



An Executive Forum on Business Change

**Industry Recommendations for DoD
Acquisition of Information Services and
SOA Systems**

July 7, 2008

SOA Acquisition Working Group

The Association for Enterprise Integration

An Affiliate of the National Defense Industrial Association

Association for Enterprise Integration
2111 Wilson Boulevard, Suite 400
Arlington, Virginia 22201
www.afei.org



2111 Wilson Boulevard
Suite 400
Arlington, Virginia 22201

Mr. David Wennergren
Deputy Chief Information Officer
Department of Defense
6000 Defense Pentagon
Washington, D.C. 20301-6000

Mr. Timothy Harp
Deputy Assistant Secretary of Defense
C3ISR and IT Acquisition
Assistant Secretary of Defense, Networks and Information Integration
6000 Defense Pentagon
Washington, D.C. 20301-6000

July 7, 2008

Gentlemen,

I am pleased to submit this product of the AFEI Executive Forum SOA Acquisition Working Group, convened to address business models and acquisition of services in a DoD service-oriented architecture. Achieving a net-centric Department of Defense with a services-oriented information environment is generally recognized as offering potential benefits in terms of cost and agility while requiring potentially significant changes to Department processes.

The purpose of this work is to explore the issues surrounding acquisition of services capabilities and provide specific recommendations on how Acquisition Strategies and Request for Proposals (RFPs) could be structured to incorporate principles of Service Oriented Architecture (SOA). It provides industry views on acquiring capabilities vice systems and, by extension, services within the SOA construct. The paper explores how services acquisition models differ from the traditional acquisition model and provides recommendations on how to leverage the best of both models.

The Association and its members are pleased to support the efforts of the DoD CIO and the defense community, and look forward to continuing to develop the concepts, ideas and recommendations contained in the document.

Respectfully,

David E. Chesebrough, P.E.
President



Acknowledgements

AFEI wishes to acknowledge the contributions of all those who made their time and skills available to complete this project. In particular AFEI thanks MaryAnn Kiefer of Level Consulting and Don Johnson of ASD (NII) who provided expert coordination and guidance as the project evolved.

AFEI also recognizes the senior leadership provided for the group by Mr. David McQueeney, Chief Technical Officer for IBM Federal, and Mr. Michael Burnett, Joint C2 Mission Systems Director, Northrop Grumman. Early in the creation of this group all recognized that discussing the impact on defense industry business models of the adoption of service-oriented architectures required input by those most affected – the providers of technology and the integrators of systems. The tireless efforts and wisdom of these gentlemen in leading and guiding this analysis are immensely appreciated.

AFEI also acknowledges the invaluable contributions of the principal authors of the report, who spent a great deal of time and energy in discussing and debating the issues, drafting the report, and in briefing their findings before the DoD Enterprise Architecture Conference, the Defense Science Board and the U.S. Navy CANES Program Office even prior to its release.

There were many others who participated in some aspect of the development of this project, and AFEI thanks them and their organizations for their support.

Finally, AFEI thanks the Office of the Assistant Secretary of Defense (Networks and Information Integration) / Chief Information Officer, DOD for turning to AFEI to provide and industry perspective on this transformational initiative of moving to service-oriented systems and service-enabled capabilities.

AFEI is deeply indebted to all of the organizations who willingly provided their best talent to participate in developing this report. This is an excellent example of the results that can be achieved when industry and government enter into collaborative efforts for the good of the community.



AFEI Executive Forum on Business Change

Industry Recommendations for DoD Acquisition of Information Services and SOA Systems

Submitted to

Mr. David Wennergren
Deputy Chief Information Officer
Department of Defense

Mr. Timothy Harp
Deputy Assistant Secretary of Defense
C3ISR and IT Acquisition
Assistant Secretary of Defense
Networks and Information Integration

July 7, 2008

Abstract

The purpose of this document is to provide specific recommendations to programs participating in a DoD services-based enterprise on how Acquisition Strategies and Request for Proposals (RFPs) could be structured to incorporate principles of Service Oriented Architecture (SOA). It provides industry views on acquiring capabilities and, by extension, services within the SOA construct. The paper explores how services acquisition models differ from the traditional acquisition model and provides recommendations on how to leverage the best of both models.

The industry recommendations contained herein are intended to assist the DoD in shaping a services and SOA-based procurement strategy that can better achieve an agile DoD enterprise.

The SOA Acquisition Working Group focused on how IT procurements for delivery of service-enabled capabilities vice traditional systems should be structured. This paper is not intended to address the general acquisition process (i.e., large scale weapon system procurements); it addresses how existing IT systems can join the larger DoD enterprise through service orientation. The focus is more on organizational and process challenges rather than technical challenges.



Principal Authors

MaryAnn Kiefer, Level Consulting

Dave McQueeney*, Chief Technical Officer, IBM Federal

Michael T. Burnett, Joint C2 Mission Systems Director, Northrop Grumman

Rob Walker*, Vice President, BEA Systems

Hans Polzer, Corporate Fellow, Lockheed Martin

Tim Pavlick, PhD., DoD Chief Architect Office of the Federal CTO, IBM

Dave Pratt, PhD., Chief Scientist/Engineer, SAIC

John Sutton, Vice President, McDonald Bradley



Industry Contributors

Kelly Brown*, EM Solutions

Art Fritzson*, Booz Allen Hamilton

Herb Kelsey*, Computer Associates

Mike Burnett, Northrop Grumman

Bert Kramer, Lockheed Martin

Sonia Schmitt, Lockheed Martin

Hans Polzer, Lockheed Martin

Renee Stevens, Mitre

Roger Loeb, IBM

John Sutton, McDonald Bradley

Beverly Seay, SAIC

Greg Gardner, Oracle

Jim Stodgill, Accenture

Susan Yochim, Northrop Grumman

Jill Scheidhauer, Northrop Grumman

Dave Miller, Mitre

Frank Petroski, Mitre

Jim Seeley, Northrop Grumman

Joan Baumstarck, EDS

* Member, AFEI Board of Directors

Government Advisors

Michael Krieger, Deputy CIO, US Army/G-6 (formerly with ODCIO)

Tim Harp, ASD (NII)

Don Johnson, ASD (NII)



Table of Contents

| | |
|---|-----|
| Executive Summary | vii |
| 1. DoD Services-Based Environment..... | 1 |
| 1.1 Services Oriented Architecture and a Services-Based Approach | 1 |
| 1.2 Why is DoD Migrating to a Services-Based Environment? | 2 |
| 2. Why Should the DoD Acquisition Model Be Different in a Services and SOA-Based Environment? | 3 |
| 3. Planning for Your SOA Acquisition and Your Business | 5 |
| 3.1 Sequencing of Capabilities..... | 5 |
| 3.2 Enterprise Engineering vs System Engineering | 8 |
| 3.3 Government/Supplier Ecosystem Business Model | 9 |
| 3.3.1 Evolving Ecosystem Acquisition Model | 9 |
| 3.3.2 Potential Ecosystem Player Roles in a Diverse SOA | 10 |
| 3.4 Transition Planning..... | 11 |
| 4. Acquisition Strategy Considerations..... | 13 |
| 4.1 Development/Execution Models..... | 13 |
| 4.1.1 Program Office Size and Functions..... | 13 |
| 4.1.2 Organizational Conflict of Interest (OCI)..... | 14 |
| 4.1.3 Implementing Effective Management Controls..... | 14 |
| 4.1.4 Risk Management..... | 15 |
| 4.1.5 Budget Planning | 16 |
| 4.2 Role of Small Business | 17 |
| 5. Request for Proposal (RFP) Considerations | 18 |
| 5.1 Leveraging the Statement of Objectives (SOO) | 18 |
| 5.2 Acquiring Services and Service Level Agreements..... | 18 |
| 5.3 Other RFP Considerations | 19 |
| REFERENCES..... | 20 |
| Appendix A. Example of an Agile SOA Model | 1 |

Executive Summary

DoD leadership is directing the Military Services and Defense Agencies to migrate IT programs to service-oriented systems and service-enabled capabilities. This approach better enables information sharing and allows rapid, more cost effective creation and deployment of new and unanticipated functions in response to dynamic mission environments. The challenges to this migration are significant: a DoD acquisition system that is predicated on a high degree of independence between systems; an ‘a priori’ lack of clearly defined system interfaces; an emphasis on development rather than operations; and a lack of incentives for system owners to provide capabilities to any user beyond their predefined base. The DoD must overcome these challenges to achieve the Service Oriented Architecture (SOA)-based environment it desires in the future.

This broad industry team is recommending an Agile Model for acquiring and implementing this future environment. The Agile Model promotes incremental definition, development and delivery of capabilities, rather than a traditional system development, that forces requirements and architecture lock-down early in the life cycle. Each increment, or spiral, delivers capabilities that encompass the SOA “stack” (SOA platform and mission services) based on a mission thread. This allows the SOA platform and mission services to be proven on the real network and systems infrastructure before the next spiral is delivered. Once each increment is fielded for operational use, the users will assist in re-prioritizing the mission content of the next spiral. This Model is also contrasted with a “big bang” delivery of the SOA platform without having mission services to evaluate the platform capabilities.

The SOA Agile Model requires new perspectives on the Government/Supplier ecosystem business model. There is a shift to new roles: Enterprise Managers, Component Developers, and Platform/Commodity Infrastructure Providers. Contracting for these roles is dependent of the granularity of the sub-roles within each; more granular sub-roles increase visibility into contractor performance but increase the need for more government oversight. Appropriate Organizational Conflict of Interest (OCI) provisions and open acquisition approaches such as open systems, standards-based reference models, and published test standards allow contractors to bid in one or more of these roles.

The Agile Model requires much greater cooperation and interaction with the contractor base than in traditional procurements. Acquisition plans and business models have to be re-assessed with each spiral to allow the Government to rebalance the contractor evaluation and management criteria (e.g., SLAs), and to ensure shared Government-contractor value and fairness. The Government will have to offer reverse SLAs (RSLAs) to contractors to guarantee the performance of existing infrastructure and platform capabilities. This high level of interdependency is not seen in the current environment.

A repository for Agile Model acquisition best-practices should be considered. Templates for acquisition artifacts such as performance specifications, SLAs, evaluation criteria, and contract clauses should be developed. Programs would be required to understand what services already exist within DoD or partner organizations before acquiring new services. Achieving this DoD services-based environment will require leadership commitment to change current acquisition processes.

1. DoD Services-Based Environment

The purpose of this document is to provide specific recommendations from industry on how Acquisition Strategies and Request for Proposals (RFPs) could be structured to incorporate principles of Service Oriented Architecture (SOA). It provides industry views on acquiring capabilities and, by extension, services within the SOA construct. The paper explores how services acquisition models differ from the traditional acquisition model and provides recommendations for DoD on how to leverage the best of both models. The recommendations are intended to be useful to acquisition programs contributing in a DoD services-based enterprise.

This paper was developed for the Assistant Secretary of Defense (Networks and Information Integration)/Department of Defense (DoD) Chief Information Officer (ASD(NII)/DoD CIO) by the SOA Acquisition Working Group of the Association for Enterprise Integration (AFEI), an affiliate to National Defense Industry Association (NDIA). The recommendations apply to the DoD and potentially to any of its mission or business partners. The benefits and constraints of net-centric data and services, and associated governance challenges, have been adequately addressed in the AFEI Data Sharing and Services Strategy Working Group white paper, *Facilitating Shared Services in the DoD*, February 2006 and the AFEI Information Sharing Working Group report, *Federated Governance of Information Sharing Within the Extended Enterprise*, January 2008, references (a) and (b). These papers are a preamble to the acquisition issues addressed by this paper and are currently available at www.afei.org.

These recommendations are intended to assist the DoD in shaping a services and SOA-based procurement strategy that can better achieve an agile DoD enterprise.

The SOA Acquisition Working Group focused on how IT procurements for delivery of service-enabled capabilities, vice traditional information systems and applications, should be structured. This paper is not intended to address the general acquisition processes that result in large-scale weapon system procurements. It addresses how existing and new IT systems can acquire service orientation. The focus is more on organizational and process challenges rather than technical challenges.

1.1 Services Oriented Architecture and a Services-Based Approach

DoD leadership is directing warfighter and business programs to migrate to service oriented systems so that capabilities can be more responsive to the changing needs of the military consumer. The DoD is motivated to achieve similar successes as industry in accelerating their ability to respond to a rapidly changing market by applying maturing SOA technologies. Many DoD programs have begun the process of service enabling their systems but have yet to extend those services beyond the boundaries of their specific program or accredited environments.

The Organization for the Advancement of Structured Information Standards (OASIS) defines SOA as: *a paradigm for organizing and utilizing distributed capabilities that may*

be under the control of different ownership domains (reference (p)). This definition indicates that SOA is not simply the application of technology or service enablement but also management of capabilities to achieve a mission given the diversity and required cooperation of different ownership domains.

A service-based approach reduces the intricacy of system applications and data by abstracting out much of the complexity. In a SOA environment, standards-based interfaces allow access to functionality and data that is encapsulated within systems. This is accomplished by decomposing tightly coupled, highly integrated systems approaches into components that can be offered at a fraction of the cost of the entire system, and deployed in significantly reduced time frames. An added benefit of the service-based approach is that, once decomposed in this fashion, new and unanticipated functions can be created through novel combinations of services.

The benefit of a well thought out SOA implementation is that it provides a platform for rapid innovation.

Maturing standards are allowing programs to service-enable their systems. Today, that enablement is generally done within the boundaries of a system, a program, or an organization. The ideal construct for the DoD would be to leverage these services across program, system, or organizational boundaries to form aggregated capabilities by applying the principles of a SOA. This requires DoD to address how to specifically procure services for consumption outside of these traditional boundaries and how program risk associated with sharing these services could be mitigated.

1.2 Why is DoD Migrating to a Services-Based Environment?

Many commercial enterprises adopt SOA as a means of breaking up inflexible IT infrastructures, which are usually characterized by monolithic, customized applications. Through the use of commercially accepted design and development standards, the definition of a new service or redefinition of an existing service can take place with reduced testing and implementation requirements.

SOA promises to deliver substantial benefits for the Federal Government. However, SOA requires significant change within Federal organizations and carries some inherent risks. The benefits stem largely from SOA's ability to deliver agility and service reuse. The risks arise from the technical and governance interdependence (e.g., end-to-end capability validation, change and version control, unanticipated cost from cross program governance) beyond the boundaries of specific programs or organizations. The net results of broad-based adoption and maturation of SOA capability throughout the Federal Government (reference c) include:

SOA offers a:

- Reduction in dependence on proprietary technologies;
- Streamlined development process; and
- Re-use of business and IT assets.

- Improved responsiveness as a result of simplified delivery of enhanced services;
- Increased information sharing through reuse of business and mission assets; and
- Improved security, transparency, and resilience as a result of consistent use of a standards-based infrastructure.

2. Why Should the DoD Acquisition Model Be Different in a Services and SOA-Based Environment?

The current DoD acquisition model is predicated on several assumptions:

- Systems are more or less independent of one another, and any dependencies are clearly defined prior to acquisition;
- System development and procurement are the major budget and acquisition risk drivers, and the Government will operate the resulting system using O&M funds;
- System owners do not require any specific incentives to provide services to each other, or to use services provided by others efficiently.

The tacit assumption underlying the current acquisition model is that programs will be appropriately funded to support a known demand. In a services or SOA-based environment, it is difficult to predict demand, especially for services that don't already exist, and to budget accordingly. In this environment programs would be expected to support whatever service requests might arise, or to negotiate specific service level agreements (SLA) between service providers and consumers on an ad-hoc, bilateral basis. The current and foreseeable budgeting process is carried out over timeframes of years. However, demand for services could be much more rapid and unpredictable. Individual programs currently have no inherent motivation or incentive to make their services available to a broader range of users. Conversely, programs have very strong motivations to avoid dependency on services outside their control to reduce risk and improve their chances of success. Negotiating and managing an ever-increasing number of service agreements with other organizations adds undesirable administrative load to programs and creates further performance management risks.

Not only does a services-based approach inherently question the assumption of systems being more or less independent of each other, it raises the new question of providing services that may be common across multiple systems. In the past, this commonality might have been addressed by government-off-the-shelf (GOTS) components that were shared across multiple programs from a developmental perspective in the acquisition process. These GOTS components generally were subsumed into each individual system's structure and budget, and then operated, and perhaps modified, independently of one another.

In contrast to most acquisitions, services don't necessarily have to be acquired as part of a complete system.

The services-based approach makes it technically feasible to have one system or program provide a common service to all the other systems or programs that have a similar functional need. There is a resource-sharing potential among operational programs that has not been feasible in traditional system development and acquisition. Shared services and infrastructure resources have the potential to facilitate information sharing among multiple systems.

The DoD has business models in place for managing shared resources, but they typically have not been applied to information services, and they generally are not dynamic or market-driven. There are few, if any, incentives for making services accessible outside a given program's boundaries or using another program's services.

More importantly, there are quite a few disincentives for doing so. The unpredictable nature of service usage and quality of service (QOS) demands by other systems makes sharing services a risky behavior, and places more emphasis on scalability and adaptability of services during development.

This increases acquisition cost without any corresponding immediate benefit realized by the system sponsor. Service developers or service providers can leverage existing infrastructure and services to provide additional functionality as incremental services.

A services-based approach suggests that multiple, semi-independent service development and procurement actions are a viable alternative for building large-scale capabilities. This alternative raises performance liability and indemnification questions when infrastructure and shared services provided by the Government or another party do not meet expected performance levels.

3. Planning for Your SOA Acquisition and Your Business

Planning for a SOA-based acquisition requires that sequencing of capabilities, roles, responsibilities, and legacy systems transitions must be defined. Implementing a SOA-based enterprise is a transformational endeavor; there are best practices, processes and organizational considerations that must be addressed. As an organization moves to an open architecture of a SOA and capabilities-based approach, it must re-think how:

- Requirements are formulated (as capabilities) and a evolutionary spiral consumption business model is followed
- Business models are executed (buying services vs. hours or systems)
- Contractual and Intellectual Property are treated (movement toward a mixed ownership model, e.g., contractor owns some things, government some, some are shared)
- Organizations are formed and governance is executed (movement toward arranging of organizations by delivered capability vs. deep hierarchies and a governance structure that mirrors the new structure)
- Functional architectures are arranged by capabilities
- Technical architectures are service-based and technical capabilities are separated horizontally by role
- Software development and engineering are reoriented to deliver incremental capabilities in a progressively more specific manner
- Logistics changes as it relates to individual capabilities/services vs. entire systems

3.1 Sequencing of Capabilities

The most important element in the deployment of a SOA-based system is to understand how the capabilities will be deployed. Not only does this affect the legacy system cut over and user acceptance, but the success of the system will reside in the ability to rapidly innovate new capabilities on the deployed SOA platform. The sequencing of capabilities is a two-fold effort.

Figure 1 illustrates the implementation sequence of SOA components as indicated by the numbers. Enterprise SOA implementations have 4 basic functional groups:

- Networking Infrastructure (identified as layer -2),
- Systems Infrastructure (identified as layer -1),
- SOA Platform (identified as layers 0, 2, 3, 4, 5), and
- Mission Services (identified as layers 1, 6, 7).

The first step, once the infrastructure is in place, is to develop and deploy the SOA platform. Once the platform has been created, the desired mission services can be

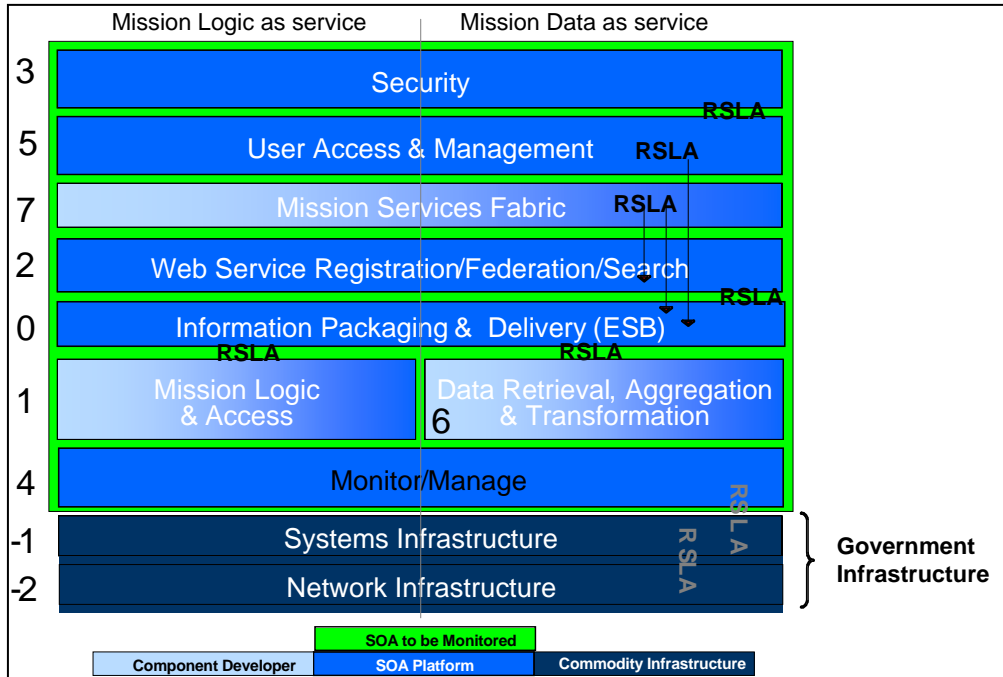


Figure 1 – SOA Implementation Sequence

developed and deployed. As shown in the Figure 1, many of the platform elements will require Service Level Agreements (SLAs) or Reverse SLAs (RSLAs)) that help to quantify and manage the interactions between the layers. Appendix A provides a more detailed example of an enterprise SOA implementation and associated sequencing.

A key challenge is to characterize and express requirements in the form of capabilities (most often expressed as mission threads or use cases), to be satisfied by services. Services should be defined in terms of their relationship to each other, mission threads, and to service groups. The key challenges are the appropriate and relevant definition of:

Mission threads allow users to understand what capabilities are contained or enabled in each release.

- Service content
- Service quality (QOS)
- Service trust
- Relationship of services, service groups and service infrastructure to each other, e.g., pre-requisite
- SLAs

Service definition and acquisition must be defined by the capabilities and these performance characteristics versus acquiring software lines of code (SLOC), Earned Value, or physical end items. It is necessary to have defined and implemented the platform as a prerequisite for a service to operate.

It is important to move away from the typical model of procuring all desired capability at once. While we think of incremental spirals in regard to mission capability, it is also important to refine our consumption of SOA infrastructure services. The state of the government commodity infrastructure (-2, -1 in Figure 1 above) upon which SOA will ride is often unknown. Hence, it will be difficult to predict the ability of that infrastructure to support a full SOA.

SOA promotes incremental definition, development and delivery of capabilities while traditional system development forces requirements and architecture lock down early in the life cycle.

The SOA Agile Model (sometimes called the Spiral Consumption Model) is an alternative to big bang acquisition. Figure 2 describes a SOA Agile Model and compares it with a more traditional development model. The Agile Model, shown in the lower half of Figure 2, dictates that *both* the (SOA) mission services *and* the SOA infrastructure services be acquired in a spiral manner. For example, the Government should order the most minimal set of SOA platform services in order to deliver, or test, the initial service-based mission capabilities. If the platform services acquired in any given procurement are minimized, a number of problems in the procurement cycle are solved (e.g., time to deliver, time to value, vendor investment and ability to test capability on the Government's existing infrastructure).

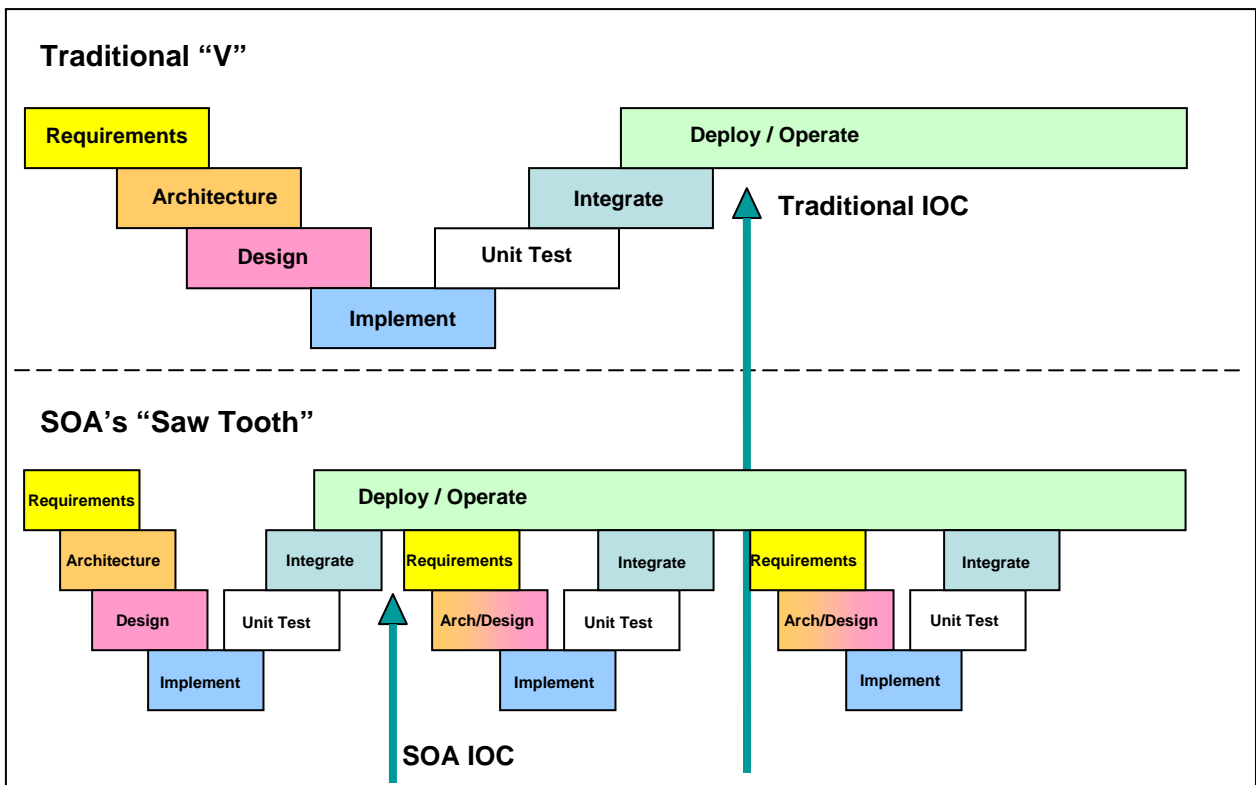


Figure 2. SOA Agile Model Compared to Traditional Development Model

Another aspect of the Agile Model is to address the business model in synchronization with the technology model. This allows for system innovation to occur at the same rate as the business and process changes. The Agile Model requires assessment checkpoints along the way. Spiral consumption of platform and mission services allows

re-thinking of what may be ordered in the next services spiral. Similarly, because the DoD business models for SOA are as yet unproven, it will be important to reassess the utility of the particular SOA business model at play in each small increment. Whether the chosen acquisition model is internal, outsourced, a managed service or some other model, it is important that the Government rebalance the evaluation and management criteria (e.g., ensuring SLAs are fair at each checkpoint).

Services and SOA-based environments promise benefits to the Government. However, to realize those benefits, it is necessary for industry to perceive the business model as being fair and one they can endorse. This will ensure service-based acquisitions are successful and deliver value to both Government and industry.

Fielding a mission thread will result in users re-prioritizing the content of the next mission thread.

Because the design and delivery of SOA-based capabilities is so different, the payment methods need to change to accommodate the new delivery models. Transferring the risk to the contractor (e.g., pay per transaction) for the SOA platform in Figure 1 makes contracting difficult because a basic infrastructure will be necessary prior to generating warfighter mission demand.

Conversely, mission specific services could be contracted for with a pay per use model. This assumes that the Government will provide the contractor with 1) an SLA for the underlying layers of technology, and 2) the constraining architectures for which the government previously contracted. This reinforces that there is an order of deployment for SOA-based systems to enable this separation of roles.

3.2 Enterprise Engineering vs. System Engineering

The scope of a SOA implementation extends beyond the traditional system and organizational boundaries. It involves a broad cross-section of an enterprise's functions and infrastructure. Enterprise engineering is the cyclic process of managing the evolution and interconnection of the enterprise. Since most, if not all, the data and services will come from sources beyond the control of a user organization, enterprise engineering helps manage the relationships, risks, and dependencies associated with the SOA model. In some cases, the needed capability might not be delivered via a system development effort; rather a change in the mission or business process might suffice.

One of the key elements of enterprise engineering is the maintenance of a test environment for prototyping, innovation, integration, and risk reduction. The more closely the test environment mirrors the operational environment, the lower the deployment risk. In addition to the typical new system tests, comprehensive regression testing must be done to ensure the new services do not break the deployed system or cause a breach of the SLAs.

Enterprise engineering must also consider information assurance (IA). Horizontally integrated systems implemented using SOA face the same IA issues that traditional systems face, but there are some unique challenges around the spiral, or "saw tooth", nature of development and management of multiple sources of services. Traditional IA certification and accreditation (C&A) processes are characterized by milestone-oriented periodic events; SOA development approaches and tools deliver capabilities

incrementally and continuously throughout the lifecycle. The traditional C&A process certifies discrete completed (or “static”) systems; SOA systems are incrementally and dynamically constructed, extended, and maintained by piecing together (composing) services obtained from different components. Implementing a SOA-based system and capabilities depends on resolving some key technical IA challenges and involving the security organizations from the very beginning of the effort.

3.3 Government/Supplier Ecosystem Business Model

The Government's acquisition approach has a major impact on industry. The advent of a true capability-based acquisition process will be a transformational change. Once the Government changes its acquisition model, the supplier community will change in response.

3.3.1 Evolving Ecosystem Acquisition Model

A new acquisition model requires changes to the business model and the contractual terms and conditions. New roles will have to be defined. This acquisition model will segment contractor roles between SOA platform and infrastructure and SOA mission services. Figure 3 addresses the shift in contractor roles as DoD evolves a SOA-based environment. The left side of Figure 3 shows the traditional roles various contractors have played in the acquisition process.

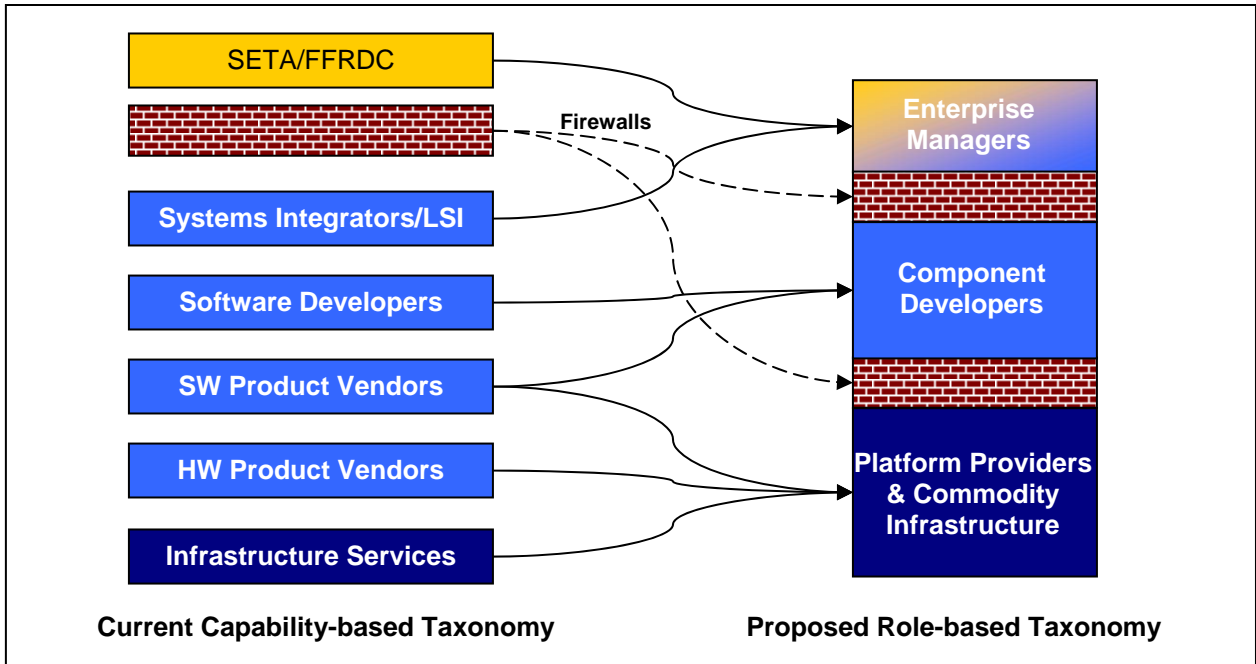


Figure 3 – Shift in Contractor Roles

The shift from systems to service/capability-based acquisition, combined with horizontal and enterprise integration and consolidation of the contractor community, is causing the shift to the model shown on the right. The business model must allow for severability by function and ownership.

The model implies that governance is not only separated by service or set of services, it is most likely separated by function. Communities of Interest (COIs) deal with a particular cluster of services to accomplish certain functions, typically mission oriented, which link enterprises. A domain is a larger grouping of functions that encompasses several COI's, i.e., a larger grouping of clusters of services (focusing on the SOA mission services).

An enterprise is a cross-domain logical construct that might be within an organization or across organizations. The enterprise provides the infrastructure and commodity components of the SOA platform as well as mission services. For example, a ship is an enterprise. This enterprise includes the logistics, finance, and surface warfare C2 domains and the SOA platform that supports them. The logistics, finance, and surface warfare C2 domains provide the architecture within which the COI's have to conform. They also provide some core services that span multiple COI services such as access control and service management. Those domains include COI services (service clusters) that require participation by elements beyond the ship. For example, these COIs might be providing purchasing and radar services.

3.3.2 Potential Ecosystem Player Roles in a Diverse SOA

In the role-based taxonomy in Figure 3, there are three major roles—Enterprise Manager, Component Developer, and Platform/Commodity Infrastructure Provider. Each of these major roles can be further decomposed into more granular sub-roles as shown in Table 1.

Depending on the scope (e.g., domain, enterprise, or COI capability) and type of acquisition, each of these sub-roles might be contracted individually, or bundled into the three major roles. Bundling simplifies the overall acquisition, but the more granular approach provides a cleaner separation of roles and responsibilities, enterprise monitoring, and management of risk. The granular approach requires a more precise monitoring and management architecture that provides more government insight but requires more effort.

Within a domain there may be specific technical standards and designs that allow particular functions to operate effectively. These domain-specific designs adhere to the domain architecture, which is within the bounds of the enterprise-wide architecture. The domain architectures and designs must be related for the capabilities to work correctly.

The more granular the roles are contracted, the greater the visibility, but more government oversight is required.

Table 1 – Role-based Taxonomy Descriptions

| Major Role | Sub-role | Description |
|---|--|---|
| Enterprise Manager | Enterprise Architecture/Engineering | These roles may be separate or contracted for as one function. Avoid contracting for a broad enterprise. The resulting large scope of the domain and the COI engineering will cross roles and create the possibility of blurred responsibilities. |
| | Domain Middleware Architecture or Engineering | Architecture for domain middleware functions could be reused in each domain and customized. Likewise the architect may engineer the function in one or more domains. |
| | Domain Architect/Engineer | This contractor would ensure that the individual domain architectures are built out into a fully functioning set of capabilities. This contractor, providing oversight and integration, would set limits and define general boundaries within which the other architects in that domain operate. This ensures that separate architecture or engineering efforts are not designing cross-purposes. |
| | COI Architect/Engineer | This contractor would be required to have mission specific knowledge. The only limit to this contractor's engineering of several domains is breadth of knowledge of warfare functions and warfare-specific technologies. |
| Component Developer | Capability Module Designer/Builder for a service group | This job is to create a capability and its requisite clusters of related mission services. |
| | Service Cluster Designer/Builder | This contractor operates with constraints previously defined by the architects/engineers. This contractor defines and delivers a related cluster of mission services using those design rules. |
| | Individual Mission Service Provider | This is a niche role with limited scope. Awarding this contract would have to balance the procurement effort required against the limited extent of the job being done. Alternatively if a services-centric blanket purchase agreement exists, it may be relatively easy for the Government to define a new service and task a contractor. |
| | Service Provisioning | Developing services is distinct from the delivery model. The options pertain to the different delivery models that the Government might consider. |
| Platform/Commodity Infrastructure Provider | Infrastructure Provider | This contractor provides the computational and network layers of the system. |
| | Infrastructure Provisioning | Provisioning of the net-centric backbone is a separate role for bundling infrastructure when needed. |

3.4 Transition Planning

The move to a SOA in an operational setting requires two prior shifts: 1) movement to an open architecture (OA), and 2) movement toward a capabilities-based approach. OA is a necessary condition for SOA as systems must first be open to interconnecting their computational ability. These resources could be consumed via hardwires, interfaces, services, or any other means. Hence OA is a necessary, *but not sufficient*, condition for SOA. An extension of OA requires that systems be designed via a capabilities-based approach and access be provided along capability lines (vs. traditional systems boundaries).

SOA deployments will have to interact with legacy systems for the foreseeable future. The coexistence with, and transition from, legacy systems must be planned. Figure 4 shows the historical evolution of systems and provides insight into the transition path.

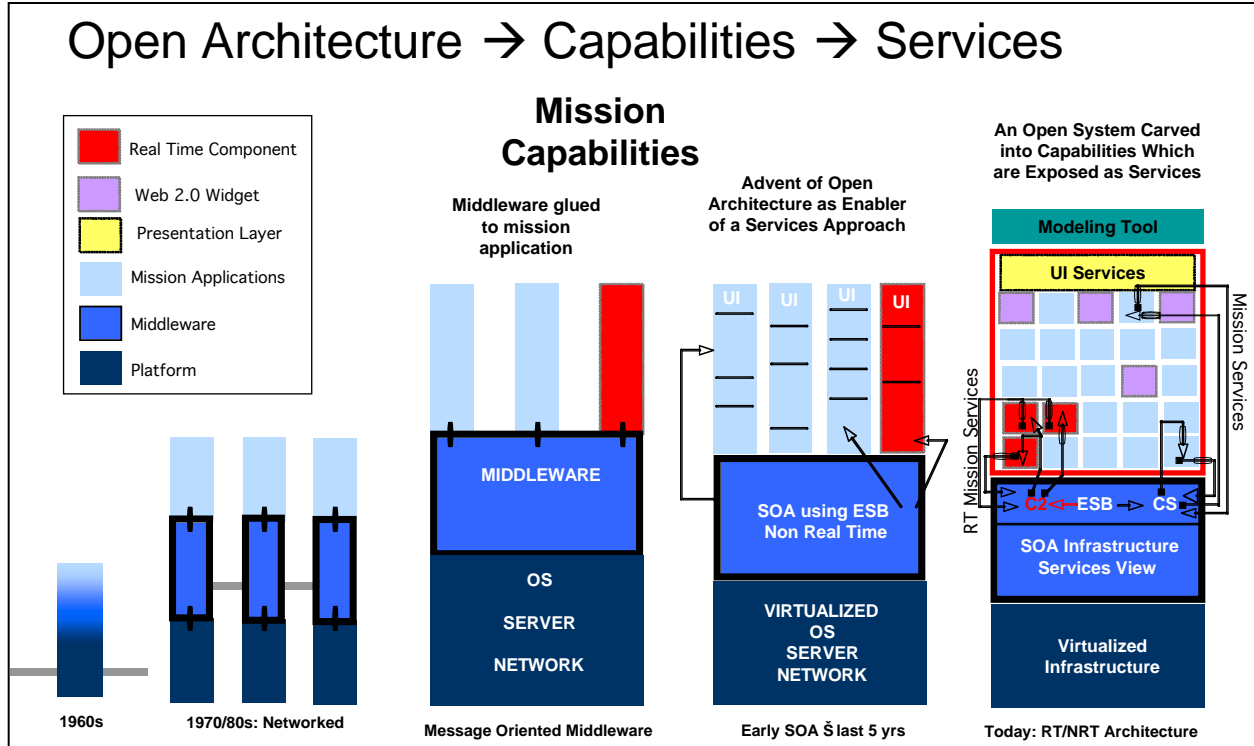


Figure 4. Evolution to Mission Capabilities via SOA

Moving from left to right, the mission applications are first “wrapped” and then decomposed into services as the infrastructure layers are commoditized. Wrapping and segmentation must be sequenced in concert with the planned capability rollout. Figure 4 shows the functional part of the model being segmented by mission capabilities. Mission services are the services coming into and going out of the mission capabilities. The OA aspect of the evolution to SOA for mission systems is shown via the horizontal segmenting of the technology layers.

4. Acquisition Strategy Considerations

As in traditional acquisitions, the priority must be appropriate stewardship of taxpayer dollars. However, there are some additional considerations that are introduced based upon the collaborative and iterative nature of SOA acquisition and deployments.

4.1 Development/Execution Models

Aligning the acquisition model roles with organizational and functional considerations will help ensure that the government can draw best of breed in the marketplace and not be constrained by the artificiality of the procurement system.

SOA will demand more cooperation and interaction with the contractor base than in traditional acquisitions.

4.1.1 Program Office Size and Functions

There are two extremes in the structure of a program office. We suggest that the acquisition agency adopt one or the other for its program office to minimize confusion of roles.

- Large program management office (PMO). This could be at the program executive officer level and encompass multiple traditional programs that are converging on a SOA implementation. If the Government PMO is large with appropriate SOA competencies, it can act as the chief engineer and chief architect. It may in-source the configuration management, integration, and operations functions to the enterprise manager (government organization). The large PMO probably does all the source selection. In this model the integration and performance risk resides with the government.
- Small PMO. This would be more likely at the traditional program level. In this case, the enterprise manager (contractor) should be a partner to the PMO, who is weak by virtue of size and limited technical capability. The enterprise manager owns the chief engineer and chief architect. The government can participate in the specification development, but the enterprise manager develops and owns them. The enterprise manager defines and executes the integration and test and operates the resulting system. In this model, the integration and performance risk resides with the enterprise manager.

The execution model should flow from the level of competency the acquiring office has in the SOA domain.

The small PMO model is recommended while the government builds its program management and engineering competency in the SOA and services space.

4.1.2 Organizational Conflict of Interest (OCI)

The enterprise manager, infrastructure provider, and component developers are the three major roles to be filled by contractors. Typically, there are firm OCI clauses in place between each of contractor roles in the capability-based taxonomy in Figure 3. The consolidation of the contractor base and evolution of the commercial IT market have reduced the number of viable contractors while increasing their capabilities. To ensure the government gets the best value and most capable system, one recommendation is to eliminate the hard-line OCI restriction and allow the provider to compete in the open architecture. Using a “firewall” allows the enterprise manager to compete to be the infrastructure provider, component developer, or both, and vice versa. This works if the SOA platform layer(s) are developed in the open, e.g., all artifacts are placed in an accessible repository and created using open standards (i.e., UML). One of the key ways to mitigate the OCI concerns between the various players of the SOA enterprise is the use of standards-based reference models and well-defined and published interfaces. Similarly, using well-defined and published test and acceptance criteria ensures transparency in vendor selection. The definitions and specifications must be published and available to all interested parties while they are being created. The requirement to create a standards-based, open system serves to mitigate the specification of a system for a company’s product, but also helps to prevent restrictive IP and vendor lock-in. One of the main PMO tasks is to ensure the openness of the system to minimize unfair competitive advantage and “proprietary lock-in.”

Open systems, defined interfaces, and published test standards reduce OCI concerns.

4.1.3 Implementing Effective Management Controls

The key to implementing effective management controls is to document a clear picture of the key drivers of the system and to build a metrics plan to measure and assess them. Given the iterative nature of a SOA-based system, the metrics (both the metric and value) will need to adapt as well. While the development of a metrics plan is beyond the scope of this document, this section will provide some insights in how those metrics can be enforced. A combination of several enforcement mechanisms normally works best for a given situation.

The management controls will need to change and adapt at each spiral.

Service Level Agreement (SLAs)

An SLA is a formal contract that exists between consumers of the service and the service provider(s). SLAs define roles and dependencies and set expectations between service consumers and producers. SLAs record the common understanding about services, priorities, responsibilities, guarantees, measurement points, the level of service metrics, and thresholds. For example, an SLA may specify the levels of availability, quality, or other attributes of the service, like billing and even penalties in the case of violation of the SLA.

The development and enforcement of performance SLAs (uptime, availability, etc.) is straightforward but needs to be well thought out since collection of metrics to assess

SLA compliance imposes a burden. SLAs that cross over functional and component boundaries (response time to a user query) are much more difficult and will require more detailed analysis in case of breach.

Risk-Reward Model

In this model, the contractor is asked to assume a greater risk in terms of investment or performance, with the expectation of having a greater reward. One of the most common ways this control is used in the commercial marketplace is for the vendor to provide a service that is paid for out of the savings accruing to the user. In the government marketplace, this translates to “share in savings” or share in cost avoidance, but requires that a realistic cost basis be computed and projected ahead of time.

Enhanced Services Model

In the current model contractor’s incentives are to provide the minimal solution that meets the requirements. As the requirements grow and change, Engineering Change Proposals (ECPs) are required, resulting in real or perceived “scope creep”, and always in cost growth. This results in delays in deployment and fielded capability.

Under the enhanced services model, the contractor will be able to exceed the requirements by providing expanded or additional services. These could be “sold” to other users as a common reusable service that others do not have to develop. The additional services could also provide the basis for further development as the system matures.

Managed Services/Software as a Service (SaaS) Model

In both of the preceding models, the shift is toward having a vendor provide a metered service to the user over the network with payment for usage charges rather than buying and owning a system. Some of these implementations are wholesale outsourcing of a fundamental and widespread capability, while others are much more atomic in nature. The key difference between the two models tends to be the level of control passed to the provider.

In a managed-service model, the responsibility for providing the entire service is passed to the vendor. In SaaS, only the software service is under the vendor’s control. Given DoD security and operational requirements, SaaS is only likely to be implemented if the Government adopts an enterprise monitoring and management strategy by “sponsoring” a SaaS instance. Government agencies may be able to provide SaaS services to other government agencies.

4.1.4 Risk Management

In a SOA and services-based environment, the focus shifts from buying things to buying capabilities. This requires decomposition of the capability into parts in which some entity can be held accountable for the parts and the whole. The more composite the capability, the more diffuse the responsibility. The more diffuse the responsibility, the more difficult it is to identify who carries the risk.

In the large PMO model, it would typically be the government as the enterprise manager. In the small PMO model, the contractor acting as the enterprise manager would assume the risk. A finer grained allocation of roles and responsibilities provides more precise

monitoring and a management architecture that improves visibility while enabling better assessment of failure mechanisms. A diffuse allocation increases risk by increasing the “seams.”

The application of modeling, piloting and testing can instill a degree of confidence in the system, thereby allowing for the enterprise manager to understand and manage the risk. For mature systems, the performance and usage characterization should be well known and can be modeled in the test environment to predict how the capability will function when it is deployed. For new capabilities or instances, the uncertainty and error thresholds of the model will be quite large. Over time, these uncertainties will be reduced as familiarity with the system grows. This is why the agile model works well for SOA systems - you build the minimum configuration, not the whole SOA, then you test it against the other variables, such as other systems, networks, clients, etc. SLAs are modified based on those results and development continues further up the stack and across the enterprise.

Performance risk increases each time a new service is added and is reduced as the system and services mature.

This approach requires a viable test environment, and the government assumes much of the risk in the early phases of the SOA system deployment. As the system matures and performance is better understood, the risk is passed to the enterprise manager and more creative cost models can be exercised. The component developer’s risk is that of non-performance in the test environment, which could result in cancellation of the component developer’s contract. Once the component developer’s services have been successfully tested, the component risk is passed to the enterprise manager. Warranty clauses could be used to protect against some undetected bug or unknown limitation in the component.

4.1.5 Budget Planning

A spiral consumption model means that a SOA-based system will spend significantly less time in the development phase and will enter the operational phases much earlier, but with less capability than a fully developed system in the traditional model. However, once the first version of the SOA-based system begins operations, users start providing inputs for future iterations.

There are budget planning implications of this approach. A shorter development phase will shift more of the development costs into what has been the operations and maintenance (O&M) phase. Since the scope of the system and content of each spiral will evolve, the amount and type of development is not fixed at the beginning of the acquisition. This model acknowledges the changes in scope and the requirement for post-deployment support upfront versus the traditional thinking that scope and content are fixed at the beginning and ECPs are required.

Early IOC means more capability will have to be developed and funded during the traditional O&M phases.

The other difference in SOA-based systems is the increased expectation of shared, or reused, services and components. System use of components developed or managed elsewhere reduces development costs but might incur recurring license, usage, or support costs over time. These costs might be born by the system being procured or

they might be born by the other component or service provider. This will require a buy/build cost tradeoff to be conducted across the domain/enterprise as part of the system/component/service acquisition process.

4.2 Role of Small Business

There are four fundamental reasons why contracts go to small business:

1. Fair competition
2. Set aside/small business goals
3. Niche business that doesn't warrant large business participation; and
4. Innovative market/capability creation and leadership

The last two reasons are key benefits of the SOA approach. As the componentization of the services increases in granularity, more and more niche areas will be created in which small business can excel and provide services and components. However, as the scale of the enterprise increases, there will be pressure to use proven services and components from larger companies. This is reflected in the current innovate/acquire cycle in the commercial IT market.

| |
|---|
| Niche services enable small business to play in the service development role. |
|---|

5. Request for Proposal (RFP) Considerations

Procuring services and capabilities may take different forms within the context of a SOA. Today, the government acquisition processes are aligned to system procurements in which the requirements satisfaction is managed within the scope of a contract. Organizational and process changes may be necessary to transition to a service or capability-based enterprise. An RFP in the future might look different with the SOA approach.

5.1 Leveraging the Statement of Objectives (SOO)

The statement of objectives (SOO) is a brief description of the basic, top-level objectives of the program being acquired with the RFP. The SOO is written to address product-oriented goals and performance-oriented requirements. The SOO defines the end-state results and gives the contractor/provider flexibility in process, design, and approach to achieve the results. In the transition from traditional system acquisitions to a SOA-based system acquisition, the emphasis of the SOO should be on mission and user objectives.

The PMO should engage prospective bidders prior to RFP release to construct a notional work breakdown structure (WBS) that emphasizes mission and capabilities. The WBS helps the system and enterprise governance structures organize the SOO to ensure all requirements have been included. It can be used to form a logical arrangement of the elements required of the SOO or the associated performance specifications. The WBS can also provide insight into what enterprise capabilities will be leveraged and reused to mitigate risk and add capability within the larger enterprise. This will be especially important in managing the development, integration, and operation of enterprise services and capabilities.

Place more focus on the enterprise, rather than just the system, as an integral part of the RFP.

In a SOA environment, a goal is provide capability more rapidly to users than the current acquisition approach. As service repositories and registries are populated with functional and data services, RFPs should require or encourage reuse and linkage with existing services before building duplicative ones. The RFP should consider the end user and the domain/enterprise as well as the unique system requirements.

5.2 Acquiring Services and Service Level Agreements

In considering the products expected from the acquisition, the contracting organization should ask the following questions:

- How much of the system will be oriented towards services to the enterprise?
- Who will use the services (consumers and providers)?
- Who will manage the services (enterprise service manager)?

- What governance will support the services (enterprise configuration board; portfolio governance; domain/COI; architect)?

Governance provides the management oversight for defining, developing, deploying and operating services and components, including standards and specifications, across the domains, enterprises, and the DoD and mission partners.

SLAs provide a metric for contract performance.

SLAs should be used in the contract and, once the service/component is in operation, they are the measurement of contractual fulfillment as described in Section 4.1.3.

The SOO should require that the prospective bidders define the boundaries and SLAs as they relate to governance. At the same time, the government may want to describe existing governance structures that will be required for the bidders to participate within the enterprise.

5.3 Other RFP Considerations

Evaluation factors influence how prospective bidders respond to the RFP. One evaluation factor that should be considered is the requirement for an operational demonstration. Prospective offerors should be challenged to demonstrate that a component or service can be operated under certain mission constraints. Demonstrations can be done locally, in simulation or via joint combat exercises.

The government should consider weighting an “enterprise” perspective versus a system perspective in evaluations. This weighting would increase the credit given to proposals that consider reuse of existing components and services rather than development of new components and services.

Special contract clauses are needed to address business factors that should be considered as part of the eco-system acquisition model. Two areas of special clauses that may need to be addressed include OCI (see Section 4) and warranty considerations. Other special clauses might address issues such as intellectual property or rights provisions, or indemnification, given the distribution of responsibility and risk across the system.

Templates should be developed with repeatable examples that can be readily inserted into the SOO, performance specifications, SLAs, evaluation factors, and contract sections of RFPs. Templates for these acquisition artifacts would be considered in developing acquisition strategies for systems and programs transitioning into the SOA and services-based DoD enterprise. Programs would be required to research and understand what components or services already existed within the enterprise before acquiring new components and services.

REFERENCES

- a) AFEI, Data Sharing and Services Strategy Working Group, Facilitating Shared Services in the DoD, February 2006
- b) AFEI, Information Sharing Working Group, Federated Governance of Information Sharing Within the Extended Enterprise, January 2008
- c) Practical Guide to Federal Service Oriented Architecture; version 0.7, Nov 13, 2007; Architecture and Infrastructure Committee, Federal Chief Information Officers Council

DoD and Government Directives

- d) Interim Guidance on DoD Information Assurance Certification and Accreditation Process (DIACAP) dated 6 July 2006. Supersedes DoDI 5200.40 and DoD 8510.1-M.
- e) DoD CIO Directive 8320.2 Data Sharing in a Net-Centric Department of Defense 12/02/2004, <http://www.dtic.mil/whs/directives/corres/html/832002.htm>
- f) DoD Instruction 8500.2, Information Assurance (IA) Implementation, 6 February 2003; <http://www.dtic.mil/whs/directives/corres/html/85002.htm>.
- g) DoD Instruction 8520.2, Public Key Infrastructure (PKI) and Public Key (PK) Enabling, 1 April 2004. <http://www.dtic.mil/whs/directives/corres/html/85202.htm>.
- h) Chairman of the Joint Chiefs of Staff (CJCSI) 6510.01D, Information Assurance (IA) and Computer Network Defense (CND), 15 June 2004. http://www.dtic.mil/cjcs_directives/cjcs/instructions.htm#6000
- i) National Security Telecommunications and Information Systems Security Policy (NSTISSP) No. 11, Fact Sheet for the National Information Assurance Acquisition Policy, July 2003. http://www.cnss.gov/Assets/pdf/nstissp_11_fs.pdf.
- j) DoD IT Standards Registry (DISR). https://disronline.disa.mil/a/DISR/DISR_reports.jsp.
- k) DISA Security Technical Implementation Guides (STIGs). <http://iase.disa.mil/stigs/index.html>

Standards/Specifications

- l) NCES User Guide, http://www.disa.mil/nces/nces_user_guide.html
- m) NCES Standards
- n) IETF RFC 2246, Transport Layer Security (TLS) 1.0
- o) IETF RFC 2459, Internet X.509 Public Key Infrastructure (PKI) Certificate and CRL Profile
- p) OASIS Reference Model for Service Oriented Architecture 1.0
- q) OASIS Security Assertions Markup Language (SAML) 1.1
- r) OASIS Universal Description, Discovery and Integration (UDDI) version 3.0.2
- s) OASIS Web Services Security (WS-Security) 1.0
- t) SOAP version 1.1 and/or 1.2
- u) W3C XML Digital Signature (XMLDSIG)
- v) Web Service Definition Language (WSDL) version 1.1
- w) WS-I Basic Profile 1.1
- x) Computer, IEEE Computer Society, October 2003, Vol 36, No. 10, pp 46-52

Appendix A. Example of an Agile SOA Model

Early DoD SOA acquisitions were typically the application of SOA to existing Programs of Record (PORs). These PORs implemented SOA to make program capabilities accessible via services. Lessons learned from early SOA implementations can be used to frame enterprise implementations. Acquisition and technology models will change as enterprise SOA implementations are undertaken.

A.1 Implementation Sequence for SOA Agile Model

The 'Agile Model' in Figure A-1 shows a best practices sequence for SOA implementation. The numbers indicate the sequence in which individual functions should be implemented.

Enterprise SOA implementations have 4 basic functional groups:

- Networking Infrastructure (identified as -2),
- Systems Infrastructure (-1),
- SOA Platform (0, 2, 3, 4, 5), and
- Mission Services (1, 6, 7).

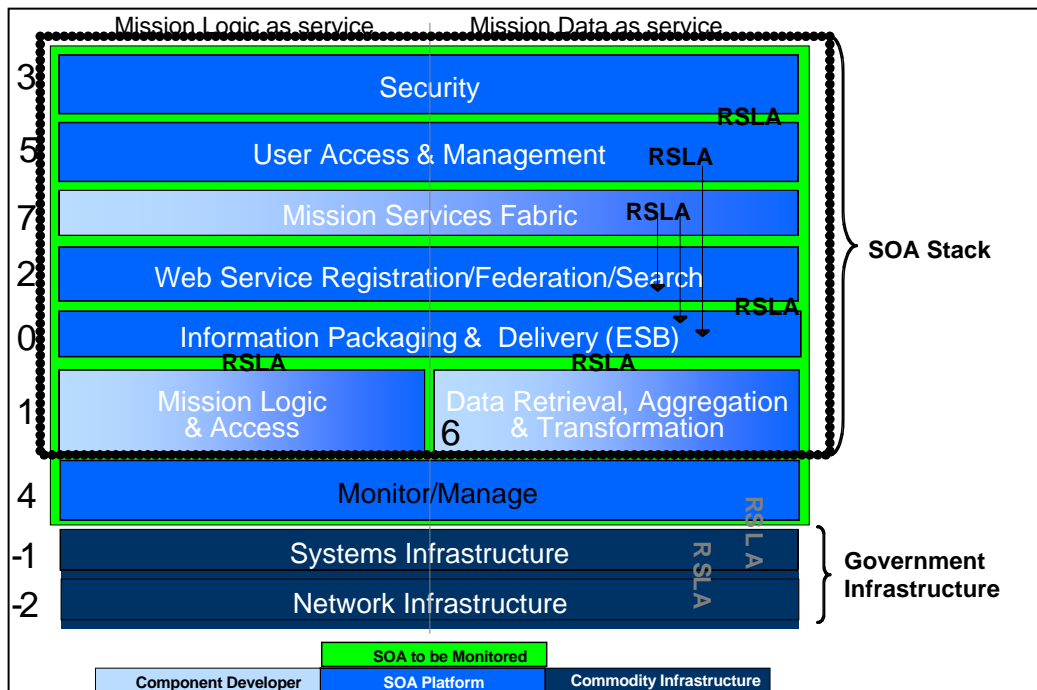


Figure A-1. SOA Implementation Sequence

This appendix will describe a sequence for implementation and a division of labor that provides for a clean separation among SOA contractors and SOA functions. The acquisition model should reflect a similar sequence of buys, and re-balancing of business models between buys.

Re-balancing of business models implies that lessons learned from earlier agile sequence implementations inform the requirements going forward and the business conditions under which the contract will exist. SLAs will change as the sequence unwinds.

In Figure A-1, networking and systems underlie the SOA stack (encapsulated by the dotted line). This figure is intentionally very granular to illustrate that different contractors could be used for each of the most granular pieces. In practice, the Government is likely to let collections of these pieces to a single contractor. An example of this grouping is indicated by the colors. It is assumed that a networking infrastructure must exist two steps prior (hence the -2 in the sequence) to the implementation of the SOA stack. Similarly, to deploy a software SOA stack, a systems (server & operating systems(s)) infrastructure is also a prerequisite. A network and systems infrastructure that function according to an agreed upon profile, e.g., SLA, are a necessary condition for any SOA deployment.

Reverse SLAs (RSLA)

Often times networking or systems acquisitions are measured by SLAs. These are “one way” contracts for service levels that the contractor owes the Government. However, often times the Government has infrastructure upon which the successful performance of these contracts depends. A RSLA is an agreement by the Government to the contractor that the Government provided infrastructure will operate within a set of performance parameters. Hence the contractor’s commitment, the typical SLA, is conditional upon the performance of the Government’s infrastructure. This dependency is acknowledged and it should be formalized.

The Government networking infrastructure is not assumed to operate with 99.9999% availability. In an operational environment, it is specifically assumed that there will be times when systems must operate in disconnected mode, or over very narrow band connections, or in intermittent fashion. However, this does not preclude describing the as-operated characteristics on the networking infrastructure in an RSLA. If one desires the SOA contractor(s) to be held to defined performance standards it is necessary to define the dependencies.

A major source of problems in prior SOA implementations has been the lack of defining the dependencies upon the Government. Networking capability must precede both the systems infrastructure and SOA implementation. Furthermore, the capability performance must be defined, if even at a gross level, in an SLA format. This RSLA must be guaranteed to the contractors further up the stack. Any dependency on existing infrastructure should be called out and defined in the most precise manner possible.

Often times the Government does not know the performance characteristics of the network or systems infrastructure, particularly as it combines with new software capabilities. The behavior of the network under the stress of new capabilities may not be known until the network is exercised. This assumption is often used in wargaming and applies here as well. This is one of the primary reasons for re-balancing the business model after each agile spiral. Both the Government and the Contractor will be more cognizant of their capabilities and responsibilities.

This variability of operations must not be ignored or minimized; it is a fact of complex IT infrastructures. SOAs are meant to evolve. If it is assumed that more will be learned about the network as we progress through the SOA implementation sequence, then the

approach to SLAs will benefit from this increased knowledge. SOA acquisitions will be structured to incorporate increasingly more operational network information and specificity. More importantly SOA acquisitions will not assume that perfect networks exist prior to the acquisition. Starting out with a coarse grained set of expectations, on both the Government and contractor sides will provide for a more exploratory mindset in the early spirals. This expectation is more consistent with what actually occurs in early SOA implementation spirals. The business model established through early spiral explorations will provide a more accurate basis for a well-defined business model for a later phase more scalable SOA.

Agile Defined

The spiral development method assumes that we really don't know what to build later until we see how the earlier builds perform. In traditional spiral software development projects it is assumed that the Government knows all of its requirements ahead of time even if they chose to implement them in an incremental fashion. In traditional spiral developments, the contracting terms and conditions are set ahead of the spirals and are not intended to change as each spiral is delivered. The only negotiation between contractor and Government, for each spiral, is to determine what functionality is included in each spiral.

Demand can also be exercised in a spiral fashion. Given that, and the variable nature of our infrastructure, later spirals cannot and should not be defined ahead of preliminary spirals. The Government's consumption of SOA should be determined experimentally as the sequence of SOA implementation shown in Figure A-1 unfolds.

SOA requirements and terms should be metered to account for the increased knowledge of infrastructure operational characteristics that will be gained through the process. The requirements and conditions will change as each spiral is executed, hence the need to re-balance the business model. The Government wants operational characteristics that it can promise to its customers. The contractor desires a set of terms that more accurately characterize the 'to be deployed in' environment. Contractors will provide more reasonably priced SOA services if they know more about the 'to be deployed in' environment.

The fundamental premise of 'Agile' is that the nature of what is desired will change as capability spirals out. Both Government and contractor should reset expectations via renegotiation at each spiral prior through minor contracting changes for the next spiral. The fundamental terms of the contract won't change. Requirements may be swapped out. SLAs and RSLAs will be refined. However the contractual framework remains essentially unchanged. Under these assumptions, it is not necessary to renegotiate entire contracts per spiral. The Government only need be open to data changes within each spiral contract mod. This concept allows for a fast, orderly execution of the contractual side of SOA implementation.

A side benefit is that the Government may be able to accelerate the time frame of the SOA implementation based on lessons learned from the prior spiral. Often the Government asks for accelerated implementation with unrealistic expectations, but the contractor doesn't understand or know the risk in advance and the implementation cannot be delivered as expected. This uncertainty is always 'priced in' the cost to the Government. In the case of Agile, the acceleration is not defined a priori but will be

based upon the collection of sufficient information from earlier spirals for both parties to manage risk.

A.2 Practical Steps to Implement the Agile Model

The Government's SOA acquisition strategy should address both the SOA platform increments as well as the mission services. Figure A-2 details the steps of SOA platform and mission services implementation.

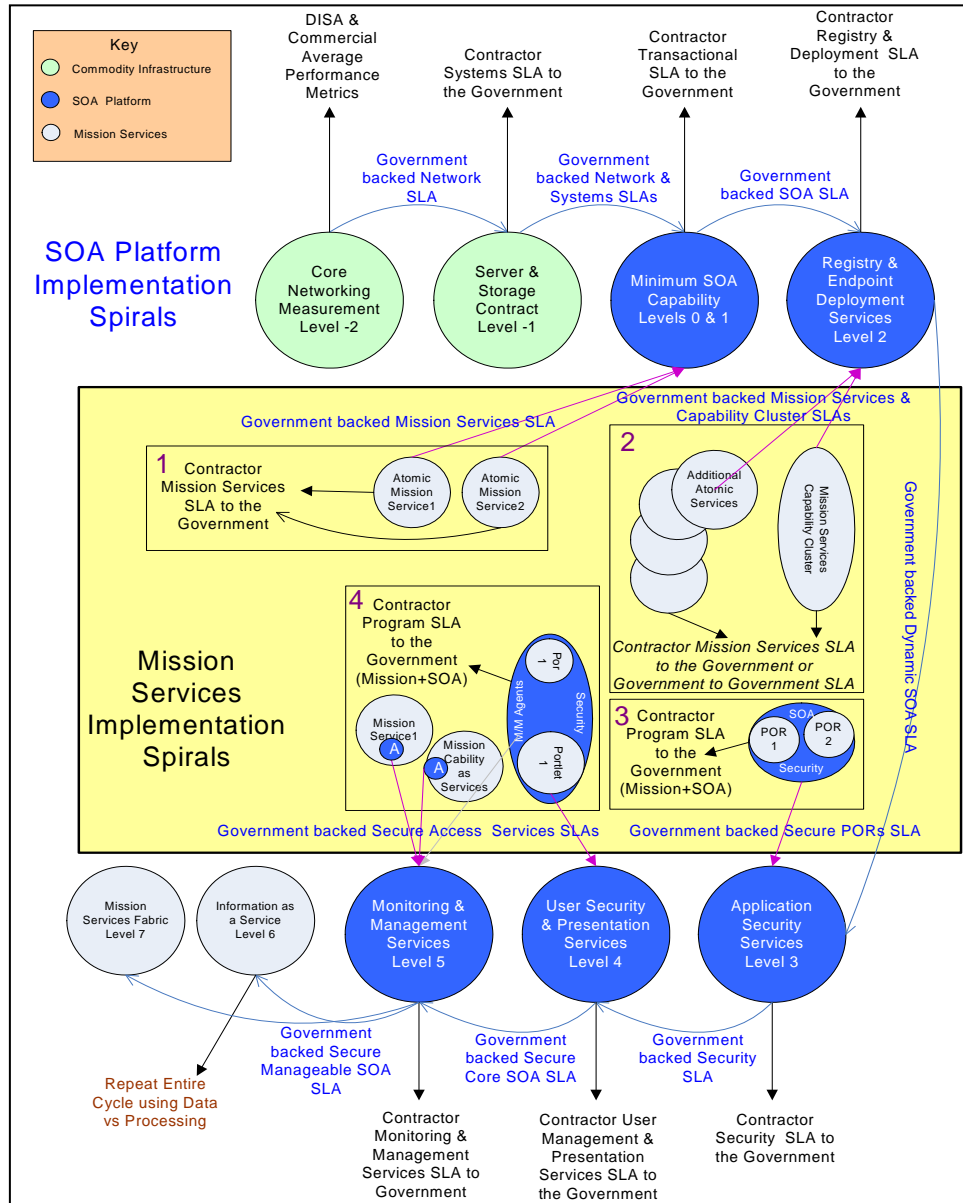


Figure A-2 Detailed SOA Infrastructure and Mission Services Implementation Sequence

In Figure A-2, the circles are associated with the steps or levels that they represent in Figure A-1. For example in Figure A-1, networking infrastructure is step or level -2; in Figure A-2, implementing core networking measurement is an apriori step (level -2) required before implementing a systems infrastructure or SOA services. In the center of Figure A-2, four mission service implementation spirals are shown.

Each of these spirals represents a phase of implementing a SOA system (e.g., spiral 1 implements minimum mission SOA services on top of levels 0 (Information Packaging and Delivery, the ESB) and 1 of the SOA infrastructure in Figure A-1). This distinction between SOA infrastructure services, which are required to deploy mission capabilities, and mission services is key to understand what to implement and when. Mission services should be deployed in the same spiral fashion as the SOA infrastructure services upon which they depend.

A Priori Spiral: Setting the Scene for SOA (Levels -2 and -1)

Level -2 and level -1 provide the baseline infrastructure support necessary to deploy SOA (Figure A-3). Measurements are needed to establish the baseline of the core network infrastructure. These measurements should be turned into a Government backed SLA to the systems provider, that is, an RSLA. The Government provides the contractor with a performance “guarantee” for the infrastructure upon which the contractor will depend. Likewise, the systems infrastructure (Level -1) contractor will provide an SLA to the Government for metrics such as server uptime, storage availability, and so on. The Government offers these metrics in an RSLA to the basic SOA contractor. The basic SOA contractor has assurances of networking and systems infrastructure performance and can now assure the Government that the SOA service basics will be available (given the RSLAs are not violated).

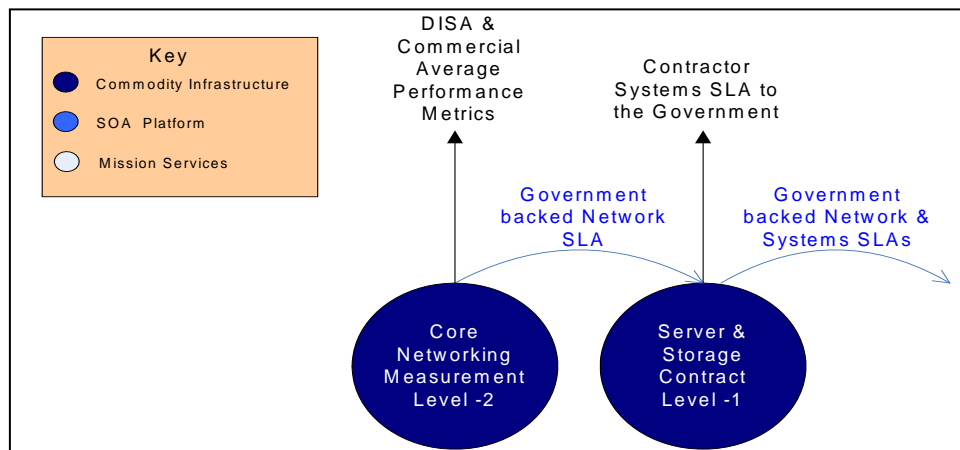


Figure A-3 Setting the Scene for SOA

Spirals 1 and 2: Establishing a Basic SOA (Levels 0, 1, and 2)

A basic SOA capability requires SOA platform services, such as Enterprise Service Bus (ESB) and Registry/Repository (R/R), as well as some mission services (Figure A-4). Mission services could be simple collect services like weather or aerial surveillance.

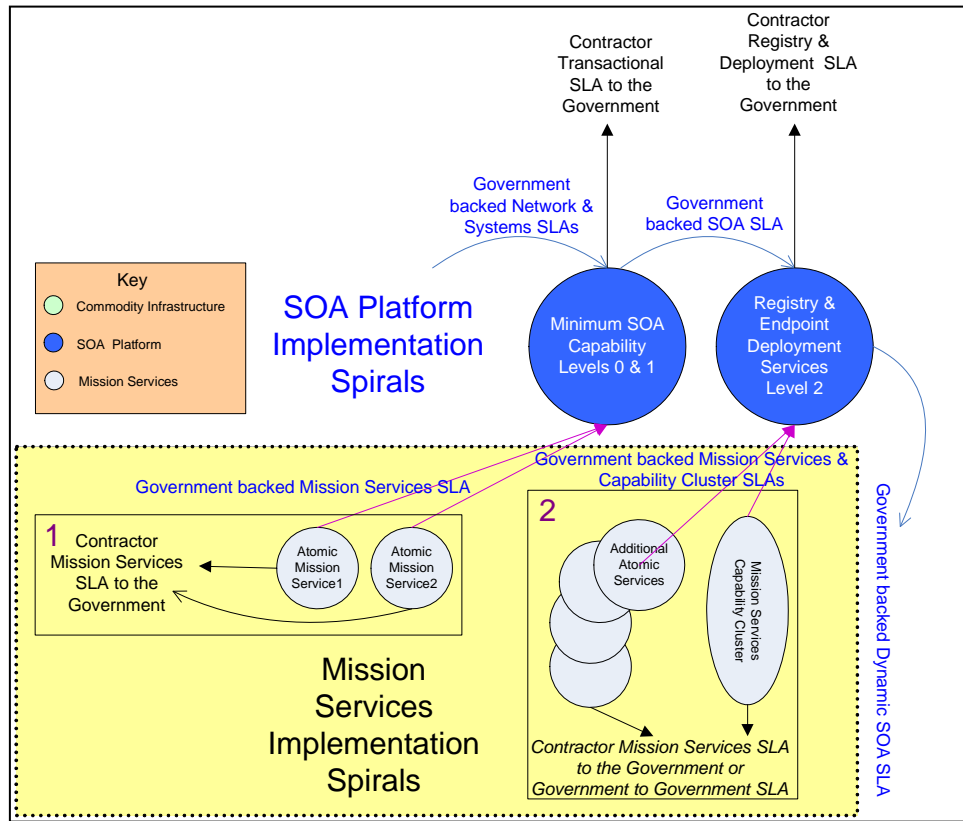


Figure A-4 Basic SOA Levels

Either of the weather or aerial surveillance mission services can be viewed as atomic services—they can stand on their own and have value to the warfighter even if they were not combined with other mission services. For levels 0, 1, and 2, the chain of RSLAs between the Government (representing the predecessor service providers) and successor service providers continues.

In this case, with SOA platform services, the R/R service provider requires that ESB services exist and satisfy their SLAs. Otherwise, the R/R service would fail. For example, consumers would not find the search service to be very useful if they searched for and found a service that they were unable to reach once the service was deployed. Similarly, if providers pushed new mission services to the R/R but the ESB did not deploy them, the R/R service fails.

Bringing on a few simple mission services at this point is a good way to understand how the minimal SOA platform services perform on the real networking and systems infrastructure. These services would have to be hard coded onto the ESB as the R/R service does not yet exist. Bringing on more services prior to learning these operational lessons may confound troubleshooting. Once the real characteristics of the networking and systems infrastructure are established, the expectations of the SOA contractor can be re-balanced to match the operational environment. At this point contractual terms

should be amended; this is the adjustment to the Government's consumption behavior that is essential to continuing the trust chain going forward.

Once the basic infrastructure is in place and the ESB (Level 0) has been proven out, then R/R services (Level 2) can be implemented. Examples of R/R services are:

- Search or find a service
- Lifecycle management of a service
- Grouping services together by functional capability or owning organization
- Service deployment

In order to exercise more complex SOA platform services it is necessary to have more complex mission services with which to develop, test, and deploy. Additional atomic, or simple, mission services are added as well as creating mission service clusters as indicated on the right in Figure A-4. Additional atomic services test out the scale of the R/R while mission service clusters test out the lifecycle management and ontology functions of the R/R.

Examples of additional atomic services may be contact alerting or mapping services. Examples of mission service clusters may be a set of services that when used together demonstrate a track management capability. This capability takes contacts the warfighter receives, performs the necessary functions to analyze them, and fuses them into a track that represents a vehicle on the move (e.g., vector of an airplane).

From this example, the desirability of managing a set of services as a group or capability can be understood. It may be useful to combine the set of services into an aggregate service called track management thereby hiding the complexity of the inner working from the warfighter. This aggregation is achieved by organizing mission services by their functional uses. Similarly, it can be understood how a particular set of mission services may be given to a group of mission service developers. Hence, it would be useful to manage them as "owned by" and managed by this group. An R/R with full lifecycle capability could provide for such ownership and it could provide for custody to pass to different groups as the set of mission services graduates through the service lifecycle. For example, ownership of the track management service(s) would pass from a development group to a separate test group as the service(s) moved from development to test. The notion of sets of services allows both functional and lifecycle groupings of services.

By bringing onboard these SOA platform and mission services, the Government can offer a 'Dynamic SOA SLA' to the contractor chosen to implement SOA levels 3, 4 and 5.

Spiral 3 and 4: Delivering a Fully Functioning SOA (Levels 3, 4 and 5)

Once core SOA services are deployed and working, it makes sense to wrap SOA security around them. Security services without actual SOA platform or mission services are not legitimate. If variations of SOA platform and mission services are not exercised, the viability of the SOA security services is not really known. Consequently, the prior steps are prerequisites to the on-boarding of SOA security as a SOA platform service.

Figure A-5 shows the mission services and SOA platform services included in the 'Fully Functioning SOA' sequence of steps. To exercise SOA security, it is important to allow requests to multiple PORs. It is necessary to interact with these PORs as part of a

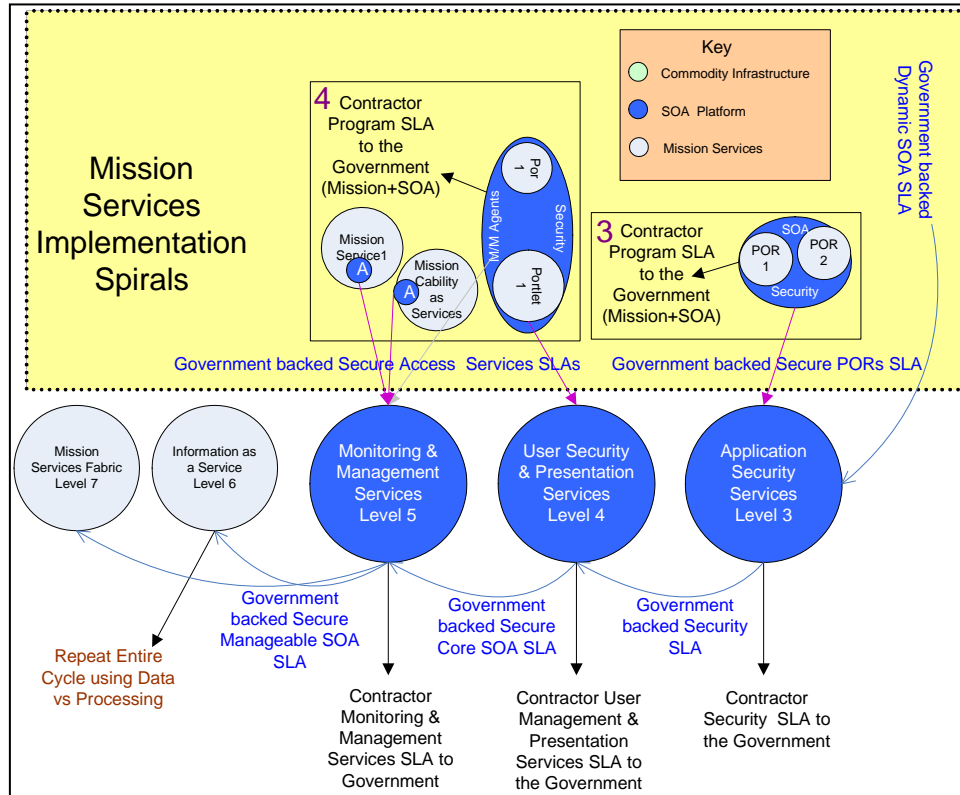


Figure A-5 Fully Functioning SOA Implementation Steps

separate SOA enclave with potentially its own SOA platform and its own SOA security implementation. SOA interoperability is feasible provided that SOA Security standards (e.g., WS-Trust, SAML) are followed.

Following the earlier examples, PORs might be Single Integrated Air Picture (SIAP) for track management clusters of services or Distributed Common Ground Station-Navy (DCGS-N) for targeting information. A full compliment of SOA security services should be exercised with the analysis (SIAP) and targeting (DCGS-N) PORs. This builds upon earlier examples of atomic mission services (e.g., weather or unprocessed aerial surveillance). It is good practice to move through the lifecycle of a warfare act, such as from ISR to analysis to command and control (C2), as mission services are brought onboard to test the operational viability and security of the underpinning SOA platform services (e.g., ESB, R/R).

The Importance of Re-Balancing Contracts

At each break point in the sequence, the SLAs/RSLAs, terms and conditions, and requirements are re-balanced to ensure that service providers within each progressive spiral have realistic, operational performance histories upon which to base their initial SLAs (which are likely to be revised later as well). Re-balancing of expectations by both the contractor and Government is essential to creating a workable SOA because interactions of technologies and mission systems will arise as the sequence of steps is executed. This is a virtue of SOA and the Agile Model allows for smarter choices to be made as the Government consumes each of the spirals. It also allows the Government to bring in new requirements at natural break points in the contracting lifecycle.

We are not suggesting that the Government go through the traditional waterfall procurement cycle, with its consequent delays, at each of the break points in the Agile Model. We are suggesting that the Government develop an omnibus SOA contracting vehicle and pre-qualify the vendors for the collection of all the spirals and levels. It is likely that certain vendors would qualify for certain spirals or levels based on past performance. Some vendors will qualify for multiple spirals. It is extremely unlikely that any single vendor would qualify to perform every spiral in entirety.

Once it has been demonstrated that machines can produce and consume mission services securely, a spiral with user facing services begins. User Security and Presentation Services will exercise different aspects of the SOA platform, e.g., Web 2.0, Portlets, LDAP, CRLs, X.509. Figure A-6 on the next page shows that additional mission capabilities will have to be brought on to exercise these SOA platform services.

Returning to the earlier example, a warfighter may request a weather report that is delivered in human readable format vs. a format for machine consumption. Or a soldier may request an intelligence product via a portlet that offers a menu of available imagery products.

Once the basic SOA functions are in place and user testing has occurred, it is time to bring on SOA monitoring and management (M&M) services. While it would be nice to have had these services from the outset for optimization and troubleshooting, M&M services are meant to cover the broad spectrum of SOA platform and mission services. SOA M&M services require greater SOA complexity to exist so they can monitor and manage security services, R/R services, POR services, and atomic mission services. If these other services go down, the M&M services provide a means to identify root cause of the failure.

M&M services also provide for threshold monitoring and initiation of proactive steps if a threshold has been breached, but not an SLA (yet). These M&M actions can be automatic, such as spawn a new cluster to handle load, or alert an administrator to inspect the source of difficulty.

Figure A-6 shows that M&M agents would need to be sprinkled across a variety of different mission and SOA platform services to fully exercise the compliment of M&M services. For example, if a targeting service becomes unavailable it is easy to understand why a warfighter would like an alert (to change his tactics) and why an administrator would as well (to exercise a fix).

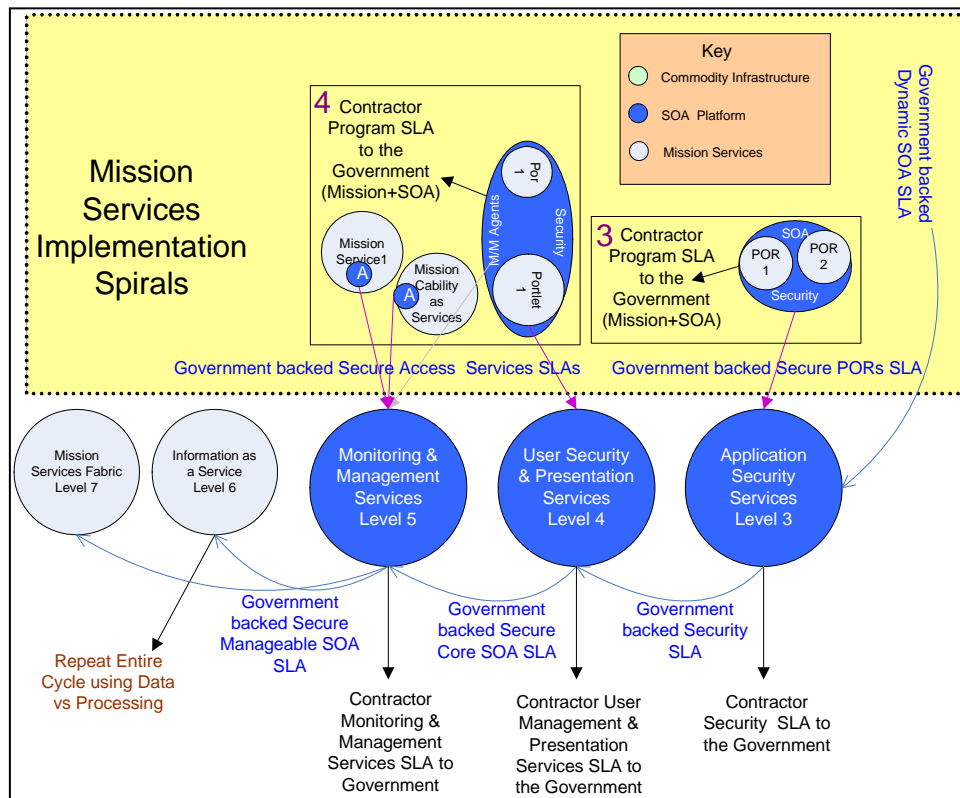


Figure A-6 Adding User-facing Services

Other SOA platform uses of M&M services are a bit more esoteric. For example, monitoring UDDI server availability to know if the 'Find a Mission Service' web service is up and running; or pinging the XML transform service to ensure that it is performing up to the SLAs. There are many good examples of why a robust Monitoring & Management service is important to a fully functioning SOA.

Later Spiral: Advanced SOA (Levels 6 and 7)

Data as a Service and Dynamic Composition of Orchestrations are two more advanced SOA capabilities whose implementations at present mostly occur in commercial marketplaces. However, it is easy to see how 'Data as a Service' could be useful in the ISR realm. Data as a Service could be brought in earlier in the spirals, but it is important to get basic SOA working prior to addressing advanced SOA. Similarly, Dynamic Orchestrations would be very useful in the ad hoc provisioning of mission threads to warfighters. Figure A-7 illustrates the relationship of these level 6 and 7 services with an SLA-backed Government infrastructure and services-oriented environment.

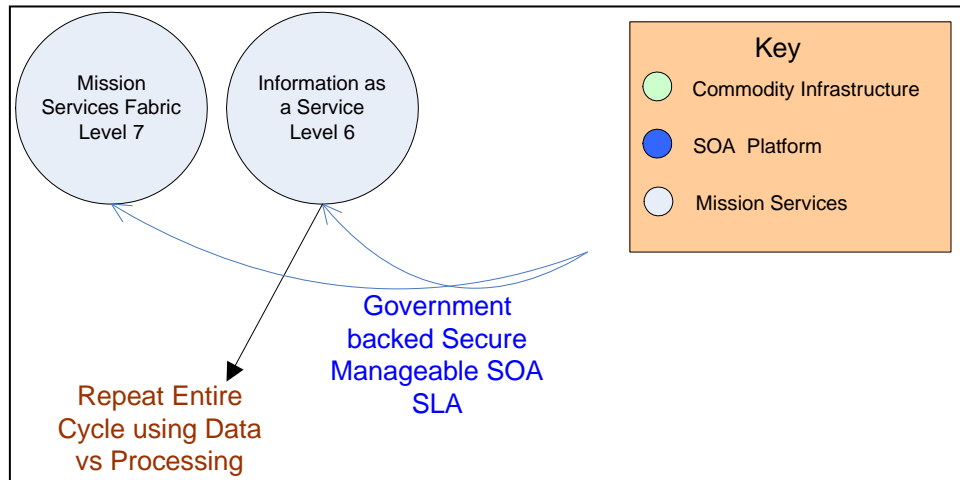


Figure A-7 Advanced SOA

Dynamic Orchestrations, require a Level 5 SOA in order to be implemented. Orchestration is an executable business process that can interact with both internal and external Web services (reference x). An orchestration defines the sequence and conditions in which one service invokes other services in order to accomplish its function. That is, an orchestration is the pattern of interactions that a service must follow in order to achieve its goal. Dynamic orchestration requires a fabric of mission services to exist that allow this composition of new processes based on existing services. Data as a Service and Dynamic orchestration are quite advanced and rely on complete understanding of the underlying infrastructure and existing services; hence, they are at the end of the spirals. We will save further discussion on implementing them for a subsequent paper while the Department works to acquire and implement basic SOA.

A.3 Relationship of Architecture & Engineering to Agile

Because the model laid out above is granular in nature it offers the Government several options vis a vis planning. The Government could have one contractor that does SOA platform architecture and design and a different contractor that does mission services architecture/design. It is good practice to separate these concerns. Within the SOA platform concerns, the Government could choose to have one contractor design the SOA basics (steps or levels 1 through 3 or 1 through 5) and have another do the advanced SOA platform architecture and design. As long as the steps are accomplished in a linear fashion, e.g., 1 through 3 versus 1, 3 and 5, then the earlier architect can have a clear plan for the later architect to follow. This is also true for architectures of mission systems that are exposed as services. Knowledge of desired mission threads and legacy systems is required to take existing, or newly developed mission services, and aggregate or orchestrate them to mission ends. Like SOA platforms, simple mission service architectures, e.g., mission service implementation spirals 1 and 2, could be done by a variety of contractors. However advanced mission services design would require a more sophisticated level of mission and mission systems knowledge.

A.4 Summary

The example in this Appendix shows the relationship between SOA infrastructure services and mission services. Important lessons learned are emphasized:

1. Implementing an entire SOA infrastructure at the outset of an SOA transformation has not worked well.
2. Implementing an SOA infrastructure to the exclusion of mission services has not worked well.
3. Assuming that the SOA is independent of the infrastructure will lead to confounding problems as implementers attempt to tease apart sources of SOA failures.

Hence the best practices approach is to:

1. Implement a bare bones SOA infrastructure with a few mission services to baseline the Government networking & systems infrastructure.
2. Develop Reverse SLAs that define the operating parameters of the Government infrastructure (upon which the SOA will ride).
3. As SLAs are developed, keep them coarse grained at the outset as the SOA infrastructure is raised and baselined. In later implementation spirals fine tune the SLAs.
4. Roll out a set of mission services that adequately exercise the successive SOA infrastructure spirals.
5. Cleanly define Government-to-contractor dependencies and contractor-to-contractor dependencies. Set incentives that encourage parties to work together to ensure dependencies work, e.g., joint incentives or sanctions.