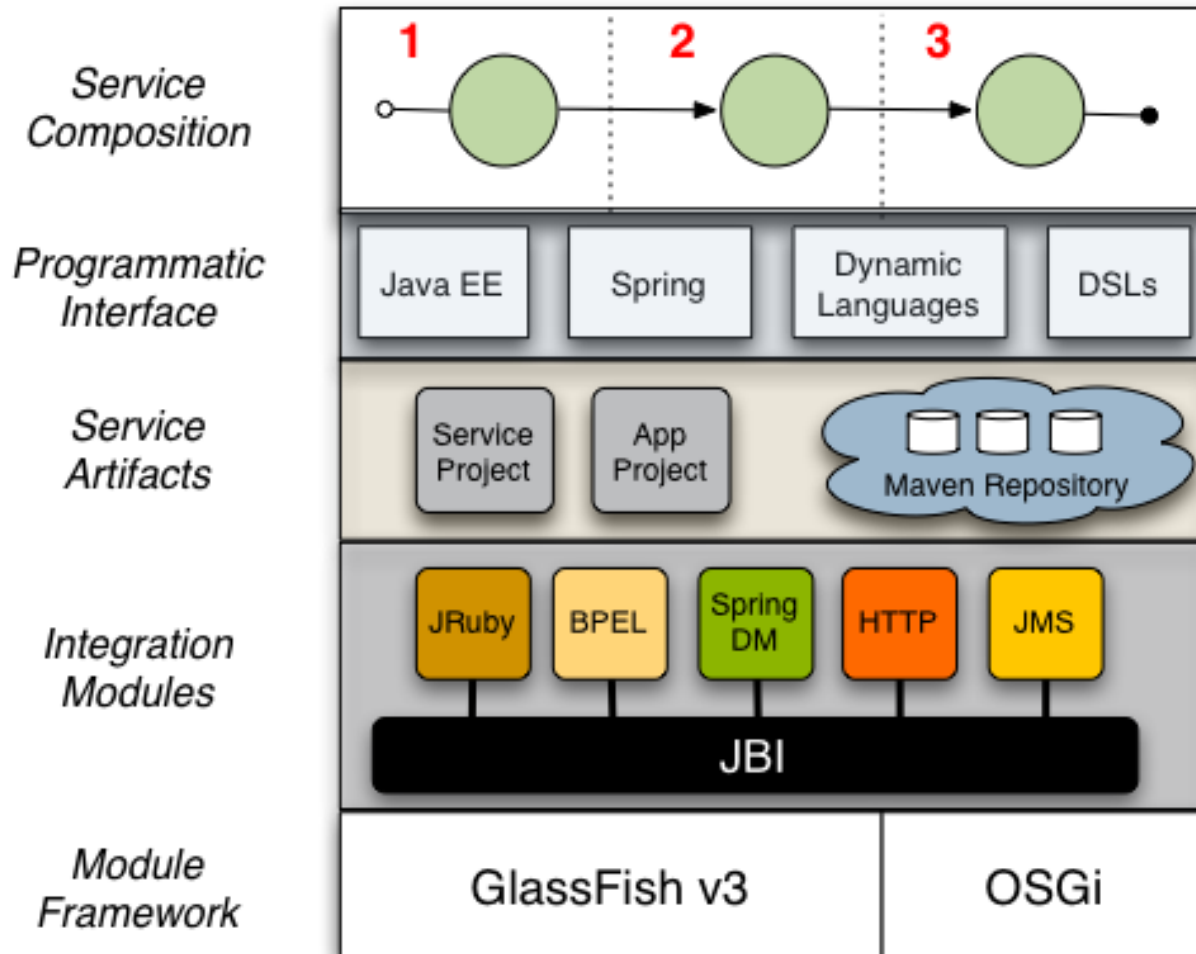
# Background

- Project Fuji represents research for the next generation of our core integration stack
  - > Some advances will flow earlier into the current production line, Open ESB v2
- Compatibility
  - > Open ESB v2 tooling can generate applications for Fuji
  - > Fuji runtime uses the same components as Open ESB v2, no separate code lines
- Goals:
  - > Keep the robust infrastructure in Open ESB v2
  - > Right-sized runtime, lightweight microkernel
  - > Ease-of-use, Productivity
  - > More Flexibility and Choices (Tooling, Languages...)

# Overview

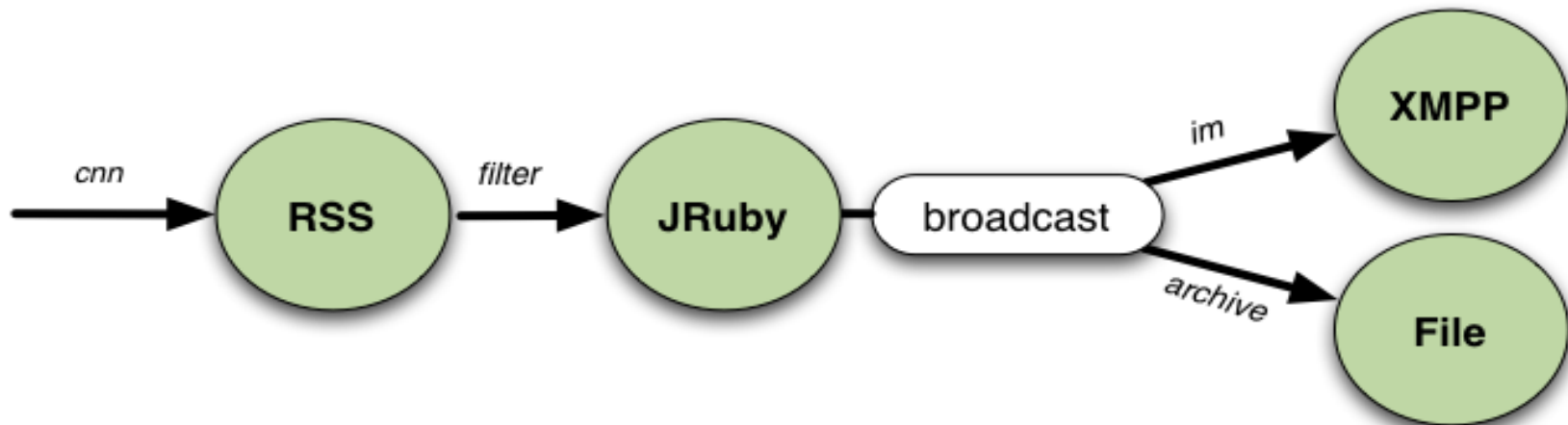
- Based on OSGi
  - > Core JBI runtime < 300kb
  - > Everything is an OSGi bundle
    - > JBI framework, components, applications, libraries, ...
  - > Can be used in any OSGi framework
- Adds (non-intrusive) Maven-based tooling
  - > Application and service archetypes
  - > Really cool plugins
    - > Service generation
    - > Application distribution
  - > Works with any Maven-enabled IDE
    - > NetBeans, Eclipse, etc.

# Architecture



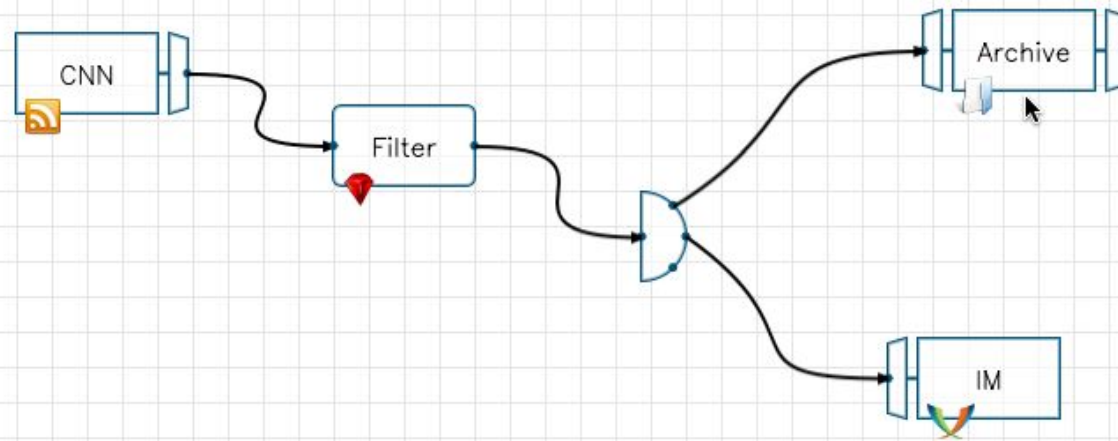
# Service Composition - IFL

```
rss    "cnn"  
xmpp   "im"  
jruby  "filter"  
file   "archive"  
  
route do  
  from "cnn"  
  to "filter"  
  broadcast do  
    route to "im"  
    route to "archive"  
  end  
end
```



# Service Composition - UI

Palette
Existing Services
<i>No existing services...</i>
Existing Adapters
<i>No existing adapters...</i>
New Service
♥ JRuby Service
New Adapter
🔍 jabber
🌈 XMPP Adapter
New Control
📡 Broadcast



# More Info...

## Jazoon Technical Presentation

- > Integration Profile for GlassFish v3  
Wednesday, June 25th 2008, 14:00 – 14:50, Arena 9

## Places to bookmark

- > [open-esb.org](http://open-esb.org)
- > [wiki.open-esb.java.net](http://wiki.open-esb.java.net)
- > [fuji.dev.java.net](http://fuji.dev.java.net)
- > [blogs.sun.com/andi](http://blogs.sun.com/andi)

## Connect with Open ESB team via

- > [users@open-esb.java.net](mailto:users@open-esb.java.net)
- > [dev@open-esb.java.net](mailto:dev@open-esb.java.net)



# Thank You

Andreas Egloff  
[blogs.sun.com/andi](http://blogs.sun.com/andi)  
[andreas.egloff@sun.com](mailto:andreas.egloff@sun.com)

