API Exposure for Integration

Executive Summary

The data integration milestone has passed successfully by implementing the Virtuoso Virtual Database. With our capital line-of-business systems running normally, and with local access to their relational engines intact, we have created the all-important unified schema through the Virtuoso VDB. That's progress.

The next order of business is to align application functions via Web Services exposure. Web Services are the accepted way of creating and combining composite services from mixed bags of applications, and is a viable and standard-based method for integrating IT operations.

Post-merger data models unified via the Virtuoso VDB grants an opportunity not commonly found in competing data integration solutions the ability to expose data, SQL stored procedures, and application code, via the ubiquitous URI using Web Services. This exposure of functions through a standardized medium provides a workable, incremental method of integrating IT systems.

Laying the Foundations for API Exposure

Knowledge of your data storage systems is a minimum prerequisite for merger integration using web services API exposure. Virtuoso provides a unified interface for external RDBMS connections, and a mechanism to inspect the stored procedures that make up the lion-share of capital LOB functions.

Virtually all contemporary applications, custom and ready-made, sit atop a popular relational database management system. This fact is emphasized because most of the data handling routines exposed via web services compliant URI's are enshrined in SQL stored procedures of these various databases.

Foundational LOB applications provide a standard programming API used for calling specific functions. Third parties also sell language specific connectors for Java and other environments.

The salient point of this practicum is that capital applications enshrine logic and data handling routines as SQL stored procedures within the underlying DBMS. Finally, with this knowledge in hand, we can reveal:

'SQL stored procedure attached via Virtuoso's Virtual Database engine, regardless of which application it serves (a custom application grown in-house or a $500k+ SAP system), may be exposed as a web service end-point.'

Using Virtuoso Universal Server, any capital line application function can be made available via a URI. Let's break it down by family:

Capital Line applications may belong to one of several species:

1. **Off-the-Shelf**: The high-end of these applications (SAP, PeopleSoft) all sit atop a big three database, such as IBM DB2, Oracle, or MS SQL Server. The mid-end accounting systems or vertical industry applications may embed storage. For those sitting on the broad shoulders of a major database, Web Services exposure will prove to be straightforward. For the more obscure verticals, thankfully, they may also provide API connections; if not, the data file can be broken open with XSLT[1], or XQuery, and converted to SQL data.

   An interesting instance is mid-end accounting programs, QuickBooks being a good example. This mega hit application started as a small business accounting package with no point of entry for customization, yet has grown to rival enterprise accounting systems - with the inclusion of a complete API.

2. **Homegrown**: A great deal of applications are custom made from the ground up. The majority of custom applications are two-tier; databases sit beneath an application layer provided by a me'lange of programming environments. Web Services takes the stage as a fairly easy path to integration, as the core of the application is likely composed of SQL stored procedures.

   Business logic bound in programming language runtimes may also be hosted and invoked within the Virtuoso Environment, while the presentation layer may eventually be extended or dispensed with in favor of a Web Based application interface. With Virtuoso, there is no need to make any rash decisions; URI Web Service invocations read and update the underlying database, and the legacy application can be held over during the conversion.

3. **Hybrid**: Larger enterprises will fall into the hybrid category. Complex business logic may span Microsoft Tools (SQL Server), languages (.net and C#), and possibly encompass a Sun Solaris implementation (Oracle on Solaris).

   The same rules apply to the DBMS - all SQL procedural code is fair game for exposing via a web services URI invocation. The application logic may, in the case of large hybrid systems, need inspection via a Java reflection[2] API, or code listings for early-bound language families. Ultimately, object languages

and frameworks of the late-bound variety (J2EE, C#) are easier to factor into web services applications than older compiled binaries.

This exposition of the foregoing system flavors sets the stage for the task at hand, exposing application functions in the form of : (system1.payrollrecordadd.com)[3]. Lets jump in.

End-Points, SOAP, and Virtual Directories

The following case of an IT driven mergers shares a common thread with an internal systems integration - each of the partner organizations/internal systems have in-place methods and applications for dealing with daily operations. An excellent example is adding a Client Record.

In our most onerous case, a systems hybrid of a commercial ERP system and a custom inventory application built atop a DBMS, we envision a case to expose two AddClientRecord API functions as a Web Service. As you might surmise from our introduction, yes, the ERP system was inherited from Company 'A', and the custom Inventory system from Company 'B'.

Three concepts define the composition of an executable web service URI:
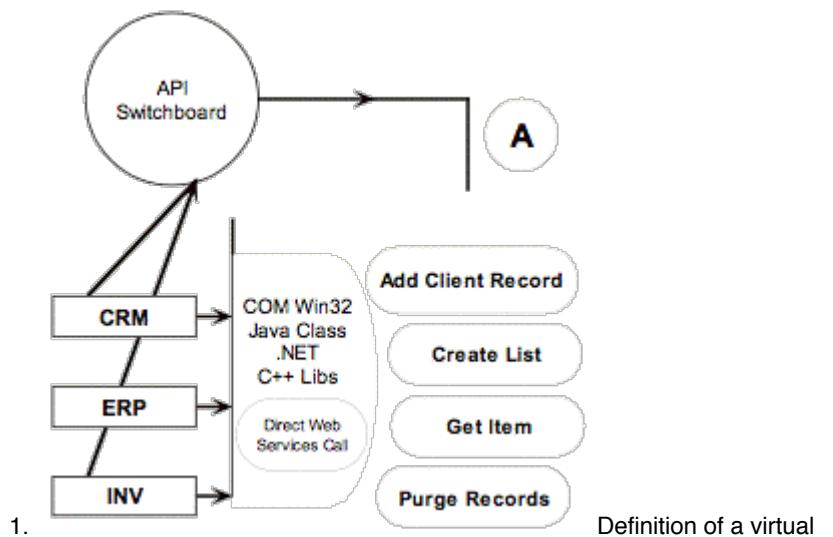
1. Definition of a virtual Directory to host the service and connect the URI to the service. Think of the virtual directory as part of the web server - a home for one SOAP end-point.
2. Composition of the service by selecting either a SQL stored procedure within the Virtuoso core DBMS engine, or an attached SQL procedure via the VDB. Alternatively, the virtual directory may host run-time code.
3. Defining the end-point as a SOAP service, with appropriate SQL account security settings.

A SOAP end point is created using Virtuoso's virtual directory- essentially a web

server executable services function. We can do this via the administrative interface or programmatically via Virtuoso's Procedure Language.

The Simple Object Access Protocol (SOAP) is an XML-based application layer for information exchange, nothing more. SOAP defines the 'traveling' format for your URI-based messages. The most common way to transport SOAP messages is HTTPS.

For those web services calls between a known caller and server - in this merger case, for instance, we may rely on the apriori knowledge of the function variables or prototypes exposed within the SOAP message body. Having access to the to the SQL objects and program source documentation may be all we need to invoke these URI end-points and integrate our systems accordingly.

For web services targeted for invocation by external callers, WSDL, the Web Services Description Language, provides a method of conveying SOAP calling and return data parameters. Virtuoso automatically generates WSDL for SOAP services, and one may wish to use this as standard procedure. In a following article, we will examine the creation and binding of multiple SOAP service invocations with WSDL.
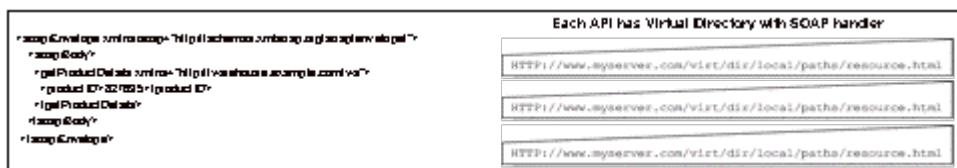
Learn More

- **Virtuoso Documentation**
- **Virtuoso Tutorials**

[1] Virtuoso Supports a complete suite of XML transformation services

[2] Inspection, Reflection, and Introspection all refer variously to methods of revealing code and data structures in Object Oriented Programs and languages, such as Java and .Net

[3] For a detailed explanation of the Web Services Nomenclature, see the excellent white paper, 'Virtuoso Web Services'.