Service Component-Based Architectures

Version 2.0

Architecture and Infrastructure Committee,
Federal Chief Information Officers Council
June 2004

The Federal CIO Council, Architecture and Infrastructure Committee

Components Subcommittee

in

Collaboration with the

Federal Enterprise Architecture Program Management Office

and

The Industry Advisory Council

Service Component-Based Architectures
Version 2.0:

June 2004

Contributors

In alphabetical order:

Tim Bass, Department of the Air Force

John Butler, Industry Advisory Council

Reynolds Cahoon, National Archives & Records Administration*

Harry Feely, Department of Education

Robert Haycock, Office of Management and Budget*

Roy Mabry, Department of Defense

David Mayo, Industry Advisory Council

John McManus, National Aeronautics and Space Administration

Patrick Mullen, Agency for International Development

Marion Royal, General Services Administration

Daud Santosa, Department of the Interior

*Co-Chairs, Federal CIO Council, Architecture and Infrastructure Committee, Components Subcommittee

Table of Contents

EXECUTIVE SUMMARY

To facilitate efforts to transform the federal government into one that is citizen-centered, results-oriented, and market-based, the Office of Management and Budget (OMB) is developing the Federal Enterprise Architecture (FEA), a business-based framework for government-wide improvement. As illustrated in Figure 1, the FEA is being constructed through a collection of interrelated "reference models" designed to facilitate cross-agency analysis and the identification of duplicative investments, gaps, and opportunities for collaboration within and across federal agencies.

This document, developed by the Components Subcommittee of the Federal CIO Council Architecture and Infrastructure Committee, compliments the SRM by discussing the concepts of service component-based architectures and placing them in the context of the FEA. The purpose of this paper is to inform agencies' thinking on development and use of enterprise architecture, in a manner consistent with component sharing and reuse, and the objectives of the FEA. The Subcommittee will evolve this document as agency enterprise architectures and the FEA mature.

Successful service-oriented, component-based architectures will greatly serve the needs of the Federal community as the community works to ensure information technology (IT) capabilities are managed in a structured enterprise-wide technology and business process lifecycle. Experience with component-based architectures has shown that reuse can be successful when the reuse efforts focus on large business level components in a collaborative environment that includes system owners, capital planners and enterprise architects.

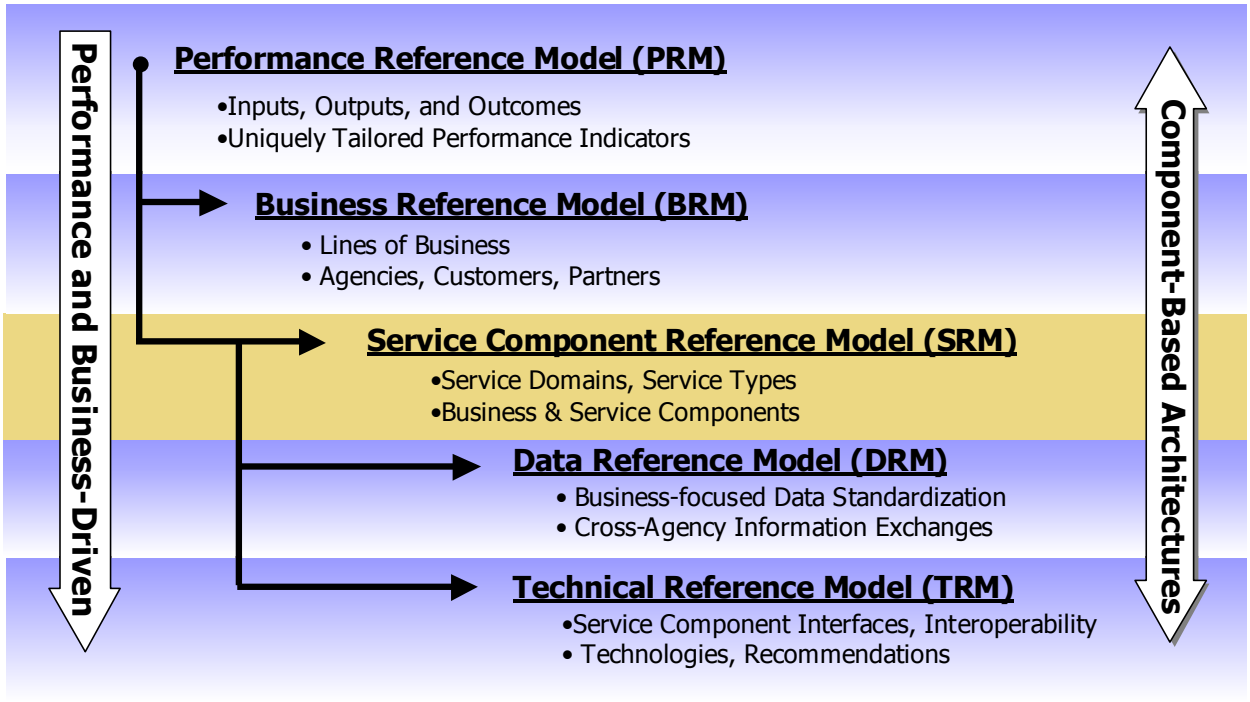


Figure 1 – The Federal Enterprise Architecture Reference Models

The FEA Service Component Reference Model (SRM) is intended for use in discovering government-wide business and application service components in IT investments and assets. It is a service component-based framework that provides—independent of business function and technology (to the extent possible)—a "leverage-able" foundation to support the reuse of business services, service components, and federated business systems.

TARGET AUDIENCE

Business leaders – focused on fielding systems that best support their mission and business needs and achieve the highest return on their IT investments

Capital Planners – responsible for defining and funding service components using an IT 300 Exhibit to support capital planning and investment control (CPIC) of federal projects or programs that may benefit from cross-agency collaboration and the re-use of agency assets. In accordance with OMB Circular A-11, Section 300, Federal agencies are required to submit to OMB an Exhibit 300 for all major IT investments.

Chief Architects – responsible for the definition and target planning of an Agency's Enterprise Architecture, working with a variety of architectural implementations (e.g., service-oriented architectures; FEA reference models; intergovernmental architectures, such as the National Association of CIO's Enterprise Architecture Development Tool-Kit, etc.).

System/Solution Architects – responsible for building / assembling service components that leverage existing capital assets and business services across the government and industry.

ORGANIZATION OF THE DOCUMENT

The remainder of this document is organized according to the following sections:

Section 1: **Introduction to Service Components** provides an overview and description of service components, component granularity, and linkages into the SRM.

Section 2: **Federal Service Components** provides a complete overview of a Federal Enterprise Component and sub-components. It discusses related concepts including business modeling, harvesting, provisioning, and service component collaboration/registry/repository.

Section 3: **Use and Maintenance** describes, at a high level, how the SRM should be used by the agencies in their EA and Capital Planning processes; how an online repository will facilitate these efforts; and how the SRM will be modified and updated to support the budget process.

Appendix:

The FEA provides a high-level overview of the Federal Enterprise Architecture, the Federal Enterprise Architecture - Program Management Office (FEA-PMO), and supporting committees.

Federal Enterprise Component Life-Cycle Perspectives provides an overview of various federal enterprise component life-cycle perspectives. The AIC Components Subcommittee is working to further refine these perspectives

Glossary defines terms used in the service component-based architecture white paper.

INTRODUCTION TO SERVICE COMPONENTS

BACKGROUND

The pros and cons of component-based architectures have been discussed and debated for some time. With the evolution of web-based technologies and the growing complexity of IT solutions, increased attention is being paid to the benefits of component-based development. Not surprisingly, the successful implementation of a component-based architecture requires the application of sound architecture principles, such as integration, interoperability, adaptability and the management of scale and flexibility.

The first step in a sound, component-based approach is to understand what a component is and what are its desired characteristics. The Federal CIO Council's Architecture and Infrastructure Committee commissioned the development of this White Paper to provide a basic understanding of how to describe and define a component. The paper represents the first attempt by the Components Subcommittee to reach a common understanding of this issue for both the Federal Government and its partners. It is the intent of this paper to initiate the discussion and mature the construct in order to promote component reuse and sharing, and enhance the collaboration opportunities that are available to all segments of the Government.

COMPONENT DEFINITION

The term "service *component*" can represent many things to many people. It can describe a complete business line such as U.S. Treasury's PAY.GOV, the U.S. Government's Central Contractor Registry (CCR), a business service supporting the validation of a Social Security Number, an application to support Content Management, or a capability that may be accessed through a technology or business interface. With multiple types of components available in industry and across governments, it became critical to the success of the SRM to define "service component" and to clarify the level of granularity that will reside within the SRM. We expand on the notion of component granularity to include return on investment (ROI).

A Component is defined as "a self contained business process or service with predetermined functionality that may be exposed through a business or technology interface."

SERVICE COMPONENT GRANULARITY

LEVEL	DEFINITION	ROI
Federated Business Component	A set of cooperating system-level components federated to resolve the business need of multiple end users often belonging to different organizations. Can be expressed as an IT 300 exhibit or a federation of IT 300 exhibits.	**High**

Business Component System	A set of cooperating business components assembled together to deliver a solution to a business problem. Can be expressed as an IT 300 exhibit.	**High**
Business Component	Represents the implementation of an autonomous business concept, business service, or business process. It consists of all the technology elements (i.e., software, hardware, data) necessary to express, implement, and deploy a given business concept as an autonomous, reusable element of a large information system. It is a unifying concept across the development lifecycle and the distribution tiers. Normally not expressed as an IT 300 exhibit, but as a sub-component of a larger business component system.	**Medium**
Distributed Component	The lowest level of component granularity. It is a software element that can be called at run-time with a clear interface and a clear separation between interface and implementation. It is autonomously deployable. Normally not expressed as an IT 300 exhibit. A distributed component provides low ROI for capital planning purposes.	**Low**
Language Class (n/a)	A class in an object-oriented programming language to build distributed components. This is NOT considered an SRM component. Normally not expressed as an IT 300 exhibit. A language class provides very low ROI for capital planning purposes.	**Very Low**

PRIORITIES FOR CAPITAL PLANNING PURPOSES

The table above illustrates the levels of granularity of service components and their priorities for capital planning investment ROI. The AIC Components Subcommittee believes that the greatest ROI is achievable when a service component can be expressed as a stand-alone IT 300 Exhibit.

The effective identification, assembly, and usage of service components allows for aggregate business services to be shared across agencies and governments. These business services provide the functionality and execution of business processes, which in turn sustain the FEA Business Reference Model (BRM) sub-functions. Service component aggregation will enable rapid building and implementation of components to support a given initiative or investment. Figure 4 illustrates the concept of aggregate services where multiple service components can support a business sub-function.

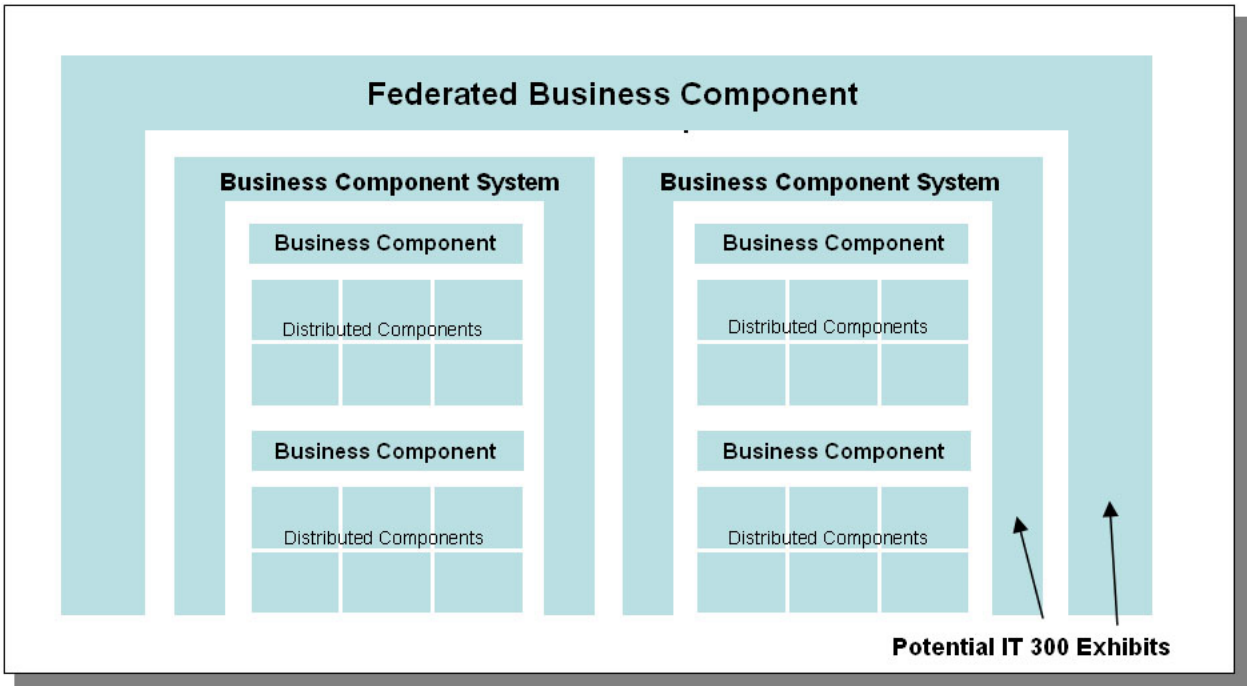


Figure 4 – Conceptual Hierarchy of Components

The SRM is decomposed into lower levels of granularity beginning from the process and application levels to the software and component and module levels. This level of decomposition provides various perspectives for stakeholders and solution architects to support the adoption of components and services within an IT initiative, asset, or investment.

DESIRED FEATURES OF A SERVICE COMPONENT

A successful service component-based architecture requires the application of sound architecture principles to the definition and composition of components. The components in the architecture should exhibit basic features, as discussed in the following section.

Encapsulation

A component should clearly separate the definition of the services that it provides from its implementation of those services. This implies that the internals of a "well-behaved" service component are hidden behind a contract between the component and the outside world.

Consumability

A component that is designated as the provider of certain services should be able to provide those services in a coherent and consistent manner to another software or business process. In other words, the component should provide those services without impeding the operations of the user. A true component can be independently developed, delivered, and installed without complex interdependencies on other external components.

Extensibility

A well-behaved service component should be extensible in both the services it provides and the way those services are provided within the component itself. A well-behaved component should be extensible to adapt to changes in the business while at the same time preserving the service provided to existing consumers.

Standards-Based

The value of a component increases in conjunction with the number of places the component is used. Standards, both technical and domain, affect this applicability in a number of ways. First, by basing the interface of a component on industry standard practices and technologies, the component is more likely to be reused. As an example, if a component is built using Cobol, reuse of that component in a .NET environment is relatively difficult and therefore is unlikely to occur. On the other hand, if a technology standard, such as WSDL, is used to create an interface for the component, it can then be used from either Java or .NET with equal ease and will therefore see greater ROI. Further, if the interface is based on a domain industry standard such as LegalXML, even greater ROI is likely since a consortium of organizations have "pre-agreed" to adhere to the standard.

The second reason standards help component reuse is that software components need a compatible execution environment. This means that the implementation also benefits from being standards based. For example, if a component is written to the Web Services Interface Standards, then it can be deployed in a fairly broad set of execution environments and therefore is more likely to be compatible.

Industry Best Practices and Patterns

A software component should embody industry "best practices" and patterns. Patterns are simply common solutions to recurring problems or issues faced in the software life cycle. Patterns typically reflect industry best practices—the convergence of approaches to solving problems. The use of patterns in components facilitates the understanding and consumption of the components.

Well Documented

A software component should be well documented to promote understanding of its capabilities and encourage its consumption. The documentation should permit architects, designers, and integrators to evaluate and consume the component. The documentation should include

models (preferably in UML; for example, use cases, class diagrams, and sequence diagrams) depicting the process and data capabilities of the component, user guides, functional overviews, and installation guides, as well as API documentation. A Test Harness should be delivered with the component to allow the consumer of the component to test each of the services or methods offered by it prior to consuming the component. If appropriate, the component should include the source code (for "white box" components) and a "management application" if the data managed by the component must be entered or updated independent of the consuming application. Finally, a component should be delivered with samples of consumption of the component to indicate how the component operates within an application environment.

Cohesive Set of Services

Components should be factored in such a way that they provide a cohesive set of services. Proper "packaging" of services makes the services easier to find and use. System developers and integrators are able to use just the right component for the need. Using components that offer too broad an array of services leads to bloated software and can result in bugs due to inadvertent use of features that are not appropriate. As an example, suppose a Party Management component included bank transaction processing and required a connection to an ACH system. While this may work for a banking system, further reuse would be limited. A system that does not need bank transaction processing, such as a case management application, would still require a connection to an ACH system. Creating appropriately factored service offerings will significantly increase the breadth of opportunities for reuse of a component.

Well-Defined and Broadly Available Licensing or SLA

A software component should be accompanied by a well-defined license or service-level agreement (SLA). The license or SLA defines the user's rights and responsibilities with respect to the component. In particular, the license or SLA should clearly articulate the intellectual property ownership for the component, the scope of usage permitted, the extent of any rights granted to modify the component or produce derived works, and the extent of any rights granted to redistribute the component. For COTS components, the copyright will usually reside with the original author of the component, but the rights to use, modify, and redistribute can vary widely.

To promote reuse of the component, the license or SLA terms should be sufficiently broad as to allow the component to be reused in contexts other than its first intended usage without having to renegotiate licensing terms. So, for example, a site- or organization-wide license would be more appropriate than a single-processor license.

FEDERAL SERVICE COMPONENTS

Business improvement and integration challenges are major concerns with the majority of large private businesses and government organizations. Enterprise architects and capital planners must collaborate to meet the demands of today's business objectives and challenges. To accomplish this, an enterprise architecture is responsible for supporting the development of new business capabilities and must be congruent with business capital planning. An architecture that provides for reuse of existing business services and rapid deployment of new business capabilities based on existing capital assets is often referred to as a service-oriented architecture (SOA).

Successful SOAs ensure that the service components and the business services that they describe are consistent and complete across the enterprise. Experience with SOA has shown that reuse is successful when the reuse efforts focus on harvesting the existing capabilities and capital assets within business lines and elements.

SERVICE COMPONENT DISCUSSION

The term "component" is often confusing when viewed from the FEA perspective. At the top of the hierarchy is a federation of business components. Federal business components contain multiple business processes or services that can be shared across agencies. At the lower levels are lower granularity service components that implement elements of the processes or services. Components at the lower levels are selected and integrated to build higher-level services. However, the purpose of this white paper is not to define a governance structure for all components in the recursive definition of components. For FEA purposes, we are primarily concerned with the optimum ROI achievable using SOA in the federal government and therefore focus on a component granularity that can be described by an IT 300 Exhibit for capital planning purposes. The attributes of a business or service component are normally very different from the attributes of a technology component. Relating the service component hierarchy to the current FEA reference models provides the linkage from the efforts at the lines of business level in the BRM and the supporting services in the SRM.

Service components, by design, separate the services they provide from the way those functions are implemented. This is true at all levels of the service component hierarchy; however for the optimum ROI to be achieved, there must be fiscal purpose and asset alignment to capital planning processes. In addition, the interface profile of a service component is critical to business integration, as it makes it possible to "hide" the internal details of the service component. This abstraction of a service component provides resource planners a common view or language to communicate with enterprise architects and program managers. It follows that the first step in an effective SOA implementation is to understand the definitions and desired characteristics of service components, both of which are discussed in the following sections.

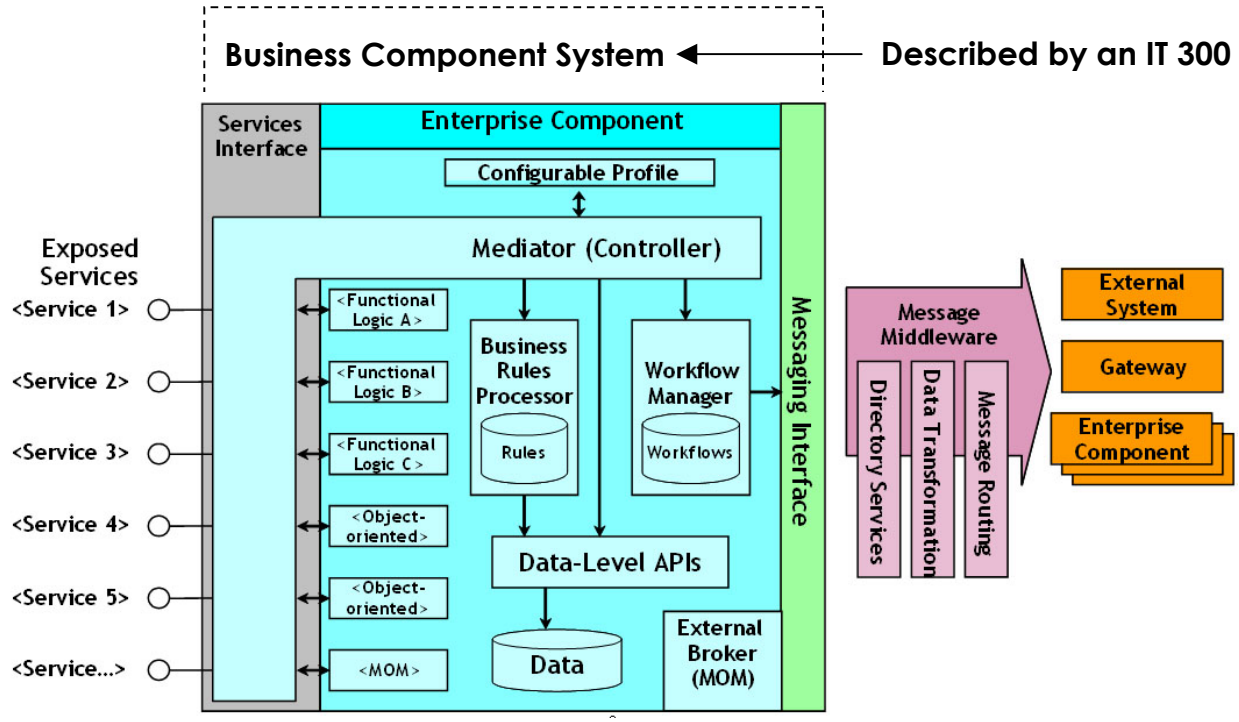


Figure 5 – A Notional Federal Enterprise Component

Figure 5 illustrates a notional Federal Enterprise Component that is also represented by a notional IT 300 Exhibit. This notional model can help facilitate the collaboration between capital planners, enterprise architects, and program managers. Federal Enterprise Components are defined in a top-down fashion by taking into account the nature and evolution of the business, its goals, conceptual model, process model, rules, and policies models. This concept is derived from the business-driven nature of the FEA BRM and the SRM. Service components defined in this fashion can be leveraged to build larger federated business components—components that can be used to resolve the business needs of multiple end users often belonging to different organizations. By exposing the business services via well-defined interfaces and encapsulating the internal technical details—including communication protocols, schema inconsistencies, data locations, etc.—enterprise components can provide services that span across the federal government.

SUB-COMPONENTS

The following notional sub-components, which could make up a Federal Enterprise Component, are technology implementation specific and are each assigned specific responsibilities:

- **Services interface**—the set of published services that the component supports. These are aligned with the business services outlined in the SRM.

- **Interface profile**—the sub-component that provides the ability to customize the component for various uses. The profile can be tailored to suit different deployment architectures well as the different sets of business rules across enterprises. The interface profile can specify the business rules and workflow that are to be executed when the component is initialized. The profile can specify the architectural pattern that complements the service component.
- **Mediator (controller)—**the sub-component that oversees the coordination processing carried out by one or more components. The mediator calls on other elements of the component that are required to satisfy the service request.
- **Business rules processor**—the sub-component that manages and executes the set of complex business rules that represent the core business activity supported by the component.
- **Workflow manager**—the sub-component that enables one component to access services on other components to complete its own processing. The workflow manager determines which external component services must be executed and manages the order of service execution.
- **Data-level application programming interfaces**—the services internal to the service component that support access to the data of record maintained within the service component. These services may span numerous distributed data sources.
- **Data**—Factual or numerical business information of record that is maintained by the service component. The encapsulated service component is fully responsible for maintaining this information.
- **Messaging interface**—the linkage from the service component to various external software modules (component, external systems, gateways, etc.) and other service components.

COMPONENT ARCHITECTURE DESCRIPTION STANDARDS

An effective service component-based architecture requires the adoption of and adherence to technical standards that help promote a common understanding of the use case of service components in an effective manner. A key goal in describing service component architectures is to provide a clear separation of concerns between the functional and the operational aspects of the architecture. A unified modeling language-based notation, as illustrated in Figure 6, is one way to describe a service component. These descriptions also form the basis of the work products or artifacts that are essential to the solutions lifecycle. The functional aspect generally addresses the following:

- Static structure of service components (application and technical), including service component interfaces
- Dynamic behavior of service components (collaboration)

The operational aspect often addresses the following:

- Static system topology (hardware, locations, etc.), including the following:
  - What runs where (placement)
  - Service-level characteristics (performance, availability, etc.)
  - Management and operation of the IT system (backup, recovery, capacity planning, etc.)

- Dynamic end-to-end behavior of the IT system (walkthroughs)

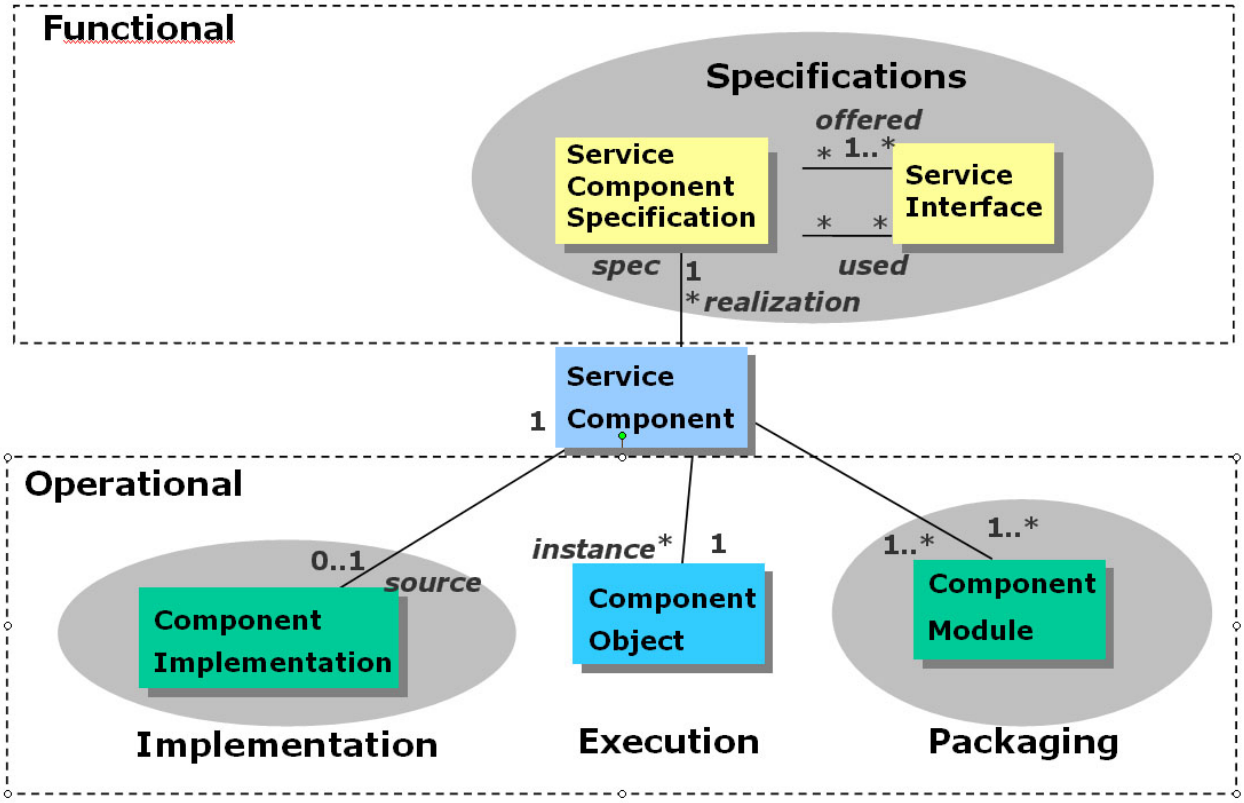


Figure 6 – Notional UML Service Component Description

FEDERAL SERVICE COMPONENT HARVESTING AND PROVISIONING MODEL

The FEA reference models are designed to be used to identify and define reusable service components and service component interfaces, and the definition of the federal service component granularity roughly corresponds to the level of granularity of an IT 300 Exhibit. Reference models are particularly used for business analysis and capital planning. Figure 7 illustrates the notional relationships of reference models to typical SOA activities.

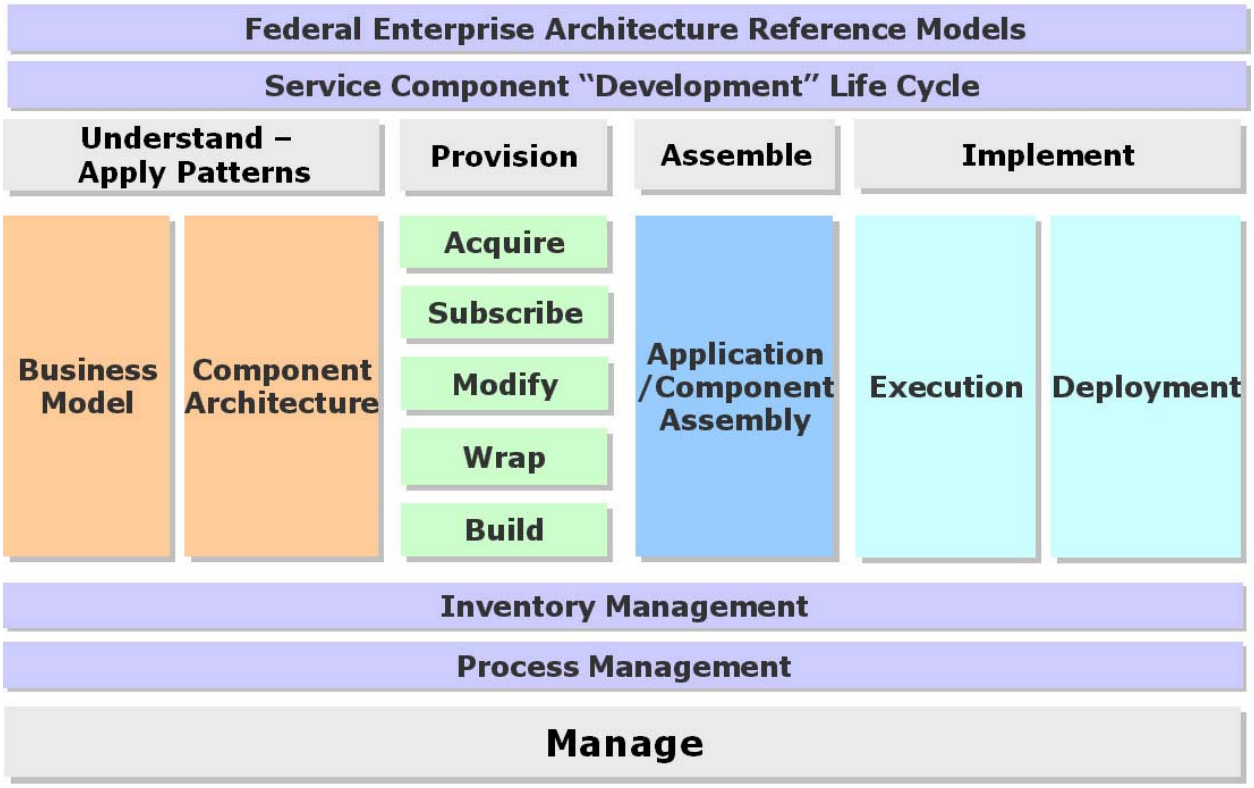


Figure 7 – Notional SOA and Interaction with the FEA Reference Models

Business Modeling

SOA activities start with business modeling, with the BRM and SRM acting as the source for structuring capabilities. The BRM and SRM are designed to bridge the gap between capital planners and enterprise architects, allowing expression of the business services and concepts that are required to support the operation of the organization.

E-business patterns identified in the BRM and SRM are also valuable assets that can be leveraged at the modeling stage to identify potential reuse patterns. The process of applying patterns begins by considering business drivers, then mapping the business pattern to application and runtime patterns and technologies. Business and integration patterns reflect the requirements and are independent of application and infrastructure topologies. Business and integra-

tion patterns are sufficiently general that they can be used for both the current and planned architectures. On the other hand, the application and runtime patterns put a stake in the ground that affect application and infrastructure design.

Applying the e-business patterns (selecting which patterns fit best) occurs in steps that parallel the layered asset model shown in Figure 8.

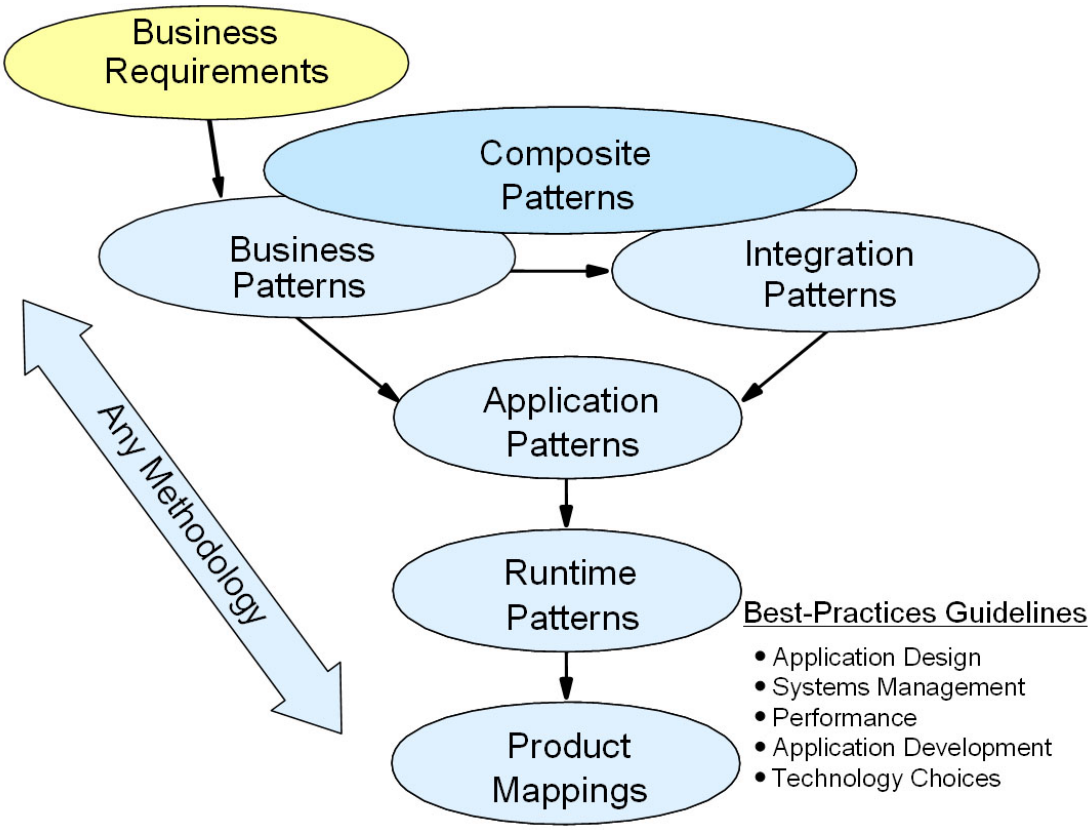


Figure 8 – Layered Asset Model

Pattern selection for component-based architectures begins with identifying the business patterns that apply to the business processes under consideration. Simple implementations may involve only one business pattern, but it is more common to find that several business patterns apply. Depending on the nature of the business requirements, process collaboration and orchestration patterns may also be needed to extend and support the business patterns.

Composite patterns represent frequently occurring combinations of business and collaboration patterns. A less frequently occurring combination of business and collaboration patterns constitutes a custom design.

Service Component Architecture

The content gathered during the business modeling phase is communicated via work products or artifacts to formulate the solution outline. At this macro-level design phase, the solution may be communicated via high-level service component models to develop a better understanding of the business domain and the solution. Service components are defined and the dependencies between collaborating service components are clearly identified. The business services exposed by these service component interfaces support the operational needs of the organization. Non-functional requirements are also taken into account to determine the constraints that must be applied to the solution. The business services are typically defined in a technology-neutral reference model, like the BRM or SRM. Again, e-business patterns are leveraged to identify existing applicable business patterns. The solution outline is followed by the provisioning, assembly, and orchestration of service components.

This staged, model-based approach can help insure both the consistency of the resulting solution and the traceability of the resulting service components back to the analysis-level of business concepts from which they were derived.

Component Provisioning and Assembly

It is also helpful to consider business solution assembly and service component provisioning working in parallel as indicated in Figure 9. This is often called "twin track development."

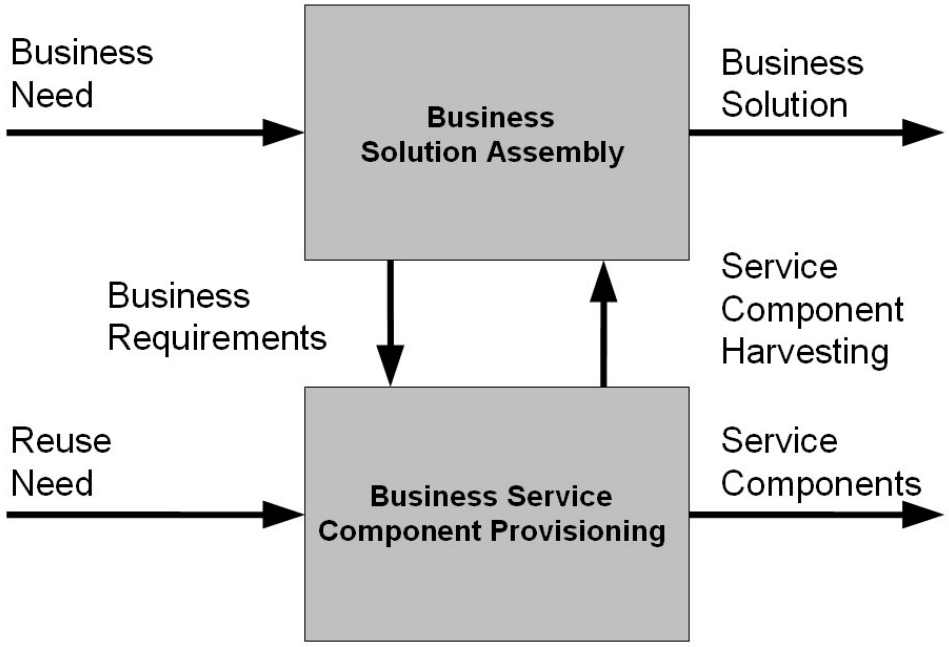


Figure 9 – Twin Track Development

Service component and asset consumers and producers follow separate processes geared to their respective needs of creating or harvesting business solutions and high-quality service components. Solution developers (asset consumers) examine existing service components and assets to potentially harvest them as reusable assets. Component producers (provisioners), on the other hand, look at requirements to be the basis for creating components or acquiring service components for reuse. Capital planners work to insure that the optimum ROI is achieved from the existing capital asset base.

The service component provisioning step may be used to determine whether the architecture calls for using preexisting service components or adopting a custom build of the required service component(s). There is a wide range of options to address service component provisioning, from extending existing frameworks, purchasing service components, and wrapping existing service components to adapting or even acquiring a subscription to a component service. A Fit-Gap analysis is a common technique, which can be applied to help define the nature of the required service components. A Fit-Gap Analysis strives to look at components within the context of requirements and makes a determination as to the suitability of the service component. Figure 10 illustrates this concept.

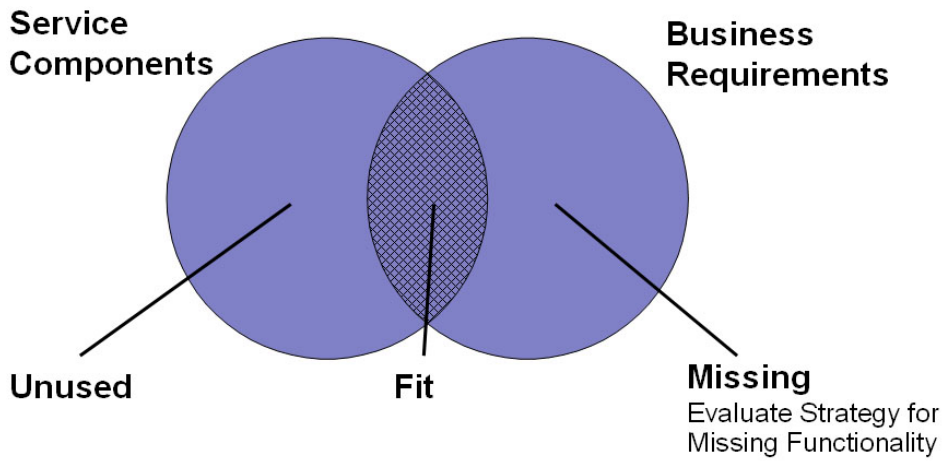


Figure 10 – Service Component "Fit Gap" Analysis

FEDERAL ENTERPRISE COMPONENT REGISTRY/REPOSITORY

After potential federal enterprise components have been identified, it is important to have a way for enterprise architects to manage the service component interface profile, publish the profile, and provide the methodology for accessing the component and other key component information. In addition, it is important to be able to understand the strategic, tactical, and operational attributes of the service component, including IT 300 Exhibit and fiscal information. In addition, when large federated system components are harvested, it is important for enterprise architects to understand the architectural pattern, or "blueprint," as to how the published service component is designed to interact with other service components. Equally important are the relationships between service components that are being used, because a failure in one

component used by many federal agencies could result in a cascading catastrophic system failure of business processes that depend on the functionality of that service component.

Because of the complexities associated with service component-based architectures and SOA, it is very important that mission- or business-critical federal components be property certified. In order to have a scaleable certification process for federal components, it is important to establish the criteria for service component certification. It follows that the registry/repository concept for Federal components will also be based on a hierarchical certification governance process.

For example, the certification process for service components will be different based on the component provisioning model. Acquired federal components would be certified differently from how developed service components are certified. Shared software service components may also be subject to a different certification criteria. These concepts are currently under development by the AIC Components Subcommittee and complement the concepts in this paper.

The FEA reference models are designed to be used to identify and define reusable service components. Service components, by design, separate the services they provide from the way those functions are implemented. This is true at all levels of the service component hierarchy. Service component-based architectures, if properly implemented, provide a framework to achieve a very high ROI for federal IT assets. The FEA SRM serves as the foundation for federal enterprise architects and capital planners to better serve the citizen by utilizing a service component-based architectural approach to federal IT asset management.

THE SRM

Definition

The SRM is a service component-based framework that can provide—independent of business function—a "leverageable" foundation for reuse of applications, application capabilities, components, and business services.

Purpose

The SRM serves to identify and classify horizontal and vertical service components that support Federal agencies and their IT investments and assets. The model will aid in recommending service capabilities to support the reuse of business components and services across the federal government.

Specifically, the SRM was created to:

- provide a framework that identifies service components and their relationships to the technology architecture of agencies across the federal government
- classify, categorize, and recommend components for the reuse of business services and capabilities across the federal government
- define existing service components that may be leveraged outside agency boundaries

- align and leverage existing federal guidance and application/architecture recommendations
- support e-Government initiatives
- evolve based on new services and components as they are discovered across industry and federal markets

Development of the SRM

In developing the SRM, the FEA-PMO leveraged previous federal architecture efforts, such as the Federal Enterprise Architecture Framework (FEAF) guidance and agency application reference models, as starting points for designing the government-wide model. Using these architectures as a point of departure, the FEA-PMO performed extensive research on industry and government application capabilities to provide a capabilities frame of reference for agencies to use.

The information contained within these sources provides concise and thorough documentation of the many services and capabilities that industry and government applications and IT investments perform. The FEA-PMO used this information to normalize and categorize service capabilities and components that support, through IT assets, the business of the Federal Government. A hierarchical structure of service domains, service types, and service components was crafted to convey a high-level categorization of capabilities. Definitions were applied to the seven service domains, the 29 service types, and the 168 supporting service components.

.

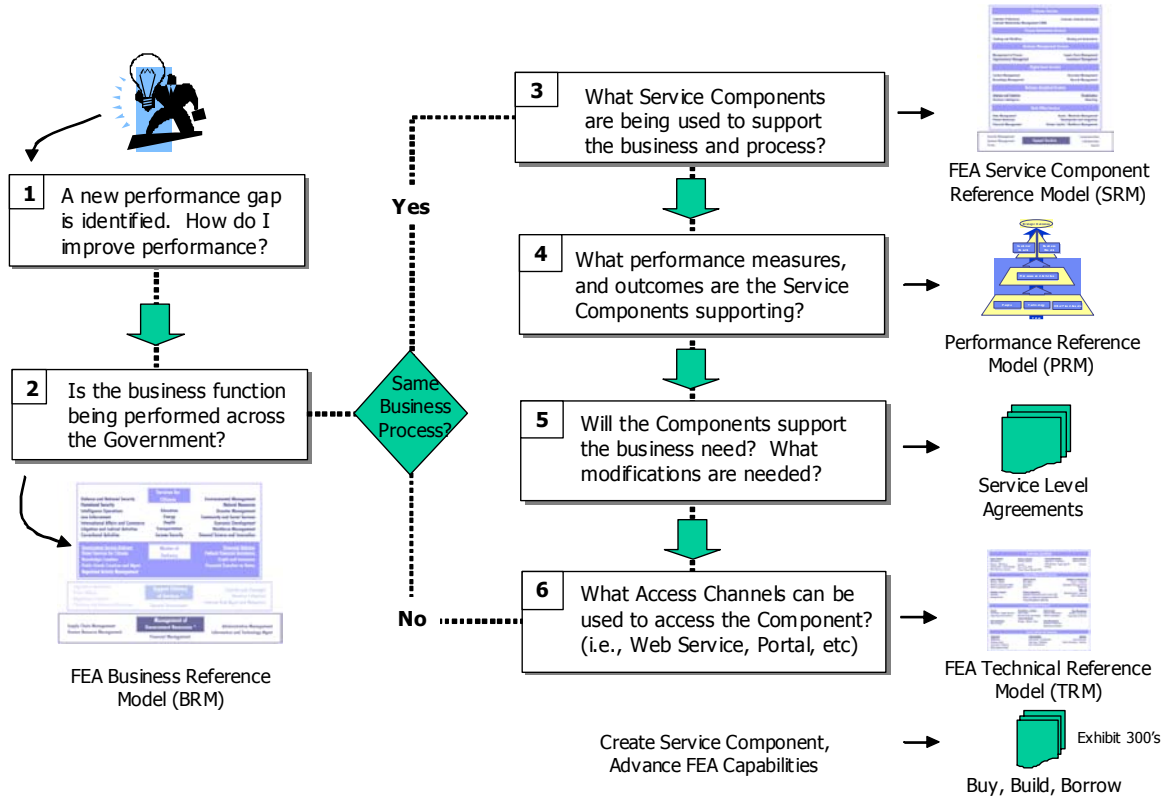
USE AND MAINTENANCE

The FEA and SRM are intended for use in analyzing investments in IT and other capital assets. As agencies plan for their IT capital investments, they will be able to access the FEA to identify:

- agencies that are building or have already built similar service components and capabilities
- agencies that are already collecting or plan to collect similar data
- suitable technologies already being used elsewhere, in support of service components
- potential collaboration partners to jointly resource a project

An illustration of the utility of the FEA and SRM are presented in Figure 11.

FIGURE 11 – FEA / SRM USE CASE EXAMPLE



The FEA will provide agencies with a powerful tool to investigate alternatives to costly (and potentially duplicative) IT capital investments *up front* and before a significant expenditure of resources. Reciprocally, OMB will be using the FEA to ensure that proposed agency IT investments are not duplicative and to analyze the architecture throughout the year to identify opportunities for cross-agency collaboration. As such, the FEA will help ensure that the federal government eliminates redundant capital investments and that agencies save time and money by leveraging reusable business processes, data stores, and IT service components.

The alignment and relationship of the SRM to agency enterprise architectures is one of the next steps towards implementing the SOA model across the federal government. Aligning the layers of the FEA Technical Reference Model (TRM) and SRM to agency technology, business (process or activity), and application architectures, enables the categorization of an agency's IT investments, assets, and infrastructure by the common definition and purpose of the service specifications and service components in the TRM and SRM, respectively.

THE FEDERAL ENTERPRISE ARCHITECTURE MANAGEMENT SYSTEM (FEAMS)

FEA analysis and maintenance are greatly facilitated through the use of an Internet-based automated EA repository and analysis tool—the Federal Enterprise Architecture Management System (FEAMS). Agencies will be given access to FEAMS and can use it in both capital planning and architecture development efforts.

In addition to storing the FEA reference models, as shown conceptually in Figure 12, FEAMS will include general information on agencies' IT initiatives. Initiative alignment to the BRM Lines of Business that they support, the service components and technology that these components leverage, and the performance metrics that they use in achieving performance objectives will be presented. It is OMB's goal that FEAMS will eventually include information on all of the appropriate capital assets in which federal agencies invest.

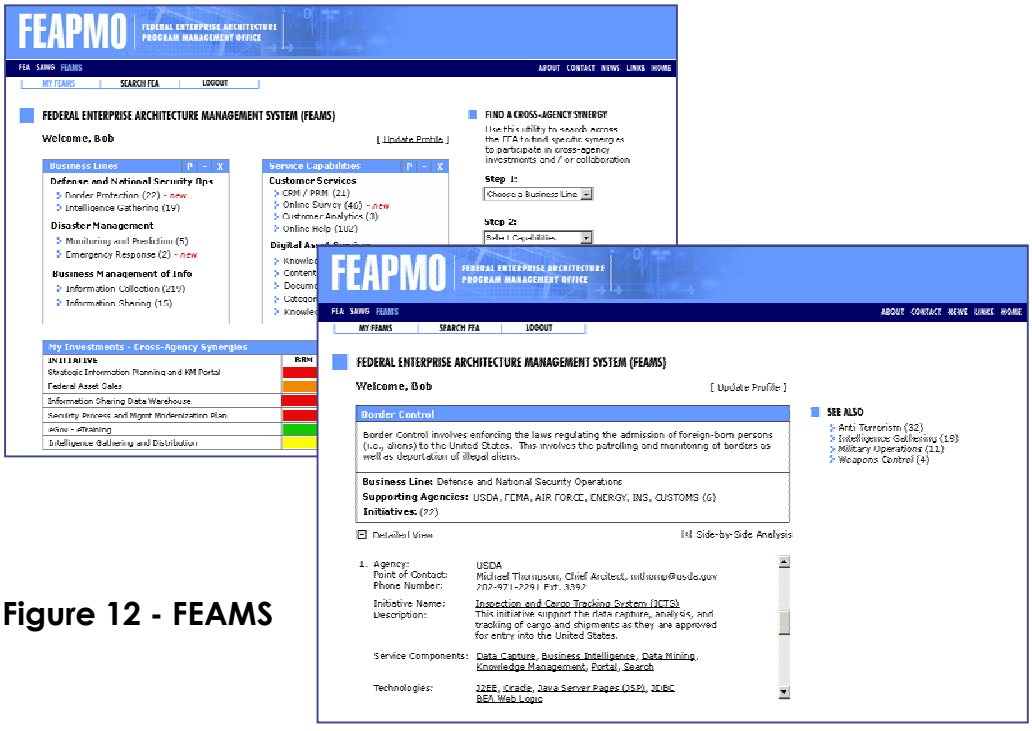


Figure 12 - FEAMS

The FEA, including the SRM, is being released to federal agencies through the FEA-PMO website, http://www.feapmo.gov. The website provides Agencies with downloadable access to the SRM in multiple electronic formats—PDF, Word, and XML. FEAMS will advance these capabilities by providing agency representatives the ability to search across FEA reference models to determine the availability of services and components they may be able to reuse as well as data and information that they may be able to share.

SERVICE COMPONENT REGISTRY/REPOSITORY AND COLLABORATION

The creation of a component repository and registry for service components is envisioned to be one of the tangible, ongoing outcomes of the FEA analysis at the service, technology, and data layers. As reusable service components are identified and harvested, and collaboration between agencies begins to take place, there will be the need for a collaboration-based repository for storing, maintaining, and sharing these service components. In addition, it is critical that federal service components are referenced to strategic, tactical, and operational information including budgeting and congressional appropriations information, including IT 300 Exhibit information.

The AIC Component Subcommittee will establish a service component registry that will be accessible over a secure extranet to initiative owners and the managing partners of e-Government initiatives. For instance, this registry might provide an area in which users can find, evaluate, share, download, and rate service components, as shown in Figure 13, as well as a directory of business functions that the service component supports. The component registry might provide the latest research and analysis surrounding the selection and recommendation of third-party/industry components that are supported by the service component-based architecture specifications.

Figure 13 – Notional Federal Component Registry and Collaboration Tool

The component repository will support collaboration and the rapid discovery of capital assets. Whenever possible, the sharing of these service components will be accomplished through standards-based, reusable, secure, portable, and interoperable technology. Service Level Agreements between the partnering agencies will help build the understanding for implementation and usage of these components. The required collaboration can be performed in a notional registry/repository, like the www.core.gov example in Figure 13.

The establishment of a component registry is widely accepted as a means by which organizations can leverage the knowledge and intellectual property across public, state and local industries. For instance, organizations, such as the National Association of State CIOs (NASCIO), have partnered with commercial companies to create the National Software Component Exchange (NSCE) for state and local governments. Other exchanges and component-service organizations offer similar services that should be leveraged when making the decision to partner and/or build a similar solution.

APPENDIX

THE FEA

The FEA is a business- and performance-based set of reference models for cross-agency, government-wide improvement. The lack of an FEA to support cross-agency collaboration was cited by the 2001 Quicksilver E-Government Task Force as a key barrier to the success of the 24 Presidential Priority E-Government Initiatives approved by the President's Management Council in October 2001.

The purpose of the FEA is to identify opportunities to simplify processes, reuse federal IT investments and unify work across the agencies and within the lines of business of the federal government. The outcome of this effort will be a more citizen-centered, customer-focused government that maximizes technology investments to better achieve mission outcomes.

Collectively, the FEA reference models will provide universal definitions and constructs of the business, performance, and technology of the federal government. The reference models will serve as a foundation to leverage existing processes, capabilities, components and technologies as government agencies build target enterprise architectures. They are designed to facilitate cross-agency analysis and the identification of duplicative investments, gaps, and opportunities for collaboration within and across federal agencies.

OMB established the FEA Program Management Office (PMO) in February 2002 to define and develop the FEA. For additional information on the FEA reference models, please refer to the FEA-PMO website, located at www.feapmo.gov.

THE ARCHITECTURE AND INFRASTRUCTURE COMMITTEE (AIC) – COMPONENTS SUBCOMMITTEE

The AIC Components Subcommittee was established to foster the identification, maturation, use, and reuse of service component-based architectures and architectural components in the federal government. The underlying objectives are to foster the basic principles of interoperability, reusability, and portability of processes, services, and service components by federal agencies and related partners and stakeholders as they transform their business processes through data sharing, e-government automation, and improved information systems.

The efforts of the Components Subcommittee will be directed toward achieving these outcomes:

- identification of business processes, service components, and technologies for reuse through analysis of the FEA Service Component and Technical Reference Models
- reduction of IT costs for federal agencies achieved through the reuse of business processes, service components, and technologies
- rapid solution development through the reuse of service components
- rapid integration of disparate business services

- development and implementation of e-government solutions based on component-based architectures

FEDERAL ENTERPRISE COMPONENT LIFE-CYCLE PERSPECTIVES

The AIC Components Subcommittee, in collaboration with the AIC Emerging Technology Sub-committee, identified numerous federal enterprise component life-cycle perspectives. The conclusion of this collaboration, supported by the Federal CIO AIC leadership, was that there is not a single federal enterprise component life-cycle perspective, but numerous perspectives depending on the view of the stakeholder. This appendix highlights a few perspectives identified by the AIC. These perspectives will be further refined and consolidated in a future AIC white paper that specifically addresses the federal enterprise component life-cycle.

Federal Enterprise Component Harvesting Perspective

In the federal harvesting perspective, federal enterprise components are harvested from executive agencies incubating or cultivating service-components. The example in this section illustrates harvesting and certifying the Central Contractor Registry as a federal enterprise component. This perspective, like the others in this appendix, is a work-in-progress.

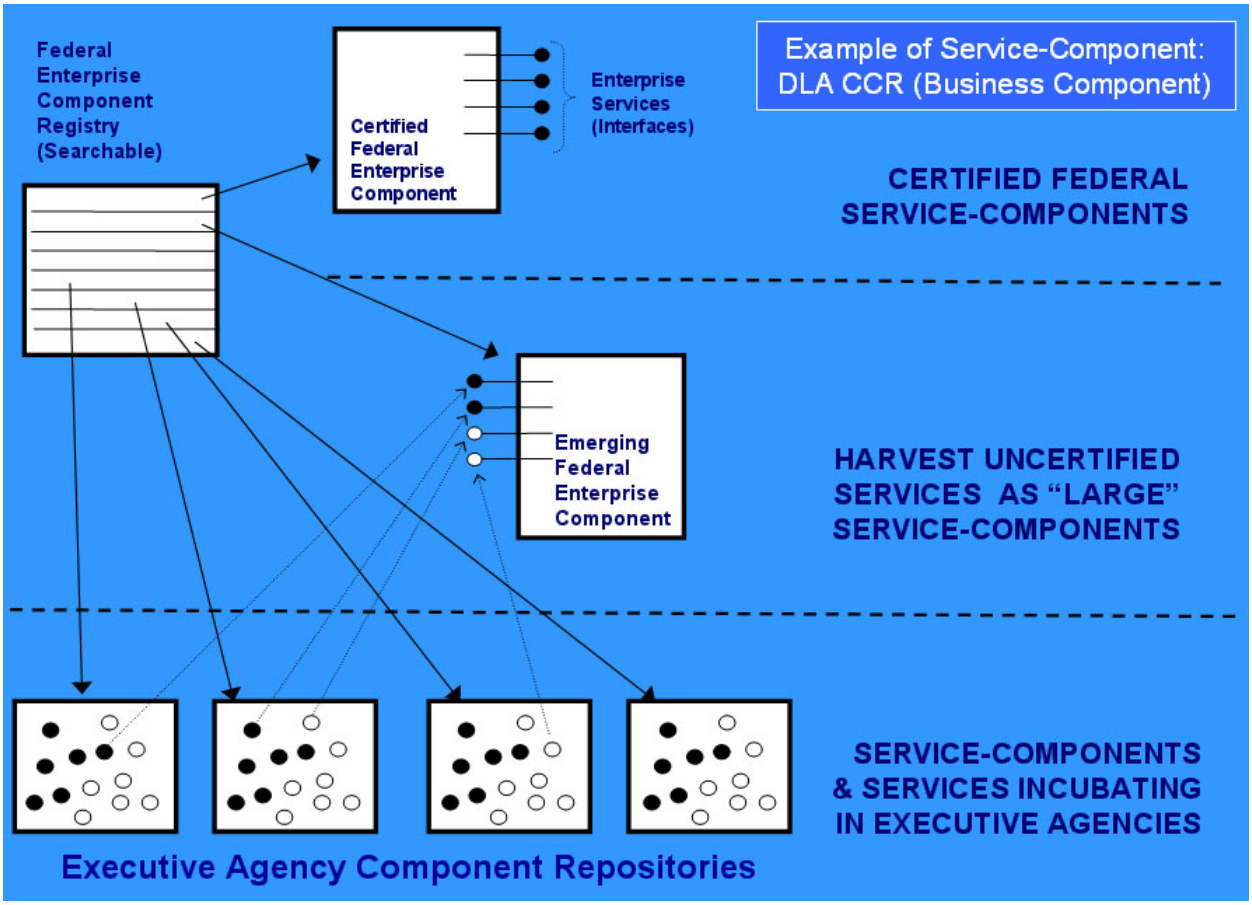


Figure A1 – Perspective: Harvesting Federal Enterprise Components

There are numerous proposed harvesting models for federal enterprise components. This perspective, showing the harvesting concept in a relationship workflow between departments, agencies, the AIC and OMB, like the others in this appendix, is a work-in-progress.

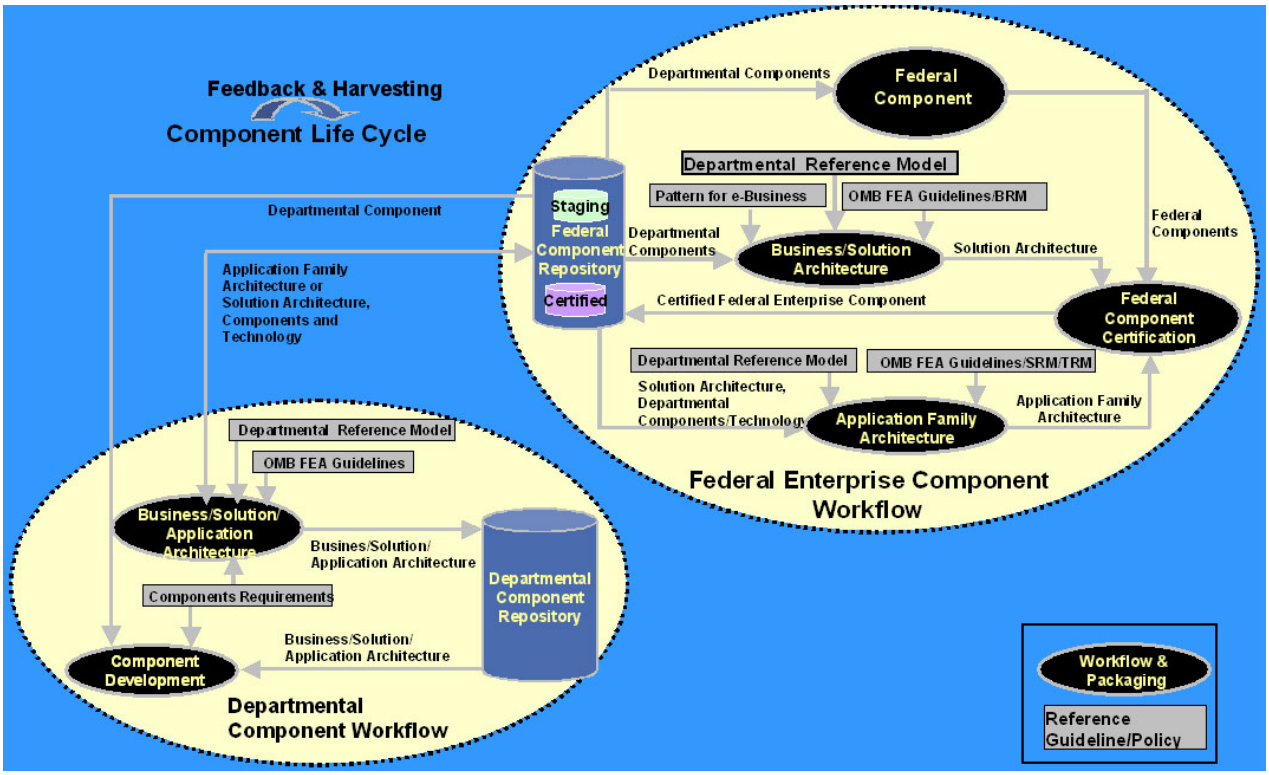


Figure A2 – Perspective: Harvesting Federal Enterprise Components

Federal Enterprise Component Emerging Technology Perspective

The AIC Emerging Technology Subcommittee is working on a component life-cycle perspective for new technologies. This perspective, like the others in this appendix, is a work-in-progress.

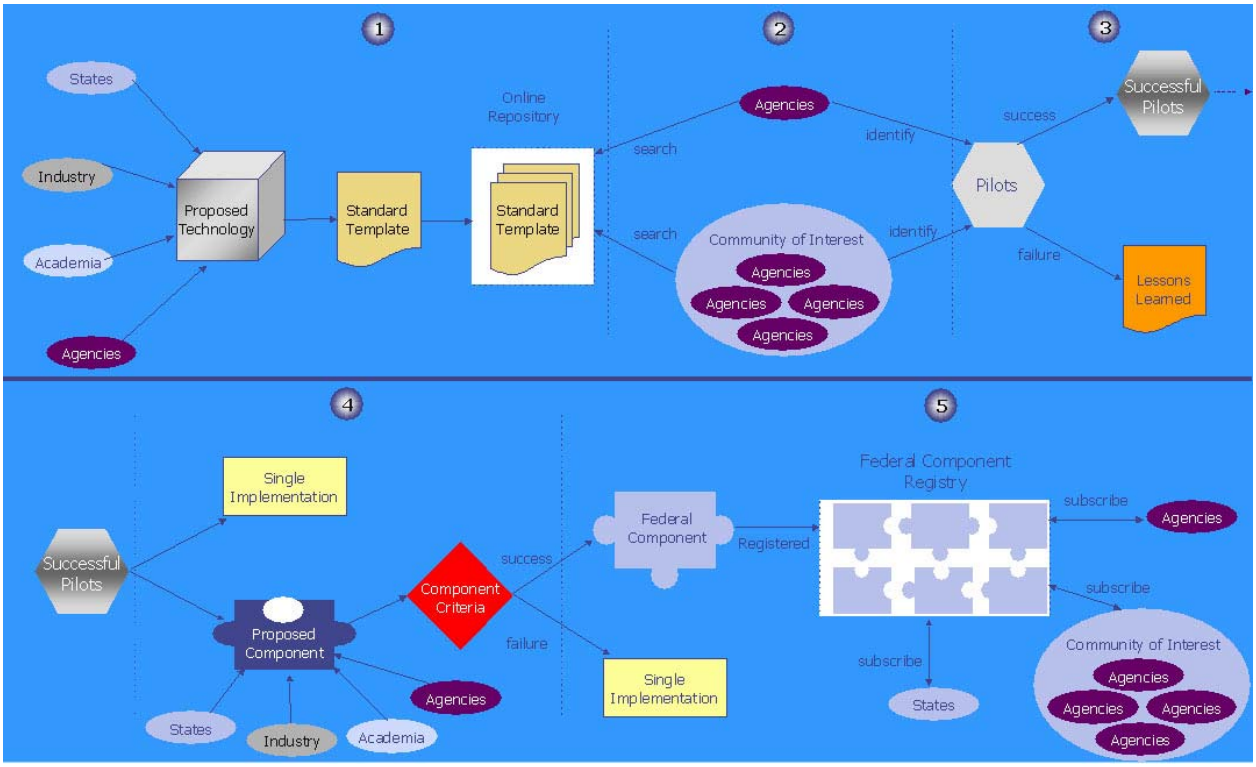


Figure A3 – Perspective: Component Life-Cycle of Emerging Technologies

Federal Enterprise Component Web Services Perspective

The AIC Emerging Technology Subcommittee is considering federal component life-cycle perspectives based on Web Services technologies. This perspective, like the others in this appendix, is a work-in-progress.

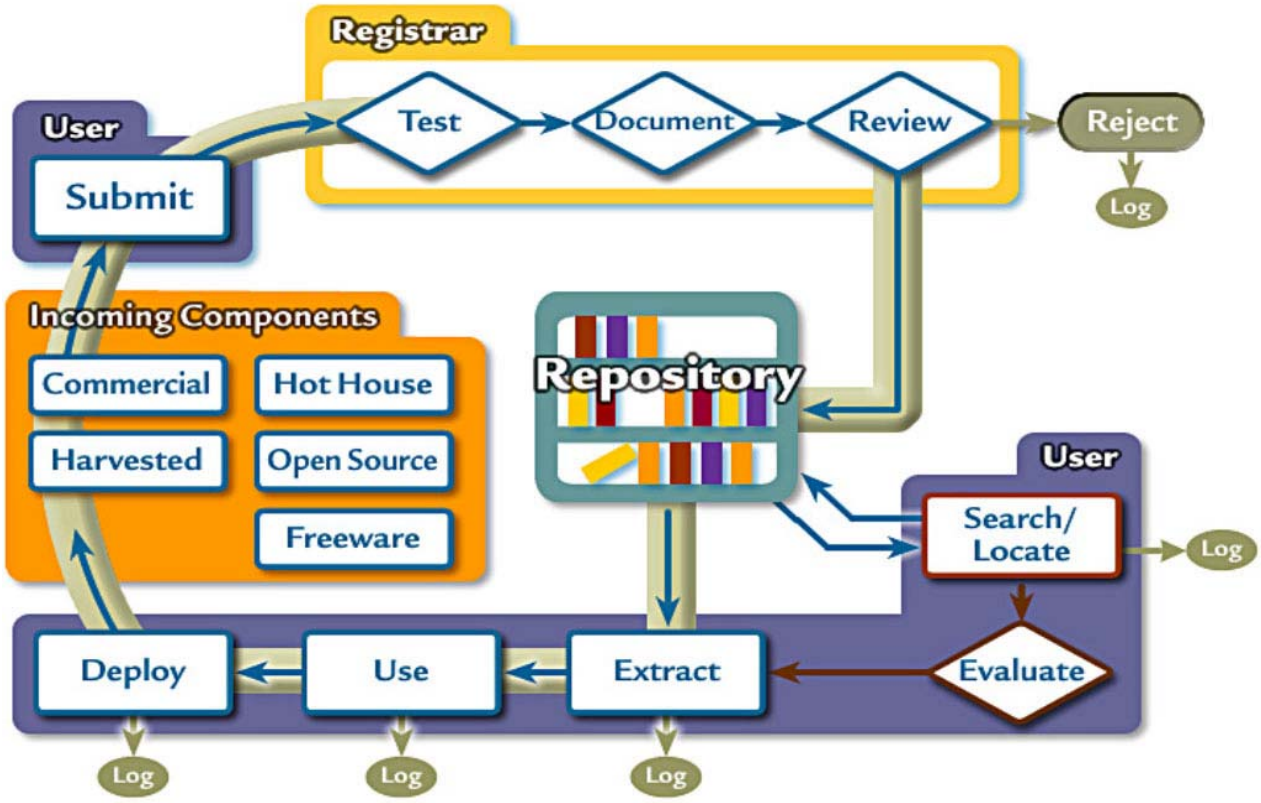


Figure A4 – Perspective: Web Service Component Lifecycle

Federal Enterprise Component AIC Subcommittees Life-Cycle Perspective

The AIC Components Subcommittee is working on a federal component life-cycle perspective that includes workflow processes across the AIC and in collaboration with OMB and federal agencies. This perspective, like the others in this appendix, is a work-in-progress.

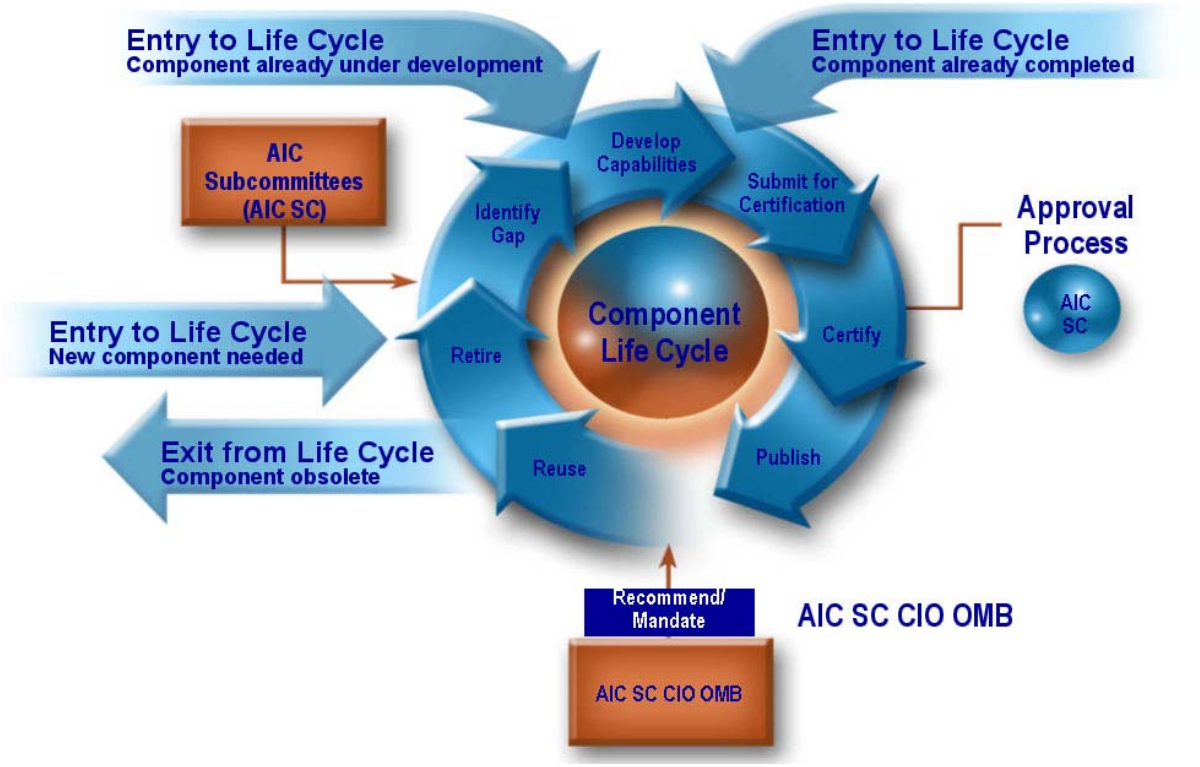


Figure A5 – Perspective: AIC Component Life-Cycle and Work Flow

GLOSSARY

Architecture: Representation of the structure of a system that describes the constituents of the system and how they interact with each other.

Application Architecture: Representation of an application and its parts, their inter-relationships and functions.

Business Component: Represents the implementation of an autonomous business concept, business service, or business process. It consists of all the technology elements (i.e., software, hardware, data) necessary to express, implement, and deploy a given business concept as an autonomous, reusable element of a large information system. It is a unifying concept across the development lifecycle and the distribution tiers. Normally not expressed as an IT 300 Exhibit, but as a sub-component of a larger business component system.

Business (Domain) Component: Organizational unit that offers business services operation based on rules of that business.

Business Component System: Set of cooperating business components assembled together to deliver a solution to a business problem. Can be expressed as an IT 300 Exhibit.

Business Logic Component: Software unit that offers small-grained business logic that has a large degree of reuse throughout the organization. Sub-components that manage and execute the set of complex business rules that represent the core business activity supported by the component.

Component: Independently deployable unit of software that exposes its functionality through a set of services accessed via well-defined interfaces. A component is based on a component standard, is described by a specification, and has an implementation. Components can be assembled to create applications or larger-grained components. For the purposes of this paper, a "component" is defined to be an entity that is roughly congruent to an IT 300 Exhibit.

Component Architecture: Internal structure of a component described in terms of partitioning and relationships between individual internal units.

Component-Based Architecture: Architecture process that enables the design of enterprise solutions using large service components. The focus of the architecture may be a specific project or the entire enterprise. This architecture provides a plan of what needs to be built and an overview of what has been built already.

Component Registry: Application designed to provide a directory of available components based on profile and or specification. Registries usually provide efficient mechanisms for searching for components in multiple ways, such as by service, price, and/or provider.

Component Repository: Application designed to store component specifications and implementations. Often provides facilities to efficiently search for and retrieve components for evaluation against desired component specifications though the search capabilities may be off-loaded to a component registry.

COTS Components: Commercial Off the Shelf (COTS) components that can satisfy business process and data requirements for large functional domains or lines-of-business. Examples of COTS components would be Enterprise Resource Planning (ERP) products such as those as offered by commercial software companies.

Data: Factual or numerical business information of record that is maintained by the service component. The encapsulated service component is fully responsible for maintaining this information.

Data-Level Application Programming Interfaces: Services internal to the service component that support access to the data of record maintained within the service component. These services may span numerous distributed data sources.

Distributed Component: Lowest level of component granularity. It is a software element that can be called at run-time with a clear interface and a clear separation between interface and implementation. It is autonomously deployable. Normally not expressed as an IT 300 Exhibit. A distributed component provides low ROI for capital planning purposes.

E-Business Patterns: Patterns for e-business are a group of proven reusable assets that can be used to increase the speed of developing and deploying net-centric applications, like Web-based applications.

Encapsulation: Hiding implementation details within a component so that an implementation is not dependent on those details.

Enterprise Architecture: Meta-architecture of an organization or the sum of all architectures within an organization.

Enterprise Component: Large-granularity business component of an organization.

Extensibility: Ability to extend the capability of a component so that it handles additional needs of a particular implementation.

Federated Business Component: Set of cooperating system-level components federated to resolve the business need of multiple end users often belonging to different organizations. Can be expressed as an IT 300 Exhibit or a federation of IT 300 Exhibits.

Federal Enterprise Component: Very coarse-grained business component of the U.S. Federal Government.

Fit-Gap Analysis: Examination of components within the context of requirements and to make a determination as to the suitability of the service component.

Component Granularity: The size of the unit of component under consideration in some context. The term generally refers to the level of detail at which component is considered, e.g. "You can specify the granularity for this service component".

Hierarchical Foundation: A foundation based on successive levels or layers.

Infrastructure Component: Software unit that provides application functionality not related to business functionality, such as error/message handling, audit trails, or security.

Interface: Mechanism by which a component describes what it does and provides access to its services. This is important because it represents the "contract" between the supplier of services and the consumer of the services.

Intellectual Property: A product of the intellect that has commercial value, including copyrighted property such as literary or artistic works, and ideational property, such as patents, appellations of origin, business methods, and industrial processes.

Interface Profile: the sub-component that provides the ability to customize the component for various uses. The profile can be tailored to suit different deployment architectures well as different sets of business rules across enterprises. The interface profile can specify the business rules and workflow that are to be executed when the component is initialized. The profile can specify the architectural pattern that complements the service component.

IT 300 Exhibit: OMB's A-11 requires submission of a business case for all major investments, including but not limited to information technology. Section 300 specifically describes the requirements for the business case, which is referred to as the Exhibit 300. The IT 300 Exhibit is submitted in the Fall and again with the President's budget and contains extensive set of screening questions and project management reporting requirements.

Language Class: Class in an object-oriented programming language to build distributed components. This is NOT considered an SRM component. Normally not expressed as an IT 300 Exhibit. A language class provides very low ROI for capital planning purposes.

Line of Business: A particular kind of commercial or government enterprise; e.g. "human resources" "financial management" "wholesale banking".

Mediator (Controller): Sub-component that oversees the coordination processing carried out by one or more components. The mediator calls on other elements of the component that are required to satisfy the service request.

Messaging Interface: Linkage from the service component to various external software modules (component, external systems, gateways, etc.) and other service components.

Notional Component: Set of services packaged into a component, derived from requirements definition. A "desired" component, prior to implementation.

Presidential Priority E-Government Initiatives: 24 e-business goals approved by the President's Management Council in October 2001.

Process Component: Software unit that implements the logic of a process.

Reuse: Any use of a preexisting software artifact (component, specification, etc.) in a context different from that in which it was created.

Service: Discrete unit of functionality that can be requested (provided a set of preconditions is met), performs one or more operations (typically applying business rules and accessing a database), and returns a set of results to the requester. Completion of a service always leaves business and data integrity intact.

Service-Component: Modularized service-based applications that package and process together service interfaces with associated business logic into a single cohesive conceptual module. Aim of a service component is to raise the level of abstraction in software services by modularizing synthesized service functionality and by facilitating service reuse, service extension, specialization and service inheritance.

Service-Component Reference Model (SRM): Service component-based framework that can provide—independent of business function—a "leverage-able" foundation for reuse of applications, application capabilities, components, and business services.

Service Interface: Set of published services that the component supports. These are aligned with the business services outlined in the service reference model.

Service-Level Agreement: A contract or memorandum of agreement between a service provider and a customer that specifies, usually in measurable terms, what services the service provider will furnish. Information technology departments in major enterprises have adopted the idea of writing a service level agreement so that services for their customers (users in other departments within the enterprise) can be measured, justified, and perhaps compared with those of external (sourcing) service providers.

Service-Oriented Architecture: Architecture that provides for reuse of existing business services and rapid deployment of new business capabilities based on existing capital assets.

Services Interface: A logical boundary that permits software services to be defined independent of the service implementation.

Solution Assembly: Process of implementing a solution by assembling the necessary components into a complete solution. This process often involves additional "glue" code to integrate the assembled components.

Test Harness: Software that automates the software engineering testing process to test the software as thoroughly as possible before using it on a real application.

Web Service: Functionality provided by a service, which is exposed using the Internet (XML, TCP/IP) as the transport mechanism. Can be internally provided as part of a suite of services or can be offered by external organizations.

Workflow Manager: Sub-component that enables one component to access services on other components to complete its own processing. The workflow manager determines which external component services must be executed and manages the order of service execution.

Wrapping: Creation of an interface around legacy functionality (code) that exposes the functionality as services via interfaces that conform to a component specification.

WSDL: An XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information. The operations and messages are described abstractly, and then bound to a concrete network protocol and message format to define an endpoint. Related concrete endpoints are combined into abstract endpoints (services).