Oracle Tuxedo in a SOA World

An Oracle White Paper
May 2014

INTRODUCTION

Service Oriented Architecture is one of the most widely accepted architectural approaches to building and integrating applications. Although the term was coined in 1994, the architectural approach wasn't widely accepted until a decade later. Today most large scale applications apply SOA in at least some part of their design and implementation.

Oracle Tuxedo is one of the original SOA platforms adopting SOA principals as part of its original design and development in 1982.

While the term may have been coined in 1994, the principles embraced by SOA existed long before then. Software development principles such as reuse, interoperability, standards compliance, and modularity have been used by software developers for decades. Architectural principles such as encapsulation, loose coupling, abstraction, autonomy, implementation and location independence, and late binding have as well been used at least since the 1980's. It is perhaps the combination of these principles that makes SOA unique.

Oracle Tuxedo is one of the original SOA platforms. Everything in Oracle Tuxedo is a service and invoked as a service. Oracle Tuxedo services are location and implementation independent, adhere to a contract, and are interoperable with other Oracle Tuxedo services and other service oriented platforms. As a result integrating Oracle Tuxedo applications into a SOA environment is natural and straightforward, regardless if the business logic is written in C, C++, Java, COBOL, Python, Ruby, or PHP. Oracle Tuxedo also provides industry leading reliability, availability, scalability, and performance. This paper will cover the various integration techniques that can be utilized to integrate Oracle Tuxedo applications into a SOA world.

Client Only Options

Several of the options available for integrating Oracle Tuxedo applications into a SOA environment are client only options. This means that the non-Oracle Tuxedo system can only act as a client to the Oracle Tuxedo application. One of the implications of this is that requests can only flow from the client environment to Oracle Tuxedo and not the other way around. Another implication is that the client's transaction context is never extended to the Oracle Tuxedo system. Any Oracle Tuxedo transactions started by the client are delegated to Oracle Tuxedo and are separate from any transaction context in which the client may be participating.

Server to Server Options

Server to Server integration options are for those environments where Oracle Tuxedo is able to make calls to the non-Oracle Tuxedo system as well as receive calls. In these environments some options support distributed transactions where transactions started in one server can be extended to the other server as part of a single global distributed transaction.

JAVA INTEGRATION

Many SOA based applications are developed using Java. Oracle Tuxedo provides two direct Java integration options. Jolt is an Oracle product offering that provides client connectivity to Oracle Tuxedo. The other option is the Oracle WebLogic Tuxedo Connector (WTC) component that ships as part of Oracle WebLogic Server (WLS). This component provides bi-directional connectivity to Oracle Tuxedo from WLS applications.

Jolt

For simple Java access to Oracle Tuxedo applications, Jolt is hard to beat. Jolt provides access to Oracle Tuxedo services from essentially any Java environment.

Jolt provides a simple to use set of Java classes that can be used in virtually any Java environment. This ranges from stand alone Java applications, applets running inside a browser, to clients running inside a Java Enterprise Edition (JEE) application server, providing great flexibility in deployment options. When used in WLS there is support for connection pooling to allow multiple clients to share the same Jolt connection to Oracle Tuxedo.

These classes support all of the Oracle Tuxedo standard client facilities such as invoking services, enqueuing and dequeuing messages to Oracle Tuxedo /Q queues, and subscribing to the Oracle Tuxedo event broker for publish and subscribe interactions. As well the Jolt classes support all of the common Oracle Tuxedo buffer types.

As a network client only product, Jolt can be used to access Oracle Tuxedo services running anywhere in an enterprise's network. Because Jolt only provides client connectivity, distributed transaction support is not available. Although Jolt applications can start an Oracle Tuxedo transaction, this transaction is delegated to Oracle Tuxedo and is separate from any Java transaction that may be active. To provide better network security, Jolt offers a relay component that allows the relay to be run on a firewall without requiring external access directly to the Oracle Tuxedo applications, thus allowing the Oracle Tuxedo application to remain safely behind the firewall.

Oracle WebLogic Tuxedo Connector

As a standard component of Oracle WebLogic Server, WTC provides out of the box capabilities to integrate Java applications with Oracle Tuxedo applications. WTC utilizes the Oracle Tuxedo domains protocol and thus can support both outgoing requests to Oracle Tuxedo as well as incoming requests from Oracle Tuxedo. Incoming requests are processed by an Enterprise Java Bean (EJB) on the

WLS side. Full bidirectional distributed transactions are supported which allows XA resource updates by WLS and XA resource updates by Oracle Tuxedo to be coordinated in a single global transaction.

WTC provides bi-directional service invocations to and from Oracle Tuxedo and Oracle WebLogic Server. With its integrated JMS bridge, applications can utilize both synchronous and asynchronous interaction styles.

WTC provides a set of Java classes that offer an API very similar to the standard Oracle Tuxedo ATMI API called JATMI, although within the context of the object oriented nature of Java. This means that developers familiar with Oracle Tuxedo will find developing WTC clients and services very natural. Access to Oracle Tuxedo services and /Q queues is available through these classes. For simpler integration with JMS, WTC provides a transactional JMS bridge that allows JMS messages to flow to Oracle Tuxedo /Q queues, messages to flow from Oracle Tuxedo /Q queues to JMS queues, and messages from JMS queues to be sent directly to Oracle Tuxedo services.

The domains capability in Oracle Tuxedo allows services to be shared across multiple separate domains. This is done using the Oracle Tuxedo domain gateway which uses a service import/export model. A domain requiring the services of a remote domain configures a connection between the domains and then imports the services it requires from the remote domain. If desired, the domain gateway can limit the services that a local domain exports to the remote domains.

In WLS a WTC server is configured to connect to one or more remote Oracle Tuxedo domains. Each WTC server can import and export services in much the same manner as the Oracle Tuxedo domain gateway does for Oracle Tuxedo. Oracle Tuxedo sees a WTC server on WLS as another Oracle Tuxedo domain. WTC and the domain gateways multiplex requests over a single network connection and multiple connections can be defined to provide automatic load balancing and failover/failback capabilities.

Oracle Tuxedo JCA Adapter

As part of the Oracle Tuxedo 11gR1 release, Oracle released a new product that provides similar capabilities as WTC but for all JEE application servers that provide a Java Connector Architecture (JCA) container. This product is a JCA 1.6 compliant resource adapter that allows bidirectional interactions between JEE applications and Oracle Tuxedo applications. Distributed transaction propagation from the JEE server to Oracle Tuxedo, and as of the Oracle Tuxedo 12c release, transaction propagation from Oracle Tuxedo to JEE is supported as well.

The adapter supports the JATMI programming interface used in WTC as well as supporting the Common Client Interface (CCI). By supporting CCI, the JCA Adapter for Oracle Tuxedo allows integration tools such as Oracle SOA Suite to be able to directly access Oracle Tuxedo services. With the Oracle Tuxedo 12c release, support for utilizing the adapter in Oracle SOA Suite is now provided in jDeveloper.

Native Java Support

In the Oracle Tuxedo 12c release, Oracle Tuxedo now supports implementing services in Java as plain old Java objects or POJOs. Developers can use JATMI to develop Oracle Tuxedo services in Java and leverage Oracle Tuxedo's scalable, highly available infrastructure to create or extend their enterprise class SOA applications. This first release of Java support includes support for JDBC to access databases as well as all standard Oracle Tuxedo features.

.NET INTEGRATION

Another popular platform for supporting SOA based applications is the Microsoft .NET framework. Oracle Tuxedo includes support of a .NET client library that can be used to access Oracle Tuxedo services from a .NET application. The .NET client is a standard .NET wrapper assembly which makes it easy to use by .NET applications. Like Jolt, it is a client only offering meaning that .NET applications can invoke Oracle Tuxedo services, but Oracle Tuxedo cannot invoke .NET applications. This also means that transactions are delegated as opposed to distributed.

The Oracle Tuxedo .NET client supports the development of Oracle Tuxedo clients in any supported .NET programming language such as C#, J#, VB .NET, and ASP.NET. The client library provides a set of object oriented interfaces that allow access to virtually all of the standard Oracle Tuxedo client functions.

WEBSPHERE MQ INTEGRATION

Many customers have chosen IBM's WebSphere MQ as their means for SOA integration. Oracle Tuxedo ships with a bidirectional MQ adapter that allows messages placed on MQ queues to be delivered to an Oracle Tuxedo service and the reply from the Oracle Tuxedo service being placed on an output MQ queue. The adapter can also advertise services that an Oracle Tuxedo application can call where the adapter takes the request and places the message on an MQ queue and replies to the caller with the corresponding reply message take from the associated reply or output MQ queue. Finally the adapter also provides a mechanism for Oracle Tuxedo applications to use the normal Oracle Tuxedo /Q APIs but the queues acted upon are MQ queues. Thus customers can use the standards based Oracle Tuxedo APIs to interact directly with MQ queues.

Applications needing staged transactional delivery of requests can use standard Oracle Tuxedo distributed transaction management to ensure exactly once processing of messages. The MQ adapter can be configured to automatically start a transaction prior to enqueuing or dequeuing a message. This transaction will be extended to any Oracle Tuxedo service that is subsequently called, thus ensuring transactional integrity of the queue with any other transactional resources such as databases.

MAINFRAME TRANSACTION INTEGRATION

Mainframe systems based upon IBM CICS or IBM IMS are often at the back end of many applications and provide core business logic and services. As customers adopt SOA there is often a need to access these backend systems as services. The Oracle Tuxedo Mainframe Adapters (TMA) provide an excellent foundation for integrating these legacy systems into a modern SOA environment.

The TMA adapters are built on the Oracle Tuxedo domains technology and allow the sharing of services between Oracle Tuxedo and the mainframe systems. The mainframe applications see Oracle Tuxedo services as though they were local applications. Thus a CICS or IMS application interacts with an Oracle Tuxedo service in exactly the same way it interacts with another remote CICS or IMS transaction. Similarly Oracle Tuxedo applications see the remote mainframe applications as local Oracle Tuxedo services and interact with them in the same manner as a local Oracle Tuxedo service. This means that all the other Oracle Tuxedo integration options can be used together with TMA to provide a wide range of SOA enablement and integration options for legacy mainframe applications.

Like other integration options built on Oracle Tuxedo's domains technology, TMA supports identity propagation and can support distributed transactions if supported by the underlying communication protocols. The adapters come in two flavors depending upon the protocol used; TMA for SNA uses the IBM SNA protocol and fully supports syncpoint level 2 distributed transactions whereas TMA for TCP supports integration with IBM mainframes but does not support distributed transactions.

With the Tuxedo 12cR2 release, new tooling is available to assist in making mainframe COBOL applications available to Oracle Service Bus (OSB). The Mainframe Transaction Publisher can generate OSB Web service proxies to mainframe transactions. It generates Java classes from COBOL copybooks that are used in a pipeline to the Tuxedo business service imported from the mainframe. This greatly simplifies the effort of integrating mainframe applications using TMA. All the necessary artifacts are generated directly from the specified COBOL copybook and automatically published to OSB.

WEB SERVICES INTEGRATION

Currently the most popular means of integrating SOA based applications is via Web services, in particular SOAP over HTTP (SOAP/HTTP). Most application platforms support Web services and provide mechanisms to both invoke Web services as well as respond to Web service requests. The Oracle Services Architecture Leveraging Tuxedo (SALT) product is an Oracle Tuxedo add-on product that provides bi-directional SOAP/HTTP support for Oracle Tuxedo applications. This product is completely configuration driven, meaning that no coding is necessary to access remote Web services or to allow remote applications to access Oracle Tuxedo services as Web services.

"We were up and running very quickly, all in all it took us less than 2 hours to get SOAP services going using SALT 1.1" – Kadaster – Dutch Land Registry

SALT provides a high performance point to point integration mechanism using open standards. It is an ideal solution for customers standardizing on SOAP/HTTP as their protocol of choice for integrating applications.

SALT supports most of the common Web services standards including SOAP 1.1 and 1.2, SOAP with attachments, MTOM, WSDL 1.1, WS-Policy, WS-Addressing, WS-ReliableMessaging, WS-Security 1.0 and 1.1, WS-Coordination, and WS-AtomicTransaction 1.0. This allows integration with most other Web services platforms including Oracle WebLogic Server, Axis for Java 2.0, Microsoft .NET framework 3.0, Oracle Service Bus, and Oracle BPEL, as well as most other popular SOAP/HTTP offerings. To simplify the use of security, SALT 12c now supports credential propagation via SAML. This can be used to support a single sign-on capability.

Accessing Oracle Tuxedo services by SOAP/HTTP

SALT provides a SOAP/HTTP gateway that accepts incoming SOAP requests and transforms the request into a native Oracle Tuxedo request. All Oracle Tuxedo buffer types are supported which allows virtually any Oracle Tuxedo service to be accessed via SOAP/HTTP. The gateway operates by using information stored in the Oracle Tuxedo service metadata repository that describes the Oracle Tuxedo service to be called. This information is also used to automatically generate a WSDL file that can be accessed via HTTP from the gateway.

"all the GISP services, running on Tuxedo, are managed by SALT. GISP manages the provision of all the mobile service for Telecom Italia, so you can understand how much the system is strategic for Telecom Italia" – Fabio Di Vita (Oracle Technical Lead in Telecom Italia)

Accessing external SOAP/HTTP services

The SALT gateway can also provide transparent access to external Web services. The gateway advertises Oracle Tuxedo services as proxies to the external SOAP/HTTP service that can be called as a normal Oracle Tuxedo service using FML32 buffers. When the gateway receives a request to one of the proxy services, the gateway transforms the FML32 buffer into the appropriate SOAP payload and then calls the external Web service.

Data transformation

In some cases it may be desirable to change the shape of the SOAP payload or to perform other custom data mapping. SALT provides a plug-in mechanism that allows customers to develop their own custom mapping between the SOAP payload and the corresponding Oracle Tuxedo buffer type.

RESTful Web Services

A major new feature in SALT 12cR2 is the support of REST based Web services. This allows the use of HTTP to directly call Tuxedo services or for Tuxedo applications to be able to make HTTP requests to external applications. The HTTP verbs GET, POST, DELETE, and PUT are mapped to Tuxedo service names. For inbound requests, HTTP content, either XML or JSON, is mapped to the specified Tuxedo buffer type, while the HTTP header information is placed into the Tuxedo header information that can be retrieved with the tpgetcallinfo() API. Upon return from the service, the reply buffer is mapped back into XML or JSON and then returned to the caller.

Similarly for outbound calls, service names associated with the HTTP verbs are advertised by the SALT gateway. Tuxedo applications call these services with the appropriate Tuxedo buffer type and the buffer is converted to either XML or JSON before the corresponding HTTP request is made. The response from the external service then transformed from XML or JSON into a Tuxedo buffer and delivered back to the caller.

SERVICE BUS INTEGRATION

Oracle Service Bus (OSB) is an enterprise service bus that supports the integration, mediation, and monitoring of a variety of types of services. Client applications call OSB provided proxies. Once the request is received it is routed via a pipeline through various stages to a business service that is provided by an external system. The stages in the pipeline can transform the contents of the message, make routing decisions based upon header and message contents, and enrich the contents of the message. Mediation allows clients to use one protocol and interaction style while the business service uses a completely different protocol and interaction style.

For each type of service supported by OSB a transport is used to handle the connection and interfacing to the specifics of the service. Standard transports like SOAP/HTTP, JMS, and file transports, are provided as well as a transport for Oracle Tuxedo. This transport allows Oracle Tuxedo applications to be easily integrated into environments using OSB. The Oracle Tuxedo transport supports Oracle Tuxedo based business services that can be accessed from any OSB proxy as well as Oracle Tuxedo proxy services which allow Oracle Tuxedo applications to call services mediated by OSB.

Used in conjunction with other transports, the Oracle Tuxedo transport allows SOAP, JMS, SMTP, REST, SAP, WebSphere MQ, and many other clients to transparently access Oracle Tuxedo based applications. Oracle Tuxedo applications can also access services offered by these same systems as though they were native Oracle Tuxedo applications.

WHICH ONE TO CHOOSE

With such a variety of integration choices, it can be difficult to choose the most suitable integration mechanism. Generally the first decision is whether a client only integration option is sufficient to meet the business requirements. The client only solutions can be used to access Oracle Tuxedo services and can start delegated transactions that do not participate in the local systems transaction management environment. If a client only option meets the business requirements, then the choice is typically determined by what the client environment will support and in the case the where the client environment supports multiple options, the choice is usually a trade-off between standards based communication or higher performing non-standards based binary protocols. For example a Java client may have three options; Jolt, WTC, and SALT. Jolt and WTC use proprietary APIs and protocols but the protocols are binary protocols providing excellent performance. SALT uses SOAP/HTTP which will incur additional processing overhead to convert to and from XML.

The table below provides a quick summary of the various options and the features they support.

Option	Client Only or Client/Server	Standards Based	Synch or Asynch	Transaction Support	Binary Protocol
Jolt	Client	No	Synch	Delegated	Yes
WTC	Client/Server	No	Both	Distributed	Yes
JCA Adapter	Client/Server	Yes	Both	Distributed	Yes
.NET Client	Client	No	Both	Delegated	Yes
MQ Adapter	Client/Server	No	Async	Distributed	Yes
SALT	Client/Server	Yes	Both	Delegated	No
OSB	Client/Server	Yes	Both	Distributed	Yes
TMA	Client/Server	No	Both	Distributed	Yes

Option	Custom Data Transformation	Service Mediation	Multiplexed Connections
Jolt	No	No	Yes
WTC	No	No	Yes
JCA Adapter	No	No	Yes
.NET Client	No	No	No
MQ Adapter	No	No	Yes
SALT	Yes	No	No
OSB	Yes	Yes	Yes
TMA	No	No	Yes

Environment	Jolt	WTC	JCA	.NET Client	MQ Adapter	SALT	OSB	TMA
Standalone Java	Y				Y	Y	Y	
Oracle WebLogic Server	Y	Y	Y		Y	Y	Y	
Java EE	Y		Y		Y	Y	Y	
.NET/ASP				Y	Y	Y	Y	
IBM CICS or IMS					Y			Y
E-Business Suite						Y	Y	
JD Edwards						Y	Y	
Siebel						Y	Y	
PeopleSoft	Y					Y	Y	
SAP					Y	Y	Y	

The above tables should help in determining what integration options are available and what options would best fit a specific set of requirements.

CONCLUSION

At an Oracle OpenWorld 2008 session on high availability, an architect for a leading financial institution with nearly $2T under management said: "In 15 years of using Tuxedo for our applications, we have never had a production outage. "

As one of the original SOA platforms, Oracle Tuxedo provides many ways to connect with other SOA platforms and environments. Applications based on Oracle Tuxedo can transparently interact with most commonly used application platforms. The combination of excellent SOA connectivity options, extreme performance, high reliability, and linear scalability, makes Oracle Tuxedo the obvious choice for C, C++, Java, COBOL, Python, Ruby, and PHP applications.

Oracle Tuxedo in a SOA World
May 2014
Author: Todd Little

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com