



# Integrating Salesforce with SAP

Patterns, Tools and Considerations

By Saleem Rauf  
Suchin Rengan  
Ralf Schundelmeier  
Sibasisha Padhy  
Siddharth Muzumdar

# Contents

<b>INTRODUCTION</b>	<b>3</b>
<b>COMMON BUSINESS SCENARIOS</b>	<b>4</b>
Accounts and Contacts Synchronization	
Bidirectional Account Creation and Maintenance	
Product Pricing and Quoting	
Opportunity-to-Order Processing	
Post-Order Processing	
Financial Data Visualization	
<b>KEY INTEGRATION CRITERIA &amp; CONSIDERATIONS</b>	<b>10</b>
Step 1 – Understand the Business Context	
Step 2 – Identify the Integration Layer	
Step 3 – Select a Services Layer Integration Approach	
Step 4 – Select an implementation Approach	
<b>INTEGRATION OPTIONS</b>	<b>13</b>
Salesforce Integration Options	
SAP Integration Options	
Salesforce & SAP Integration Options by Layer	
<b>INTEGRATION DECISIONS</b>	<b>19</b>
Authoritative Sources	
Integration Pattern Decision Matrix	
<b>REFERENCE ARCHITECTURE</b>	<b>22</b>
<b>SUMMARY</b>	<b>23</b>
<b>NEED MORE INFORMATION?</b>	<b>24</b>
<b>HOW CAN SALESFORCE HELP?</b>	<b>25</b>
<b>ABOUT THE AUTHORS</b>	<b>26</b>
Glossary	
<b>APPENDIX</b>	<b>27</b>
Salesforce Integration Capabilities	
Integration Patterns	

# Introduction

This white paper provides an overview of the most common business scenarios, key integration criteria, and considerations for integrating Salesforce with SAP. In addition, it introduces integration options, approaches, and patterns. The framework laid out here will help CIOs, CTOs, chief architects, and enterprise architects define the optimal integration approach and architecture to integrate Salesforce with SAP.

As one of the most popular enterprise resource planning (ERP) solutions on the market, SAP has enabled organizations to optimize many of their business functions in finance, human capital management, manufacturing, sales and distribution, supply chain, and supplier relationship management.

Salesforce is the leading Customer Success Platform, and organizations are using it as a system of engagement to provide the right information at the right time by unlocking back-office data and processes.

Sales, service, and marketing teams have a 360-degree view of their customers that ultimately leads to an enhanced customer experience.

This white paper is a guide to achieve this in a fast and cost-efficient way.

## CHAPTER 1

# Common Business Scenarios

Based on the large number of successful implementations of Salesforce, including integrations with SAP, we have identified common business use cases that our customers look to implement.

At a high level, here are common areas that Salesforce can integrate with SAP for data and process integration:

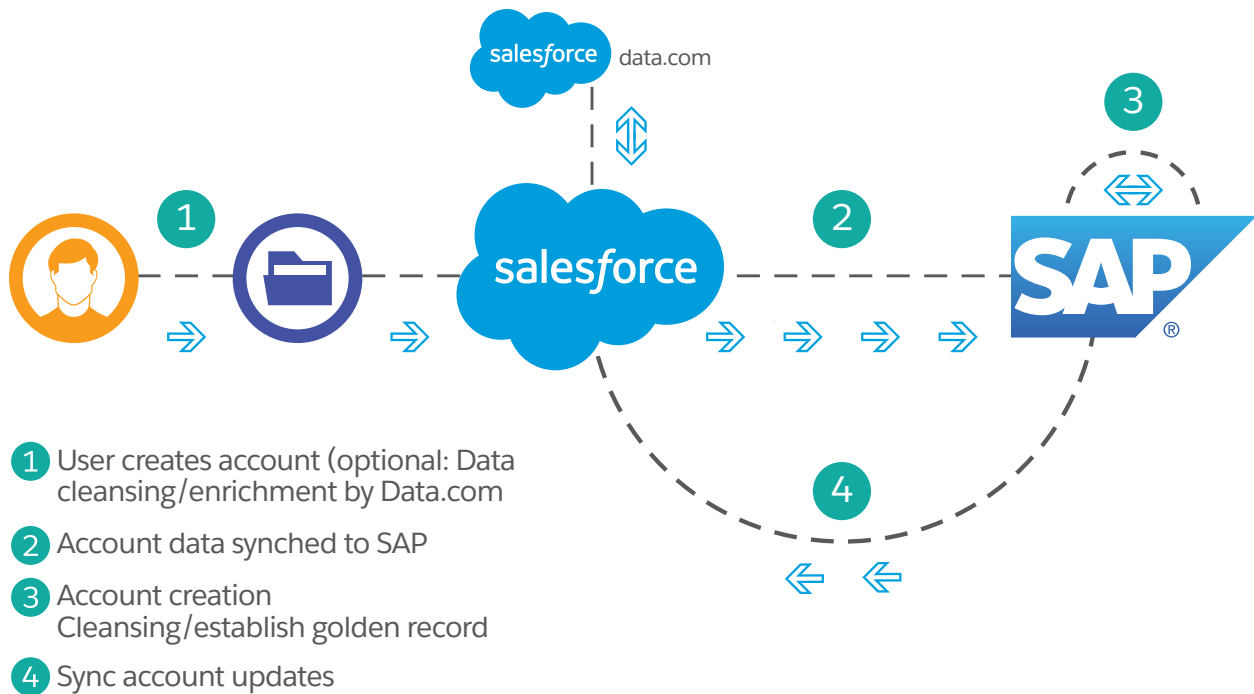
- Accounts/Account Partners
- Product & Pricing
- Order Management
- Demand Management
- Warranty Management
- Financial Data

## Accounts and Contacts Synchronization

One of the primary requirements for basic integration between these two systems is synchronizing account and contact data from SAP to Salesforce to ensure that Salesforce is up to date. We have observed with our larger enterprise customers that accounts are “mastered” outside of Salesforce in either a master data management (MDM) platform or another system of record, such as SAP. Salesforce is generally the system of record for contacts.

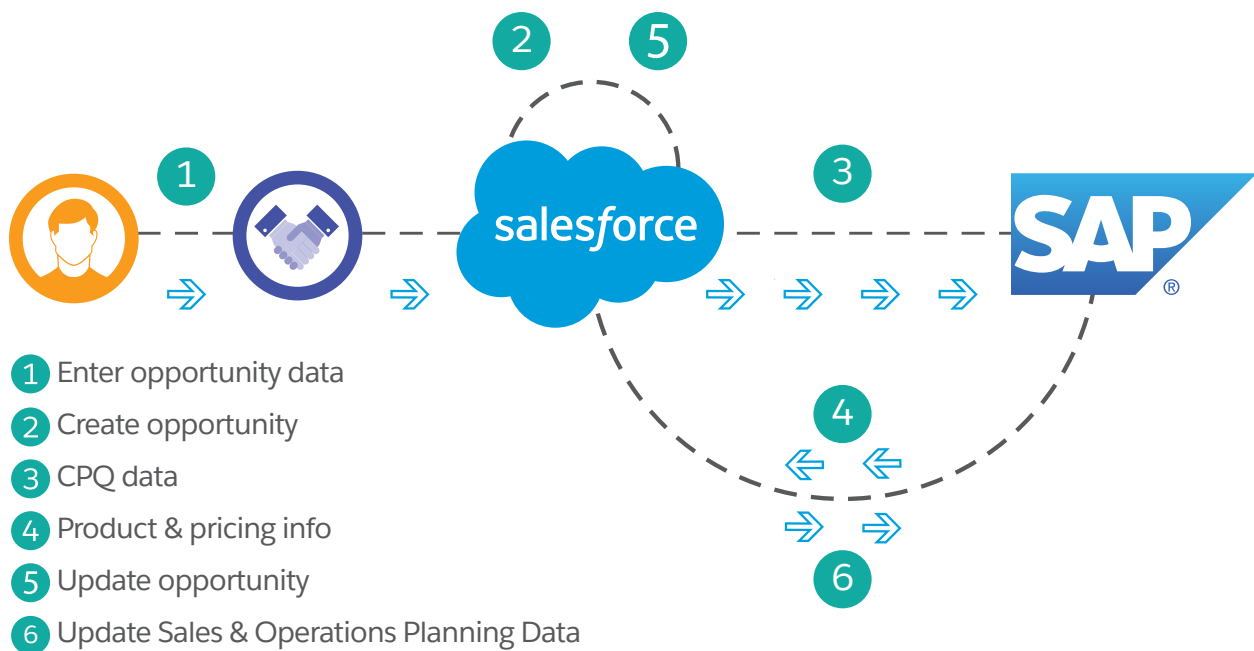
## Bidirectional Account Creation and Maintenance

Users need the ability to create and maintain accounts using the Salesforce platform. SAP account data and/or account data in an MDM platform need to stay in sync and obey certain controls and rules. Data stewards might also manually correct or authorize data changes before committing to the system of record.



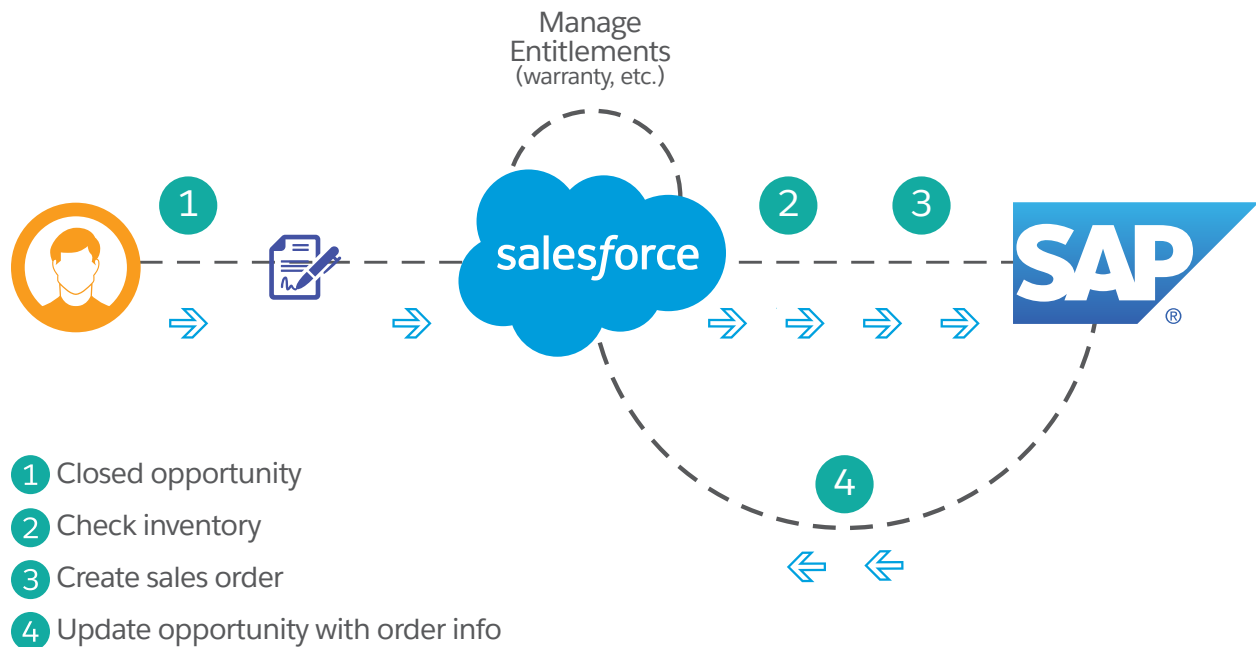
## Product Pricing and Quoting

Salesforce users, such as sales reps who are working on opportunities, need to generate quotes using products and price book information. They also need to associate those quote records to their corresponding opportunities in Salesforce. Product and pricing information is typically managed in SAP and needs to be synced with Salesforce.



## Opportunity-to-Order Processing

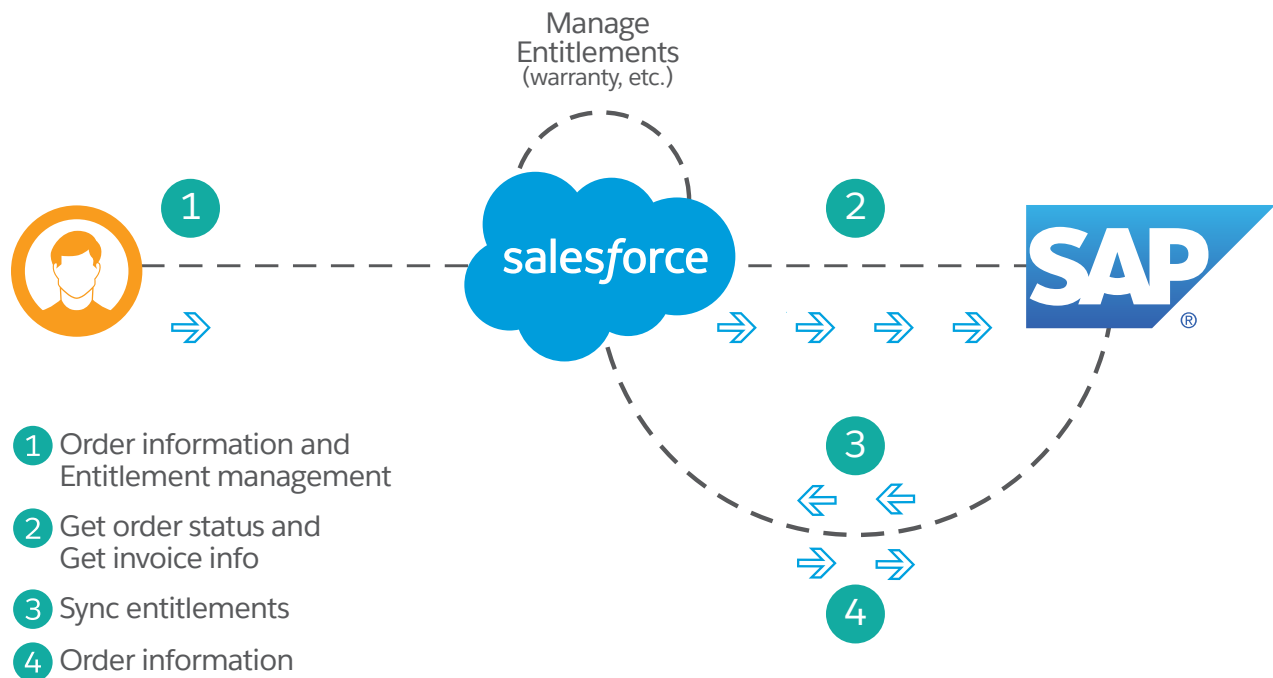
Salesforce users, such as sales reps and call center agents, need to process opportunities – from initiation to order confirmation and execution in SAP – through the Salesforce UI. This business process has multiple steps, such as checking inventory, closing the opportunity by creating an order, creating a contract, setting up the payment terms, and setting up entitlements.



## Post-Order Processing

After an order is processed in SAP, additional processes need to be supported that require integrations between Salesforce and SAP. For example:

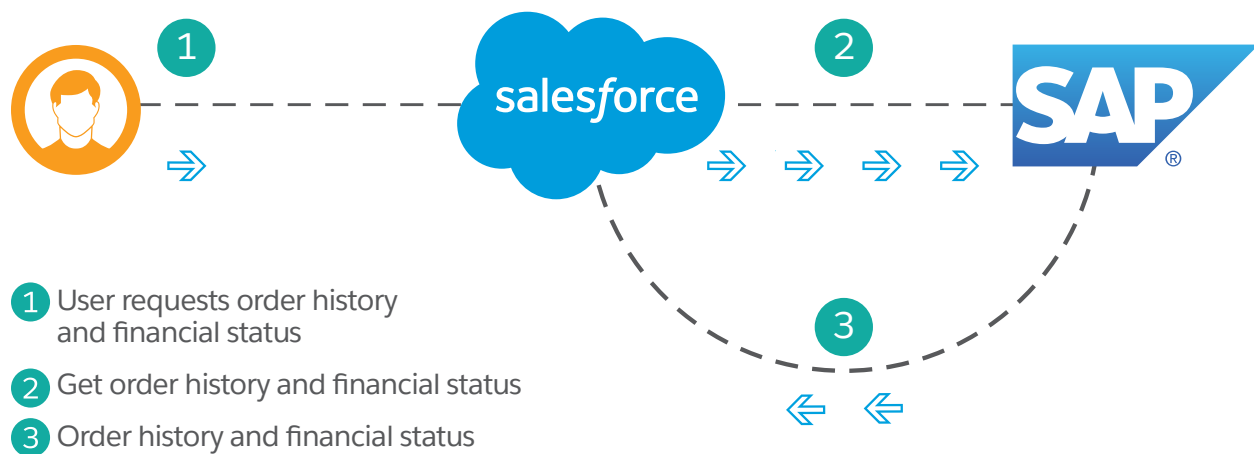
- Order status updates
- Managing contracts
- Managing entitlements
- Invoicing





## Financial Data Visualization

Every time a contact or account record is viewed in Salesforce, all relevant customer transactional data, such as related order history and the current financial status, needs to be accessible in real time to the Salesforce user. In many cases, real-time drill down into orders is a requirement.



**CHAPTER 2**

# Key Integration Criteria & Considerations

Before we dive into solutions, we'd like to introduce a high-level, four-step process to successfully integrate Salesforce with SAP. First and foremost, understand the three main business requirements that the integration should support.

## Step 1 | Understand the Business Context

**WHAT DATA ENTITIES ARE INVOLVED?**

Data entities describe the data objects, such as invoices, products, orders, and customers.

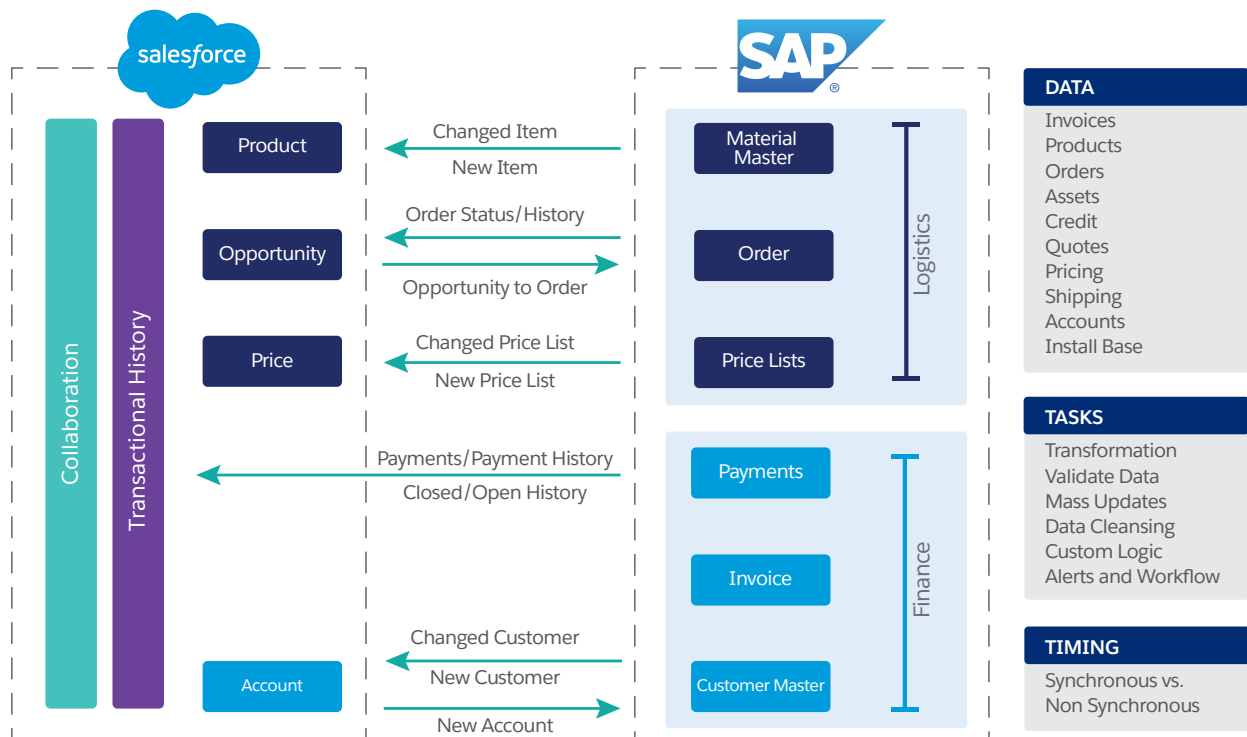
**WHAT ACTIONS/TASKS NEED TO BE PERFORMED?**

Tasks describe the activities that are being performed with these objects, such as transformations, validations, updates, and deletes.

**WHAT ARE THE TIMING REQUIREMENTS?**

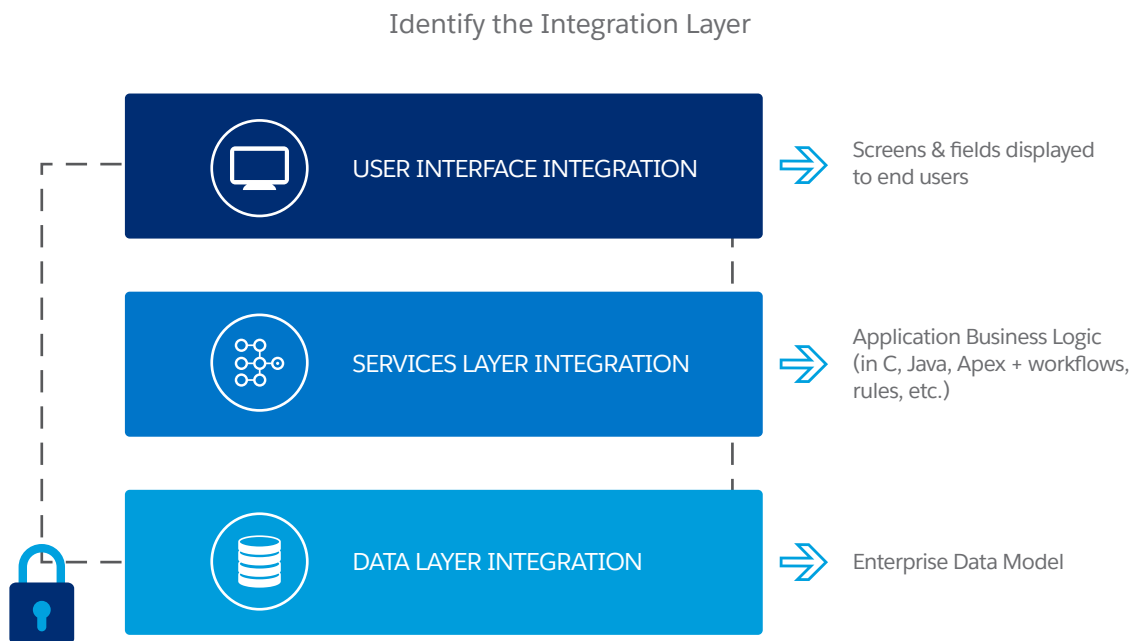
Timing describes whether the tasks are performed synchronously or asynchronously, whether they're batch- or event-driven, and how frequently they're performed.

Integration Scenarios Between Salesforce and SAP



## Step 2 | Identify the Integration Layer

The integration layer defines at which tier (in a three-tier architecture) the integration will be implemented. Selecting the correct integration technology layer is an important decision that significantly affects the cost, speed, and overall performance of your integration project. To understand which layer is appropriate for you, you need to know what options are available from each solution provider.



## Step 2 | Identify the Integration Layer – *continued*



### USER INTERFACE INTEGRATION

Integration at the user interface layer is also known as a mashup or composite integration. This approach is growing in popularity because of the simplicity of the implementation and its rapid time to value. A mashup combines data from different sources and displays data from one application in the user interface of another. Mashups generally use Web services to grab data from remote applications. More advanced solutions can also send information back when bidirectional access to data is required. Displaying data from the original source application is easier than transferring large volumes of data between applications and helps to address certain compliance issues.



### SERVICES LAYER INTEGRATION

Integration at the services layer continues to gain momentum due to simple access to modern application APIs, such as the Enterprise Services from SAP and Force.com APIs from Salesforce.

For integration at the services layer, applications are connected to each other via APIs. Instead of accessing data directly, these interfaces provide an abstraction layer and a series of rules for how the data is presented and shared with external applications. Integration solutions at the services layer often send and receive XML or JSON messages, either synchronously, where the sender waits for a reply, or asynchronously, where the message is sent to one or more recipients and the application flow continues without reply. A middleware solution is often used in this case to orchestrate communication; however, the use of middleware is not mandatory.



### DATA LAYER INTEGRATION

Connections between applications at the data layer are still the most widespread and commonly used form of integration, particularly among SAP customers who are very familiar with their master and transaction data and the underlying structure. Data integration generally involves moving data back and forth between databases of two or more applications, either in real time or in nightly batches.

Because Salesforce doesn't allow direct access to the underlying database, the entire integration solution runs at the data layer via the Force.com API. Integration between this API and the company's local applications is extremely common because it's simple, fast, cost-effective, and doesn't require development on the Salesforce side.

Although SAP technically allows direct access to the underlying database, this is not the recommended approach. As noted above, SAP provides access to business data via APIs. Data integration of SAP and Salesforce applications therefore typically occurs at the services layer.

## Step 3 | Select a Services Layer Integration Approach



### POINT-TO-POINT DIRECT INTEGRATION

In theory, the key interfaces provided by SAP and Salesforce can be directly accessed by software applications. In these types of architectures, consumers and providers are tightly coupled. This makes the integrations very brittle as the consuming applications need to change as the underlying environment changes over time.



### SERVICE-ORIENTED ARCHITECTURE (SOA) STACK

SOA stacks are an alternative for enterprise application integration. A large number of vendors offer these integration platforms. The primary advantage of utilizing an SOA stack is that it creates a relatively robust integration architecture that can address most use cases. Applications are loosely coupled so when changes are needed they can be quickly addressed. Application maintenance costs are lower than in point-to-point approaches, and the overall platform is more reliable.



### INTEGRATION PLATFORM AS A SERVICE (IPAAS)

As a cloud-based platform, iPaaS supports cloud-to-cloud, cloud-to-on-premises and even on-premises-to-on-premises integration scenarios. Also, iPaaS can handle business-to-business (B2B), electronic data interchange (EDI), and extract, transform, and load (ETL) integration requirements. As with SOA stacks, the applications are loosely coupled, providing more stability and flexibility.

## Step 4 | Select an Implementation Approach

Your choices are to “build or buy,” though in most cases the solution is a hybrid of both. Buy a solution from one of the many Salesforce integration partners, using Salesforce App Cloud-provided integration approaches and APIs or using integration toolsets to build your own custom integration from scratch.

**CHAPTER 3**

# Integration Options

## Salesforce Integration Options

To integrate with other enterprise systems, most customers use one or more of the following capabilities:

**SALESFORCE CONNECT**

This integration approach supports OData (Open Data Protocol) and can be used to connect and integrate various systems with ease – quickly and without any coding on the Salesforce side – via direct configuration in the administration interface of the Salesforce environment. It facilitates integration not just with SAP, but also with other systems for which an OData producer is available. This includes products such as Microsoft SharePoint or middleware solutions to integrate additional components. As OData is a standardized protocol, many software companies have already implemented appropriate integration points for OData. In case the systems to be integrated with do not support OData, you can use the Apex Connector Framework to develop your own custom adapter. This connector can retrieve data from external systems and synthesize data locally. Salesforce Connect then represents that data in Salesforce external objects, which your users and the Force.com platform interact with. The data that you expose as external objects from Apex can come from anywhere, for example you can retrieve data via Apex callouts, Web service callouts, SOQL, and SOSL.

**FORCE.COM APIS**

Use the Force.com API and a selection of innovative development tools to build the required integrations yourself. Salesforce provides toolkits for the most common programming languages – including Java, .Net, Ruby on Rails, and PHP – so that you have the programming flexibility to build any integration. You use the Force.com APIs directly or via a middleware platform.

Force.com works with all major middleware platforms. The list of integration solutions on the Salesforce AppExchange includes a large number of certified middleware solutions, many of which provide specific Salesforce CRM connectors.

## Salesforce Integration Options – *continued*



### APEX

You can use Apex to create Web services that clients can invoke. You develop the Apex methods and logic, then expose them as Web services. External applications can invoke these Web services to perform custom logic.

You can also use Apex callouts to invoke external Web services. With callouts, Apex provides integration with external web services that utilize SOAP and WSDL, or HTTP services (RESTful services). You can import WSDLs to auto-generate the corresponding Apex classes. Additionally, Apex supports HTTP services to use `HttpRequest` and `HttpResponse` objects to invoke the external Web service.



### FORCE.COM CANVAS

Force.com Canvas enables you to easily integrate a third-party application in Salesforce. Force.com Canvas is a set of tools and JavaScript APIs that you can use to expose an application as a canvas app in Salesforce. This means that you can take your new or existing applications and make them available to your users as part of their Salesforce experience.

Instead of redesigning and reintegrating your external applications, you can now use these tools to integrate your technology within Force.com Canvas.

The third-party app that you want to expose as a canvas app can be written in any language. The only requirement is that the app has a secure URL (HTTPS).



### VISUALFORCE

Visualforce is the component-based user interface framework that allows developers to build sophisticated, custom user interfaces that can be hosted natively on the Force.com platform. The Visualforce framework includes a tag-based markup language that's similar to HTML and a set of server-side standard controllers that simplify basic database operations such as queries and saves.

Visualforce uses the traditional model-view-controller (MVC) paradigm, with the option to use auto-generated controllers for database objects, providing simple and tight integration with the database. You can write your own controllers, or extensions to controllers, using Apex code. Via custom Apex controllers/ classes, calls can be made to external Web services to retrieve and display third-party data side by side with Salesforce data.

## Salesforce Integration Options – *continued*



### LIGHTNING COMPONENTS

The Lightning Component framework is a UI framework for developing dynamic Web apps for mobile and desktop devices. It's a modern framework for building single-page applications that are engineered for growth. The framework supports partitioned, multi-tier component development that bridges the client and server. It uses JavaScript on the client side and Apex on the server side.

Components are the self-contained and reusable units of an app. They represent a reusable section of the UI, and can range in granularity from a single line of text to an entire app. The framework includes a set of prebuilt components. Via custom Apex Controllers/Classes, calls can be made to external Web services to retrieve and display third-party data side by side with Salesforce data.



### WORKFLOW OUTBOUND MESSAGING

Outbound messages let you specify that changes to fields within Salesforce can cause messages with field values to be sent to designated external servers.

An outbound message is an action for workflow rules in Salesforce. Workflow rules watch for specific kinds of field changes and trigger automatic Salesforce actions, such as sending email alerts, creating task records, or sending an outbound message.

The key benefit of using outbound messages for implementing integrations with an external system is that the configuration is completely driven on the Salesforce side.



## SAP Integration Options

SAP provides standard documents, data structures, and function modules to support integration from SAP to SAP, and from SAP to third-party software solutions such as Salesforce. The most frequently used options are described below.



### INTERMEDIATE DOCUMENTS (IDOCs)

IDocs is a standard data format defined by SAP for the exchange of information between SAP and non-SAP applications. IDocs are typically used when information needs to be sent to or from SAP without notification requirements. IDocs are primarily used to transfer master data in and out of SAP, for example, to retrieve suppliers, cost centers, activity types, and logistics information such as the *material master* or *bill of materials*.



### BUSINESS APPLICATION PROGRAMMING INTERFACE (BAPI)

BAPIs are defined interfaces that can be called by SAP applications and non-SAP applications. BAPIs are typically used in synchronous scenarios where two-way communication is required. For example, creating new orders or changes to existing orders could be done via a BAPI.



### ODATA

OData (Open Data Protocol) emerged into an open Web services standard, which provides an open protocol for sharing data. SAP supplements the data types that are used by OData with extra information from the ABAP Data Dictionary. Typically, an OData service is built based on BAPIs, meaning that the BAPI is exposed and consumed using the OData format.



### ADVANCED BUSINESS APPLICATION PROGRAMMING (ABAP)

ABAP as a foundation for many applications offers a broad range of integration and connectivity technologies for remote SAP and non-SAP systems. Universal Internet protocols, such as HTTP(s), and data formats, such as XML and SOAP, can be used as well as SAP-proprietary protocols and formats such as RFC/BAPI, IDoc, and ALE/EDI. Developers can expose ABAP-based functionality as a Web or enterprise service by publishing the service definition in the Enterprise Service Repository, creating a server-side proxy, and implementing the service using the ABAP programming language.

## SAP Integration Options – *continued*



### ABAP WEB SERVICES

SAP NetWeaver Application Server for ABAP provides a standardized architecture and a set of tools for creating Web services and related objects. Existing BAPIs or remote-enabled function modules can be used to set up Web services, or you can develop new Web services in the Object Navigator of SAP NetWeaver AS for ABAP. There are predefined settings for securing Web services or the transports being used. ABAP Web services can be used for communication between SAP systems and between SAP and non-SAP systems.



### SAP JAVA CONNECTOR (SAP JCO)

SAP JCo can execute function calls that are inbound (such as when a Java client calls BAPI or RFM) and outbound (such as when ABAP calls a Java server). JCo supports synchronous, transactional, queued, and background RFC. The SAP JCo can be used either with SAP business connector (integrated component) or as a standalone component to connect an external Java application.

## Salesforce & SAP Integration Options by Layer

The table below is a tool to help you quickly determine which Salesforce and SAP technology to use for the integration layer you have chosen.

Integration Layer	SAP Technology	Salesforce Technology
<b>User Interface Integration</b>	<ul style="list-style-type: none"> <li>- SAP Enterprise Portal</li> <li>- SAP NetWeaver Composition Environment</li> <li>- UI screens based on Web Dynpro (WD)</li> </ul>	<ul style="list-style-type: none"> <li>- Visualforce Pages (Mashups)</li> <li>- Lightning Components</li> <li>- Force.com Canvas</li> </ul>
<b>Services Layer Integration</b>	<ul style="list-style-type: none"> <li>- SAP APIs (RFCs and BAPIs)</li> <li>- iDocs</li> <li>- Enterprise Services</li> <li>- Custom Web Services created with SAP NetWeaver</li> <li>- OData Connector (NetWeaver Gateway)</li> <li>- NetWeaver PI/PO</li> </ul>	<ul style="list-style-type: none"> <li>- Force.com APIs (Web service interfaces, SOAP, and REST)</li> <li>- Force.com Apex Callouts</li> <li>- Outbound Messages</li> <li>- AppExchange 3rd party Middleware</li> <li>- AppExchange 3rd party API Gate ways</li> </ul>
<b>Data Layer Integration</b>	<ul style="list-style-type: none"> <li>- Direct access to the underlying database</li> <li>- SAP Data Services</li> </ul>	<ul style="list-style-type: none"> <li>- Salesforce Connect</li> <li>- Force.com APIs</li> <li>- Data Loader</li> </ul>

## CHAPTER 4

## Integration decisions

## Authoritative Sources

Before we jump into the details of which patterns are best suited for each use case, let's highlight the role that each system plays for the entities that are required to implement the business requirements.

Entities	Data Providers		
	Authoritative Data Source <sup>1</sup>	Contributing Systems <sup>2</sup>	Consumers <sup>3</sup> (Read-Only)
Accounts	SAP	Salesforce	N/A
Contacts	Salesforce	SAP	N/A
Opportunity	Salesforce	N/A	N/A
Products	SAP	N/A	Salesforce
Price books	SAP	N/A	Salesforce
Contracts	SAP	Salesforce	N/A
Entitlements	Salesforce	N/A	N/A
Orders	SAP	N/A	Salesforce
Quotes	SAP	N/A	Salesforce
Order Line Items	SAP	N/A	Salesforce
Financial Data (payment history, credit)	SAP	N/A	Salesforce
Invoices	SAP	N/A	Salesforce
Shipping Information	SAP	N/A	Salesforce
Inventory	SAP	N/A	Salesforce
Assets	SAP	N/A	Salesforce

<sup>1</sup> Authoritative Data Sources: Cohesive set of data assets that provide trusted, timely, and secure information to support a business process. Information is visible, accessible, understandable, and credible to information users [Source: MIT].

<sup>2</sup> Contributing Systems: Systems that create, update, and delete data but must sync with Authoritative Data Sources to create the golden copy of the record.

<sup>3</sup> Consumers: Systems that reference/read data to support business processes but do not modify it.

## Integration Pattern Decision Matrix

The following table lists the patterns to help you determine the pattern that best fits your integration requirements. Please refer to the appendix for a detailed description of these integration patterns.

Use Case	Data Source › Target	Type			Timing		Key Patterns to Consider
		UI Integration	Services Layer Integration	Data Layer Integration	Sync	Async	
Account Creates/ Update	Salesforce › SAP		✓		✓		Remote Process Invocation – Request and Reply
Account Creates/ Update	Salesforce › SAP		✓			✓	Remote Process Invocation – Fire and Forget
Account Creates/ Update	SAP › Salesforce			✓		✓	Batch Data Synchronization
Account Creates/ Update	SAP › Salesforce		✓		✓		Remote Call-In
Initial Account Load	SAP › Salesforce			✓		✓	Batch Data Synchronization
Product & Price Books	SAP › Salesforce			✓		✓	Batch Data Synchronization
Order Creation	Salesforce › SAP		✓		✓		Remote Process Invocation – Request and Reply If order creation is a long-running transaction, consider using continuations or using Remote Process Invocation – Fire and Forget followed by a Remote Call-In to update Salesforce with order details such as Order ID and Status.
Order History & Financial Data (no reporting requirements in Salesforce)	SAP › Salesforce			✓	✓		Data Mashup
Order History & Financial Data (no reporting requirements in Salesforce)	SAP › Salesforce	✓			✓		UI Mashup
Order History & Financial Data (no reporting requirements in Salesforce)	SAP › Salesforce			✓		✓	Batch Data Synchronization
Order Status Updates	SAP › Salesforce		✓		✓		Remote Process Invocation – Request and Reply
Order Status Updates	SAP › Salesforce		✓			✓	Batch Data Sync depending on frequency requirements (e.g., frequent updates vs. nightly batch updates).

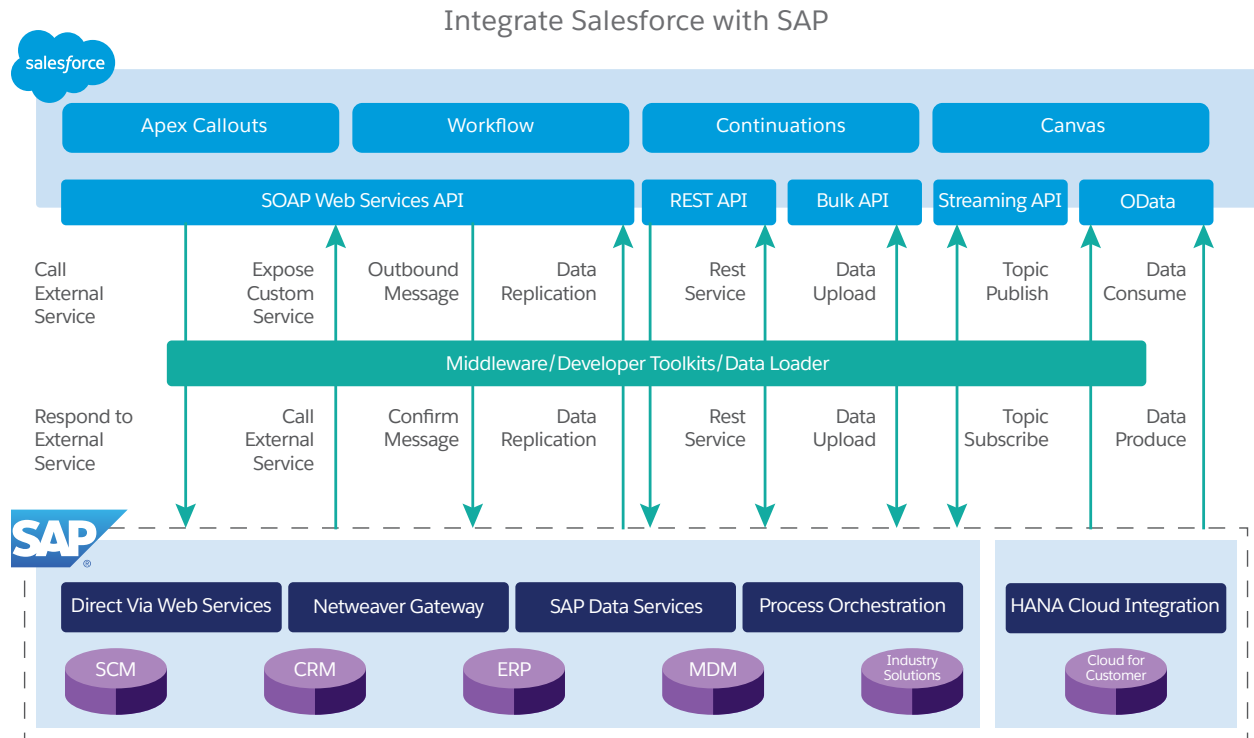
## Integration Pattern Decision Matrix – *continued*

Use Case	Data Source › Target	Type			Timing		Key Patterns to Consider
		UI Integration	Services Layer Integration	Data Layer Integration	Sync	Async	
Financial Data (Payment History, Credit Status, etc.)	SAP › Salesforce	✓			✓		UI Mashup
Financial Data (Payment History, Credit Status, etc.)	SAP › Salesforce		✓		✓		Remote Process Invocation – Request and Reply
Financial Data (Payment History, Credit Status, etc.)	SAP › Salesforce			✓	✓		Data Mashup
Invoices	SAP › Salesforce	✓			✓		UI Mashup
Invoices	SAP › Salesforce			✓	✓		Data Mashup
Shipping Information	SAP › Salesforce		✓		✓		Remote Call-In
Shipping Information	SAP › Salesforce		✓			✓	Batch Data Sync depending on frequency requirements (e.g., frequent updates vs. nightly batch updates).
Inventory	SAP › Salesforce		✓		✓		Remote Process Invocation – Request and Reply
Inventory	SAP › Salesforce			✓	✓		Data Mashup

CHAPTER 5

# Reference Architecture

The figure below provides a reference approach to integrate Salesforce with SAP.



## Summary

When it comes to supporting the requirements of your business through integration between Salesforce and SAP, consider the following to ensure a successful implementation:

- Understand the functional and nonfunctional requirements.
- Define your key integration considerations and criteria. Think about what your landscape looks like and what your enterprise architecture standards are.
- Understand the key integration options that are supported by Salesforce and SAP.
- Select an integration approach that aligns well with your enterprise.
- Based on all of the above, select the best integration patterns for your business requirements.
- Define which key performance indicators (KPIs) your business needs to monitor to ensure a successful implementation.
- Monitor and ensure the business requirements and KPIs are being met.



## Need more information?



Complete Salesforce integration reference

[DOWNLOAD PDF](#)



Salesforce integration for developers

[LEARN MORE](#)



Step-by-step basic integration tutorial

[LEARN MORE](#)



Salesforce Connect tutorial

[LEARN MORE](#)



Customer Success Stories – Unlock your SAP back office with real-time integration

[WATCH VIDEO](#)

## How can Salesforce help?



[CONTACT US](#)

Salesforce offers a number of Products and Services to help customers, from the smallest to the largest enterprise, successfully plan and implement integrations with enterprise systems. If you are interested in any of these offerings, please speak to your customer success management team.

## About the Authors



### **SIDDHARTH MUZUMDAR**

Senior Director

Sid is the global subject matter expert (SME) on Integration Architecture and works with customers on defining their integration strategy globally.



### **SUCHIN RENGAN**

Director

Suchin is with the advisory services group and works with large enterprise customers in driving the App Cloud strategy and adoption.



### **RALF SCHUNDELMEIER**

Director of Product Management

Ralf is with App Cloud at Salesforce and is responsible for Enterprise Application Integration and Data Import products.



### **SALEEM RAUF**

Senior Program Architect

Saleem is with the advisory services group and works with largest, most complex enterprise customers as a strategic advisor driving enterprise transformation initiatives for our customers.



### **SIBASISHA PADHY**

Program Architect

Sibasisha works with the advisory services group and helps enterprise customers with transformational initiatives.

## APPENDIX

## Salesforce Integration Capabilities

Capability	What it's for?	When to use it?	Protocol	Data Format	Communication
<b>REST API</b>	Accessing objects in your organization using REST.	You want to leverage the REST architecture to integrate with your organization. No WSDL requirement. Well-suited for browser-based applications, mobile apps, and highly interactive social applications.	REST	JSON, XML	Synchronous
<b>SOAP API</b>	Integrating your organization's data with other applications using SOAP.	You have pre-existing middleware services that need to work with WSDLs and XML data.	SOAP/WSDL	XML	Synchronous
<b>Bulk API</b>	Loading or deleting large numbers of records.	You have over a million records to process and speed is a requirement.	REST	CSV, XML	Asynchronous
<b>Streaming API</b>	Providing a stream of data reflecting data changes in your organization.	You need near real-time notifications of when records are created or updated.	Bayeux	JSON	Asynchronous (stream of data)
<b>Apex REST API</b>	Building your own REST API in Apex. Exposes Apex classes as RESTful Web services.	You need to build custom JSON responses or you want to expose custom functionality that you implemented in Apex.	REST	JSON, XML, Custom	Synchronous
<b>Apex SOAP API</b>	Creating custom SOAP Web services in Apex. Exposes Apex classes as SOAP Web services.	You need to build custom XML responses or you want to expose custom functionality that you implemented in Apex.	SOAP/WSDL	XML	Synchronous
<b>Salesforce Connect</b>	For high-velocity data residing on external source so the data is never stale.	You need to display real-time data in Salesforce from external systems (e.g., Display order history for an account from an ERP system in Salesforce).	ODATA	JSON	Synchronous
<b>Apex Callouts</b>	Integrating data stored in other applications by calling external Web services.	You need to display data stored in external systems or update external systems based on actions in Salesforce.	SOAP/WSDL/ REST	XML/JSON	Synchronous/ Asynchronous
<b>Workflow Outbound Messaging</b>	Configuration-driven approach to update other applications when a change is made in Salesforce.	You need to update an external system when changes are made to objects in Salesforce but do not want to write custom code.	SOAP/WSDL	XML	Asynchronous
<b>Canvas</b>	Enables you to easily integrate a third-party application in Salesforce.	You have existing Web apps that you would like to integrate with Salesforce but do not want to rewrite them.	HTTPS	HTML	Synchronous
<b>Visualforce</b>	Enables you to build custom user interfaces to satisfy your business requirements.	You need to display data stored in external systems side by side with Salesforce data without bringing it into Salesforce or have the need to implement customized user interfaces.	HTTPS	HTML	Synchronous

## APPENDIX

# Integration Patterns

Patterns are a proven way to capture expert knowledge in fields where there are no simple “one size fits all” answers, such as application architecture, object-oriented design, or message-oriented integration. Each pattern poses a specific design problem, discusses the considerations surrounding the problem, and presents an elegant solution that balances the various forces or drivers.

Pattern	Scenario
<b>Remote Process Invocation – Request and Reply</b>	Salesforce invokes a process on a remote system, waits for completion of that process, and then tracks state based on the response from the remote system.
<b>Remote Process Invocation – Fire and Forget</b>	Salesforce invokes a process in a remote system but doesn't wait for completion of the process. Instead, the remote process receives and acknowledges the request and then hands off control back to Salesforce.
<b>Batch Data Synchronization</b>	Data stored in Force.com should be created or refreshed to reflect updates from an external system, and when changes from Force.com should be sent to an external system. Updates in either direction are done in a batch manner.
<b>Remote Call-In</b>	Data stored in Force.com is created, retrieved, updated, or deleted by a remote system.
<b>UI Update Based on Data Changes</b>	The Salesforce user interface must be automatically updated as a result of changes to Salesforce data.
<b>UI Mashup</b>	The Salesforce user interface displays existing Web applications as a mashup on the Salesforce platform.
<b>Data Mashup</b>	The Salesforce user interface displays data stored in an external system by using external objects without having to physically move the data into the Salesforce platform.

# Glossary

Term	Description	Source/Resource Link
ALE	Application Link Enabling ensures a distributed but integrated SAP installation. It comprises the controlled exchange of business messages and consistent data management in loosely-linked SAP application systems.	<a href="http://help.sap.com/saphelp_nw70/helpdata/en/b3/4bf60b-4bab1343a03c004d2cb3566d/content.htm">http://help.sap.com/saphelp_nw70/helpdata/en/b3/4bf60b-4bab1343a03c004d2cb3566d/content.htm</a>
Apex	Apex is a strongly typed, object-oriented programming language that allows developers to execute flow and transaction control statements on the Force.com platform server in conjunction with calls to the Force.com API. Using syntax that looks like Java and acts like database stored procedures, Apex enables developers to add business logic to most system events, including button clicks, related record updates, and Visualforce pages. Apex code can be initiated by Web service requests and from triggers on objects.	<a href="https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_intro_what_is_apex.htm">https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_intro_what_is_apex.htm</a>
API	Application Programming Interface	<a href="https://en.wikipedia.org/wiki/Application_programming_interface">https://en.wikipedia.org/wiki/Application_programming_interface</a>
Async	Asynchronous API/Service Calls - This type of service execution may require a significant amount of time to process a request. However, the client that invoked the API Call/Web service can continue with other processing in the meantime rather than wait for the response. Asynchronous operations are extremely useful for environments in which a service can take a long time to process a client request.	
B2B	Business-to-business	<a href="https://en.wikipedia.org/wiki/Business-to-business">https://en.wikipedia.org/wiki/Business-to-business</a>
BAPI	A Business Application Programming Interface is a precisely defined interface providing access to processes and data in business application systems, such as R/3.	<a href="http://help.sap.com/saphelp_46c/helpdata/en/a5/3ec8464ac-011d1894e0000e829fbbd/content.htm">http://help.sap.com/saphelp_46c/helpdata/en/a5/3ec8464ac-011d1894e0000e829fbbd/content.htm</a>
CPQ	Configure Price Quote	
CRM	Customer Relationship Management is an approach to managing a company's interaction with current and future customers. The CRM approach tries to analyze data about customers' history with a company, in order to better improve business relationships with customers, specifically focusing on retaining customers, in order to drive sales growth.	<a href="https://en.wikipedia.org/wiki/Customer_relationship_management">https://en.wikipedia.org/wiki/Customer_relationship_management</a>
EDI	Electronic Data Interchange is an electronic communication method that provides standards for exchanging data via any electronic means. By adhering to the same standard, two different companies or organizations, even in two different countries, can electronically exchange documents, such as purchase orders, invoices, shipping notices, and many others.	<a href="https://en.wikipedia.org/wiki/Electronic_data_interchange">https://en.wikipedia.org/wiki/Electronic_data_interchange</a>
ERP	Enterprise Resource Planning is a category of business-management software – typically a suite of integrated applications that an organization can use to collect, store, manage and interpret data from many business activities, including: product planning, cost, manufacturing, or service delivery, marketing, and sales.	<a href="https://en.wikipedia.org/wiki/Enterprise_resource_planning">https://en.wikipedia.org/wiki/Enterprise_resource_planning</a>

# Glossary

Term	Description	Source/Resource Link
Force.com Canvas	The Force.com Canvas SDK is an open-source suite of JavaScript libraries and provides simple methods that use existing Salesforce APIs (SOAP API, REST API, Chatter REST API) so you can build a seamless end-user experience inside of salesforce.com.	<a href="https://developer.salesforce.com/page/Force.com_Canvas">https://developer.salesforce.com/page/Force.com_Canvas</a>
HTML	HyperText Markup Language, commonly referred to as HTML, is the standard markup language used to create web pages.	<a href="https://en.wikipedia.org/wiki/HTML">https://en.wikipedia.org/wiki/HTML</a>
HTTP	The Hypertext Transfer Protocol is an application protocol for distributed, collaborative, hypermedia information systems. HTTP is the foundation of data communication for the World Wide Web. Hypertext is structured text that uses logical links (hyperlinks) between nodes containing text.	<a href="https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol">https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol</a>
HTTPRequest / HTTPResponse	HTTP functions as a request-response protocol in the client-server computing model. A web browser, for example, may be the client and an application running on a computer hosting a web site may be the server. The client submits an HTTP request message to the server. The server, which provides resources, such as HTML files and other content, or performs other functions on behalf of the client, returns a response message to the client. The response contains completion status information about the request and may also contain requested content in its message body.	<a href="https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol">https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol</a>
IDOC	Intermediate Document (SAP)	<a href="http://scn.sap.com/docs/DOC-26192?rid=/webcontent/uuid/0082e169-0c37-2b10-49ac-ab422af20c9a">http://scn.sap.com/docs/DOC-26192?rid=/webcontent/uuid/0082e169-0c37-2b10-49ac-ab422af20c9a</a>
IPASS	iPaaS is a suite of cloud services enabling customers to develop, execute and govern integration flows. Under the cloud-based iPaaS integration model, customers drive the development and deployment of integrations without installing or managing any hardware or middleware.	<a href="http://www.gartner.com/it-glossary/information-platform-as-a-service-ipaas">http://www.gartner.com/it-glossary/information-platform-as-a-service-ipaas</a>
JAVA	Java is a general-purpose computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible.	<a href="https://en.wikipedia.org/wiki/Java_(programming_language)">https://en.wikipedia.org/wiki/Java_(programming_language)</a>
JSON	JavaScript Object Notation	<a href="http://www.json.org/">http://www.json.org/</a>
KPI	Key Performance Indicators	<a href="https://en.wikipedia.org/wiki/Performance_indicator">https://en.wikipedia.org/wiki/Performance_indicator</a>
Lightning Components	The Lightning Component framework is a UI framework for developing dynamic web apps for mobile and desktop devices. It's a modern framework for building single-page applications engineered for growth. The framework supports partitioned multi-tier component development that bridges the client and server. It uses JavaScript on the client side and Apex on the server side.	<a href="https://developer.salesforce.com/docs/atlas.en-us.lightning.meta/lightning/">https://developer.salesforce.com/docs/atlas.en-us.lightning.meta/lightning/</a>
MDM	Master Data Management	
Middleware	Middleware is the software that connects software components or enterprise applications. Middleware is the software layer that lies between the operating system and the applications on each side of a distributed computer network. Typically, it supports complex, distributed business software applications.	<a href="https://en.wikipedia.org/wiki/Middleware">https://en.wikipedia.org/wiki/Middleware</a>

# Glossary

Term	Description	Source/Resource Link
<b>MVC</b>	Model-view-controller is a software architectural pattern mostly (but not exclusively) for implementing user interfaces. It divides a given software application into three interconnected parts, so as to separate internal representations of information from the ways that information is presented to or accepted from the user	<a href="https://en.wikipedia.org/wiki/Model-view-controller">https://en.wikipedia.org/wiki/Model-view-controller</a>
<b>.NET</b>	.NET Framework (pronounced dot net) is a software framework developed by Microsoft that runs primarily on Microsoft Windows.	<a href="https://en.wikipedia.org/wiki/.NET_Framework">https://en.wikipedia.org/wiki/.NET_Framework</a>
<b>OData</b>	Open Data Protocol	<a href="https://en.wikipedia.org/wiki/Open_Data_Protocol">https://en.wikipedia.org/wiki/Open_Data_Protocol</a>
<b>PHP</b>	PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language.	<a href="https://en.wikipedia.org/wiki/PHP">https://en.wikipedia.org/wiki/PHP</a>
<b>REST</b>	In computing, representational state transfer (REST) is the software architectural style of the World Wide Web.REST's coordinated set of constraints, applied to the design of components in a distributed hypermedia system, can lead to a higher-performing and more maintainable software architecture.	<a href="https://en.wikipedia.org/wiki/Representational_state_transfer">https://en.wikipedia.org/wiki/Representational_state_transfer</a>
<b>RFC</b>	Remote Function Call (RFC) is the standard SAP interface for communication between SAP systems.	<a href="https://en.wikipedia.org/wiki/Remote_Function_Call">https://en.wikipedia.org/wiki/Remote_Function_Call</a>
<b>Ruby on Rails</b>	Ruby on Rails, or simply Rails, is a web application framework written in Ruby under MIT License.	<a href="https://en.wikipedia.org/wiki/Ruby_on_Rails">https://en.wikipedia.org/wiki/Ruby_on_Rails</a>
<b>S&amp;OP</b>	Sales and Operations Planning	
<b>SAP JCO</b>	The SAP Java Connector (SAP JCo) is a middleware component that enables the development of SAP-compatible components and applications in Java. SAP JCo supports communication with the SAP Server in both directions:inbound (Java calls ABAP) and outbound calls (ABAP calls Java).	<a href="https://help.sap.com/saphelp_nwpi711/helpdata/en/48/70792c-872c1b5ae10000000a42189c/content.htm">https://help.sap.com/saphelp_nwpi711/helpdata/en/48/70792c-872c1b5ae10000000a42189c/content.htm</a>
<b>SAP Netweaver Gateway</b>	SAP NetWeaver Gateway is a technology that provides a simple way to connect devices, environments and platforms to SAP software based on market standards.	<a href="http://scn.sap.com/docs/DOC-35472">http://scn.sap.com/docs/DOC-35472</a>
<b>SAP Netweaver PI</b>	SAP NetWeaver Process Integration facilitates the integration of business processes that span different departments, organizations, or companies. Think of an application component being part of the value chain of a business application or a business process.	<a href="https://help.sap.com/saphelp_nw73ehp1/helpdata/en/ac/5ea-c92a34e45a2831ee59e9bce0f8d/frameset.htm">https://help.sap.com/saphelp_nw73ehp1/helpdata/en/ac/5ea-c92a34e45a2831ee59e9bce0f8d/frameset.htm</a>
<b>SAP Netweaver PO</b>	Process Orchestration provides a tool infrastructure to model and design business processes - from the overall process flow and the sequence of activities up to the interfaces and data types that are needed to integrate SAP systems and external systems.	<a href="https://help.sap.com/saphelp_nw73ehp1/helpdata/en/b5/0a32b8e-f054513a3f7ca52f4c8a7d5/frameset.htm">https://help.sap.com/saphelp_nw73ehp1/helpdata/en/b5/0a32b8e-f054513a3f7ca52f4c8a7d5/frameset.htm</a>
<b>SCM</b>	Supply chain management (SCM) is the management of the flow of goods and services. It includes the movement and storage of raw materials, work-in-process inventory, and finished goods from point of origin to point of consumption.	<a href="https://en.wikipedia.org/wiki/Supply_chain_management">https://en.wikipedia.org/wiki/Supply_chain_management</a>

# Glossary

Term	Description	Source/Resource Link
SOA	Service-Oriented Architecture is an architectural style that supports service-orientation. Service-orientation is a way of thinking in terms of services and service-based development and the outcomes of services"	<a href="https://en.wikipedia.org/wiki/Service-oriented_architecture">https://en.wikipedia.org/wiki/Service-oriented_architecture</a>
SOAP	Simple Object Access Protocol is a protocol specification for exchanging structured information in the implementation of web services in computer networks.	<a href="https://en.wikipedia.org/wiki/SOAP">https://en.wikipedia.org/wiki/SOAP</a>
SOQL	Salesforce Object Query Language	<a href="https://developer.salesforce.com/docs/atlas.en-us.soql_sosl.meta/soql_sosl/">https://developer.salesforce.com/docs/atlas.en-us.soql_sosl.meta/soql_sosl/</a>
SOSL	Salesforce Object Search Language	<a href="https://developer.salesforce.com/docs/atlas.en-us.soql_sosl.meta/soql_sosl/sforce_api_calls_sosl.htm">https://developer.salesforce.com/docs/atlas.en-us.soql_sosl.meta/soql_sosl/sforce_api_calls_sosl.htm</a>
Sync	Synchronous API/Service Calls - This type of service execution provides an immediate response to a query. In this situation, the client will wait until the server sends back the response message. The advantage of using the synchronous service is that the client application knows the status of the Web service operation in a very short time.	
UI	User Interface	<a href="https://en.wikipedia.org/wiki/User_interface">https://en.wikipedia.org/wiki/User_interface</a>
Visualforce	Visualforce is the component-based user interface framework for the Force.com platform. The framework includes a tag-based markup language, similar to HTML. Each Visualforce tag corresponds to a coarse or fine-grained user interface component, such as a section of a page, or a field.	<a href="https://developer.salesforce.com/page/An_Introduction_to_Visualforce">https://developer.salesforce.com/page/An_Introduction_to_Visualforce</a>
Web Dynpro	Web Dynpro ABAP is the SAP standard UI technology for developing Web applications in the ABAP environment. It consists of a runtime environment and a graphical development environment with special Web Dynpro tools that are integrated in the ABAP Workbench	<a href="http://help.sap.com/saphelp_nw70/helpdata/en/f6/501b-42b5815133e10000000a155106/frameset.htm">http://help.sap.com/saphelp_nw70/helpdata/en/f6/501b-42b5815133e10000000a155106/frameset.htm</a>
WSDL	WSDL is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information. The operations and messages are described abstractly, and then bound to a concrete network protocol and message format to define an endpoint.	<a href="https://www.w3.org/TR/wsdl">https://www.w3.org/TR/wsdl</a>
XML	Extensible Markup Language	<a href="https://en.wikipedia.org/wiki/XML">https://en.wikipedia.org/wiki/XML</a>

What can Salesforce App Cloud do for you?  
Talk to an expert, call us at 1-844-463-0828.

[LEARN MORE](#)