



Fiorano
Enabling change at the speed of thought

www.fiorano.com

Message Driven SOA -- Enterprise Service Oriented Architecture

Dynamic Web Services

Creating a unified computing infrastructure

--- Atul Saini

AMERICA'S

Fiorano Software, Inc.
718 University Avenue Suite
212, Los Gatos,
CA 95032 USA
Tel: +1 408 354 3210
Fax: +1 408 354 0846
Toll-Free: +1 800 663 3621
Email: info@fiorano.com

EMEA

Fiorano Software Ltd.
3000 Hillswood Drive Hillswood
Business Park Chertsey Surrey
KT16 0RS UK
Tel: +44 (0) 1932 895005
Fax: +44 (0) 1932 325413
Email: info_uk@fiorano.com

APAC

Fiorano Software Pte. Ltd.
Level 42, Suntec Tower Three 8
Temasek Boulevard 038988
Singapore
Tel: +65 68292234
Fax: +65 68292235
Email: info_asiapac@fiorano.com

Entire contents © Fiorano Software and Affiliates. All rights reserved. Reproduction of this document in any form without prior written permission is forbidden. The information contained herein has been obtained from sources believed to be reliable. Fiorano disclaims all warranties as to the accuracy, completeness or adequacy of such information. Fiorano shall have no liability for errors, omissions or inadequacies in the information contained herein or for interpretations thereof. The opinions expressed herein are subject to change without notice.

DYNAMIC WEB SERVICES

Creating a unified computing infrastructure

Executive Summary

In today's business environment, vital information exists in many disparate systems throughout the enterprise. These include:

- Customer Relationship Management (CRM) systems
- Financial and accounting systems
- Enterprise Resource Planning (ERP) systems
- Web servers
- Legacy systems

This information must be accessible from multiple touch-points. These systems typically have evolved internally in isolated silos. Despite their focus on product lines and departmental functions rather than customer needs, these systems nonetheless contain strategic information.

Additional systems containing important information are commonly acquired through mergers and acquisitions. So, businesses urgently seek integration and Business Process Management (BPM) solutions that can extend the longevity and increase the productivity of prior investments.

Several platforms today implement Web services — most notably Java 2 Enterprise Edition (J2EE) application servers from various vendors as well as Microsoft's .NET. Web services provide a standard's based solution to access data both within and across enterprises. Implementing integration solutions based on Web services allows enterprises to dramatically reduce the traditionally high costs associated with EAI, since Web services solutions are based on industry-standards and require no specialized knowledge of proprietary applications and products. As such, many enterprises are committed to Web service based solutions for EAI and Business Process Management (BPM).

Platform requirements for EAI

Not all Web services platforms can be used 'out of the box' for Enterprise Application Integration (EAI). The problems involved in implementing distributed EAI and BPM solutions impose particular requirements on the underlying platform. The rest of this section discusses these requirements, which are essential for efficiency and extensibility.

Fully distributed, peer to peer event-based architecture

Real world business processes are typically distributed across multiple applications and hardware and software systems; they're also event-based, in that the processes are typically chained together via a series of events. As such, a platform that endeavors to integrate business processes within and across the enterprise needs to be fully distributed across the particular nodes on which the processes run. An effective EAI/BPM platform must have a peer-to-peer architecture, developed with a software engineering methodology that emphasizes the variables of security, control, dynamic accessibility and flexibility. The platform must also be symmetric, which implies that the same event based infrastructure software needs to run on all machines within the enterprise. Most current EAI and BPM solutions control the business processes through a centralized hub. Changes to applications, or additions of new applications, require changes at the centralized hub. Further, all data exchanged between applications needs to traverse the central hub. This type of topology restriction is inefficient and inflexible, and leads to bottlenecks in a distributed system.

Process route transparency

An EAI infrastructure platform should offer complete transparency of information flow between the different processes that compose the solution. To provide flexible change management capabilities without any programming, individual components within the system should be oblivious to the routing of information between processes.

Flexibility through service-based architecture

An EAI platform should make it simple to deploy, manage and change participating processes. This implies a component-based architecture, in which applications are composed of 'coarse-grained' components loosely bound to each other via event-based messaging, with each service potentially running in a separate process. This enables a rapid deployment model, decreasing the lead time required for solution implementation. The architecture should also support tools that enable on-the-fly modification of processes, so analysts can change and instantly redeploy processes to meet changing business requirements.

Remote deployment, monitoring and management

An EAI infrastructure platform should provide a means for deploying, managing and monitoring processes over the network. It should also support real-time monitoring of data and events. This will reduce significantly the time and cost of deploying distributed business processes.

Enterprise standards support

Support for existing enterprise standards for data exchange, messaging and BPM dramatically reduce the total cost of ownership of EAI infrastructure over time. It precludes the need for specialized or proprietary knowledge in deploying solutions. Since content needs to be exchanged between business partners, extensible Markup Language (XML) messages and documents should be the desired format. Most businesses want to leverage existing infrastructure for messaging, components, services, enterprise data, Lightweight Directory Access Protocol (LDAP) directory servers, e-mail systems, etc., so an EAI platform needs to support all these standards easily.

Problems with current web service architectures

While interest in Web services for EAI is high, current Web services platforms suffer from many disadvantages that limit their utility for EAI problems. The principal problems of existing Web services platforms (including J2EEbased application servers and Microsoft .NET) are:

Inefficient request/reply semantics

Existing Web services platforms are typically based on request/reply semantics. Each Web service is contacted using a Simple Object Access Protocol (SOAP) request, which blocks indefinitely until the result is returned to the caller. The platforms are based on polling and require manual coding to capture the output of the Web service and pass it on to a different Web service or application within a workflow. The lack of an event-enabled bus for data flow between Web services makes existing Web services platforms ineffective for EAI solutions.

Centralized, non-scalable architectures

Current EAI solutions control the business processes through a centralized hub. Changes to applications, or new applications, require changes at the centralized hub. Further, all data exchanged between applications needs to traverse the central hub. This topology leads to bottlenecks in a distributed system and causes significant inefficiency and inflexibility.

Lack of remote deployment, monitoring and logging

Typical distributed systems have many services running on different machines across the network. If any of these fails, other components in the system need to be alerted. An EAI platform should provide built-in services for such distributed monitoring, together with remote tracing and logging facilities.

Another requirement is for the automatic deployment of services on remote nodes, which results in decreased maintenance and implementation costs. Existing centralized Web services platforms have no support for remote service deployment since the spokes of the network have no infrastructure-level support for deployment, resulting in increased lead times for solution development and deployment.

Using a services operating platform

Current Web services infrastructure platforms don't solve core problems related to EAI and BPM.

These problems are best solved using event-enabled Web services, which we refer to as dynamic Web services. These are best implemented on top of the fully distributed, event-enabled infrastructure we refer to as a Services Operating Platform (SOP). An SOP facilitates the composition, deployment and modification of distributed enterprise applications using visual tools. A powerful advantage of an SOP is that it provides a single cohesive platform for a company's EAI, B2B integration, BPM, collaboration and general distributed computing requirements.

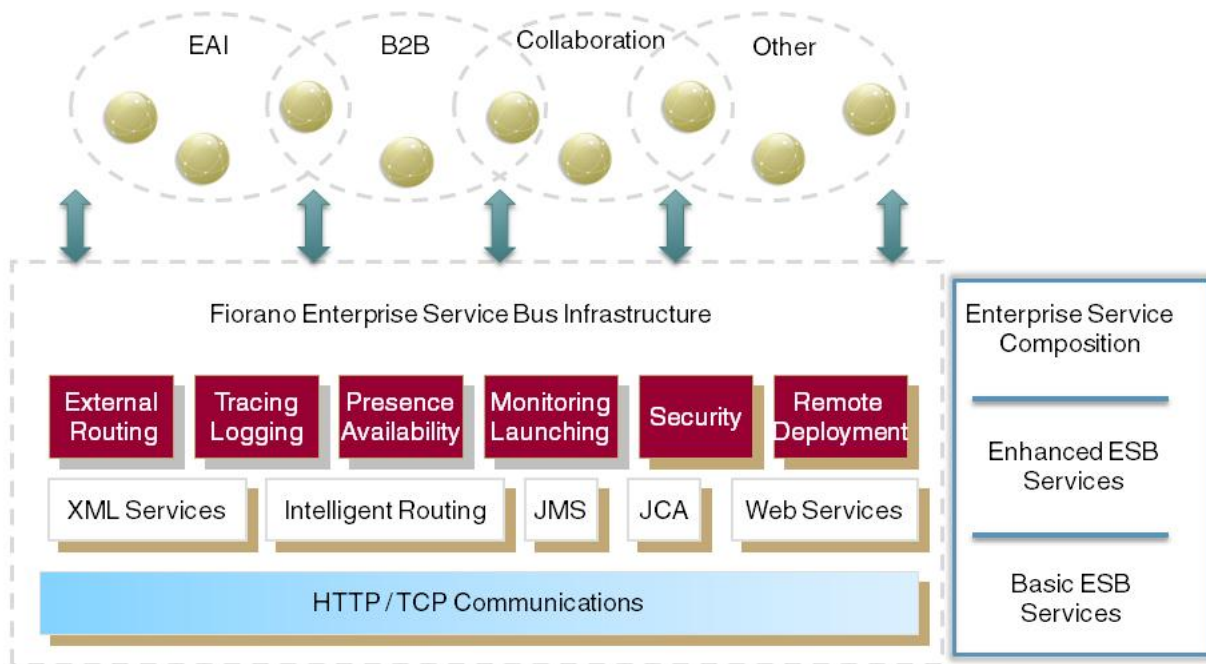


Figure 1: Typical SOP Architecture - Replicated on Each Node of the Network

Figure 1 shows the architecture of a typical SOP node. In a typical enterprise, a daemon based on this architecture runs on each network node, with all the daemons being connected via a message bus, as illustrated in Figure 2. As shown in Figure 1, an SOP provides built-in support for event-based messaging, transparent process routing, remote monitoring, tracing and logging, scheduling, presence and availability, security and remote deployment.

An SOP is a pure infrastructure platform that allows implementation of software services of any kind, including Web services. It differs from existing Web services implementations in that it requires a fully distributed peer-to-peer, message enabled infrastructure instead of a centralized request/reply infrastructure. An SOP offers these advantages to customers:

Service-based application composition

Applications deployed on top of an SOP are composed of one or more services (or components), each of which may optionally run in a separate process. Services may be written in any language and communicate with each other via XML messages. This allows the construction of flexible, easy-to-modify systems.

On-the-fly business process deployment and changes

Dynamic Web services implemented on top of an SOP are 'coarse-grained' services that are ideally suited for easy change and replacement. Further, services implemented on an SOP are oblivious to the routing of information between service instances. This makes it easy for business

processes to be modified on-the fly by dynamic service replacement or additions. An SOP also supports the run-time deployment of services, allowing changed business processes to be deployed across the network instantly, reducing solution deployment costs by an order of magnitude compared to traditional ones.

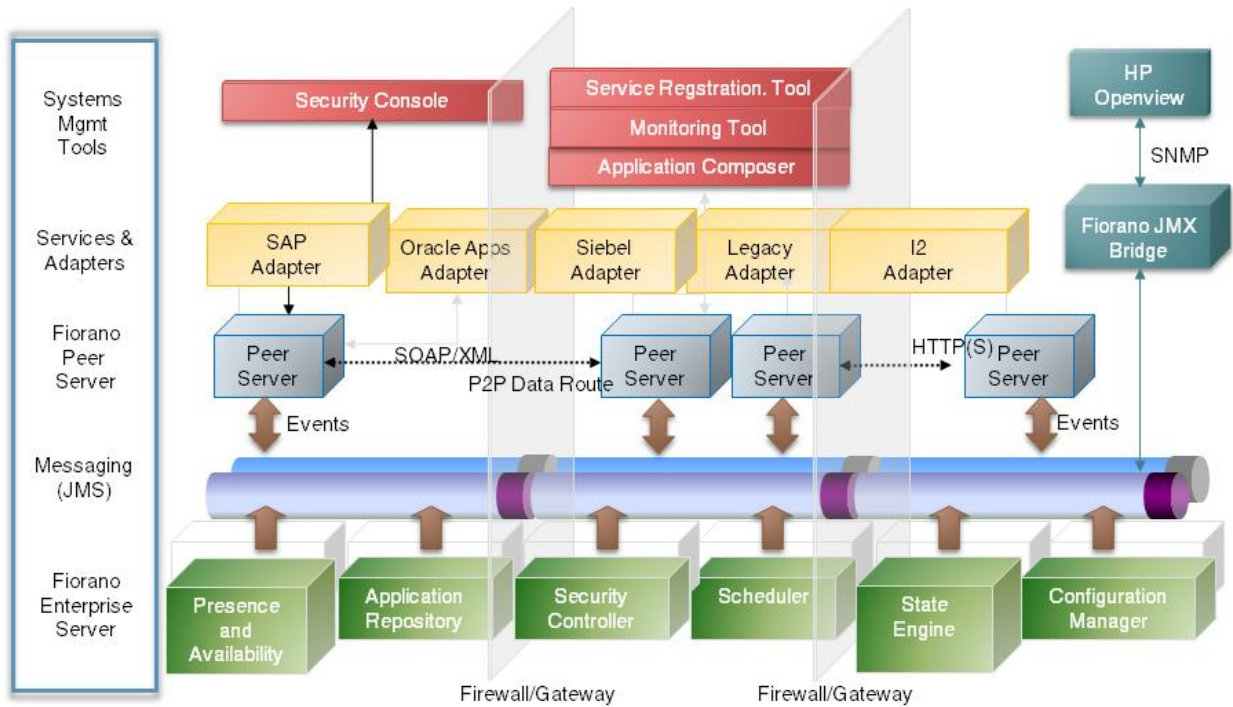


Figure 2: Detailed SOP System Architecture

Fault tolerance, scalability, reliability

The symmetric, peer-to-peer architecture of an SOP provides a highly scalable, reliable platform with no single point of failure. The absence of a centralized hub gives application developers the ultimate flexibility in defining the network topology of choice to route data directly between services as required.

Component-level security

An SOP gives administrators full control over which services are executed where while ensuring security. The SOP lets administrators set any number of security attributes for each service, and provides tools to configure security settings on any peer server across the network.

Run-time monitoring, tracing, and logging

An SOP includes native, service-level support for run-time monitoring, tracing and logging. All services can be monitored using visual tools. Trace levels can be dynamically changed within already deployed services and debug logs can be routed to software tools on any node. These features greatly simplify the development, deployment and debugging of distributed applications running on an SOP.

Support for visual process composition and change-management tools

An SOP includes Application Program Interfaces (APIs) to support the development of software tools that allow rapid composition of distributed applications by using ‘drag-and-drop’ visual icons from a services palette to construct a business process on the application composer. The same environment also allows monitoring of all services across the distributed network.

Support for enterprise standards

An SOP uses standards-based Java Messaging Service (JMS) to exchange standard XML messages. Using eXtensible Style sheet Language Transformation (XSLT), data can be translated between Fiorano ESBTM (Fiorano Enterprise Service Bus) and any other application or service format. A component Solution Development Kit (SDK) lets users connect any existing components or services for use within the SOP. Other enterprise standard such as Simple Mail Transfer Protocol (SMTP) and Simple Network Management Protocol (SNMP) enables integration with other enterprise systems.

Support for general-purpose services

An SOP also supports integration of components that aren't implemented specifically as Web services. A general-purpose service can have any constant number of input and output ports, while a Web service logically only has one input and one output. Support for general services gives enterprises additional flexibility in their integration efforts. Often, it's easier to model legacy system code as a general-purpose software service than specifically as a Web service

Summary

Existing approaches to the EAI, BPM and related problems of enterprises don't fully satisfy the underlying infrastructure requirements these problems demand. Service-based architectures, implemented on top of a fully distributed, peer-to-peer event-driven system, bring a whole new level SOPs implemented this way address the needs of BPM at the application and human interaction levels. It addresses these needs within and across enterprises, and across multiple domains, including some areas that have traditionally been served by distinct solutions. An SOP thus allows an enterprise to create a unified computing infrastructure for all its distributed computing requirements,

About Fiorano Software

Fiorano Software (www.fiorano.com) is a leading provider of enterprise class business process integration and messaging infrastructure technology. Fiorano's network-centric solutions set a new paradigm in ROI, performance, interoperability and scalability. Global leaders including Fortune 500 companies such as Boeing, British Telecom, Credit Agricole Titres, Lockheed Martin, NASA, POSCO, Qwest Communications, Schlumberger and Vodafone among others have used Fiorano technology to deploy their enterprise nervous systems.

Note: This whitepaper has been reproduced from the original, which was published in eAI Journal.