

Oracle® Service Architecture Leveraging Tuxedo (SALT)

Product Overview

11g Release 1 (11.1.1.2)

April 2011

ORACLE®

Copyright © 2006, 2011, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Oracle SALT Overview

Understanding Oracle SALT	-1
Understanding Oracle SALT Web Services.	-2
What Are Web Services?	-2
Why Use Oracle SALT?	-2
Understanding the Oracle SALT SCA Container	-3
Understanding Oracle SALT Web Application Server	-4
Oracle SALT Release History	-5
Release 1.1.	-6
Release 2.0.	-6
Release 10g R3 (10.3)	-6
Release 11g R1 (11.1.1.1.0).	-7
Release 11g R1 (11.1.1.2.0).	-8
Release 11g R1 (11.1.1.2.2).	-9
Oracle SALT Components	-9
Oracle SALT Gateway (GWWS)	-10
WSDL Assistant Utilities	-10
SCA Container APIs and Utilities.	-11
Oracle SALT Web Server Plug-ins	-11
Oracle SALT Use Cases	-11
Use Case 1: Exposing Native Oracle Tuxedo Services as Web Services	-12
Use Case 2: Invoking Web Services from Oracle Tuxedo Applications.	-12
Use Case 3: Connecting Oracle Tuxedo Domains Using SOAP Protocol	-13
Use Case 4: SCA to SCA Communication	-15
Use Case 5: SCA Components Calling an Existing Oracle Tuxedo Service	-15
Use Case 6: Oracle Tuxedo ATMI Calling SCA Components	-16
Use Case 7: Ruby Web Applications.	-16

Use Case 8: Python Web Applications	-17
Use Case 9: PHP Web Applications	-17
Configuring Web Services with Oracle SALT	-17
Invoking Oracle Tuxedo Services Using Web Service Client Toolkits	-17
Invoking Web Services Using Oracle Tuxedo Programming Interfaces	-18
Oracle SALT Supported Standards.	-18
SOAP Standards	-18
SCA and SDO Standards.	-20
What's Next?	-21

Oracle SALT Overview

The following sections provide an overview to the Oracle SALT product:

- [Understanding Oracle SALT](#)
- [Oracle SALT Release History](#)
- [Oracle SALT Components](#)
- [Oracle SALT Use Cases](#)
- [Configuring Web Services with Oracle SALT](#)
- [Oracle SALT Supported Standards](#)
- [What's Next?](#)

Understanding Oracle SALT

Oracle Service Architecture Leveraging Tuxedo (SALT) is an add-on product option for Oracle Tuxedo, enabling Oracle Tuxedo applications to participate in SOA environments. Oracle SALT has two major components: native Web services stack and SCA container.

Oracle SALT allows external Web services applications to invoke Oracle Tuxedo services as Web services, and Oracle Tuxedo applications to invoke external Web services. Oracle SALT does not require any coding to achieve this. In addition, Oracle SALT includes an SCA container, which allows you to develop new SOA applications focusing on business logic, while still taking advantage of Oracle Tuxedo infrastructure. SCA container also helps with effective reuse of existing application assets.

Understanding Oracle SALT Web Services

Oracle SALT complies with standard Web service specifications (SOAP 1.1, SOAP 1.2, and WSDL 1.1), allowing Oracle SALT to interoperate with other Web service products and Oracle SALT Overview development toolkits. Oracle Tuxedo applications can easily integrate with Web services applications using Oracle SALT.

What Are Web Services?

Web services are a set of functions packaged into a single entity made available to other systems on a network. They can be shared and used as a component of distributed Web-based applications. The network can be a corporate intranet or the Internet. Other systems, such as customer relationship management (CRM) systems, order-processing systems, and other existing back-end applications, can call these functions to request data or perform an operation. Because Web services rely on standard technologies which most systems provide, they are an excellent means for connecting distributed systems together.

The software industry has evolved toward loosely coupled service-oriented applications that interact dynamically over the Web. The applications break down the larger software system into smaller modular components, or shared services. These services can reside on different computers and can be implemented by vastly different technologies. They are packaged and made accessible using standard Web protocols, such as XML and HTTP.

Web services share the following properties that make them easily accessible from heterogeneous environments:

- Web services are accessed using widely supported Web protocols such as HTTP.
- Web services describe themselves using an XML-based description language.

Web services communicate with clients (both end-user applications or other Web services) through simple XML messages that can be produced or parsed by virtually any programming environment or manually, if necessary.

Why Use Oracle SALT?

Oracle SALT is a native Oracle Tuxedo Web service integration solution. It reduces Oracle Tuxedo/Web service integration costs and decreases conversion processes that may exist with other solutions for accessing Oracle Tuxedo services. It enables seamless connectivity between Oracle Tuxedo applications and external Web service applications.

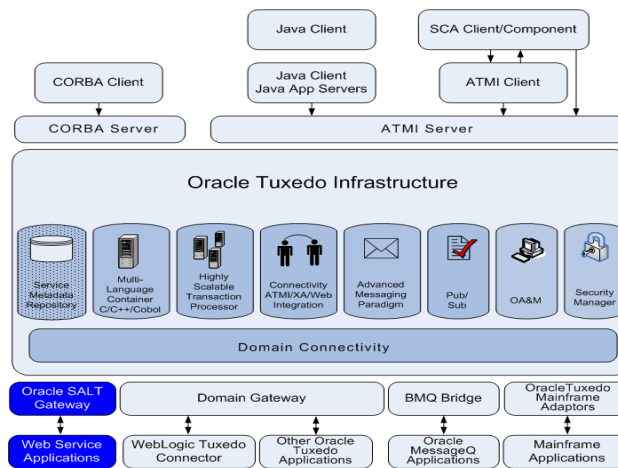
Oracle SALT allows existing Oracle Tuxedo services (inbound) to be easily exposed as Web services without additional programming tasks. It also allows you to create native Oracle Tuxedo applications that access external Web services (outbound) transparently.

Major Web services benefits include:

- Interoperability among distributed applications that span diverse hardware and software platforms
- Easy, widespread access to applications using Web protocols
- A cross-platform, cross-language data model (XML) that facilitates developing heterogeneous distributed applications

Figure 1 illustrates how the Oracle SALT gateway is used in the Oracle Tuxedo framework.

Figure 1 Oracle SALT Gateway/Oracle Tuxedo Infrastructure



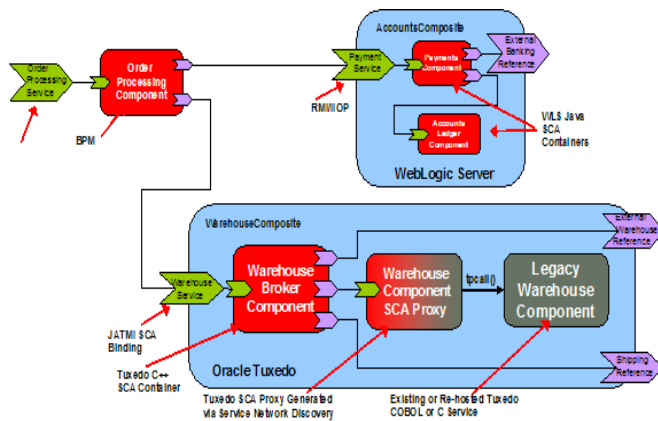
Understanding the Oracle SALT SCA Container

Oracle SALT provides an SCA Container for new application development. The SCA container, which is based on standard SCA programming and assembly model, allows customers to focus on business logic without having to learn many middleware APIs. Use of SCA standard allows you to use SCA tools to develop and assemble applications. The SCA container still leverages all the benefits of Oracle Tuxedo infrastructure, such as reliability, availability, scalability and performance.

SCA container also improves interoperability and extensibility of existing and new Oracle Tuxedo applications by making it easier to interoperate with SOA environments.

Figure 2 describes an application based on SCA assembly and programming model. The application contains many services, offered by SCA components. These components include components hosted in SCA container as well as legacy components accessed from SCA components.

Figure 2 SCA Application



Understanding Oracle SALT Web Application Server

Oracle SALT provides an HTTP server plug-in (`mod_tuxedo` for Apache/Oracle HTTP Server and `tux_nsapi` for iPlanet) that forwards HTTP requests to Tuxedo services. Replies from Oracle Tuxedo services are then sent back to HTTP clients such as web browsers by the same plug-in.

In addition, Oracle SALT provides a System Server (`WEBHNDLR`) that is capable of running PHP, Python and Ruby web applications, such as Symfony (PHP), Django (Python) and Rails (Ruby). HTTP requests can also be handled directly by regular C/C++ Tuxedo servers using the CGI-like format which is counterpart to the HTTP server plug-in

Oracle SALT Web Application server provides the following support:

- Apache 2 Web Server Plug-in Support

Web Server plug-ins in Apache 2 format, for interfacing Apache HTTP server 2, Oracle HTTP Server or iPlanet Server and Oracle Tuxedo-based web applications.

- Web Gateway Interface Support

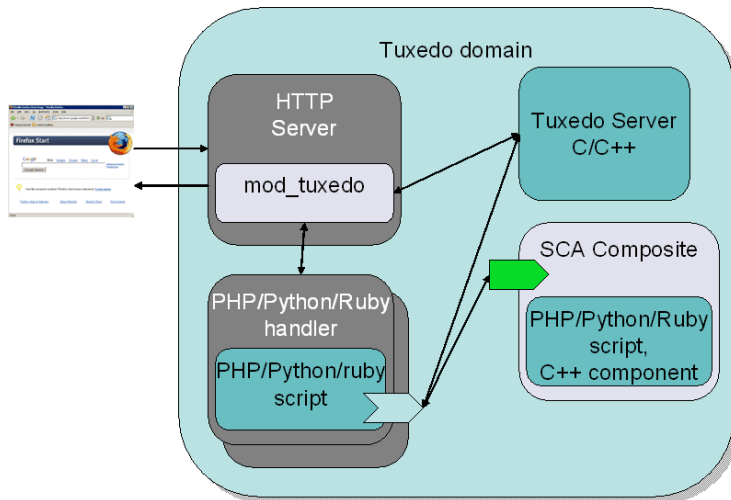
Support of a Web Gateway Interface in order to develop Oracle Tuxedo services that generate dynamic HTML pages.

- Web Server Plug-in Support

Web Server plug-ins in Apache 2 format, for interfacing Apache HTTP server 2, Oracle HTTP Server or iPlanet Server and Oracle Tuxedo-based web applications.

Figure 3 shows a high-level view of the Oracle SALT Web application server architecture.

Figure 3 Oracle SALT Web Application Server Architecture



Oracle SALT Release History

Oracle SALT is the latest add-on to the Oracle Tuxedo product family. Developed in 2006, Oracle SALT is designed to provide a seamless Oracle Tuxedo solution of integrating Oracle Tuxedo applications and standard Web services application. With the addition of an SCA container, SALT is also designed to better integrate existing Oracle Tuxedo applications in SOA environments, as well as design better SOA applications from the ground up with SCA.

Release 1.1

Release 1.1 is the initial Oracle SALT release. Made available in 2006, SALT 1.1 introduced the following major features:

- Inbound Service Support
 - Permits Web service applications to invoke native Oracle Tuxedo services
- HTTP and HTTP over SSL Transport Support
- Asynchronous and Reliable Messaging Support

Release 2.0

The Oracle SALT 2.0 release incorporates significant enhancements based on the SALT 1.1 release. SALT 2.0 introduced the following features:

- Outbound Service Support
- Extended WS-* Standards Support
- SOAP Message Transmission Optimization Mechanism Support (MTOM)
- Oracle Tuxedo TPFAIL Support for Web Services
- Extensible Data Type Mapping and Message Conversion
- Multiple Encoding Support
- Configuration-Driven Deployment
- Leveraging the Oracle Tuxedo Service Metadata Repository
- Data Type Mapping and Message Conversion
- Asynchronous and Reliable Messaging
- Web Service Security Support

Release 10g R3 (10.3)

Oracle SALT 10g Release 3 (10.3) introduced the following features:

- Service Component Architecture (SCA) Programming

SCA provides a new programming model that aims at simplifying component re-use and seamless communications between components. The SALT 10g Release 3 (10.3) SCA container enables new programming model and leverages the most valued Oracle Tuxedo features (such as reliability, availability, scalability, and performance). Oracle SALT 10g Release 3 (10.3) introduces the following SCA features:

- Client-side binding for SCA invocations over ATMI and SOAP
 - Server-side binding for serving SCA requests made over ATMI and SOAP
 - Client-side binding for SCA invocations from Java environments
 - Development and runtime tools: Commands to build and deploy SCA clients and servers as well as commands for runtime administration. For more information, see the [SALT 10g Release 3 \(10.3\) Command Reference Guide](#).
 - Authentication and authorization for SCA services
 - Global transactions
 - Thread-safe SCA/SDO clients and servers
 - SCDL schema validation
 - Support for simple data types
 - Support for complex data types using SDO
 - Automatic data transformation to/from Oracle Tuxedo buffer types
 - Support for multi-byte characters using multiple encoding
- **Service Contract Discovery**

Automatically discover service contract information at run time. The generated information can be put into metadata repository automatically or to a file which can then be loaded manually into the metadata repository using the `tmloadrepos` utility. For more information, see [Configuring an Oracle SALT Application](#) in the *Oracle Salt Administration Guide*.
 - **Access Log for All Incoming Requests**

Assists Oracle Tuxedo client administrators to monitor application validity at runtime. You can record application high water client count, current client count, and named users.

Release 11g R1 (11.1.1.1.0)

Oracle SALT Release 11g R1 (11.1.1.1.0) introduced the following features:

- Python and Ruby Support

Python and Ruby support in SALT SCA provides a simple SCA client API for each language to perform SCA calls from Python or Ruby client programs, and an SCAHOST system server to allow access to Python or Ruby components. Python, Ruby or C++ components can now call or be called to/from other Python, Ruby or C++ components.

For more information, see [Oracle SALT SCA Programming](#) in the Oracle SALT Programming guide.

- SCA Structure support

Provides additional C++ structure functionality for improved performance.

For more information, see [Oracle SALT SCA Programming](#) in the Oracle SALT Programming guide.

- Scatuxgen Metadata Generation Tool

The Scatuxgen Metadata Generation Tool parses C++ interfaces as used to develop SCA components for the SCA runtime functionality introduced in SALT 10gR3. It generates Oracle Tuxedo metadata repository interface data, and optionally a Web Services Definition File (WSDL) document.

For more information, see the [Oracle SALT Command Reference Guide](#).

- WS-TX Support

Provides bi-directional transactional interoperability between Web Services and Oracle Tuxedo applications. Applications transparently make use of the GWWS system server transactional capabilities.

Outbound XA transactions are transparently propagated outside of an Oracle Tuxedo domain and mapped one-to-one to WS-TX transactions. Inbound WS-TX transactions are propagated into an Oracle Tuxedo domain and mapped to XA transactions.

For more information see, [WS-TX Support](#) in Oracle SALT Interoperability.

Release 11gR1 (11.1.1.2.0)

Oracle SALT 11g Release 11g R1 (11.1.1.2.0) introduced the following features:

- SOCKS Proxy Support

SOCKS proxy support provides a configurable element associated to outbound endpoints and allows configuring a SOCKS server address that will proxy outbound connections according to the SOCKS V4 and V5 protocol.

Inbound support is not necessary as SOCKS proxies incoming connections without listening endpoints being aware of it. For more information, see the [Oracle SALT Deployment File Reference](#) in the Oracle SALT Reference Guide.

The `wSDLcvr` command is enhanced to connect to a SOCKS proxy server. For more information, see the [Oracle SALT Command Reference](#) in the Oracle Salt Reference Guide.

Release 11g R1 (11.1.1.2.2)

Oracle SALT 11g Release 11g R1 (11.1.1.2.2) introduces the following features:

- Apache 2 Web Server Plug-in Support

Web Server plug-ins in Apache 2 format, for interfacing Apache HTTP server 2, Oracle HTTP Server or iPlanet Server and Oracle Tuxedo-based web applications.

- Web Gateway Interface Support

Support of a Web Gateway Interface in order to develop Oracle Tuxedo services that generate dynamic HTML pages.

- PHP, Python and Ruby Web Scripts Support

- PHP scripts supported directly
- Python scripts supported via WSGI
- Ruby scripts supported via Rack
- PHP, Python and Ruby applications using most web frameworks such as Zend framework, Symfony, Django or Rails run as-is in an Oracle Tuxedo environment.

- PHP SCA Script Support

Support for PHP scripts as SCA client and SCA components, for SOA integration.

For more information, see the [Oracle SALT Administration Guide](#).

Oracle SALT Components

Oracle SALT consists of the following major components:

- [Oracle SALT Gateway \(GWWS\)](#)
- [WSDL Assistant Utilities](#)
- [SCA Container APIs and Utilities](#)

- [Oracle SALT Web Server Plug-ins](#)

Oracle SALT Gateway (GWWS)

The Oracle SALT provided Oracle Tuxedo system server (GWWS), connects with other Web service applications via SOAP over HTTP/S protocol. The GWWS server acts as an Oracle Tuxedo gateway process and is managed in the same manner as general Oracle Tuxedo system servers. Each GWWS server has bi-directional (inbound/outbound) capability. The GWWS server:

- accepts SOAP requests from Web service applications and issue Oracle Tuxedo native calls to Oracle Tuxedo services.
- accepts Oracle Tuxedo ATMI requests and issues SOAP calls to Web Service applications.

You can have multiple GWWS instances in one Oracle Tuxedo domain. The same functionality for multiple GWWS instances is provided by specifying the same Oracle SALT configuration to improve throughput and failover protection. You can also group multiple GWWS instances in different configuration files for different purposes.

When the GWWS server boots, it loads the specified SALT configuration file and Oracle Tuxedo service contract information from the Oracle Tuxedo Service Metadata Repository.

The GWWS server also acts as a simple HTTP Web server for WSDL document and XML Schema file download.

WSDL Assistant Utilities

The Web Services Description Language (WSDL) is an XML-based specification that describes a Web service. A WSDL document describes Web service operations, input and output parameters, and how a client application connects to the Web service. Oracle SALT provides two utilities (`tmwsdlgen` and `wsdlcvt`) to map Oracle Tuxedo applications and Web Service WSDL descriptions.

WSDL Generator from Oracle Tuxedo Definitions

When using Oracle SALT to publish Oracle Tuxedo services as Web services, you do not need to compose a WSDL document manually; it is automatically generated as part of the SALT Web service development process. The generated WSDL document can be integrated using Web service development tools, or can be published to a UDDI server.

There are two ways to obtain a WSDL document:

- Use [tmwsdlgen](#) (the WSDL document file generating utility).
- Download the GWWS server generated WSDL document via HTTP(S).

WSDL Converter to Oracle Tuxedo Definitions

To support external Web Service applications, external WSDL documents need to be converted. The Oracle SALT conversion utility, [wsdlcvt](#), converts external WSDL documents to Oracle Tuxedo specific definition files (SALT Web Service Definition file, Oracle Tuxedo Service Metadata Repository Definition file and FML32 Field Table Definition file).

The SALT Web Service Definition file can be imported into a SALT Deployment file and utilized by a particular GWWS server. The Oracle Tuxedo Service Metadata Repository Definition file and FML32 Field Table Definition file provide service interface descriptions for Oracle Tuxedo client programming.

SCA Container APIs and Utilities

See the [Oracle SALT Command Reference Guide](#).

Oracle SALT Web Server Plug-ins

Oracle SALT dynamic languages system server (`WEBHNDLR`), takes requests sent by the Web server plug-in (`mod_tuxedo` or `tux_nsapi`) and processes them using PHP, Python or Ruby Web applications (no Oracle Tuxedo-specific API required).

- `mod_tuxedo`: this module forwards requests from Apache or Oracle HTTP Server to Tuxedo.
- `tux_nsapi`: this NSAPI plug-in forwards requests from iPlanet Web Server to Oracle Tuxedo.

Oracle SALT Use Cases

The following sections describe the most common Oracle SALT Web services use cases:

- [Use Case 1: Exposing Native Oracle Tuxedo Services as Web Services](#)
- [Use Case 2: Invoking Web Services from Oracle Tuxedo Applications](#)
- [Use Case 3: Connecting Oracle Tuxedo Domains Using SOAP Protocol](#)
- [Use Case 4: SCA to SCA Communication](#)

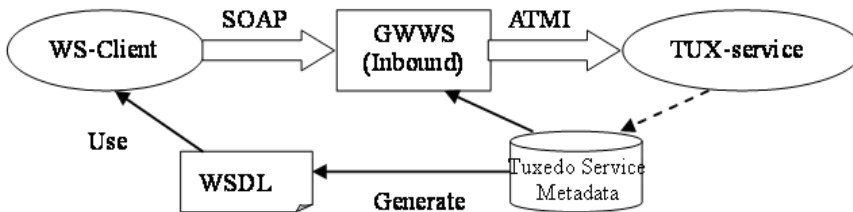
- Use Case 5: SCA Components Calling an Existing Oracle Tuxedo Service
- Use Case 6: Oracle Tuxedo ATMI Calling SCA Components
- Use Case 7: Ruby Web Applications
- Use Case 8: Python Web Applications
- Use Case 9: PHP Web Applications

Use Case 1: Exposing Native Oracle Tuxedo Services as Web Services

Native Oracle Tuxedo services can be exposed as Web services using standard Web service SOAP protocol. The GWWS server accepts SOAP requests through HTTP/S and then converts them into Oracle Tuxedo ATMI calls. SALT generates a WSDL document that describes the open standard Web service interfaces for Oracle Tuxedo services. The Oracle Tuxedo Service Metadata Repository is used to define Oracle Tuxedo service contract information. This is an “inbound” use case.

Figure 4 illustrates a generic inbound Web service call.

Figure 4 Exposing Oracle Tuxedo Services as Web Services (Inbound)



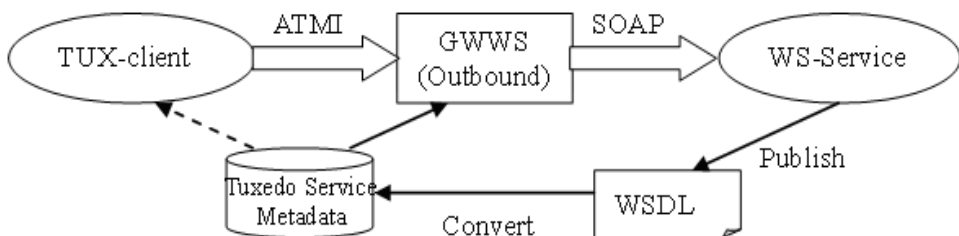
Use Case 2: Invoking Web Services from Oracle Tuxedo Applications

Web service applications can be imported into an Oracle Tuxedo domain, advertised as Oracle Tuxedo services through the GWWS server, and invoked from Oracle Tuxedo applications. SALT converts and maps each `wsdl:operation` as a particular Oracle Tuxedo service. The GWWS server advertises the mapped services (called SALT proxy services), and accepts Oracle Tuxedo ATMI requests from Oracle Tuxedo applications.

The Oracle Tuxedo Service Metadata Repository is used to store converted Oracle Tuxedo service contract information and helps Oracle Tuxedo programmers understand what type of Oracle Tuxedo buffers are expected for the imported SALT proxy services. This is an “outbound” use case.

Figure 5 illustrates a generic outbound Web service call.

Figure 5 Invoking Web Services from Oracle Tuxedo Applications (Outbound)



Use Case 3: Connecting Oracle Tuxedo Domains Using SOAP Protocol

Oracle SALT also allows you to connect two *different* Oracle Tuxedo domains using GWWS servers as an alternative to using /T domain. The GWWS server in the calling domain works in an outbound direction, the GWWS server in the receiving domain works in an inbound direction.

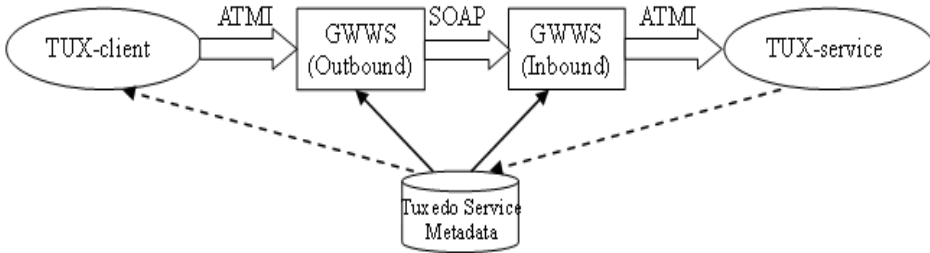
The receiving Oracle Tuxedo domain must propagate the Oracle Tuxedo service definition to the calling Oracle Tuxedo domain. This means that the calling domain Oracle Tuxedo Service Metadata Repository must contain the Oracle Tuxedo service definition file that runs in the receiving domain.

Note: This should be set up *manually*. The Oracle Tuxedo Service Metadata Repository infrastructure does not currently provide automatic propagation between Oracle Tuxedo domains.

The WSDL document is not required. Oracle SALT provides simple configurations to allow two GWWS servers to work together for domain connectivity using SOAP protocol without needing to exchange WSDL documents.

Figure 6 illustrates how to use Oracle SALT to connect two domains.

Figure 6 Connecting Two Oracle Tuxedo Domains with SOAP protocol



Two GWWS servers should not be used to create connections within the *same* Oracle Tuxedo domain, see [Figure 7](#). Also, a single GWWS server cannot connect to itself, see [Figure 8](#).

In either scenario, the GWWS server advertises the same Oracle Tuxedo services which are already advertised by other application servers. This might result in *dead-loop* service dispatching.

WARNING: It is strongly advised that you carefully plan and configure your Oracle SALT application to avoid these scenarios.

Figure 7 Two GWWS Servers Making a Connection Within the Same Oracle Tuxedo Domain

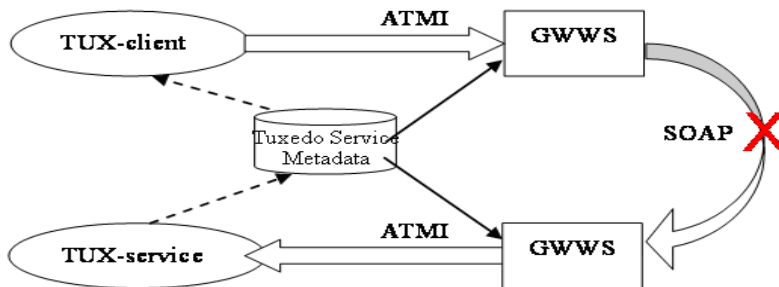
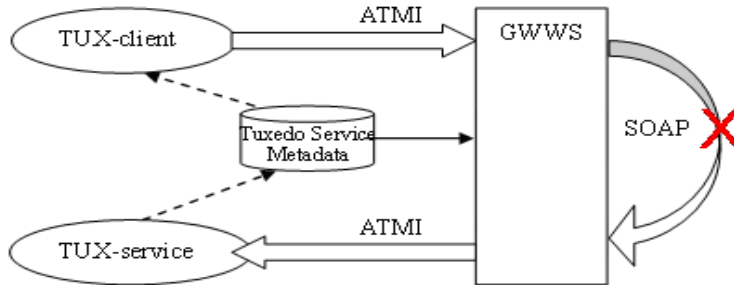


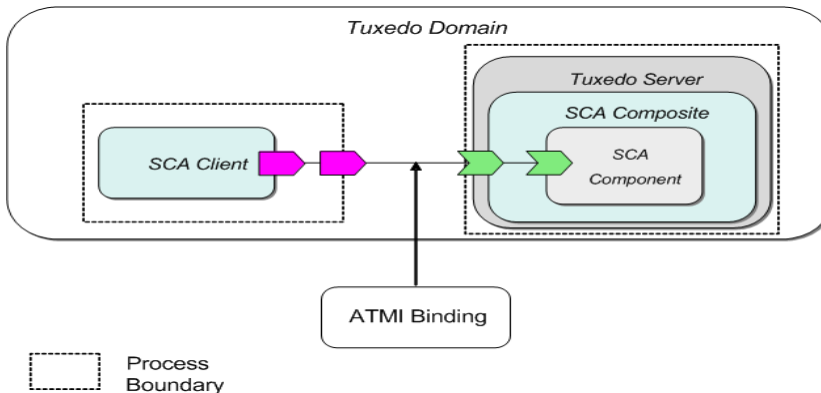
Figure 8 Single GWWS Server Making a Connection to Itself



Use Case 4: SCA to SCA Communication

Oracle SALT provides an infrastructure that allows developing components that conform to the Services Component Architecture (SCA) specification. These components may interact natively, or leverage the performance and high-availability of the Oracle Tuxedo framework, by communicating using a native ATMI binding, a WorkStation protocol based binding, or a Web-Services binding as shown in [Figure 9](#).

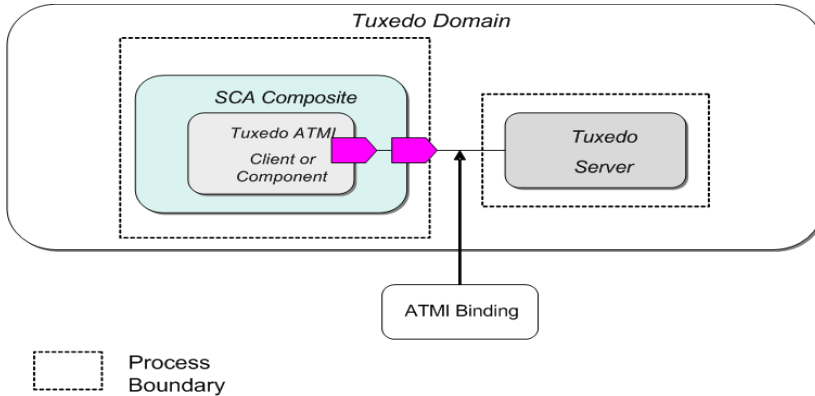
Figure 9 SCA to SCA Communication



Use Case 5: SCA Components Calling an Existing Oracle Tuxedo Service

Newly developed SCA components can interact with existing Oracle Tuxedo ATMI services by using the ATMI binding, as shown in [Figure 10](#).

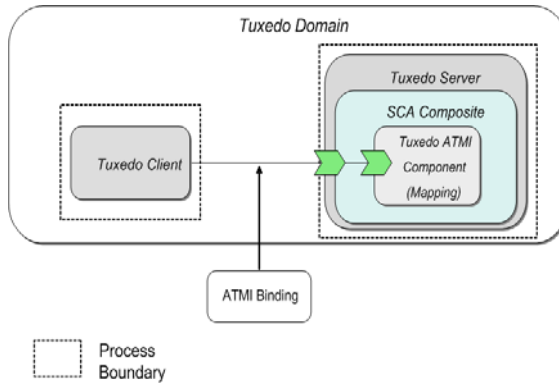
Figure 10 SCA Components Calling an Existing Oracle Tuxedo Service



Use Case 6: Oracle Tuxedo ATMI Calling SCA Components

Conversely, existing Oracle Tuxedo clients can interact with newly-developed SCA components by being exposed with the ATMI binding, as shown in [Figure 11](#).

Figure 11 Oracle Tuxedo ATMI Calling SCA Components



Use Case 7: Ruby Web Applications

A Ruby web application created using Rails can run directly inside Tuxedo without any code modification.

Use Case 8: Python Web Applications

A Python Web application created using Django can run directly inside Tuxedo without any code modification.

Use Case 9: PHP Web Applications

A PHP web application created using Symfony can run directly inside Tuxedo server without any code modification.

Configuring Web Services with Oracle SALT

The following steps are used typically when you configure Web services using Oracle SALT:

1. Configure Inbound Oracle Tuxedo Services
 - a. Define Oracle Tuxedo application services using the [Oracle Tuxedo Service Metadata Repository](#).
 - b. Compose one or more Oracle SALT Web Service Definition Files (WSDF).
2. Configure Outbound Web Services
 - a. Convert an external WSDL document into Oracle Tuxedo the following components: SALT Web Service Definition file, Oracle Tuxedo Service Metadata Repository Definition file, and FML32 Field Table Definition file.
 - b. Resolve potential naming conflicts for the auto-generated service names and FML32 field names.
3. Load all Oracle Tuxedo Service Definitions into the Oracle Tuxedo Service Metadata Repository using `tmloadrepos`.
4. Compose the Oracle SALT Deployment File for both inbound and outbound services.
5. Add the `TMMETADATA` and `GWWS` servers to your Oracle Tuxedo `UBBCONFIG` file.
6. Boot the Oracle Tuxedo application.

Invoking Oracle Tuxedo Services Using Web Service Client Toolkits

1. Client end user downloads the WSDL document file from the GWWS server.
2. Client end user generates client-side stubcode from the WSDL document file with a SOAP development kit.

3. Generate client-side program.
4. Run the client to invoke the Web service with SOAP messages.

Invoking Web Services Using Oracle Tuxedo Programming Interfaces

1. Create an Oracle Tuxedo client/server program according to the generated Oracle Tuxedo Service Metadata Definition file. The client program can be written in any Oracle Tuxedo supported client-side programming language (C/C++, Java, COBOL, .NET, and so on).
2. Compile and deploy the Oracle Tuxedo client/server program.
3. Run the Oracle Tuxedo application to invoke the external Web service applications.

Oracle SALT Supported Standards

Oracle SALT support the following standards:

- [SOAP Standards](#)
- [SCA and SDO Standards](#)

SOAP Standards

- Standards for transmitting data and Web service invocation calls between the Web service and the user of the Web service.
 - SOAP 1.1
For more information, see:
<http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>
 - SOAP 1.2
For more information, see:
<http://www.w3.org/TR/soap12-part0/>
 - SOAP with Attachment
For more information, see:
<http://www.w3.org/TR/SOAP-attachments>
 - MTOM
For more information, see:
<http://www.w3.org/TR/soap12-mtom/>

- A standard for client applications to find a registered Web service and to register a Web service.
 - UDDI 2.0
For more information, see:
<http://uddi.org/pubs/ProgrammersAPI-V2.04-Published-20020719.htm>
- Standards for describing the Web service to clients so they can invoke it.
 - WSDL 1.1
For more information, see:
<http://www.w3.org/TR/2001/NOTE-wsdl-20010315>
 - WS-Policy
For more information, see:
<http://specs.xmlsoap.org/ws/2004/09/policy/ws-policy.pdf>
<http://specs.xmlsoap.org/ws/2004/09/policy/ws-policyattachment.pdf>
- Standards for Web Service infrastructure.
 - WS-Addressing
For more information, see:
<http://www.w3.org/Submission/2004/SUBM-ws-addressing-20040810/>
 - WS-ReliableMessaging
For more information, see:
<http://specs.xmlsoap.org/ws/2005/02/rm/ws-reliablemessaging.pdf>
<http://specs.xmlsoap.org/ws/2005/02/rm/WS-RMPolicy.pdf>
- Standards for Web Service Security.
 - WS-Security 1.0
For more information, see:
<http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf>
<http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0.pdf>
<http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0.pdf>
 - WS-Security 1.1
For more information, see:

<http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf>
<http://www.oasis-open.org/committees/download.php/16782/wss-v1.1-spec-os-UsernameTokenProfile.pdf>
<http://www.oasis-open.org/committees/download.php/16785/wss-v1.1-spec-os-x509TokenProfile.pdf>

- Standards for Web Service Transactions.
 - WS-Coordination 1.0
For more information, see:
<http://schemas.xmlsoap.org/ws/2004/10/wscor/>
 - WS-AtomicTransaction 1.0
For more information, see:
<http://schemas.xmlsoap.org/ws/2004/10/wsat/>

SCA and SDO Standards

Oracle SALT supports the following SCA and SDO standards:

- SDO for C++ Specification V2.1 published December, 2006:
For more information, see:
<http://www.osoa.org/display/Main/Service+Data+Objects+Specifications>
- OSOA Specifications
 - SCA C++ Client and Implementation V0.95
For more information, see:
http://www.osoa.org/download/attachments/35/SCA_ClientAndImplementationModelforCpp_V0.95.pdf?version=1
 - SCA Assembly Model V0.96
For more information, see:
http://www.osoa.org/download/attachments/35/SCA_AssemblyModel_V096.pdf?version=1
- SCA Java Implementations
 - Tuscany - SCA Java Development Guide: for developers contributing code

For more information, see:

http://www.osoa.org/download/attachments/35/SCA_TransactionPolicy_V1.0.pdf?version=1

What's Next?

After becoming familiar with the Oracle SALT Product Overview, refer to the following topics for installing, configuring, and running Web services using the SALT product:

- Install the Oracle SALT product.

For an explanation of how to install the product, refer to the *Oracle SALT Installation Guide*

- Configure and administer the Oracle SALT product.

For an explanation of how to configure and administer the product, refer to the *Oracle SALT Administration Guide*

- Program Web Services with the Oracle SALT product and develop new applications using the Oracle SALT SCA Container.

For an explanation of how to program with SALT, refer to the *Oracle SALT Programming Guide*

- Oracle SALT Web service and SCA samples

For Oracle SALT Web service application samples, refer to the *Oracle SALT Sample Guide*.

Oracle SALT Overview