Systems Engineering at MITRE

SERVICE-ORIENTED ARCHITECTURE (SOA) SERIES

Seven Significant Challenges for Federal Leaders Employing SOA

Larry Pizette, Geoffrey Raines & Steve Foote

Ed Kenney & Rob Mikula

MITRE

Executive Summary

As Federal leadership approaches new and emerging Service-Oriented Architecture (SOA) efforts, several challenges need to be addressed in order to realize the full value of the investment. While the list of topics is potentially quite large, this paper focuses on seven key issues that will need to be understood early in SOA efforts in order to set the foundation for long-term success. The topics are Demonstrating Value, Governance, Acquisitions, Security, Testing, Runtime Management, and Service Reuse.

Demonstrating value—When embarking on emerging SOA efforts, Federal leadership will likely need to justify the expenditure and convey to the stakeholders what value they can anticipate from making the investment. There may be differing perceptions of the value that the SOA will bring, and the Federal leadership will likely need to address these concerns.

Strategies for addressing concerns include:

- Aligning the benefits with the business goals
- Demonstrating value incrementally
- Balancing the competing demands of implementing necessary SOA infrastructure (e.g., security, mediation, middleware) with new capabilities for the business

Governance—Successful governance is essential to establishing trust and realizing the value of SOA efforts. In the Government domain, organizations may not realize that they need to develop a set of rules, responsibilities, and processes to have a successful business relationship between producers and consumers for essential services. Establishing these rules will make the difference between success and failure. Governance should focus on:[1]

- Portfolio management to determine which services should be built, which should be shared, and how the ongoing operation will be funded.
- Configuration management to determine. whether any service can go live on the network
- Operational usage of services to include developing service-level agreements (SLAs) when necessary, as well as monitoring services and escalation policies.
- Testing and version management for establishing solid business relationships between consumers and producers and establishing trust in the quality and availability of services. Full visibility and agreed-upon testing criteria are essential.

Acquisitions—Federal acquisition teams need to consider whether services can be utilized from external organizations or purchased from contractors rather than developing the capabilities themselves. If the services are bought from a contractor, the Government team has the new challenge of trusting critical services hosted by a non-Government entity. Steps that the Government team can take to improve acquisition of services are:

- Determine whether acquiring the capability via a service is a viable alternative.
- Acquire services with a focus on the capability need and SLA, and not on the implementation.
- Utilize performance-based contracting techniques.
- Gain an understanding of contractor incentives and risks.

Security—Information sharing is enabled by protecting and securing the information being shared using an SOA.[2] This security challenge can be successfully conquered by dividing it into three major areas and systematically tackling each one: empowering unanticipated users, establishing trust

across organizational boundaries, and mitigating newly exposed vulnerabilities. If SOA is being used to implement an information sharing strategy that requires access privileges for unanticipated users, Federal leaders and security architects will need to establish enterprise-wide authentication and authorization mechanisms in order to support this type of use. Attribute-based access control and other modern security techniques can be leveraged to provide this capability.

- SOA enables organizations to *technically* communicate and collaborate seamlessly, but it does not ensure that such interoperability will be condoned *socially* and *politically*. Having adequate security, including a common certification and accreditation process, contributes to building trust, which can facilitate social and political acceptance.
- Distributed computing architectures, like SOA, provide increased vulnerability due to new attack vectors. To help mitigate the vulnerability, contemporary Web Service offerings require the use of additional security measures, such as robust input data validation, XML schema validation, and application layer firewalls, in order to reduce the risk of a service-enabled application being exploited by an external attacker.

Testing—Adequate testing is essential to establishing trust between providers and consumers. The required level of testing and certification should be incorporated into the enterprise governance so that service consumers have a level of trust in the quality and resiliency of the service. In practice, service and system testing can be very different in an SOA effort. While a benefit of SOA is that subscribers do not need access to the inner workings of services, this model will likely be a change for test teams that are accustomed to having access to the entire system. Additionally, it may be challenging to test capabilities with a full understanding of the network characteristics and the impact that other programs may have on inter-dependent services. Steps that Federal leadership can take to mitigate the risks are:

- Ensure comprehensive SLAs are in place for dependent services and test, up to and beyond the limits of the SLAs.[3]
- Demonstrate system behavior under high load, such as in the opening days of an armed conflict or national emergency, or under non-standard conditions, such as a degraded or down network or external service failure.
- Institute governance criteria to enforce service testing requirements.[4]
- Consider modifying test processes tailored to an SOA. For example, as Forrester Research suggests, add a level of testing for services.[5]
- Ensure services can scale for unanticipated usage.

Runtime management—Runtime management is essential to the successful adoption of an SOA. In the past with legacy stovepiped systems, runtime management was focused on the status of systems under the control of a single entity. In an SOA, runtime management needs to focus on the delivery of services by providers and the consumption of services by consumers within the boundaries of the metrics stipulated in the SLAs. Runtime management is essential for building trust and development of successful business relationships. Specific steps that Federal leadership can take to maximize the value of an SOA with runtime management are:

- Ensure SLAs are in place with metrics that can be measured and are important to the business, such as volume of data and transactions, timelines of transactions, and availability of data.
- Establish agreements for sharing of metrics and detailed measurements.
- Ensure agreement on how metrics will be calculated and where measurements will be taken. Due to network propagation delays, performance metrics can be legitimately different for consumers and producers.

Service reuse—Service reuse is considered one of the significant benefits of an SOA. Service reuse in an SOA context means reusing the service without rebuilding it or having to operationally host it. (In contrast, in legacy systems, reuse of software generally means incorporating the code as part of the system under development.) Steps that Federal leadership and an enterprise can take to maximize reuse are:

- Manage as a portfolio of services to identify and enforce opportunities for reuse.
  - Ensure a process is in place to investigate the reuse of services, prior to the decision to build new services.
  - Look for opportunities to develop generic, reusable services based on business need.

- Utilize governance to establish trust (e.g., process for changing interface, testing requirements).
- Align funding resources with usage of services so that there is an incentive for Federal leadership to make services available on the network.
- Ensure that there is a portal of information and a registry for the services to facilitate the ease of reuse.

Conclusion

While there are many challenges to obtaining the full benefit of an SOA, Federal leaders who explore these seven issues will be better positioned to identify opportunities and avoid risks. This increased understanding may help their organizations to fully realize the benefit of making an SOA investment.

For more information on SOA, see http://www.mitre.org/soa.

Table of Contents

THE BIG PICTURE: As Federal leadership approaches new and emerging SOA efforts, there are several challenges that they will need to address to realize the full value of the investment. By exploring these issues, they will be better positioned to identify opportunities, avoid risks, and make informed decisions on SOA investments.

Seven Significant Challenges for Federal Leaders Employing SOA

Larry Pizette
Geoffrey Raines
Steve Foote

Ed Kenney
Rob Mikula

Introduction

***Objective*—**The objective of this paper is to provide insight to Federal leadership on seven significant issues that will need to be addressed to realize the full value of SOA techniques. We anticipate that Federal leadership could use this information, along with tailored MITRE support, to gain a more in-depth understanding of these challenges, and as a result, make better, more informed decisions on SOA investments.

For assistance on deciding whether an SOA applies to your requirements, see *MITRE's Perspective on Emerging Industry SOA Best Practices.*

***Intended audience*—**This paper is intended to be used by Federal leadership and MITRE engineers and managers who provide SOA-related guidance to leadership level, Federal Government customers. The information is intended to be used as a starting point for developing more tailored guidance for sponsor-specific challenges. To ensure that this paper continues to accurately capture the emerging best practices of successful SOA implementations, the content will continue to evolve. As MITRE's engineers develop new best practices and the commercial industry makes advances, the authors of this paper will update the content. They welcome feedback and contributions for future editions of this document.

Demonstrating Value

***The challenge*—**Today's Federal leadership teams can find themselves facing tough portfolio investment trade-off decisions when applying contemporary SOA techniques and the various technologies used to implement these architectures. Demonstrating SOA's immediate value to end-user communities is not always straightforward. Many of the benefits of successful SOA implementations occur at a level below the user-facing presentation layer, and consequently, these benefits are not immediately visible to end users. Leadership teams are trying to minimize the cost of enterprise portfolios as well as the resources required for system operations and maintenance. They are trying to put agile architectures in place that can respond more quickly to requirements changes, technology upgrades, and changing missions. Further, when attempting to demonstrate the value of applying limited agency resources to the implementation of business services, they may experience tension between several competing resource demands. Common competing needs and challenges include:

- **Technical team requesting more infrastructure:** IT staff will likely point to the need for additional infrastructure to develop a viable SOA platform for the enterprise. Infrastructure needs may include security, messaging, workflow tools, redundancy and failover, and data access. They may identify specific technologies such as

enterprise services buses (ESBs), business process execution language (BPEL) engines, or data access layers. While these technologies may be needed, they tend to be expensive to implement and by themselves usually do not provide new capabilities for end users.

- **Users demanding more capability:** Program managers and users of Federal Government systems are interested in new capabilities that will help them to achieve their existing objectives more efficiently and achieve new goals. They are not necessarily interested in investing significant amounts of resources and time into infrastructure, which they do not necessarily fully understand or perceive as having value to them.
- **Programs may be unable to identify the value:** While there is significant value to an SOA approach for an entire enterprise, specific programs and service providers may not be able to concretely identify the value to their programs.
- **Funding model may be inconsistent with the value proposition:** If the funding for capabilities is done at the program level, programs may not have the incentive to provide service capabilities to the enterprise. While consumers' costs will decrease and the overall enterprise can reduce costs, the service providers' responsibilities and costs will likely increase, as they provide capabilities to more customers than they did in the past, causing a disincentive. This problem can occur when the overall enterprise and consumers reap the value that producers can deliver, if the producers are not adequately funded for their increased responsibilities. As stated by Oracle, "In the pre-SOA world, budgets were allocated in silos at the project, group, or department level. SOA, on the other hand, is about sharing capabilities as services and leveraging assets across the enterprise. Thus, SOA requires new policies and procedures (including chargeback models) for funding services and architecture." [6]
- **Vendor marketing may be causing confusion:** A significant number of commercial off-the-shelf (COTS) vendors are selling products that provide middleware and other infrastructure that can be used in the implementation of these architectures. These vendors are vying for limited Federal dollars. Some of the vendor information communicated may be helpful. However, some of the information causes confusion, as Ronald Schmelzer from ZapThink identifies: "The SOA spin cycle is now churning at full speed, generating significant froth in the market … Many end users find themselves lost in all this turbulence, bobbing from one vendor's SOA marketing pitch to another, confused between different implementation and architectural approaches to making SOA work, leaving them dizzied, dazed, and confused." [7]
- **Concern whether SOA will be successful or adequately funded:** A study sponsored by BEA Systems, Inc. (acquires by the Oracle Corporation, January 2009), indicated that leaders run into concerns when demonstrating the value of SOA to their organizations. The study indicated that the largest roadblock to justifying an SOA in their organizations was "Lack of confidence in the big SOA payoff" as identified by 54 percent of respondents.[8] Similarly, the study showed "Securing funding" and "ROI not strong enough" were considered roadblocks (47 and 42 percent, respectively).

> *"the connection between business and IT will deliver the promised business value … with flexibility as the driver and SOA as the enabler."*
>
> —Sandy Carter, *The New Language of Business, SOA and Web 2.0.*

SOA's value proposition—As identified in our white paper, *Leveraging Federal IT Investment Using SOA,* SOA builds on past computer engineering approaches to provide an architectural approach for enterprise systems, oriented around offering networked services to consumers. SOA, as implemented through the common Web Services standards, offers Federal senior leadership teams a path forward, given a diverse and complex IT portfolio. It allows for incremental and focused improvement of their IT support systems, where value can be demonstrated while the requisite infrastructure is being built. Leaders can demonstrate the value of SOA through the following:

- **Identify the goals:** In order to demonstrate value, it is useful to identify the objectives that drive the choice of service orientation for your

enterprise architecture. The goals should support the business and operational objectives. Example goals may include organizational agility, reduction of costs through elimination of redundant capability, and better customer service.

Surveys identify that companies are looking to SOA for a variety of business goals beyond just cost savings. Forrester Research identifies that application and business flexibility are more frequently cited as goals than cost savings for both large enterprises and small and medium businesses.[9] This finding is consistent with the BEA-sponsored white paper written by GCR Custom Research. They identify the most common goals from industry leaders with billion-dollar-plus companies in North America and Europe; their finding shows that the combination of improved customer service and faster time to market is the primary goal 44 percent of the time.[10] In this study, cost savings is only identified as a primary goal 30 percent of the time. Similarly, an IBM survey states, "75 percent of the respondents said the primary reason for implementing SOA is to meet new business goals, versus 25 percent that cited fixing existing business problems." [11]

- **Describe the value of the SOA in terms of the goals:** The value of the SOA should be described in terms of the goals, focused on the business and operational objectives. For example, if cost reduction is an objective, an all-encompassing financial analysis would be appropriate. However, if the primary goal of the SOA is to deliver organizational agility, then a quantitative return on investment (ROI) analysis may not be the most meaningful metric. Regardless of the specific business or operational goals, clearly it is important to start with the business and operational objectives rather than the IT objectives.

- **Identify examples of the benefits:** When benefits extend beyond cost savings, such as organizational agility, you should identify examples of new capabilities that could be provided by the SOA and explain how they will positively impact the business and operations. These examples will help non-SOA experts understand the kinds of capabilities that can easily be realized through an SOA. Since increased flexibility and the ability to facilitate organizational agility are perhaps the most important benefits of SOA, it is difficult to provide a satisfying financial analysis for new approaches or technologies with these benefits. Ronald Schmelzer from ZapThink indicates that "while reducing costs and increasing reuse provide clear ROI for SOA, increasing business agility is the most promising benefit of SOA as well as the most difficult to quantify." [12]
- **Demonstrate value iteratively:** Accumulated "best practices" indicate that it is best not to develop a system in a "big-bang" approach; it is helpful to demonstrate value and mitigate risk by pursuing an iterative approach. As significant SOA value flows from improvements to business processes, the iterations should be focused on improving specific business processes and combining new capabilities with only the infrastructure needed to implement the new capability. This approach can be integral to mitigating risk as well. Heather Havenstein from ComputerWorld reports, "Many companies that have achieved early success with SOA focused first on small, incremental projects that show immediate returns to the business." [13] John deVadoss from Microsoft suggests starting small and avoid the "big bang" approach indicating, "The challenge with the big-bang approach is [it] tends to diverge from the business very rapidly. There is a lot of risk that gets built up." [14]

SOA—Demonstrating value—In short, employing SOA to develop Federal IT capabilities can have significant value for the Federal enterprise. By coupling the benefit analysis with the business needs and demonstrating to the stakeholders how the envisioned capabilities can satisfy their needs, Federal leadership can determine if an SOA is the right approach for their organizations. If it is the right choice, an iterative approach to both applications and infrastructure will allow the Federal project leader to demonstrate value incrementally.

Governance

The challenge—Federal Government organizations often find themselves making significant investments in SOA, but without effective governance, the benefit of the SOA may not be fulfilled. SOA introduces the need for enterprise-wide coordination, monitoring, and enforcement where minimal interaction between stakeholders may have occurred in the past. The policy and procedures that guide this interaction are fundamental to the establishment of

"Implementing SOA requires a cultural shift in the way people work together, building closer coordination between IT and business functions, and a sharper focus on delivering value to the whole enterprise rather than simply within a functionality silo."

—BEA Organization and Governance Planning Service

trust and successful usage of services between producers and consumers of services. However, Federal leadership may be unaware of the governance that's required. Since legacy stovepiped systems generally run on dedicated hardware with a known customer base, they rarely require the type of governance that SOA requires.

Organization for the Advancement of Structured Information Standards (OASIS) states that, "SOA Governance should be considered an extension of existing IT Governance that deals with the decision rights, processes and policies that are put into place to encourage the adoption and operation of a SOA that may cross ownership boundaries." [15] Similarly, Bobby Woolf from IBM characterizes governance as the "means of establishing and enforcing how a group agrees to work together." [16] Some of the challenges are:

- **Uncertainty about lifecycle changes in service:** When Federal leaders do not control a critical resource, they may have concerns about changes in the service, such as a shut down at "end of life." Larry Fulton of Forrester Research highlights the importance of service lifecycle management in governance when he writes, "SOA service life-cycle management is a fundamental SOA governance activity." [17]
- **Lack of control of maintenance and upgrade activities:** Maintenance and upgrade activities can result in periods of instability for systems, including services. Generally, programs try to avoid upgrades during critical periods. For example, a Federal military organization would not likely want to make any changes before planned exercises, and a Federal financial organization would not likely want to make any changes at the very end or very beginning of the fiscal year.
- **Unspecified testing:** As systems migrate to SOA, program managers may be concerned that the service they are using has not been adequately tested and that they can't test it themselves.[18] A user of a service can only inspect a service through its interface. The system behind the interface is generally a black box, which cannot be code coverage tested by consumers. Fulton includes testing in SOA service lifecycle management.[19]
- **Uncertain syntax and semantics of data:** When a business process is controlled by one system or one organization, it is relatively easy to understand the syntax and semantics of data. But when utilizing services across a large enterprise, these data challenges become more difficult. As a simple example, currency, speed, location, and time can be represented in a multitude of ways. Data is a key part of governance.
- **Unspecified technologies and standards employed:** When establishing interoperability, consumers and producers may be concerned that they will be trying to collaborate with organizations that employ different technologies. Much of the value of SOA relies on interoperability standards, which are still evolving, and may not be standardized across the organization. These standards can range from choice of technology (e.g., REST vs. SOAP[20]) to the more subtle, such as utilizing different versions of the Web Service Interoperability (WS-I) profiles.[21] Further, many COTS vendors are deploying solutions with some open and some proprietary standards embedded within the products.
- **Information assurance:** When establishing an enterprise-wide SOA, it is important to ensure that systems and data are protected from malicious activity while still providing interoperability and access to legitimate consumers.
- **Operational management:** When relying on external services, consumers will need to have visibility into the status and performance of dependent services. This information is necessary to facilitate trust and must be measurable and understood across the enterprise.
- **Registering services:** In order to share services, the enterprise will need to have a common approach to registering services in a repository. This information is needed for integrating new

services with those already identified in the repository, and for finding the location of the service at run time (i.e., the "endpoint").

As Cutter Consortium notes, "SOA environments introduce a whole new set of governance questions that need to be answered. Who owns the services? How do we select services? Which services do we publish to the external community? How should the services be orchestrated? How do we control the service operation? How do we control the service lifecycle? And so on."[22] These questions and governance challenges are common and cannot be ignored. "One thing is certain: lack of governance can be a serious impediment to success … through 2010, the lack of working governance arrangements will be the most common reason for the failure of SOA projects," states Gartner Group's Paolo Malinverno.[23] Focus on good governance is clearly essential to a successful enterprise SOA.

Steps to SOA governance—There are some important steps to take to establish SOA governance within a Federal enterprise. While each step has a different focus, each also contributes to maximizing the value of the SOA, aligning the SOA investment with business and operational needs, and establishing trust within the enterprise community.

- **Communicate the need for SOA governance to senior leadership:** While most governance guides start with "planning and understanding current governance structures," in the Federal space another step must happen first. The leaders within the enterprise must be made aware that the SOA paradigm requires governance, and they must devote resources and time to establishing governance processes. Without an understanding of this step, subsequent steps will not likely be possible.
- **Establish a governance process:** Once the leaders within the organization understand the importance of governance and its value, it is important to establish a roadmap for implementation. IBM recommends that governance be implemented in four phases: plan, define, enable, and measure.[24] BEA states the importance of SOA governance as a discipline: "Though many companies that are implementing SOA already have some form of IT governance program in place, SOA makes many new demands in terms of the service lifecycle, technology standards, team roles, and resident skill sets. In this context, SOA governance should be approached as a discipline in itself in order to ensure successful SOA implementation."[25] The enterprise-wide governance should be comprehensive, addressing the issues listed above, including policies and processes that facilitate the establishment of trust and successful business relationships.
- **Portfolio management:** It is important to establish portfolio management governance in order to allocate resources efficiently. For example, it is necessary to determine whether an unlimited number of services will be introduced onto the network or whether the number and types of services will be controlled. Considerations for this control include cost, quality, and pedigree of data. For example, if an organization had multiple "update customer address" services, there could be different address information in different services and deterioration in the overall quality of customer data.
- **Operational management—ensure visibility and collect measurements:** For governance, if it is not visible or measurable, it is not likely to have the desired positive effect on the organization. Therefore, it is important to utilize portals or other information-sharing vehicles to convey the status of operational services and those under construction. For operational services, examples of measurable items include "up time" and "response time." However, it is important to be precise about when the response time is measured (e.g., providers' response time will be different from users' perceived response time due to network propagation).

SOA governance—Conclusion—Ultimately, the purpose of governance is to maximize the value of SOA to the organization. Therefore, it must be dynamic and evolve to fit the enterprise needs. Oracle discusses good governance effectively: "Good governance practices are often similar to those followed by successful open source projects: The community itself decides how it will govern itself. Governance models handed down to the team are often self-defeating if the team does not buy into the model. The team needs to see a clear tie between the governance model and improved results. In other words, the governance model should measure things that matter and hold team members accountable for things that will affect the overall team results."[26] Oracle makes a compelling point. Good governance needs to be understood, accepted, and effectively

implemented by the organization. The first step toward achieving this is to evangelize the need for governance with an understanding of its many challenges and benefits.

Acquisitions

The challenge—With SOA becoming an increasingly popular approach for managing large IT portfolios, interdisciplinary teams across the Federal Government are building their first acquisitions for SOA-based services, components, and supporting infrastructure. Many of the governing procedures and rules for these acquisitions were established years before the existence of service orientation as a concept for large software systems; therefore, some parts of the procurement process are not well matched to these acquisitions.

> *"The old practices of procurement to satisfy Government requirements are being squeezed by a strong need for agility, visibility of information and our ever-decreasing ability to pay to 'reinvent the wheel.'"*
>
> *—Jeff Simpson, Government Computer News*

Every procurement team for a Federal SOA acquisition must include what the Government is buying in its strategy. One example of this is the issue of service granularity.[27] As Forrester Research writes, "SOA's focus on business and application services changes the definition of an application, because services are more granular than complete business solutions. This creates a mismatch between the way an application architect may try to solve a business problem and the processes that are entrenched in the Government procurement model."[28] Procurement teams, used to buying large software applications, may struggle with buying collections of enterprise components, in terms of the operational models, technical support models, and equitable cost reimbursement.

Also, Federal procurement teams must determine whether the Government is buying a service that is run on a vendor's computing infrastructure within a vendor's facility and accessed over a communications line, such as a wide area network (WAN), or is asking a vendor to run a service within the Government's own infrastructure. The former situation, often referred to as a managed service provider (MSP), does not completely align well with the Federal Acquisition Regulations (FAR), especially in cases when the service is not a commercial offering and did not exist prior to a Government request for proposal (RFP). In this case, the Government must create the service, making the Government the only or primary user of the service.

Performing a Federal acquisition for a portion of an SOA implementation brings some unique challenges:

- **Service or software product?** Buying or licensing commercial products is a well-understood process within Federal acquisition communities. In contrast, buying network-based services from an MSP does not share the same precedent, especially if the SOA service is provided from a wholly vendor-owned infrastructure. Buying services requires the careful delineation of service behaviors and expected service levels and requires a good deal of advance planning and requirements definition from the purchaser, especially when the needed services are customer-unique. Long procurement times exacerbate the difficulty in predicting service definitions accurately.
- **Whose risk?** Contracting officers have long recognized that increased risk drives more cost. Contract cost is increased not only in the actual realization of risk in operations, but also in the perception of the potential for risk during the bidding process. Consequently, depending on the nature of the work, particular contract types have been established that shift the emphasis of risk between the two contracting parties (Government and vendor). For example, in general terms, time-and-materials contracts are known to be riskier for the Government and may be discouraged by Federal senior leadership, while fixed price contracts are riskier for the vendor. For fixed price contracts, vendors will price risk back into their bids to the Government. Given this context, the procurement team must

define a contracting method that equitably distributes risk between the service consumer and provider, depending on the type of service being acquired. Contractual risk in providing a service should not be placed solely on the vendor, or the costs will be unreasonable.

Of course, risk is not confined to contract type. If the Government asks for a unique service not widely available in the commercial marketplace, and the Government will be the offering's only major consumer, there is substantial risk in creating that new service.

- **Whose capital?** Infrastructure for services essential to the mission of a Federal organization is expensive. If the services run 24/7 and offer highly reliable services, a good deal of capital is required to establish them. In a truly commercial setting, capital investment is recouped by having a diverse set of paying customers who buy a service. In many Government scenarios, the Government is the only customer of a service due to unique legal requirements, unique mission, or unique security requirements. In essence, the capital required to create Government-unique services becomes the Government's investment, whether purchased through a vendor or not.
- **Understanding contractors' risk?** In order to establish an optimal business relationship, it is necessary to structure the acquisition of services so that it is fair and the acquisitions team is cognizant of the risk and the resulting costs. Currently, many Federal IT contracts are written with five-year durations. Of these five years, often only one or two are considered a "base period," with the rest being contract "option" years. Option years do not have to be exercised and can be cancelled at the Government's discretion for reasons that have nothing to do with vendor performance. In practice, these option years mean that the vendor cannot be sure the contract will be in force beyond the base period, and if major capital investment is required for performance of the contract, the only sure way to recoup the cost is to receive this payment in the base period. If the Government is one of many customers, this is not a significant issue. But if the Government is the only service customer, or the dominant customer for a unique service, then risk increases. Uncertainty results in higher costs to the buyer.
- **Where are the prior templates and examples?** One can quickly scan the Government's FedBizOps[29] website and see many examples of successful templates for traditional software acquisitions and support contracts. They can be downloaded by Federal staff and reused as needed. In fact, most agencies and Federal professionals have an archive of prior tried-and-true material for borrowing acquisition text. This text often has been improved by decades of lessons learned and comes with an implied set of language-specific prior case law. Currently, service-oriented acquisitions do not have the same resources.
- **How is security provided?** For some agencies, unique Federal security requirements can make it difficult for vendors to provide the same commercial service offerings to fulfill Government requirements. Changing the commercial offering for one potential customer adds risk for the vendor.

Federal acquisition regulation elements—Most service acquisitions must fit into a combination of three elements: a) existing FAR, b) agency-defined extensions, such as the Defense FAR Supplement (DFARS),[30] and c) agency-defined local contracting practices. Acquisitions can be full and open, meaning that they are open for bidding to Government registered contractors, or they can be existing indefinite delivery/indefinite quantity (IDIQ) contracts that are open to a limited number of contractors. Regardless of the acquisition vehicle and agency-specific formats, a few essential elements must be present to produce competitive bids. Historically, Section C of an RFP contains a description of the statement of work (SOW) and describes requirements in terms of what needs to be accomplished. Section L describes instructions on how the offer is to be constructed, and Section M describes how the offer will be evaluated. Though there have been many agency variations on the titles of these key elements, procurement teams generally find themselves turning their service-oriented requirements into document collections that can be memorialized in a binding contract. Performance-based contracting approaches use variations of these document types to define a contract for a vendor service offering.

Performance-based contracting—Many of the current trends in performance-based contracting work well with the acquisition of SOA services. For example, according to the Office of Management and Budget (OMB), "Performance-based service

contracting (PBSC) emphasizes that all aspects of an acquisition be structured around the purpose of the work to be performed as opposed to the manner in which the work is to be performed ... . It is designed to ensure that contractors are given freedom to determine how to meet the Government's performance objectives, that appropriate performance quality levels are achieved, and that payment is made only for services that meet these levels." [31] This performance-based approach is true to the underlying spirit and architecture of delivering a service in SOA, which focuses on the result of the service, not on specifying an implementation or a description of "how" the service's work is to be done.

Consumers of SOA services care most about the service's interface and its performance characteristics. Similarly, PBSC also focuses on the performance characteristics of the vendor's service to the Government. OMB states, "The key elements of a PBSC Performance Work Statement (PWS) are: a statement of the required services in terms of output; a measurable performance standard for the output; and an acceptable quality level (AQL)."

PBSC and SOA both use the term "service." Given that the term is drawn from two different contexts, a useful parallel can be recognized. For PBSC, the service is a vendor offering being acquired by the Government that provides utility for the Government. In an SOA context, the service is a function or capability, typically run across the network, which provides utility for the consumer. In both cases, the service has a defined behavior or outcome that provides value that can be measured and has defined service performance levels of some type. Consequently, defining SOA services to be acquired in a PBSC contracting framework is easier than the proscriptive SOW of the past because we focus on the interface rather than the implementation. In the past, we might have been tempted to define how exactly the service is to be accomplished, which is contradictory to the component-based concepts of an SOA. Typical SOA services are a black box, providing capability to the consumer.

OMB writes, "Performance-based contracting methods are intended to ensure that required performance quality levels are achieved and that total payment is related to the degree that services performed meet contract standards." [32] The key is that service outcomes are to be measured and expectations are defined. OMB states further, "The definitions of standard performance, maximum positive and negative performance incentives, and the units of measurement should be established in the solicitation." Both of these ideas have a parallel in an SOA service. As an SOA service provider, one carefully defines the offering to the enterprise. Service performance requirements drive the quantity of underlying infrastructure run by the service provider, and that therefore drive the provider's cost. If a contract is crafted to provide an SOA service to the enterprise, the expected service levels will drive the estimated cost of the service and should be considered carefully.

Challenge decomposing application into services—The services required for an SOA are driven by a business analysis of an organization's requirements. While SOA improves the clarity of the items to be acquired, by unambiguously defining their behavior and service levels, getting to this level of detail requires the Government to have a very firm concept of the service to be procured. Given the extended timelines of many procurement efforts, anticipating these service definitions can be a challenge. Forrester Research states that, "SOA changes the definition of an application, breaking it up into a composite of discrete, reusable services. Thinking of an application as a set of services throws a significant monkey wrench into the vision for SOA adoption. Should procurement practices support smaller, discretely defined processing components?" [33] Regardless of the size of the processing components, the service definitions need to be driven by the steps in the business process.

Recommendations—Federal leaders may consider the following recommendations when acquiring SOA services:

- Consider using PBSC frameworks and document templates to acquire SOA services in a Federal context. The acquisition definition should center around the service interface and the SLA. Define how the service should perform, not how it is implemented.
- Perform a rigorous reuse market analysis and determine if the SOA service can be bought commercially or if the service is a one-of-a-kind creation for the Government.
- Independent Government cost estimates (IGCEs) are considered best practice in estimating the total cost of procurement effort. A comprehensive IGCE will consider program risks as a cost driver. If the Government is the predominant,

or only, consumer of the SOA service to be acquired, acquisitions teams should consider the incentives from the vendor's point of view. How much time does the vendor have to make back their investment? The Government group creating the RFP should understand the profit incentives and risk consideration from the vendor perspective in order to acquire successfully for the Government.

- Consider who will own the underlying infrastructure used to provide the SOA service—the Government or the vendor? If the contract should need to be terminated for any number of reasons, including performance or convenience, what will the Government own in the end?
- Consider any special security requirements that may drive your organization away from commercial approaches to providing the service. Diverging from commercial best practices and industry standards will tend to increase costs and risk.

Security

The challenge: Enabling information sharing with security—A primary objective of applying service orientation to a system's architecture is to facilitate broader user access to information stored within that system.[34] This objective gives rise to the challenge: how to enable information sharing while protecting and securing the information being shared.[35] This challenge can be successfully addressed by dividing it into three major areas and systematically tackling each one: empowering unanticipated users, establishing trust across organizational boundaries, and mitigating newly exposed vulnerabilities.

Empowering unanticipated users—In systems without SOAs, all users are known *a priori*. This known information allows the system to control access to resources in a straightforward manner. Authentication, which establishes trust in a user's identity, is performed using locally stored credentials (i.e., usernames and passwords). Authorization, or determining an authenticated user's right to access a resource, is achieved by using access control lists (ACLs) based on user identity or by assigning each user a role with specific access privileges. When this legacy model was being used, Federal leaders could readily trust that the users on their systems were authenticated and approved for the appropriate level of access.

In contrast, if SOA is being used to implement an information sharing strategy that requires access privileges for unanticipated users, Federal leaders and security architects will need to establish enterprise-wide authentication and authorization mechanisms in order to support this type of use. When employed, this approach requires each service to authenticate legitimate but unanticipated users, and authorize them accordingly using a set of access control policies. As this is put in place, Federal leaders and security architects will need to establish mechanisms within their architectures to provide their own information services with enterprise-wide authentication and locally enforced authorization to support access by unanticipated users.

Leveraging enterprise security services—In an SOA, information services should be accessed via a policy enforcement point (PEP) responsible for enforcing security policy decisions.[36] The policy decisions themselves are decided on by a policy decision point (PDP), the entity responsible for access control policy decisions required for allowing or denying access to a resource. When an information service is accessed, the PDP verifies the validity of the user's authentication and that the user has the necessary attributes required for accessing the particular information service according to the applicable access control policies.

To support the concepts of PEPs and PDPs, many organizations provide authentication and authorization security controls as enterprise-wide services that can be leveraged uniformly by all information services. This approach enables security management to be performed either centrally or in a federated manner, as appropriate, enabling uniformity and consistency. It also supports the rapid deployment of new information services without requiring additional (redundant) security code to be implemented for each new information service. And for distributed environments that suffer from periodic loss of connectivity, a PDP can locally cache the validity of users' credentials as well as users' attributes so that policies can still be enforced even when access to enterprise security services becomes temporarily unavailable.

Attribute-based access control (ABAC)—The difficult aspect of implementing the security approach

described above lies in the effort needed to write access control policies. These cannot be based on a user being part of an existing security group, or having an existing security role, because that would require prior knowledge of the user. So, many organizations have chosen to implement "attribute-based access control."

ABAC is a security architecture pattern that is commonly used to support unanticipated users.[37] The concept of ABAC is to provide access to a requester (e.g., a user or another information service) based on the attributes of the requester provided by a trusted source, such as an enterprise attribute server.[38] When a requester attempts to access an information service, the PEP for that service sends the requester's identity to the PDP. The PDP's relevant policies state which attributes are required to allow access. Then, the PDP retrieves the requestor's attributes from the enterprise attribute server and applies them to the policies to reach an allow/deny decision. The PDP then returns its decision to the PEP where the decision is enforced.

By defining and implementing policies based on attributes, instead of user identities or roles, a system can securely control access to information services for a user without prior knowledge of that user. A security administrator can implement a system policy that authorizes access to resources based on characteristics of the requester. For instance, if a policy restricts access to an information service to "no foreign dissemination," then the attribute required to allow access would be "citizenship = 'US.'"

It is recommended that Federal leaders invest the necessary time and effort into determining their security policies and the set of attributes required to support the authentication and authorization needs of their enterprises. Selecting attributes becomes a challenge when those attributes must be consistent with those of external business partners.

Establishing trust across organizational boundaries—One of the core benefits of service orientation is ease of interoperability. SOA enables disparate organizations to technically communicate and collaborate seamlessly, but it does not ensure that such interoperability will be condoned socially/politically. To maximize the value realized by this paradigm, Federal leaders must successfully establish trust relationships with business partners.

Tony Baer from SAIC highlighted this challenge, "While trust was implicit for traditional IT applications, for SOA, it must be made explicit. For instance, when intermediaries are involved, the service provider must depend on the intermediary to vouch for the original requestor. To avoid reinventing the wheel when defining access privileges for each new service, a standard mechanism for communicating trust becomes essential for SOA."

The concept discussed earlier of empowering unanticipated users by using enterprise security services and attribute-based access control provides a firm technical foundation on which trust can be established across organizational boundaries. But each organization also needs to be confident that the other organizations it interoperates with have adequately secured their respective information systems and services. Confidence builds trust, and a common certification and accreditation process builds confidence among organizations.

Achieving certification and accreditation—While SOAs provide characteristics that are beneficial for enterprise systems, they also complicate the existing Federal certification and accreditation (C&A) processes. These complications include increased likelihood of incremental deployment, difficulty in defining system boundaries, interactions with external users and services not under the system's configuration management control, and unanticipated users.

It is important for Federal leaders to realize that SOA is an approach for developing and designing business and mission capabilities. Although there may be aspects of SOA that stress the existing C&A processes, fundamentally current processes still apply. When disparate organizations agree politically to use a common certification and accreditation process (based on consistent security practices such as attribute-based access control), then it becomes possible to establish trust across those organizational boundaries.

Mitigating newly exposed vulnerabilities of applications and services—With all distributed computing architectures, including SOA, as external access to system capabilities becomes available, vulnerability will increase due to open ports and new attack vectors. This vulnerability occurs because adversaries have the ability to interact with the system externally, potentially in such a way as to exploit

software vulnerabilities within internal applications and processes. Even a single exploitation of one of these newly exposed vulnerabilities can undo all prior attempts at establishing trust across organizational boundaries.

Contemporary Web Service offerings utilize well-defined XML interfaces, and while there is increased vulnerability due to open ports,[39] the frequent use of robust input data validation in contemporary SOA implementations provides significant risk mitigation. Additionally, XML schema validation and application layer firewalls also serve to reduce the risk of an internal application being exploited by an external attacker.

Federal leaders need to ensure that sufficient resources (i.e., time and money) are allocated to mitigate the exposure of these new vulnerabilities so that the levels of cross-organizational trust and interoperability they have worked so hard to achieve can be maintained.

Conclusion—It is possible to enable new, unprecedented levels of information sharing by dividing the challenge into more manageable pieces and conquering each one systematically. By enabling access for unanticipated users, establishing trust across organizational boundaries, and mitigating the newly exposed vulnerabilities inherent with distributed computing architectures, an organization can successfully leverage SOAs and achieve improved interoperability.

Testing

> *"Thoroughly testing services is not an option."*
>
> —Serge Lucio, IBM Rational

Why is testing SOA different?—Federal leaders are faced with the challenge of providing high-quality services to their customers while delivering capabilities more quickly and reducing costs across their portfolios. These factors may point leaders toward adopting an SOA; however, they will not fully embrace this approach for their mission-critical applications unless they can trust in the quality of the services being provided and that they will be available when most needed. Since SOA has characteristics that are different than testing legacy systems under a single organizational umbrella, it is important to understand these new challenges:

- **Ownership of dependent services:** The consumer of a service may not be able to test the consumed service beyond the quality of service (QoS) specified in an SLA. While this approach saves the consumer the cost of verifying and validating the software and hardware used to deliver the service, it may put Federal leaders, who are used to rigorously testing all aspects of their system, in an uncomfortable position. This aspect of SOA requires establishing trust in the organization delivering the service as it is desirable to be able to test beyond agreed volumes to understand how the system behaves under severe load (e.g., does it fail gracefully or just stop working) and whether the services can scale to appropriately meet demand. The ability to test beyond anticipated volumes should be a consideration in determining the acquisition strategy and may be a factor in determining whether the service should be provided by a managed service provider or by the Government.
- **Ownership of a dedicated test environment or access to test services:** When an organization tests a system that utilizes external services, it is not likely to have control over a dedicated test environment that encompasses these external services. The situation is further compounded if the service provider does not have a specific service dedicated for consumers to use just for testing. For example, without a service to use for testing, a) real data may need to be used that is sensitive or Government classified, resulting in increased time and cost to protect the data, b) update actions may inadvertently change real data (e.g., names and addresses could be erroneously updated), and c) the consuming organization may not be able to conduct volume, throughput, or error testing against an operational service.
- **Ownership of network and uncertainty of network impacts:** When using an SOA or any large distributed system, the network will introduce uncertainty into the performance of services accessed remotely. This problem may be compounded by unreliable networks that are not owned by service consumer or provider organizations. A challenge when testing SOAs is to understand the usage scenarios and to ensure

that test environment conditions replicate the types of network performance that may be experienced within an operational environment.

- **Information assurance (IA):** SOAs and large distributed systems that are exposed to external organizations—allowing legitimate but unanticipated system use—present different security test challenges than those that are implemented as standalone systems. There is also a natural tension between information assurance mechanisms and a system's performance needs.
- **Unanticipated usage:** Services may be used in ways that were not envisioned by the initial designers, requiring the system to scale to meet the volumes. If the ability to scale is not built into the service and tested, it can cause significant problems when meeting future demand.

Considerations—Because adequate testing is an essential element in establishing trust between business partners and confidence in the capabilities of systems dependent on services, Federal leaders may consider the following suggestions:

- **Understand the behavior of the system in non-standard conditions:** Federal leadership teams have unique concerns when using a network-based service architecture for mission-critical needs. This due diligence into the service provider's capabilities and implementation architecture goes beyond the traditional SOA concepts of a "black box" service, where the implementation details remain hidden to the consumer. When a non-standard event occurs, such as a national emergency or the opening days of an armed conflict, a system may experience network degradation or loss of network connectivity, or the dynamic need for additional processing resources. For mission-critical capabilities in these circumstances, it is important to have gone beyond the black box service consumer point of view and have understood the architecture and behavior of the service implementations, and dependencies on network capabilities and external services. The external services themselves may be under significant extra load. Some suggestions to consider are:
  - **Develop a strategy for a degraded network or service outage:** Federal leaders should consider the implications to the operation or business if provider services were not available or were degraded and establish a strategy for continuing mission-critical operations under these circumstances.
    - Consider the capabilities that are essential for successful operation during an outage; not all capabilities may be needed during a non-standard event.
    - Emulate or simulate degraded network conditions to understand the behavior of the system with reduced or no network connectivity.
    - Perform negative testing to understand the implications of service outages on mission-critical capabilities.
    - Analyze the service implementation architecture for the actual redundancy of service offerings on the network. Multiple independent instances of the service implementation generally support higher overall reliability of the capability.
    - Analyze the modularity of the individual service architectures to determine how additional processing resources can be rapidly assigned in periods of peak demand. For example, a first layer Web Service tier, implemented through contemporary web servers, can be rapidly scaled up either through the replication of a standard server image or by virtualization techniques. This requires that the software implementing the service in the web server tier be constructed for this modularity from the start, and that local load balancing is expected and planned for.
    - Incorporate the strategy for continuing operations of mission-critical capabilities into system test plans. For example, if an alternate source of information will be utilized in the event of a network outage, test the system with the alternate data source.
  - **Establish increased usage options:** Federal leaders should consider writing into SLAs the terms for scaling volume and throughput beyond the standard processing needs, and whether it can be done ad hoc (if burst capacity is needed) or if it requires significant advance notice. If increased usage is available, incorporate it into test plans to determine service behavior and implications for delivering overall system capabilities. Testing should identify whether provider services scale appropriately within the specified QoSs,

gracefully degrade, or break down completely at increased usage levels.
  - **During system operation, monitor the system and network performance:** Federal leaders should consider incorporating instrumentation and monitoring the system and network for early indications of problems. If problems are identified, having a strategy for a fast response and remediation of the issues may prove invaluable.
- **Test the system on the network:** Consider testing the system on a representative network (employing WAN simulators if necessary) or on the actual network before putting the system into operational usage to ensure that the services work in an integrated fashion as expected. This level of testing can be used to ensure that the system will meet user requirements in a geographically distributed environment with network throughput and latency constraints.
- **Independently test services:** Services should be tested independently. David Linthicum writes for ZapThink, "You need to test [services] with a high degree of independence, meaning that the services are both able to properly function by themselves, or as a part of a cohesive system." [40] Developers of systems may not always envision that their services will be used externally with other organizations, but they should consider testing their services with this usage in mind. Carey Schwaber from Forrester Research recommends testing each service in a "services testing" layer and goes on to recommend testing the services layer with an independent test team. "Because there's no way to know where or when they will be used, services should be bulletproofed with an additional round of testing by an independent testing team—whether internal or even outsourced. This organization should test services in isolation, even when the delivery team has already done so. This is not common practice today." [41]
- **Incorporate SLAs into test planning:** Whether testing a system as a consumer or a provider, it is essential to have a firm understanding of SLA requirements in order to adequately write and execute test plans. This knowledge is needed for both functional testing and other types, such as volume testing. For service providers, the test team needs to ensure that the service is capable of delivering the agreed QoS for each customer under load. Therefore, even if SLAs have not been established, the test team needs to have an understanding of planned SLAs and how the service will scale to meet performance requirements.
- **Require testing through governance and provide visibility:** Governance that requires testing can be utilized to ensure the quality of services placed on the network. Schwaber states, "Implement governance mechanisms to enforce service testing requirements. Once service testing standards are defined, fulfillment of these requirements should be required before the service can be checked into a software configuration management (SCM) system or a services registry. These repositories should contain pointers to relevant test assets." [42]
- **Explore modeling and simulation:** Federal leaders can leverage modeling and simulation (M&S) as part of a broader strategy to ensure that they will be able to deliver business capabilities, mitigate risk, and realize the value of their investments. Since thoroughly testing an SOA across a full range of distributed geographic locations, service owners, and network owners may not be possible, it can be advantageous to explore M&S, recognizing that M&S has limitations. Under nominal operating conditions, with predictable network performance, user activity, and service performance, M&S can prove helpful in understanding complexities that may be difficult to test on operational systems. However, as complexities and demands placed on computing infrastructure significantly increase, the extreme conditions that may result can exceed the applicability range of SLAs, so models based only on the encapsulated, abstracted, "black box" view of service behavior that SLAs provide will yield either indeterminate or erroneous results.
- **Test IA:** Given the shared nature of SOA and the IA challenges listed above, extensive IA testing should be included to ensure that access is provided to the desired users while keeping services secure from malicious activity. Additionally, it is important to place a special emphasis on performance testing of the system utilizing security controls representative of those that will be employed operationally in order to properly evaluate the ultimate performance of the system.

Conclusion—Testing SOAs is different from testing previous generation, stovepipe systems due to the inherent dependency on the network and business model of service consumers and providers. There are challenges in gaining the trust of Federal leaders that the services will function according to their SLAs. To address these challenges, new approaches need to be followed, which include governance for minimum test requirements and layered testing. For governance, SLAs should be known prior to testing to determine the load to place on external services. The autonomous nature of service providers and service consumers necessitate multiple levels of testing, from basic unit tests of individual services through functional testing of orchestrated services. There may also be opportunities for third parties to verify service functionality to help establish trust across the enterprise.[43, 44]

Runtime Management

The challenge—In order to establish trust among service consumers and providers, many early adopters of SOA have struggled with the runtime ability needed to verify that the capabilities offered by service providers are operationally meeting the requirements of the enterprise. Runtime management is not new to IT; it is utilized in network management and distributed systems. However, it has taken on increased importance as organizations have migrated critical capabilities to services that may be external to their organizations. In the past, runtime management required understanding whether distributed applications, servers, or nodes on the network were up or down and was dominated by mature products such as HP Software (formerly called HP Openview[45]) or custom solutions. With the migration to SOA, runtime management has new characteristics related to the performance, ownership, and control of services. If an organization does not own a service, it is likely that there will be little visibility into the service beyond the interface contract and quality of service specified by the SLA.

Examples of runtime management challenges are:

- Lack of governance agreement
- Lack of appropriate metrics or visibility into the metrics
- Lack of clear recourse if obligations are not met
- When problems arise, little data may be available to perform post-mortem analysis or conduct any troubleshooting.
- The monitoring points for gathering metrics may be unclear. Stakeholders or vendors may have lack of clarity or a bias in determining monitoring points, including data to report, the analysis, and resulting reports.

Whether the runtime SLAs are being met in a satisfactory manner may rely on the particular point of view of the stakeholder. It may be very common for stakeholders to agree that a service should respond with some output in an expected period of time. Determining the best means to manage services requires a combination of tools to accurately provide required metrics and processes to resolve disputes.

Complexity of distributed SOA environment—Runtime management in the Federal enterprise adds an additional element of complexity, as services may be hosted in multiple locations with different performance characteristics for different requesters. Ronald Schmelzer of ZapThink captured this challenge when he stated that "there might be multiple, distributed implementations of a Service throughout the network, and so managing Service operational performance becomes more like managing a Service 'grid' than individual, discrete Services."[46] This complexity heightens the need for clarity in the definition of performance metrics and SLAs.

SLAs area prerequisite to successful runtime management—Establishing trust between consumers and producers through successful SOA runtime management requires enterprise governance and SLAs, followed by continuous performance monitoring. The SLA defines the terms and responsibilities for both the consumer and the producer, and is a prerequisite to successful runtime management. As a result, the consumer understands his own responsibilities (e.g., maximum volume of transaction requests) and the provider understands her responsibilities (e.g., response times, pedigree of data, "up time"). The SLA should be as precise as possible to avoid misunderstandings. Periodically, SLAs and business relationships should be reexamined to ensure that they are meeting the organizations' objectives and business needs.

Selecting a runtime management solution—In order to meet business and operational needs, SOA runtime management should be viewed based on

the requirements of the enterprise rather than on the metrics that any particular COTS tool offers. Randy Heffner of Forrester Research cautions avoiding a product-based approach. He also adds, "When evaluating SOA and Web Services management solutions, it is important for enterprise architects to consider the embedded SOA management features of their existing application platforms and integration platforms. As a group, standalone SOA management solutions offer the best and most comprehensive capabilities, but some of the embedded solutions come quite close." [47]

SOA runtime management ideally should be able to extend beyond services as components meeting SLAs to incorporate visibility into successful execution of the business. Schmelzer articulates, "From the business perspective, the only aspect of performance that really matters is whether the IT ecosystem is meeting current business requirements." Therefore, it is essential that the runtime management extend beyond whether a service is up or down to incorporate a more comprehensive view of meeting their customer needs.

> *"From the business perspective, the only aspect of performance that really matters is whether the IT ecosystem is meeting current business requirements."*
>
> *—Ronald Schmelzer, ZapThink*

Recommendations—The following recommendations should be considered when establishing runtime service monitoring:

- Perform monitoring and testing at regular intervals and at multiple points throughout the network.
- Utilize measurable metrics that are meaningful to the business. A low number of meaningful, measurable metrics are better than having many metrics that are hard to measure.
- Avoid metrics misunderstandings—all metrics should include where the metrics are captured, how the metrics are captured, and the specific calculation for measuring the service level.
- Plan for changes—structure the agreement to allow for improvement of the service level during the term of the contract.

Service Reuse Aspects of Portfolio Management

The challenge—The ability to place functionality in the form of services on an enterprise network, and reuse the same capabilities among multiple business processes, is a chief benefit of a service-oriented approach. In plain terms, reuse in the service context means not rebuilding a service, but rather the using again, or invoking of, a service built by someone else. It is to the enterprises' advantage to enable service reuse and keep the total portfolio of individual applications as small and maintainable as possible.

However, encouraging reuse across an enterprise brings with it a number of management considerations and challenges, such as building trust among service providers and consumers, being able to correctly define a service's behavior, and ensuring that mission-critical services are adequately resourced and redundant. By managing the collection of services as a portfolio, a leadership team can save considerable resources in IT operations and maintenance. Note that the enterprise as a whole saves resources when services are managed as a portfolio. For example, every time a project decides to reuse services rather than construct redundant services, savings accrue to the enterprise. Since a system's maintenance costs often exceed the cost to build them, over their lifetime, the Federal enterprise saves not only in the establishment of a new service but also in the 20-plus-year maintenance lifecycle of the service. One web vendor stated, "Web Services reuse is everything: on top of the major cost savings … reuse means there are fewer services to maintain and triage. So reuse generates savings—and frequency of use drives value in the organization." [48] In a properly managed portfolio, this savings more than offsets the cost of building and operating the service to be usable in multiple business processes at the outset—a cost that accrues to the project that initially builds and operates the service.

Reuse of a service differs from source code reuse in that the external service is called from across the network and is not compiled into local system libraries or local executables. The provider of the service continues to operate, monitor, and upgrade the service as appropriate. Changes can be made to individual services and fielded in "rolling deployments," as opposed to massive block upgrades as is

often done today. Thanks to the benefits of contemporary Web Service technologies, the external reused service can be in another software language, use a completely different multi-tiered or single-tiered machine architecture, be updated at any time with a logic or patch modification by the service provider, represent 50 lines of Java or 5 million lines of COBOL, or be mostly composed of a legacy system written 20 years ago. In these ways, service reuse is very different from the source code reuse of the past.

Some aspects of reuse remain unchanged. The consumer of the service still needs to trust the reliability and correctness of the producer's service. The consumer must be able to find the service and have adequate documentation accurately describing the behavior and interface of the service. Performance of the service is still key. ZDnet states, "Converging trends and business necessity—above and beyond the SOA 'vision' itself—may help drive, or even force, reuse. SOA is not springing from a vacuum, or even from the minds of starry-eyed idealists. It's becoming a necessary way of doing business, of dispersing technology solutions as cost effectively as possible. And, ultimately, providing businesses new avenues for agility, freeing up processes from rigid systems."[49]

Portfolio approach—Every team leading an SOA implementation has to grapple with its fundamental approach for offering services to the enterprise. Is the organization going to foster a "command economy" or a "market economy"? In an SOA command economy, a leadership chain of command (e.g., enterprise CIO office) determines the services—sometimes through an enterprise architecture—that will be made available to the enterprise and allocates the funding and IT resources for their implementation and operation. While this approach can minimize service duplication in a portfolio, it requires one part of the organization to have an understanding of consumer requirements across the entire organization. In a market economy analogy, services can enter or exit the enterprise based on consumer demand without central planning.

In practice, some mixture of the two approaches is often wise. For example, individual service providers, who have been successful and have a deep understanding of their customers and data sources, should be allowed to continue offering services. In this market approach, service providers can enjoy the success of correctly matching customer requirements, or endure the consequences of forecasting incorrectly. The enterprise CIO must also ensure from a command point of view that the enterprise has a reasonable IT portfolio, gaps in services capabilities are being filled somewhere in the organization, architectural commonality is being preserved, and mission-critical resources are adequately resourced. Successful SOA efforts will support innovation by the participants, while also ensuring a comprehensive set of reused services and standards compliance. The challenge is finding the balance.

Every enterprise managing projects evolving toward SOA has to grapple with its fundamental approach toward service reuse. Unfortunately, as of the writing of this paper, it appears that little has changed in the basic approach that most Government organizations have taken toward program management and reuse. Programs typically evolve as individual stovepipes, with traditional systems engineering defining interfaces, across which there is occasional reuse. Decision making is driven by the traditional cost, schedule, and risk management criteria, which leads to "stovepipe SOAs"—services developed within a program, visible only within the program, operating on program-developed infrastructures. This is the natural result, based on the incentives in today's Government program management approaches.

The alternative approach is to manage groups of projects as portfolios of services. This approach begins with a portfolio manager developing an enterprise architecture (EA), driven by the business processes within the portfolio. The portfolio manager can use the business process to drive a decomposition of the processes into services; the portfolio manager then uses this service layer architecture to determine gaps and overlaps among competing programs developing services to deliver capability. In cases where services have established track records, the portfolio manager can be very directive in establishing reuse in the enterprise architecture, and then using this EA to resource the program developing and operating the service to respond to the demand across the portfolio. In other cases, a portfolio manager may decide to create an explicit competition, resourcing more than one alternative to manage risk across the enterprise, eventually providing sustaining funding to successful services and "starving out"

unsuccessful services. Finally, in a complex system, emerging services may appear within the portfolio that were not anticipated; portfolio managers should also identify these successes and resource them appropriately as well.

Reuse costs—Barry Boehm provided two useful formulas when estimating the costs of software systems reuse. One formula is from the provider's point of view, while the other is from the consumer's perspective.[50]

Provider focused formula:

$$\text{Relative Cost of Writing for Reuse (RCWR)} = \frac{\text{Cost of Developing Reusable Asset}}{\text{Cost of Developing Single-Use Asset}}$$

Consumer formula:

$$\text{Relative Cost of Reuse (RCR)} = \frac{\text{Cost to Reuse Asset}}{\text{Cost to Develop Asset from Scratch}}$$

Jeffery Paulin examined large-scale SOA service providers to estimate the value ranges for these formulas in practice.[51] His data shows that RCWR ranges between 1.15 and 2.0 with a median of 1.2, while RCR ranges between .15 and .80 with a median of .50. In other words, Paulin's work suggests that creating a generic reusable software component for a broad audience takes more resources (15 percent to 100 percent more) than creating a less generic point solution. The cost of reuse therefore shifts to the providers, and benefits the consumers. Consumers spend less (median 50 percent less) to reuse the service than to create their own. We can see from these formulas that as the enterprise decides to fund service providers, there is great benefit in maximizing the number of consumers for an operational service.

Reuse measurement—Federal leadership should measure reuse as part of a periodic portfolio management assessment. Actional writes, "Reuse is not only a key benefit of SOA, but also something that you can quantify. You can measure how many times a service is being used and how many processes it is supporting, thus the number of items being reused. This enables you to measure the value of the service. With a little work, you can calculate the service cost savings for each instance of reuse, including saved architecture and design time, saved development time, and saved testing time."[52] The assessment of reuse can be effectively integrated into the information repository used for service discovery in the organization, the enterprise catalog.

"Reuse is not only a key benefit of SOA, but also something that you can quantify. You can measure how many times a service is being used and … this enables you to measure the value of the service."

—Actional

Recommendations—
The following recommendations should be considered when enabling enterprise reuse of services:

- **Manage services as a portfolio:** Portfolio management should be used to manage the collection of projects developing services. The development of an EA that permits the assessment of capability gaps and overlaps is critical to the decision making of the portfolio manager. Portfolio managers should leverage all the tools at their disposal to both resource, direct, and encourage reuse.
- **Appropriate governance:** Some service reuse will hinge on consumer trust. It must be more advantageous for the consumer to call someone else's service rather than to build their own. To engender trust, the enterprise should govern the SOA implementation with a focus on building and communicating attributes such as reliability, availability, and accuracy.
- **Measurement:** Measurement of reuse in the enterprise should be performed. Federal organizations should assess the degree of reuse they are achieving over time. An EA can provide the structure for this examination. Application requirements that are perceived to be unique and that thwart reuse should be examined with a critical eye.
- **Marketplace:** Create a marketplace for the free-form exchange of information about a service between service providers and consumers. In

practice, this marketplace can be accomplished by creating a portal where both providers and consumers can interact. Issues of concern by a service consumer can then be effectively communicated to a provider, and resolutions can be communicated to the entire community.

References

[1] Adopted from Forrester Research, January 24, 2008, Defining SOA Service Life-Cycle Management by Larry Fulton
http://www.forrester.com/Research/PDF/0,5110,43723,00.pdf

[2] W3C Web Services Architecture
http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/

[3] SLA in this context is defined as a requirements agreement and not necessarily a legally binding contract.

[4] Schwaber, C., R. Heffner, M. Daniels, July 17, 2006, SOA Raises The Stakes For Software Quality, How Software Testing

[5] ibid

[6] http://www.oracle.com/technologies/soa/docs/oracle-soa-governance-best-practices.pdf

[7] Schmelzer, R. July 4, 2007, SOA Infrastructure Patterns and the Intermediary Approach.
http://www.zapthink.com/report.html?id=ZAPFLASH-200774.

[8] http://www.bea.com/content/news_events/white_papers/BEA_Costs_and_Benefits_GCR_Survey_Final.pdf

[9] Heffner, R., February 28, 2007, Planned SOA Usage Grows Faster Than Actual SOA Usage, Business Data Services, North America, Europe, and Asia Pacific.

[10] http://www.bea.com/content/news_events/white_papers/BEA_Costs_and_Benefits_GCR_Survey_Final.pdf

[11] http://www-03.ibm.com/press/us/en/pressrelease/21844.wss

[12] http://zapthink.com/report.html?id=ZAPFLASH-20050127

[13] http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=267564

[14] http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=267564

[15] http://wiki.oasis-open.org/soa-rm/TheArchitecture/Governance

[16] Woolf, B., ISSW WebSphere J2EE Consultant, 13 Jun 2006, Updated July 2007, Introduction to SOA Governance, IBM.
http://www-128.ibm.com/developerworks/ibm/library/ar-servgov/ Bobby Woolf

[17] Fulton, L., R. Heffner, C. Salzinger, K. Smillie, Defining SOA Service Life-Cycle Management Understanding a Core SOA Governance Process, Forrester Research.
http://www.forrester.com/Research/Document/0,7211,43723,00.html

[18] http://wiki.oasis-open.org/soa-rm/TheArchitecture/Governance

[19] Fulton, L., R. Heffner, C. Salzinger, K. Smillie, Defining SOA Service Life-Cycle Management Understanding a Core SOA Governance Process, Forrester Research.
http://www.forrester.com/Research/Document/0,7211,43723,00.html

[20] _Note that SOAP is no longer considered an acronym. For more information, see
http://www.w3.org/TR/2003/REC-soap12-part0-20030624/#L1153

[21] http://www.ws-i.org/

[22] http://www.cutter.com/content/itjournal/fulltext/2007/06/itj0706b.html

[23] Afshar, M., May 2007, SOA Governance: Framework and Best Practices, Oracle White Paper.

[24] William A. Brown, Senior IT Architect, IBM Enterprise Architecture & Technology CoE, SOA CoE, Garry Moore, Consulting IT Architect IBM Global Services, William Tegan, Associate Partner, Application Innovation, IBM Global Services, SOA governance—IBM's approach, IBM Corporation.
ftp://ftp.software.ibm.com/software/soa/pdf/SOA_Gov_Process_Overview.pdf

[25] Services Data Sheet BEA Organization and Governance Planning Service™ Accelerating SOA Adoption in the Enterprise,
http://www.bea.com/content/news_events/white_papers/BEA_SOA_Org_Gov_Planning_Service_ds.pdf

[26] http://www.oracle.com/technologies/soa/docs/oracle-soa-governance-best-practices.pdf

[27] Granularity in this context means the layer in the business process and not the calling architectural layer, frequency of use, or amount of data transferred.

[28] Gene Leganza, Forrester Research, "US Federal Enterprise Architects Are Committed To SOA, But Procurement Gets Complicated," April 12, 2006

[29] http://www.fedbizopps.gov/

[30] http://www.acq.osd.mil/dpap/dars/dfarspgi/current/index.html

[31] http://www.whitehouse.gov/omb/procurement/pbsa/guide_pbsc.html#chapter1

[32] http://www.whitehouse.gov/omb/procurement/0703pbsat.pdf

[33] Gene Leganza, Forrester Research, "US Federal Enterprise Architects Are Committed To SOA, But Procurement Gets Complicated," April 12, 2006

[34] OASIS Reference Model for Service Oriented Architecture
http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf

[55] W3C Web Services Architecture
http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/

[36] ISO10181, The Access Control portion of the ISO Security Framework, International Standards Organization

[37] April 2004, Security Design Patterns, The Open Group.

[38] November 2005, Security Patterns within a Service-Oriented Architecture, IBM Corporation.

[39] The Open Web Application Security Project
www.owasp.org/index.php/OEWASP_Top_Ten_Project

[40] ZapThink Design and Validate SOA in a Heterogeneous Environment, The Foundations of SOA July 2008 Analyst: David Linthicum, http://www.zapthink.com/report.html?id=WP-0171

[41] Schwaber, C., R. Heffner, M. Daniels, July 17, 2006, SOA Raises The Stakes For Software Quality, How Software Testing Methods Must Change To Suit Service Orientation, Forrester Research.
http://www.forrester.com/Research/Document/0,7211,39947,00.html

[42] ibid

[43] Lithicum, D., J. Murphy, J., Key Strategies for SOA Testing

[44] A Guide to Successful SLA Development and Management, Gartner Strategic Analysis Report

[45] http://www.hp.com/education/sections/network.html

[46] Schmelzer, R., April 16, 2008, What does Service Performance Mean? Document ID: ZAPFLASH-2008416 | Document Type: ZapFlash.

[47] Heffner, R., October 22, 2007, Embedded SOA Management Solutions, A Supplement To The Forrester Wave™: Standalone SOA And Web Services Management Solutions, Q4 2007

[48] http://www.actional.com/resources/whitepapers/SOA-Worst-Practices-Vol-I/Web-Services-Reuse.html

[49] http://blogs.zdnet.com/service-oriented/?p=699

[50] Boehm, B.
http://sunset.usc.edu/GSAW/gsaw99/pdf-presentations/breakout-2/boehm.pdf

[51] Jeffery Poulin, J., 2006, The ROI of SOA Relative to Traditional Component Reuse, Logic Library.

[52] http://www.actional.com/resources/whitepapers/SOA-Worst-Practices-Vol-I/Web-Services-Reuse.html

www.mitre.org

Approved for Public Release
Distribution Unlimited
Case Number: 09-0172
Document Number: MTR090010

MITRE