Fiorano
Enabling change at the speed of thought

www.fiorano.com

Scaling Enterprise SOA Deployments

Benefits of a Message-Driven Approach Powering Next-Generation SOA Deployments

--- 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

SCALING ENTERPRISE SOA DEPLOYMENTS

Benefits of a Message-Driven Approach Powering Next-Generation SOA Deployments

Executive Summary

The first wave of integrating storage, compute and networking hardware helped businesses move from client/server to internet-based peer-to-peer networks. A second wave of integrating applications on top of the hardware infrastructure promised to deliver unprecedented economies of scale. In today's enterprise IT model, applications exposed as services need to be integrated seamlessly with other applications distributed across the network to generate the best operational efficiencies. Messaging-oriented middleware is at the heart of enabling such a frictionless integration between a business' core assets: its applications and data residing on the network. However, integrating multi-vendor applications with diverse infrastructure and legacy applications is a daunting task.

In a recent report –a leading analyst reports that Global 3500 firms will spend an average of $6.4-Million in 2007 on Service-Oriented Integration and process management budgets, *and less than 35% of the projects come in on time and on budget.*

So why are SOA project schedules so unpredictable? Incumbent first-generation SOA *solutions* consist of a patch- work of point-products that are struggling to meet the performance, scalability, flexibility and security demanded by modern distributed business systems. In contrast to an age of irrational exuberance, throwing more hardware and customized software at the problem is no longer a viable option for most businesses.

Since early 2003, the Enterprise Service Bus (ESB) has emerged as a standards-based Services Orchestration Platform, typically built on a distributed peer-to-peer service-grid architecture that enables high-performance, scalable and affordable SOA solutions. A distributed Enterprise Service Grid typically consists of the following server components:

Enterprise Server (ES): The ES forms a centralized repository for security policies, business-process configuration and logs, monitoring, debugging facilities and reusable services
Peer Servers (PS): These lightweight servers can be installed on hand-held devices as easily as on back-end enterprise servers. With a local store-and-forward service, PSs enable a peer-to-peer data transport thereby generating huge economies of scale
MQServer: A standards-based JMS messaging server that supports an arbitrary number of PSs communicating with an arbitrary number of ESs – thereby enabling a *brokered peer-to-peer services* orchestration platform, in which control information flows through a centralized server (the ES) while data flows concurrently between connected Peer Servers (PSs).
Services Orchestration Tools: A set of developer-centric tools that enable the composition, governance, monitoring, debugging and creation of reusable services
Built-in Adapters: ESBs typically include hundreds of available pre-built adapters for packaged as well as legacy applications.

The distributed architecture of a modern Enterprise Service Grid provides ease of use and real-time responses from integrated services as discussed in the remainder of this document.

Scalable Performance

Applications and Data are the lifeblood of a business. Faced with dynamically changing customer, partner and supplier requirements, businesses need to integrate a large number of applications – exposed as services - in multiple data formats across dispersed geographic locations. Traditional middleware integration solutions consist of Business-Process GUIs hooked into a distributed services layer as shown below. An arbitrary number of standalone applications can be integrated to represent a workflow that could model, for instance, a financial transaction processing package. The hub-and-spoke messaging

middleware also uses standards based protocols (TCP/IP, HTTP, others) to process and manage the messages supporting a set of workflows.

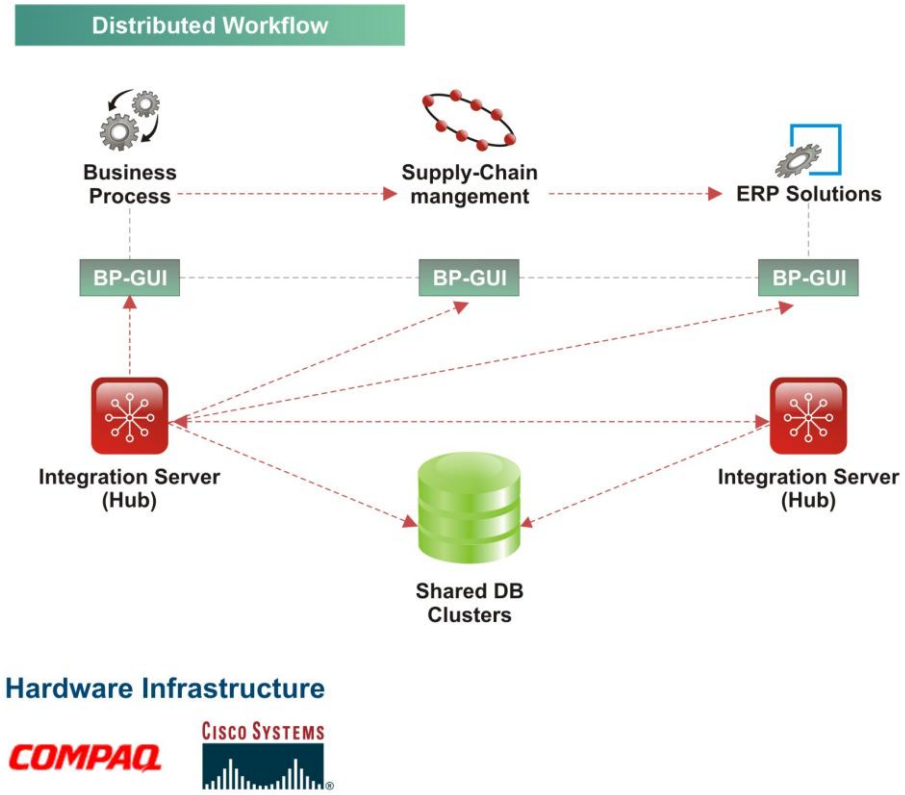

Figure 1: Traditional Hub-and-Spoke messaging architectures

All messaging traffic between the distributed Services consisting of control and data packets must traverse the HUB - also referred to as an Integration Server (IS). As traffic requirements scale, the IS becomes a performance bottleneck and a single point of failure. IS clusters can help create redundancies that increase availability – at a high cost of increased complexity and budgets.

Most IS-clusters rely on multicast mechanisms to relay messages. The multicast mechanism is very inefficient in terms of bandwidth usage, especially when delivering certified/guaranteed messages – as demonstrated in several benchmarks [1]. While composing Services into composite applications remains the key value of an SOA solution, underlying hub-and-spoke architectures are weak in delivering scalable performance and guaranteed message delivery at high rates.

The **Brokered Peer-to-Peer** architecture of a **modern distributed Service-Grid platform** provides a near-linear scalable performance at highly affordable costs. The distributed Service-Grid architecture enables the splitting up of data traffic amongst an arbitrary number of lightweight peer servers, and control traffic with enterprise servers supporting all centralized functions. An Enterprise Service Grid Network consists of Peer Servers (PS) connected to a set of Enterprise Servers (ES) via a JMS messaging backbone as shown in Figure 2 below.

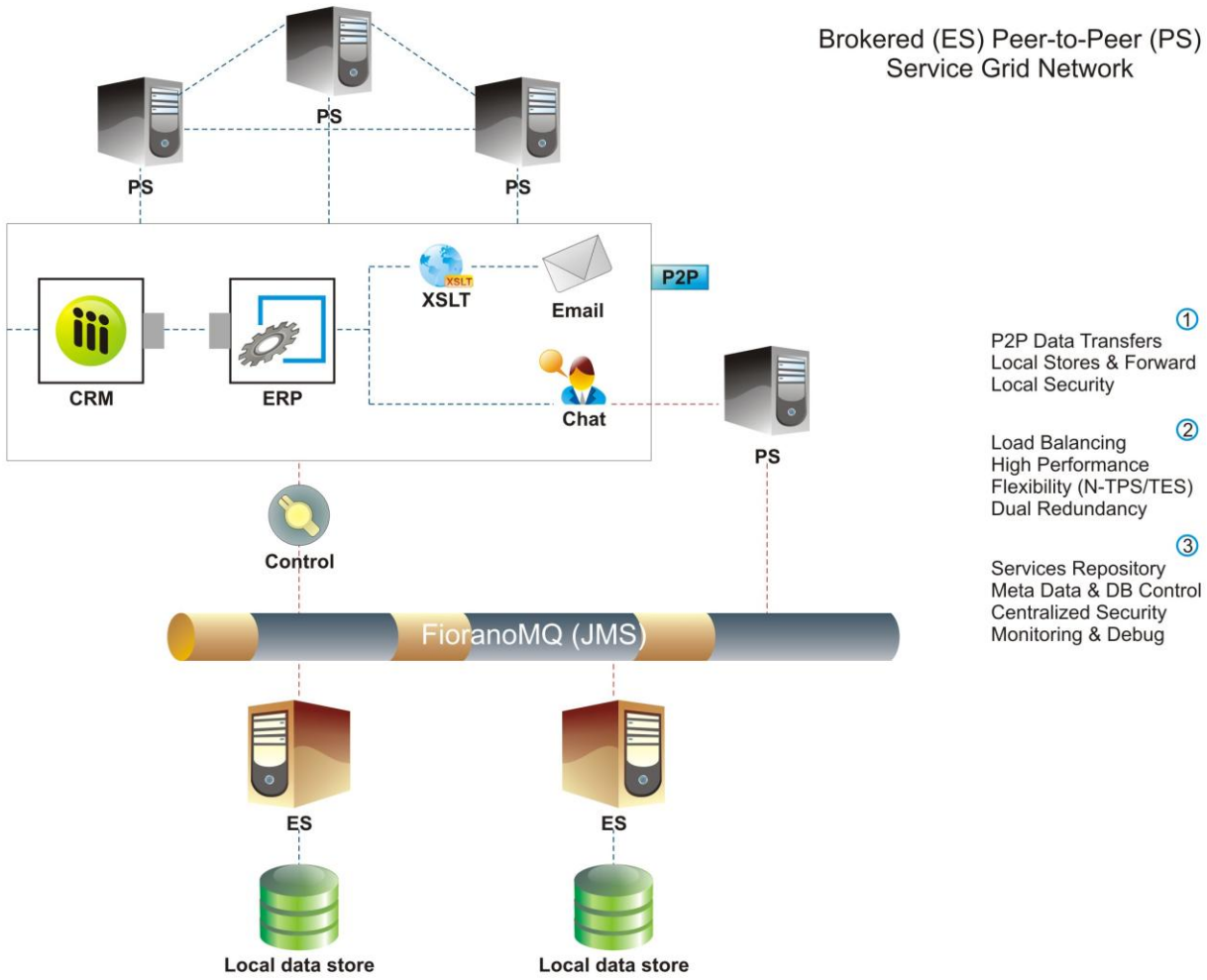

Figure 2: ESB Peer Network consisting of Servers (ES and PS) and MQ transport

As seen in Figure 2 above, the distributed Peer-to-Peer architecture of an Enterprise Service Grid provides several benefits:

1. **Store and Forward**. Peer Servers (at $1K/each) with their local store/forward mechanisms and a micro-messaging server enable fine-grain control over performance scalability of data traffic
2. **Enterprise-Class Backbone**. JMS compliant MQ servers typically support over 10000 messages/sec and a large number of concurrent connections thereby providing a very scalable, enterprise-class messaging backbone. With a dual-redundant configuration that supports fail over, the MQ backbone enables deploying an arbitrary number of PS (data traffic) and ES (control traffic) at very affordable costs. This enables a build-as-you-grow scalability model with no downtimes or added complexities
3. **Enterprise Servers** (ES) are constantly updated with all control traffic from any of the PS peers. The MQServer arbitrates between outgoing traffic from the ES to the PS servers and performs a many-to-one de-multiplexing for this traffic. The JMS-compliant MQServer typically also supports sophisticated load-balancing algorithms such as round-robin, weighted round-robin and Intelligent Load Balancing (ILB) – to provide efficient usage of all PS and ES deployments in the system.

The distributed Service-Grid architecture inherently provides for a highly scalable performance envelope. SOA architects can achieve very fine-grained control to match their performance requirements without trading off scalability or reliability goals as measured by the following parameters:

- Number of messages per second (aggregate throughput of the system)
- Size of the messages (5KB, 100KB, 1MB, 5MB)
- Number of concurrent applications/users supported
- Messaging Quality of Service (MQOS): delivery class (normal v/s guaranteed delivery)
- MQOS: Total Latency (total application-to-application latency via Integration Servers)

A Service Grid Peer Network consisting of two ES and a handful of PS servers can deliver on-par performance of 100-500 messages/sec at a fraction of the cost of traditional hub-and-spoke based solutions as shown in Figure 3.

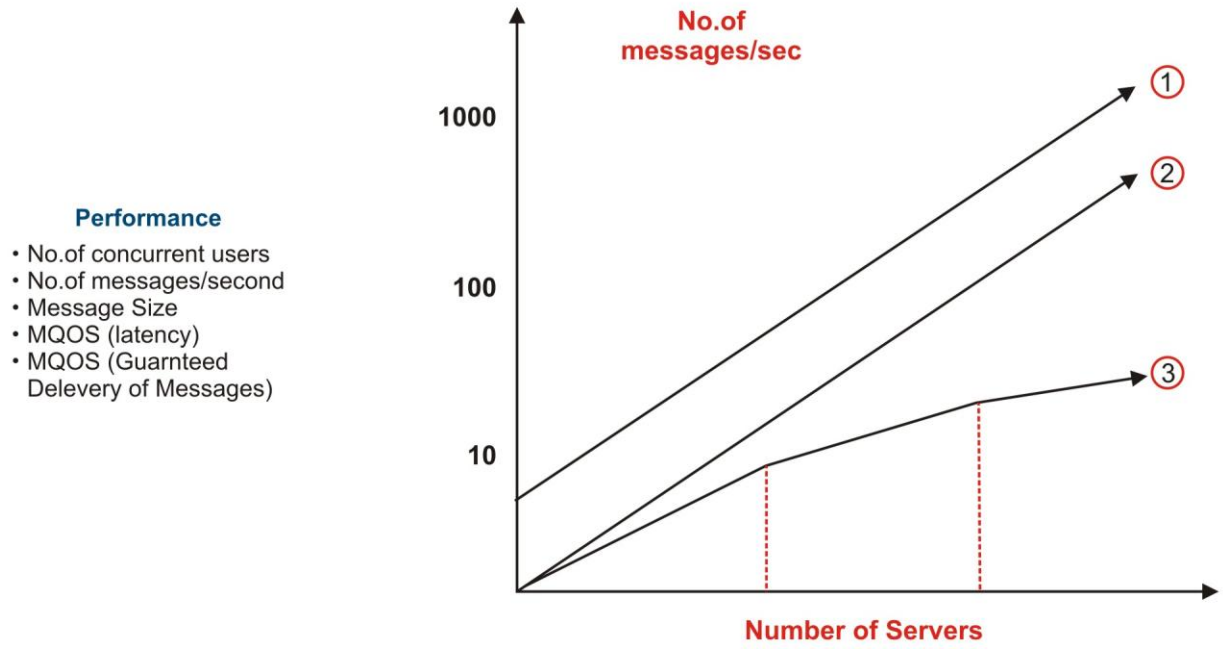

Figure 3: Scalable Performance with an ESB versus traditional EAI solutions

Peer-to-Peer Service Grids deliver unprecedented levels of throughput, capacity and MQOS, easily outscoring the nearest competitors by a factor of 10X. When performance benchmarks are normalized to the cost of software (and in some cases, the minimum hardware required), a P2P Service Grid delivers even higher returns on investment for the Performance benchmarks as seen in the plot in Figure 3 above.

1. Indicates a total cost of ($1K/PS * 20 + $25K/ES = $45K)
2. Indicates a total cost of ($1K/PS * 10 + $25K/ES * 2 = $60K) *and* provides a failover configuration in case of an ES failure.
3. Indicates an average cost of ($300K x 3 x 2 = $1800K) with three Integration Servers to provide an on-par performance of 100-500 messages/sec.

High Availability

System-level high availability is a function of the availability of its components. While evaluating SOA platform solutions, it is imperative that each of the SOA solution components is designed for and enhances the overall availability ratings with complete fail-over capabilities:

- Centralized Repositories (Configuration, Monitoring, Logs and Debugging data-stores, Governance policies, other centralized functions and tasks)

- Distributed Processes (GUIs that help compose and deploy workflows, any local store-and-forward data queues and associated software, distributed hardware)
- Messaging backbone
- Adapters to various applications
- Service-level fail-over (It should be possible to re-launch a failed service on a different machine in the case of distributed architectures for business/web-services)

The distributed P2P Service Grid architecture enables clear visibility at each of the component levels of an SOA infrastructure solution as listed above and shown in Figure 3 below.

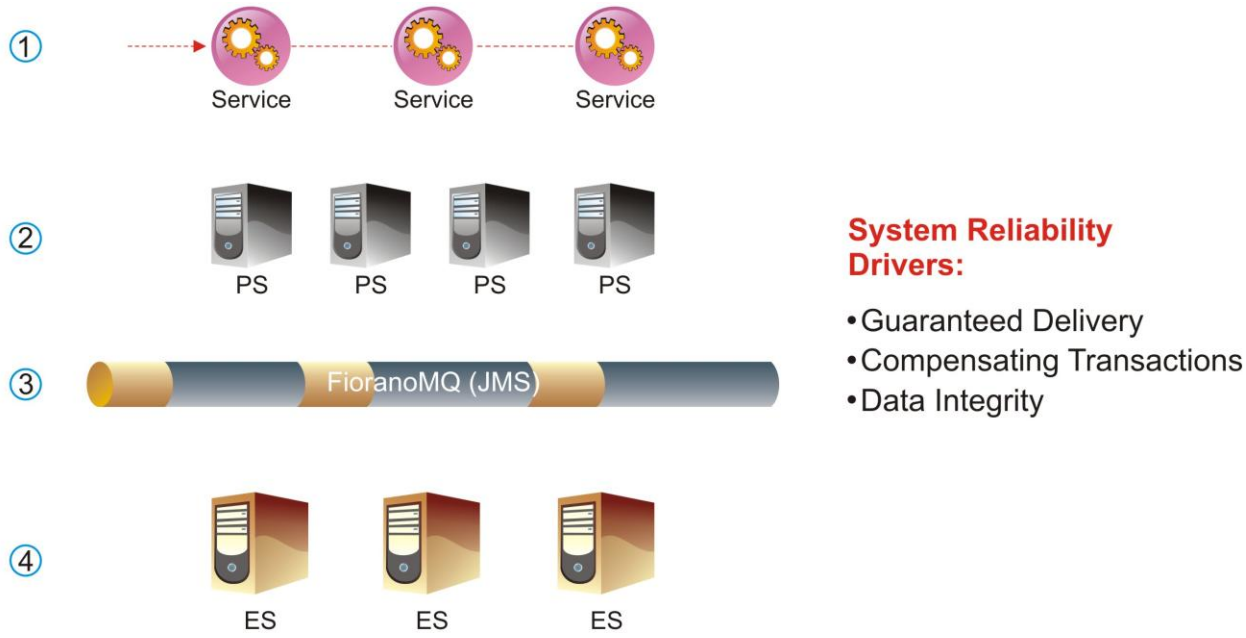

Figure 4: Enterprise Service Grid system components supporting overall high availability

The brokered peer-to-peer architecture of an Enterprise Service Grid enables any PS to designate another as its secondary or tertiary backup in case of failure. Mechanisms built into each PS monitor the health of its hardware and ensure that in case of hardware failure, all completed and pending transactions fail-over to the designated secondary transparent to the applications themselves.

Control traffic from a PS - for example, the creation and registration of a new service in the workflow - is immediately sent to multiple ES's via the MQ messaging backbone. Each ES has its own data-store – unlike hub-and-spoke solutions that have shared data-stores between Integration Servers. In case of an ES failure, a designated secondary ES transparently takes over its functionality, with no loss of transaction-level details.

The MQ messaging backbone that includes intelligent load-balancing of messaging traffic and supports both the ES and PS instances, is itself built as a redundant system to support fail-over. Thus, the Enterprise Service Grid enables a self-healing network of peers that protect against any component-level failure.

Traditional hub-and-spoke architectures in contrast necessitate replicating expensive Integration Servers to achieve reliability through redundancy. Even the touted "Federated Integration Servers" from Tibco or the "Distributed Peer-to-Peer" solutions from IBM rely on shared memory implementations that become a single point of failure. Increasing reliability raises costs and complexities dramatically, which in turn can adversely affect overall performance and scalabilities.

Experience with a variety of customer implementations shows that the Enterprise Service Grid's self healing networks offer the highest levels of affordable reliability under various real-world operational scenarios.

Security

Messaging middleware is a critical component of any enterprise's overall security policy framework. Messages in transit between participating publishers and subscribers need to be secure and tamper-proof. Messages stored in local queues need to be secured as well. Messages once consumed need not be stored and should be purged from all databases that could be vulnerable to inspection by malicious agents.

An Enterprise Service Grid can be easily integrated into a wide range of security policies as shown below.

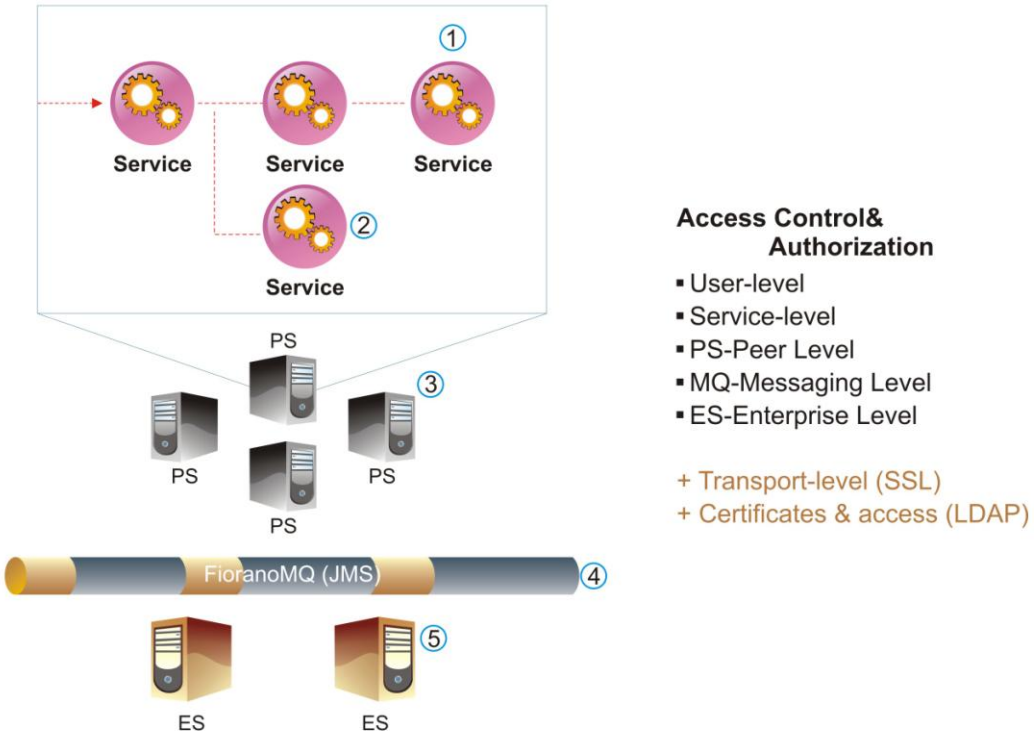

Figure 5: Easy integration and support of overall security policies with a Service Grid

An Enterprise Service Grid's Security and Governance policy includes empowering security administrators to limit access and control of a business process at the user, service, and component levels. Fine-grained security support enables a seamless security implementation inside and outside corporate firewalls. Messages can be securely transported using HTTPS and SSL protocols (client and server-side certificates and authentication supported). Security certificates can be stored redundantly in multiple Enterprise Servers (ES) across the Enterprise Service Grid network.

Extensibility

An SOA solution requires a vast array of reusable services and adapters. Adapters are software programs that deal with translating data stored in proprietary formats for applications such as PeopleSoft, Oracle DB and SAP – into standards for data transport such as XML. Initiatives such as JCA and JBI (Java Business Integration) have enabled standards to demystify adapter issues. Incumbent integration

solutions that build on hub-and-spoke architectures and adapters to integrate applications can only position themselves as extending with their adapters – deeper into the Integration space.

In contrast, integration is only *one* of several areas of application with an Enterprise Service Grid in an enterprise deployment. An IT-staff can write Enterprise Services easily in a wide variety of development languages (including Java, C, C++ and C# in most typical cases), and deploy these Services over the Service-Grid network as intelligent software agents to assist in automating the management of several IT tasks.

Consider, for example, a set of SMTP and Pop/iMAP dual-redundant servers as shown in Figure 6 (next page). Incoming emails are stored in disk space on the SMTP servers. In the event of a disk failure, these email messages need to be restored. Dual-redundant servers can provide for message backup; however, the mechanism with several scripts is process intensive, and a complexity overhead for IT staff. A more robust and easier alternative is to deploy PS agents on the SMTP servers and write service-based composite applications for failure recovery.

IT engineers can leverage the local store/forward of the Service Grid peer servers to save email messages that are not yet committed to a failed disk. Since the Service Grid peers are capable of real-time data transfers, these messages can be stored within the PS queues without being stored on the SMTP server disks. The distributed PS network can be thus leveraged to implement failover in a less complex manner.

This simple idea can be extended to solve several fail-over issues throughout an IT-stack. If the SMTP servers in the example above were substituted with Storage or Networking devices, the Enterprise Service Grid can power sophisticated – and yet simple to use – IT management networks.

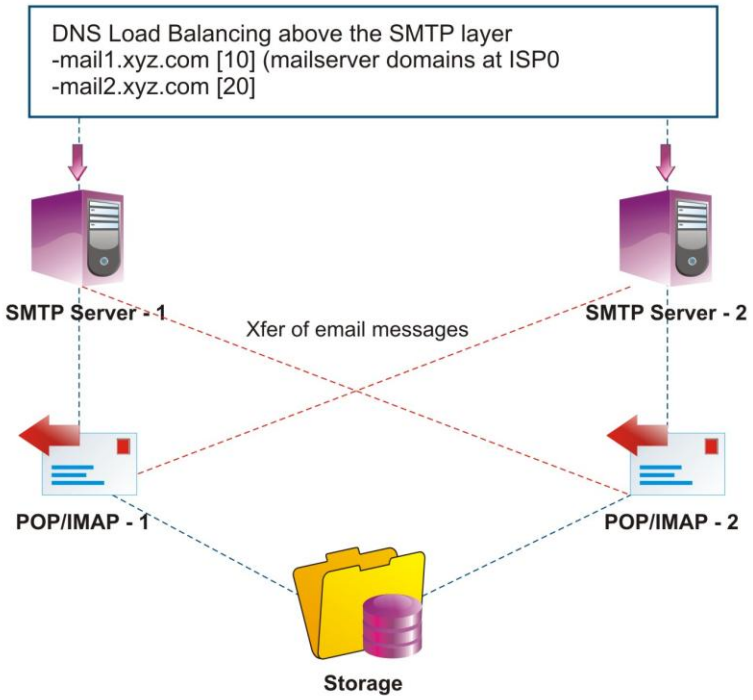

Figure 6: Distributed Peer-to-Peer IT Asset Management System

The Enterprise Service Grid's built-in security support at the Service, PS, ES and transport levels enables its applicability to solving a wide variety of distributed problems. PS daemons installed on machines inside and outside corporate firewalls, as seen in Figure 6, can help with standards-based remote monitoring, deployment and debugging tasks.

Mainstream collaborative applications can utilize the Service Grid as an integration and administration platform by building distributed application-specific services and adding to their Services Palette containing reusable services. The widespread adoption of a single platform and reusable services leads to dramatic savings in the Total Cost of Ownership (TCO) across enterprise solutions.

Summary

A leading analyst estimates that the global-3500 companies will spend an average of USD $6.4-Million each on their SOA projects in 2007. Further, as per Gartner, 65% of these projects take longer and can cost more than the projected budgets. One of the key reasons for this unpredictability and expensive costs is the weak hub-and-spoke messaging architecture powering the SOA projects. Modern Enterprise Service Grid's, with their brokered peer-to-peer architecture offer the next generation of efficiencies, ease of use and savings for a wide array of distributed computing projects including the SOA market as summarized in Table 1 below.

Table 1: Architecture-level benefits summary with an Enterprise Service Grid versus leading-edge SOA alternatives

Strategic Requirements	Service Grid: Architecture-Level Sustainable Advantages	Incumbent architecture-level weaknesses
Scalable Performance	**Data:** Peer-to-peer scalability within the Peer Servers ($1k/each) **Control:** Real-time replication of centralized information in Enterprise Servers ($25k/each) Messaging backbone: JMS-compliant MQ with built-in load-balancing and support for over 10000 msg/sec; amenable to be clustered as needed. **NET:** Fine-grained control over data and control messaging traffic with near-linear scalability in performance	Data and Control routed on a hub-and-spoke architecture. Even solutions claiming to be distributed architectures rely on shared memory semantics at the Integration server levels – thereby limiting scalability and performance.
High Availability	Self-healing ESB Network ensures a *24xForever* level of availability built-into the distributed network at no additional cost.	Requires redundant Integration-servers adding to the costs and management/deployment complexities
Security	Role-based security with fine-grain access control at the user, workflow, host (machine hosting the workflow) and individual service levels	Weak security models, difficult to integrate into enterprise security policy frameworks
Extensibility	Integration is only one of the potential areas of application. Enterprise Service Grid's are being considered for solving fail-over issues in routine IT management tasks, as well as deploying a variety of new solutions within an enterprise via flexible composite applications.	Limited to Integration usage.

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.