



DEMO VERSION

Salesforce

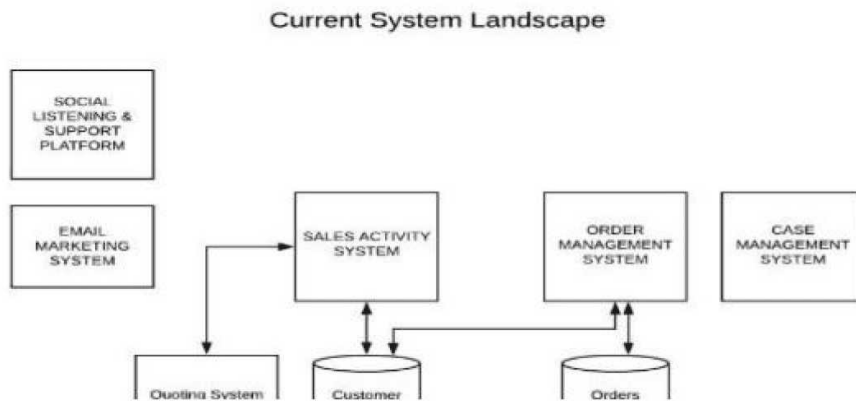
Plat-Arch-204 Exam

Salesforce Certified Platform Integration Architect

Exam Latest Version: 25.0

Question 1. (Single Select)

A large business-to-consumer (B2C) customer is planning to implement Salesforce CRM to become a customer-centric enterprise. Below is the B2C customer's current system landscape diagram.



The goals for implementing Salesforce include: Develop a 360-degree view of the customer. Leverage Salesforce capabilities for marketing, sales, and service processes. Reuse Enterprise capabilities built for quoting and order management processes. Which three systems from the current system landscape can be retired with the implementation of Salesforce?

- A: Order Management, Case Management, and Email Marketing
- B: Sales Activity, Order Management, and Case Management
- C: Email Marketing, Sales Activity, and Case Management

Correct Answer: C

Explanation:

Comprehensive and Detailed 250 to 350 words of Explanation From Salesforce Platform Integration Architect documents: In the role of a Salesforce Platform Integration Architect, evaluating a legacy landscape requires a clinical mapping of current system functions against Salesforce's native capabilities, while strictly adhering to the "Constraints and Goals" provided by the business. The objective here is to maximize the ROI of the Salesforce implementation by consolidating redundant systems into the core platform.

According to Goal 2, the business intends to utilize Salesforce for Marketing, Sales, and Service

processes. Salesforce is architected to handle these three domains through its core clouds: Marketing Cloud (replacing the legacy Email Marketing System), Sales Cloud (replacing the Sales Activity System), and Service Cloud (replacing the Case Management System). By consolidating these three specific functions into Salesforce, the organization achieves Goal 1, which is the creation of a 360-degree view of the customer. When these activities occur on a single platform, the data is unified, eliminating the silos that existed in the previous landscape.

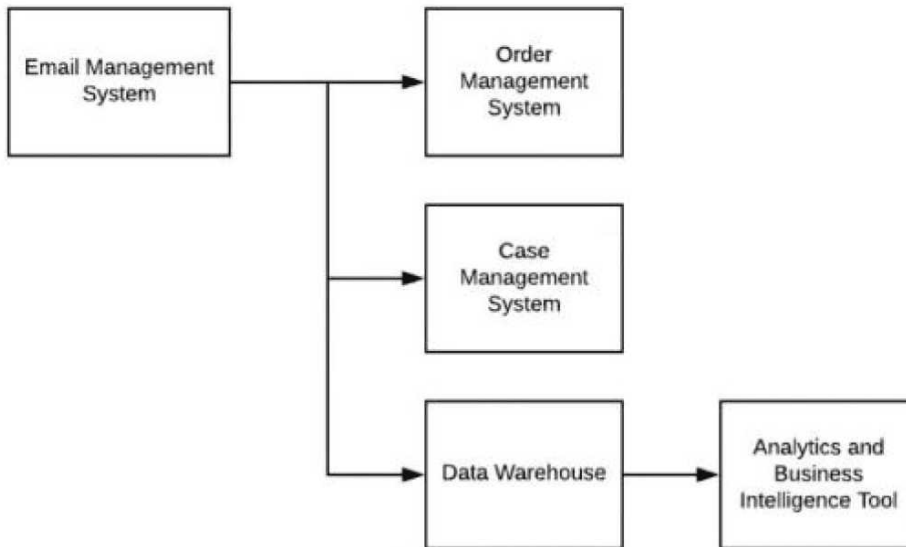
However, the architect must also respect the technical constraints defined in Goal 3, which explicitly states the need to reuse enterprise capabilities built for quoting and order management. In architectural design, this designates the "Quoting System" and the "Order Management System" as systems of record that must remain in the future-state landscape. These systems likely contain complex, proprietary logic or are tightly coupled with back-end ERP systems like SAP Business Suite, making them "non-negotiable" for retirement at this stage.

Therefore, because Email Marketing, Sales Activity, and Case Management map directly to Salesforce's primary strengths and are not excluded by the "reuse" requirement, they are the three systems that should be retired. This strategic retirement simplifies the integration architecture, allowing the architect to focus on building robust integration patterns (such as Request-Reply or Fire-and-Forget) between Salesforce and the remaining Quoting and Order Management systems.

Question 2. (Single Select)

An enterprise customer is planning to implement Salesforce to support case management.

Current System Landscape



Below is their current system landscape diagram. Considering Salesforce capabilities, what should the integration architect evaluate when integrating Salesforce with the current system landscape?

- A: Integrate Salesforce with Data Warehouse, Order Management and Email Management System.
- B: Integrate Salesforce with Order Management System, Data Warehouse and Case Management System.
- C: Integrate Salesforce with Email Management System, Order Management System and Case Management System.

Correct Answer: A

Explanation:

An Integration Architect's primary responsibility when evaluating a landscape for a new Salesforce implementation is to identify the system of record for each business process and determine which legacy systems will be replaced by Salesforce. In this scenario, the customer is implementing Salesforce specifically to support case management.

According to the provided landscape diagram, the Case Management System currently exists as a standalone entity. Since Salesforce Service Cloud provides native, best-in-class case management capabilities, this legacy system is the primary candidate for retirement. Retiring the legacy Case Management system avoids data fragmentation and ensures that Salesforce

serves as the single source of truth for support interactions.

However, for Salesforce to function effectively as a new case management hub, it must integrate with the remaining surrounding systems:

Email Management System: This system likely handles inbound customer communications. An architect must evaluate integrating this with Salesforce (via Email-to-Case or a specialized connector) so that incoming emails automatically generate or update cases.

Order Management System (OMS): Support agents often need to view order history or status to resolve customer inquiries. Integrating Salesforce with the OMS allows for a 360-degree view, enabling agents to see relevant order data directly within the Salesforce case console.

Data Warehouse: For long-term reporting, trend analysis, and a unified customer profile, case data from Salesforce needs to be pushed to the Data Warehouse. This ensures that the Analytics and Business Intelligence Tool downstream can report on support metrics alongside other enterprise data.

Therefore, the architect should evaluate integrations with the Data Warehouse, Order Management, and Email Management System. Option B and C are incorrect because they suggest integrating with the "Case Management System," which is the very system being superseded by Salesforce's native capabilities. By focusing on the integration of these three supporting systems, the architect ensures a seamless transition where Salesforce is fully enriched with the necessary external data to drive support excellence.

Question 3. (Single Select)

An enterprise architect has requested the Salesforce integration architect to review the following (see diagram and description) and provide recommendations after carefully considering all constraints of the enterprise systems and Salesforce Platform limits.

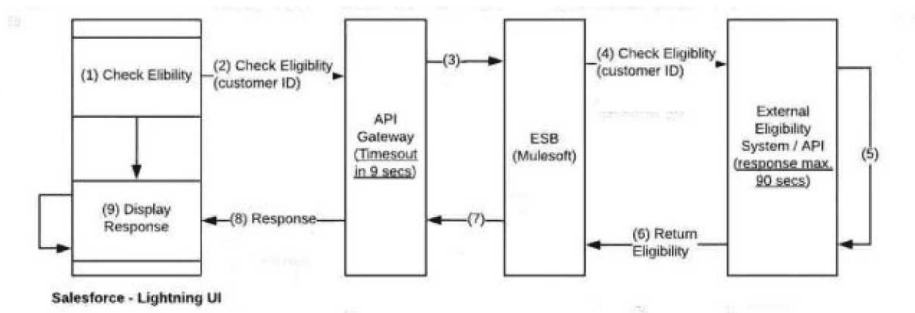
About 3,000 phone sales agents use a Salesforce Lightning user interface (UI) concurrently to check eligibility of a customer for a qualifying offer.

There are multiple eligibility systems that provide this service and are hosted externally.

Their current response times could take up to 90 seconds to process and return.

These eligibility systems are accessed through APIs orchestrated via ESB (MuleSoft).

All requests from Salesforce traverse the customer's API Gateway layer, which imposes a timeout constraint of 9 seconds.



Which recommendation should the integration architect make?

- A: Recommend synchronous Apex callouts from Lightning UI to External Systems via Mule and implement polling on an API Gateway timeout.
- B: Implement a "Check Update" button that passes a requestID received from ESB (user action needed).
- C: Create a platform event in Salesforce via Remote Call-In and use the empAPI in the Lightning UI to serve 3,000 concurrent users when responses are received by Mule.

Correct Answer: C

Explanation:

In this architectural scenario, the Integration Architect must navigate two critical technical "bottlenecks": the 9-second API Gateway timeout and the 90-second backend processing time. Since the backend takes significantly longer than the gateway allows for a synchronous connection, a standard Request-Reply pattern will fail. Furthermore, having 3,000 concurrent agents perform synchronous callouts would risk hitting Salesforce's concurrent long-running request limits.

The most scalable and user-friendly solution is to implement an Asynchronous Request-Reply pattern using Platform Events and the empAPI.

When an agent clicks "Check Eligibility," Salesforce sends an initial asynchronous request to the ESB (MuleSoft). The ESB immediately acknowledges receipt with a 202 Accepted status, freeing up the Salesforce UI thread and avoiding the API Gateway's 9-second timeout. Once the backend eligibility system completes its 90-second process, MuleSoft acts as a client to

Salesforce, performing a Remote Call-In to publish a specific Platform Event containing the result and the original Request ID.

On the frontend, the Lightning UI uses the empAPI (Enterprise Messaging Platform API) to subscribe to the streaming channel for that Platform Event. Because the empAPI uses CometD technology to maintain a single long-lived connection, it can efficiently push the response to the agent's screen the moment it arrives, without requiring the agent to manually refresh or click a "Check Update" button (as suggested in Option B). This provides a "real-time" feel despite the long backend latency.

Option A is non-viable because synchronous polling would exacerbate the load on the API Gateway and likely lead to governance limit issues within Salesforce. By using Platform Events and empAPI, the architect ensures the solution remains within Salesforce's execution limits while providing a seamless, automated experience for a high-volume call center environment.

Question 4. (Single Select)

Northern Trail Outfitters is in the final stages of merging two Salesforce orgs, but needs to keep the retiring org available for a short period of time for lead management as it is connected to multiple public website forms. The sales department has requested that new leads are available in the new Salesforce instance within 30 minutes. Which approach requires the least amount of development effort?

A: Call the Salesforce REST API to insert the lead into the target system.

B: Use the Tooling API with Process Builder to insert leads in real time.

C: Use the Composite REST API to aggregate multiple leads in a single call.

Correct Answer: A

Explanation:

In an org merger scenario, a Salesforce Platform Integration Architect must often design interim solutions that balance speed of delivery with minimal development overhead. The requirement here is for leads to be synced within a 30-minute window. This is a relatively low-latency requirement that does not strictly necessitate a real-time, event-driven architecture, which typically requires more complex infrastructure like Platform Events or middleware.

Option A, calling the Salesforce REST API to insert leads into the target system, represents the standard, most straightforward approach. Since both systems are Salesforce orgs, the retiring org can be configured to make an outbound REST call to the new org's standard Lead endpoint. This can be achieved with a small amount of Apex code (such as a trigger or an invocable method called by a Flow). This approach is considered the "least development effort" because it leverages the standard REST API, which is already enabled and authenticated via a Connected App in the target org. It requires no custom API development in the target system and uses standard JSON payloads.

Option B is incorrect because the Tooling API is intended for managing metadata, system settings, and developer tools, not for standard transactional data movement like Lead insertion. Option C, using the Composite REST API, is an optimization technique designed to group multiple requests into a single call to save on API limits. While efficient for high-volume scenarios, it introduces additional development complexity regarding payload construction and bulk error handling that exceeds the "least effort" requirement for a simple Lead sync. Therefore, a standard REST call is the most direct path to meeting the 30-minute SLA with minimal coding.

Question 5. (Single Select)

A subscription-based media company's system landscape forces many subscribers to maintain multiple accounts and to log in more than once. An Identity and Access Management (IAM) system, which supports SAML and OpenID, was recently implemented to improve the subscriber experience through self-registration and single sign-on (SSO). The IAM system must integrate with Salesforce to give new self-service customers instant access to Salesforce Community Cloud.

A: OpenID Connect Authentication Provider and Just-in-Time (JIT) provisioning¹

B: OpenID Connect Authentication Provider and Registration Handler²

C: SAML SSO and Registration Handler

Correct Answer: B

Explanation:

To provide "instant access" and a seamless experience for Community (Experience Cloud) users, the architect must choose an authentication and provisioning strategy that handles user

creation on-the-fly. While both SAML and OpenID Connect (OIDC) are viable for SSO, OpenID Connect is the modern standard for consumer-facing "Social" or external identity integrations because it is built on OAuth 2.0.

The critical component for "self-service" is the Registration Handler. When an OpenID Connect Authentication Provider is configured in Salesforce, you must associate it with an Apex class that implements the `Auth.RegistrationHandler` interface. This handler is executed during the SSO flow if the user does not already exist. It provides the architect with full programmatic control to:

Match the incoming identity to an existing Contact or Account.

Create a new Contact record if one doesn't exist.

Provision a new User record with the correct Profile, Permission Sets, and Locale settings.

Link the User to the correct Account hierarchy, which is vital for Community security models.

Option A suggests Just-in-Time (JIT) provisioning, which is a declarative way to create users. However, JIT is often too rigid for Experience Cloud requirements, as it has limited ability to perform complex data lookups or handle the specific linking of Contacts to Accounts required for external users. Option C is technically mismatched in common Salesforce terminology; while SAML uses JIT, the Registration Handler is the native, specific mechanism designed to work with Authentication Providers (like OIDC). By using B, the company ensures that a subscriber logging in for the first time via the IAM system is instantly and accurately provisioned in Salesforce, eliminating the need for multiple accounts.

ExamsIndex

Demo PDF Complete

Your Plat-Arch-204 Demo (5 Questions)

Get the Complete Version

Full Questions with Detailed Explanations

Interactive Web-Based Exams Available

To get 30% off, use Coupon Code: NEWYEAR30

<https://examsindex.com/exam/plat-arch-204>