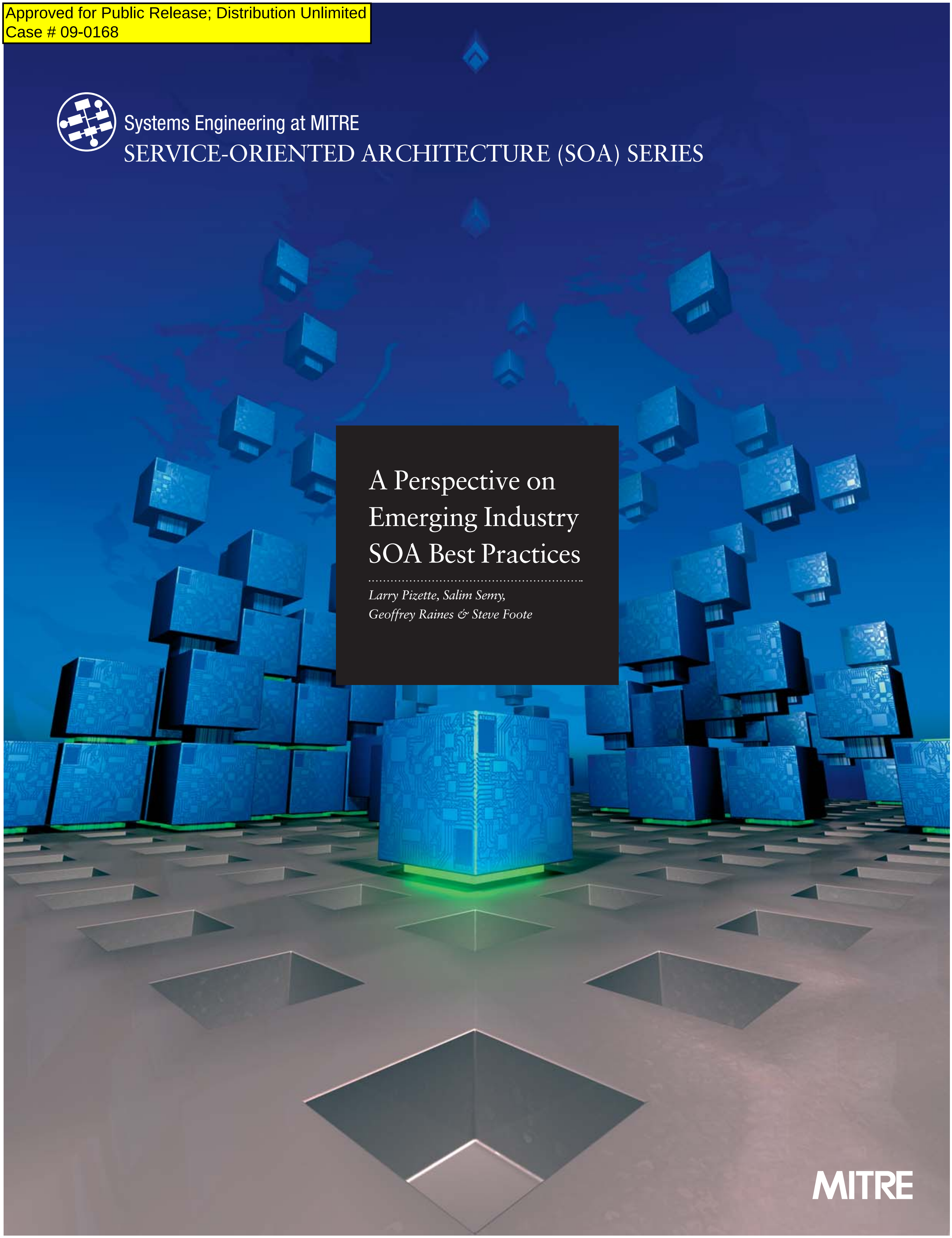
Systems Engineering at MITRE
SERVICE-ORIENTED ARCHITECTURE (SOA) SERIES



A Perspective on Emerging Industry SOA Best Practices

Larry Pizette, Salim Semy,
Geoffrey Raines & Steve Foote

MITRE

Executive Summary

Service-Oriented Architecture (SOA) practices are intended to create an agile, integrated information technology (IT) infrastructure that is scalable, reliable, and can rapidly respond to an organization's changing needs by employing loosely coupled and dynamic services. An SOA approach enables business needs to drive an organization's strategic IT decisions. As a result, SOA allows a business to become more efficient in meeting its current business needs and more agile in meeting future (and possibly unknown) business needs. It is important to note, however, that an SOA is neither a panacea nor something that can be purchased. There are challenges with employing SOA techniques effectively, some of which are explored in this paper.

Given the anticipated benefit of delivering business and operational value (e.g., cost savings, improved business processes, increased accessibility to information), SOA has become a high-priority focus area for the Federal Government. As the various organizations in the Government research adopting SOA, they often struggle with fundamental questions:

- **Why should we adopt an SOA approach for our IT portfolio?** Is an SOA the best approach for our organization?
- **What are the inhibitors to an SOA approach?** What causes the failure of SOA initiatives and how do we avoid these pitfalls?
- **How do we know if we are implementing an SOA correctly?** The SOA approach is broad and can be implemented with many tools, standards, and commercial products. What are the engineering-related tradeoffs of the various approaches?
- **What results should we expect from implementing an SOA?** A clear definition of success is necessary to set the expectations of the SOA initiative, relate them to the organization's business goals, and use them in determining the strategic benefit of the investment. What set of metrics are necessary to quantify the results of an SOA initiative?

As the Federal Government starts to evolve their architectures to a service orientation on a large scale, project leaders within the Government will look for the lessons learned by the IT industry and Government organizations regarding governance and technology. The practical experiences that clearly demonstrate the benefits of an SOA approach will continue to emerge.

This paper documents a variety of best practices and key characteristics of successful SOA implementations based on the authors' analysis of SOA case studies and industry reports. Ten emerging best practices for successful SOA implementations from industry are included below.

Determine if an SOA is the best approach—It is important to realize that an SOA is not always applicable for business and technical reasons. Furthermore, even if an SOA approach is adopted as an architectural pattern, particular standards are better suited for certain situations than others. Some aspects of an enterprise SOA deployed on the network using contemporary Web Service standards are not well suited for certain types of systems (e.g., real-time components). Some of these constraints are unique to contemporary SOA (e.g., TCP/IP transport underlying Web Services) and some are inherent to all distributed systems on the network (e.g., security and network latency). While it is possible to engineer around many of these challenges, the further the architecture moves away from commercial and contemporary SOA standards, the greater the cost. In addition, the program will be less likely to realize the benefits of SOA.

Start SOA activities with the focus of solving business and operational challenges—When used properly, an SOA enables business goals to drive IT decisions. SOA efforts that are purely focused on meeting policy requirements or implementing the underlying technology without a focus on business processes may fail. This focus on business processes can lead to the consolidation of services and the infrastructure through reuse, increased business agility gained from implementing business processes that are composed of services, and easier integration of new and legacy systems to compose business processes.

Employ services to support key business process steps—Most Government organizations typically build their own organization-specific capabilities to meet their needs, which often leads to redundancy, increased costs, and interoperability issues. By focusing on business process steps, which are reused across the enterprise, the reuse of services becomes a natural outcome of the architecture. These step-based business process services enable an organization's needs to be reliably met by another organization's capabilities. Key factors in developing services include identifying the right services to support business process steps, loose coupling, making services visible to consumers and developers, and establishing contracts between service providers and the services' consumers (i.e., the users).

Examine your data, realizing that SOA does not solve data problems and it may expose them[1]—
The flexibility of SOA in decoupling applications from data may expose issues with data quality, ensuring data availability, and the semantic differences of data. In the end, shared services share data; unless providers and consumers agree on the data that constitutes the payload of a service, shared services will not be possible. When deploying an SOA, it is best to keep a focus on data, paying special attention to the governance for maintaining data quality, maximizing data availability by tracking data management issues associated with service-level agreements (SLAs), defining a common data or abstraction layer, and focusing on mappings between internal schemas and a common vocabulary across the community. Commercial-off-the-shelf (COTS) tools may facilitate these efforts.

Start small, learn, and evolve—Employing the "big bang" approach to SOA adoption is unlikely to be successful, due to the lack of knowledge within an organization about how to do it right the first time, and the varying levels of maturity and growth patterns in different parts of the organization. While the nature of Federal Government acquisitions pushes us toward large projects, the focus should be on small, incremental releases. SOA initiatives should begin by addressing a real business problem, focusing on piloting the architecture, ensuring clearly defined success criteria exist, and capturing the lessons learned to educate the enterprise and improve future SOA implementations. Narrowing the initial scope of an SOA implementation to one or two business processes will help keep it to a manageable and realistic size. This strategy will reduce the time it takes an organization to realize value from its SOA investment.

Have a long-term vision—It is natural to focus on employing an SOA approach to satisfy a particular business need for a certain set of customers, but it is impossible to fully anticipate future users or their needs. An SOA implementation that is scalable and capable of expanding in scope and requirements will ensure it meets future and unanticipated needs. Considerations include monitoring the services, ensuring scalability of the infrastructure and services, developing an appropriate testing strategy, building security that is scalable at the enterprise level, and establishing a governance framework.

Ensure governance is a key component of the SOA—SOA techniques can be applied to individual projects, but the changes necessary for an enterprise-wide adoption can only be achieved by putting the right policies and processes in place to bridge the enterprise architecture (EA) with the business strategy. Governance is an essential element of an SOA; it creates, communicates, and enforces policies, defines roles and responsibilities, and aligns IT investments with business goals. Furthermore, lessons learned can help evolve the governance strategy. Some examples of key performance indicators to measure the effectiveness of governance include the number of applications using shared services, compliance with key aspects of a reference architecture, and adoption of enterprise standards. Solid governance is essential to establishing trust.

Integrate security throughout the SOA lifecycle—
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 (if SOA will be used to implement an information sharing strategy that requires access privileges for unanticipated users), establishing trust across organizational boundaries, and mitigating newly exposed vulnerabilities. Federal leaders and security architects may need to establish enterprise-wide authentication and authorization mechanisms to support access by unanticipated users. Attribute-based access control and other modern security techniques can be leveraged to provide this capability.

The successful implementation of an SOA requires that the right security mechanisms are applied to the right services. Also, security should be balanced with other considerations, such as performance and scalability.

Set your expectations on the return on investment of implementing an SOA—One motivation for moving to an SOA is the promise of cost reduction. While cost savings can be a realistic expectation, an organization should expect upfront costs when using SOA techniques for the first time. These upfront costs can be due to the learning curve associated with implementing a new approach, the lack of technology skills and familiarity, the lack of mature industry standards, and limited user enterprise management abilities. Cost savings may occur at the enterprise-level eventually, but not necessarily at the project level. Furthermore, cost savings is not always the primary objective. Another motivation for moving to an SOA is the ability to rapidly deploy capabilities. For example, in the DoD, the flexibility derived from SOA implementations can accelerate the deployment of urgently needed capabilities to warfighters, resulting in more effective mission executions and possibly saving lives.

Examine an SOA approach as part of a networked enterprise—The ability to leverage IT resources across the network to adapt to evolving requirements and to rapidly deliver new functionalities to meet users' needs is at the core of a networked enterprise. SOA practices can help realize this vision by establishing shared services and service composability; however, to support these goals, the Federal Government must put in place the appropriate governance (i.e., establish and enforce policies on how services are developed, made available, secured, operated, and used by the enterprise). Furthermore, the Federal Government should focus on establishing trust, in addition to the financial and organizational aspects of governance within a networked enterprise.

The SOA best practices described above are intended to serve as a baseline for successful SOA implementations. They illustrate that an SOA is more than a system or software architecture; SOA changes the character and agility of the underlying IT infrastructure that is available to an organization's senior leadership team and decision makers. While technology is a key part of employing SOA techniques, other IT management issues (e.g., changing the organization's culture toward providing and consuming services and implementing effective governance processes to continually align the IT portfolio with business requirements) are equally important.

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

Table of Contents

THE BIG PICTURE: SOA best practices are intended to create an agile, integrated information technology infrastructure that's scalable, reliable, and can rapidly respond to an organization's changing needs by employing a portfolio of loosely coupled and dynamic services. SOA, however, is not a panacea. As with any large-scale systems integration effort, there are challenges with employing SOA techniques effectively.

A Perspective on Emerging Industry SOA Best Practices

Larry Pizette
Salim Semy
Geoffrey Raines
Steve Foote

Introduction

SOA practices are intended to create an agile, integrated information technology (IT) infrastructure that is scalable, reliable, and can rapidly respond to an organization's changing needs by employing loosely coupled and dynamic services. An SOA approach enables business needs to drive an organization's strategic IT decisions. As a result, SOA allows a business to become more efficient in meeting its current business needs and more agile in meeting future (and possibly unknown) business needs. It is important to note, however, that an SOA is neither a panacea nor something that can be purchased. There are challenges with employing SOA techniques effectively, some of which are explored in this paper.

Given the anticipated benefit of delivering business and operational value (e.g., cost savings, improved business processes, increased accessibility to information, etc.), SOA has become a high-priority focus area for the Federal Government. As various Government organizations consider adopting an SOA approach, they are likely to struggle with a few fundamental questions.

- **Why should we adopt an SOA approach for our IT portfolio?** Is an SOA the best approach for our organization?
- **What are the inhibitors to an SOA approach?** What causes the failure of SOA initiatives and how do we avoid these pitfalls?
- **How do we know if we are implementing an SOA correctly?** The SOA approach is broad and can be implemented with many tools, standards, and commercial products. What are the engineering-related tradeoffs of the various approaches?
- **What results should we expect from implementing an SOA?** A clear definition of success is necessary to set the expectations of the SOA initiative, relate them to the organization's business goals, and use them in determining the strategic benefit of the investment. What set of metrics are necessary to quantify the results of an SOA initiative?

As the Federal Government starts to evolve their architectures to a service orientation on a large scale, project leaders within the Government will look for the lessons learned by the IT industry and Government organizations regarding governance and technology. The practical experiences that clearly demonstrate the benefits of an SOA approach continue to emerge, so the established best practices are also evolving.

Objective—This paper documents a variety of best practices and the key characteristics of successful SOA implementations based on the authors' analysis of SOA case studies and industry reports. It documents a set of emerging best practices for organizations considering adopting an SOA-based approach. It also provides insight to the key issues and potential pitfalls that need to be addressed as an organization considers implementing an SOA.

This paper presents ten emerging best practices that the authors believe are necessary to successfully implement an SOA based on collective experiences gathered from the commercial sector. Throughout the paper, the authors emphasize that an SOA is more than a system or software architecture; an SOA changes the character and agility of the underlying IT infrastructure that is available to an organization's senior leadership team and decision makers. Thus, while technology is a key part of implementing an SOA, other IT management issues are equally important, such as changing the organization's culture toward providing and consuming services and implementing effective governance processes to continually align the IT portfolio with business requirements.

The paper discusses many aspects of SOA, identifies the major considerations for each best practice, and provides references to substantiate the conclusions.

Intended audience—This paper is intended to be used by MITRE's engineers and Federal Government leadership. This knowledge is intended to be leveraged, providing a starting point for developing more tailored guidance for domain-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 as industry makes advances, the authors plan to update the content of this paper. The authors welcome your feedback and contributions for future editions of this paper.

Determine If SOA Is the Best Approach

While SOA can provide the benefits of reuse, agility, and loose coupling, these benefits are not always the software architect's first priorities. For example, when designing real-time systems, as performance requirements become constrained to seconds or milliseconds (e.g., flight safety systems, communication systems), software application design decisions should be based on the required system performance. In general, this leads to tighter coupling between system components to ensure the predictability and reliability of all aspects of the system. Often real-time system implementations are largely custom-designed to suit particular performance requirements, and they are not good candidates for general enterprise service reuse. Furthermore, the benefits of agile composition tend not to apply in real-time systems. Changes to these systems are tightly controlled to meet key mission requirements.

Distributed architectures—Under constrained networks, a distributed system may not be the best solution. Many of the issues that are frequently associated with contemporary SOA technologies are inherent to distributed systems and are not specific to SOA (e.g., security constraints with open ports, timing delays with network communication, etc.). Distributed architectures, including distributed SOA implementations, work best when the underlying network is robust, reliable, and available. The network should be sufficient in facilitating communication between the various components of the system. There are several characteristics that define the quality of the network, including bandwidth, reliability, and connectivity; deficiencies in any characteristic can make it very difficult to invoke services across the network. Any constraints placed on the network limit the effectiveness of the distributed system.

Web Service implementation—The benefits of industry standards and supporting tools make Web Services an appealing approach to implement an SOA. Contemporary technologies and standards for Web Services include SOAP,[3] a common extensible markup language (XML) messaging protocol; Web Services Description Language (WSDL), a standard XML-based description of service interfaces; and Universal Description, Discovery, and Integration (UDDI), a universal directory for discovery and invocation of services. While Web Services are prominent in many SOA implementations, they are not ideal under all circumstances due to the overhead of SOAP messages and the need for open ports.

Start SOA Activities with the Focus of Solving Business and Operational Challenges

The SOA approach enables business needs to drive IT decisions. SOA provides organizations with the ability to organize and use distributed capabilities (i.e., services) that may be controlled by different stakeholders,[4] enabling them to become more:

- Effective in meeting current business needs
- Capable of meeting future (or currently unknown) business needs

As reported by ZapThink, organizations that adopted SOA experienced the time- and cost-savings benefits from the consolidation of services (i.e., shared services), improved flexibility in rapidly responding to new business needs (i.e., business agility), and the streamlined integration across systems (i.e., simplified integration).[5] To reap these benefits, an SOA implementation should focus on addressing business problems rather than focusing entirely on technology or Web Services.

> *SOA efforts that do not focus on solving business problems are often at risk of failing.*

SOA efforts that do not focus on solving business problems are often at risk of failing. For example, if SOA techniques are employed purely to meet policy requirements, it often leads to a minimal effort implementation, or just enough to "check the box," and it can deviate from the business focus. For this reason, if policy is the primary driver of deploying an SOA, the organization should rethink their motivations.

A successful SOA deployment is about more than technology. Projects often declare SOA implementations successful after they provide Web Service interfaces for their existing systems. Wrapping systems with Web Service interfaces alone does not result in an SOA implementation, and it does not provide major benefits to the overall business. While common practice, it is not necessary to use Web Service technology to implement an SOA. Aberdeen Group's research points out that only about 50 percent of SOA initiatives at best-in-class companies are Web Service-based.[6]

Organizations will see the benefits of shared services, business agility, and simplified integration if their SOA adoption strategies are motivated by, and aligned to, resolving business problems.

***Shared services*—**SOA enables organizations to benefit from reuse by consolidating existing services or by updating existing services to deliver new functionalities. Fundamentally, it is the reuse of business process steps that drives the reuse of services. This reduces the overhead cost of maintaining redundant services, and it promotes efficiency (i.e., time and cost savings) in developing a new business process. This is not an instantaneous process; it involves maturation over a number of stages, from improvements in implementing local best practices to shared service economies of scale. IBM's Shared Service Maturity Matrix[7] allows organizations to characterize their shared services environment. IBM's analysis of the private sector indicates that the full benefit of shared services is only achieved in the final stage, but incremental progress through the intermediate stages can derive partial benefits.

One example of service consolidation is Verizon.[8] Looking to reduce inefficiencies in software development, Verizon focused on the 250 most important business transactions that the company performed, such as determining customers' credit histories and looking up customer information. On average, each transaction had been deployed five to 25 times. This duplication decreased the productivity of developers and created unnecessary ongoing maintenance costs. Verizon developed and institutionalized shared, reusable Web Services internally and externally with telemarketing partners to exchange customer data. Within a year of becoming operational, this consolidation effort helped Verizon cut their IT budget in half by eliminating the redundant systems they had inherited from the merger of Bell Atlantic and GTE.

Another example of service consolidation is Guardian Life Insurance. [9] Faced with the problem of multiple application silos with little attention to business goals or reuse, they employed an SOA to make the disparate technologies across the silos work together, focusing their IT efforts on developing new applications rather than reworking old ones, and leveraging their investments in legacy systems. The SOA implementation was aimed at modernizing the administration of three systems: benefits, claims processing, and policyholder administration. Within 28 months of their SOA implementation, Guardian Life Insurance developed about 60 services, 50 of which were used by all three systems, resulting in a 30 percent savings in their application development budget.

Two additional case studies that demonstrate the value of an SOA approach that is based on shared services include Countrywide Financial[10] and Standard Life Group.[11]

***Business agility*—**SOAs are designed to allow existing IT assets to be leveraged more effectively and to provide agility in developing new business processes through common interface standards and composition of services. The term "service orchestration" refers to composing reusable and newly developed services to support business processes. Implementing business processes as a set of composed services allows the organization to continuously utilize IT assets for the most critical business drivers, which increases the responsiveness of their IT in meeting new business needs. This leads to developing more business capabilities quickly at a lower cost, and the ability to absorb and integrate new business partners.

ING Card employed an SOA when they built an application for their customers, enabling it to link to new websites, implement new product features, and maintain credit scoring rules easily and quickly.[12] The agility they achieved from the SOA approach allowed them to provide these capabilities to customers in multiple countries. This often involved customizing processes, products, and credit scoring for each country. ING Card achieved this agility with a layered architecture by employing a service concept and by parameterizing some of the business software.

British Telecom is another case study that demonstrates the value of an SOA based on business agility.[13]

***Simplified integration*—**SOA enables integration by abstracting the service implementation behind the interface that is used to invoke the service. Contemporary SOA implementations frequently use standards-based interface components, removing the need for developers to understand the underlying technologies used by the individual services. This standardization makes IT more flexible; standard interfaces enable services to be composed as business processes that can be reused. The increased flexibility and simplicity enables companies to be more efficient in their integration efforts for both modernized and legacy applications.

Merrill Lynch developed a Service-Oriented Legacy Architecture (SOLA) to reuse their multi-billion dollar investment in legacy systems in their newer distributed systems.[14] Since the majority of Merrill Lynch's business runs on mainframe systems, it was imperative that they incorporate these systems as part of their SOA initiative. Their SOLA helped them take a holistic approach to incorporating legacy systems into their SOA environment. Merrill Lynch's SOLA, which exposed 420 customer information and control system applications as Web Services, resulted in a ten-fold improvement in performance time and the number of transactions that could be processed. Processing about two million Web Service transactions daily, Merrill Lynch estimated a savings of $500,000 to $2 million per application through cost avoidance and direct savings.

Other case studies that demonstrate the value of an SOA based on a simplified integration objective include Transamerica Life Insurance,[15] Sony Pictures Entertainment,[16] and eBay.[17]

Employ Services to Support Key Business Process Steps

Commercial and Government organizations typically create IT capabilities that meet their own specific needs and requirements. For large organizations, this can result in unintended inefficiencies, including the following:

- **Redundancy:** Multiple projects re-creating the same or similar functionality.
- **Increased operational costs:** Each redundant capability may be managed and maintained separately, which leads to increased long-term sustainment costs.
- **Interoperability issues:** Integration across the enterprise becomes more challenging when implementations use a variety of standards and commercial products.

For the Federal Government, the Office of Management and Budget (OMB) spent considerable time working on the issues listed above and put processes in place to reduce redundant IT portfolios. For example, many Enterprise Architecture (EA) requirements for Exhibit 300 described in the OMB document "Planning, Budgeting, Acquisition, and Management of Capital Assets," have these goals in mind. The purpose of Exhibit 300 is to coordinate OMB's efforts to collect agency information for Congressional reports, and to ensure that business cases for investments are tied to the mission statements, long-term goals, and objectives. In essence, it is used to ensure that a strong business case is provided prior to all IT investments.[18]

Contemporary SOA approaches facilitate a reduction in an organization's IT portfolio redundancy by providing key capabilities that are commonly used across business processes as services (e.g., validating a credit card service), making them readily available to users across the enterprise. By focusing on business process steps that are used across the enterprise, the reuse of services becomes a natural outcome of the architecture. Once a capability is offered for reuse, an organization's needs may be met by the capabilities that are offered by another organization.

Identifying the right services—It is imperative that services be specified at the right level of the business process (i.e., the breadth of functionality provided by the service) so they can easily be mapped to business process steps. As ZapThink points out, organizations must size their services correctly to maximize reuse and minimize unnecessary expenses.[19] This design effort entails focusing on business requirements, identifying business problems of the right size (i.e., a subset of the business requirements that still provide value to the business), and appropriately scoping both service and infrastructure development. Articulating a business problem includes clearly documenting the challenges, identifying the potential opportunities and risks associated with the problem, and determining the costs associated with solving the problem. To demonstrate the business value of employing SOA techniques early, business analysts should focus on a subset of the business problem that delivers optimal business value with limited risk and can be addressed within a short duration (e.g., six- to nine-month timeframe). Services should also be appropriately specified so they can be reused and orchestrated with other services.

Abstraction and loose coupling—The engineering concept of abstraction is at the core of the services approach. The detailed steps to fulfill the service are only known by the service provider; the software, databases, languages, and central processing units running the service are hidden. The service consumers only see the service interface, which enables them to ask for, or to invoke, each service.

Abstraction can occur in several ways. For existing legacy systems, developers can build service wrappers for existing code, which hides the legacy system behind a contemporary Web Services interface. Similarly for new services, the developers can create software services using contemporary Web Service standards. In either case, abstraction makes service consumers blind to the implementation technology used by service providers. It promotes loose coupling between service providers and service consumers, enabling organizations to absorb and integrate new business partners and customers more easily than traditional point-to-point integration. Abstraction introduces simplicity for the consumer and provides the service provider with the flexibility to implement the service in a manner that suits their environment.

There may be particular circumstances, however, where tight coupling is a valid approach. Phil Wainewright explains that "tight coupling is still very important when your process is very stable and you want maximum operational performance … loose coupling comes into play when you want flexibility."[20] Even when tight coupling is required, the engineer can make every attempt to localize and isolate tight coupling so as to not unnecessarily constrain overall system interactions.

Making services visible—As highlighted by Luc Clement, "if users cannot easily find business services, the promise of SOA is largely lost. If developers cannot readily find and reuse services, they essentially don't exist."[21] A key aspect of emphasizing visibility is making service descriptions available to support visibility, accessibility, and understandability. Furthermore, as Clement explains, a platform-neutral, standards-based registry that publishes business services is an essential part of service visibility. Publishing service descriptions and policies to a registry serves as a system of record for an SOA, and it supports management and oversight of the services.

It is unlikely that real-time discovery of services unknown to the consumer will have a significant impact in the Federal space in the near term, given issues of trust, security, and reliability. During the development phase, there is sufficient time to investigate the service offering, establish a trusted relationship, agree to a service-level agreement (SLA), and perform testing as needed. During execution of an operational work flow, there usually isn't sufficient opportunity to perform these activities in real time.

Establishing contracts between service providers and service consumers—The autonomous nature of service providers and service consumers necessitates establishing contracts to manage expectations. These contracts, often referred to as SLAs, are defined by Gartner as "the contracted measures the service recipient will use to accomplish its key business objectives. This agreement sets service provider and recipient expectations, describes the products or services to be delivered, identifies contacts for end-user problems, and specifies the metrics by which the effectiveness of service activities, functions, and processes will be measured, examined, changed, and

Section	Description
Executive summary	This is a summary section describing the general purpose of the document to meet or exceed the service-level measurements that are mutually agreed on. This should include the purpose of the document and the duration of the agreement. It should define the stakeholders or ownership for the service-levels agreed on within the enterprise and the scope of the areas that are included.
Description of the services	Within this section is a detailed description of each of the services and the committed performance levels associated with them.
Service-level management	Numerous processes need to be documented regarding the management of service-levels, including measurement tracking and reporting, business continuity, problem escalation guidelines, service/charge requests, new services implementation, approval process, and the service-level review process
Service-level definitions	For each functional area, a minimum number of key SLAs should be included. A sample of the description of the data points that should be prepared for each SLA are: Definition: The key business service (function/process/procedure) that is being measured, reported, and continuously improved. Measurement time frame: The days, dates, and times when the defined SLA is to be measured, usually indicating the inclusion or exclusion of recognized national holidays. Assumptions/responsibilities: Statement of specific requirements that must be met by the IS organization and business units to remain in compliance with the SLA. Service-level metric: Relevant measurement of required work performed by the IS organization. Although these service-levels are commonly measured in percentage terms, IS organizations need to design pertinent measurements that can be expressed in terms of business performance. Measurement formula: Description of mathematical formula and example. Reporting measurement interval period: Reporting period for measurement that determines exceeding, meeting, or not meeting target SLAs. Data sources: Location(s) from where data is collected, including a description of what is collected, where it is collected, how it is stored, and who is responsible for it. Escalation activity: Describes who is notified and under what conditions as out-of-compliance situations occur, including day-to-day and measurement period out-of-compliance situations. Escalation management: Identifies to whom the out-of-compliance activities are forwarded on recognition. Contractual/exceptions/penalties/rewards: Describes, and refers to, any contractual exceptions, penalties, and rewards that are included in the contract. Reward/penalty formula: Description of mathematical formula and example. If the enterprise employs severity or priority codes, generally they would be described within this section.
Appendixes	Appendixes are used to include additional information that might be relevant to the agreement, such as the hardware and software supported.

Figure 1. SLA Template[22]

controlled.”[23] SLAs are essential to delivering services. Realistic SLAs, adherence to agreements, and incentives are necessary to ensure that consumers’ needs are satisfied. A sample SLA template, provided by Gartner, is shown in Figure 1.

Examine Your Data, Realizing That SOA Does Not Solve Data Problems and May Expose Them[24]

David Linthicum explains that since data is the most important component of SOA, it is important to think about how it is managed.[25] Decoupling applications from data is one way SOA enables flexibility in IT to better support business processes. This allows data services to be created and makes data more widely accessible by other applications and services within and outside an organization. Implementing an SOA by itself does not solve data problems. While SOA makes data more accessible, it might still be unusable due to data quality, availability, and differences in community vocabularies. For example, community agreement on vocabulary can facilitate the movement away from point-to-point interfaces toward “publish once/subscribe everywhere” interfaces. Insufficient attention to data may compromise the value achieved from an SOA implementation.

Data semantics—Semantic interoperability is essential to a successful SOA approach because it enables service providers and consumers to exchange information.[26] Without semantic agreement, data is exchanged without consistent meaning, which leads to misinterpretation and data corruption, causing the data to become unusable. Two approaches that address this problem are defining a common schema for widespread use across an organization's services, and allowing providers and consumers of services to individually reconcile semantic differences. The common schema approach can be challenging to implement because it requires a unified agreement across a large organization. The second approach results in tightly coupling data representation with services, which jeopardizes the agility provided by service orchestration. Addressing data semantic interoperability issues effectively is likely to require an approach somewhere between these two solutions.

Microsoft Corporation's Pat Helland distinguishes between data inside and outside the service boundary,[27] explaining that the data outside a service that is sent between services as messages must be defined in a way that is understandable by both the sending service and the receiving service. This should be the focus of semantic integration. Defining data loose couplers (i.e., the minimal set of agreements on data to facilitate translation across vocabularies) will help with semantic integration of data outside the service boundary.

The inconsistencies between the producer and consumer vocabularies could be addressed by developing a common vocabulary. This shared vocabulary could be used to help expose and interchange data between providers and consumers. While it is impossible to identify all potential consumers of a service, or to agree on a single vocabulary shared by all users, a common vocabulary would allow new consumers to understand data semantics and representation and align their vocabulary accordingly. As an example, the DoD has promoted the establishment of Communities of Interest (COIs) as part of their Net Centric Data Strategy to develop a common vocabulary within a COI.[28]

Data quality—Many Federal agencies employing an SOA to bridge multiple incompatible systems find that data quality issues can seriously undermine the value of an SOA.[29] From a data provider perspective, exposing data via services may reveal internal data issues, including redundant, inconsistent, or stale data. While traditionally hidden and accounted for by tight data and application coupling, data quality issues may be exposed through the loose coupling of data and applications associated with SOA. Delivering a high-quality SOA requires high-quality data. Without quality data, SOA is just an efficient way to share poor-quality data.

> An SOA data strategy should include establishing a centralized enterprise data management function to treat data as an enterprise asset.

As the *Federal Times* points out, data governance is critical to maintaining quality data.[30] Data governance includes procedures, policies, standards, and technologies to manage the availability, quality, consistency, and security of data. Furthermore, an SOA data strategy should include establishing a centralized enterprise data management function to treat data as an enterprise asset,[31] and focus on establishing the roles and responsibilities of particular data management.

Data availability—One inherent characteristic of an SOA is the ability to provide distributed services over a network that supports a heterogeneous, and perhaps unpredictable, set of service consumers. The unpredictable nature of consumers and the network over which the services operate make effective data management and scalability an important aspect of SOA.

To help address data problems and to ensure a manageable scope, it is important to prioritize which data is exposed and to understand how it provides business value. Furthermore, it is important to ensure that underlying capabilities and processes exist to support data availability (e.g., scalable data replication and management infrastructure) and quality (e.g., data governance) because managing data accessibility for countless consumers can become overwhelming.

As David Linthicum pointed out, SOA projects should start with data because those that start with services must often redesign and redefine services after they figure out their data layer, which can significantly change the initial services. [32]

Start Small, Learn, and Evolve

Inappropriate scoping is often a primary cause of failure for an SOA project (i.e., employing a "big bang" approach).[33] Since SOA is an aspect of enterprise architecture, architects often seek to make their initial SOA adoption an enterprise-wide effort, even when it is not justified by a business case. Such an approach, with a complete infrastructure overhaul and replacement of all existing systems at once, is unlikely to result in a successful SOA. It is also unlikely that an organization will have all of the knowledge up front on how to adopt an SOA within their organization or to do it right the first time. Furthermore, as organizations grow and become more distributed, as in the case of Countywide Financial,[34] each business unit is faced with different growth patterns and technology lifecycles. Implementing a successful company-wide SOA across these heterogeneous environments at once is not possible. SOA adoption is an iterative process that takes time and requires education across the enterprise.

With SOA, it is best to start small, learn, and then evolve. An incremental approach allows lessons learned to be collected before adopting an SOA in a large way; it minimizes business risks and allows returns to be realized incrementally. However, when scoping SOA efforts, it is important to address a meaningful business problem, focus on piloting the architecture, clearly state the desired outcome and measures of success, and capture lessons learned, which can be leveraged in future efforts.

Address a real business problem—While it is important to start small, it is also important to start an SOA project by addressing a problem that, when resolved, will make a difference. A well-scoped SOA project is usually a sign that the organization is focusing on a meaningful business problem, resolving it in a service-oriented way, and learning lessons about best practices and potential pitfalls to inform future SOA efforts. Focusing on problems that address a high-value business need or pain points will help the SOA effort be successful and expand across the enterprise.

To identify a business problem that can be resolved by implementing an SOA, it is important to understand and articulate the organization's business goals. This approach helps focus the SOA effort on key business processes, which can subsequently be analyzed and decomposed to develop specifications for services. This process underscores the importance of collaboration between IT and business. For example, to support business-based SOA, IBM developed a business-driven development approach, which is a role-based business process for developing SOA solutions.[35] It stipulates the collection of requirements and the capture of business processes written by business analysts. The resulting process models and artifacts are then used by architects to design systems, by developers to develop services, by integration developers to implement service flow and deployment, and by the quality assurance team to use in their testing efforts.

Pilot the architecture—ZapThink identified one common SOA pilot pitfall as the desire to build new functionality that resolves a particular business problem, which focuses heavily on the new functionality or the development of standards-based interfaces rather than the architecture.[36] The SOA approach is about developing an architecture that provides a comprehensive, high-level plan for how business interacts with IT. Specifically, SOA should help a company build IT that can respond to unpredictable changes in the needs of the business or users. Implementing an SOA is largely about figuring out how to identify the right services at the right level of granularity[37] so they map to business process steps, determining which services can be reused, and assuring loose coupling through contracted interfaces and policies.

Identifying the right services often takes considerable time and is likely to be the most difficult and important step in the successful deployment of an SOA. Shanxi Mobile used IBM's Service-Oriented Modeling and Architecture model,[38] employing both a top-down process to think about the changes to business processes that must take place in concert with bottom-up thinking about infrastructure.[39] One business application took two months to identify the flows and services, five months for the specification and related flows, and one month for the realization. Refining the process is important to a meaningful SOA deployment.

In practice, an SOA pilot will focus on more than architecture. SOA should be validated with a working implementation, which involves developing new functionalities and upgrades for the infrastructure. Shanxi Mobile incorporated an Enterprise Service Bus (ESB) into their infrastructure to allow them to make the connection between the top-down business process and bottom-up system component perspectives. They employed the ESB to build adapters that worked with existing systems, which were then available as services to implement the business process.

When determining the technologies to employ in an SOA, an architect should view an ESB as a set of capabilities that fulfills infrastructure needs rather than as a single COTS product or a set of tools that must be purchased together from a single vendor. An ESB solution is not required for an SOA, but it can provide the appropriate prebuilt plumbing to connect services, supporting service management, messaging, workflow, and many other capabilities. This functionality can be provided by a single vendor or by using multiple independent products. In the former approach, an integrated, comprehensive suite of infrastructure products from a single vendor may fit the needs of the organization. However, the latter approach may help maintain vendor neutrality (i.e., allowing an easier transition from one vendor solution to another), reduce costs, and provide flexibility in the implementation (i.e., not being tied to a specific vendor for the overall implementation). The SOA architect should try to avoid becoming overly tied to a single vendor's ESB solution.

Prior to implementing an SOA, it is important to realize that efforts and resources will likely need to be shared across the development of business and infrastructure capabilities. This requires that resources be adequately prioritized to ensure a balanced approach across business activities and infrastructure upgrades. Some best practices to ensure a balanced and reasonably scoped effort include the following:

- **Leverage existing systems:** Rather than discarding legacy systems, focus on bringing new technologies and capabilities to the development effort, determine what can be leveraged from existing systems, and wrap them to expose capabilities as part of the SOA implementation. Verify that these activities are justified by a business case.
- **Employ only the necessary standards and technologies:** Focusing on a particular business problem means focusing the implementation on particular technologies and specifications that support developing a solution for that business problem. It is not necessary to employ every standard and technology at once, especially during the initial pilot.

One key to a well-scoped SOA effort is identifying the overlap between business objectives and existing legacy systems to identify opportunities to leverage legacy systems to accomplish business goals.

Define success criteria—Defining the criteria of success is crucial to understanding when an SOA achieves success and to quantify the SOA deployment's return on investment. ZapThink identifies three commonly used criteria to measure success:[40]

- **A complete architectural design:** Within the scope of the pilot, a design that includes details about service contracts, invocation styles, and other elements of a working SOA implementation are necessary. This design may serve as a starting point for follow-on SOA efforts. The Intel Corporation's[41] design consisted of a proto-integration framework and methodology and a set of SOA development practices to coach the teams who were developing the services and to integrate COTS products into this framework. Intel spent about a year developing the baseline services for very basic SOA functions and the integration capability.
- **Reusable business services:** Measure the level of reusability of services developed during the SOA's initial deployment. Intel focused their SOA implementation on reuse, measuring the success of their SOA on the cost savings achieved by reuse (i.e., the value of reuse in dollars). Intel quantified the value of the reuse of services by measuring the number of reusable services and the number of consumers using the services. The cost savings was in excess of tens of millions of dollars.
- **Governance criteria:** An initial SOA deployment should have a governance approach that handles assigning roles and responsibilities, requires policy creation and management, and evolves with the SOA implementation. The pilot should provide this initial governance framework, which can then be leveraged and evolved in future efforts. For example, Intel developed an internal services portfolio planning process that provided business processes for identifying services and prioritizing service development and deployment. The process included intercepting business solution projects to identify common services across implementations and to define a service delivery

timeline. Also included in the governance framework were best practices for developers to use in designing services to foster service-oriented development and reuse.

Capture lessons learned and educate the enterprise—An important reason for incrementally developing an SOA (i.e., starting with a well-scoped SOA pilot) is to capture the lessons learned and educate the enterprise.[42] There is a great deal of knowledge and many lessons learned that can be gained from starting small, understanding the potential pitfalls and best practices, and learning from them before moving onto a large-scale enterprise adoption. It is important to look for knowledge management practices to capture the lessons learned and share them across the enterprise. By advocating and supporting the collection and dissemination of best practices and patterns, Federal decision makers, architects, and developers can reduce the uncertainty and risk in determining when, where, and how to apply an SOA.

Shanxi Mobile's SOA implementation illustrates the importance of educating the enterprise. From their SOA initiative, the principal lesson learned was that people matter most.[43] Processes start with people, not systems—if system developers do not understand what, why, and how people do things, they will not develop systems that are relevant to the people or the enterprise. Helping people see the big picture and learning to understand the needs of others, including customers, will foster cooperation. Persuading people to cooperate on an emerging SOA initiative may be difficult. Business units should recognize that their silo does not necessarily reflect the total picture and that an enterprise requirement that crosses business functions will offer more options to everyone.

Have a Long-Term Vision

Starting small is important, but in doing so, it is important not to lose sight of the big picture. An initial SOA implementation may focus on satisfying a particular business need or providing a service to a particular set of consumers. However, it is impossible to predict how the services will be used and who will use them in the future. Building an SOA for today can result in tightly coupling services for specific service consumers, resulting in stovepipes with pair-wise connectivity between systems. Furthermore, without planning for future or unpredictable usage, SOA may fail to meet future service demands. Therefore, SOA implementations should be designed with the expectation that requirements will evolve. SOA should be built to allow for scalability and expansion, both in the scope and requirements of the SOA deployment.

> *Starting small is important, but in doing so, it is important not to lose sight of the big picture.*

Manage services through monitoring—Managing services is essential given the unpredictable nature of a service's usage. Monitoring services can provide a measure of how well management goals are being met and aid in pinpointing and resolving problems. Some examples of service goodness measures that can be monitored include:

- **Service performance and availability:** Comparing the service execution time and downtime to SLAs helps identify inconsistencies and indicates when scaling the infrastructure is necessary to meet higher demands.
- **Service access:** Implementing authentication and authorization capabilities to monitor and control access to the service.
- **Service auditing:** Logging a service's activity helps determine usage patterns, intrusions, and tracks the service's functionality.
- **General policy compliance:** To effectively implement the governance of services requires, at a minimum, a semi-automated infrastructure to monitor the services' compliance to the defined policies.

As defined by Forrester, SOA management consists of the "software infrastructure to ensure that the production operation of SOA-based services delivers on Quality of Service (QoS) expectations for technical performance and availability and, optionally, on QoS for security, business operations, and general policy compliance."[44]

Forrester identifies two alternatives to SOA management solutions: standalone and embedded. Standalone solutions are software systems that monitor the SOA's infrastructure and are often available as single products. An alternative is to use existing application platform monitoring capabilities that may be adequate for near-term SOA monitoring (i.e., an embedded management solution). The embedded management approach uses existing component-level monitoring capabilities as part of an overall service management solution. As a best practice, embedded management solutions should be considered as part of the initial SOA implementation, which can then evolve to a standalone solution if necessary.

Choose implementations that are easily scalable—In David Linthicum's blog, "Scalable SOA Solutions Continue to Emerge,"[45] the writer points out that many recently employed SOA technologies and approaches were not tested with higher application, information, and service management traffic loads. Rather, they focused more on getting solutions up and running rather than on performance or scalability. Unfortunately, Linthicum indicates, choosing the wrong technology or approach leads to problems that may be painful to fix.

In SOA implementation design and technology, decisions should not preclude scalability. Considerations should include the following:

- **Infrastructure:** As services are deployed, the infrastructure should support short-term goals and allow for long-term extensibility. For example, infrastructure virtualization technologies should be considered to minimize complexity and retain the focus on agility and flexibility.[46] This technology can be used to create virtual servers and storage devices for application computing and storage, allowing application processes to be executed across a pool of resources. This leads to a separation of the infrastructure and applications, which leads to greater flexibility.
- **Service development:** Providing guidance for service developers will help ensure that scalable services suitable for business process orchestration purposes are built. While immediate goals may be oriented toward a specific business process, the reuse of these services within other processes will require an appropriate level of granularity of services.
- **Testing:** Adopt testing practices to test the performance (e.g., load testing) and to perform functional, interoperability, and vulnerability testing to ensure that the SOA implementation is robust, scalable, integrated in the environment, and secure.[47]

A good example of a scalable SOA implementation, from the infrastructure perspective, is Wachovia's Corporate and Investment Banking division's SOA implementation.[48] Wachovia's goal was to employ an SOA to break down silos of highly customized financial products and services while leveraging existing legacy systems in the horizontal integration process. The potential fluctuations in demand at different times among various divisions provided potential for wasted or insufficient resources that could lead to poor performance. Their solution, a grid computing-based infrastructure, provided the framework to allow for scalability and the effective use of computing resources to reduce costs and improve performance.

Build security from the ground floor—Due to the number of potential environments, domains, and platforms that would be crossed while executing an SOA business process, a federated approach to security may be adopted. In a decentralized SOA environment, simple perimeter controls and centralized security models may be ineffective. It is difficult and expensive to include security as an add-on capability. The fundamental security areas to address include authentication and identity management, authorization and confidentiality, integrity, availability, non-repudiation, auditing and monitoring, security administration, and policy management.

Establish a governance framework—An enterprise-wide adoption of an SOA is only possible if the policies and procedures governing its adoption are implemented and if the implementation is aligned with business goals.

Ensure Governance Is a Key Component of the SOA

Successfully implementing the changes necessary for enterprise-wide SOA adoption requires putting policies and processes in place, which are largely driven by business needs. Organizational commitment to governance is necessary for SOA initiatives to be successful.

SOA governance serves three primary purposes.

- **Defining roles and responsibilities:** Establishing roles and assigning responsibilities is very important to promoting ownership and accountability.

- **Aligning IT investments with business goals:** Business-driven IT governance ensures that IT is responsive to business needs. It helps define processes and policies that contribute to success, including identifying new services to support business goals and determining priorities and funding for establishing services.
- **Creating, communicating, and enforcing policies:** Governance provides consistency across an organization's service lifecycle management (e.g., modeling, assembly, deployment, and management), adherence to standards and best practices, reuse of services, and enforcement of operational policies across SOA runtime environments.

An SOA Center of Excellence, consisting of both business and IT professionals, can serve many of these SOA governance objectives, by establishing ways to measure the effectiveness of governance and facilitating the governance activities discussed below.

Oracle's SOA Governance Framework[49] defines SOA governance as the interaction between policies, decision makers, and processes that ensure an SOA's success. As identified in Oracle's SOA Governance Framework, key leverage points for SOA governance include architecture, technology infrastructure, information, financial, portfolios, people, project execution, and operations.

Architecture—An architectural approach sets up a minimal set of constraints to ensure consistency in service implementations, which results in services with improved interoperability, composability, and reusability when working on a shared infrastructure. As a best practice, an IT organization should consider developing a reference architecture to align architectural principles across lines of business. Governance should include policies for architectural documents, guidelines for service designs, processes to assess adherence to the reference architecture, and processes for reviewing and updating the architecture.

Technology infrastructure—Technology should be identified, sourced, and managed. Failure to enact technology policies may result in unjustified technology investments, redundant capabilities, or incomplete or non-functioning services with poor interoperability unfit for enterprise-wide adoption. As part of SOA governance, policies should be established to ensure that an infrastructure platform, which may include messaging, security, and other utility services, is employed by all projects. Consistency in developing services and migrating legacy systems and ensuring that investments in the infrastructure coincide with business goals should also be included in SOA guidance. For example, to maintain interoperability, policies should be created to ensure that technical solutions adhere to industry standards to maintain vendor neutrality.

Information—An effective SOA implementation requires addressing data quality and interoperability issues; otherwise, any unresolved issues can lead to poor data quality and inconsistent data. Governance should include defining data ownership, setting up a data architecture, establishing policies and guidelines to adhere to data standards chosen by the enterprise or domain, establishing policies to conform to data quality metrics, establishing security policies for data access, and developing SLAs for data services. As a best practice, it is useful to establish an enterprise data management group to establish data governance policies and procedures that cut across multiple business domains and monitor enforcement of the governance.

Financial—SOA projects usually have a steep ramp-up cost because creating reusable services requires more thoughtful designs and thorough testing. As a result, service providers bear a large portion of the costs rather than consumers, which is hard for service providers to justify in the short term. Financial decisions and policies should be designed to balance these costs. Funding should be allocated to sustain the development of common business and utility services and for the shared infrastructure. As a best practice, central funding should be considered for services that cut across program and/or organizational lines. In addition, an incremental funding strategy should be developed for individual SOA projects in which future investments are driven by their demonstrated business value.

Portfolios—Employing effective policies for governing the enterprise portfolio of services is important for ensuring consistency among service lifecycles, aligning project milestones with the SOA strategy, planning business and utility services, and ensuring that the existing enterprise applications are appropriately leveraged when developing services. As a best practice, the SOA approach should focus on high-value business processes driven by multi-year strategic plans that integrate business and IT goals.

People—Users and providers across the enterprise community can be the key drivers or resistors of an SOA implementation. Policies that encourage behavior in support of the SOA should be part of a governance approach. This includes assigning responsibilities and empowering people to drive process improvements; training people on the architecture, development, testing, and deployment of services; creating incentives for reusing services; and establishing an enterprise architecture group for SOA adoption. As a best practice, organizations should provide enterprise SOA training for their architects, developers, project managers, business analysts, and quality assurance teams.

Project execution—Since some projects are not well suited for an SOA implementation, policies and guidelines should be put in place to make these determinations. Policies are useful for prioritizing projects and aligning them with the SOA's roadmap. Addressing the funding, ownership, and management of services will drive consistency in service implementations and the reuse of services. Tackling the creation, storage, and retrieval of shared SOA artifacts and formalizing the governance of the lifecycle of services, business processes, and business rules are also important. As a best practice, using a service registry to catalogue existing services can help with reuse.

Operations—Shared services raise interesting challenges in runtime management. The runtime operation of services should be enforced with policies to ensure that SLAs are met. These include the operational model for services (e.g., who pays when additional resources are needed for a service), monitoring services to ensure they are meeting demands, handling policy exceptions and violations, and defining and enforcing runtime policies (e.g., security, access, logging, billing policies, and service reliability). As a best practice, it is helpful to initially centralize SOA operations and support. As the SOA matures, deploying a service management solution to enforce runtime policies in a more distributed manner should be considered.

SOA governance is critical to the success of an SOA effort. While SOA governance, especially early in the process, is important, defining an effective governance authority is unlikely to occur until the organization sees the value of the SOA. As part of the "start small, learn, and evolve" strategy, leaders should focus on motivating the organization to adopt the SOA approach and define success metrics and key performance indicators to determine the success of the governance strategy. Example metrics may include the number of applications using shared services, compliance with key aspects of a reference architecture, and adoption of enterprise standards.

Integrate Security Throughout the SOA Design

There are a variety of techniques that can be employed to significantly mitigate the increased threats associated with exposing services on the network when they are applied throughout an enterprise infrastructure. All security mechanisms involve tradeoffs, and effective security requires a cohesive, system-wide approach. Security should be considered when the architecture is first designed because of the tight coupling between security mechanisms and other architectural choices. Alan Radding explains that "like governance, security has to be considered from the start." [50] By addressing these needs early, an SOA implementation can be secured.

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.[51] This objective gives rise to the challenge: how to enable information sharing while protecting and securing the information being shared.[52] This can be successfully addressed by dividing it into three major areas and systematically tackling each one: empowering unanticipated users (if SOA will be used to implement an information sharing strategy that requires access privileges for 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 will be used to implement an information sharing strategy with 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.

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 and 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 ability to empower unanticipated users by using enterprise security services and attribute-based access control provides a firm technical foundation upon which trust can be established across organizational boundaries.[53] 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.

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.

> Confidence builds trust, and a common certification and accreditation process builds confidence among organizations.

Contemporary Web Service offerings utilize well-defined XML interfaces, and while there is increased vulnerability due to open ports,[54] 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 that an internal application will be 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.

Ensure the right security for each service—In many ways, protecting services in an enterprise is like protecting any other resource. Some resources require stringent security against sophisticated threats, while others may exist in relatively stable environments without any requirement for strict control. Services do not always need the same level of security, which is an important perspective in the development of an SOA.

For a successful SOA implementation, the right security mechanisms should be applied to the right services. Security should be balanced with other considerations (e.g., performance and scalability).

Set Your Expectations on the Return on Investment

A recent Cutter Consortium report[52] analyzed the state of SOA and surveyed 97 global firms to examine the adoption of SOA across industry. In their analysis, they surveyed corporations on the benefits they expect to gain from implementing an SOA. Some of the responses were as follows:

- 67 percent reported increased business flexibility.
- 64 percent expected cost-effective development of composite applications.
- 58 percent indicated lower software integration costs.
- 51 percent expected greater economies and efficiencies through the reuse of code.
- 51 percent would like to turn software from an inhibitor of business changes to an enabler.
- 44 percent reported an improved return on investment (ROI) on existing IT applications.

While cost savings is a primary motivation for adopting an SOA, the authors of this study also reported that a fundamental challenge to SOA adoption is determining how to pay for the SOA, given steep learning curves, a lack of technology skills and familiarity, a lack of mature industry standards, and a lack of user enterprise management ability (i.e., governance) to plan integrative SOA implementations. An organization adopting SOA approaches needs to have a realistic expectation on how much investment is needed and the expected ROI.

Expect upfront costs**—**In reality, an SOA will initially cost more money than traditional architectures. To estimate this cost, David Linthicum recommends a budget-to-budget best practice (i.e., budget some time to understand your domain in detail to help you make more informed budgeting decisions).[53] The cost of an SOA is typically expressed as follows:

> **Cost of SOA** = Cost of Data Complexity + Cost of Service Complexity + Cost of Process Complexity + Enabling Technology Solution

Costs are driven by multiple technical and organizational factors. From a technical perspective, new technologies will be employed with interfaces, layers, utility services, and adapters that must be supported. There are associated engineering costs for the design, including scalability and the generality of interfaces. On an organizational level, business analysis will have initial costs that will ultimately yield significant results. SLAs, cost recovery mechanisms, governance boards (e.g., service lifecycles), and enterprise security rules should be established to allow access and trust relationships to form. Most of the costs are not for the SOA implementation, but for areas that were previously ignored (e.g., business process modeling). An SOA just highlights the need and allows for cleaner alignment.

There are additional costs for designing the service for agility, which go beyond the initial investment. An assessment by Ken Vollmer and Mike Gilpin of Forrester Research shows SOA-based developments can be twice the cost of traditional approaches when viewed solely with respect to building a particular application component. Typical client-server approaches only optimize the current set of requirements and reuse (i.e., generalization of the interface) is rarely done. With an SOA, since designing the service (rather than the business logic or capability) is a first-order engineering item, it is an add-on cost to the software design, so the monetary payoff is generally not evident (e.g., savings based on ease of maintenance or reuse) until after the deployment. Initially, the SOA will cost more than traditional approaches.

The monetary savings may not be local**—**The cost savings associated with an SOA may not be realized by every part of an enterprise. Many of the costs associated with the SOA approach are from consolidating silos of applications with redundant functionality and data throughout organizations.[57] Case studies that demonstrate consolidating services using an SOA were discussed earlier. Service reuse leads to reduced maintenance costs, which results in cost savings across the enterprise. Along similar lines, SOA also encourages platform-agnostic services, enables virtualization of the infrastructure, and allows organizations to choose the most cost-effective, best-of-breed application components for particular functions.

Monetary savings may not be realized immediately**—**While an exact payback period is difficult to quantify, the win will initially come from the business analysis and work flow activity.

Long-term cost savings and the advantages over adversaries will come through the organizational agility that the SOA will provide and the ability to share information in accordance with net-centric tenets. The value of this organizational agility may be hard to measure, but some quantifiable financial rewards may include consolidating services to eliminate redundant capabilities, lower integration costs, and service reuse.

Savings may be more than monetary—In many cases, especially within the Federal Government, monetary savings is often not the primary concern. For example, within DoD warfighting systems, flexibility in system implementations can speed up adding and modifying capabilities within systems, which accelerates the deployment of urgently needed capabilities to warfighters. The end result is a faster delivery of improved capabilities and more effective executions of missions, which can potentially save lives.

Without clear monetary cost savings, incentives are important—Without clear and immediate cost savings, it is imperative to understand the stakeholder's incentives, which can come in two forms. First, the providers or consumers should see direct value. Part of this includes effective mechanisms that provide incentives for appropriate behavior among service providers and consumers. Alternatively, a visionary who has authority will see the broader benefit that provides the SOA's direction (i.e., there is support from the executive leadership). This typically occurs when there is a compelling business need to initiate change, and leadership support drives the change.

Examine SOA as Part of a Networked Enterprise

At the core of a networked enterprise is the ability to leverage IT resources that can adapt to evolving requirements and rapidly deliver new functionality to meet users' needs. As Federal agencies move toward a networked enterprise, a fundamental shift is required to move away from the acquisition and management of stand-alone systems with pair-wise connections. A refocus is needed for organizing distributed capabilities to foster easier reusability and composability to deliver new functionality in a timely and cost-effective manner. SOA may be useful for a networked enterprise, but an SOA alone is likely insufficient.

SOA may be useful for a networked enterprise, but an SOA alone is likely insufficient.

The DoD's Net-Centric Environment (NCE) provides a useful example. The DoD's concept of an NCE is defined as a framework for full human and technical connectivity and interoperability that allows all users and mission partners to share the information they need, when they need it, in a form they can understand and act on with confidence, and that protects the information from those who should not have it.[58] The DoD Net-Centric Services Strategy (NCSS)[59] defines four goals to achieve this vision.

- **Provide services:** Information and functional capabilities need to be available as secure services on the network.
- **Use services:** Existing services should be used to satisfy mission needs before creating duplicative capabilities.
- **Govern the infrastructure and services:** Policies and processes should be established for a single set of common standards, rules, and shared secure infrastructure and services throughout the DoD's enterprise to ensure interoperability.
- **Monitor and manage services via Global Information Grid (GIG) Network Operations (NetOps):** Services should be implemented in accordance with DoD's GIG NetOps Strategy and concept of operations to ensure situational awareness of the NCE.

The NCSS cites shared services and SOA as one way to achieve NCE's goals. Many of the goals are aligned with the advice presented in this paper on SOA. Services should be available, usable, and managed through policies and monitoring. These guidelines can help improve operational effectiveness, enable an improved, standards-based approach to achieve information sharing, and increase the service provider's agility through cost- and resource-effective reuse of capabilities.

SOA alone is insufficient for an NCE within the DoD or, more broadly, for a networked enterprise within the Federal Government. The enterprise should establish and enforce how the service's building blocks will be made available, secured, operated, and used. This challenge includes resolving issues of trust, funding, and governance. For example, an NCE entails establishing trust between peer units with services managed outside the vertical chain of command within the DoD.[60] Michelsen distinguishes between vertical integration (i.e., where one works within the organizational chain of command boundaries) and horizontal integration, which involves working across vertical chains of command. He points out that before realizing the value of reusable services, agencies should establish trust horizontally, put appropriate funding vehicles in place, and develop the right incentives to motivate change (e.g., openness and sharing are critical to realizing the net-centric vision).

Given the promise of enabling a networked enterprise through SOA, organizations should first ensure the SOA implementation supports the networked enterprise, consider the issues of establishing trust, and the financial and people-related aspects of governance.

Conclusion

This paper presents a set of best practices and key characteristics based on industry experience for successful SOA implementations. The intent of this paper is to leverage industry guidance on SOA by identifying a manageable set of best practices that are relevant to the Federal Government. Government leaders and MITRE engineers can use these best practices to ensure that customers are receiving advice aligned with industry trends and experiences.

Central themes of these best practices include the following:

- SOA is not just an architecture initiative or a technology implementation effort; a successful SOA implementation requires cultural changes in an organization. Organizations should shift their focus to service-oriented thinking and implement effective governance to align their IT portfolio with business drivers.
- Adopting an SOA is an incremental process; successful implementations focus on solving business and operational challenges in a scalable way.
- Data and security are important SOA considerations; insufficient attention to data or security can compromise the success of the SOA effort.
- Benefits of an SOA largely occur at the enterprise level and may not be evident in a single project or be purely monetary.

References

[1] Gartner, Inc., Ted Friedman, “Where Data Meets SOA: Data Services”
http://www.ebizq.net/webinars/6953.html 28 Jun. 2006

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

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

[4] OASIS, “Reference Model for Service-Oriented Architecture 1.0” Organizations will see the benefits of shared services, business agility, and simplified integration if their SOA adoption strategies are motivated by and aligned to resolving business problems.
http://www.oasis-open.org/committees/download.php/19679/soa-rm-cs.pdf 2 Aug. 2006

[5] ZapThink, Ronald Schmelzer, “The ROI for SOA”
http://www.zapthink.com/report.html?id=ZAPFLASH-20050127 27 Jan. 2005

[6] Aberdeen Group, Peter Kastner, “The Composite Applications Benchmark”
http://www.aberdeen.com/summary/report/benchmark/RA_CompositeApps_RS-PK_3716.asp Dec. 2006

[7] IBM Corporation, Kathryn Gould and Amit Magdieli, “Optimizing Government Effectiveness Through Shared Services: Perspectives from IBM Corporation”
http://www-03.ibm.com/industries/government/doc/content/resource/thought/2365833109.html Apr. 2007

[8] Heather Havenstein, “Hammering Out Web Services Links”
http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=101084&intsrc=article_pots_bot 18 Apr. 2005

[9] InfoWorld, Galen Gruman, “SOA Ensures Guardian Gets It Right”
http://www.infoworld.com/article/05/05/02/18FEsoaguardian_1.html?s=feature 2 May 2005

[10] InfoWorld, Galen Gruman, “Countrywide Financial Simplifies Lending”
http://www.infoworld.com/infoworld/article/05/05/02/18FEsoacountrywide_1.html 2 May 2005

[11] Network World, James Kobielus, “The ROI of SOA”
http://www.networkworld.com/techinsider/2005/101005-roi-of-soa.html?page=2 10 Oct. 2005

[12] Open Group, Alcedo Coenan, “SOA Agility in Practice”
http://www.opengroup.org/projects/soa-case-studies/uploads/40/11405/SOA_Agility_in_Practice.pdf 4 Nov. 2005

[13] IT Week, James Murray, “Interview: BT Cuts Costs with Service-Oriented Architecture”
http://www.itweek.co.uk/articles/print/2162201 11 Aug. 2006

[14] Enterprise Systems, Leigh Alexander, “Case Studies: SOA in an Age of Legacy Integration”
http://www.esj.com/enterprise/article.aspx?EditorialsID=2150&pg=1 19 Sept. 2006

[15] InfoWorld, Leon Erlanger, “Transamerica Turns Silos into Services”
http://www.infoworld.com/infoworld/article/05/05/02/18FEsoatransamerica_1.html 2 May 2005

[16] Microsoft Corporation, “Case Study: Sony Pictures Entertainment”
http://www.microsoft.com/casestudies/casestudy.aspx?casestudyid=48976 2 Oct. 2005

[17] ZDNet, Joe McKendrick, “Middleware Is Cool Again (just ask eBay)”
http://blogs.zdnet.com/service-oriented/?p=675 3 Aug. 2006

[18] Executive Office of the President, Office of Management and Budget, “Planning, Budgeting, Acquisition, and Management of Capital Assets”
http://www.whitehouse.gov/omb/circulars/a11/2002/part7.pdf June 2002

[19] ZapThink, Ronald Schmelzer, “Right-Sizing Services”
http://www.zapthink.com/report.html?id=ZAPFLASH-20051115 15 Nov. 2005

[20] ZDNet, Joe McKendrick, "Where and Why Do We Still Need Tight Coupling?" http://blogs.zdnet.com/service-oriented/?p=591 13 Apr. 2006

[21] Loosely Coupled, Luc Clement, "Risks of Running SOA Without Registry" http://www.looselycoupled.com/opinion/2005/clem-risk-gov0228.html 28 Feb. 2005

[22] Gartner, Inc., R. Matlus and K. Brittain, "Creating a Service-Level Agreement for an IS Organization" http://www.tarrani.net/shared/GartnerCreatingSLA4ISOrg.pdf 21 Jan. 2002

[23] Gartner, Inc., William Maurer, et al., "A Guide to Successful SLA Development and Management" http://www3.gartner.com/DisplayDocument?id=314581 16 Oct. 2000

[24] Gartner, Ted Friedman, "Where Data Meets SOA: Data Services" http://www.ebizq.net/webinars/6953.html 28 Jun. 2006

[25] InfoWorld, David Linthicum, "Using a Common Data Model with SOA" http://weblog.infoworld.com/realworldsoa/archives/2007/07/using_a_common.html 3 Jul. 2007

[26] IBM Corporation, Mei Selvage, et al., "Achieving Semantic Interoperability in a SOA" http://www.ibm.com/developerworks/webservices/library/ws-soa-seminterop.html 23 Jun. 2006

[27] Microsoft Corporation, Pat Helland, "Data on the Outside vs. Data on the Inside" http://msdn2.microsoft.com/en-us/library/ms954587.aspx#dataoutsideinside_topic1

[28] http://www.defenselink.mil/cio-nii/docs/Net-Centric-Data-Strategy-2003-05-092.pdf

[29] Federal Times, Jim Pruden and Ashutosh Kulkarni, "Commentary: Improving Data Quality Is the First Step in Info Sharing" http://federaltimes.com/index.php?S=2483627 16 Jan. 2007

[30] Federal Times, Jim Pruden and Ashutosh Kulkarni, "Commentary: Improving Data Quality Is the First Step in Info Sharing" http://federaltimes.com/index.php?S=2483627 16 Jan. 2007

[31] SOA World Magazine, Tieu Luu, et al., "SOA Data Strategy–Vital to a Successful SOA Transformation" http://webservices.sys-con.com/read/233667.htm 15 Jun. 2006

[32] InfoWorld, David Linthicum, "Using a Common Data Model with SOA" http://weblog.infoworld.com/realworldsoa/archives/2007/07/using_a_common.html 3 Jul. 2007

[33] ZapThink, Ronald Schmelzer, "What's NOT Killing SOA?" http://www.zapthink.com/report.html?id=ZAPFLASH-20071016 16 Oct. 2007

[34] InfoWorld, Galen Gruman, "Countrywide Financial Simplifies Lending" http://www.infoworld.com/infoworld/article/05/05/02/18FEsoacountrywide_1.html 2 May 2005

[35] IBM Corporation, Jean Wang, "Develop SOA Solutions for Healthcare Organizations Using Business-Driven Development" http://www.ibm.com/developerworks/webservices/library/ws-soa-bddhealth/index.html 15 Sept. 2006

[36] ZapThink, Jason Bloomberg, "The SOA Pilot Pitfall" http://www.zapthink.com/report.html?id=ZAPFLASH-2005711 12 Jul. 2005

[37] In this context, granularity is referred to in terms of the business process rather than the technical architecture or the amount and frequency of data flowing over the network.

[38] IBM Corporation, Ali Arsanjani, "Service-Oriented Modeling and Architecture" http://www.ibm.com/developerworks/library/ws-soa-design1/ 9 Nov. 2004

[39] IBM Corporation, Gang Chen, "SOA and the value received when business drives IT decisions" http://www-01.ibm.com/software/success/cssdb.nsf/CS/CPOR-79V4M8?OpenDocument&Site= 14 Dec. 2007

[40] ZapThink, Jason Bloomberg, "The SOA Pilot Pitfall" http://www.zapthink.com/report.html?id=ZAPFLASH-2005711 12 Jul. 2005

[41] The SOA Magazine, "Intel Chief Architect Steve Birkel on SOA Practices and Intel's SOA Adoption" http://www.soamag.com/I12/1107-1.asp 6 Nov. 2007

[42] IBM Corporation, Ali Arsanjani, "Service-Oriented Modeling and Architecture"
http://www.ibm.com/developerworks/library/ws-soa-design1/ 9 Nov. 2004

[43] IBM Corporation, Gang Chen, "SOA and the Value Received When Business Drives IT Decisions"
http://www-01.ibm.com/software/success/cssdb.nsf/CS/CPOR-79V4M8?OpenDocument&Site= 14 Dec. 2007

[44] Forrester, Randy Heffner, "Embedded SOA Management Solutions"
http://www.forrester.com/Research/Document/0,7211,43206,00.html 22 Oct. 2007

[45] InfoWorld, David Linthicum, "Scalable SOA Solutions Continue to Emerge"
http://weblog.infoworld.com/realworldsoa/archives/2007/06/scalable_soa_so.html 3 Jun. 2007

[46] IBM Corporation, "Perspectives on Grid: Virtualization as a Foundation for SOA Environments"
http://www.ibm.com/developerworks/grid/library/gr-soavirt/ 26 Jun. 2006

[47] Software Mag.com, Rizwan Mallal, "Pillars of SOA Testing"
http://www.softwaremag.com/L.cfm?Doc=0206-ThoughtLeadership-R-Millal Feb. 2006

[48] Align Journal, Ryan Bagnulo, "Case Study: SOA Delivers for Wachovia—With Interest"
http://www.alignjournal.com/index.cfm?section=article&aid=466 14 Nov. 2007

[49] Oracle Corporation, Mohamad Afshar, "SOA Governance: Framework and Best Practices"
http://www.oracle.com/technologies/soa/docs/oracle-soa-governance-best-practices.pdf May 2007

[50] Application Development Trends, Radding, Alan. "The SOA-wise Enterprise"
http://www.adtmag.com/article.aspx?id=18695&page=1 1 Jun. 2006

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

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

[53] Enterprise security services and Attribute Based Access Control are described in more detail in our white paper, Seven Significant Challenges for Federal Leaders Employing Service-Oriented Architecture.

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

[55] Cutter Consortium, Andrew Schwarz, "The State of SOA Today"
http://www.cutter.com/architecture/fulltext/reports/2008/01/index.html Jan. 2008

[56] InfoWorld, David Linthicum, "How Much Will Your SOA Cost? Here Are Some Guidelines."
http://weblog.infoworld.com/realworldsoa/archives/2006/11/how_much_will_y.html 19 Nov. 2006

[57] Network World, James Kobielus, "The ROI of SOA"
http://www.networkworld.com/techinsider/2005/101005-roi-of-soa.html 10 Nov. 2005

[58] Department of Defense, "DoD Net-Centric Environment Joint Functional Concept Version 1.0"
http://www.dtic.mil/futurejointwarfare/concepts/netcentric_jfc.pdf 7 Apr. 2005

[59] Department of Defense Chief Information Officer, "DoD Net-Centric Services Strategy"
http://www.defenselink.mil/cio-nii/docs/Services_Strategy.pdf 4 May 2007

[60] CrossTalk, John Michelsen, "For Net-Centric Operations, the Future Is Federated"
http://www.stsc.hill.af.mil/crosstalk/2007/09/0709Michelsen.html Sept. 2007

MITRE

www.mitre.org

MITRE