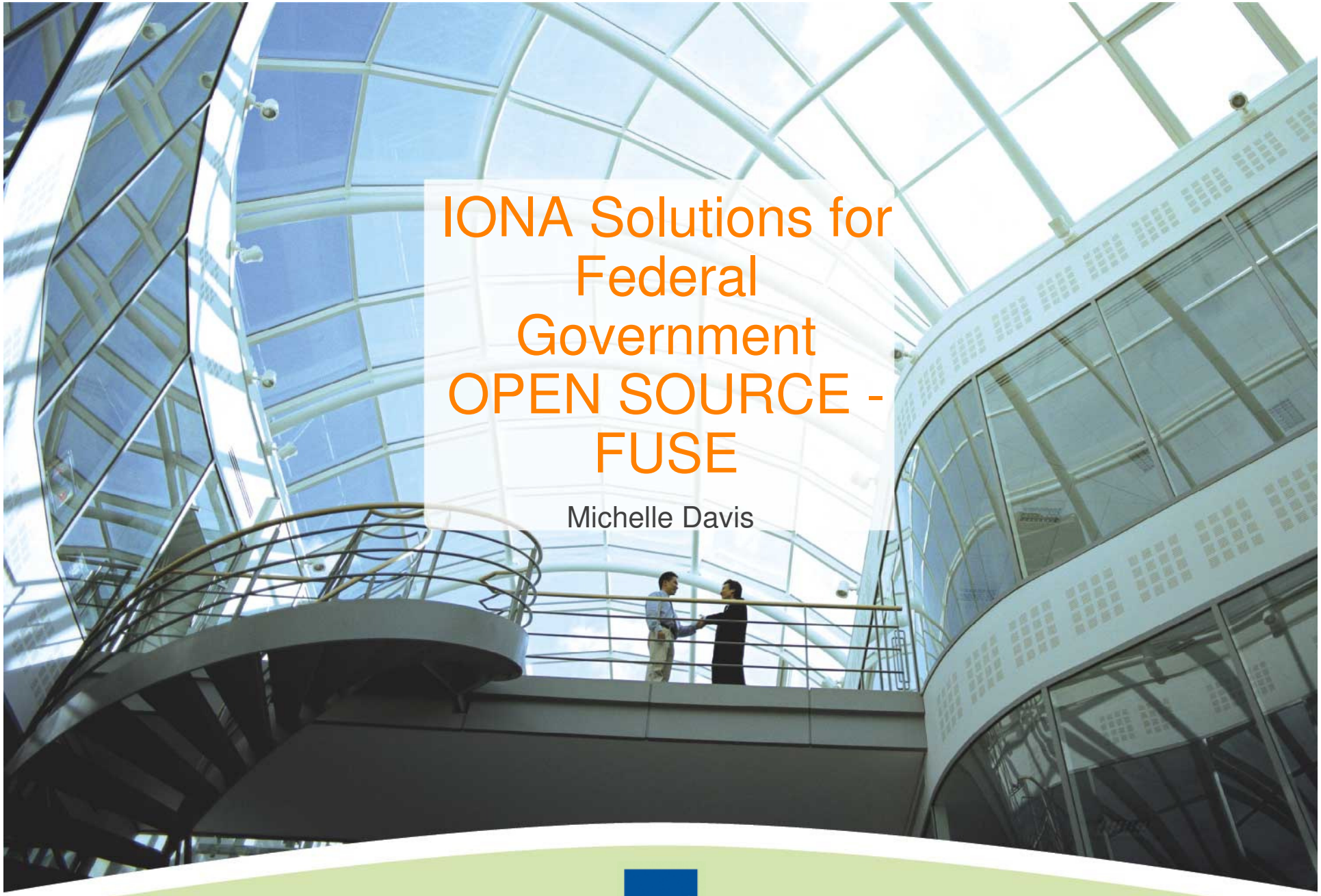


IONA Solutions for Federal Government OPEN SOURCE - FUSE

Michelle Davis

IONA

Making Software Work Together™

Agenda

- About IONA, Open Source & SOA
- FUSE SOA Infrastructure
  - FUSE Message Broker
  - FUSE ESB
  - FUSE Services Framework
  - FUSE Service Mediation
- Summary

Making Software Work Together

Customers include world's largest firms

- 80% of Global Telecom
- 70% of Financial Services in Global 100
- Blue Chip Partners

Worldwide presence

- EMEA HQ in Dublin, Ireland
- US HQ in Massachusetts
- APAC HQ in Tokyo, Japan

Solid business with a history of profitable growth

- Founded in 1991
- Publicly traded since 1997
- Strong balance sheet

NASDAQ:IONA

The IONA Approach

- Deliver high performance integration software for mission critical applications
- Make heterogeneity an asset, not a liability
- Deliver on the value proposition of standards

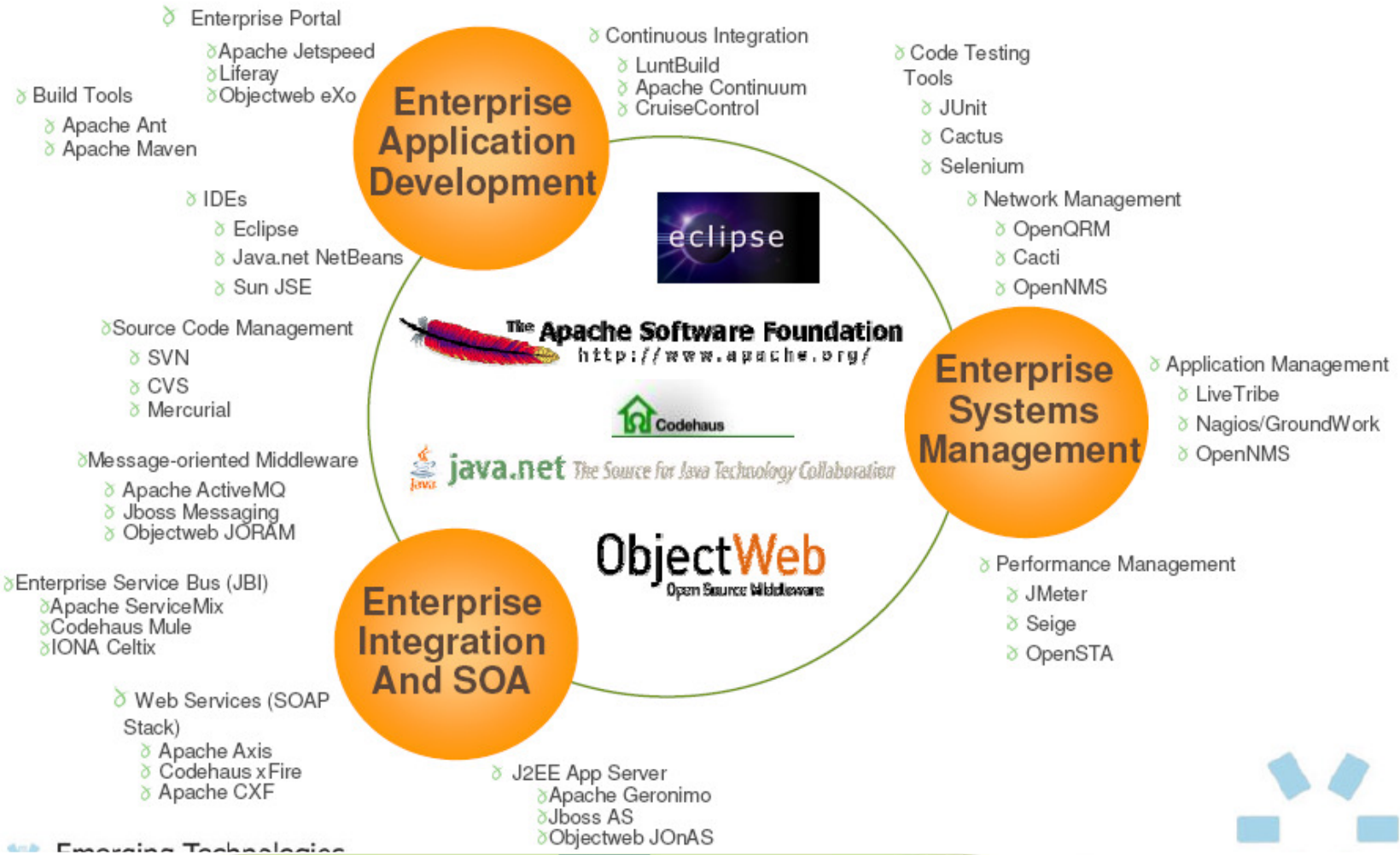
IONA Customers

IONA Open Source Strategy

Market leadership in Open Source SOA Infrastructure

- To provide an on-ramp for a variety of types of users, developers, and architects which allows flexibility on adoption paths which in turn assist in finding the right fit for SOA infrastructure
- To leverage the benefits of an Apache open source license and community
- To drive adoption Open Source

Open Source …. Rising Adoption



IONA

Making Software Work Together™

* Slide by Chariot Solutions

What is SOA? (Recap)

- SOA **should be** business oriented
- SOA is a **way of thinking**
- SOA is **not** Web Services
- **Loosely coupled** architecture that uses **messaging**
- **Enriched** by creating composite apps
- Move from batch to **real-time**

Some Goals of SOA

- Build reusable business components (Services)
- Reliable
- Manageable

  (Service locators/directories)
- Provide agility and flexibility

Why SOA? – For the Business Owner

- To gain some type of ROI
  - Internal (for the organization)
  - External (for citizens or partners)
- To be competitive
  - Time to market
  - More robust applications

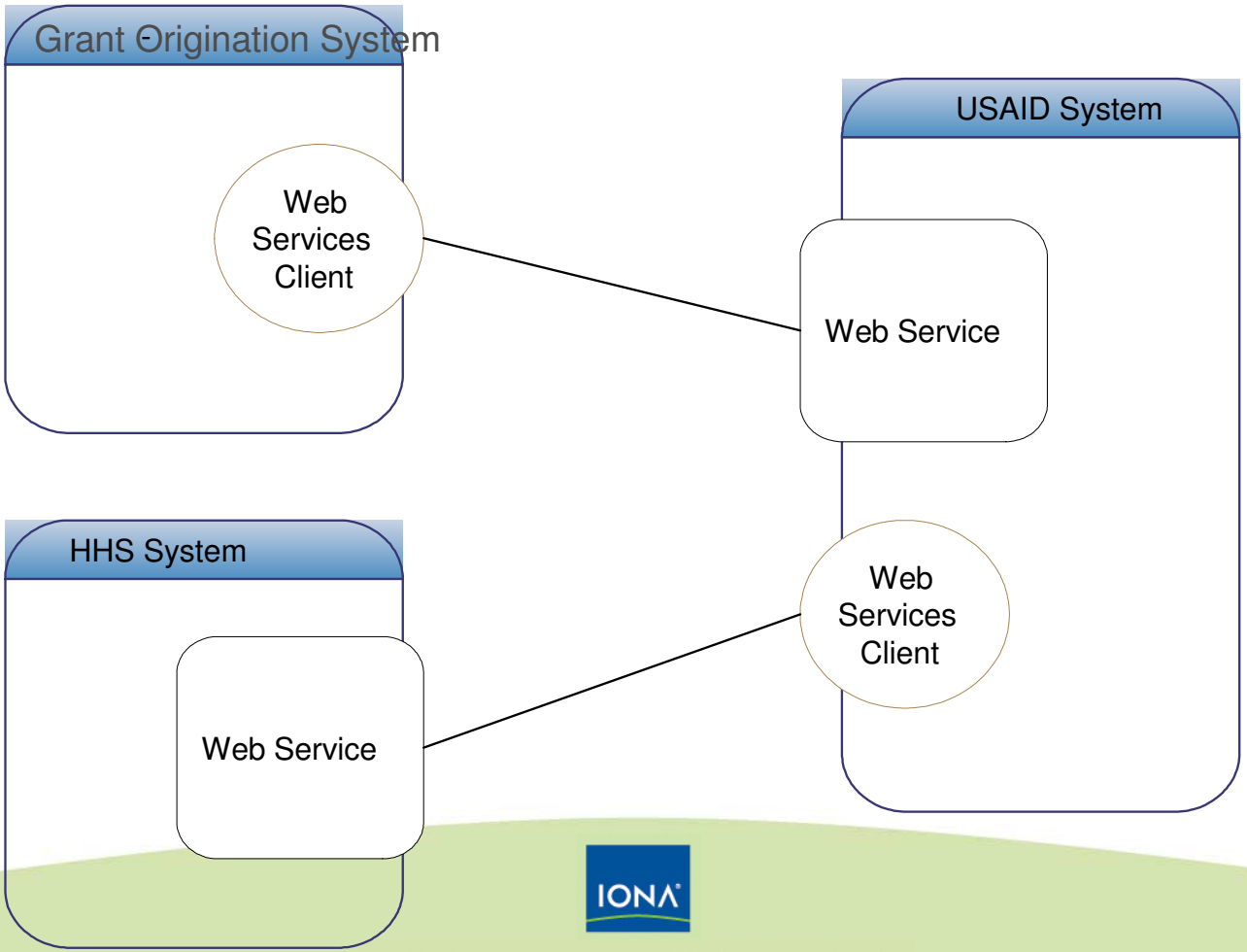
SOA Pilot 1 – Grants Approval System

- Routes grants applications between Heath and Human Services and USAID systems
  - Real-time response
  - Loosely coupled interface
    - XML - not a file layout anymore
- Replaces file-based delivery service
  - Batch

SOA Pilot 1 – Grant Approval System

- SOAP Web Service
  - WSDL
  - HTTP
- Pre-ESB
  - Hard-coded connectivity
  - Not as configurable as the ESB approach

SOA Pilot 1 – Diagram Grant Approval Service



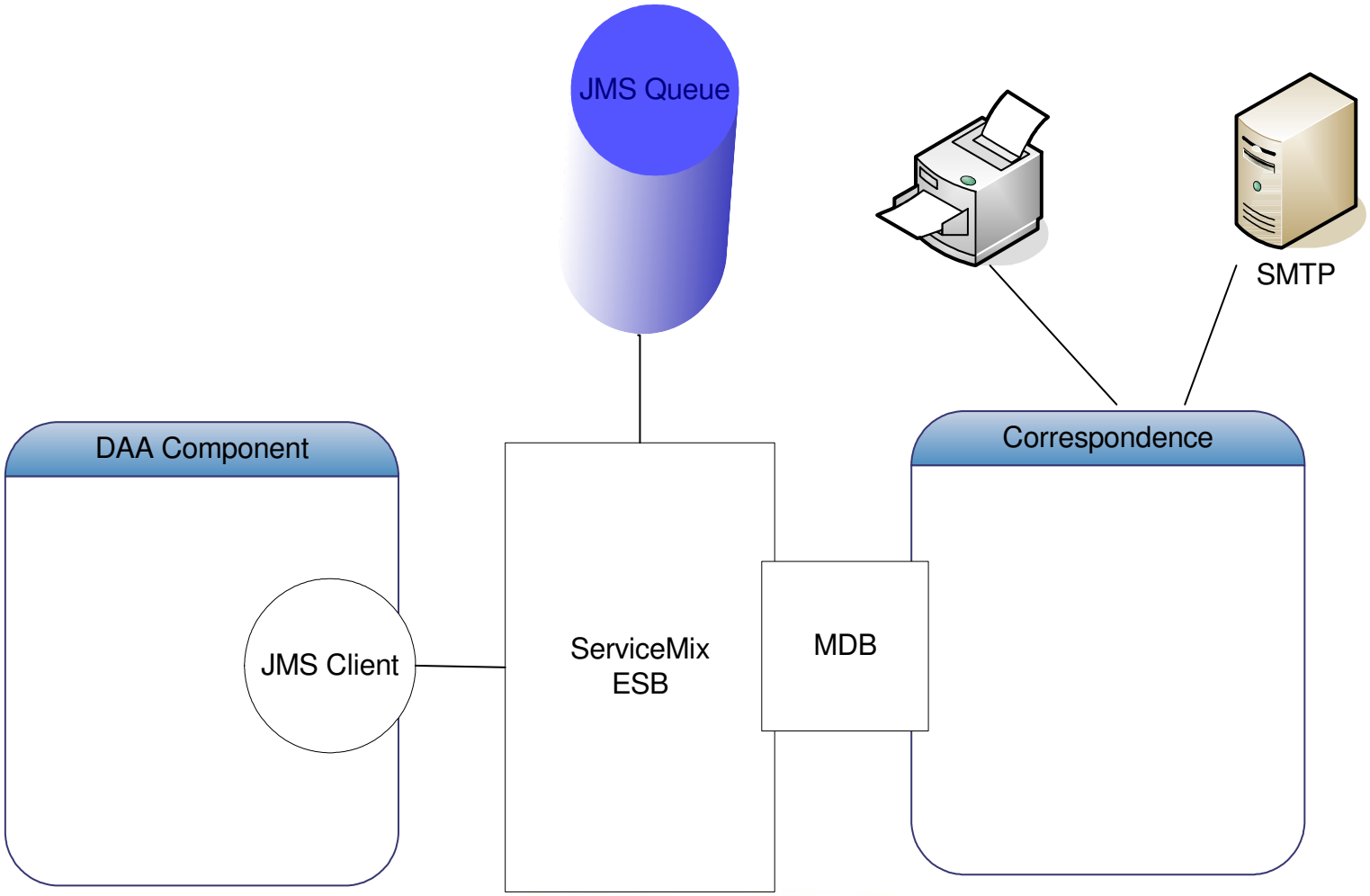
SOA Pilot 2 – Data Correspondence Service

- Provides letter generation upon request
  - Letter format based on templates
    - Grant letter for applicant
    - Grant report for OMB
    - Grant request letter for USAID
  - Request does not have to wait for response
    - Asynchronous
    - Document generation can be time-consuming
  - Output types
    - Text, HTML, PDF
  - Destinations
    - Email, Printer

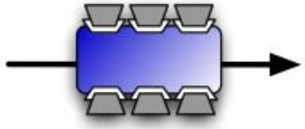
SOA Pilot 2 – Technical Architecture - Data Correspondence Service

- EJB Message-Driven Bean
- JMS queuing
- Open source environment
  1. ServiceMix - ESB
  2. Geronimo – J2EE application server
  3. Tomcat - Web container
  4. ActiveMQ – JMS provider

SOA Pilot 2 – Diagram Correspondence Service



FUSE SOA Product Line



FUSE ESB

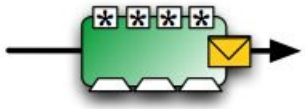
Based on Apache ServiceMix (JBI)

Deployment Container for integration components in a standards based way.

FUSE Message Broker

Based on Apache ActiveMQ (JMS)

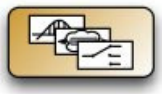
Reliable Messaging: High performance, reliable messaging for tying together Services across a SOA or Integration Pattern



FUSE Services Framework

Based on Apache CXF (JAX-WS)

Services Enablement: Lightweight, pluggable, extensible services framework and service enablement



FUSE Mediation Router

Based on Apache Camel

Integration Components: Easily defined process routing based on standard Enterprise Integration Patterns definitions

The FUSE Stack

Cohesive SOA stack forming around open source

- **ActiveMQ** as the message bus
- **CXF** (JAX-WS RI) as the SOAP stack with mediation
- **ServiceMix** as the JBI / ESB hosting all the integration components
- **Camel** as the mediation and enterprise integration pattern support.

- **Tooled by Eclipse and SOA Tools Project (STP)**

Fuse Message Broker

- Based on the Apache ActiveMQ project, the leading open source solution for JMS Messaging.

  - JMS 1.1 compliant implementation
  - High Performance
  - Flexible & extensible
  - Master/Slave HA
  - Networks of brokers - scalability
  - Native code client libraries for C, C++, .NET and Java
  - Scales to 1000's/sec transactions

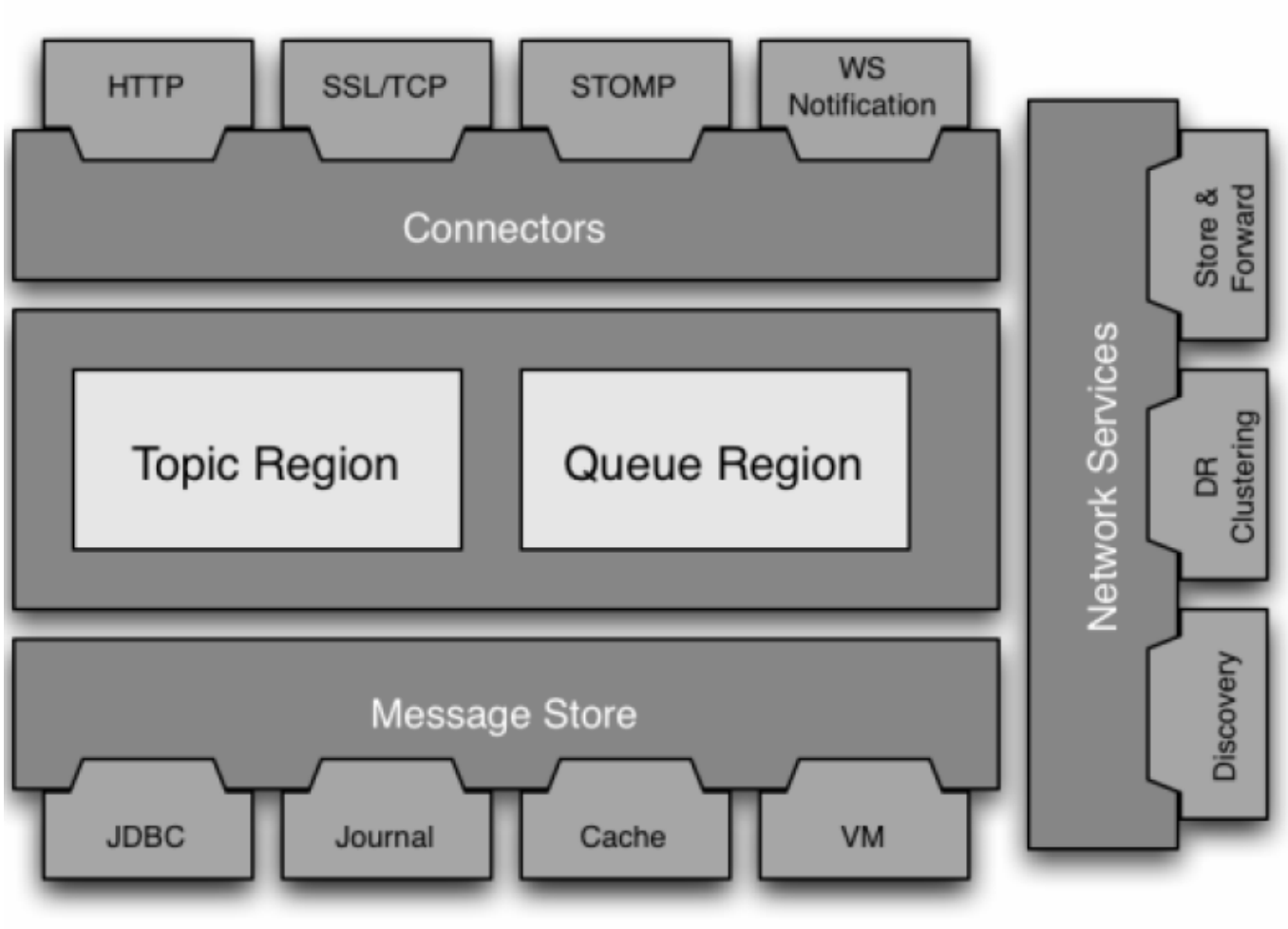
FUSE Message Broker

Apache ActiveMQ

- Enterprise class Message Bus (MOM)
- Standards-based:
  - JMS 1.1, J2EE 1.4, JCA 1.5, JTA and XA
  - C, C++, C#/.Net, Python, Perl, PHP, Ruby and Java
- Integrated to:

  Geronimo, Spring, Tomcat, JBoss and any J2EE 1.4 container (e.g., WebLogic/WebSphere)
- Fastest open source JMS provider by some margin and close, or better than, the proprietary alternatives
- Highly scalable messaging
  - Load balancing, Clustering, peer-to-peer, master/slave HA and federated networks support
  - Distributed Destinations, smart routing features like exclusive queues and message groups

ActiveMQ Architecture



FUSE ESB

- Based on the Apache ServiceMix project. The leading open source JBI Enterprise Service Bus.
  - JBI 1.0 implementation
  - Connects virtually any enterprise application and platform via open standards.
  - Supports existing middleware via standard interfaces.
  - Enables robust, extensible SOA

ServiceMix ESB

- Supports JBI (JSR 208)
  - Container, binding components and service engines
- Connectivity
  - Works with any SOAP stack and WS-* provider : CXF, Axis, Synapse, SCA/Tuscany, WSIF, ActiveSOAP
- Transports for HTTP, JMS, JCA, JTA, RSS, Jabber, Email, Files etc
- Smart Routing
  - Content based, Rules based, XPath based, Itinerary based
- Transformation
  - XSLT, Scripting, Java code
- Platform
  - Runs in any JVM, Geronimo, JBoss, J2EE
- Manage services & monitor endpoints
- Hot deploy BPEL engines, BPEL processes

Pluggable Architecture - JBI

- The Java Business Integration specification provides interface standards for integration services, enabling a pluggable architecture in which multiple providers may be called upon for key integration requirements.
- The pluggable JBI architecture allows organizations to use their preferred service solutions in their SOA. Any standard JBI-compliant Service Engine or Binding Component may be deployed to the FUSE ESB JBI container, and FUSE ESB components may be deployed to other JBI-compliant ESBs.

Components

Components communicate through mediated Message Exchanges

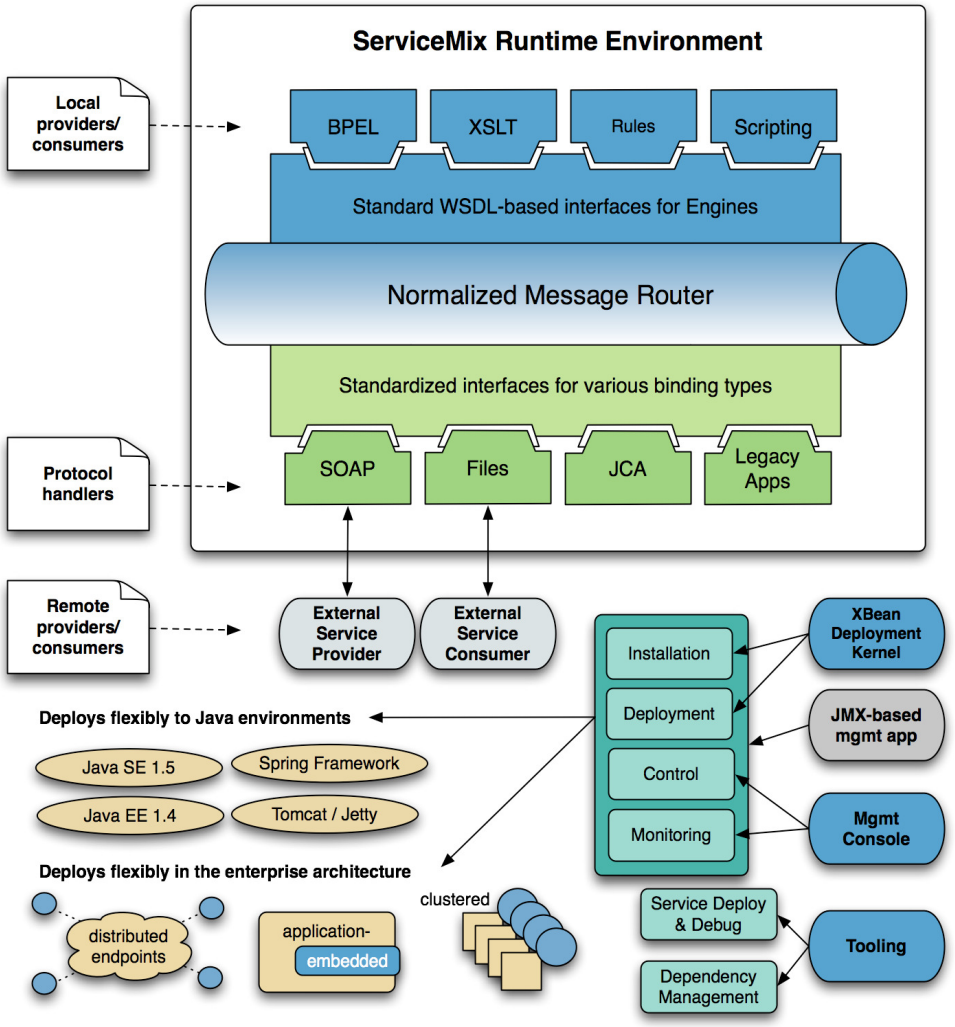
- Service Engine (SE) Components
  - Plug-in components provide business logic, transformational services etc.
  - Some SE's can contain other SE's, “container of containers”
  - **ServiceMix** provides SE's for JMS, EJB, Drools, HTTP and many many more
- Binding (BC) Components
  - Provide connectivity to components external to the JBI environment
  - Send and receive messages for some particular protocol and transports
  - Provide a sort of adapter for messages and the transport and protocol

IONA Open Source : : FUSE ESB Product

Three perspectives of FUSE:

At the top is the JBI-based internal structure of the ServiceMix runtime environment. Local providers and protocol handlers are incorporated through standards-based interfaces, to act as Service Engines and Binding Components for message-processing. External services are connected via the Binding Components.

The deployment illustration on the lower left shows that ServiceMix instances can be distributed, embedded with applications, clustered, and deployed to virtually any Java runtime environment.



On the lower right, we see how the ServiceMix environment is deployed, managed and monitored through standard tools and protocols.

Application Programmer models

Choice of POJO Service models

- JSR 181 / JAX-WS
- SCA and Apache Tuscany for a new simplified POJO model for SOA
- EJB 3
- Spring Remoting

JSR 181 / JAX-WS

- Defines annotations on services to describe how to expose them over SOAP & as WSDL endpoints
- Uses JAXB 2.0 as the XML marshalling layer
- The new standard for working with POJOs and WS
- JAX-RPC successor
- Java 5 dependency

FUSE Services Framework

- Advanced Services Enablement for SOA
  - Quickly service enable existing or new systems
  - Services are extensible for easy integration within a SOA or integration pattern
- Based on Apache CXF
  - Web Services Standards Support
  - Frontends
  - Ease of use
  - Flexible Deployment
  - Binary and Legacy Protocol Support

Support for Standards

- JAX-WS, JAX-WSA, JSR-181, and SAAJ
- SOAP 1.1, 1.2, WS-I BasicProfile, WS-Security, WS-Addressing, WS-RM and WS-Policy
- WSDL 1.1 and 2.0
- MTOM

Multiple Transports, Bindings, Data Bindings, and Formats

- Bindings: SOAP, REST/HTTP
- Data bindings: JAXB 2.0, Aegis.(XMLBeans, Castor and JiBX will be supported in CXF 2.1)
- Formats: XML, JSON
- Transports: HTTP, Servlet, JMS, and Jabber transports
- Extensibility API allows additional bindings for CXF, enabling additional message format support such as CSV and fixed record length

Flexible Deployment

- Lightweight containers: deploy services in Tomcat or Spring-based containers
- JBI integration: deploy as a service engine in a JBI container such as ServiceMix, OpenESB, etc.
- SCA integration: deploy in an SCA container such as Tuscany
- J2EE integration: deploy services in J2EE application servers such as Geronimo, JOnAS, JBoss, WebLogic, and WebSphere
- Standalone Java client/server

Support for Multiple Programming Languages

- Full support for JAX-WS 2.0 client/server programming model
- JAX-WS 2.0 synchronous, asynchronous and one-way API's
- JAX-WS 2.0 Dynamic Invocation Interface (DII) API
- Support for wrapped and non-wrapped styles
- XML messaging API
- Support for JavaScript and ECMAScript 4 XML (E4X) - both client and server
- Support for CORBA with Orbix and Yoko
- Support for SCA withTuscany
- Support for JBI with ServiceMix

Code Generation

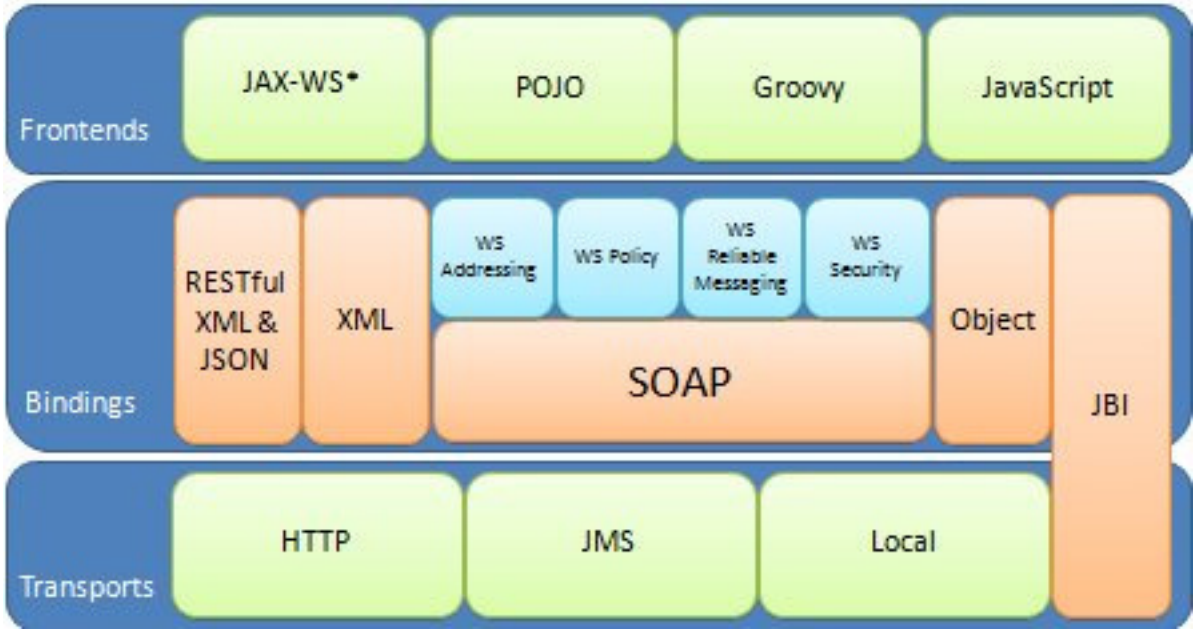
- Java to WSDL
- WSDL to Java
- XSD to WSDL
- WSDL to XML
- WSDL to SOAP
- WSDL to service

FUSE Services Framework Architecture

The overall CXF architecture is primarily made up of the following parts:

- **Bus**: This is the backbone of the CXF architecture.
- **Messaging & Interceptors**: These provide the low level message and pipeline layer upon which most functionality is built.
- **Front ends**: Frontends provide a programming model to create services (e.g. JAX-WS).
- **Services**: Services host a Service model which is a WSDL-like model which describes the service.
- **Bindings**: Bindings provide the functionality to interpret the protocol (i.e. SOAP, REST, Corba).
- **Transports**: Destinations and Conduits make up the transport abstraction that CXF uses to achieve transport neutrality.

FUSE SERVICES FRAMEWORK ARCHITECTURE



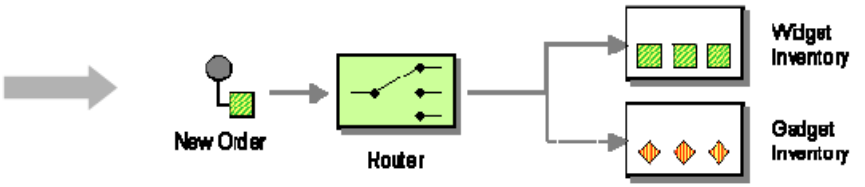
FUSE Mediation Router

- Built upon the Plain Old Java Object (POJO)-based Apache Camel project, FUSE Mediation Router provides a lightweight, rules-based and transport-neutral routing and mediation engine that leverages Enterprise Integration Patterns.
- Defines a high-level language to describe EIPs

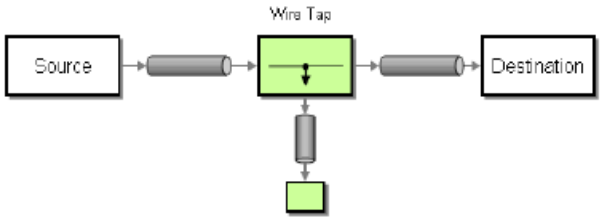
  This language very closely maps to components that work inside Spring 2.

Enterprise Integration Patterns

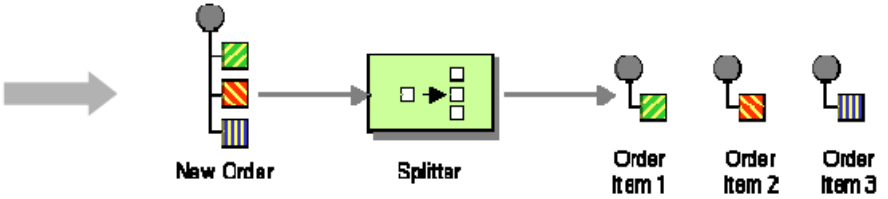
- Enterprise Integration Pattern (EIP)
  - Use routing patterns to handle Message Exchange
- Some patterns are
  - Content-Based Router



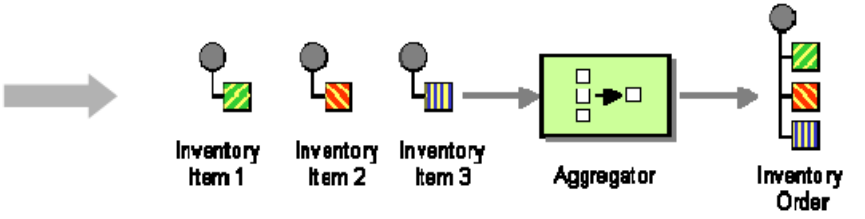
  - Wire Trap



  - XPath Splitter



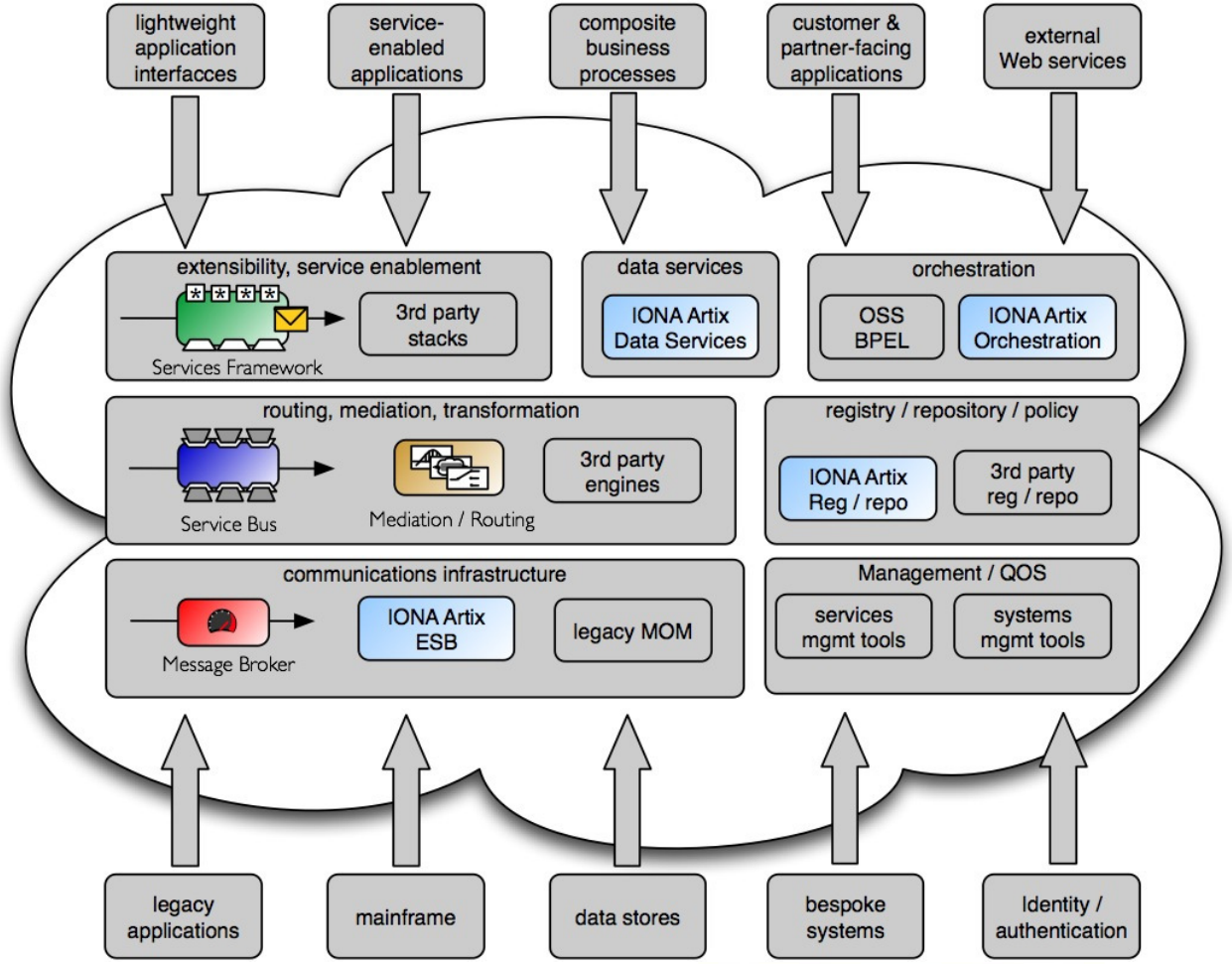
  - Split-Aggregator



  - …

Summary

IONA FUSE In A Distributed SOA Backplane



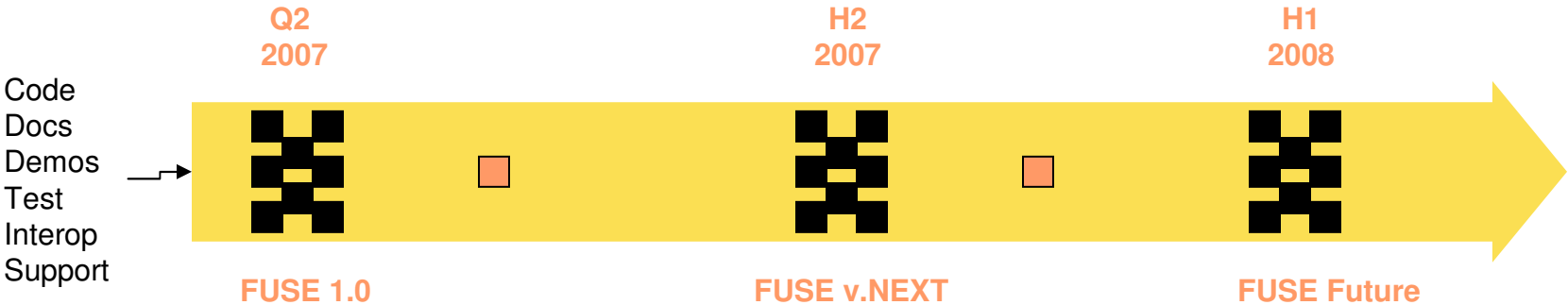
The open source components form the core of the SOA Backplane, yet support the use of additional open source and commercial technologies.

Each organization's deployment and configuration will vary, depending on business and technical requirements.

Components interoperate with an ecosystem of complementary technology components, both open and closed

IONA Open Source : : Roadmap

FUSE Product Development 2007-2008



FUSE 1.0 Themes

- Repackage FUSE stack into discrete products aligned with Apache projects
- IONA services and support team trained up on FUSE offerings
- Open.iona.com infrastructure launched
- Bi-weekly cumulative fixpacks

FUSE v.NEXT Themes

- Synchronize with community projects
- Increased integration and additional samples of FUSE products together
- Code simplification
- Test cases and documentation for all supported features
- JBI tools on Eclipse 3.3
- New management tools
- New EIP (Camel) tools

FUSE Future Themes

- OSGi deployment model across FUSE portfolio
- Continued support for JBI 1.0, new support for JBI 2.0
- Enhanced clustering

Open Source the IONA way

Open Source License FUSE ESB is available under an open source license that is based on the Apache License, Version 2.0.	The FUSE product license allows FUSE ESB to be used at no charge, and distributed as a part of any open source or commercial solution. The source code is publicly available and can be modified in any way. Modifications to FUSE ESB may be published or kept confidential.
Professionally Supported IONA provides complete 24x7 support, plus training and technical services for FUSE ESB.	Enterprises can use FUSE ESB as the runtime backbone of their SOA with confidence that their infrastructure is backed-up by an enterprise-class support team. Additional training, consulting and customization of the platform are readily available from the product's architects.
Community Strength FUSE ESB has support and contribution from a leading community of enterprise Java developers.	With an active base of contributors and users, FUSE ESB has been deployed in a wide variety of IT environments, and proven in thousands of business applications.

'...under the right circumstances, groups are remarkably intelligent, and are often smarter than the smartest people in them.'

Making Software Work Together™

For More Information

For IONA Jumpstart Program go to

http://www.iona.com/Jumpstart

Robert Kilker, robert.kilker@iona.com
Michelle Davis michelle.davis@iona.com

Open Source distributed SOA infrastructure download at
http://open.iona.com
