MuleSoft

WHITEPAPER

Top 5 Salesforce integration patterns

Best practices for integrating Salesforce

Table of contents

Introduction

We are in the midst of the fourth industrial revolution. Ninety-seven percent of businesses have joined the race to deliver connected customer experiences. Delivering these experiences requires more than just providing products and services in a timely manner. It requires creating organic connections with real people at every touchpoint of their journey.

As the world's No. 1 customer relationship management (CRM) platform, Salesforce enables companies to seamlessly connect customer experiences across every touchpoint in the business: sales, service, marketing, and more.

The challenge many businesses face is how to connect Salesforce with their core systems and applications such as databases, enterprise resource planning (ERP) systems, and custom applications. Though every integration scenario is unique, there are several common requirements and issues that developers must resolve. This whitepaper describes the strategies, in the form of patterns, that many developers use to integrate with Salesforce.

Top 5 Salesforce integration patterns

Integration plays a critical role in creating seamless customer experiences across channels, departments, and systems. To speed up Salesforce integration, organizations typically apply integration applications based on five patterns. These patterns are the most logical sequence of steps to solve specific types of integration problems, documented from real-world use cases.

An integration application includes a pattern and a business use case. The pattern is the use of the generic process for data movement and handling. The business use case is comprised of the value obtained from an integration.

When thinking about the format of a simple point-to-point, atomic integration one can use the following structure:

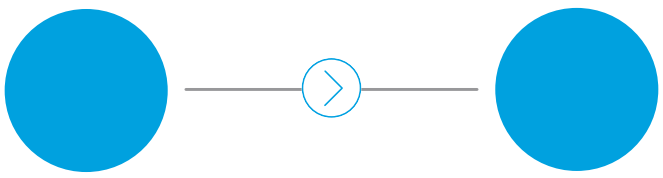
Application(s) A to application(s) B – object(s) – pattern

To templatize common Salesforce integration needs, base patterns, or best practices, it must first be established to make integrations atomic, reusable, extensible, and easily understandable. The essence of a pattern includes some combination of at least one of the following elements:

- **A source system** where data resides prior to execution.
- **The criteria** which determines the scope of data to be copied, moved, or replicated.
- **Transformation** from one format to another.
- **A destination system** where the data will be inserted.
- **Results capture** which compares the final state with the desired state.

The five most common integration patterns are migration, broadcast, aggregation, correlation, and bidirectional synchronization.

Pattern No. 1: migration

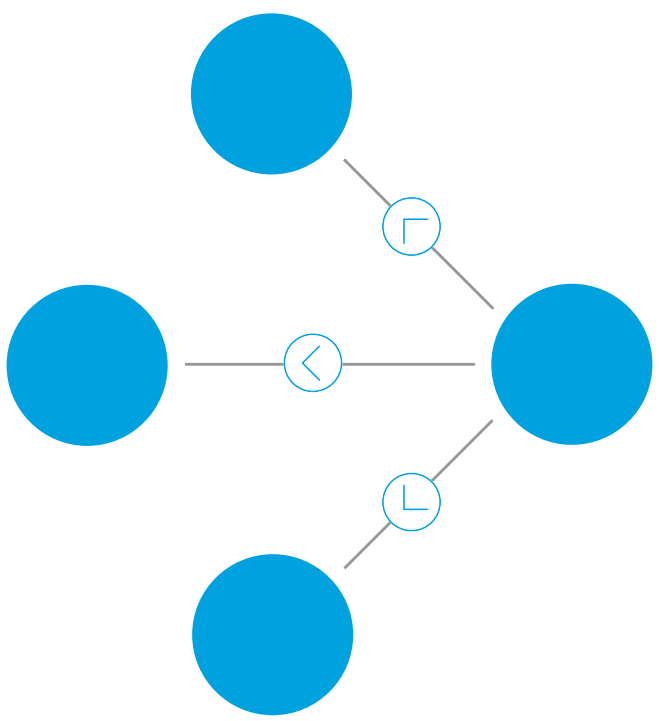


Data migration is the act of moving a specific set of data at a point in time from one system to another. A migration pattern allows developers to create automated integration services for functionality that will be shared across multiple teams. Developers can set configuration parameters to pass into an API call so that the migration application can dynamically migrate scoped Salesforce data in or out of Salesforce either on command or in the form of an API. Creating reusable services for frequent migrations can save an exceptional amount of time for development and operations teams.

There are numerous Salesforce integration scenarios that may require a migration pattern including, migrating data from a legacy CRM system to Salesforce or from one Salesforce organization to another, backing up a master data set, consolidating CRM systems, populating Salesforce product data from SAP, and more. Migration patterns are tuned to handle large volumes of data, process many records in batches, and have to fail gracefully in case of error(s).

Migrations are essential to any data system and are executed extensively in any organization that has data operations. Migrations keep data agnostic from the tools used to create it, view it, and manage it and ensure that data is not lost when tools are changed.

Pattern No. 2: broadcast



Broadcast is the act of moving data from a single source system to many destination systems in real time, near real time, or on an ongoing basis. Essentially, it is one-way synchronization from one to many. Typically “one-way sync” implies a one-to-one relationship; however, the broadcast pattern can also be a one-to-many relationship.

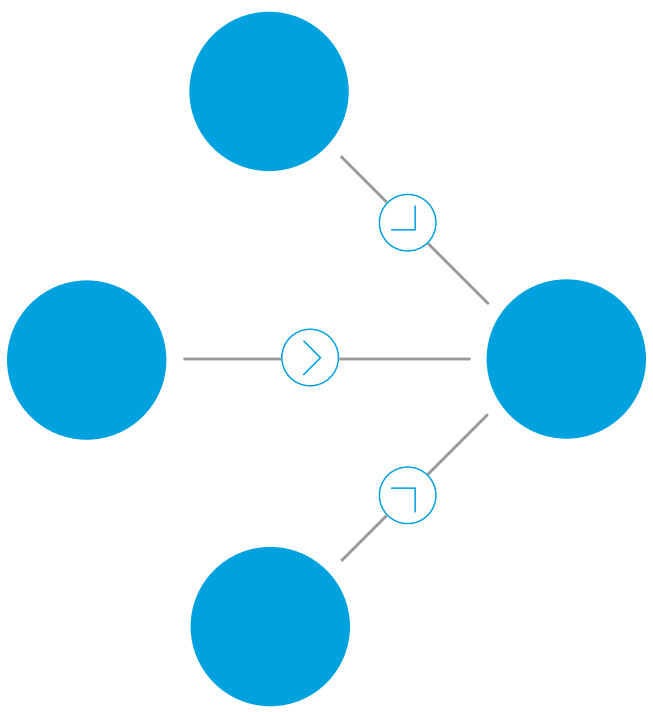
In contrast to the migration pattern, the broadcast pattern is transactional. It executes logic only for items that have been recently modified and is optimized for processing records as quickly as possible. Broadcast patterns are highly flexible and used to keep data up to date between multiple systems, across time. A broadcast pattern should be highly reliable to avoid losing critical data in transit. Reliability is also necessary to employ integrations with low human oversight across mission-critical applications as broadcast patterns are usually initiated by a push notification or a scheduled job.

The broadcast pattern allows for the immediate transfer of customer data between systems, whether from two instances of Salesforce or between Salesforce and other systems. For instance, the pattern can broadcast update to a customer

record between Salesforce organizations or can enable an action in Salesforce to immediately translate into order fulfillment processing. Examples of use cases for the broadcast pattern include the following:

- A sales order in SAP should be created when an opportunity is marked as CLOSED WON in Salesforce.
- Real-time data needs to be synchronized from Siebel to Salesforce.
- Insert/update to a Salesforce record must be reflected in other adjacent enterprise applications.

Pattern No. 3: aggregation



Aggregation is the simplest way to extract and process data from multiple systems into one application or report in real time. The alternative to aggregation is to run multiple migrations on a daily basis, which requires manual maintenance to keep data accurate, synchronized, and up to date. Broadcasting data from multiple systems, though real time, does require maintaining a database to store replicated data to be queried. By using an integration template built on an aggregation pattern, developers can easily query multiple systems on demand and merge data sets to use data however and whenever needed. This way, one can create or store reports in .csv or other formats of choice.

Examples of uses for the aggregation pattern include the following:

- Creating a dashboard that pulls data from multiple Salesforce instances, while ensuring data consistency.
- Updating Salesforce with data from multiple back-end systems such as ERP and issue-tracking systems.
- Building APIs that collect and return data from multiple systems, or report across multiple systems.

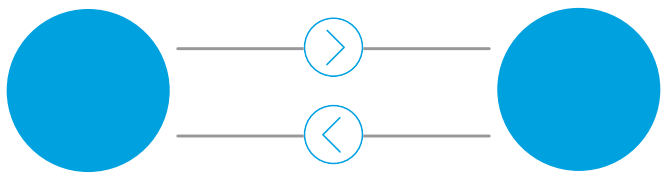
Since the aggregation pattern allows developers to extract and process data from multiple systems and merge them into one application, data is always up to date, does not get replicated, and can be processed or merged to produce any desired data set or report. This avoids the need to have a separate operational database for merged content and makes reports available in any format or within any repository.

The aggregation pattern is particularly helpful in the following scenarios:

- When creating orchestration APIs that get data from multiple systems to "modernize" legacy systems by processing data into one response.
- When creating reports or dashboards which have to pull data from multiple systems and create an experience with that data.
- When systems used for compliance or auditing purposes need to have related data from multiple systems. With the aggregation pattern, compliance data can be collected from multiple systems, but be housed in a central, compliant repository.

When using the aggregation pattern to integrate data and applications, key considerations include collecting data, the scope of the source data and insert data, merging multiple data sets, formatting data, and any additional destinations. For example, when collecting data, there are two ways to do so: either create an application - that listens for messages from multiple systems and aggregates them in real time, or create an application - that is triggered by an event. When combining multiple data sets, it is important to consider how to merge them and how to present the data in the final report or destination system.

Pattern No. 4: bidirectional sync



Bidirectional sync is the act of uniting two or more data sets from two or more different systems to behave as one system that recognizes the existence of different data sets. This type of integration is useful when different tools or different systems, which are needed in their own right and for their own specific purposes, must accomplish different functions on the same data set. Using bidirectional sync to share the data set enables the use of both systems, while maintaining a consistent real-time view of the data across systems.

Enterprises can use bidirectional sync integration to:

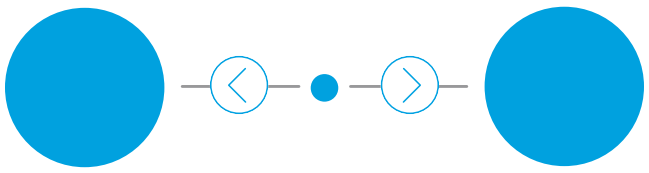
- Optimize organizational processes.
- Closely align data representations to reality in both systems.
- Reduce the compound cost of manually addressing inconsistencies, lack of data, or the impact on business as a result of inconsistencies.
- Hand pick and customize a suite of tools with best-of-breed applications used collectively for specific organizational aims.

Applied to Salesforce, bidirectional sync supports examples such as integration of multiple systems that contribute to operational efficiencies and a streamlined quote-to-cash flow. Salesforce can play the role of the primary system of records or a secondary system where all data are synchronized to.

Bidirectional sync integration enables each system to perform optimally while maintaining data integrity across both synchronized systems. It can provide flexibility to modularly add and remove two or more systems that subspecialize inside a domain as storage. And sync is particularly advantageous when object representations of reality must be comprehensive and consistent.

Bidirectional sync is a pattern that can unify the most relevant and best performing applications to enhance Salesforce functionalities and result in overall growth in sales.

Pattern No. 5: correlation



Correlation and bidirectional sync are very similar but the patterns have one critical difference. Whereas bidirectional synchronization aims to replicate the same data elements in two locations, correlation is used to associate disparate data records without copying the data itself. Bidirectional synchronization will create new records if they are found in one system and not the other. The correlation pattern is not discerning in terms of origination of objects. It will agnostically synchronize objects as long as they are found in both systems.

Correlation is useful for cases in which two groups or systems only want to share data, but only if they both have records representing the same items or contacts in reality. The correlation pattern is most useful when extra data is more costly than beneficial as it scopes out the “unnecessary” data. For example, hospitals in the same health care network may want to correlate patient data for shared patients across hospitals, but it would be a privacy violation to share patient data with a hospital that has never admitted or treated the patient.

With the correlation pattern, the most important consideration is the definition of the term “same” across records. This definition can vary by industry and consequences for unclear definitions also vary. For example, in targeting offers to customers, the same name may be close enough; however, in a hospital, relying on a name could have serious consequences if

two patients have the same name and different courses of treatment. The table below illustrates what can occur when the definition of “same” is too strict, too lax, or accurate across correlation and bidirectional sync.

	Just right	Too strict	Too loose
Bidirectional sync	Union of data sets	Duplicate created	Wrong records Merged
Correlation	Intersection of data sets	Not synchronized	Wrong records Merged

The correlation pattern allows shared account data to be synchronized across applications, including Salesforce instances, either across an organization or between a company and a partner. Working internally, it can also allow for synchronization of customer data entered by two different teams or even team members of the same department.

The Salesforce integration patterns we just reviewed cover five of the most common integration scenarios. However, there are many more integration patterns preferred for other use cases — for example, data virtualization is a synchronous example of aggregation whereby disparate back end data sources are composed and exposed through a single API to appear in Salesforce as if they are native objects. The next section details where you can find hundreds of templates, connectors, and APIs for Salesforce integrations.

Get started: Anypoint Templates for Salesforce integration

Each integration pattern follows a consistent structure. Templates are packaged integration patterns that address the most common use cases — helping developers integrate faster to save time.

MuleSoft's Anypoint Templates for Salesforce integration are integration applications built to be configured, customized, extended, and reused. Anypoint Templates are:

- **Complete for atomic use cases:** They are comprehensive and focused on the main base unit of value, but are compounded by adding flows in parallel or serial order.
- **Reusable:** Anypoint Templates conform the base-to-base patterns leveraged in many variations of the same base problem.
- **Extensible:** Anypoint Templates are designed to grow, containing limited field mappings, data scopes, insert statements, definitions of "same," and transformations to adhere to specifics within each enterprise.
- **High-quality:** They are built and tested with production quality in mind.
- **Elegant:** Flows are built to read like an integration story so that they are easily understood.
- **Documented:** Clear, complete documentation developers helps to quickly understand and initiate templates.
- **Easy-to-find and use:** Search and browse templates for hundreds of use cases in Anypoint Exchange and import directly into Anypoint Studio, the design environment of the Anypoint Platform.

MuleSoft's Anypoint Templates for Salesforce integration streamline and simplify the process of:

- Merging data after acquiring other entities.
- Moving off of a legacy CRM system.
- Updating Salesforce with ERP data in real time.
- Ensuring consistency of data across Salesforce instances and other enterprise applications.
- Creating APIs that pull data from multiple systems.

Anypoint Platform allows for automated updates of contacts, accounts, products, leads, opportunities, and users in Salesforce when changes occur or new information appears in ERP or CRM applications, databases, or other Salesforce instances. Integration templates enhance consistency of data and records in real time across applications, geographies, business units, departments, and more.

Better Together: MuleSoft + Salesforce

Over the years, MuleSoft has invested in understanding Salesforce with blueprints, connectors, and expertise to accelerate your connectivity. Today, MuleSoft and Salesforce are better together with a wide range of offerings to power your digital transformation such as:

- 100+ pre-built Salesforce connectors and assets, each built with proprietary knowledge of use cases for unmatched connectivity and reliability.
- First to market connectivity to new Salesforce products and API revisions as part of a unified product experience.
- One place for all teams to save, share, and discover assets in Salesforce through Anypoint Exchange.

With more support and a clear product vision, MuleSoft and Salesforce will increase developer productivity and allow you to build connected experiences faster.

About MuleSoft

MuleSoft, a Salesforce company

MuleSoft's mission is to help organizations change and innovate faster by making it easy to connect the world's applications, data, and devices. With its API-led approach to connectivity, MuleSoft's market-leading Anypoint Platform™ empowers over 1,600 organizations in approximately 60 countries to build application networks. By unlocking data across the enterprise with application networks, organizations can easily deliver new revenue channels, increase operational efficiency, and create differentiated customer experiences.

For more information, visit **mulesoft.com**