Treasury Board of Canada Secretariat
Secrétariat du Conseil du Canada

Government of Canada

Service Oriented Architecture

Series

Primer

Chief Information Officer Branch
Enterprise Architecture and Standards Division

March 2006

Draft Version 0.6

Document History

Version	Date	Participant	Role	Comment
0.1 – 0.4	Dec. 22, 2005	Mike Giovinazzo	Author	Initial Outline, draft content and alignment with latest Statement of Direction
0.5	Jan. 13, 2006	Wesley McGregor	Author	Additional content and refinement
0.5B	Feb 23, 2006	Richard Bryson	Contributor / Reviewer	Review and incorporate concepts and comments
0.6	Mar 14, 2006	Wesley McGregor	Editor	Final edits and incorporation of B. Murray edits

TBS RDIMS # 379503

Contacts

Treasury Board Secretariat, CIO Branch

Wesley McGregor
Enterprise Architect, Government of Canada
Enterprise Architecture and Standards Division
mcgregor.wesley@tbs-sct.gc.ca

Richard Bryson
Senior Director, Architecture Domains
Enterprise Architecture and Standards Division
bryson.richard@tbs-sct.gc.ca

Gary Doucet
Chief Architect, Government of Canada
Executive Director
Enterprise Architecture and Standards Division
doucet.gary@tbs-sct.gc.ca

Table of Contents

Table of Figures

1.0 Preface

The Chief Information Officer Branch (CIOB) of the Treasury Board Secretariat is providing guidance on the adoption of Service Oriented Architecture for the Government of Canada (GC SOA). The goal of this guidance is to have departments adopt a common approach to the emerging standards and models of Service Oriented Architecture. SOA is a useful approach to GC program design, strategic business planning as well as systems design; it is not simply a technological advancement. It is also a means of achieving the GC goals for service modernization, horizontal service delivery and greater interoperability.

This document is part of a series related to the GC SOA. Collectively the GC SOA series provides CIOB's overall direction on behalf of the Government of Canada with regards to the adoption of Service Orientation.

2.0 About this Document

The Government of Canada is in the midst of a transformation on a scale never seen before. This government wide renewal process has consequences in both the public facing program areas as well as in internal service delivery. Simply put, the Government of Canada wishes to engage its citizens more fully while at the same time being more cost effective with the resources it currently has and consumes.

To support these goals, the Chief Architect and the Chief Information Officer of the Government of Canada recognize the need for a service-based whole-of-government approach to a business, information and technological infrastructure that supports the vision of a “service oriented” Federal Government. For instance, the newly instantiated Service Canada service bureau clearly encompasses such an approach not only in name but also by design. The services that Service Canada provides are the public facing extension of the services offered by many other federal government departments. The advent of Service Canada will allow citizens to see their government as a seamless portfolio of services that reinforce one another and together combine to deliver the outcomes that meet their needs.

“Service Orientation” is a powerful paradigm for organizing and utilizing distributed capabilities that might be under the control of different organizational owners. In a service-oriented environment, organizations make resources available to a community as independent services that can be accessed in a standardized way. But to successfully achieve a service-oriented vision, there must be a consistent way of defining and implementing services across the federal government. A proven way to do this is to create them like a set of interoperable business components that can be flexibly mixed-and-matched to cost effectively achieve the desired outcomes and deliver on the many mandates of government.

The expression "*Government of Canada Service Oriented Architecture*" (GC SOA) is the term CIOB is using to introduce a series of concepts that will help make it possible to deliver on this promise in a practical manner. To some audiences, in particular the more technically literate readers, it might appear that this GC SOA is akin to any typical *Service Oriented Architecture* (SOA as it is widely known in the Information Technology industry). While there is a definite draw from SOA concepts, TBS sees the GC SOA as being different in two key areas: context and application.

SOA is widely accepted as a best practice for information and technological design. The GC SOA however embraces a broader landscape and is the result of augmenting the technical concepts to also incorporate the business levels. Hence the GC SOA focuses beyond the usual technology realms of SOA.

Lastly, the GC SOA is also more precise in its application and includes specific guidance and flavouring as appropriate to the Canadian federal government helping to ensure that the government delivers its services in a more agile, interoperable, effective and efficient manner.

2.1 Purpose

The purpose of this document is to provide a foundational overview of the GC SOA so all readers have a common and consistent understanding of the topic. Upon reading this document, readers should have an appreciation of SOA in general, why the Government of Canada is pursuing this strategy and appreciate the business focus and specific layers of the GC SOA that sets it apart from typical SOA models.

2.2 Audience

This document is intended for general audiences and provides a high level view of the GC SOA. This document is a prerequisite for all additional guides and references related to the topic of GC Service Oriented Architecture. This document does not go into the level of technical detail needed to implement the GC SOA from a practitioner's point of view. The goal is to ensure all audiences come away with a good foundation background of the GC SOA and a consistent understanding of the basic concepts.

It also assumes the reader is already familiar with the **Statement of Direction for the GC SOA in the Government of Canada**, as such it does not reiterate the benefits of service orientation, nor the rationale and motivation for adopting the GC SOA.

2.3 Scope

The scope of this document is the introduction of the basic concepts of Service Oriented design. It covers the topic from the perspective of the Canadian Federal Government. It is not the intent of this document to reiterate the volumes of excellent industry literature regarding SOA, its usage and benefits.

3.0 GC SOA Overview

3.1 Introduction

The GC SOA is CIOB's design of a holistic Service-Oriented enterprise model that starts at the business level and permeates down though all levels of an organization down through the technology layers that support the business.

The delivery model for service orientation is based on the theory of the marketplace. If a service has value, then a consumer will use it and if a service has a lot of value, then a lot of consumers will use it. This makes a service a very important and reusable building block for planning and designing government programs that achieve desired outcomes.

The marketplace also leads to a dynamic community of Service Providers and Service Consumers. In general, entities (people and organizations) that offer capabilities act as *service providers*. Those with needs who make use of services are referred to as *service consumers.* Together the service providers and service consumers are sometimes referred to jointly as *service participants.*

In the sharing and reuse of services, the participants should (but not always) derive mutual benefits. In arms length transactions, providers can amortise their investments over a broader community while the consumers have ready access to desired services without the need to create them from the ground up. In a closed community like the federal government, the service participants work together to achieve common goals such as: Results for Canadians, Horizontal Service Delivery, Service Modernization and greater efficiency in the management of the public purse. In either scenario, they collectively benefit from each other's presence.

At present, the main emphasis of the GC SOA has been restricted to the reuse and interoperability aspects of run-time services only. For instance, services such as a mailroom services, a help desk or a correspondence tracking system are all reusable in a run-time fashion since with some negotiation they can presumably handle the phones calls, mail or tracking for a new community.

There are also many opportunities for reuse during the planning and design phases. For instance, a best practices guide for strategic planning, a template for service level agreements, or a reference model for secure network design are examples of reusable assets that are of value when planning or designing a service. These types of non-operational (runtime) services are not immediately consumable and are not part of the initial scope.

3.2 Setting the Business Context

Many technical papers have been written on SOA and organizations are already deriving success and benefits from its implementation. It is becoming clear however that having the right suite of reusable services is best achieved when the services are informed by and mapped to the business. In other words, the services should be developed with specific business contexts (or requirements) in mind.

If services are developed with a different set of assumptions then their reuse will be limited. For instance, if someone offers a delivery service, then presumably that service has a high degree of reuse for anyone else trying to move goods. If those goods however include ice cream then all of a sudden the potentially shareable delivery service might not be of value. The designers may have anticipated the ice cream and similar goods and chosen to build the entire delivery service based on climate-controlled trucks. That design however, might include unnecessary overheads for dry goods. With proper business context and a clear appreciation of how the services will be used, it is much easier to make the proper decisions and trade-offs in designing the services.

TBS has already invested in an extensive set of methods and techniques for analysing and expressing the business of government in a common way. The Governments of Canada Strategic Reference Model (GSRM) prescribes a common language used to describe the operations of the public sector of Canada from several perspectives, so individual departments, "clusters" of departments, and central agencies, can more clearly describe themselves and thus "map" common services and the business processes that support them. This GSRM reference model is the first step in achieving a higher degree of interoperability and service reuse. By conceiving and designing services in a common manner, with the intent to make them reusable, the highest degree of benefit from service orientation can be achieved. The GSRM serves as a whole-of-government standard for expressing business services and creating a system of record that can be used in cataloguing services as well as supporting reviews of strategic business designs and implementations.

The GSRM offers 19 distinct service output types that can be used to categorize the many distinct services offered by the numerous federal departments and agencies. These service types are used as the initial classification for business and technical services. As an analogy, recall from your high school physics classes that there are only 6 types of basic machines (the lever, wheel & axle, inclined plane, wedge, pulley, and screw). All other machines, from your basic office stapler to an orbiting space shuttle, are simply the reuse and aggregation of these 6 simple machines.

The 19 GSRM service types have been determined to be the basic building blocks used to deliver all the outputs needed to achieve the outcomes desired for the many federal programs. Individual services can be combined into higher-level services and expressed using models known as SIAMs (Service Integration and Accountability Models). These models can be used to communicate the accountability relationships between the various service providers and service consumers utilized and derived by the Service Negotiation process.

The GSRM also offers generic process patterns for each service output type. This allows for the identification and standardization of services at a more granular level. Drilling into the service process patterns allows departments to identify possible areas of reuse at a more detailed level. For instance each of the service patterns include a process called “collects and accounts for [service name] output fee”. With the growth of the Internet, and emphasis on GOL, many departments were increasingly executing this pattern online using credit cards. That spawned the birth of a new service called the RGBB (Receiver General Buy Button). By factoring out a common process and elevating it to the level of a shared business service, PWGSC was able to offer a common approach to collecting fees via online credit cards without the need to have each department reinvent the wheel in this area.

The GC SOA, as a layered architecture, employs the same principles to combine and reuse services to create ever more powerful and beneficial services of increasingly higher order. The reality of this highlights the value of service orientation.

Once we take a process and elevate (and encapsulate) it to the level of a service, there is typically a series of processes that it in turn precipitates. For instance, the RGBB needed to offer departments a means to query a given credit card transaction, perhaps offer refunds or make other billing corrections when necessary. The result is the RGBB evolves into an application with a rich set of functionality all designed to support and manage a shared business service.

The GC SOA therefore starts with the GSRM as a common approach for defining government programs and decomposing them into business services and business processes. This then provides a starting point to contextualize the creation of automated solutions that support the business. Developing an application is after all not about writing code, it is really about addressing and satisfying the business requirements.

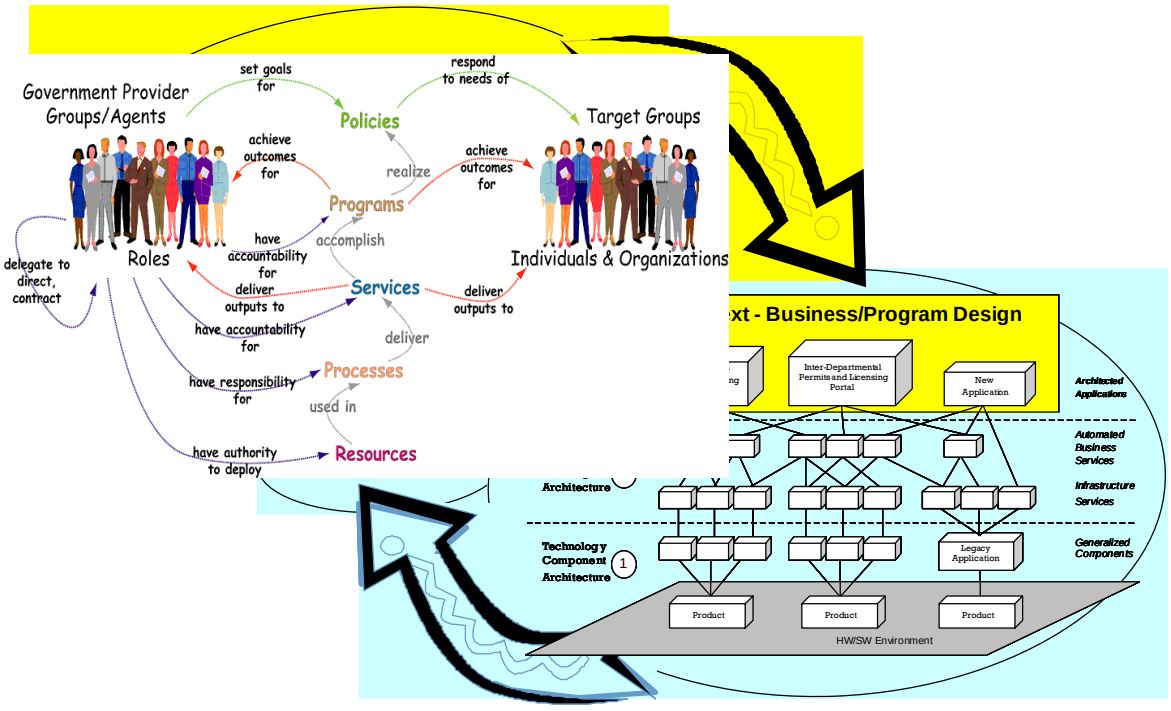

Figure 1 Linking the GC SOA to the GSRM

Thus GSRM based services and processes are often addressed by the implementation of business applications (the top layer of the GC SOA stack). These applications and their many subordinate components (in the SEA and TCA layers) in turn become part of the GSRM resources that help support the business needs.

In the Information Technology (IT) world, business needs are typically addressed through the building (or acquisition) of computer applications. By properly mapping the applications to the business, the organization's agility and adaptability is significantly increased. Clearly if there is a one-to-one mapping between an application and a business function then dropping or changing a business function would have impacts on the one underlying application and presumably not cause a chain reaction across many applications. Conversely, if an application addresses a wide multitude of business functions, the application is constantly being impacted every time some element of the business changes.

3.3 Deriving Value from the GC SOA

As demonstrated earlier, there are significant advantages to being able to take a complex machine and express it as a series of simpler machines. In the IT world, complex systems can be expressed as a series of small discrete units called components. When service-orientation is applied to an application, we increase its resilience to change and make the individual components more reusable. This decomposition process is repeated resulting in several layers. The magic is to try and manage the overall process by having a clear approach to decomposing the application in a systematic manner and have a rationale for knowing how to organize the many components into distinct layers. This is the heart of the GC SOA reference architecture.

By introducing these layers in a formal manner, industry has shown the value of Service Orientation to be in its ability to:

- Facilitate the manageable growth of large scale enterprise systems;
- Provide a simple scalable paradigm for organizing large networks of systems that require interoperability;
- Minimize trust assumptions among providers and consumers to promote greater business agility and autonomy; and
- Integrate functionality across ownership boundaries.

The key to making these benefits real is to generate momentum for the creation and reuse of services. Collaboration is needed to have providers create and register a portfolio of reusable services, plus make it practical for them to be discovered and shared by potential consumers. This collaboration model can be represented in the following manner.

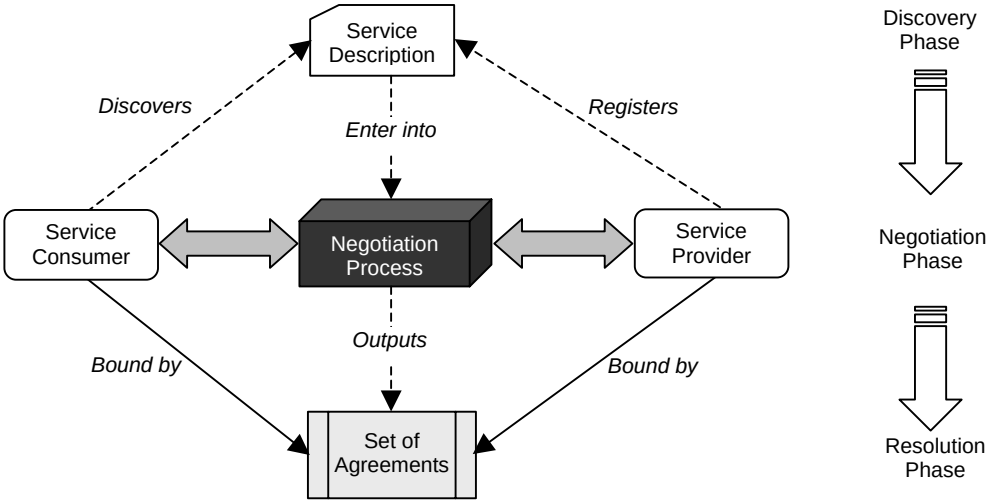

Figure 2 Service Negotiation process

4.0 The Three Layers of the GC SOA

The GC SOA specifies three distinct layers within an overall business context (the virtual fourth layer). These three layers represent an industry-recognized approach to the layering of any IT architecture. The main alteration to the broader SOA models commonly seen in the industry today is the encapsulation and naming of the layers such that each layer's name reflects what it comprises and the motivation for encapsulating it.

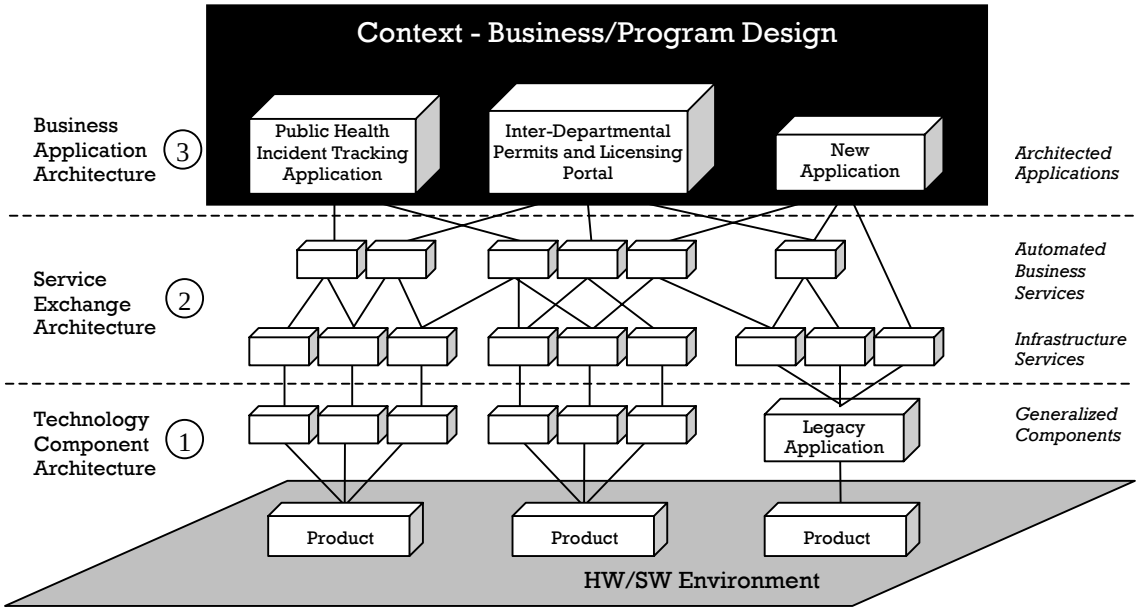

Figure 3 The Three layers of the GC SOA

4.1 Layering Benefits

The GC SOA architecture is broken down into these layers in recognition of the intuitive and well-known reasons noted below.

- It is easier to understand one layer at a time as a coherent whole rather than grasping the complexity of the entire architecture from top to bottom.
- Dependencies between layers are minimized and interconnections between entities are reduced to a manageable set.
- Typically standards are introduced to layer interfaces so that the maximum number of consumers can utilize the services provided by the supporting layer. The Open Systems Interconnection reference model is a widely used example of just such a layering.
- As one moves up the layers (or stack if you will), the level of abstraction increases hiding the details and complexity thereby allowing one to conceptualize and solve higher order business problems.
- With standards firmly in place, layers can be replaced or interchanged with other layer providers. For example in the OSI model, should a layer be a TCP/IP layer, there are multiple TCP/IP stack providers one can choose from.

Lastly, like any building, once a layer or floor is built, it is usually easier to build another layer right on top (of course only if it makes sense to do so).

4.2 The Individual Layers

A lot of thought has been given to the question “why are three layers necessary to describe the GC SOA?” The answer is multi-faceted in that:

- Each layer provides the necessary services for the layer above it
- Each layer is self-describing, and
- Each layer embodies a particular architectural or technological concept

The three layers defined within the GC SOA are:

Layer 1 - Technology Component Architecture

Layer 2 - Service Exchange Architecture

Layer 3 - Business Application Architecture

Each layer will now be discussed in turn.

4.2.1 Technology Component Architecture – Layer 1

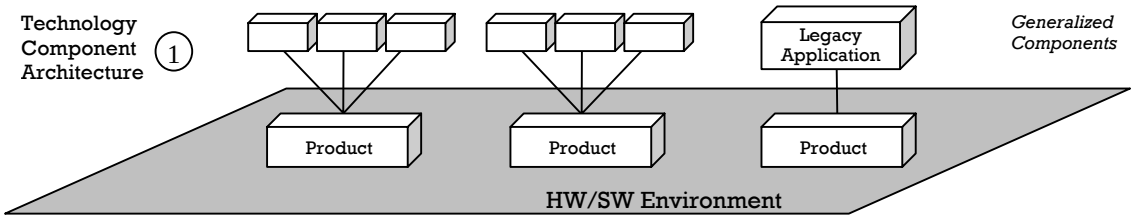

Figure 4 Technology Component Architecture (TCA) – GC SOA Layer 1

The Technology Component Architecture (TCA) contains vendor specific products, services and their supporting architectures and identifies the lowest level components that can be re-used across the GC ”out-of-the-box”. In other words, you can reuse it as-is. There is no formal way to use just part of it, or alter its intended behaviour. In essence, it is a black box that does what it does. This layer contains all of the hardware and software required to expose the building block components (regardless of complexity and size) offered by the vendors. Examples of these components might be the Java language or computer video card. They are what they are; you use them as they exist and have little opportunity to alter their capabilities without involving their vendor/supplier. Each of the products are assumed to be self contained and non-interacting. The expectation of interoperability in this layer is not warranted as each vendor has the right to use standards appropriate for their own implementation. This does not mean however, vendors could not and should not try to interoperate with each other directly, and in fact some actually succeed in doing so, it simply is not a requirement that the Government of Canada places on vendors.

An important consideration in this layer is the degree of granularity. In other words, how bite-sized are the components that have been made available by the vendor/builder. More granularity implies more flexibility in customizing a system, because there are more, smaller increments (granules) from which to choose.

Legacy applications that standalone and are built directly on vendor platforms can also be considered a part of the TCA. These in-house applications are treated exactly the same way as vendor products due to the fact that the only significant differentiating factor is who developed it. Of course, infrastructure products offered by vendors typically are more easily brought into the service-oriented model, but with today’s adapter technologies and available efficient custom coding techniques, applications also can be considered technology and process service providers.

4.2.2 Service Exchange Architecture – Layer 2

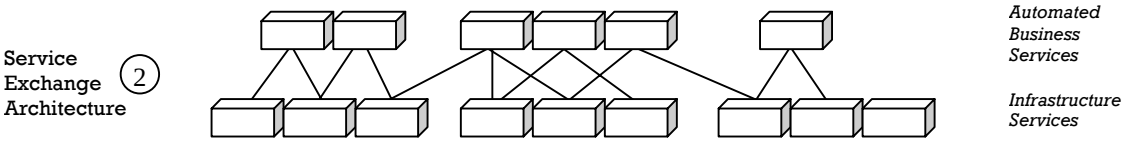

Figure 5 Service Exchange Architecture (SEA) – GC SOA Layer 2

The Service Exchange Architecture (SEA) is the middle layer that supports the one-to-one mapping of a component offering in the TCA layer to what is termed an infrastructure service or *i-service*. For example, task management and calendaring are components of e-mail, which can be represented, treated and utilized separately. Usually what transpires is that a standardized service template is made available to wrap the vender specific service in order to conform to any GC wide standard. These infrastructure services would then be used as building blocks upon which other value-add theme-based services (e.g. travel or health services) could be built and exposed.

Automated Business Services, also called *composite services* in the IT industry, are services composed of other services and they represent the key value-add proposition of the SEA. Typically composite services deliver key business functions or processes that are part of some higher order application or program delivery vehicle. The higher order composite service consumer may be other internal services, applications or external services for citizens or business.

Regardless of usage, the recursive nature of services calling services allows for program complexity where warranted or simplicity when desired. Figure 5 has been simplified and simply shows a single composite layer referred to as the Automated Business Services although a composite service can consist of other composite services to any depth.

Of special significance to this layer is the distinction between services exposed using proprietary or industry standard interfaces vs. services which are exposed using GC wide approved interfaces. Examples of industry interfaces include ebXML, Web Services and traditional EAI (**E**nterprise **A**pplication **I**ntegration). An example GC wide approved interface might be the XML based interface to the Secure Channel services broker. In instances when there is no need to tailor an industry standard, the GC approved standard might be identical to the industry standard. One such example is the GC adopting POP3 as the interface between any e-mail client program and the many GC mail servers.

Figure 6 TCA to SEA common service exposure, shows vendors exposing an interface within the TCA as a component. Although many vendors are moving to less proprietary interfaces which is very positive, to be considered an *i-service* in the SEA layer, TCA component functionality must be offered via a GC wide approved interface.

In the SEA layer, the GC SOA can simply adopt industry interfaces that are sufficiently broad such that all departments can be assured of interoperability by selecting such services from fully compliant vendors and sources. In other instances the GC SOA will promote “wrapping” the components and their interfaces with a GC wide standard defined centrally. This will insulate departments from the actual service providers at the TCA layer and provide a strong degree of vendor independence.

When such wrapping does occur, potentially the message header (as part of the service call on the wire) will be considered along with the message contents (also called the payload). In some cases the wrapping will simply provide insulation in the header portion. In other instances, the payload will also be manipulated by the wrapping interface.

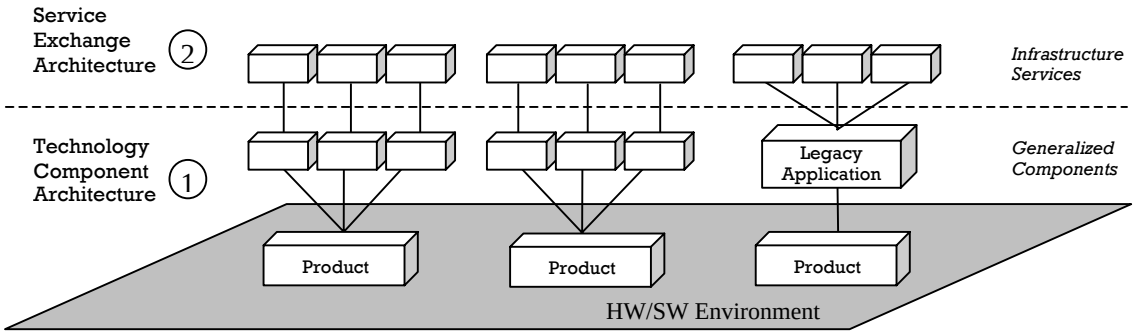

Figure 6 TCA to SEA common service exposure

4.2.3 Business Application Architecture – Layer 3

Figure 7 Business Application Architecture (BAA) – GC SOA Layer 3

The Business Application Architecture (BAA) is at the top of the architecture stack. In essence, it allows business owners to package a selection of GC (or non-GC) services to be used in a new value proposition for the crown. Typically the scope of an application is flexible and it can be defined to include a little more or a little less functionality depending on the time and resources available. Applications can be created through new code, acquired code, or repurposing existing code. For example, by repurposing existing departmental and cross-jurisdictional health related services, one could create a composite real-time and location independent health and travel advisory service, including food recalls, air quality, and medical clinic information for use in a single window for travellers. Such contemporary public services would leverage new and existing public and/or private services depending upon the service levels and access controls associated with them.

By their very nature, applications within the BAA layer are incapable of "direct" reuse. The users of the applications are generally people suggesting the BAA contains those elements of the architecture related to service delivery. To support inter-operability between applications in this layer, the applications must utilize services located in the Service Exchange Architecture layer where information exchange is made possible. In other words the BAA layer is the presentation layer along with any associated interaction logic and other non-reusable aspects of a business application.

The BAA is also decomposed into its own distinct layers. These layers are connected via an Application Services Bus (ASB), which can be proprietary and/or local to the application. When interactions to the SEA layer are involved then an Enterprise Services Interface (ESI) provides a standardised model for external service consumption.

To ensure the lower level components are as reusable as possible the BAA maintains in its internal Process layer and Persistence layer the business specific rules and data that are distinctive to this business instance. When a process layer function is common across many BAA instances then it too, becomes a candidate to move down the GC SOA stack to the SEA layer and gets exposed as a new service.

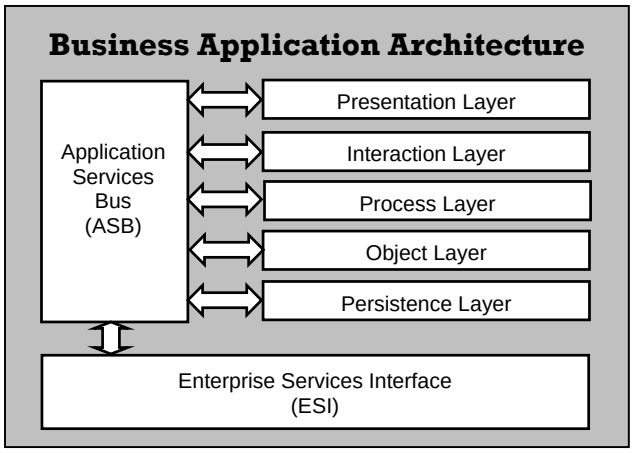

Figure 8 BAA internal layers and communication mechanisms

The following is a brief overview of the layers internal to the BAA.

Presentation Layer Displays content, prompts and results on whatever device is being used (e.g. browser, cell phone, PDA, 3270 screen etc) including transformation functions to support accessibility technologies

Interaction Layer Deals with end user guidance and management of data capture, processes and inter-service interoperability (back button, help, change menu, etc)

Process Layer Manages the overall logic and workflow for the session. Includes models of business processes that enable service consumers and service providers to have a shared understanding of each process and to select processes and services based on responses from the service partner

Object Layer Handles any BAA specific business rules such as data validation. Includes items that are modeled in accordance with a COI adopted object model standards (e.g. business information entities defined according to the UN\CEFACT specifications and guidelines for core components – an application of ISO 11179).

Persistence Layer Handles information retrieval and storage needed to maintain session continuity including technologies such as digital signatures and encryptions that safeguard the integrity of objects interchanged between service exchange participants and the guidance on storing them.

The BAA contains a range of applications, from the tightly focused single purpose application to the large multi-purpose behemoth. The BAA also contains applications that are truly unique or “one-of” to a particular business unit or program or even a department. In the past, without a strong service orientation and well-defined standards, the tendency was to build larger and larger applications that covered a wide breadth of business needs. This helped ensure that all the various pieces of the solutions could interact and function as a whole.

The GC SOA will have the tendency to foster the growth of smaller more focused applications, once people break out of the mind set that the project (or program) and the application are different. For instance, in the above figure, we may have a health and travel advisory project with the key deliverable from this project being sometimes thought of as the health and travel advisory application. In a service-oriented world, the “application” will be more synonymous with a portfolio or collection of components related to health and travel brought together for the expressed business purpose of providing a comprehensive advisory service. If and when the business services are reused (or changed) then the top-level application(s) can also be refocused.

A significant amount of the BAA architecture focuses on the design-time aspects of application design since the models and patterns applicable to it can be shared and re-used amongst a number of applications during the development phase. It is these models and patterns that will be the focus of discussion in the detailed documents to come.

4.3 Another Perspective on the 3 Layers

The GC SOA can be viewed from another perspective for those still perplexed as to how the layers fit together.

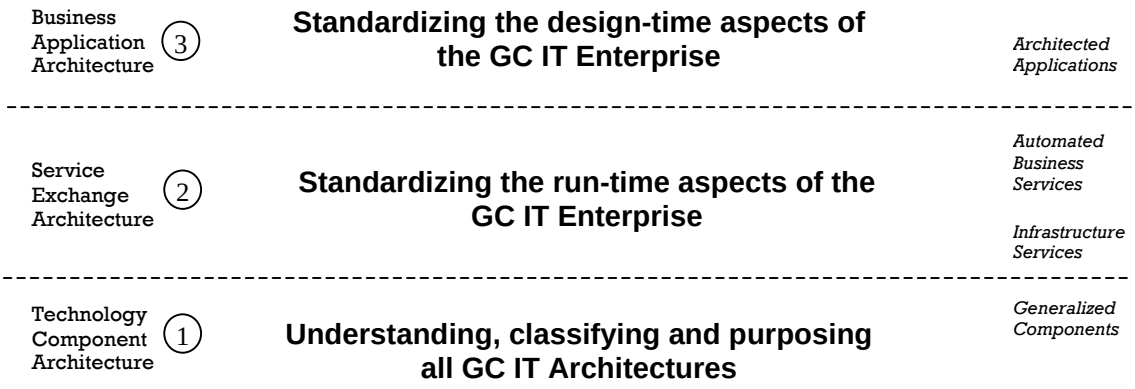

Figure 9 GC SOA alternate perspective

Although Figure 9 shows a simplistic segmented view of design-time and run-time partitioning, by no means is it true that all of the design is done at the top layer. The design of services is still done in layer 2, but the main focus or output of layer 2 is an interoperable run-time environment.

5.0 Transitioning to SOA

The primary goal of service-orientation is to move away from large monolithic applications that traditionally do not have well defined components accessible independent of the top-level application in which they are embedded. Figure 10 below illustrates the type of closed application that permits virtually no reuse.

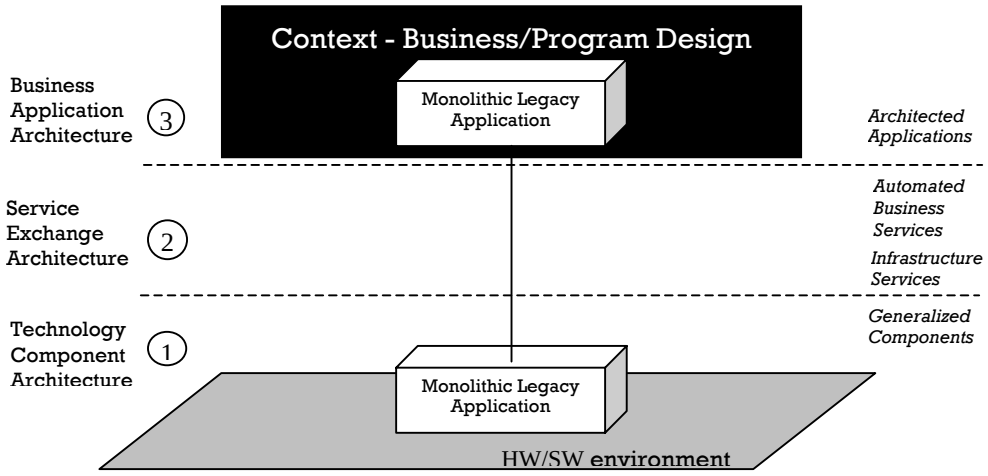

Figure 10 Legacy application classification paradox

Notice that the application straddles both the BAA and the TCA layers. This is acceptable since the application has aspects that reside in both layers. Normally the only way to interface to such an application (without changes) is by imposing some logic over its existing user interface (such as keyboard macros or screen scraping).

Instead the goal is to build applications out of application components in the BAA layers using as many "black box" components in the TCA layer, always accessing them via a formal SEA layer.

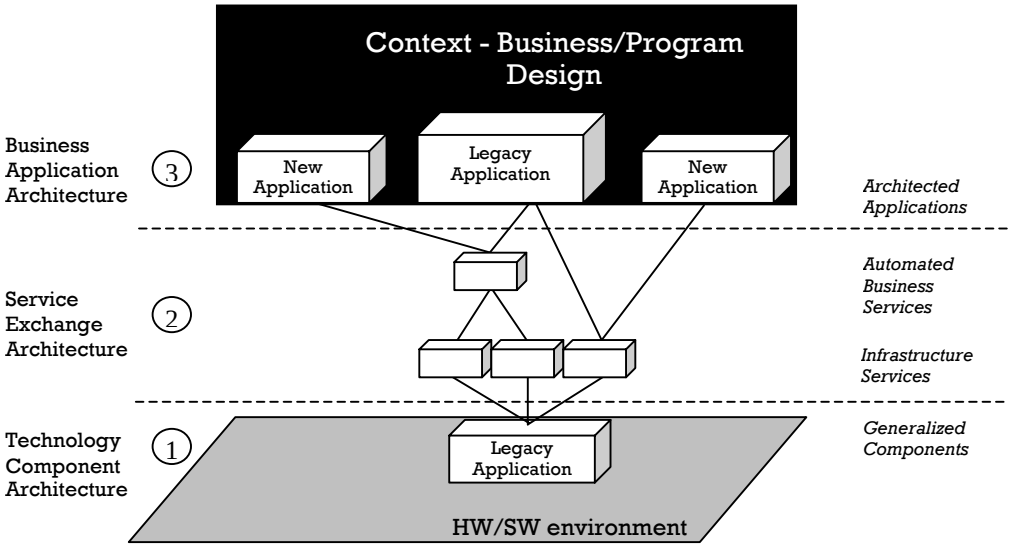

Figure 11 Introduction of the SEA to a legacy application

Applications should thus be built or retrofitted to make available discrete elements of reusable logic, accessible through the SEA Layer. In this example the application still performs all its original function but some useful application components have been isolated and exposed for external use via new SEA interfaces. As a result, two new applications can leverage some of the proven benefits previously buried inside the legacy application. The overall goal is to move as much of the new logic down the layers, towards the TCA. The result is a growing collection of reusable "black boxes" of functionality that act as future building blocks for broader deployment, inside and outside of the organization.

6.0 Next Steps

Upon reading this primer the reader is encouraged to delve into those specific areas that relate to their own interests and needs.

Executives, business planners, program analysts and other non-technical staff will then want to look to the **GC SOA Business Context/Design Primer** for specific guidance regarding the application of the GC SOA in the business realm.

Technical readers should look to the documentation entitled the **GC SOA Model** for specific details regarding the GC SOA reference model, as a start in guiding their application and component designs.

The **GC SOA Overarching Principles** offers high-level guidance in the adoption of the GC SOA and complements the other documents in this series. These principles are applicable to the business as well as the technology layers of the architecture.

7.0 Appendices

7.1 Acronyms and Abbreviations

Term	Definition
API	Application Programming Interface
ASB	Application Services Bus
BAA	Business Application Architecture
BTEP	Business Transformation Enablement Program
CIOB	Chief Information Officer Branch
EA	Enterprise Architecture
EASD	Enterprise Architecture and Standards Division
EDA	Event Driven Architecture
ESI	Enterprise Services Interface
GC	Government of Canada
GSRM	Governments of Canada Strategic Reference Model
HTML	Hyper Text Mark-up Language
HTTP	Hyper Text Transfer Protocol
IM	Information Management
IP	Internet Protocol
IT	Information Technology
MDA	Model Driven Architecture
PWGSC	Public Works and Government Services Canada
SE	Service Exchange
SEA	Service Exchange Architecture
SIAM	Service Integration and Accountability Model
SLA	Service Level Agreement
SOA	Service Oriented Architecture
TBS	Treasury Board Secretariat
TCA	Technology Component Architecture