

# Veeva CRM Integration

## Technical Documentation Guide

### 1 Overview

This document outlines the technical documentation requirements for a Veeva Partner integration with Veeva CRM. It is intended as a guide on the common components of product documentation for integrations into the Veeva software and should not be considered exhaustive. Veeva partners should determine what information is appropriate and pertinent to their integration and expand on those topics.

### 2 Audience

The technical documentation document is designated for Veeva Internal Audience only. The document will be evaluated by the Veeva Product Alliances team and Veeva CRM Product Management team. This document will also be used by Veeva Services/Implementation teams working on customer deployments involving the partner integration. For this purpose, it is recommended the document assume the reader does not have prior knowledge of the integration.

### 3 Suggested Components

#### 3.1 Table of Contents

#### 3.2 Integration Definition

Version number being certified

Integration Overview

Project Stakeholders / Technical Contacts

Business Justification

#### 3.3 Integration Capabilities

To help us understand the scope of the integration, please list and describe all the integration points or functionality the integration will cover.

#### 3.4 Architecture

A complete description of the integration architecture.

- How is the Integration hosted, who are the cloud vendors, middleware, etc

Provide a complete Architecture diagram

- In a table below the diagram, define and describe each Architecture component, its role and capabilities

Integration Flow

- Outline the step by step flow of events through the architecture diagram.

### 3.5 Workflows

Process Flow Diagram

- Integration may involve separate processes. Make sure there is a diagram for each.

Logical Data flow

- Make sure to include the business processes that deal with each area of integration outlined in section 3.3.

### 3.6 Authentication and Session Management

- Describe the Authentication model used by the integration
  - o Type of User
  - o Single Sign On – OAuth or SAML via API
  - o How are credentials stored
- How are Sessions handled
  - o Expiration
  - o Re-Authentication
  - o It is a best practice to cache and reuse the session ID to maximize performance rather than obtaining a new session ID for each call.
- Permissions required
  - o In a table or tables, outline all Salesforce User permissions required for the integration to function.

### 3.7 Data access

- For each integration point:
  - o What objects / fields are being accessed (table format)
  - o What data is required / optional
  - o What data is passed back to Veeva
  - o Describe the Usage of data for each process

### 3.8 Error Handling

- Approach
  - o Define the error handling strategy for each part of the integration (UI, background process, etc..)
  - o Specify how the errors are trapped, reported and handled
  - o Mechanism for resolution
  - o Clear direction for Administrators on error resolution
  - o Outage handling
- Error Logging

- How can errors be traced through the integration
- Do the logs provide actionable information
- Failure recovery
  - Can the integration recover automatically
  - What steps are required to re-run the integration
  - Re-Processing failed transactions

### 3.9 Efficiency

- Using the API in an efficient manner is a major concern to Veeva customers who must contend with Salesforce API limits in their ORG's.
- Describe how the integration aims at reducing the number of API calls necessary to achieve a certain task
  - Using same session
  - Bulk API where appropriate
  - Careful use of nested queries
- How much API traffic can be expected from a typical implementation
- Benchmarking data, typical number of calls in 24hr period.

### 3.10 Setup and Installation

- Identify the resources required to get the integration live at a customer
  - What teams need to participate from the Customer side (IT, Admin, Business), from Partner side (Implementation, Services) and from the Veeva side (Services, Account Teams)
  - Varying levels of implementation – Is there a quick start package
  - A tiered approach to implementation
  - Outline plans for configuration testing
- Configuration - Create a table showing every required permission at object and field level that the integration requires to function
  - Consider data in related objects
  - Think of the correct level of access required for each object and field (CRUD)
  - Consider supporting objects, like User and Account
- Flexibility
  - Describe how custom fields are handled for a customer
  - List and describe any fields that are hard coded
- Installation
  - Outline the distribution method for the integration
  - Describe how the integration is packaged