

# Considering Salesforce.com? What You Need to Know as an Oracle e-Business Suite Customer

By Ravi Balakrishnan

*Prisio Technologies*

## Introduction

Oracle e-Business Suite customers are either increasingly considering using salesforce.com or are currently using this application for sales force automation. The decision to use Salesforce.com has its own merits and demerits, which will not be addressed in this paper. The opportunities and challenges when using two disparate applications that do not integrate out-of-the-box are discussed.

This paper is structured by business object that is a candidate for integration with the corresponding implementation considerations on both Salesforce.com and Oracle e-Business Suite. A brief description of the features available and recommendations for configuration of both applications is presented. In this paper, salesforce.com denotes Salesforce.com sales cloud.

The candidate business objects discussed are:

- Customers
  - Parties
  - Accounts
  - Addresses
  - Contact
  - Contact Points (Email, Phone, Fax etc.)
- Quotes
- Territories
- Orders
- Invoices
- Price list
- Items (or Products in salesforce.com)
- Contracts and Install Base

Security and authentication of transactions is a frequently asked question when dealing with such integration. A brief discussion on security considerations is provided at the end of this paper with a brief discussion on use of middleware.

## Customer Integration

Customer integration is the most challenging of integrations which requires a detailed business process analysis.

## Directionality

The first step is to determine if the integration has to be unidirectional or bi-directional. Unidirectional interface mandates data in one system as the master and the other as slave. In a unidirectional interface, customer data is created in the master application and the updates in the master cascade down to the slave. In a bi-directional interface data can be updated in either system and will cascade to the other.

Most customers use unidirectional customer interface as bi-directional interfaces are rather difficult to develop and maintain and also causes conflicts when various users update the same data. The decision is based on several factors:

- Is there a centralized customer maintenance team that can create customers when requested by a salesperson?
- How is the customer data created? Does it come from external sources (conferences, events) as leads?
- How large is the data and how many customers are being created?
- How fluid is the customer data? Are changes made frequently? To what entities?
- How long is your sales cycle? Can the salesreps wait for a reasonable time before the customer data is present in Salesforce.com?
- What kind of customer model is used? Is Oracle Trading Community Architecture implemented and leveraged?
- Does the Organization have any address validation tools in place like DNB, Vertex on Oracle and Jigsaw on SFDC? If there are validation tools on only of the systems, then this will be good candidate for being the Master.

Depending on the process maturity and unique requirements of a customer, decision of the directionality of the interface should be made. It should also be recognized that the implementation decision may be to leave certain elements open for bi-directional updates – say contact names, contact address, phone number, email etc. and enforce stricter controls on customer and account data.

## Data Model

Difference in the customer data model between salesforce.com and Oracle e-Business Suite poses a significant challenge when designing and developing integration. A brief overview of the data model is provided here, so that readers appreciate the complexity.

### Oracle TCA (Trading Community Architecture)

Oracle e-Business Suite TCA is a framework that provides great flexibility for an organization to model customer data based on unique needs. The framework allows organizations to model customers with various degrees of complexity starting with a simple Customer (or Parties) using a single account and address to complex relationship where B2B and B2C customer data can be modeled showing levels of relationships.

Figure 1 shows an abridged version of the TCA model.

TCA provides many features like:

1. One-to-many relationship between customers or Parties.
2. Organization or person type parties.
3. Relationships between an organization and person either as a contact or as an employee.
4. Support for both B2B and B2C transactions with the ability to create accounts at both the organization and person level.
5. Support for multiple accounts per party with different billing and shipping addresses.
6. Multiple addresses and address types.
7. Supports addresses at both the party (an entity that may or may not have financial relationship with the organization) or an account (which is an entity that has financial relationship with the organization)
8. Several contact points (or communication types) at person, organization level.

Several Oracle e-Business suite customers have leveraged the TCA model including all the flexibility it offers. However, this creates a significant challenge when integrating with external applications like Salesforce.com.

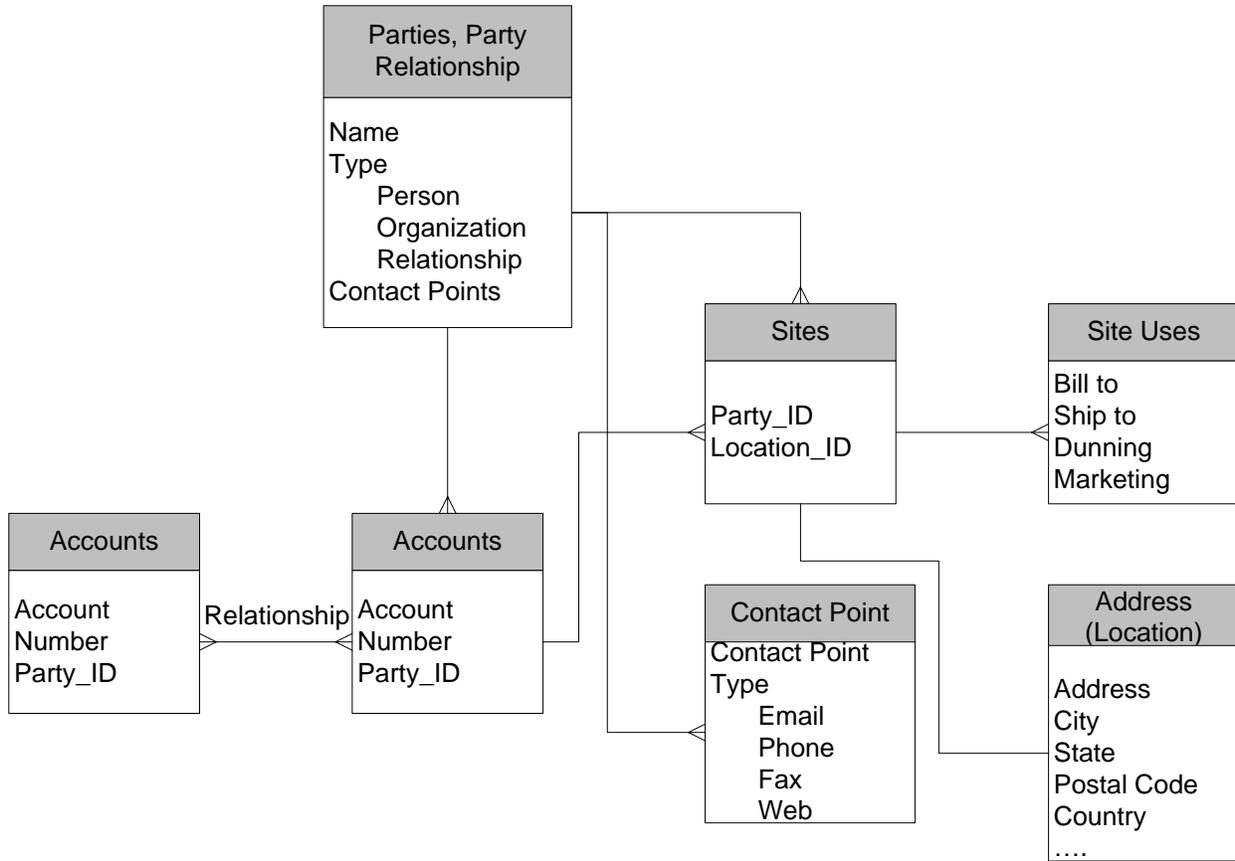


Figure 1. Simplified Oracle TCA Model

**Salesforce.com Account Data Model**

Salesforce.com uses a simpler model of Accounts and Contacts with their associated email, phone numbers, and addresses.

Salesforce.com supports a one level Account relationship (Parent of) and a reporting relationship definition on the contact. A simplified data model of salesforce.com account and contact is shown in Figure 2.

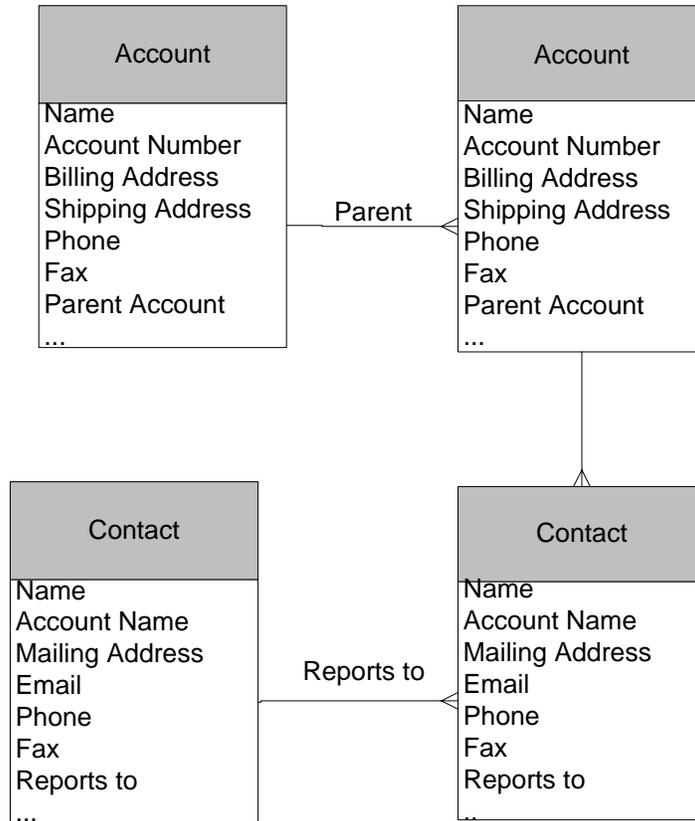


Figure 2. Salesforce.com Account and Contact Data Model

With such disparate data model across the two applications, it is imperative to analyze the existing data on either application – specifically on Oracle e-Business suite before defining the mapping. This paper presents a few data mapping scenarios for customers.

### Customer/Account Mapping Types:

#### Account to Account Mapping

In this type of mapping, salesforce.com account is mapped to an Oracle customer account. This is applicable to customers that use traditional Oracle “Customer” data model where each customer has an account. The mapping elements are shown in Figure 3.

This mapping assumes:

- Every party in Oracle has an account.
- There is a single hierarchy of customers in Oracle e-Business Suite i.e. a party has a single account, which has a single bill to and ship to address.
- An account is always created in Oracle when an account is created in salesforce.com.

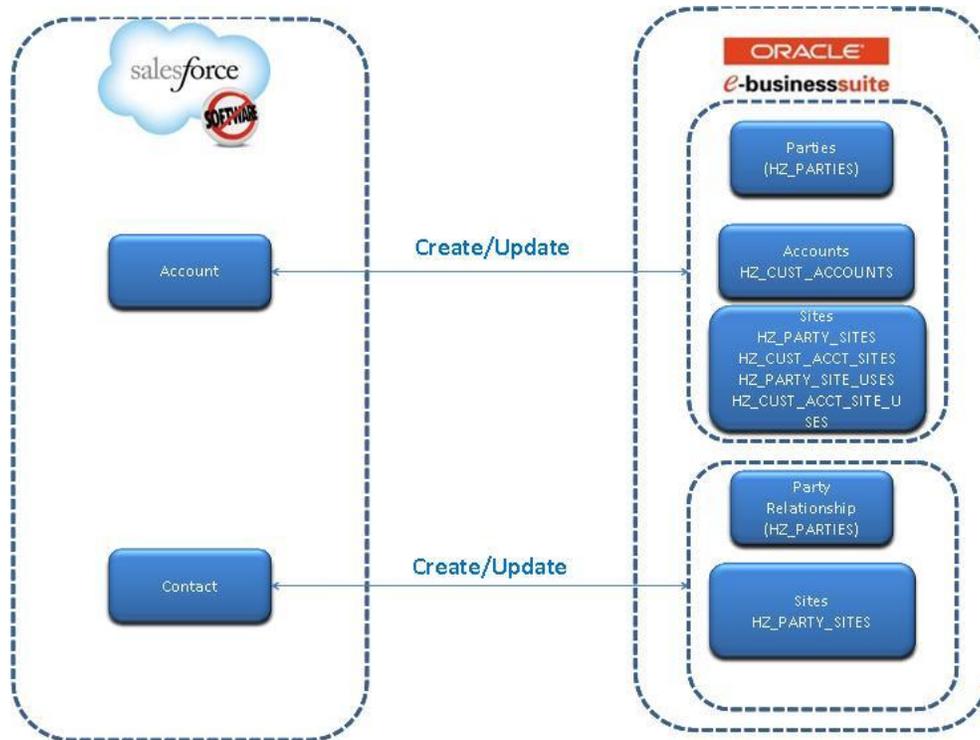


Figure 3. Account to Account Mapping

### Account to Oracle Party Mapping

In this mapping scenario, Salesforce.com account is mapped to a party record in Oracle. The schematic is shown in Figure 4. This mapping assumes:

- There is a single hierarchy of customers in Oracle i.e. a party has a single account, which has a single bill to and ship to address.
- Interface is triggered when a party is created in Oracle or when an account is created in salesforce.com
- Account is created in Oracle when a financial transaction is executed in Oracle like an Order placement, contract execution etc.
- Account creation in Oracle does not trigger an interface transaction.

### Address Mapping

In Oracle, an address could either be a party address or an account address with its site uses. Depending on the customer integration and directionality, salesforce.com address is mapped. Inherently, Salesforce.com supports only the bill to and ship to site uses. An account site is created when account to account mapping used with a corresponding party site. A party site alone is created when Account to Oracle Party mapping used. Account site creation does not trigger a transaction in salesforce.com.

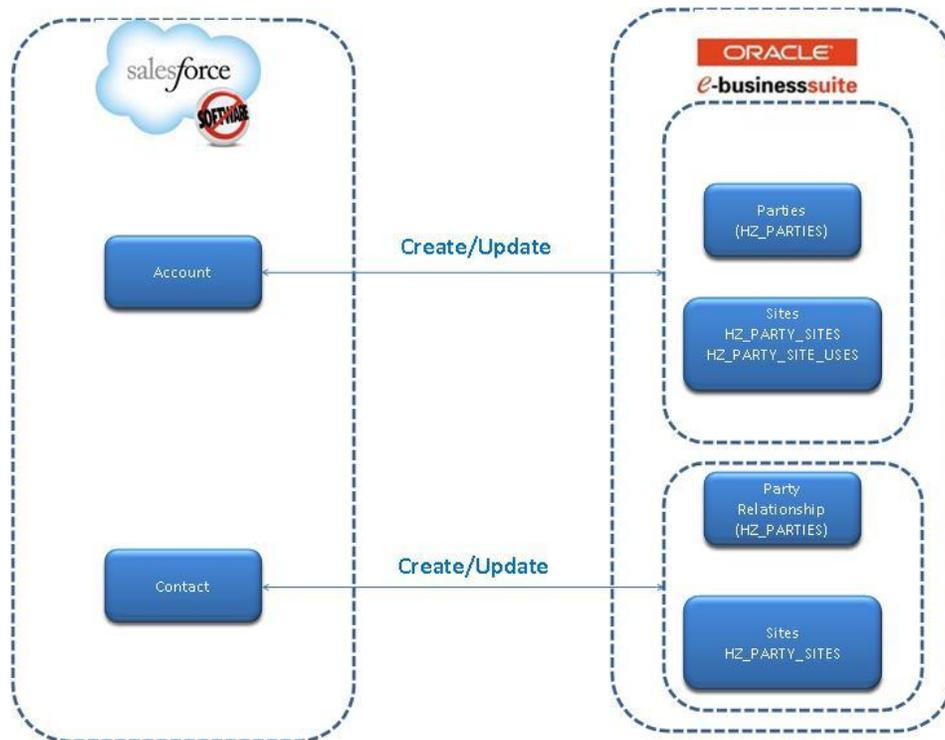


Figure 4. Account to Oracle Party Mapping

### Contact Mapping

Both Salesforce.com and Oracle support multiple contacts per account or customer. In Oracle contacts or person parties could have multiple addresses and accounts, which is not supported inherently in salesforce.com. If your Organization has both B2B and B2C type of a business model and require maintaining this relationship, some customization is required.

### Scenarios

The below table outlines different scenarios and where gaps could exist if using Oracle e-Business suite or Salesforce.com out-of-the-box.

#	Selling Model	Customer Model/Process	Analysis
1	B2B or B2C	Traditional Oracle Customer data model not leveraging TCA.	Use Salesforce.com to Oracle Account Mapping. <b>For B2C:</b> Person Account used in salesforce.com. This is not available as a default and has to be requested. Person Accounts are specific record types in salesforce.com
2	B2B or B2C	Using TCA but there is only one account per party in Oracle. Account created at the time a financial transaction is executed.	Use salesforce.com to Oracle party mapping. <b>For B2C:</b> Person Account used in salesforce.com. This is not available as a default and has to be requested. Person Accounts are

#	Selling Model	Customer Model/Process	Analysis
			specific record types in salesforce.com
3	B2B and B2C	Traditional or TCA model used with one account per party.	Can use either salesforce.com account to Oracle party or Oracle account mapping. <b>For B2C:</b> Person Account used in salesforce.com. This is not available as a default and has to be requested. Person Accounts are specific record types in salesforce.com
4	B2B or B2C	Leveraging TCA model using Party relationships, account relationships.	Requires building custom objects to replicate in salesforce.com
5	B2B and B2C	Leveraging TCA model using Party relationships, account relationships	Requires building custom objects to replicate in salesforce.com

### Custom Objects in Salesforce.com

Based on the customer integration needs, some custom objects may be needed in salesforce.com. The objects are listed below. This list could vary based on business requirement and is provided only as a general guideline.

#	Object	Description	Recommended Fields
1	Party Relationship	Custom object to support one to many party relationship.	Account (lookup value from accounts object) Related Account (lookup value from accounts object) Relationship Type (Picklist)
2	Communication or contact points	To store multiple contact points. Contact Point detail can be displayed on the hover on the contact or account screen.	Association Type (Picklist either account or contact) Related Object (Lookup value either account or contact) Communication type (Picklist) Email, Fax, URL, Phone
3	Additional Address	Salesforce.com standard account and contact object supports one billing and one shipping address.	Association type (picklist either account or contact) Address type (Picklist, values depends on business requirement like Installed Address, Marketing address etc. )

#	Object	Description	Recommended Fields
			Address (of Address datatype)

## Account Merge

Both Salesforce.com and Oracle offer the ability to merge duplicate accounts or parties as the case may be. It is highly recommended to perform merges only in one application and remove access to merge in the other. Performing merges in both applications is very difficult to manage.

The recommended practice:

- For single directional integration, where Oracle or Salesforce.com is the master and the other the slave. Perform merge in the master application first. Identify the corresponding candidates to be merged in the slave application and perform the merge manually.
- For bi-direction integration perform merge manually in both applications.
- Oracle provides API for initiating party merges. For additional information on the merge API please refer the SOA Gateway (Integration Repository) user's guide. This however is not a recommended practice. Organizations should have a robust customer merge strategy in place and perform merges in both applications manually.
- After performing merges ensure the mapping keys (example Oracle party\_id on the salesforce.com account record) is correct after merge is performed. A post update script may be required to keep the merged records in synch.

## Quoting Integration – Presentation Layer

Opportunity to Quotes conversion is a standard business in most sales cycles. Salesforce.com offers a simple quoting functionality out-of-the-box, which captures data from the opportunity and supports printing a quote document from the opportunity. Pricing functionality in salesforce.com is very limited and does not support automatic discounting for repeat orders, inclusion of free items (for example, buy one get one free) which Oracle supports using Advanced Pricing. This functionality may not be utilized by all customers, but if used, it is easier with Oracle's Quoting engine rather than implementing this functionality in Salesforce.com.

Quoting functionality is usually done with presentation layer integration. In presentation layer integration, users are seamlessly transferred to Oracle Quoting module from salesforce.com.

Key considerations/steps when designing the presentation layer integration:

- Determine if presentation layer integration is the right solution
  - Do you have a Quoting process in place?
  - Who enters the Quotes/Orders in your organization?
  - How complex is your pricing model?
  - Do you use Oracle Configurator module and is it part of your quoting process?
- Determine the authentication mechanism when control is transferred from salesforce.com to Oracle.
  - Should a DMZ box be in place?
  - Is authentication done through Active Directory validation?
  - How is the context (User ID, Responsibility ID and Session ID) passed to Oracle?
  - Salesforce.com provides delegated authentication web services API to support authentication through Oracle or Active Directory.
- Determine the Opportunity stage where Quote creation will be enabled

- What data is required to be passed to Oracle?
  - Account
  - Opportunity Product line
  - Opportunity Contacts
- Is there any user interface change required in Oracle Quoting so that users experience the same look and feel when switching from salesforce.com to Oracle?

## **Territory Synchronization**

Both Oracle e-Business Suite and Salesforce.com provides robust territory management functionality. Territory synchronization across these applications is required only when Territory management is used for other purposes like compensation computation.

Salesforce.com does not provide Territory APIs out of the box, so territory APIs have to custom built.

## **Historical Information/Bulk Transfers**

Several other customer-centric objects like Order History, Invoice History, Contracts, Install Base records etc. are generally uni-directional and can be imported using bulk or standard APIs (depending on data volume) available with salesforce.com. In general, custom objects should be built to display this information and can leverage territory management rules to secure access to this data.

Custom objects can be bulk loaded in salesforce.com passing the right reference to metadata.

## **Release Considerations**

Salesforce.com CRM applications is upgraded every three months. These releases could include a new version of the API. In general, API versions are not upgraded with each salesforce.com update but follows its own cycle.

For example, the current Winter '12 release includes Web Services API version 23 and the Summer '11 release included API version 22. If the integration is built on version 23, it will continue to work in the future releases of salesforce.com. Salesforce.com supports the API for three year duration from the time of first release at which time the APIs could be de-supported.

Oracle e-Business Suite undergoes frequent and the release cycle depends on the user Organization policies and procedures. Based on the APIs used on the Oracle e-Business suite, it is recommended to test the integration before migration to production.

## **Use of Middleware for Integration**

Several customers considering salesforce.com integration with Oracle are faced with the question if middleware is necessary for integration. There are several middleware products available and the use depends on organization's specific requirement. This has to be evaluated against the cost of procurement, installation, implementation and maintenance of the middleware software.

The following features are generally available in most middleware products:

- Metadata driven mapping
- Data translations without coding
- Integration mapping and flow building using Microsoft Office products.
- Advanced scheduling capabilities

## Integration Case Study

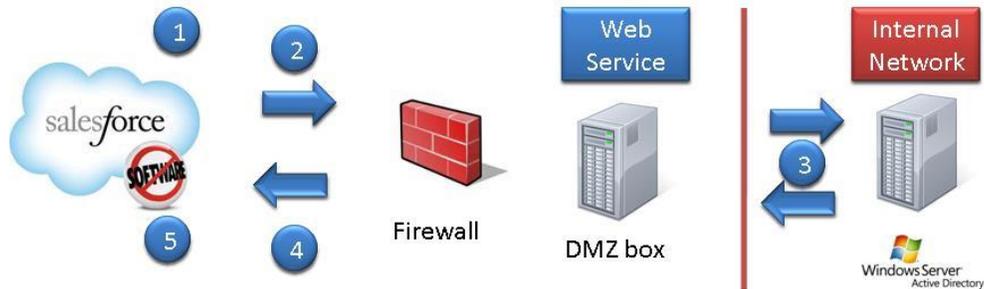
This case study outlines the salesforce.com integration with Oracle e-Business suite. The client name and demographics are withheld at the request of the client.

- Number of salesreps: 7000
- Customer location: US
- Industry: Office automation Product and Services
- Oracle e-Business Suite version: Release 12.1.2
- Integration enabled:
  - Customer unidirectional Integration with Salesforce.com customer to Oracle party mapping and Oracle as the Master.
  - Oracle Quoting presentation layer integration
  - Historical Orders, Contracts and Install Base
  - Territory Synchronization using custom APIs
- Dedicated customer management team.

The integration project was launched along with implementation of salesforce.com. The customer has attempted to implement Oracle sales applications in several versions (11.5.X) without success. The problems centered on process adoption and the culture change required in the sales organization. The customer decided to use salesforce.com to support 7000 salesreps dispersed nationwide. The project was implemented in phases for a pilot group of salesreps. Several customizations were needed to make salesforce.com work for this organization.

### Integration Architecture

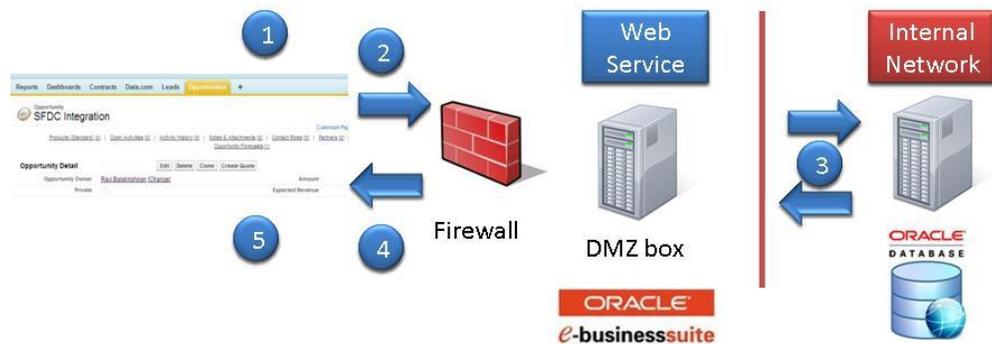
The integration architecture is shown in the Figures below. Figure 5 shows the Delegated Authority functionality implemented in salesforce.com using Microsoft Active Directory authentication. Figure 6 and 7 show the integration architecture for Quoting Integration and Context switching for Quoting access respectively.



- 1 User enters username and password in salesforce.com
- 2 Salesforce.com passes the credentials to the DMZ box
- 3 DMZ box passes these credentials to Active Directory through web service
- 4 Web Service returns pass/fail back to Salesforce.com
- 5 If passed salesforce.com allows user through else denies access

Figure 5. Delegated Authentication

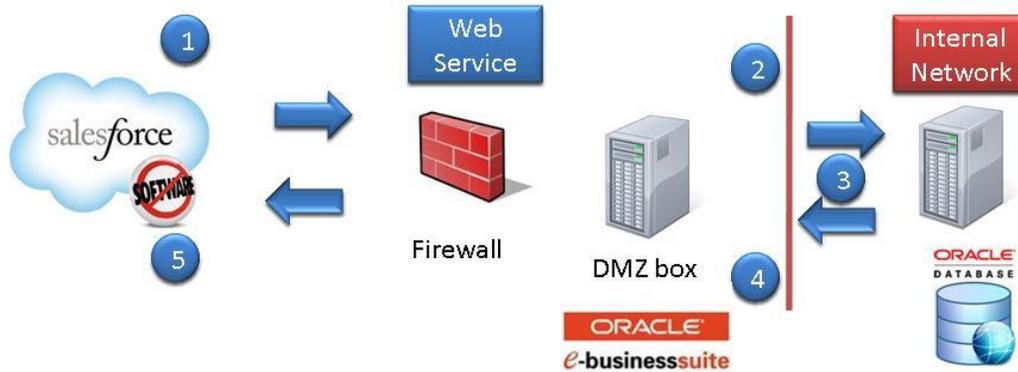
Delegated Authentication is fully supported by salesforce.com. When a user enters username and password, salesforce.com sends this authentication information in a web services API that can be used to validate the authentication and return a pass/fail message back. Delegated authentication is generally in place where single sign on is used. In addition, salesforce.com also issues security tokens for unknown PCs and can secure access by authorized IP ranges.



- 1 User clicks Create Quote on the salesforce.com opportunity page
- 2 Salesforce.com passes the credentials to the DMZ box
- 3 DMZ box passes these credentials to Active Directory through web service. Web service also checks status of Oracle EBS
- 4 Web Service returns Employee ID/Resource ID if the user is still active
- 5 If passed SFDC lets user to the Oracle R12 and opens the Quote page

Figure 6. Presentation Layer Quoting Integration

The customer used presentation layer integration with Oracle Quoting. Oracle HTML quoting screens was customized to provide the same look-and-feel as salesforce.com to give salespersons a seamless experience. Two main aspects of the quoting integration were the authentication and context switching. Context switching is providing the correct application ID, user ID and responsibility ID context to Oracle quoting so that the application is able to get the required profile options, roles and security rules.



- 1 A new Tab is launched from salesforce.com to display Oracle Quoting with employee ID
- 2 A new user session is created using Guest User
- 3 Database is checked for user ID and corresponding employee ID
- 4 If record is found, context is switched to the new User ID, Responsibility and Application ID
- 5 If passed salesforce.com then Quoting UI is displayed.

Figure 7. Quoting Integration: Context Switching

### Implementation Challenges

The project had some significant challenges during implementation and ongoing maintenance.

- **Data Quality**  
Prior to the implementation and migration of customer data to salesforce.com from Oracle an extensive data quality exercise was undertaken. Customer already had a dedicated customer master team in place. Identification of duplicates and merges were performed before migration of data. This continues to be a challenge when maintaining disparate customer data.
- **Data Volume**  
Customer had several thousands of party records to migrate to salesforce.com. The amount of contract and install base volume posed a significant challenge in terms of validation and reconciliation and assignment based on territories. If bring over historical records like Invoices, Orders, Contracts and Install Base for viewing in salesforce.com consider this like a data migration activity and bring enough information that is useful to the salesperson. Normally old invoices/orders may not be useful information in a sales pursuit.

### Project Checklist

In order to get the integration right and tailored for your business needs, consider the following steps in your integration project.

1. The focus of the integration project should be on getting the customer integration right. Decide on directionality, the master application.
2. Setup a process for customer update.
3. Determine if a middleware is required in your environment. If yes, go through the product selection.
4. Determine security and authentication requirements.
5. Focus of data quality and cleanup. Plan to cleanup data in Oracle before migration to salesforce.com
6. Identify other data required in salesforce.com like Orders, Contracts etc.
7. Determine custom objects required in salesforce.com
8. Build the objects, integration, test and implement.

## **About the Author**

Ravi Balakrishnan is an experienced information technology professional. He is currently involved in managing ERP/CRM implementation and integration projects in Oracle e-Business Suite and Salesforce.com. He co-founded Prisio Technologies, a value driven services firm providing implementation, upgrade and integration solutions in salesforce.com and Oracle e-business suite.

Prisio Technologies offers accelerated and configurable Salesforce.com – Oracle integration solution using direct web services based integration without the need of a middleware. For additional information and inquiries, please contact Ravi at [ravib@prisiotechologies.com](mailto:ravib@prisiotechologies.com).