

# ORACLE SERVICE ARCHITECTURE LEVERAGING TUXEDO

## KEY FEATURES AND BENEFITS

### FEATURES

- Web services calls to and from Oracle Tuxedo with extensible data mapping
- RESTful Web services with JSON, XML or HTML payload
- Auto transformation to/from Tuxedo buffer types
- Easy-to-use Tuxedo Services Console for all configuration needs
- Support for custom http headers
- SOAP/http with WS-specs implementations
- MTOM support for binary data
- WS-Security , WS-AT, WS-RM
- Single-sign-on with other applications when using Oracle Access Manager
- SAML Single Sign-On
- XML schema for custom buffer types
- Data transformation tracing
- SOCKS proxy enabled
- MIB and dynamic configuration
- Web based configuration tool
- Integrated monitoring tools
- ECID propagation

*Oracle Service Architecture Leveraging Tuxedo (SALT) is a standards-based, bi-directional Web services gateway. SALT provides transparent and reliable access to existing Oracle Tuxedo services as RESTful and SOAP/http Web services with a high-performance, easy-to-use, and configuration-driven model. In addition, Oracle Tuxedo applications can call external RESTful and other Web services as Tuxedo services, with the gateway seamlessly transforming the call using HTTP protocol. The result is widespread access to applications using Web services without any Web services specific coding, enabling interoperability among distributed applications that span diverse hardware and software platforms.*

### SOA-Enable Oracle Tuxedo Applications Without Re-Coding

Oracle SALT allows organizations to integrate mission-critical applications written in C, C++, COBOL and Java into service-oriented architectures (SOA) without costly rewrites . Leveraging a configuration-driven approach eliminates the need for programming changes and allows companies to use existing enterprise assets and to lower their total cost of ownership (TCO).

As a native Oracle Tuxedo Web services implementation, Oracle SALT reduces the run-time conversions that might exist with other solutions for accessing Oracle Tuxedo services.

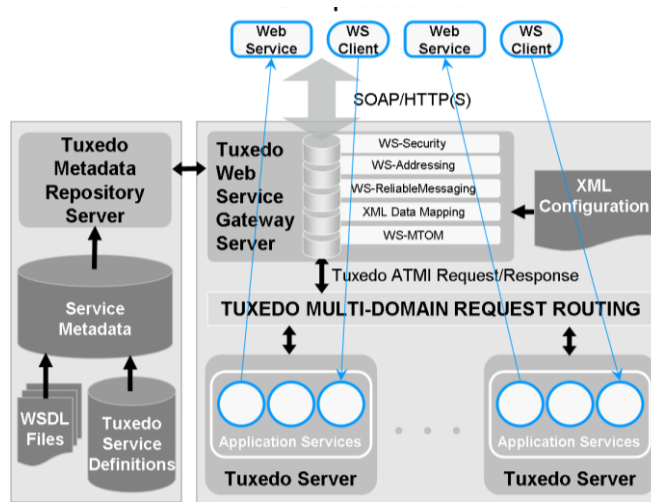
### Create an Extensible Web Services Architecture

Oracle SALT and Oracle Tuxedo allow enterprises to develop composite (or hybrid) end-to-end solutions that leverage the scalability and robustness of Oracle Tuxedo combined with the extensibility of SOA. Uniting these two environments simplifies a broad spectrum of transaction-processing solutions via powerful, open standards-based APIs. With a high-performance, easy-to-use, configuration-driven model for bidirectional Web services integration, developers can publish existing Oracle Tuxedo services as standard Web services using the SOAP/HTTP(S) protocol.

With Oracle SALT, Oracle Tuxedo services can transparently call external Web services as if calling another native Tuxedo service. In addition to basic Web services protocols, the application complies with most primary Web services specifications, including SOAP, Web Services Description Language (WSDL), WS-Addressing, WS-Security, WS-AtomicTransaction, and WS-ReliableMessaging. This improves TCO by leveraging existing skill sets, tools, and information assets.

**BENEFITS**

- Enable interoperability among distributed applications that span diverse hardware and software platforms
- Simplify access to Oracle Tuxedo applications using standard Web services protocols
- Reduce complexity and time-to-market for new Web services with configuration driven approach
- Atomic transactions ensure data consistency
- Facilitate the development of heterogeneous distributed applications with a cross-platform, cross-language data model
- Lower your TCO by leveraging existing IT assets without the need for costly rewrites



**Figure 1. Oracle SALT in an enterprise services network**

Oracle SALT can be integrated with Oracle Service Registry and Oracle Enterprise Repository to publish Tuxedo services metadata for broad access within the enterprise, enabling their use in Oracle BPEL PM, Oracle Business Rules, and Oracle Service Bus as well as any 3<sup>rd</sup> party users of the Registry and Repository.

Web Services Architecture Features and Benefits	
Bi-directional calls to Web services	Allows Oracle Tuxedo application services to be accessed via Web services, and Tuxedo services to call external Web services
Configuration-driven deployment	Supports a configuration-driven style of deployment allowing you to specify a list of Oracle Tuxedo services that can be exposed as Web services
Metadata repository	Maintains Tuxedo service contract metadata and automates the creation of WSDLs for publishing Tuxedo services as Web services.
Web services gateway (GWWS)	Handles Web services protocols and performs automatic SOAP message conversion to and from Tuxedo buffers; acts as an Oracle Tuxedo gateway process and can be managed in the same manner as general Oracle Tuxedo servers
WSDL document generation and publishing	Automatically generates WSDL documents that can be integrated using Web services development tools or published to a Universal Description, Discovery, and Integration (UDDI) server
Seamless upgrade	Enables use of existing WSDL, configuration files, and service metadata without any manual changes to upgrade to newer versions
MTOM support for binary data	Implements optimized MIME multiple/related serialization of SOAP messages for Oracle Tuxedo and X_OCTET CARRAY buffers
Multiple encodings for inbound requests	Supports multiple character encodings in addition to previously supported UTF-8 encoding for SOAP messages
Forward user data with TPFALL	Provides capability to send user data back to the Web services client in a TPFALL instance by way of a SOAP fault message

WS-AtomicTransaction implementation	Supports WS-AtomicTransaction and WS-Coordination to allow Tuxedo applications to participate in or coordinate atomic Web services transactions
XML schema for custom buffer types	Supports the use of XML schema to describe the content of custom buffer types to convert custom buffers to and from XML
XML validation	Delivers the mechanism to describe format/content of XML fragments
Custom http headers	Allows application to customize http headers for additional flexibility in application architecture when accessing SOAP/http Web services

RESTful Web services allow 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.

<b>RESTful Web Services Features and Benefits</b>	
Bi-directional calls to Web services	Allows Oracle Tuxedo application services to be accessed as RESTful Web services, and Tuxedo services to call external RESTful Web services
Configuration-driven deployment	Supports a configuration-driven style of deployment allowing you to specify a list of Oracle Tuxedo services that can be exposed as RESTful Web services. And specify list of external RESTful Web services to be accessed from Tuxedo application
Metadata repository	Maintains Tuxedo service contract metadata, which helps automate creation of RESTful Web services.
Web services gateway (GWWS)	Handles http protocol and conversion to and from Tuxedo buffers; acts as an Oracle Tuxedo gateway process and can be managed in the same manner as general Oracle Tuxedo system servers
Custom http headers	Allows application to customize http headers for additional flexibility in application architecture when accessing RESTful Web services
CRUD support	Allows mapping to Create, Read, Update, Delete functions which maps to http's POST, GET, PUT, DELETE respectively
JSON, XML and HTML	RESTful web services can accept JSON, XML and HTML payload, which is transformed into Tuxedo buffer types by SALT gateway for incoming requests. For outgoing requests Tuxedo buffer is transformed into JSON, XML or HTML payloads to access external RESTful Web services

### **Optimize Performance, Scalability, and Interoperability**

Oracle SALT improves performance with significantly faster response times while offering full support for multiple interaction styles, including inbound/outbound, synchronous/asynchronous, and request/response. SALT supports XML and key WS standards, and interoperates with multiple Web services platforms, including Oracle

**RELATED PRODUCTS AND SERVICES**

Oracle Tuxedo delivers a robust platform to run high-volume applications across distributed, heterogeneous computing environments, enabling transactions that stretch from customer-facing, business-critical applications to back-office processes, across any system, anywhere in the world.

**RELATED PRODUCTS:**

- Oracle Tuxedo
- Oracle Tuxedo System and Applications Monitor (TSAM)
- Oracle Tuxedo Mainframe Adapters
- Oracle Tuxedo JCA Adapter
- Oracle Tuxedo Message Queue
- Oracle Tuxedo Application Rehosting Workbench
- Oracle Tuxedo Application Runtime for CICS and Batch
- Oracle Tuxedo Application Runtime for IMS
- Oracle Tuxedo Application Runtime for Batch

WebLogic Server, Oracle Service Bus, Apache Axis, and .NET.

<b>Performance, Scalability and High Availability Features and Benefits</b>	
Asynchronous and reliable messaging	Provides asynchronous communication model and complies with WS-Addressing specification; also supports reliable message delivery conforming to WS-ReliableMessaging specification
Scalability	A single Oracle Tuxedo domains can support thousands of concurrent clients using services that generate external Web services calls
Overlapped requests	Handles multiple concurrent requests with high throughput – sufficient to support all application services in an Oracle Tuxedo domain
Scalability	Can have multiple instances of the SALT Gateway process to handle same set of Web services requests
Runtime Stats	In addition to command line tools, provides programmable management information base (MIB) API for configuration and runtime statistics
Dynamic configuration	Allows configuration to be changed without having to restart SALT Gateway eliminating any downtime requirement
Tracing	Extensive tracing of data transformation to/from Tuxedo buffer types
ECID propagation	Allows Web service invocations to be traced across Oracle product stacks
Monitoring	Allows monitoring of Web services via Oracle Enterprise Manager

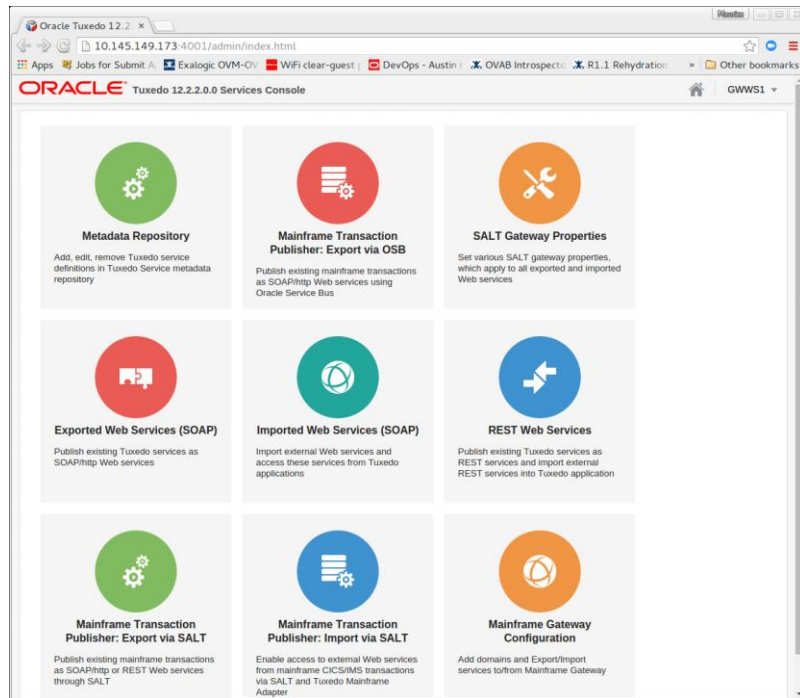
**Security**

Oracle SALT uses the Oracle Tuxedo security framework for authentication, protects point-to-point transactions using Secure Socket Layer (SSL) protocols, and can route traffic through SOCKS proxies.

<b>Security and Platform Support Features and Benefits</b>	
HTTP basic authentication	The gateway server supports Oracle Tuxedo user profiles passed from the Web services client via HTTP basic authentication protocol.
WS-Security support	The gateway server supports the WS-Security and WS-SecurityPolicy protocols to allow inbound and outbound authentication by username and X509 tokens, and digitally signing messages.
Single Sign-On	The gateway server also supports SAML 1.1 and 2.0 with sender vouches confirmation method to allow single sign-on authentication.
SSL link-level security	The gateway server supports HTTPS requests and link-level security, enabled by specifying an SSL host and port number in the Oracle SALT configuration file.
SOCKS proxy	The gateway server can use a SOCKS proxy server to establish connections, adding an extra layer of protection.

**Web browser based Tuxedo Services Console**

Oracle SALT provides configuration-only way to expose or access Web services. The Tuxedo Services Console, embedded in the base product, provides a point and click approach to create the required configuration to expose Tuxedo services as Web services, eliminating the need to understand structure of the configuration files and manually edit those.



**Figure 2. Oracle Tuxedo Services Console interface**

Oracle SALT uses the service metadata repository for Tuxedo service definitions. Using the Tuxedo Services Console, service definitions for Tuxedo services can be automatically populated or edited. Tuxedo services are then grouped as operations to become part of a Web service.

Tuxedo Services Console Features and Benefits	
Out-of-the-box experience	The console is embedded into the Oracle SALT gateway and does not require any additional product or add-on installation.
Service Definition Discovery	For discovery and automatic population of service definitions in the metadata repository.
Service Definition Editor	A service metadata editor to add/edit/delete service definitions directly.
Web Service Definitions Editor	Web service definitions group Individual services (operations) into an actual Web Service.  Policies can be configured here at the service, operation or individual message level.
Security	SSL and authentication can be used to protect access to the configuration tool.
Import WSDLs	Imports WSDLs to create configuration for accessing external Web services
Import RESTful Web services	Import JSON definitions so that these external RESTful Web services can be accessed from Tuxedo applications
Test client	A simple test client for testing RESTful Web services
Property Editor	Setup SALT gateway properties

**Contact Us**

For more information about Oracle SALT, please visit [oracle.com](http://oracle.com) or call +1.800.ORACLE1 to speak to an Oracle representative.



| Oracle is committed to developing practices and products that help protect the environment

Copyright © 2010-2016, Oracle and/or its affiliates. All rights reserved.  
This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.  
Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.