
# Overview of LogicBlaze FUSE


07/06 Version 1.2

## Overview of LogicBlaze FUSE

Welcome to LogicBlaze FUSE, the open source SOA platform. This overview document gives a general introduction to Service Oriented Architecture, and the unique capabilities that LogicBlaze FUSE can provide as you integrate and provide access to your systems and services. This brief overview concludes with an outline of the structure of the LogicBlaze FUSE platform.

LogicBlaze FUSE is a distribution of open source components that, taken together, provide a runtime platform for SOA. LogicBlaze FUSE is built around the ServiceMix enterprise service bus (ESB), and provides it with a set of complementary technologies that support the essential runtime requirements of a service-oriented architecture.

# 1. LogicBlaze FUSE: A Platform for Service Oriented Architecture

*Service-Oriented Architecture* is a unique way of thinking about systems: specifically, how to provide access to systems and integrate their operations to support real-world business processes. The purpose of SOA is to achieve a closer alignment of IT capabilities with operational requirements, by defining the processes that "business users" must support, and composing those processes from component services.

> The expression *Service Oriented Architecture* (capitalized, or abbreviated "SOA") refers to a set of principles or patterns of software design. However, each organization that is "implementing SOA" develops its own unique set of services, and organizes and runs them according to its specific technology requirements and policies (in addition to the overall principles of SOA). To avoid confusion, we describe the general principles as "SOA" (capitalized), and we refer to the specific implementation of SOA practices by an organization as its "service oriented architecture" (no caps).

A *service* is typically defined as "an operation whose interface is abstracted from its implementation," which is to say that to access and invoke a service to perform a function, you don't need to know what technology lies behind the service. Many services in a service oriented architecture are "Web services," that is, services that define their interfaces and transport their data according to commonly agreed XML-based standards such as WSDL and SOAP. However, many organizations provide some services (or some parts of certain composite services) using other technologies, such as JCA, EDI or FTP. In addition, an organization may use Web service standards in conjunction with other technologies, such as encapsulating message data in a SOAP envelope, but transmitting it over JMS.

The distinctive characteristic of SOA is that regardless of how the organization chooses to expose the capabilities of its systems as services, these services exist within a common framework that specifies how services are developed, organized, identified, accessed, connected, monitored, managed, etc. When we say that LogicBlaze FUSE is a "platform for Service Oriented Architecture," we mean that LogicBlaze FUSE provides the core technologies that the organization uses to implement that framework, ensure that services execute according to the design of the framework, and improve and expand the framework over time.

# 2. Why SOA?

SOA is often summarized as the more nimble successor to the previous generation of business integration technologies, such as enterprise application integration (EAI). While it is true that integration architects who had previously implemented EAI tend to focus on SOA these days, the migration from EAI to SOA is only one part of the SOA picture. Many organizations that never considered EAI are eagerly pursuing SOA. There's more to SOA than just "next generation EAI." Organizations look to SOA for a number of key benefits:

**Re-use**: The most commonly identified goal for SOA is to produce an extensive set of business services that can be used and re-used in any number of different business processes.

**Flexibility/Agility**: Agility refers to a business's ability to rapidly adapt IT operations to changing business requirements. Flexibility refers to the ability to implement change across all relevant systems environments and business organizations.

**Modularity**: Both Re-use and Agility are driven by modularity, the ability to abstract services as meaningful components, with the appropriate degree of granularity.

**Monitoring and Management**: To be effective in the enterprise, SOA requires a sophisticated view into the operation of distributed processes. At the same time, because the processes correspond to specifically defined business activities, SOA also offers a real-time view into business activity.

**Interoperability**: By relying on standards-based interface definitions, organizations can extend business processes to include partners and customers where appropriate.

**Leverage**: The principles of SOA are guided by a focus on extending the capabilities of existing systems, not replacing them. This is not only true of business applications, but existing middleware and integration technologies as well.

# 3. Advantages of LogicBlaze FUSE

**The First Open Source SOA Platform Solution**

- Lower cost, for both initial adoption and expansion over time.
- Access to all source code, and freedom to modify code as required.
- Access to an active development community, with no lock-in to any specific vendor.

**LogicBlaze-Certified Distribution**

- Code versions are validated for quality and interoperability.
- No hidden license issues - All components are released under the Apache 2.0 License unless specifically identified as an exception.
- "Whole product" support: LogicBlaze provides documentation, demonstration examples, support for the entire platform.

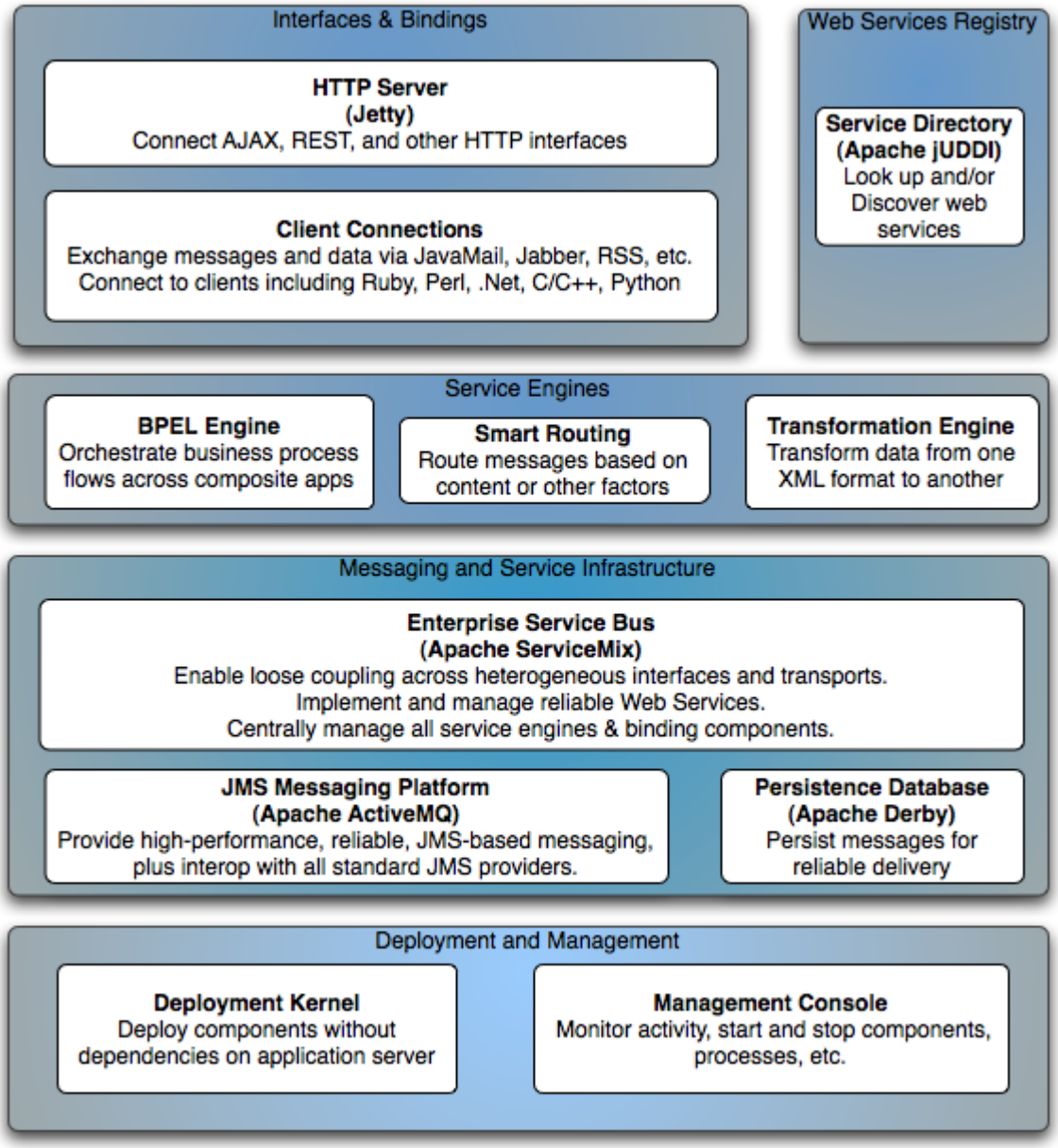
**Standards-based Design**

- Compatibility with existing systems - open source and commercial - through Java and Web services standards.
- Commitment to support standards going forward - both Java and Web services.
- Ability for each component to evolve or be swapped out independently of others.

**Professional Support and services from LogicBlaze**

- Enterprise-class support according to industry-standard service-level agreements.
- Continuous innovation and rapid issue resolution directly from members of the developer community.
- Unique model for support (the CoRE Network) including close collaboration and knowledge transfer.

# 4. Structure of LogicBlaze FUSE



## Deployment and management

To effectively Deploy and install the components of the LogicBlaze FUSE SOA Platform, a platform-independent installer is provided. One of the key advantages of the deployment kernel is its platform-independent mechanism, which is helpful to minimize dependencies.

The Management Console is a dashboard providing visibility into, and control over, the deployed components of the SOA platform, as well as service processes and system capacity.

## Messaging and service infrastructure

The heart of LogicBlaze FUSE is the ServiceMix enterprise service bus (ESB).

The ActiveMQ messaging platform provides reliable messaging, persistence and scalability through a JMS-compliant messaging system.

For persistence the messaging infrastructure includes the Apache Derby database.

## Service engines

**Orchestration**: LogicBlaze FUSE provides a complete orchestration engine, for managing service processes through workflow processes defined in standard BPEL files.

**Smart Routing**: LogicBlaze FUSE supports "smart routing," which includes capabilities such as dynamically determining the best route for message delivery or service execution; content-based routing; script-based routing driven by a rules engine, etc.

**Transformations**: Message and data transformations and supported through XSLT.

## Interfaces and bindings

Connect via both standard interfaces including WSDL, JMS, JCA etc. as well as other interfaces including RSS, AJAX (REST), IRC (Jabber), FTP, etc.

## Service registry

The Apache jUDDI provides a UDDI-compliant registry for service lookup.