Comparing and Contrasting SOA Variants for Enterprise Application Integration

Today’s enterprise applications must be highly scalable and equipped for growth — adaptable and extensible enough to manage both current and future business systems and processes. With service-oriented architecture, IT organizations have a variety of options for integrating their infrastructure — from point-to-point and hub and spoke approaches, to an enterprise service bus architecture.

Executive Summary

Today, many organizations employ legacy technologies and frameworks to support a wide array of complex applications. The good news is that these systems have stood the test of time, remain fully functional, and continue to deliver exceptional value to the enterprise. But as businesses expand and technology continues its rapid evolution, systems must keep pace. This requires a tightly integrated, updated infrastructure that can support enterprise solutions. Such an approach is far more desirable than building new systems from scratch, which can be costly, time-consuming and subject to scheduling constraints. It also eliminates the need to take down existing applications to facilitate the migration of new ones.

Application integration solutions are made to evolve with technology’s irrepressible advances. By staying one step ahead of technology’s growth curve, the vendor community can offer organizations’ IT departments numerous options for integrating, reusing and customizing their business’s application infrastructure. Traditionally, these areas of the enterprise relied on proprietary vendor solutions to unify applications — forming the basis for enterprise application integration (EAI). However, even tried and true approaches have long lacked the ability to address many key technical requirements (particularly infrastructure security). They have also lacked the agility to embrace technological change, and the resources and tools needed to standardize existing applications that flow into, within and out of the enterprise.

As large organizations expand their business, the need to extend their existing applications and adopt newer and optimized business processes is often critical. Integration thus must ensure that applications can scale to new levels of performance, and accommodate increasing user loads and data throughput. This calls for a highly scalable and extensible EAI model that addresses the risks and issues related to traditional enterprise application integration.
In this white paper, we will discuss traditional EAI and compare relevant service-oriented architecture (SOA) solutions, highlighting their salient features. A sample solution that utilizes different aspects of SOA to effectively minimize EAI risks is offered.

Traditional EAI

Enterprise application integration has formally existed since the early 2000s. However, the core problem it tries to address has existed even longer.

In the process of computerizing, or automating, business processes, organizations invested a considerable sum of money, and developed numerous applications that addressed requirements in each business unit or functional process. However, the need for interoperability among incongruent applications soon emerged as a serious business and technical challenge. Porting data from one application to another was compulsory in order to automate the business process flow. EAI provided the loose coupling of applications by standardizing the input/output (I/O) of devices and improving the overall usability and functionality of applications. Different vendors provided tools to bridge this gap (e.g., WebMethods, Microsoft’s BizTalk, etc.). These included middleware — a layer of systems that allows applications to integrate and interact. Middleware assumed the communications role by permitting the applications to interoperate, and mapping corresponding data formats and protocols. Nonetheless, there were challenges.

Drawbacks of Traditional EAI

Traditional EAI lacked key features that many organizations needed, including:

- Support for data and network security.
- Loose coupling and minimization of interwoven implementation.
- A standard definition of information input/outputs.
- Agility to embrace newer technologies and frameworks.
- Ability to contend with complex integration challenges.

Service-Oriented Architecture

The concept of SOA, which originated in the mid-1990s, has recently evolved to eliminate many of the aforementioned EAI drawbacks. However, it is important to remember that SOA is the definition of principles and methodology, rather than a technology in and of itself. Components are developed individually to expose a service (i.e., reusable code that executes a functional task and can do a meaningful amount of work by itself without depending on other technical components). SOA facilitates services residing on the network to be published, discovered and invoked by each other. The main difference of integration here is interlinked standards rather than interwoven implementation. Each service provides and adheres to a standard of operations, and supports heterogeneous systems through the adoption of pre-defined communication protocols.

The “service” in SOA does not mean “Web services,” although Web service (using J2EE or .NET) is the commonly used technology to implement SOA. Other SOA implementation options use messaging systems (JINI, COBRA, etc.) to enable service discovery and service execution. However, application integration via Web services has become the implementation norm for many organizations.

Head to Head: SOA vs. EAI

SOA and EAI both propose architectural solutions to integrate enterprise systems — ranging from legacy to present-day technology. However, there are some key differences (as shown in Figure 1).

We will now take a look at three prominent models of integration.

Point-to-Point Architecture

Point-to-Point, or P2P, is among the key approaches for enabling application interoperability. This approach, as the name suggests, directly connects the different services of a system (see Figure 2).

Among the highs and lows of this approach are:

- **Fast and efficient:** With no intermediaries, P2P is one of the fastest ways for data to flow through integrated services.
- **Simple:** Each service knows when and how to call the other service, which keeps the architecture simple.
- **Large number of connections to be supported:** For a network of n nodes, there will be \( n^2 \) connections (e.g., 10 nodes = 45 connections). That’s a lot of maintenance.
- **High degree of dependency:** If the standard of interaction changes for one service, all connecting services must change.
Point-to-point architecture is best suited when the number of services that are integrated are not as focused on maintaining the communication protocol. It is typically used in intra-business areas where the control of the network and its components lies within a team. P2P is compatible with situations where data latency is not tolerated.

**Hub and Spoke Architecture**

Maintenance overhead of \( n^2(n-1)/2 \) links stands as the weakest aspect of P2P architecture. The hub and spoke model overcomes this by assigning a service that interacts with other services and acts as a hub. Any service (called a spoke) that needs to be integrated into the system must be integrated with the hub alone. The hub acts as a message broker, or service broker, and communicates with other spokes (see Figure 3).

The salient points of the hub and spoke architecture are:

- **Centralized data management:** The hub manages the spokes and retrieves data from them.
- **Decentralized processing:** The individual spokes are responsible for processing the data.
- **Reduced maintenance:** For a system of \( n \) nodes, there are only \( n \) connections to maintain.

### Comparing SOA and EAI

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Solution level</strong></td>
<td>EAI integrates through system output; integration occurs at the system level. SOA enables a wider range of enterprise applications to be integrated.</td>
</tr>
<tr>
<td><strong>Driving force</strong></td>
<td>EAI is technology-driven, and concentrates on the technical aspects of the integration process. SOA is business-driven, and takes a standards-based approach to business process integration.</td>
</tr>
<tr>
<td><strong>Approach</strong></td>
<td>EAI takes a bottom-up approach – propagating changes from one system to a cluster of systems. SOA takes a top-down approach. Standards are defined for various integrations and taken to each integration point.</td>
</tr>
<tr>
<td><strong>Magnitude of changes</strong></td>
<td>EAI works with a system of changes to integrate via a common network. SOA provides components to minimize changes to application code.</td>
</tr>
<tr>
<td><strong>Vendor dependency</strong></td>
<td>EAI standardization depends on the vendor, and hence increases vendor dependency. SOA works on standards, and neutralizes (to some extent) vendor dependency.</td>
</tr>
</tbody>
</table>

*Source: Cognizant*

Figure 1

Point-to-point services

Hub and Spoke services

**Source:** cognizant 20-20 insights
The hub and spoke model best suits both intra- and inter-business areas where data accuracy is not a primary concern. Domain data, or data that is not transactional in nature, can be shared using this network, since the data does not change frequently. The hub and spoke model best suits both intra- and inter-business areas where data accuracy is not a primary concern. Also, small organizations that do not have a full-scale IT infrastructure can utilize the hub and spoke model to take advantage of its lower cost of maintenance and IT overhead.

Enterprise Service Bus/Bus Architecture

In the hub and spoke architecture, the addition of a new service or changes to an existing service involves changes to the hub, as well as to the spoke. Also, the hub’s integration points are separated from the integration server and message backbone. A recent approach, known as bus architecture, evolved from the hub and spoke model. With the enterprise service bus, the bus acts as the transport platform between services. It provides an integration server and adapters for interlinking services. Adapters for various types of integrations are provided (e.g., adapters for database integration, adapters for file operations).

Bus architecture highlights include:

- **Data standard transformation**: Any component can be designed with an adapter and be integrated with an existing bus; data formats are automatically translated to the agreed standards.
- **High scalability**: The classification of integration endpoints into the integration server, adapters and message backbone scales better compared with the hub and spoke.
- **Implementation encapsulation**: The intermediate adapters encapsulate the details of the implementation, such as the framework, technology, etc., of the service provider from the service consumers.
- **Security mechanism**: Each adapter can have its own security framework for authorization and authentication.
- **Protocol bridging**: The sync providers support protocol conversion (e.g., JMS to HTTP), thereby helping to integrate existing systems without changes.
- **Complexity**: With more intermediary components, the complexity of implementation and maintenance also increases. The technical support needed to maintain systems ranges widely.
- **Performance impact**: With more intermediary components, communication performance can also degrade.

The enterprise bus model fits best in an organization where applications are based on various technology platforms and have different standards of communication. Generally, this...
model is best suited for large organizations that have the resources to implement and maintain the entire bus and necessary plug-in adapters (see figure 5 for a full comparison).

Creating An Enterprise Solution

Given the emergence of various models, as well as their benefits and drawbacks, we will now examine where and how each approach fits into an organization’s EAI roadmap. An enterprise application demands a truly scalable and extensible system that not only focuses on current business processes, but also has the scope to include new processes and systems as they evolve.

Take supply-chain applications, for instance. Different business units (BU) within the organization will have related processes and somewhat distinct goals. For building an enterprise application, the organization should take advantage of existing atomic systems and integrate them. The architecture supporting the integration needs to be decided carefully – weighing both the merits and the shortfalls of the integration model.

Each business unit will also have a greater degree of process control inside its operations. Hence, a P2P integration model should work best here. Fewer services will nullify the disadvantages of tight coupling and help keep maintenance costs low.

Data from each BU must flow to a central hub at a frequency that is indirectly proportional to the tolerance for latency. For example, pricing or contract data may be changed once a year, while Point of Sale (POS) data is received once a month. Shipping data, however, is received on a daily basis. The hub should have the capability to store and publish data to spokes as and when requested. This will help to decouple the business units from each other and reduce direct dependencies from another service provided to another business unit.

An enterprise service bus (ESB), with features that include data transformation, routing and orchestration, is best suited for separating user applications and service providers’ systems.

Model Comparisons: Weighing the Various Options

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Point to Point</th>
<th>Hub and Spoke</th>
<th>Enterprise Service Bus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complexity of Development</td>
<td>Medium (easily coded but common end point details need to be finalized for use by each client).</td>
<td>Medium (hub needs to accept individual protocols and transform).</td>
<td>Low (Open standards ensure less development effort).</td>
</tr>
<tr>
<td>Complexity in Maintenance</td>
<td>High (each connection must be coordinated for future changes).</td>
<td>Medium (hub’s maintenance alone becomes focus for overall availability).</td>
<td>Less (individual service or adapter’s failure does not affect overall function).</td>
</tr>
<tr>
<td>Coupling</td>
<td>Tightly</td>
<td>Need to know the protocol of the hub.</td>
<td>Loose coupling by use of adapters.</td>
</tr>
<tr>
<td>Scalability</td>
<td>High</td>
<td>Restricted by hub infrastructure.</td>
<td>High</td>
</tr>
<tr>
<td>Extensibility</td>
<td>Low (Each node must know the end points’ protocol).</td>
<td>High (new spokes easily added without affecting existing ones).</td>
<td>High</td>
</tr>
<tr>
<td>Security Mechanism</td>
<td>Up to individual service.</td>
<td>Hub can do the implementation.</td>
<td>Built-in mechanism with adapters.</td>
</tr>
<tr>
<td>Data Latency</td>
<td>Nil, real-time.</td>
<td>Present, depends on frequency of updates to hub.</td>
<td>Depends on messaging backbone.</td>
</tr>
<tr>
<td>Performance</td>
<td>High (no overhead).</td>
<td>Depends on the hub’s infrastructure.</td>
<td>Some dip; depends on adapters.</td>
</tr>
</tbody>
</table>

The enterprise bus model fits best in an organization where applications are based on various technology platforms and have different standards of communication.
including P2P and hub and spoke integration models. With security mechanisms built-in, the ESB isolates entire business processes and the application tier from one other — minimizing security risks. ESB facilitates a Web-based application and a handheld PDA device using different protocols of communication to access the same shipping service, for example. Another application might help in placing the order and planning the packaging/shipment of the same – with each depending on a service spread across business units – orchestrating various services through the ESB. The ESB can be configured to act seamlessly according to the source, communication protocol, orchestration requirements and destination.

**Looking Forward**

SOA helps streamline the integration of services across an organization – making them truly extensible and scalable to ever-changing business environments. In a large enterprise, the ability to migrate to a new set of standards for systems and applications will not be feasible for the reasons referenced earlier. Traditional EAI products can help these organizations update and integrate the capabilities of their legacy systems.

But with an efficient adoption of SOA, the benefits surpass the cost and effort of the change in the long run. There is no silver bullet; the appropriate integration model must be chosen – one that keeps data latency, maintainability, necessary hardware/software requirements and scalability factors top of mind.
Footnotes
1 Integrated SOA Governance. http://www.soacom/
3 Good-bye Hub and Spoke, Hello ESB? http://dotnet.sys-con.com

About the Author
Rajesh Pattabiraman is an Architect with Cognizant’s Advanced Solutions Practice, within its Advanced Solutions Group. He has over 10 years of industry experience, and specializes in J2EE architecture and design solutions and SOA integration for enterprise applications. He has played the role of technical lead and architect for various project teams. Rajesh is currently responsible for solution delivery, performance engineering and application modularization of a research application developed by a large and leading U.S.-based legal information services provider. He has a bachelor’s degree in electronics and communication engineering from PSG College of Technology, Anna University, Coimbatore. Rajesh can be reached at Rajesh.Pattabiraman2@cognizant.com.

About Cognizant
Cognizant (NASDAQ: CTSH) is a leading provider of information technology, consulting, and business process outsourcing services, dedicated to helping the world’s leading companies build stronger businesses. Headquartered in Teaneck, New Jersey (U.S.), Cognizant combines a passion for client satisfaction, technology innovation, deep industry and business process expertise, and a global, collaborative workforce that embodies the future of work. With over 50 delivery centers worldwide and approximately 162,700 employees as of March 31, 2013, Cognizant is a member of the NASDAQ-100, the S&P 500, the Forbes Global 2000, and the Fortune 500 and is ranked among the top performing and fastest growing companies in the world. Visit us online at www.cognizant.com or follow us on Twitter: Cognizant.