



### Office of Technology Strategies (TS) Architecture, Strategy & Design (ASD)

#### A VA Executive's Guide to the Application/Services Layer

##### INTRODUCTION

CTS Note #3 introduced Service Oriented Architecture (SOA) and provided a general description of how modern applications are built with multiple "layers" often referred to as the Presentation, Services and Data Layers. This note dives deeper into the concept of SOA by examining the Services Layer in more detail.

##### BACKGROUND

The simple explanation of the Services Layer is that it is the part of a modern application that actually does all the work. It is how Web pages are able to interact with databases and where the majority of data manipulations and calculations take place. It is also the part of a system that does most of the work to ensure the application is secure.

In short, the Services Layer does all of the heavy lifting. For example, when a user tries to search for information in a modern application, the Web pages that make up the Presentation Layer—the pages in which

they enter search terms or click a link—simply pass those search terms to the Services Layer (See Figure 1). Behind the scenes, multiple services do all the hard work needed to fulfill the request: Where does it search? How many answers will it return? Is the user authorized to access the answers?

##### SERVICES LAYER BENEFITS

Using discrete services to perform applications tasks completely independent of the Presentation Layer provides many advantages, including:

- Elimination of point-to-point connections
- Reusability of shared services

Legacy applications built without this "separation of concerns" into different "layers" rely on multiple point-to-point connections to other applications and data stores. These interfaces require a lot of time and resources to create, modify and maintain. This makes modifying or adding new functionality expensive, time consuming and complicated.

With a Services Layer, instead of each application "hard-coding" each of those connections, applications reuse shared services that are much less expensive to build, operate and modify.

Applications that use services can simply mix and match different services from within the Services Layer or

This newly established office within OIT's Architecture, Strategy & Design (ASD) interacts not only with the ASD pillar offices, but also with multiple stakeholders within OIT and with strategic offices across the enterprise. TS works closely with IT and business owners to capture business rules and provide technical guidance as it relates to Data Sharing across the enterprise, specifically for inter-agency operability.

change the sequence in which they call them, to rapidly provide users with new and different capabilities. Additionally, new applications can be built much more rapidly by creating new Presentation Layers that reuse existing services.

##### Decreased Time and Cost

The Services Layer provides the enterprise with a set of reusable services, which modern applications access to perform any number of necessary functions to provide capability to users. This decreases the time and cost to create custom functionality for each distinct application.

##### Increased Information Agility

One of the key challenges VA faces in the next five to 10 years is the ability to share and use information, including information from other federal or private entities. Services allow an organization to access more of its own data, and enable external applications to share or use common functionality and data.

##### Improved Security

Service Layer security offers an agile approach that focuses on the communication

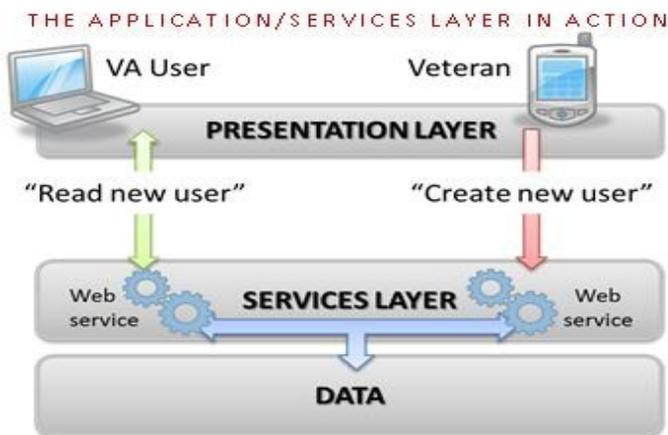


Figure 1 shows how a Veteran creates a new user account using web services in an application services layer. A VA user reads the new user via the same services layer, which interacts with the data layer so that neither the Veteran nor the VA user needs to know the details of the data layer.

(Continued on page 2)

### A VA Executive's Guide to the Application/Services Layer, cont'd

between systems (messages), and on specific services that interact with information (end-to-end). The Services Layer can orchestrate multiple security services for one interface (e.g., authenticating users as well as encrypting data).

#### HOW ARE SERVICES BUILT?

Enterprise services may be developed as part of a specific project, but must be designed to operate independent of any specific application. In VA, the majority of our data is either created, read and modified by many different applications, or, (unfortunately) duplicated by them. This is because many different VA users have similar business needs. Therefore, it is critical that all services be built as potential "shared services."

#### User Requirements

Users generally think in terms of business functionality and not in terms of the series of discreet services sequenced in a particular order to achieve that functionality. This makes it difficult to translate user requirements into service design. However, using an Agile Development methodology to break functional requirements into specific and finite application functions makes documenting service requirements much easier. Application developers need reusable services to be truly "Agile" and SOA provides maximum benefit when used in an Agile environment.

#### TYPES OF SERVICES IN THE SERVICES LAYER

Create, Read, Update and Delete (CRUD) Services: These are simple interactions between Presentation and Data Layers via the Services Layer. One user may create, read or modify (including delete), and another user can view those changes when the record is accessed. The service allows multiple applications to modify a single set of data,



Figure 2 shows two different RESTful web service calls for finding user data. The left part shows the output of an application making an HTTP POST operation to a VA server add a record for Joe. The right part shows an HTTP GET operation to the same VA server for obtaining a record for Joe.

which can be immensely useful in workflows that rely on multiple systems.

Composite Services: These perform smaller sets of tasks within a larger process. For instance, if a user requests information, a composite service will authenticate the user with a separate data store before a CRUD service returns the appropriate information, ensuring they are authorized to view that information. Composite services can also help CRUD services return separate data elements for the same user request (e.g., savings and checking accounts in a single online banking dashboard).

#### ADDITIONAL KEY CONCEPTS

Representational state transfer (REST): A Web-service architecture in which services are designed to focus on calling system resources regardless of the state of those resources (See Figure 2). This "statelessness" improves Web service performance by including all necessary information in the RESTful Web service call. This is a fast-growing alternative to more established Web services, and is used by companies like Google and Facebook because it is designed for mobile and Web applications

processing form-based data. In VA, we use REST for similar purposes, specifically in designing user interfaces and their interactions with form-based information in the Data Layer (e.g., DD 214).

Simple Object Access Protocol (SOAP): Provides a simple mechanism for sending information in XML format over a network in Web service interactions between applications. SOAP is not programming- or implementation-specific, meaning it can be used in interactions between a variety of applications and systems. In VA, we use SOAP for system-to-system interfaces that share information that is not form-based.

Application programming interface (API): At a basic level, APIs specify how software components interact with each other. In a Web service environment, the API is what allows a user's application to perform actions via a remote call to another system's components (e.g., search its data store).

If you have any questions about the Services Layer, don't hesitate to ask CTS ([askCTS@va.gov](mailto:askCTS@va.gov)) for assistance or more information. Check out earlier CTS Note editions [here](http://vawww.blog.va.gov/oit360) ([vawww.blog.va.gov/oit360](http://vawww.blog.va.gov/oit360)).