



An Oracle White Paper
November 2012

Cloud Integration – A Comprehensive Solution

Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Introduction	4
Putting Cloud Integration in Context	6
Introducing Oracle Fusion Middleware.....	7
A SOA Foundation	8
Types of Integration	8
Cloud Integration Examples	9
Oracle Fusion Applications Integration using Oracle SOA Suite	9
Salesforce Integration using Oracle SOA Suite	10
Workday Integration using Oracle SOA Suite	11
Service Aggregation.....	12
Service Virtualization.....	13
Security Considerations	15
Unified Monitoring and Management	15
Conclusion	16

Introduction

Cloud computing solutions are widely hailed as a way to reduce capital expenditures and move to an inexpensive, subscription-based IT model. However, many organizations don't stop to consider all of the nuances of integrating cloud applications with their existing information systems. While deploying a cloud app or subscribing to a cloud-based service may be relatively straightforward, how will this new IT asset fit in with the rest of the enterprise, including on-premise systems and other cloud applications? What is the system of record from which data will be derived? Which business processes are involved? Do you need an enterprise data model that is independent of the cloud data model?

These questions are relevant for any type of integration project, with or without a cloud deployment. However, they are particularly relevant in today's business world, where individual departments and lines of business sometimes subscribe to cloud and Software as a Service (SaaS) applications without the oversight of the IT department, and without always adhering to best practices. This departmental autonomy introduces new challenges in the areas of security, reliability, quality and the potential for a sharp rise in the number of disparate toolsets.

SaaS-only vendors (not providing an on-premise option) typically provide packaged integration tools with basic capabilities to manage the exchange of data for limited scenarios. When combining these integration tools from multiple vendors however, you may find yourself managing disparate user interfaces and writing a lot of custom code using a variety of different programming languages. While many cloud models are masqueraded under the guise of simplicity, expanding an on-premise application infrastructure to include SaaS applications in an ad hoc fashion can ultimately increase the complexity of your enterprise. In order to achieve the promised benefits of cloud computing, such as greater flexibility and lower costs, a cohesive vision for unifying SaaS applications with on-premise information systems is highly recommended.

Thus instead of focusing only on quick connectivity, CIOs must ensure that all aspects of cloud integration align with their strategic vision for IT, with attention to audits and compliance, security standards, and governance. Many cloud vendors showcase simplistic integrations that demonstrate basic connectivity while ignoring the real integration challenges that the IT team will face, including system configuration issues, application customizations, non-functional requirements, reliability, scalability, and security.

Application programming interfaces (APIs) differ greatly from one cloud app to another. Each vendor enforces its own mechanisms for security, message delivery, metadata definitions, query criteria, object semantics, and object schema. It is helpful to have a universal way to mask these technical details and enforce consistency at a logical level. Ideally, developers should be able to mediate among multiple applications from one centralized integration platform that leverages the various cloud vendor interfaces to automate both inbound and outbound connectivity. Having a robust, centralized integration platform minimizes the complexity associated with managing the APIs from various cloud vendors.

This white paper describes how to integrate on-premise and cloud applications, including Oracle Cloud applications, with this type of pragmatic, universal perspective. While Oracle, Salesforce, and Workday are used as examples, the principles apply to many vendors. The hub for these integrations are the Oracle SOA Suite and Oracle data integration products, part of Oracle Fusion Middleware, a unified platform that accommodates all types of information systems, deployment models, SaaS vendors, and Platform as a Service (PaaS) infrastructure, anchored by a cohesive set of tools for development, management, security, and governance.

Putting Cloud Integration in Context

Integration entails exchanging information among systems to achieve a specific business objective such as synchronization of customer account data for fulfilling orders. The location of an application and its data—whether in your data center or under the auspices of a cloud provider—does not change this basic business need. While many enterprises have committed some level of investment to the cloud, most of these organizations have to deal with on-premise systems in tandem—or fuse data from another cloud app.

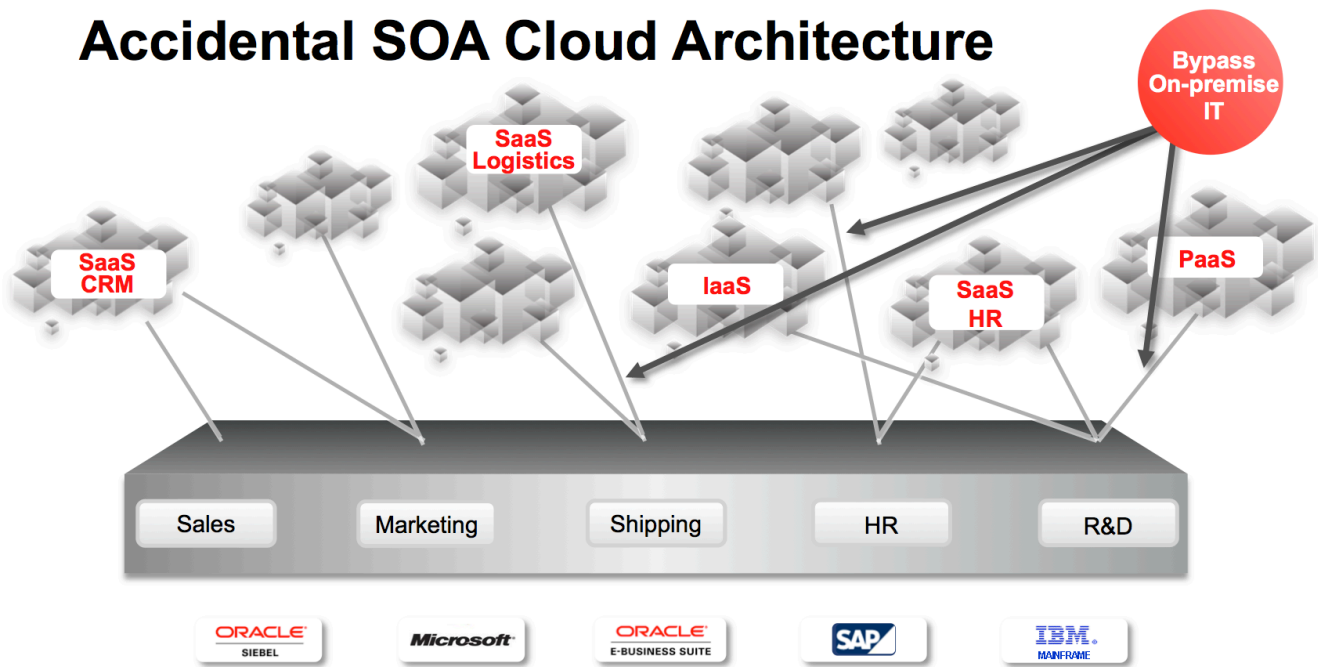
For example, an enterprise that relies on an on-premise CRM application might acquire a company that uses a SaaS-based CRM application. Account managers need to be able to access data from both systems in a cohesive way, share data between these systems, and ultimately establish an authoritative system of record that encompasses all customers. Diversity also creeps into the enterprise when an individual department subscribes to its own SaaS based CRM application. Departmental users may enjoy the ease and convenience of a rapid cloud deployment, yet they probably still need to access data from the corporate CRM system—and possibly merge the two.

Similarly, HR systems, payroll systems, and incentive compensation systems typically need to exchange information so that employees are properly compensated and paid. These are classic integration scenarios that IT pros have been dealing with for decades. As companies move one or more of these business functions to the cloud, creating the required connectivity takes on a new dimension. For example, your main employee data might reside in an on-premise HR system. If you utilize SaaS-based services for payroll, talent management, incentive compensation, or other employee-related functions, you will need to move data to and from the cloud.

Most enterprises have spent years avoiding the data “silos” that inhibit productivity. IT has had its fill of new integration paradigms, from CORBA to Client/Server to Web services, EAI and SOA. After decades of locking down critical issues such as interface definitions, governance, reliability, transaction management, exception handling, and transaction monitoring, is it time to reinvent the wheel yet again for the cloud era? Do cloud applications represent a new silo?

Without proper planning, it’s easy to end up with an “Accidental SOA Cloud Architecture” characterized by point-to-point connections from individual departments to the cloud, bypassing any well-established integration architecture. These custom interfaces are fraught with problems, often tactical rather than strategic, leading to brittle connections that are difficult to upgrade when endpoints change.

Accidental SOA Cloud Architecture



Business units bypassing IT best practices (security, SLAs, monitoring, etc.) resulting in an "Accidental SOA Cloud Architecture"

Once these applications are deployed, lack of centralized monitoring and management causes extra work for system administrators. And with no clear ownership or centralized accountability for particular integrations, the enterprise ends up hiring and maintaining a diverse set of skilled resources to create and maintain the interfaces.

Cloud data and functions often must be shared with on-premise information systems. Similarly, private clouds that depend on local data often need to feed that data to cloud-based applications, or trigger events through the firewall as part of a multi-step workflow. Balancing these initiatives requires careful planning if you don't want to end up with a new set of "spaghetti integrations" that are inflexible and difficult to manage.

Analysts and other industry experts often cite integration as one of the barriers to adoption of cloud services, especially for apps that need to exchange logic and information in this way. It's no wonder that SaaS integration has become one of the most sought after skills among IT professionals, including technical architects who understand SaaS connectivity and functional architects who understand the intricacies of the data being exchanged. SaaS integration challenges may in fact be bigger than the old integration challenges you faced with on-premise CRM and ERP systems. While the integration issues haven't changed, the cloud introduces additional complexities that need to be specifically accounted for.

Introducing Oracle Fusion Middleware

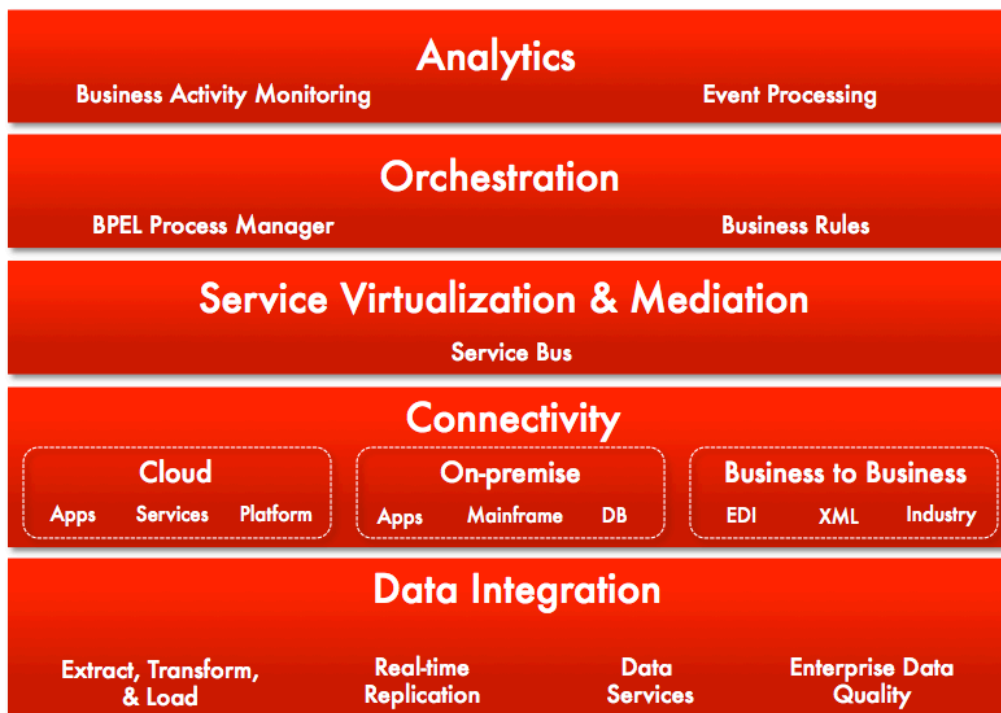
Fortunately, with Oracle Fusion Middleware, one set of integration tools can handle all of these integration scenarios, with direct and interchangeable connections to cloud, on-premise, and legacy systems. Oracle SOA Suite, Oracle Data Integration and other components of the Oracle Fusion Middleware family provide a cohesive set of integration capabilities to simplify diverse IT environments.

Oracle's support for Web services enables you to easily connect to on-premise and cloud-based services through one cohesive middleware platform. Developers can initiate Web services calls to and from cloud applications and connect to on-premise information systems via adapters that mask the technical nuances between applications and data models such as connectivity options, session management, authentication, and authorization. Whether the systems at hand include

PeopleSoft, Oracle E-Business Suite, JD Edwards, SAP, Salesforce, or other common applications, Oracle exposes simplified web services to invoke the on-premise applications. Modern applications built using standards-based web services technologies such as Oracle Fusion Applications don't need such adapters. Oracle also offers native connectors to legacy databases and file systems.

A SOA Foundation

Since most cloud applications support Web Services, a Service Oriented Architecture (SOA) is an ideal platform to address cloud integration. Many established SOA concepts, considerations, and design patterns are even more relevant in the world of cloud integration than in an “on-premise-only” scenario. Oracle SOA Suite simplifies connectivity by enabling developers to set up and manage universal services and orchestrate them into composite applications and business processes. It utilizes an enterprise service bus (ESB) as the foundation for shared services, process orchestration, event processing, and business activity monitoring so that organizations can have visibility into their entire application infrastructure as they adopt additional cloud services in the future.



A unified approach to integration with Oracle SOA Suite and Data Integration components

Types of Integration

Cloud integration scenarios fall into a couple of familiar models. *Application integration* refers to interfaces that are event-driven, near real-time, and impact business operations. For example, if you are an online retailer selling products to customers, orders accepted through a customer-facing website must be routed to a back-end fulfillment system, which initiates processes related to order confirmation, inventory management, and shipping. The processes are relatively similar whether the front-end system is on premise or in the cloud: the CRM application sends information to the destination application, such as routing an order to a back end fulfillment system.

Data Integration scenarios come into play when you are moving large volumes of data in batch mode, such as when your sales database is migrated to the cloud to populate a cloud-based contact management system. In these instances you must consider the volume and frequency of data integration – daily, nightly, weekly, etc. You might begin with a batch upload of

historical data followed by periodic data updates each day, week, or month. If these integrations are done on-premise, security is less of an issue. If you are interfacing with a cloud app, the security requirements will undoubtedly be stricter.

For both real time and batch-level interfaces, developers can always set up point-to-point interfaces. However, the protocols, file formats and metadata will vary from one cloud vendor to another. Rather than manually coding for these distinctions, they will save time by coding once to a data integration platform that can mask these distinctions, especially as the number of cloud providers in the IT ecosystem expands.

Oracle provides a simple and consistent method to create and maintain these interfaces. It doesn't matter if the interfaces are from SaaS to SaaS, SaaS to on-premise, on-premise to SaaS, or on-premise to on-premise, Oracle can connect to any kind of SOAP or REST Web service using Oracle Fusion Middleware.

Cloud Integration Examples

Most integration tools let you make a Web service call from your on-premise system to the cloud. However, new deployment options (on-premise, hosted, cloud), more client programming languages (RoR, Python), an increase in the use of mobile clients (requiring light weight web services/REST), and intensified security considerations require the mediation of a robust middleware environment. Most integration vendors support popular Web services standards, with WSDLs (Web Services Description Language) that you can download and use immediately to interface their applications with your surrounding information systems. However, each Web service has different metrics, object structures, and security requirements.

It's easy to utilize these Web service frameworks with Oracle Fusion Middleware. Three examples are given below.

Oracle Fusion Applications Integration using Oracle SOA Suite

To invoke Web services from Oracle Fusion Applications, you would follow these steps:

1. Obtain the WSDL from the Oracle Enterprise Repository for Fusion Applications cloud hosted instance (<http://fusionappsoer.oracle.com>) - This requires a free Oracle login. Search for Asset Type = "ADF Service" and enter the name of the service, such as Account. From the results list, pick the desired service, such as Trading Community Account. Clicking on the Details tab will provide a link to the Abstract WSDL as well as the URL pattern for finding the Concrete WSDL in your Fusion Applications environment. Simply replace the hostname and port and download the WSDL. Note that not all the services are available in SaaS mode since some services are "internal" and not exposed on the external virtual host; that is, they are inside the "firewall". To ensure that the service you have selected can be invoked from external clients, click on the Taxonomy tab of the service definition and ensure that Keyword=External is present.
2. In Oracle JDeveloper, create a SOA project with a BPEL process that needs to integrate with Oracle Fusion Applications. Create a partner link based on the WSDL obtained in step 1.
3. Apply a compatible security policy to this partner link. If your environment has an SSL enabled environment, you can use the out-of-the-box oracle/wss_username_token_over_ssl_client_policy. You can also choose the equivalent SAML policies (for identity propagation) or WS-Security message protection enabled policies if these have been enabled in your environment.

To get invoked via outbound services from Oracle Fusion Applications, you would follow these steps:

4. Set up a Receive activity in a BPEL process with a SOAP endpoint and implement the process based on the WSDL and XSD documented by Oracle Fusion Applications in the Application Composer Object Workflow Outbound Message page. This can be considered as an intermediary web service that understands the Fusion objects and mediates the requests to other on-premise or cloud applications as needed—for example, to

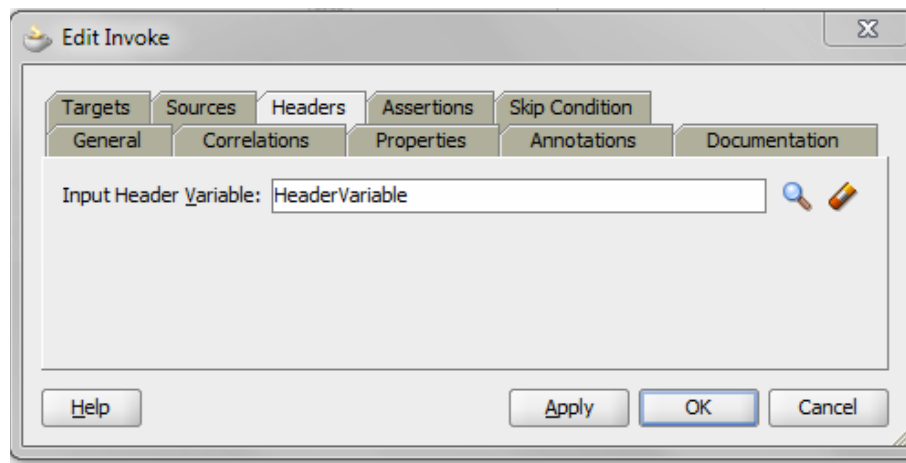
synchronize new customers created with the on-premise Master Data Management system or to create them as Contacts in the Oracle RightNow CX cloud service.

5. Deploy the composite and obtain the endpoint URL.
6. In Oracle Fusion Applications, register the endpoint URL in the Application Composer Object Workflow page in the Outbound Message page activity for the object. You can now receive this outbound message in your SOA composite and process it as per your required business process.

Salesforce Integration using Oracle SOA Suite

To use the Salesforce security model you can follow these steps:

1. Log into your Salesforce instance as an administrator and obtain the Enterprise WSDL. This WSDL has the definitions of the various web service operations as well as the standard and custom objects. The Enterprise WSDL is available at [setup | Develop | API | Enterprise WSDL](#)
2. There are two specific considerations when integrating with Salesforce.
 - a. The service end point on Salesforce is dynamic and hence cannot be hardcoded at the service definition time.
 - b. After calling the login web service a dynamic session id is created and that session id needs to be passed to Salesforce on every web service invoked thereafter. That is the security mechanism used by Salesforce for validating the web service call.
3. In Oracle JDeveloper, create a project with a BPEL process that needs to integrate with Salesforce. Create a partner link based on the WSDL obtained in step 1.
4. Create a variable for handling the dynamic endpoint based on message type *EndpointReference*, from the schema imported from <http://schemas.xmlsoap.org/ws/2003/03/addressing>.
5. Create another variable, say *HeaderVariable*, to handle the session id, based on the message type *Header*, from the Enterprise WSDL.
6. Create an Invoke activity for calling the *login* operation and create new variables for the request and response.
7. Create an Assign activity before the Invoke operation to update the user and password in *login request* variable.
8. From the *login response*
 - a. update the endpoint URL variable
 - b. copy the session id into the *HeaderVariable*
9. Every Invoke activity in the BPEL that calls a web service operation to Salesforce subsequently has to pass the session id. This is done in the Invoke activity Header tab as shown below:

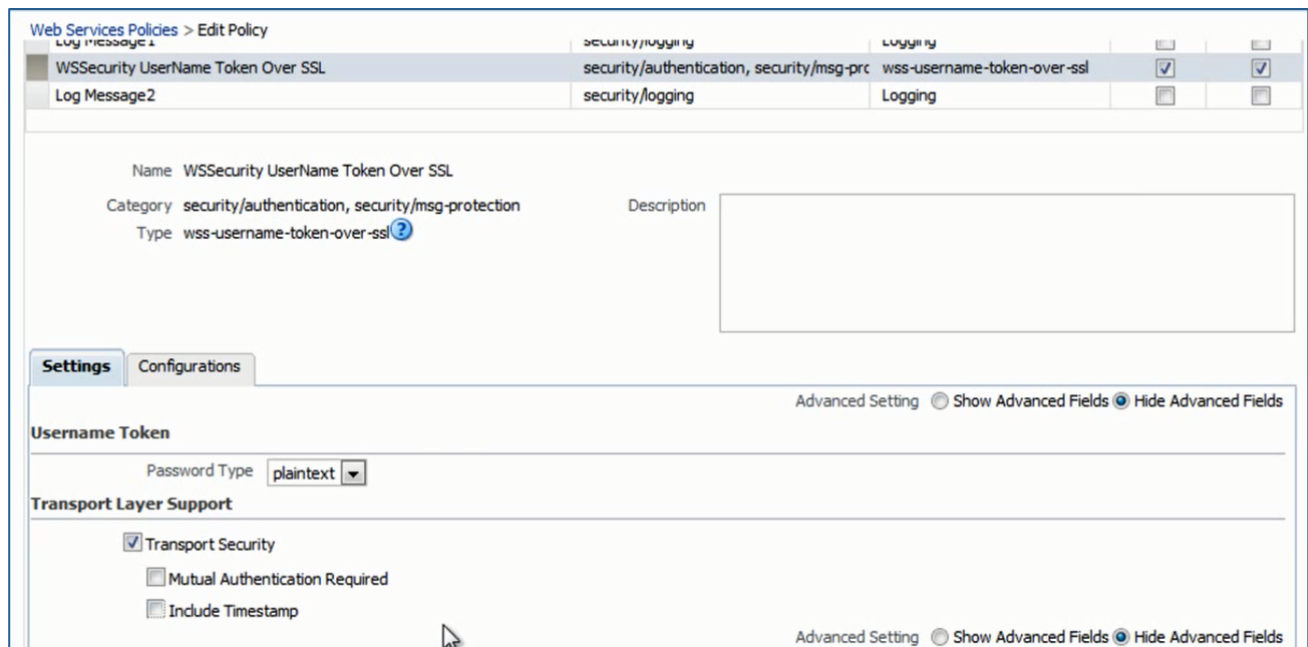


10. Salesforce can call a web service (a Receive activity in BPEL) by creating an outbound message on Salesforce. To consume this outbound message the steps are as follows:
 - a. Create the outbound message in SF and use a dummy end point.
 - b. Obtain the outbound message WSDL.
 - c. Create a BPEL that exposes a SOAP endpoint or a mediator based on the WSDL obtained in step b.
 - d. Update the endpoint in SF based on the real endpoint created during step c.
 - e. Activate the outbound message in SF.
 - f. You can now receive this outbound message in BPEL and process it as per your requirements.

Workday Integration using Oracle SOA Suite

To integrate with the Workday cloud, follow these steps:

1. Obtain the WSDL that contains the web service operations that you would like to integrate into your BPEL process. Workday has categorized the web service operations based on functionality. For example, if you want to integrate with the applicant-related web service operations you will need to obtain the Staffing WSDL.
2. Create a partner link based on the WSDL obtained in step 1.
3. Apply a custom security policy to this partner link. This security policy is based on the out-of-the-box oracle/wss_username_token_over_ssl_client_policy with the timestamp option unchecked.
 - a. Use Oracle Enterprise Manager to navigate to the domain.
 - b. Right-click on the domain and click on Web Service > Policies.
 - c. Make a copy of the oracle/wss_username_token_over_ssl_client_policy and uncheck the timestamp flag as shown in the image below.



Web Services Policies > Edit Policy

Name	Category	Type	WSS Username Token Over SSL
WSSecurity UserName Token Over SSL	security/authentication, security/msg-protection	wss-username-token-over-ssl	<input checked="" type="checkbox"/>
Log Message2	security/logging	Logging	<input type="checkbox"/>

Name: WSSecurity UserName Token Over SSL
 Category: security/authentication, security/msg-protection
 Type: wss-username-token-over-ssl

Description:

Settings | Configurations

Advanced Setting ☐ Show Advanced Fields ☒ Hide Advanced Fields

Username Token

Password Type: plaintext

Transport Layer Support

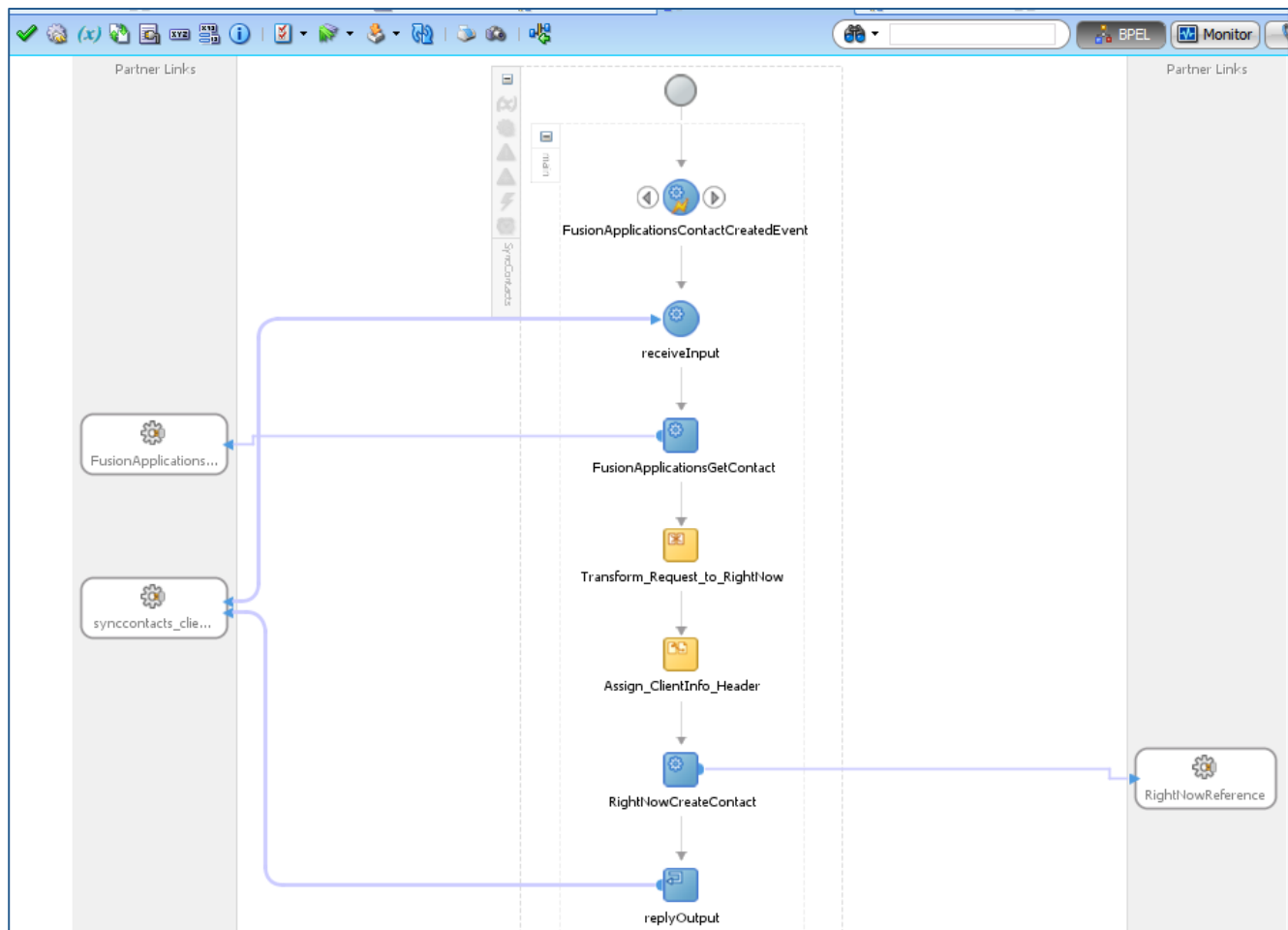
☒ Transport Security
☐ Mutual Authentication Required
☐ Include Timestamp

Advanced Setting ☐ Show Advanced Fields ☒ Hide Advanced Fields

- You will need to apply a security policy (user and password) for the Web services that will be called from Workday. That will have to be set up using the web service management functionality within Oracle SOA Suite. You will need these credentials when you set up the outbound EIB in workday.

Service Aggregation

Most APIs provided by cloud vendors are at a low level of granularity. Customers must tie these discrete APIs together through a design pattern known as service aggregation to create the high-level data-exchange functions that constitute an inbound service. Oracle Fusion Middleware supports this process through a graphical user interface using the familiar Business Process Execution Language (BPEL). Oracle BPEL Process Manager allows users to orchestrate service invocations to various heterogeneous systems and then expose these aggregations via a single service call. The Oracle BPEL engine can orchestrate service invocations across multiple service providers, coordinating the interaction among and between business processes.



Oracle BPEL process manager shown with service level orchestration across multiple on-premise and cloud applications

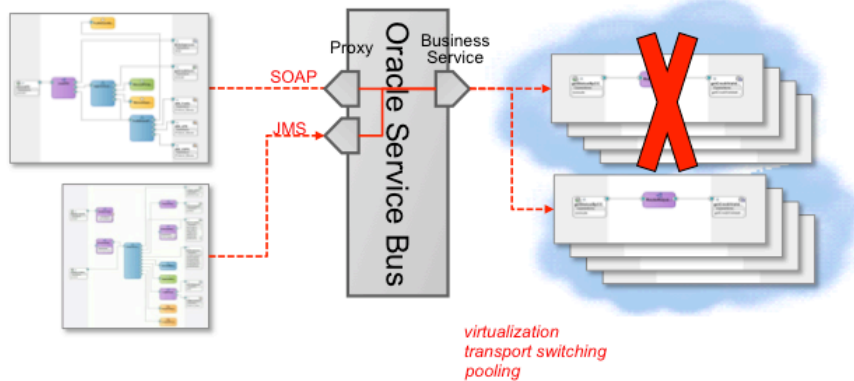
Service aggregation lets application designers operate at a much higher level, ignoring discrete granular details. For example, a developer can simply drag and drop multiple web service invocations to handle chatty API conversations with a single application (e.g. Taleo service) and expose a web service to clients. Developers can also perform data enrichment, for example to add data to sparse records and convert them to complete records, as is expected by an ERP system. In addition, the developer can also expose a service that interacts with multiple applications (shown in the screenshot above), creating new value-added business functionality that can be easily reused across the enterprise.

Service Virtualization

Service Virtualization provides a similar level of abstraction for groups of service calls. In these situations, Web services calls don't go directly to the applications, but rather to an enterprise service bus that sends them on to the correct applications. This type of virtual architecture allows a client to be loosely coupled with a service provider, permitting an enterprise to easily add and remove cloud vendors. The integration platform interfaces with the service bus, and the service bus interfaces with the providers. If a provider changes its API definition, that change is handled by the service bus, which has the flexibility to transpose dozens of different data types and network protocols into a common format, with no changes to the on-premise applications.

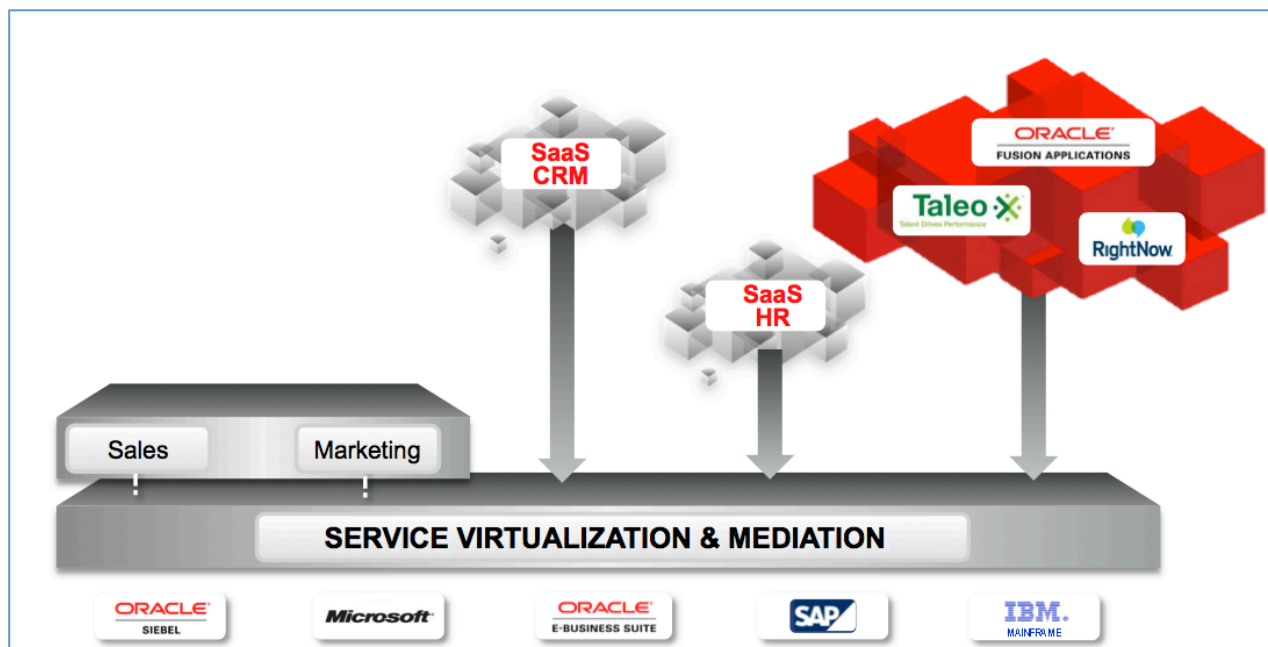
This is especially crucial in today's SaaS market, where Tier 1 cloud vendors tend to release two to four versions of their products each year. Interfacing with these new versions without have service virtualization in place put a tremendous burden on the IT personnel who are overloaded with keeping their systems in sync with these frequent updates.

Decouple Clients from Providers with Oracle Service Bus



For example, a company that deploys a Payroll system in a private cloud might later decide to utilize the SaaS version of the same application. Simple modifications to the service virtualization layer will shield the enterprise from having to make major adjustments. The middleware platform is comprehensive and flexible enough to automatically account for differences between the two deployments. Similarly, if your cloud vendor is acquired or goes out of business, you have lowered your risk by creating this layer of separation between your infrastructure and the cloud. Service Virtualization sets you up to adopt the best cloud vendors for your needs with minimal changes.

Service Virtualization is especially useful if you customize a SaaS instance. Once you make changes, the cloud vendor gives you a WSDL containing these customizations. You must update every client interface that utilizes that particular API. However, if all of your clients interface to the shared services layer, the middleware platform will handle the transformation to the API call, including any new customizations that have to be updated.

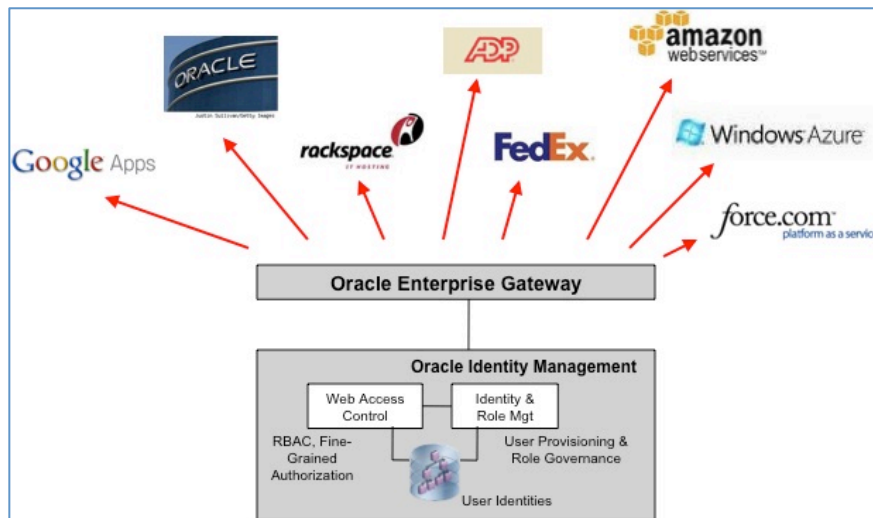


Sales and Marketing departments interfacing with Oracle Cloud, 3rd party cloud vendors, and on-premise applications through service virtualization thereby eliminating point-to-point integration challenges

Security Considerations

When on-premise enterprises expand to include cloud applications, security concerns draw increased attention. Security policies that protect information at the transport layer, message layer and authorization layer have the typical on-premise security topics as well as some new ones. IT pros must continually address threats like denial of service attacks and intrusions from rogue sites. SOX, HIPAA, PCI-Data Security, and other regulations require companies to establish privacy and data integrity policies when communicating through cloud services.

Oracle Enterprise Gateway manages all of these connections, including the API keys that connect Cloud services such as Amazon Web Services (AWS). It applies critical governance controls for service access, usage, and availability and aggregates multi-domain services such as Force.com and Google Apps. This approach also helps mitigate delays and outages that are outside of your control by caching frequently accessed data.



Oracle Enterprise Gateway working in conjunction with Oracle Identity Management

Oracle Enterprise Gateway is tightly integrated with Oracle Access Manager, Oracle Entitlements Server, Oracle Web Services Manager, and Oracle SOA Suite to provide transport- and application-level security for web services requests.

In conjunction with the web services management features within Oracle SOA Suite, IT administrators can set policies to authorize and authenticate Web service invocations such as requiring a username/password, a SAML token, or encryption for sensitive data. These security definitions are loosely coupled with the implementation of the associated services, which means administrators can easily change or update the policies without writing a single line of code. The policies can be applied interchangeably among on-premise and cloud applications. Finally, because Web services security is closely related to identity management, Oracle Enterprise Gateway enables you to apply unified security policies for the Oracle Identity Management offering.

Unified Monitoring and Management

Whether you are supporting enterprise cloud apps or traditional IT applications, you need to be able to proactively monitor these business applications and their underlying IT infrastructure. The long-term goal is to manage cloud resources as business services rather than just a collection of technical components. End-to-end oversight is especially important when business processes are distributed among on-premise and cloud-based systems. For example, an order entry transaction

might be initiated by an on-premise system, trigger updates to a cloud-based CRM system, and then log an inventory check in a legacy warehouse management system. If something goes wrong during any part of this workflow, it may be difficult to determine what caused the problem. The Oracle SOA Management Pack, available within Oracle Enterprise Manager, allows you to see the entire flow of the message through one management console.

This page shows the flow trace of the instance across composites and SOA Infras monitored by Cloud Control. To view the audit trail of an individual component instance, click on the component instance.

ECID 42a82e661fff3ff4:-74749514:1365aef9e18:-8000-0000000000004d11

> SOA Infras (1)

> Faults (0)

Trace

Click a component instance to view its detailed audit trail.

Component Instance	Type	State	Time	Composite	Instance Id	SOA Infra
orderprocessor_client_ep	Service	Complete	Mar 28, 2012 3:35:11 PM (UTC-)	soaFusionOrderDemo/C410587	/Farm08_FOD_domain/FC	
OrderProcessor	BPEL Component	Complete	Mar 28, 2012 3:35:12 PM (UTC-)	soaFusionOrderDemo/C410587	/Farm08_FOD_domain/FC	
StoreFrontService	Reference	Complete	Mar 28, 2012 3:35:11 PM (UTC-)	soaFusionOrderDemo/C410587	/Farm08_FOD_domain/FC	
StoreFrontService	Service	Complete	Mar 28, 2012 3:35:11 PM (UTC-)	soaFusionOrderDemo/C410588	/Farm08_FOD_domain/FC	
CustomerAndOrderService	BPEL Component	Complete	Mar 28, 2012 3:35:11 PM (UTC-)	soaFusionOrderDemo/C410588	/Farm08_FOD_domain/FC	
StoreFrontService	Reference	Complete	Mar 28, 2012 3:35:11 PM (UTC-)	soaFusionOrderDemo/C410587	/Farm08_FOD_domain/FC	
StoreFrontService	Service	Complete	Mar 28, 2012 3:35:11 PM (UTC-)	soaFusionOrderDemo/C410589	/Farm08_FOD_domain/FC	
CustomerAndOrderService	BPEL Component	Complete	Mar 28, 2012 3:35:12 PM (UTC-)	soaFusionOrderDemo/C410589	/Farm08_FOD_domain/FC	
CreditCardAuthorizationService	Reference	Complete	Mar 28, 2012 3:35:12 PM (UTC-)	soaFusionOrderDemo/C410587	/Farm08_FOD_domain/FC	
RequiresApprovalRule	Decision Service Component	Complete	Mar 28, 2012 3:35:12 PM (UTC-)	soaFusionOrderDemo/C410587	/Farm08_FOD_domain/FC	
PartnerSupplierMediator	Mediator Component	Complete	Mar 28, 2012 3:35:12 PM (UTC-)	soaFusionOrderDemo/C410587	/Farm08_FOD_domain/FC	

Unified end-to-end flow trace in a single interface within Oracle Enterprise Manager Cloud Control 12c

This complete cloud lifecycle management solution allows you to quickly set up, manage and support enterprise clouds and traditional Oracle IT environments, with consistent monitoring from application to disk. It includes discovery capabilities to identify all the elements of an IT environment, as well as capacity-planning tools to advise IT professionals about how to view the environment as a shared infrastructure. A wizard-driven, role-based management console helps them manage everything from capacity planning to chargeback.

What happens if you lack a universal repository for managing SOA activities? SaaS-only vendors typically point you to detailed API documentation or a very simplified page listing all WSDLs. However without clarity about service expected usage patterns, SLAs, role-based access control visibility, and dependency analysis of the underlying integration metadata for change management, the integration effort multiplies quickly as your cloud initiatives expand.

Oracle Enterprise Repository acts as the single source of truth for information surrounding SOA assets and their dependencies and provides a common communication channel for the automated exchange of metadata and service information between service consumers, providers, policy decision points, and other governance tools. Rather than dealing with these issues on a case-by-case basis, this repository provides visibility into the entire SOA ecosystem and its dependencies, including assets in planning and development.

Conclusion

Despite the temptation to simplify cloud integration down to a two-page white paper or a five-minute demo, a successful cloud integration strategy goes well beyond simple connectivity. In this white paper, we have described a comprehensive cloud integration solution, which starts with connectivity but includes the equally important topics of service virtualization, security, security gateways, service aggregation, and unified monitoring and management.

One lesson that has proven itself to be true over the last 20 years of software integration is to avoid rigid, unique, point-to-point connections between application providers and clients. Instead create logical connections to a service bus or virtualization later, so that you don't have to continually re-code each interface as your information systems change and evolve. This is especially important today, given the rapid churn in the SaaS market. Having a flexible integration platform helps enterprises to maximize their options and lower risk.

When moving from the on-premise world to a hybrid world that includes cloud vendors, external companies become an integral part of your infrastructure. Not all of these vendors uphold the same policies, procedures, and principles that you have established in your own business. One of the biggest issues in using public clouds is integrating them with on-premise applications. To solve this issue, you need to enable universal connectivity to a common platform that can seamlessly “bridge” dissimilar technologies in a consistent way.

Oracle Fusion Middleware components, including Oracle SOA Suite, Oracle Data Integrator, Oracle Enterprise Repository, and Oracle Enterprise Manager, deliver a cohesive and flexible integration platform bolstered by connectors and adapters to hundreds of popular databases and applications. This unified approach is superior to the patchwork strategy of managing an increasing number of disparate integration toolsets provided by each SaaS-only vendor. No matter how your information systems evolve, Oracle can mitigate the complexity with a cohesive integration platform that accommodates all types of cloud services, applications, and infrastructure.



Cloud Integration – A Comprehensive Solution
November 14 2012

Author: David Baum, Rajesh Raheja, Bruce Tierney, Vijay Pawar

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



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

Copyright © 2012, 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 and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 1112