Oracle® Service Architecture Leveraging Tuxedo (SALT)

Product Overview

10*g* Release 3 (10.3)

January 2009

ORACLE

Service Architecture Leveraging Tuxedo Product Overview, 10*g* Release 3 (10.3)

Contents

Oracle SALT Overview

Oracle SALT Overview

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

Understanding Oracle SALT

Oracle Service Architecture Leveraging Tuxedo (SALT) is an add-on product option for Tuxedo, enabling 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 Tuxedo services as Web services, and 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 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. 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 Tuxedo Web service integration solution. It reduces Tuxedo/Web service integration costs and decreases conversion processes that may exist with other solutions for accessing Tuxedo services. It enables seamless connectivity between Tuxedo applications and external Web service applications.

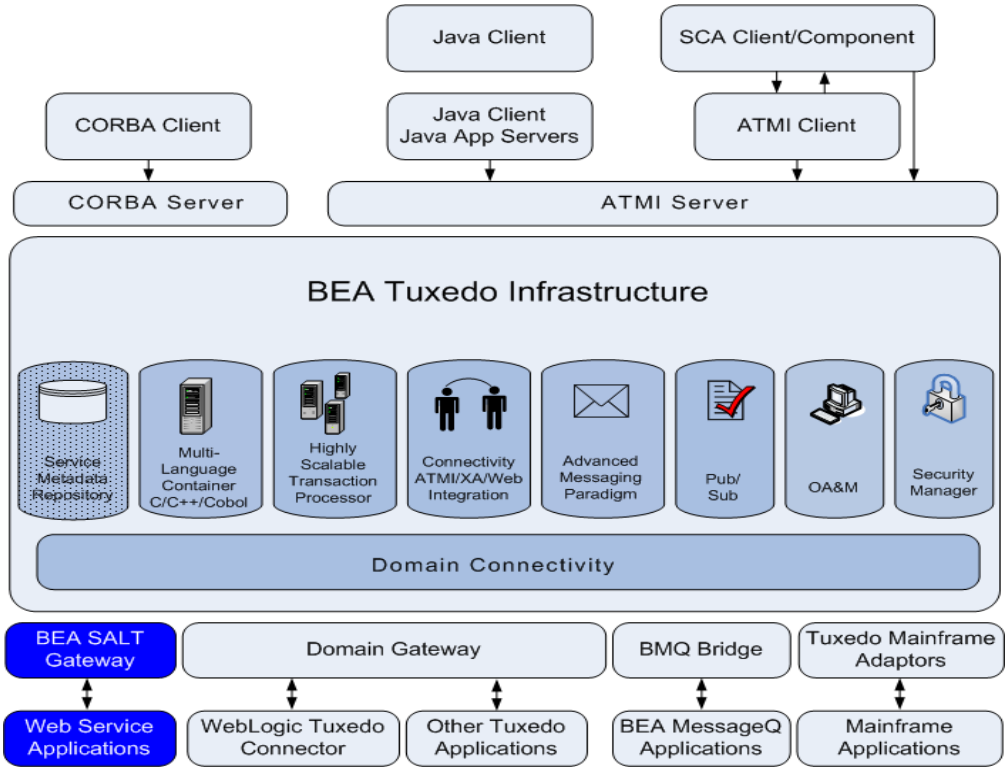
Oracle SALT allows existing Tuxedo services (inbound) to be easily exposed as Web services without additional programming tasks. It also allows you to create native Tuxedo applications that access external Web services (outbound) transparently.

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 Tuxedo framework.

Figure 1 Oracle SALT Gateway in Tuxedo Infrastructure



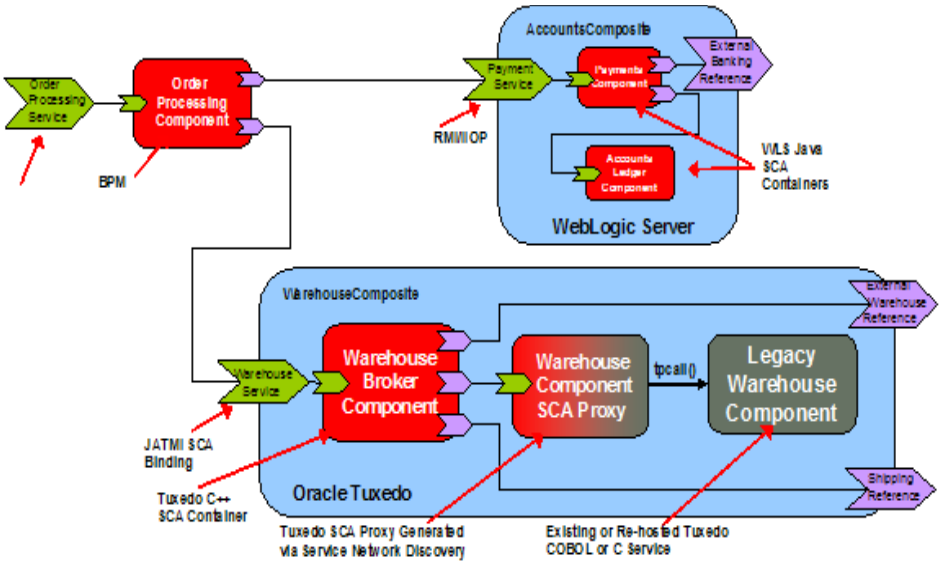
Understanding the Oracle SALT SCA Container

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 Tuxedo infrastructure, such as reliability, availability, scalability and performance.

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

Figure 2 shows 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



Oracle SALT Release History

Oracle SALT is the latest add-on to the Tuxedo product family. Developed in 2006, Oracle SALT is designed to provide a seamless Tuxedo solution of integrating Tuxedo applications and standard Web services application. With the addition of an SCA container, SALT is also designed

to better integrate existing 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 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)
- Tuxedo TPFAIL Support for Web Services
- Extensible Data Type Mapping and Message Conversion
- Multiple Encoding Support
- Configuration-Driven Deployment
- Leveraging the Tuxedo Service Metadata Repository
- Data Type Mapping and Message Conversion
- Asynchronous and Reliable Messaging
- Web Service Security Support

Release 10*g* Release 3 (10.3)

SALT 10*g* Release 3 (10.3) introduces 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 10*g* Release 3 (10.3) SCA container enables new programming model and leverages Tuxedo's most valued features, such as reliability, availability, scalability, and performance. SALT 10*g* 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 10*g* 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 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 Tuxedo client administrators to monitor application validity at runtime. You can record application high water client count, current client count, and named users.Automatic Service Definition Discovery

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 Gateway (GWWS)

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

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

You can have multiple GWWS instances in one 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 Tuxedo service contract information from the 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 Tuxedo applications and Web Service WSDL descriptions.

WSDL Generator from Tuxedo Definitions

When using Oracle SALT to publish 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 Convertor to 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 Tuxedo specific definition files (SALT Web Service Definition file, 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 Tuxedo Service Metadata Repository Definition file and FML32 Field Table Definition file provide service interface descriptions for Tuxedo client programming.

SCA Container APIs and Utilities

Table 1 describes the SALT 10*g* Release 3 (10.3) SCA Container APIs and utilities.

Table 1 SALT Container APIs and Utilities

Utility/API Name	Description
`tuxscagen utility`	Helps generate C++ interface header files and XML-based SCDL composite and component files from the Tuxedo Service Metadata Repository. It can also generate Java interface classes for JATMI client.
`buildscaclient`	Compiles and links the SCA client programs.

Table 1 SALT Container APIs and Utilities

Utility/API Name	Description
`buildscacomponent`	Used to build shared libraries containing the SCA components (as application code). It builds individual SCA components from source code. The command reads SCDL source, finds the component(s) in the composite(s) file(s) specified, parses the corresponding `.componentType` file(s) and produces corresponding executable libraries, in the same location as the `.componentType` files. **Note:** `buildscacomponent` is typically run before `buildscaserver`.
`buildscaserver`	Parses SCDL definitions to produce a Tuxedo-deployable server along with the following Tuxedo-deployable elements: • Shared libraries containing SCA components • SCDL configuration It also automates the configuration of SALT Web service artifacts when the SCDL code contains the `<binding.ws>` element.
`scaadmin`	Provides statistics and management of SCA components
`scapasswordstore`	Manages the `password.store` file used by SCA components which refer to Tuxedo-based services.
`setSCAPasswordCallback(3c)`	Sets the callback for retrieving a password associated with an identifier in a `<binding.atmi>` or `<binding.jatmi>` element.
`mkfldfromschema/mkfld32fromschema`	The `mkfldfromschema` and `mkfld32fromschema` commands take an XML schema as input and produce a field table suitable. This table is processed by the `mkfldhdr` or `mkfldhdr32` command or is loaded by other programs that need it. • `mkfldfromschema` is for use with 16-bit FML • `mlfld32fromschema` is for use with 32-bit FML
`mkviewfromschema/mkview32fromschema`	Takes an XML schema as input and produce a view file. This view file can be processed by the `viewc` or `viewc32` command. • `mkviewfromschema` is used with 16-bit views • `mkview32fromschema` is used with 32-bit views.

Oracle SALT Use Cases

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

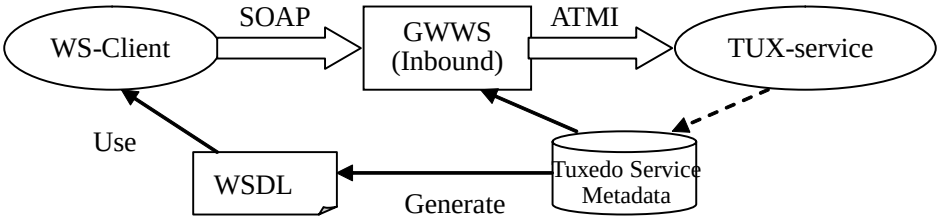
- Use Case 1: Exposing Native Tuxedo Services as Web Services
- Use Case 2: Invoking Web Services from Tuxedo Applications
- Use Case 3: Connecting Tuxedo Domains Using SOAP Protocol
- Use Case 4: SCA to SCA Communication
- Use Case 5: SCA Components Calling an Existing Tuxedo Service
- Use Case 6: Tuxedo ATMI Calling SCA Components

Use Case 1: Exposing Native Tuxedo Services as Web Services

Native 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 Tuxedo ATMI calls. SALT generates a WSDL document that describes the open standard Web service interfaces for Tuxedo services.The Tuxedo Service Metadata Repository is used to define Tuxedo service contract information. This is an “inbound” use case.

Figure 3 illustrates a generic inbound Web service call.

Figure 3 Exposing Tuxedo Services as Web Services (Inbound)



Use Case 2: Invoking Web Services from Tuxedo Applications

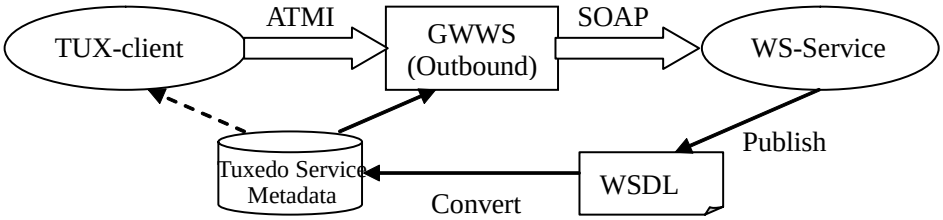
Web service applications can be imported into a Tuxedo domain, advertised as Tuxedo services through the GWWS server, and invoked from Tuxedo applications. SALT converts and maps

each `wsdl:operation` as a particular Tuxedo service. The GWWS server advertises the mapped services (called SALT proxy services), and accepts Tuxedo ATMI requests from Tuxedo applications.

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

Figure 4 illustrates a generic outbound Web service call.

Figure 4 Invoking Web Services from Tuxedo Applications (Outbound)



Use Case 3: Connecting Tuxedo Domains Using SOAP Protocol

Oracle SALT also allows you to connect two *different* 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.

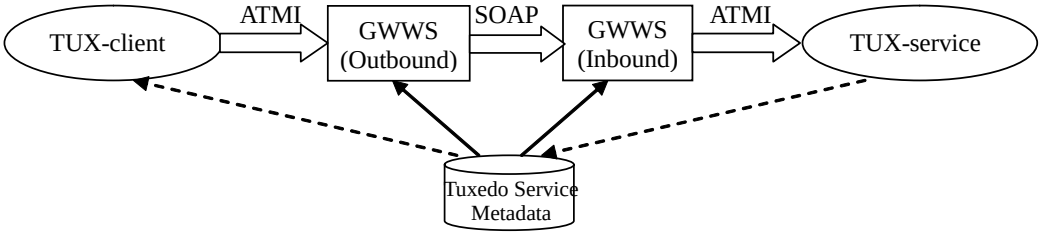
The receiving Tuxedo domain must propagate the Tuxedo service definition to the calling Tuxedo domain. This means that the calling domain Tuxedo Service Metadata Repository must contain the Tuxedo service definition file that runs in the receiving domain.

Note: This should be set up *manually*. The Tuxedo Service Metadata Repository infrastructure does not currently provide automatic propagation between 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 5 illustrates how to use Oracle SALT to connect two domains.

Figure 5 Connecting Two Tuxedo Domains with SOAP protocol



Two GWWS servers should not be used to create connections within the *same* Tuxedo domain, see Figure 6. Also, a single GWWS server cannot connect to itself, see Figure 7.

In either scenario, the GWWS server advertises the same 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 6 Two GWWS Servers Making a Connection Within the Same Tuxedo Domain

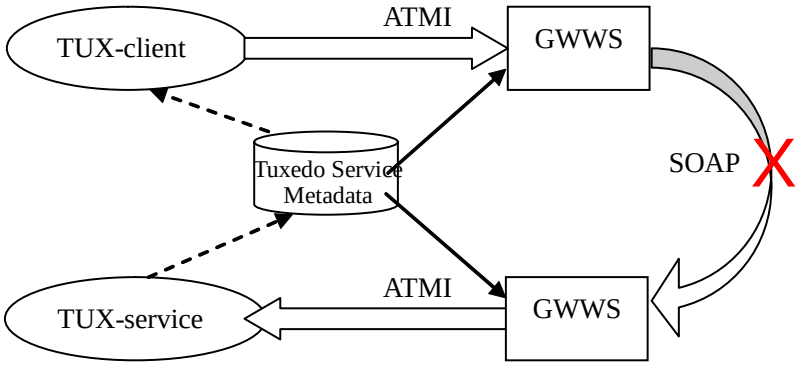
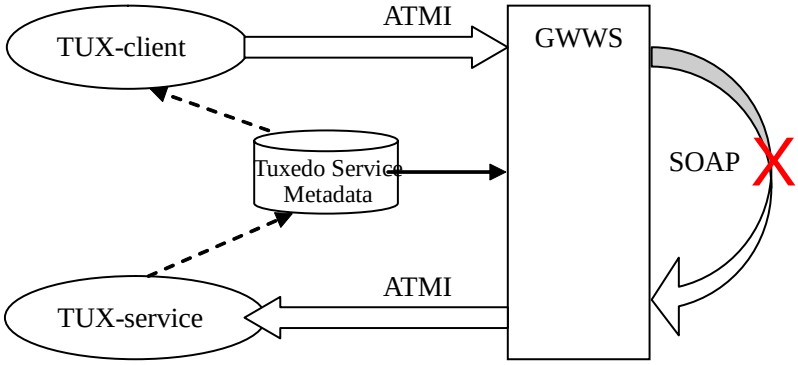


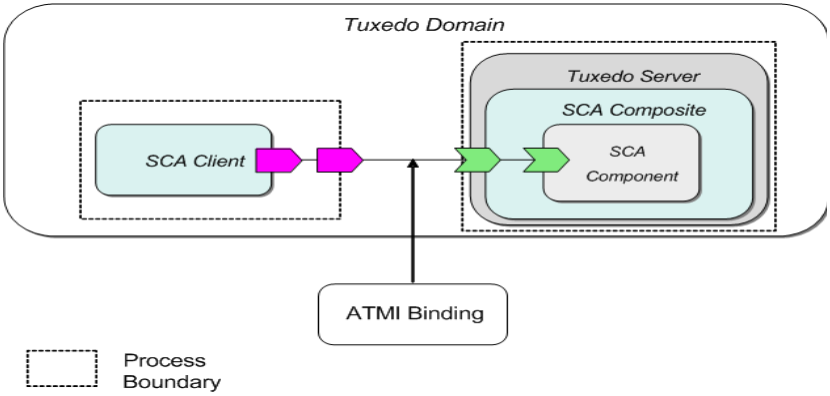
Figure 7 Single GWWS Server Making a Connection to Itself



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 Tuxedo framework, by communicating using a native ATMI binding, a WorkStation protocol based binding, or a Web-Services binding as shown in Figure 8.

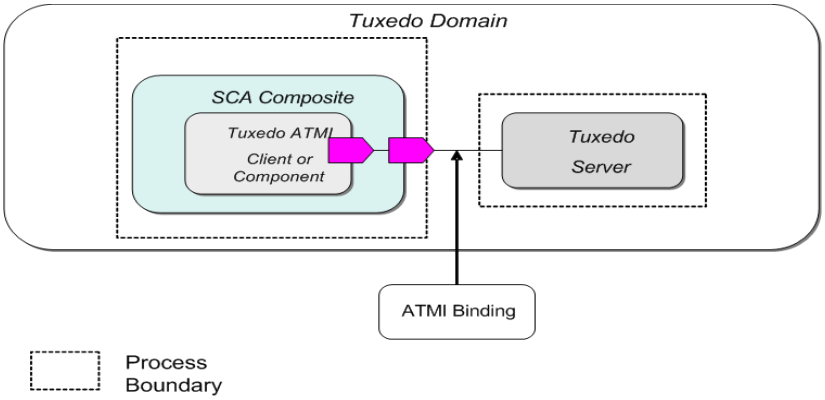
Figure 8 SCA to SCA Communication



Use Case 5: SCA Components Calling an Existing Tuxedo Service

Newly developed SCA components can interact with existing Tuxedo ATMI services by using the ATMI binding, as shown in Figure 9.

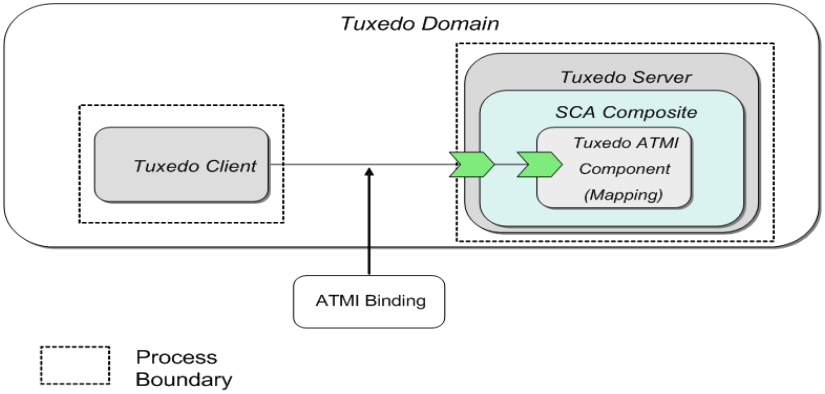
Figure 9 SCA Components Calling an Existing Tuxedo Service



Use Case 6: Tuxedo ATMI Calling SCA Components

Conversely, existing Tuxedo clients can interact with newly-developed SCA components by being exposed with the ATMI binding, as shown in Figure 10.

Figure 10 Tuxedo ATMI Calling SCA Components



Configuring Web Services with Oracle SALT

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

1. Configure Inbound Tuxedo Services
   a. Define Tuxedo application services using the 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 Tuxedo the following components: SALT Web Service Definition file, 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 Tuxedo Service Definitions into the 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 Tuxedo `UBBCONFIG` file.
6. Boot the Tuxedo application.

Invoking 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 Tuxedo Programming Interfaces

1. Create a Tuxedo client/server program according to the generated Tuxedo Service Metadata Definition file. The client program can be written in any Tuxedo supported client-side programming language (C/C++, Java, COBOL, .NET, and so on).
2. Compile and deploy the Tuxedo client/server program.

3. Run the 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

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 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.*