Order on Behalf of a Customer

Salesforce, Spring ’24

@salesforcedocs

Last updated: November 21, 2023

CONTENTS

ORDER ON BEHALF OF A CUSTOMER

Place orders on behalf of shoppers who need assistance because they can't access the Internet or find the products they want to purchase.

Get Started

	Explore system architecture related to this solution. • B2C Industry Blueprint • B2C Reference Architecture • B2C Solution Architectures
	Take Trailhead modules related to this solution. • Salesforce Solution Kits: Quick Look • Customer 360 Guide for Retail: Quick Look • Customer 360 Guides: Quick Look

This solution kit helps you:

- Increase completed purchases.
- Give your shoppers a personalized experience when you connect Commerce Cloud and Service Cloud.
- Provide exceptional customer service experiences.
- Increase shopper engagement.

Required Products

- Commerce Cloud (SFRA or SiteGenesis)
- Service Cloud

Implement This Solution

Order on Behalf Solution Workflow
Learn how data flows through the configurations to order on behalf of a customer.

Design Considerations

Keep these design considerations in mind when you order on behalf of a customer.

Integration Reference Implementation

Integration reference implementations are developer enablement frameworks that accelerate cross-cloud integration by providing code, configuration, and implementation patterns. Use the b2c-crm-sync integration reference implementation to order on behalf of a customer.

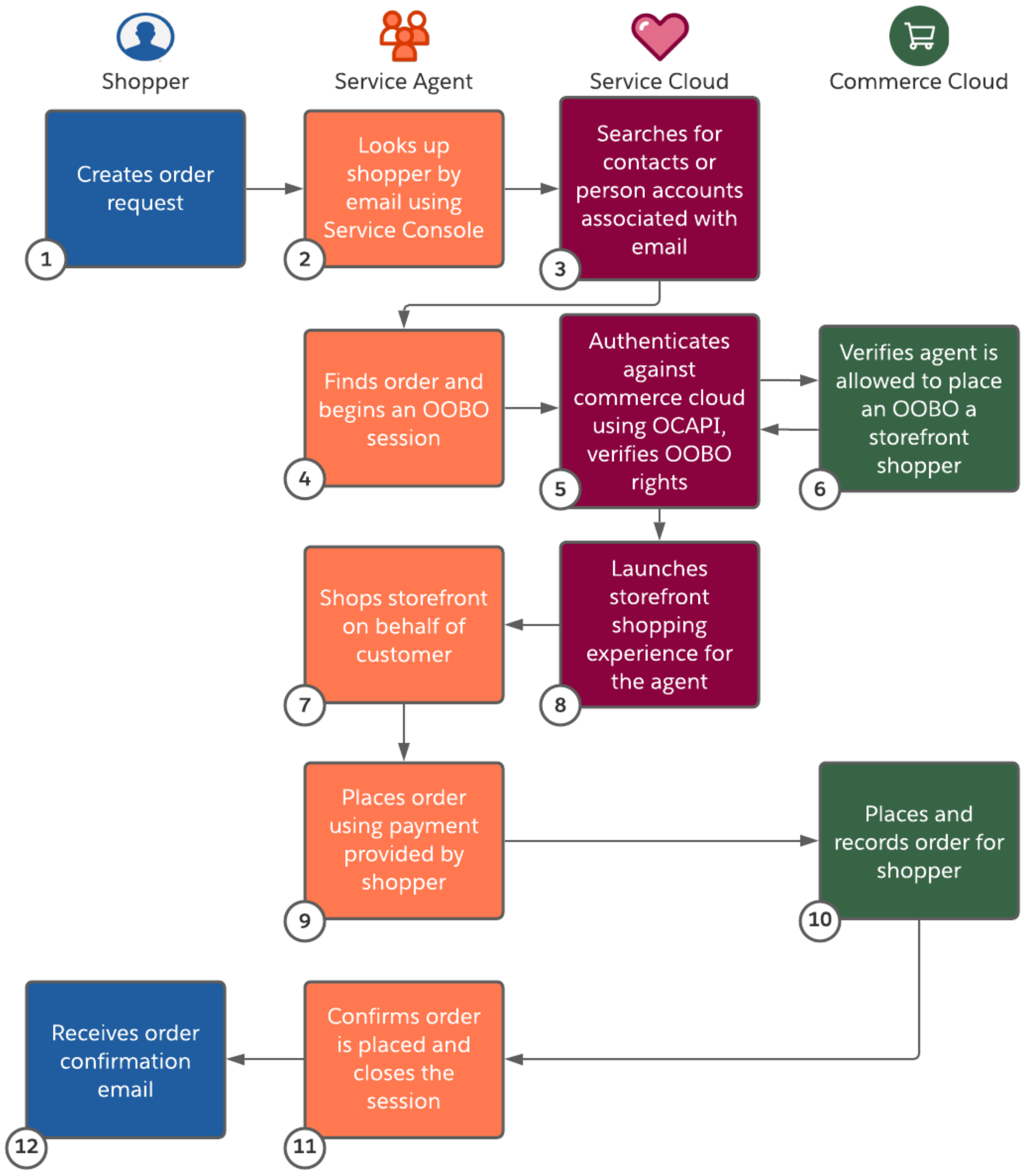
Configurations

Use these configurations to order on behalf of a customer.

Order on Behalf Solution Workflow

Learn how data flows through the configurations to order on behalf of a customer.

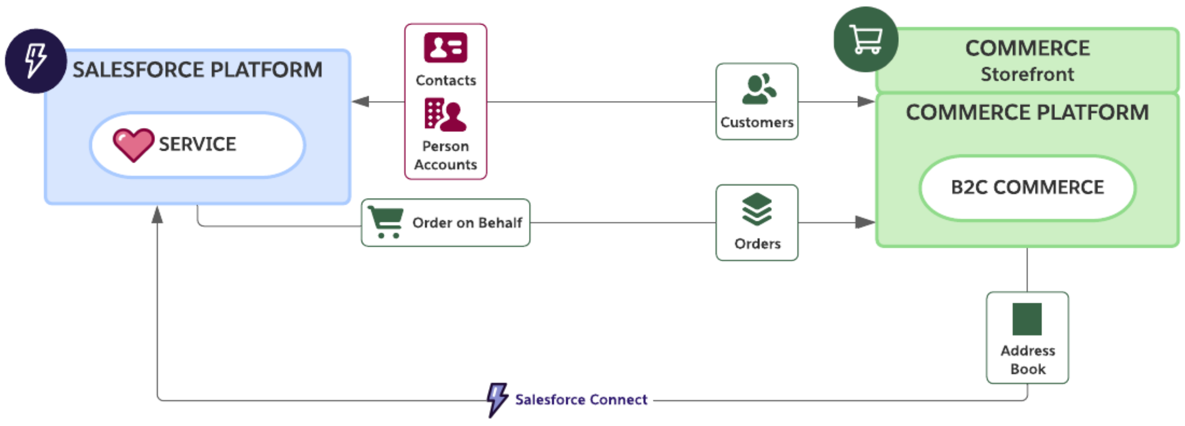
Workflow


Shopper
Service Agent
Service Cloud
Commerce Cloud
1 Creates order request
2 Looks up shopper by email using Service Console
3 Searches for contacts or person accounts associated with email
4 Finds order and begins an OOBO session
5 Authenticates against commerce cloud using OCAPI, verifies OOBO rights
6 Verifies agent is allowed to place an OOBO a storefront shopper
7 Shops storefront on behalf of customer
8 Launches storefront shopping experience for the agent
9 Places order using payment provided by shopper
10 Places and records order for shopper
11 Confirms order is placed and closes the session
12 Receives order confirmation email

1. The shopper contacts a service agent with an order request.
2. The agent looks up the shopper using their email address via the Service Console.
3. Service Cloud searches for any person accounts or contacts associated with the shopper's email address.
4. The agent initiates an Order on Behalf Of (OOBO) session via Lightning Quick Action.

5. Service Cloud authenticates against Commerce Cloud using the Open Commerce API verifying that the agent has OOBO rights.
6. Commerce Cloud verifies that the agent has the right to place an OOBO for a storefront shopper.
7. Service Cloud launches the storefront shopping experience for the agent with OOBO enabled.
8. The agent shops the storefront on behalf of the shopper.
9. The agent places the order using the payment information provided by the shopper.
10. The order is placed and recorded for the specific shopper.
11. The agent confirms the order is placed with the shopper and closes the OOBO session.
12. The shopper receives the order confirmation email generated for the OOBO placed order.

Understand the Flow of Data



Functionality Considerations

The b2c-crm-sync reference implementation:

- Supports both person accounts and account and contact customer models within the Salesforce Customer 360 platform.
- Supports synchronization of registered Salesforce B2C Commerce customer profiles between the Salesforce Customer 360 Platform and Salesforce B2C Commerce. The synchronization happens near real time via a scheduled B2C Commerce job.
- Supports OOBO shopping for customer service representatives configured and launched from within the Salesforce platform for both registered and anonymous shoppers.
- Uses Salesforce B2C Commerce open commerce REST APIs to interact with B2C Customer profiles.
- Supports multiple B2C sites and customer lists.

Implementation Considerations

The b2c-crm-sync reference implementation:

- Requires a B2C Commerce sandbox and Salesforce DevHub capable of creating scratch orgs.
- Requires a Salesforce Enterprise Edition configured with Salesforce Connect to take advantage of the federated B2C customer address book feature included with b2c-crm-sync.
- Uses node.js enabled CLI and Salesforce CLI commands. Salesforce CLI is one of the Salesforce DX (Developer Experience) tools.
- Works with existing B2C Commerce storefronts.

- Requires that Business Manager users representing customer service agents have B2C Commerce permissions to log in and place orders on behalf of registered storefront shoppers.
- Requires setup of B2C Commerce Access Keys as an alternative form of authentication when logging in to Business Manager via external applications. Applications include WebDAV File Access, UX Studio Agent, User Login, OCAPI, and Protected Storefront Access.
- Requires that Salesforce B2C Commerce environment's OCAPI data API permissions are enabled to support remote interactions.
- Requires extending the B2C Commerce instance's OCAPI permissions to allow the Salesforce org to create a storefront session for the OOBO shopping experience.
- Requires an integration user with administrative rights configured in Service Cloud.
- Enables functionality for agents from the customer details display in Service Cloud.
- Requires a Service Cloud connected app supporting OAuth authentication.
- Registers B2C Commerce as a remote site.
- Lets you launch from within Service Cloud. Include the Commerce Cloud customer number and internal customer ID in the person or standard account.

Related Content

Review this solution’s use case and purpose.

- Order on Behalf of a Customer on page 1

Take the next steps in this implementation.

- Design Considerations
- Integration Reference Implementation
- Configurations

Design Considerations

Keep these design considerations in mind when you order on behalf of a customer.

Storefront Shoppers

- When the agent launches the Order on Behalf capability, the Commerce Cloud Customer ID registers the session. The Service Cloud person account or contact uses the Commerce Cloud Customer ID to launch the Order on Behalf capability.
- B2c-crm-sync extends the B2C Commerce OOBO capability to anonymous storefront shoppers by creating B2C Commerce customer profiles. Service agents use these profiles to authenticate against B2C Commerce in anonymous shopping sessions.

Commerce Cloud Authentication

- The OOBO shopping experience requires that service agents in the Salesforce platform authenticate against the B2C Commerce environment before creating the shopping session. A named credential manages this authentication for each user via the Salesforce platform.
- Update access keys every 365 days.

Map Custom Fields

- Map only the fields for which you need data.
- Use B2C integration field mappings to map fields for contacts or person accounts.

- Use the b2c-crm-sync alternative page layouts for contacts and person accounts as a starting point. Consider how to customize your existing Salesforce customer layouts. These layouts expose the initial set of custom fields added to contacts and person accounts to support synchronization with B2C Commerce.
- Every REST call between Commerce Cloud and Service Cloud counts towards API governor limits.
- To request a governor limit increase, contact your Salesforce Account Executive.

Note: You can include order details in Service Cloud by integrating with the Salesforce Order Management System (OMS) or your own supported OMS.

Related Content

Review earlier steps in this solution.

- Order on Behalf Solution Workflow

Take the next steps in this implementation.

- Integration Reference Implementation
- Configurations

See Also

- B2C Custom Hooks Overview
- B2C Order.xsd Salesforce XML Schema
- B2C Sample order.xml

Integration Reference Implementation

Integration reference implementations are developer enablement frameworks that accelerate cross-cloud integration by providing code, configuration, and implementation patterns. Use the b2c-crm-sync integration reference implementation to order on behalf of a customer.

After you sign in to GitHub, the b2c-crm-sync solution designed by Salesforce Architects facilitates the integration between Salesforce B2C Commerce Cloud and Service Cloud. The reference implementation provides a framework to integrate the clouds by using public REST APIs to share and sync data.

Before implementing, sign into GitHub and download b2c-crm-sync. Follow the installation instructions available from the repository ReadMe.md file.

Are you a new Commerce Cloud customer or partner and don't have access to the GitHub repository? Get started with the Commerce API.

General Information About Reference Implementations

- Reference implementations are developer-enablement frameworks that accelerate cross-cloud integration by providing code, configuration, and implementation patterns.
- Reference implementations support a core set of use cases that you can extend to support other customer-driven use cases.
- require customization and configuration in Service Cloud and Commerce Cloud. The Commerce Cloud storefront requires customization as part of the integration.
- Implementation and validation require operational and administrative experience with Service Cloud.

- Plan your implementation as you would any other B2C Commerce Cloud feature by collecting requirements, capturing work tasks, and making task estimates.

What Your Company Can Do with This Reference Implementation

- Share views of customers and order data between both clouds
- Offer self-service through automated case creation using the storefront
- Facilitate agent-to-customer conversations that encourage cross-sell and up-sell opportunities
- Real-time peer-to-peer data synchronization for customer, order, and case data between Commerce Cloud and Service Cloud
- Source code that allows for storefront or Service Cloud customization based on service needs
- Faster integration time to market for both clouds when specifically targeting the supported use cases

Related Content

Review earlier steps in this solution.

- Order on Behalf Solution Workflow
- Design Considerations

Take the next steps in this implementation.

- Configurations

Configurations

Use these configurations to order on behalf of a customer.

These configurations let your agents place orders on behalf of shoppers using the Order Lightning components in Service Cloud and Commerce Cloud.

Note: Before starting configurations, synchronize B2C Commerce and Service Cloud using the b2c-crm-sync reference implementation. See the Integration Reference Implementation section of this document for the GitHub repository wiki instructions link.

Permission Settings

- Before giving agents Order on Behalf permissions, verify that the agents are Service Cloud and Commerce Cloud users with Order on Behalf rights.
- To give the correct permissions, use "Login_On_Behalf," "Login_Agent," and "Create_Order_On_Behalf_Of" business manager functional permissions.
- For more information about permissions, see the B2C Roles and Permissions and Functional Permissions pages.
- Define the agent account authentication settings for external systems in Service Cloud. For more information, see Store Authentication Settings for External Systems.

Lightning Page Layout

1. Drag the Launch Shopping Cart Lightning component to the Contact and Person Account page layouts.

   Note: You can launch OOBO from the Contact Detail or Person Account Detail page layouts. Additionally, if you would like to allow OOBO from a case, you can build a flow from any record that references the contact ID and pass it into B2CCommerce_QuickAction_Contact_OrderOnBehalfOf.

2. Review integration mappings to ensure they align with your business requirements.

   Note: You can view order details in Service Cloud by using Salesforce OMS or your own supported OMS.

Related Content

Review earlier steps in this solution.

- Order on Behalf Solution Workflow
- Design Considerations
- Integration Reference Implementation