Shrinking the Enterprise with SOA

A framework to help companies to more effectively create a distributed service-oriented architecture that reduces costs, optimizes infrastructure and facilitates more manageable sharing of services across the enterprise.

Executive Summary

The concept of a service-oriented architecture (SOA) has been around more than 15 years. There are several ways to handle an SOA, but some approaches are complex to maintain and difficult to sustain over time. This white paper addresses the challenges and provides simplified solutions to maintain data that enable the free flow of information across the enterprise.

Visualizing data in any organization is critical in establishing the infrastructure required to simplify SOA support. Organizations typically have numerous data points. The smoothest possible transition is possible if each data point in an organization can communicate in a universal language such as XML or JSON. Pushing them to Web-services architecture ensures that every data point in an organization is accessible and secured.

Data challenges enterprises face include:

- **Redundancy**: Much of the data that flows across an organization is redundant. In fact, applications, hardware, resources, time and other related resources used for maintaining and provisioning are often redundant in ways that are difficult for organizations to detect.
- **Finding the right data**: Locating data is another painful and difficult task in many enterprises. There are many parameters to assess. For example, customer information or product details can be maintained in many places, but finding the right information requires locating the appropriate team and numerous detailed meetings.
- **Accessing data**: To access data across the enterprise, organizations typically face the following challenges:
  - Inconsistent standards.
  - Data trapped in legacy applications with minimal and non-customized interfaces.
  - A lack of separate applications or back-end jobs to maintain the data.
  - Reading the data in the right format requires additional development work.
- **Data security challenges**: In reality, enterprise SOA is much more difficult because services may be deployed across multiple domains of ownership. To make interactions possible, mechanisms have to be present to convey semantics, to declare and enforce policies and contracts and to use constraints for data passed in and out of the services as well as expressions for the behavior models of services. The ability to understand both the structure and semantics of data passing between service end points is essential for all parties involved.
• Monitoring the services developed in SOA: As a result of this challenge, many specialized platforms are evolving to meet these needs.

A Framework, Tools and Dashboard to Visualize SOA

How can organizations address the above problems? Our work with companies across industries over the last dozen years on SOA projects makes clear that organizations need to build a framework with a set of tools that are portable, easily customizable and generic to deploy. Web services offer great promise to reduce complexity and enhance application interoperability. Many organizations typically end up with numerous useful Web services and are often unaware which ones provide what functionality due to a lack of tools and architecture.

Many organizations recognize that their Web services and functionality are either partially or fully built-in with at least more than five applications. As a result, data often becomes redundant and has multiple sources for the same information, despite the fact that many organizations have strong policies to control and secure the data. Moreover, some organizations lack data access standards, and in some cases the code is so closed that it’s difficult to understand the logic that underlies it.

All of the above problems are common. This white paper focuses on remedies for them.

Defining and Designing an Effective Enterprise SOA Solution

The building blocks within our SOA framework used to address the aforementioned challenges are as follows:

- Portal.
- Security bots.
- SLA management.
- Search indexer.
- Dashboard.
- Executive dashboard.
- Agent brokers.
- Data sniffer.
- Data pickers.

Anatomy of an SOA Solution

![Diagram of SOA Solution](image1.png)

Figure 1
A general SOA framework can be deployed at organizations (see Figure 1). To understand how SOA works, please refer to other white papers and articles across the Internet.

To design an effective solution for framework SOA, organizations need to consider numerous factors. One of the critical objectives is to create a cost-effective, portable and generic solution that can be deployed across enterprises.

To keep maintenance costs low, our SOA framework uses a decentralized, agent-based approach compared with a broker-based system where all service requests are handled by a server or set of servers. Custom-developed agents or small code libraries sit on all its application servers to intercept various pieces of information about the services, such as usage, response time and IP address. As these data points flow back and forth, they are recorded and stored locally and transmitted on a regular basis via a log file to the framework database. The agents support the subscription, management and dashboard layers over the top of the existing SOA framework. The subscription layer allows developers to publish and consume Web services and to set up SLAs regarding their service.

A decentralized SOA framework covers four processes:
- Portal.
- Dashboard.
- Brokers.
- Agents.

**Building an Effective SOA**

![Diagram of SOA processes]

**SOA Portal**
The portal's main purpose is to register services (using industry conventions) and validate functionality. The description of the Web services and their taxonomy and details will help other developers find the right components. Every enterprise Web service must be registered so it can be properly monitored (see Figure 2).

**Agents and Brokers**

Once the Web service is registered, certain background processes are necessary for it to be part of an enterprise SOA. Agents and brokers are installed on the server where the Web service is located. Agents and brokers are a set of lightweight components that decentralize SOA layout. Agents are mainly created in J2EE and .Net to conform with the way most Web services are constructed. Five different activities are handled by an agent once installed on the server of the hosted Web service.

Services are instrumented via the SOA framework agent to provide secure access control and monitoring of service usage events. In this approach, the SOA framework infrastructure is highly distributed, with agents colocated within application containers (the Java or .NET application engines on which the application services run within the organization's infrastructure). This arrangement allows the SOA framework to scale horizontally in conjunction with the application infrastructure. Agents are added only when new services are exposed. The centralized broker model does not require agents, and therefore does not intrude on the actual application.
Security Management: The SOA framework uses “username tokens” (based on Web services WS-Security model) to provide authentication and authorization. Depending on how the service is instrumented, these activities are performed at run-time by the agent or by the centralized broker. Agents and the broker have the ability to cache the credentials and authentication information to improve performance. The security service itself is deployed as a framework service that is accessed by agents and the broker.

For services that are used across business units, a two-way secure sockets layer (SSL) at the transport layer is usually added to provide a second layer of security. Services that are exposed to external customers and partners are accessed via public-network-facing gateways. These gateways use username tokens for authentication and authorization; they also require “digital-signature”-based encryption for privacy protection and data integrity.

SOA Framework: Return on Investment

The benefits and returns on investment for a SOA framework can be broadly classified into three categories:

- The cost and “speed to market” benefits associated with reuse of services.
• The cost benefits associated with system consolidation and legacy system “decommissioning.”

• The cost and efficiency benefits associated with the direct exposure of these services to external customers and B2B partners.

The cost and “speed to market” benefits associated with service reuse are obvious as it is usually significantly faster to reuse a service compared to writing it from scratch. For any of these benefits to be realized, there must be a significant level of adoption of services and SOA across the company. The simplicity of the SOA framework’s distributed agent model and the associated portal environment helps considerably with its service adoption.

Looking Forward
A distributed SOA model can be widely adopted within any enterprise that supports service creation and management. While the internal adoption and growth of SOAs has been phenomenal, the platform has also enabled cross-business-unit communication within an organization with B2B partners. As the SOA framework overcomes challenges, it should evolve in three ways:

• **Continued functional additions via framework services:** Expanding the SOA framework services beyond security and logging, and creating new framework services for SLA contract management, service discovery and even the dynamic provisioning of services.

• **Migrating from service standardization to information standardization:** Integrating an SOA framework with an enterprise data services platform to unlock information from legacy data sources and expose them as data services.

• **Moving toward full business automation:** Total integration of service composition and business orchestration tooling into the platform to automate entire business processes – from contract definition to service creation.

References
• Kyle Gabhart, Bibhas Bhattacharya, Service Oriented Architecture Field Guide for Executives.


About the Author
Sameer Kulkarni is a Senior Associate, Technology Consultant and Business Analyst within Cognizant’s Advanced Systems Group. He has 13-plus years of experience in systems analysis, programming, architecture, testing and implementation across multiplatform applications. Sameer has worked as a developer and architect on all the Microsoft technologies since 2000. He received his masters in computer applications from Pune University. Sameer can be reached at Sameer.Kulkarni4@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 75 development and delivery centers worldwide and approximately 178,600 employees as of March 31, 2014, 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.