

Salesforce Certified Integration Architect

Salesforce Integration-Architect

Version Demo

Total Demo Questions: 34

Total Premium Questions: 346

Buy Premium PDF

<https://passqueen.com>

support@passqueen.com

Topic Break Down

Topic	No. of Questions
Topic 1, Integration Architecture	58
Topic 2, Integration Patterns	79
Topic 3, Salesforce Integration Capabilities	124
Topic 4, Security	71
Topic 5, Architecture Fundamentals	13
Topic 6, Mix Questions	1
Total	346

QUESTION NO: 1

Universal Containers (UC) support agents would like to open bank accounts on the spot. During the process, agents execute credit checks through external agencies. At any given time, up to 30 concurrent reps will be using the service. Which error handling mechanisms should be built to display an error to the agent when the credit verification process has failed?

- A. Handle Integration errors in the middleware in case the verification process is down, then the middleware should retry processing the request multiple times.
- B. In case the verification process is down, use fire and forget mechanism instead of Request and Reply to allow the agent to get the response back when the service is back online.
- C. Handle the error in the synchronous callout and display a message to the agent.

ANSWER: A

Explanation:

Handle Integration errors in the middleware in case the verification process is down, then the middleware should retry processing the request multiple times. This approach places the external-service resilience logic in the integration layer, where it can use a controlled, bounded retry policy for transient conditions such as a temporary network interruption or a short-lived outage at the credit agency. The middleware can apply retry intervals, enforce a maximum number of attempts, capture diagnostics, and prevent the credit agency's implementation details from leaking into Salesforce.

When the retry limit is reached, the middleware should return a clear, structured failure response to Salesforce. Salesforce can then present an understandable error to the agent during the account-opening interaction, allowing the agent to communicate that verification could not be completed and take the appropriate next step. Centralizing this behavior also provides consistent logging, monitoring, and error-response mapping across credit agencies. The retry policy must remain bounded and respect the synchronous interaction's timeout expectations so that agents receive a timely outcome while transient failures receive an appropriate recovery attempt. MuleSoft documents reconnection and retry behavior for connectivity failures in its [Reconnection Strategy documentation](#). Salesforce also documents how integrations can handle and surface external-service errors through [Named Credential error handling](#).

QUESTION NO: 2

What should an integration architect recommend to ensure all integrations to the Northern Trail Outfitters' company portal use SSL mutual authentication?

- A. Enable My Domain and SSL/TLS
- B. Enforce SSL/TLS Mutual Authentication
- C. Generate a certification authority (CA) signed certificate

ANSWER: B

Explanation:

Enforce SSL/TLS Mutual Authentication is correct because it applies a platform-level requirement that API clients mutually authenticate during the TLS handshake. Mutual TLS extends ordinary server-side TLS validation by requiring the calling integration to present a trusted client certificate. Salesforce validates that certificate before allowing the API connection to proceed, so the portal can verify both the identity of Salesforce and the identity of each connecting client. This is especially appropriate where integrations access sensitive company-portal resources and the organization needs certificate-based client authentication in addition to normal credentials or tokens. The architect should ensure that each authorized integration has an appropriate client certificate, that certificate chains are trusted, and that certificate issuance, rotation, revocation, and expiration monitoring are handled operationally. Enforcing the requirement centrally provides consistent protection across API integrations rather than relying on each integration team to implement mutual TLS independently. Salesforce documentation describes mutual authentication as TLS authentication in which both endpoints present certificates, and identifies its use for securing API connections. See [Salesforce Help: Mutual Authentication](#) and [Salesforce Developer Documentation: Mutual Authentication](#).

QUESTION NO: 3

A Salesforce customer is planning to roll out Salesforce for all their Sales and Service staff.

Senior Management has requested that monitoring is to be in place for Operations to notify any degradation in Salesforce performance.

How should an integration consultant implement monitoring?

- A. Use Salesforce limits API to capture current API usage and configure alerts for monitoring.
- B. Use APIEVENT to track all user initiated API calls through SOAP, REST or BULK APIs.
- C. Identify critical business processes and establish automation to monitor performance against established benchmarks.
- D. Request Salesforce to monitor the Salesforce instance and notify when there is degradation in performance.

ANSWER: C

Explanation:

Identify critical business processes and establish automation to monitor performance against established benchmarks. Effective operational monitoring starts with the business capabilities that users depend on, such as creating a case, searching for an account, converting a lead, or completing an order-related integration. The team should define measurable service-level indicators for these processes, including response time, transaction completion rate, error rate, queue depth, and integration latency. A baseline taken during normal operation makes it possible to detect meaningful degradation rather than merely reporting isolated platform activity.

Automation should execute representative synthetic transactions or collect telemetry from the relevant integration and application components at regular intervals. It can then compare the results with agreed thresholds and alert Operations when a benchmark is exceeded. This approach provides actionable notifications tied to user impact and supports rapid triage across Salesforce, middleware, external services, and network dependencies. Salesforce also provides platform status information, but customer-side monitoring of critical end-to-end processes is necessary because availability alone does not measure the performance experienced by Sales and Service users. See Salesforce guidance on [monitoring Salesforce service performance](#) and the [Salesforce Trust Status site](#).

QUESTION NO: 4

Northern Trail Outfitters (NTO) uses different shipping services for each of the 34 countries it serves. Services are added and removed frequently to optimize shipping times and costs. Sales Representatives serve all NTO customers globally and need to select between valid service(s) for the customer's country and request shipping estimates from that service.

Which two solutions should an architect propose?

Choose 2 answers

- A. Use Platform Events to construct and publish shipper-specific events.
- B. Invoke middleware service to retrieve valid shipping methods.
- C. Use middleware to abstract the call to the specific shipping services.
- D. Store shipping services in a picklist that is dependent on a country picklist.

ANSWER: B C

Explanation:

"Invoke middleware service to retrieve valid shipping methods." is correct because the available carriers vary by country and change frequently. A middleware-owned service catalog can determine the customer's country, apply current eligibility and business rules, and return only the shipping methods that are presently available for selection. This keeps the Salesforce

user experience current without requiring frequent configuration or deployment changes whenever NTO adds, removes, or changes a carrier.

“Use middleware to abstract the call to the specific shipping services.” is also correct because the middleware layer provides a stable integration interface for Salesforce while isolating carrier-specific endpoint URLs, authentication schemes, payload formats, routing logic, and response transformations. Salesforce can submit a normalized estimate request, while middleware selects and invokes the appropriate country-specific provider and returns a normalized estimate response. This façade approach reduces coupling to external services and makes individual carrier changes much less disruptive to Sales Representatives and Salesforce integrations. Salesforce’s integration architecture guidance describes using appropriate integration patterns and decoupling systems where needed: [Salesforce Integration Patterns](#). Salesforce also documents middleware as an integration layer for connecting Salesforce with external systems: [Salesforce Integration Services Overview](#).

QUESTION NO: 5

A company has an external system that processes and tracks orders. Sales reps manage their leads and opportunity pipeline in Salesforce. The company decided to integrate Salesforce and the Order Management System (OMS) with minimal customization and code. Sales reps need to see order history in real-time. The legacy system is on-premise and connected to an ESB. There are 1,000 reps creating 15 orders each per shift, mostly with 20-30 line items. How should an integration architect integrate the two systems based on these requirements?

- A. Use Salesforce standard object, REST API, and extract, transform, load (ETL).
- B. Use Salesforce custom object, custom REST API, and extract, transform, load (ETL).
- C. Use Salesforce external object and OData connector.

ANSWER: C

Explanation:

Use Salesforce external object and OData connector. Salesforce Connect provides data virtualization: external data can be displayed and related in Salesforce without copying the OMS order and line-item records into Salesforce storage. This directly supports the need for sales reps to view current order history in real time, because Salesforce retrieves the external-object data from the configured external data source when it is accessed rather than relying on a periodically replicated dataset.

The OMS is on premises but already connected to an enterprise service bus, making the ESB an appropriate layer to expose a standards-based OData service to Salesforce. Salesforce Connect supports OData external data sources, enabling the integration to use declarative external objects and mappings instead of building and maintaining custom synchronization logic. This is especially appropriate for the stated order volumes and line-item counts, since historical order data can remain mastered in the OMS while Salesforce presents it in the sales workflow. The approach minimizes Salesforce data-storage consumption and avoids the operational burden of loading, reconciling, and keeping a large replicated order-history dataset current. Salesforce documentation describes external objects as objects that map to data stored outside Salesforce and identifies OData as a supported external data source type. See [Salesforce External Objects](#) and [Connect to an OData Service](#).

QUESTION NO: 6

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. Email Marketing, Sales Activity, and Case Management
- B. Order Management, Case Management, and Email Marketing
- C. Sales Activity, Order Management, and Case Management
- D. Sales Activity, Email Marketing, and Order Management

ANSWER: A

Explanation:

Email Marketing, Sales Activity, and Case Management is correct because these are the functional areas explicitly targeted for replacement by Salesforce capabilities. Salesforce provides marketing functionality to manage audience engagement and campaign journeys, sales functionality to manage activities and interactions throughout the customer lifecycle, and Service Cloud functionality to manage customer cases and support processes. Consolidating these capabilities in Salesforce supports the stated customer-centric objective by bringing marketing, sales, and service interactions closer to a unified customer record, which contributes to a 360-degree customer view. Salesforce describes Customer 360 as connecting customer data across applications so teams can use a more complete customer view: [Salesforce Customer 360](#). Salesforce Service Cloud also includes case management capabilities for tracking, routing, and resolving customer issues: [Salesforce Service Cloud](#). In contrast, the stated architecture direction is to reuse the enterprise quoting and order-management capabilities rather than replace them. Therefore, the systems supporting email marketing, sales activity, and case management are the appropriate retirement candidates after the Salesforce implementation and required integration work are completed.

QUESTION NO: 7

An Integration Architect has designed a mobile application for Salesforce users to get data while on the road using a custom UI. The application is secured with OAuth and is currently functioning well. There is a new requirement where the mobile application needs to obtain the GPS coordinates and store it on a custom geolocation field.

The geolocation field is secured with Field Level Security, so users can view the value without changing it.

What should be done to meet the requirement?

- A. The mobile device makes a SOAP API inbound call.
The mobile device receives a REST Apex callout call.
- B. The mobile device makes a REST API inbound call.
- C. The mobile device makes an Apex REST inbound call.

ANSWER: C

Explanation:

The mobile device makes a REST Apex inbound call. An Apex REST resource provides a custom HTTPS endpoint that the OAuth-authenticated mobile application can invoke to submit the device's GPS coordinates. The endpoint can validate the request, identify the applicable Salesforce record, and perform the update server-side. This is appropriate when the mobile user must be permitted to send a narrowly defined business operation, while the field remains non-editable through normal user-facing and standard API mechanisms.

Apex generally executes database operations in system mode, so object- and field-level permissions are not automatically enforced for the DML operation. The implementation should therefore be deliberately limited to the intended record and geolocation field, validate that the caller is authorized for the business action, and avoid accepting arbitrary field names or record updates from the client. The Apex REST class is exposed with `@RestResource`, and its HTTP method can deserialize the submitted latitude and longitude before updating the custom geolocation field. Salesforce documents that Apex REST exposes Apex classes as RESTful web services and that Apex does not automatically enforce CRUD and field-level security; developers can enforce those controls explicitly where needed. See [Apex REST documentation](#) and [Apex security and field-level access documentation](#).

QUESTION NO: 8

A company's security assessment noted vulnerabilities on the un managed packages in their Salesforce orgs, notably secrets that are easily accessible and in plain text, such as usernames, passwords, and OAuth tokens used in callouts from Salesforce.

Which two persistence mechanisms should an integration architect require to be used to ensure that secrets are protected from deliberate or inadvertent exposure?

Choose 2 answers

- A. Encrypted Custom Fields
- B. Named Credentials
- C. Protected Custom Metadata Types
- D. Protected Custom Settings

ANSWER: B D

Explanation:

Named Credentials provide the appropriate Salesforce-managed persistence for endpoint authentication details used by callouts. A named credential keeps the endpoint definition and authentication configuration separate from Apex and other declarative artifacts, so code refers to the named credential rather than embedding usernames, passwords, tokens, or authorization-header construction. Salesforce supports external credentials and principals for securely associating authentication methods and access with a named credential. This reduces the chance that secrets are exposed through source control, package contents, debug logs, or code review.

Protected Custom Settings are appropriate for sensitive package-specific configuration that must be persisted and accessed by code in the package namespace. When protected in a managed package, the setting is not exposed to package subscribers, helping prevent accidental viewing or modification of values such as integration-specific keys and tokens. The integration architect should also ensure that access is limited to the package's trusted code path and that deployment and operational processes do not export secret values into logs or repositories. Together, these mechanisms separate confidential integration values from ordinary, readable application configuration and hard-coded implementation details. See Salesforce's [Named Credentials documentation](#) and [Custom Settings documentation](#).

QUESTION NO: 9

An enterprise customer that has more than 10 Million customers has the following systems and conditions in their landscape:

- A. Enterprise Billing System (EBS) - All customer's monthly billing is generated by this system.
- B. Enterprise Document Management System (DMS) Bills mailed to customers are maintained in the Document Management system.
- C. Salesforce CRM (CRM)- Customer information, Sales and Support information is maintained in CRM.

ANSWER: B C

Explanation:

"Enterprise Document Management System (DMS) Bills mailed to customers are maintained in the Document Management system." and "Salesforce CRM (CRM)- Customer information, Sales and Support information is maintained in CRM." correctly identify distinct systems of record for information with materially different business purposes. The document management system is the appropriate authoritative repository for generated and mailed bill documents, including their retention, retrieval, and document-lifecycle requirements. Salesforce CRM is appropriately positioned as the operational system for customer, sales, and support data, where users need current customer context to sell, service, and manage relationships.

For an enterprise landscape at this scale, an integration architecture should explicitly define ownership by data domain and synchronize or expose only the data required by each consuming system. This avoids creating competing masters for customer-facing data and allows the CRM to remain focused on operational interactions while the document platform manages high-volume document storage and access. Salesforce's integration guidance describes selecting patterns based on the source of truth, data volume, latency, and consumer needs. The Salesforce Architect integration-pattern guidance is available at [Salesforce Integration Patterns](#), and Salesforce's API architecture guidance is available at [Salesforce Architect Guide](#).

QUESTION NO: 10

A company's security assessment noted vulnerabilities on the un managed packages in their Salesforce orgs, notably secrets that are easily accessible and in plain text, such as usernames, passwords, and OAuth tokens used in callouts from Salesforce.

Which two persistence mechanisms should an integration architect require to be used to ensure that secrets are protected from deliberate or inadvertent exposure?

Choose 2 answers

- A. Encrypted Custom Fields
- B. Named Credentials
- C. Protected Custom Metadata Types
- D. Protected Custom Settings

ANSWER: A B

Explanation:

Encrypted Custom Fields provide an appropriate org-level persistence mechanism when a secret must be stored as application data. Salesforce encrypts values in encrypted field types at rest, and access to the unmasked value is controlled through field-level security and the *View Encrypted Data* permission. This substantially reduces the risk that credentials or tokens are casually exposed through record pages, reports, exports, or configuration reviews. Architects should still restrict access to these fields to the smallest possible set of users and integration processes. Salesforce documents encrypted fields and their access behavior in its [Encrypted Fields documentation](#).

Named Credentials are the preferred persistence mechanism for authentication details used by outbound callouts. A named credential keeps the endpoint and authentication configuration together while allowing Apex and other callout mechanisms to refer to the credential by name rather than embedding usernames, passwords, client secrets, or OAuth tokens in source code or ordinary configuration. For OAuth-based integrations, Salesforce can manage authorization and token use as part of the external credential and named credential configuration. This supports separation of code from secrets and enables administrators to control credential access and rotation without modifying deployed callout logic. See Salesforce's [Named Credentials overview](#).

QUESTION NO: 11

Customer is evaluating Platform Events solution and would like help in comparing/contrasting it with Outbound Message for a real-time / near-real time needs. They expect 3,000 consumers of messages from Salesforce.

Which three considerations should be evaluated and highlighted when deciding between the solutions?

Choose 3 answers

- A. Both Platform Events and Outbound Message offer declarative means for asynchronous near-real time needs. They aren't best suited for real-time integrations.
- B. In both Platform Events and Outbound Messages, the eventmessages are retried by and delivered in sequence, and only once. Salesforce ensures there is no duplicate message delivery.

C. Message sequence is possible in Outbound Message but not guaranteed with Platform Events. Both offer very high reliability. Fault handling and recovery are fully handled by Salesforce.

D. Number of concurrent subscribers to Platform Events is capped at 2,000. An Outbound Message configuration can pass only 100 notifications in a single message to a SOAP end point.

E. Both Platform Events and Outbound Message are highly scalable. However, unlike Outbound Message, only Platform Events have Event Delivery and Event Publishing limits to be considered.

ANSWER: A D E

Explanation:

Both Platform Events and Outbound Message are declarative, asynchronous integration capabilities that support near-real-time processing. This is an important architectural distinction when the business requirement is truly synchronous, real-time request/response behavior: neither mechanism should be positioned as the primary solution for that type of interaction. For the stated scale, the concurrent-subscriber limit is particularly relevant. Platform Events support up to 2,000 concurrent CometD subscribers per organization, so an expected population of 3,000 consumers requires an alternative consumption design, such as a smaller number of shared subscribers that distribute events downstream. Outbound Messages can include up to 100 notifications in a single SOAP request, which is likewise relevant to endpoint throughput and batching design.

Platform Events also require explicit capacity planning for event publishing and delivery allocations. Publishing is governed by the organization's platform-event publishing entitlement, while delivery consumption is counted when events are delivered to subscribers. These limits can materially affect a high-volume fan-out design, especially because each delivery to a subscriber contributes to delivery usage. Salesforce documents these allocation considerations in the [Platform Event Allocations](#) guide and documents outbound-message behavior in the [Outbound Messaging Developer Guide](#).

QUESTION NO: 12

A healthcare services company maintains a Patient Prescriptions System that has 50+

million records in a secure database. Their customer base and data set growing rapidly. They want to make sure that the following policies are enforced:

1. Identifiable patient prescriptions must exist only in their secure system's database and encrypted at rest.
2. Identifiable patient prescriptions must be made available only to people explicitly authorized in the Patient Prescriptions System assigned nurses and doctors, patient, and people explicitly the patient may authorize.
3. Must be available only to verified and pre-approved people or legal entities.

To enable this, the company provides the following capabilities:

1. One-time use identity tokens for patients, nurses, doctors, and other people that expire within a few minutes.
2. Certificates for legal entities.
- . RESTful services.

The company has a Salesforce Community Cloud portal for patients, nurses, doctors, and other authorized people. A limited number of employees analyze the identified data in Einstein

Analytics.

Which two capabilities should the integration architect require for the Community Cloud portal and Einstein Analytics?

Choose 2 answers

- A Identity token data storage
- B. Bulk load for Einstein Analytics
- C. Callouts to RESTful services
- D. Encryption in transit and at rest

Answer: B, C

Explanation:

- A. Bulk load for Einstein Analytics
- B. Callouts to RESTful services
- C. Encryption in transit and at rest

ANSWER: A B

Explanation:

Bulk load for Einstein Analytics is required because the analytics users need to work with a de-identified subset of the prescription data rather than query or store identifiable prescription records in Salesforce. A bulk ingestion pattern is appropriate for moving large volumes of prepared, de-identified data into Analytics, particularly given the source system's scale and rapid growth. The integration should ensure that de-identification occurs before the analytics dataset is loaded and that the dataset contains only the fields necessary for analysis.

Callouts to RESTful services are required for the Community Cloud portal because the Patient Prescriptions System remains the authoritative location for identifiable prescription data. Salesforce can invoke the secure system's REST API at request time, presenting the applicable short-lived identity token and allowing the prescription system to make the authorization decision for the patient, assigned clinicians, and explicitly authorized parties. This approach supports data minimization by avoiding replication of identifiable prescriptions into the portal while allowing verified users to access current information through the source system's controlled services. Salesforce documents outbound HTTP integrations through Apex callouts at [Salesforce Apex Callouts](#); Salesforce's Analytics data-ingestion guidance is available at [Integrate Data for CRM Analytics](#).

QUESTION NO: 13

A company has several external applications using Salesforce APIs. The Integration Architect must provide operations staff with visibility into API consumption and detailed audit information about API activity.

Which two capabilities should the Integration Architect recommend?

Choose 2 answers

- A. Use the Limits resource to monitor API consumption.
- B. Use Event Monitoring API event logs to analyze API activity.
- C. Use debug logs as the primary long-term API usage monitoring solution.
- D. Use Apex test execution results to measure API consumption.

ANSWER: A B

Explanation:

Use the Limits resource to obtain current API usage and remaining allocations programmatically. An operational monitoring service can periodically retrieve the organization limits and alert administrators before an integration reaches a consumption threshold.

Use Event Monitoring API event logs for detailed analysis of API activity. Event Monitoring captures API-related events that can be used to investigate client activity, request patterns, user context, and operational anomalies. Together, limits monitoring and event-log analysis provide both capacity visibility and diagnostic information. See [Limits REST API Resource](#) and [Event Log File REST API Resource](#).

QUESTION NO: 14

An Integration Architect has built a solution using REST API, updating Account, Contact, and other related information. The data volumes have increased, resulting in higher API calls consumed, and some days the limits are exceeded. A decision was made to decrease the number of API calls using bulk updates. The customer prefers to continue using REST API to avoid architecture changes.

Which REST API composite resources should the Integration Architect use to allow up to 200 records in one API call?

- A. sObject Collections
- B. sObject Tree
- C. Batch
- D. Composite

ANSWER: A

Explanation:

sObject Collections is the appropriate REST composite resource because its collection endpoints support processing up to 200 sObject records in one request. This directly addresses the requirement to reduce API-call consumption while retaining the existing REST-based integration architecture. The integration can send a collection of records to create, update, upsert, or delete through the relevant sObject Collections endpoint, rather than issuing an individual API request for each Account, Contact, or related record.

For update operations, the request body contains a `records` array whose entries identify the target sObject type and record ID, along with the fields to change. The API also provides an `allOrNone` setting, allowing the architect to select transactional behavior appropriate for the integration: either roll back the entire collection on an error or permit successful records to be committed while reporting individual failures. This batching pattern can materially lower daily API usage without requiring migration to Bulk API or a different integration design. Salesforce documents the supported collection operations and the 200-record request limit in its [sObject Collections REST API reference](#). Additional implementation details are available in the [Composite Resources overview](#).

QUESTION NO: 15

Northern Trail Outfitters sends order-submission messages from Salesforce to a fulfillment application. The fulfillment application can be unavailable for several hours, but orders must not be lost and duplicate submissions must not create duplicate fulfillment orders.

Which two design elements should an Integration Architect recommend?

Choose 2 answers

- A. Use a durable middleware queue to persist and retry order messages.
- B. Require idempotent processing using an external order identifier or idempotency key.
- C. Make Salesforce retry the synchronous callout until the fulfillment application responds.
- D. Use the Salesforce record ID as the only error-handling mechanism.

ANSWER: A B

Explanation:

A **durable queue in middleware** decouples Salesforce from the availability of the fulfillment application. Middleware can accept or subscribe to the order message, persist it until the fulfillment service is available, and apply controlled retry and monitoring policies. This prevents a temporary downstream outage from requiring Salesforce users to resubmit orders manually.

The fulfillment application should **support idempotent order processing** using an immutable external order identifier or idempotency key. If middleware retries after a timeout or uncertain response, the fulfillment system can recognize that the same order was previously accepted and avoid creating a duplicate fulfillment transaction. See Salesforce's [Integration Patterns guidance](#).

QUESTION NO: 16

Service agents at Northern Trail Outfitters use Salesforce to manage cases and B2C Commerce for ordering. Which integration solution should an architect recommend in order for the service agents to see order history from a business-to-consumer (B2C) Commerce system?

- A. Salesforce B2C Commerce to Service Cloud Connector
- B. REST API offered by Commerce Platforms
- C. MuleSoft Anypoint Platform

ANSWER: A

Explanation:

The Salesforce B2C Commerce to Service Cloud Connector is the appropriate recommendation because it is a purpose-built integration for connecting B2C Commerce customer and order information with the Service Cloud agent experience. It enables service representatives to access a shopper's commerce context, including order history, without requiring them to leave Salesforce while working a case. This directly supports the requirement for agents to see order details from the commerce system and deliver more informed customer service.

The connector is designed to reduce the implementation effort and ongoing maintenance associated with building and operating a bespoke point-to-point integration. It uses the established B2C Commerce and Service Cloud integration model to expose commerce capabilities in the service workflow, allowing an architect to favor a supported, reusable Salesforce solution where the requirement aligns with its capabilities. The implementation can also provide connected customer information and service-oriented commerce actions, helping agents resolve order-related inquiries in the same workspace where they manage cases.

Salesforce's B2C Commerce developer documentation provides the platform context for these integrations: [B2C Commerce Developer Documentation](#). The connector's implementation resources are maintained by Salesforce Commerce Cloud: [Service Cloud Connector repository](#).

QUESTION NO: 17

A company in a heavily regulated industry requires data in legacy systems to be displayed in Salesforce user interfaces (UIs). They are proficient in their cloud-based ETL (extract, transform, load) tools. They expose APIs built on their on-premise middleware to cloud and on-premise applications.

Which two findings about their current state will allow copies of legacy data in Salesforce?

Choose 2 answers

- A. Only on-premise systems are allowed access to legacy systems.
- B. Cloud-based ETL can access Salesforce and supports queues.

C. On-premise middleware provides APIs to legacy systems data.

D. Legacy systems can use queues for on-premise integration.

ANSWER: B C

Explanation:

“Cloud-based ETL can access Salesforce and supports queues.” is a key enabling finding because an ETL platform that can connect to Salesforce can perform the load portion of a replication process. Queue support further enables decoupled, asynchronous processing, which is useful when legacy extracts are large, arrive intermittently, or must be retried without tying up a synchronous request. Salesforce’s asynchronous bulk-processing capabilities are designed for high-volume data ingestion and are commonly used by ETL-based integrations.

“On-premise middleware provides APIs to legacy systems data.” supplies the required extraction path. Since the legacy systems are accessed through middleware APIs that are exposed to cloud applications, the cloud ETL tool can retrieve the source data, apply any required mappings or transformations, and write a Salesforce-side copy. Together, these findings establish both ends of a batch replication architecture: controlled API access to the on-premise source and a scalable, asynchronous mechanism for loading Salesforce. This approach supports displaying locally stored, integrated data in Salesforce UIs while retaining the legacy system as the originating source of the data. Salesforce’s integration guidance discusses selecting data-integration approaches based on data volume, latency, and system connectivity; its Bulk API documentation describes asynchronous processing for large data loads. See [Salesforce Data Integration Decision Guide](#) and [Salesforce Bulk API 2.0 Developer Guide](#).

QUESTION NO: 18

A company that is a leading provider of courses and training delivers courses using third

party trainers. The trainer for the company has to be verified from 10 different training

accreditation verification agencies before providing training for the company. Each training accreditation agency has its own response time, which could take days to confirm a trainer.

The company decided to automate the trainer accreditation verification process by integrating to the agencies web services.

What is the recommended approach to automate this process?

A. Use salesforce external service to make the call out, Salesforce external service should check the verification agencies until the result is verified, then update the trainer status to "verified".

B. Create a trigger on the trainer record to make a Callout to each verification agencies, write business logic to consolidate the verification then update the trainer status to verified".

C. Make an apex callout using @future annotation to make the call out to all different agencies. The response should update the trainer status to "verified".

D. Use middleware to handle the call out to the 10 different verification services, the middleware will handle the business logic of consolidating the verification result from the 10 services, then make a call-in to Salesforce and update the verification status to "verified".

ANSWER: D

Explanation:

Use middleware to handle the call out to the 10 different verification services, the middleware will handle the business logic of consolidating the verification result from the 10 services, then make a call-in to Salesforce and update the verification status to "verified". This is appropriate because the process is long-running, requires aggregation of responses from numerous independent external systems, and each agency can respond on a different timetable. Middleware can persist the verification request and its correlation identifiers, independently manage retries and timeouts, receive or poll for delayed results, and apply the required aggregation rule only after the necessary agency outcomes are available. It also decouples Salesforce from the agencies’ individual protocols, availability characteristics, and potentially multi-day processing windows. Once the consolidated outcome is ready, middleware can securely invoke Salesforce through an API to update the trainer’s verification

status. This architecture keeps Salesforce focused on its system-of-record role while placing cross-service orchestration in an integration layer designed for durable asynchronous processing. Salesforce's integration architecture guidance describes middleware as a suitable layer for orchestration, transformation, and routing across multiple systems. See [Salesforce Integration Patterns](#) and [Salesforce Architecture Gallery](#).

QUESTION NO: 19

A global financial company sells financial products and services that include the following:

1. Bank Accounts
2. Loans
3. Insurance

The company has a core banking system that is state of the art and is the master system to store financial transactions, financial products and customer information. The core banking system currently processes 10M financial transactions per day. The CTO for the company is considering building a community port so that customers can review their bank account details, update their information and review their account financial transactions.

What should an integration architect recommend as a solution to enable customer community users to view their financial transactions?

- A. Use Salesforce Connect to display the financial transactions as an external object.
- B. Use Salesforce Connect to display the financial transactions as an external object.
- C. Use Salesforce External Service to display financial transactions in a community lightningpage.
- D. Use Iframe to display core banking financial transactions data in the customer community.

ANSWER: A

Explanation:

Use Salesforce Connect to display the financial transactions as an external object. Salesforce Connect is appropriate because the core banking platform remains the system of record for a very large, continuously changing transaction dataset. It exposes eligible external data in Salesforce through external objects, allowing Experience Cloud users to access transaction information without replicating the full transaction history into Salesforce. This avoids the storage, synchronization, latency, and operational concerns that would accompany copying millions of daily financial transactions into the Salesforce database.

External objects contain metadata and use an external data source to retrieve records on demand. The integration architect can configure an appropriate adapter and external object relationships so that a logged-in customer sees only transaction records associated with that customer or account. Access must be designed with the bank's authentication, authorization, sharing, and data-protection requirements in mind, but the core principle is that Salesforce provides the community-facing experience while the banking platform retains ownership of the transactional data. Salesforce documents Salesforce Connect and external objects at [External Objects](#) and provides implementation guidance at [Salesforce Connect Developer Guide](#).

QUESTION NO: 20

Universal Containers (UC) is a global financial company. UC support agents would like to open bank accounts on the spot for a customer who is inquiring about UC products.

During opening the bank account process, the agents execute credit checks for the customers through external agencies. At a given time, up to 30 concurrent rewill be using the service for performing credit checks for customers.

What error handling mechanisms should be built to display an error to the agent when the credit verification process failed?

- A. In case the verification process is down, Use mock service to send the response to the agent.
- B. Handle verification process error in the Verification Webservice API in case there is a connection issue to the Webservice if it responds with an error.
- C. Handle integration errors in the middleware in case the verification process is down, then the middleware should retry processing the request multiple times.
- D. In case the verification process is down, use fire and forget mechanism instead of request and reply to allow the agent to get the response back when the service is bar online.

ANSWER: B

Explanation:

Handle verification process error in the Verification Webservice API in case there is a connection issue to the Webservice if it responds with an error. The account-opening flow requires the credit decision synchronously: the agent needs an immediate, actionable result before the bank-account process can continue. Therefore, the service endpoint invoked by Salesforce should detect both transport-level failures, such as timeouts or an unavailable endpoint, and application-level failures returned by the credit agency, such as an error HTTP status or an invalid response payload. It should translate those conditions into a clear, controlled error response that the Salesforce user interface can present to the agent. This avoids exposing low-level technical details while ensuring that the agent knows the verification was not completed and can follow the appropriate business process, such as asking the customer to try again later or referring the case for review. Salesforce callouts expose HTTP status codes and response bodies so integration code can evaluate the external service response and handle failures explicitly. Salesforce also documents that callouts can fail because of network and endpoint conditions, which should be handled by the calling integration logic. See [Salesforce Apex Callouts documentation](#) and [Salesforce Named Credentials callout guidance](#).

QUESTION NO: 21

Northern Trail Outfitters submits orders to the manufacturing system web-service. Recently, the system has experienced outages that keeping service unavailable for several days.

What solution should an architect recommend to handle errors during these types of service outages?

- A. Use middleware queuing and buffering to insulate Salesforce from system outages.
- B. A Use Platform Event replay and custom scheduled Apex process to retrieve missed events.
- C. Use @future job and custom scheduled apex process to retry failed service calls.
- D. Use Outbound Messaging to automatically retry failed service calls.

ANSWER: A

Explanation:

Use middleware queuing and buffering to insulate Salesforce from system outages. This pattern decouples order submission in Salesforce from the availability of the manufacturing web service. Middleware can accept and durably persist each outbound order message, then deliver it to the manufacturing system when that system becomes available. For an outage lasting days, the queue provides a controlled backlog instead of requiring Salesforce to maintain synchronous callouts, repeatedly execute Apex, or risk losing requests. A resilient middleware implementation should use durable storage, retry policies with backoff, monitoring and alerting, and idempotency or a correlation identifier so a retried message does not

create a duplicate order. It should also support a dead-letter or exception process for messages that cannot be delivered after defined retry thresholds. This is aligned with Salesforce integration guidance to use asynchronous, loosely coupled messaging for systems that may be temporarily unavailable, while allowing Salesforce users and business processes to continue creating orders. Salesforce's integration patterns documentation describes queue-based asynchronous processing as a way to separate message production from consumption and improve resilience: [Salesforce Integration Patterns](#). Salesforce also describes middleware as an integration layer for routing, transformation, and reliable delivery between systems: [Salesforce Middleware Decision Guide](#).

QUESTION NO: 22

Northern Trail Outfitters wants to improve the quality of call-outs from Salesforce to their REST APIs. For this purpose, they will require all API clients/consumers to adhere to REST API Markup Language (RAML) specifications that include field-level definition of every API request and response payload. RAML specs serve as interface contracts that Apex REST API Clients can rely on.

Which two design specifications should the Integration Architect include in the integration architecture to ensure that Apex REST API Clients unit tests confirm adherence to the RAML specs?

Choose 2 answers

- A. Call the Apex REST API Clients in a test context to get the mock response.
- B. Require the Apex REST API Clients to implement the `HttpCalloutMock`.
- C. Call the `HttpCalloutMock` implementation from the Apex REST API Clients.
- D. Implement `HttpCalloutMock` to return responses per RAML specification.

ANSWER: A D

Explanation:

Call the Apex REST API Clients in a test context to get the mock response. is required because an Apex unit test must execute the client code that constructs the HTTP request, performs the callout, and deserializes the returned payload. This makes the contract-oriented behavior testable without sending traffic to the actual external REST endpoint. The test can then assert that the client correctly processes the representative response defined by the RAML contract.

Implement `HttpCalloutMock` to return responses per RAML specification. is also required because Salesforce tests do not make live HTTP callouts. A class implementing `HttpCalloutMock` supplies the HTTP response used during the test after it is registered with `Test.setMock`. The mock response body, status code, and headers should reflect the RAML-defined response contract. In addition, the mock implementation receives the outbound `HttpRequest`, so it can validate relevant method, endpoint, headers, and serialized payload fields against the expected contract before returning the RAML-compliant response. This provides deterministic tests of both the outbound request shape and inbound response handling. Salesforce documents this testing pattern in its [Apex HTTP callout testing documentation](#) and the [HttpCalloutMock interface reference](#).

QUESTION NO: 23

Northern Trail Outfitters wants to use Salesforce as a front end for creating accounts using the lead-to-opportunity process.

1. An order is created in Salesforce when the opportunity is closed and won, but the back-end ERP system is the data master for order,

2. Customer wants to be able to see within Salesforce all the stages of order processing like Order Created, OrderShipped, Order Paid that are within the retention window.

Which two message durability considerations should an Integration Architect make when designing a solution to meet these business requirements?

Choose 2 answers

- A. When subscribing to Salesforce Event bus, ReplayID is used with a value of -2 to be able to see old and new events.
- B. High-volume event messages are stored for 24 hours (one day).
- C. When subscribing to Salesforce Event bus, ReplayID is used with a value of -1 to be able to see new events.
- D. High-volume event messages are stored for 72 hours (three days).

ANSWER: A D

Explanation:

Using a ReplayID value of -2 when subscribing to the Salesforce event bus enables a subscriber to receive retained event messages, beginning with the earliest event available in the retention window, and then continue receiving newly published events. This supports reliable recovery when an ERP integration subscriber is restarted, temporarily unavailable, or needs to rebuild the recent order-status timeline shown in Salesforce. The subscriber can therefore process the retained Order Created, Order Shipped, and Order Paid status events that remain available before processing subsequent real-time updates.

High-volume platform event messages are retained for 72 hours. This retention period is a core durability constraint: the integration architecture must ensure that subscribers resume and replay required messages within that three-day window. For an ERP system that is the master for orders, publishing order-processing changes as events and consuming them with replay gives Salesforce a timely view of order progress while allowing recovery from transient consumption failures. Architects should also persist the last successfully processed replay position externally so the subscriber can resume from a known checkpoint during the available retention period. Salesforce documents replay behavior and retention in its [event message durability guide](#) and [event replay guide](#).

QUESTION NO: 24

A US business-to-consumer (B2C) company is planning to expand to Latin America. They project an initial Latin American customer base of about one million, and a growth rate of around 10% every year for the next 5 years. They anticipate privacy and data protection requirements similar to those in the European Union to come into effect during this time. Their initial analysis indicates that key personal data is stored in the following systems:

- 1. Legacy mainframe systems that have remained untouched for years and are due to be decommissioned.
- 2. Salesforce Commerce Cloud Service Cloud, Marketing Cloud, and Community Cloud.
- 3. The company's CIO tasked the integration architect with ensuring that they can completely delete their Latin American customer's personal data on demand.

Which three requirements should the integration architect consider?

Choose 3 answers

- A. Manual steps and procedures that may be necessary.
- B. Impact of deleted records on system functionality.
- C. Ability to delete personal data in every system.
- D. Feasibility to restore deleted records when needed.
- E. Ability to provide a 360-degree view of the customer.

ANSWER: A B C

Explanation:

“Manual steps and procedures that may be necessary,” “Impact of deleted records on system functionality,” and “Ability to delete personal data in every system” are essential requirements for a customer-data erasure capability. A compliant deletion process begins with data discovery: the organization must identify where a person’s data exists across Salesforce clouds, integrations, and legacy platforms, including replicated or derived records. The process must be able to execute or coordinate deletion in each applicable system and maintain appropriate evidence that the request was handled.

Deletion can also affect operational processes, data relationships, reporting, entitlement history, identity resolution, and integrations. The architect therefore needs to assess those dependencies and design a controlled process that fulfills the request without creating unexpected service failures or inconsistent downstream records. Because older mainframes may lack modern APIs or may be near decommissioning, the operational design may legitimately include documented manual activities, ownership, approvals, validation, and exception handling rather than assuming all erasure can be automated.

This approach aligns with the GDPR right to erasure, which requires controllers to erase personal data when applicable, and Salesforce guidance on managing personal data and privacy requests. See the [GDPR Article 17 right to erasure](#) and [Salesforce personal data management documentation](#).

QUESTION NO: 25

business requires automating the check and update of the phone number type classification (mobile vs. landline) for all in-coming calls delivered to their phone sales agents. The following conditions exist:

1. At peak, their call center can receive up to 100,000 calls per day.
2. The phone number type classification is a service provided by an external service API.
3. Business is flexible with timing and frequency to check and update the records (throughout the night or every 6-12 hours is sufficient).

A Remote-Call-In pattern and/or Batch Synchronization (Replication via ETL: System -> Salesforce) are determined to work with a middleware hosted on custom premise.

In order to implement these patterns and mechanisms, which component should an integration architect recommend?

- A. ConnectedApp configured in Salesforce to authenticate the middleware.
- B. IoConfigure Remote Site Settings in Salesforce to authenticate the middleware.
- C. An API Gateway that authenticates requests from Salesforce into the Middleware(ETL/ESB).
- D. Firewall and reverse proxy are required to protect internal APIs and resource being exposed.

ANSWER: A

Explanation:

ConnectedApp configured in Salesforce to authenticate the middleware. is the appropriate component because the on-premises middleware is the client initiating authenticated API requests into Salesforce. Both the Remote Call-In pattern and batch replication from an ETL or ESB process require a secure, supported identity and authorization mechanism for the external integration client. A Salesforce connected app provides the OAuth configuration, including a consumer key and consumer secret, and defines the scopes that govern API access. The middleware can obtain an access token and use Salesforce REST, SOAP, Bulk, or other APIs to query and update the phone records according to the scheduled processing model.

For unattended server-to-server processing, the OAuth client credentials flow is commonly suitable when the integration is associated with a dedicated integration user and the connected app is configured with the needed API scopes and

authorization policies. This separates the middleware's credentials and permissions from those of individual sales agents, supports least-privilege access, and enables operational governance such as connected-app policies and login history. Salesforce documents connected apps as the mechanism for integrating external applications through OAuth and documents client credentials for server-to-server authorization. See [Salesforce Connected Apps Overview](#) and [OAuth Client Credentials Flow](#).

QUESTION NO: 26

A company that is a leading provider of courses and training delivers courses using third-party trainers. The trainer for the company has to be verified by 10 different training accreditation verification agencies before providing training for the company. Each training accreditation agency has its own response time, which means it could take days to confirm a trainer. The company decided to automate the trainer accreditation verification process by integrating it with the agency's web service1s. What is the recommended approach to automa2te this process?3456

- A. Use middleware to handle the callouts to the 10 different verification services; the middleware will consolidate the verification results from the 10 services. Then, make a call-in to Salesforce and update the verification status to "verified".
- B. Use Salesforce External Services to make the callouts; Salesforce External Services should check the verification agencies until the result is verified. Then, update the trainer status to "verified".
- C. Make Apex callouts using the @future annotation to call all different agencies.

ANSWER: A

Explanation:

Use middleware to handle the callouts to the 10 different verification services; the middleware consolidates their verification results and then calls in to Salesforce to update the trainer's verification status to "verified". This is the appropriate integration architecture because the verification process is long-running, requires communication with multiple independent external services, and may remain incomplete for days. Middleware provides a durable orchestration layer that can submit requests in parallel or according to required sequencing, persist correlation and process state, receive asynchronous callbacks, apply retry and error-handling policies, and aggregate all agency outcomes into one final business result. Salesforce can initiate the process without holding an open transaction, while the middleware manages the workflow until it reaches completion. Once all required verifications are available, the middleware uses a Salesforce API call to update the trainer record. This decouples Salesforce from agency-specific latency and protocols and supports reliable recovery if an external service is temporarily unavailable. Salesforce's integration guidance identifies middleware as a useful layer for orchestration, transformation, routing, and handling complex interactions with multiple systems. See [Salesforce Integration Patterns](#) and [Salesforce REST API Developer Guide](#).

QUESTION NO: 27

Northern Trail Outfitters is replacing CometD-based Streaming API consumers with Pub/Sub API consumers for high-volume platform events.

Which two capabilities of Pub/Sub API should the Integration Architect consider?

Choose 2 answers

- A. It uses gRPC and Protocol Buffers for event publishing and subscription.
- B. It supports replaying retained events using replay IDs.
- C. It provides synchronous CRUD operations on Salesforce records.
- D. It eliminates the need for subscribers to handle duplicate events.

ANSWER: A B

Explanation:

Pub/Sub API uses gRPC and Protocol Buffers, which provides an efficient event-consumption interface for high-volume event workloads. Subscribers use the API to receive event messages from Salesforce and acknowledge progress through the subscription protocol.

Pub/Sub API supports replaying retained events using replay IDs. A subscriber can store its processing position and request events from that point after an interruption, provided the requested events remain within Salesforce event retention. The subscriber must still be designed for retry and duplicate-safe processing. See the [Pub/Sub API overview](#) and [Subscribe to Events](#).

QUESTION NO: 28

A US business-to-consumer (B2C) company is planning to expand to Latin America. They project an initial Latin American customer base of about one million, and a growth rate of around 10% every year for the next 5 years. They anticipate privacy and dataprotection requirements similar to those in the European Union to come into effect during this time. Their initial analysis indicates that key personal data is stored in the following systems:

1. Legacy mainframe systems that have remained untouched for years and are due to be decommissioned.
2. Salesforce Commerce Cloud Service Cloud, Marketing Cloud, and Community Cloud.
3. The company's CIO tasked the integration architect with ensuring that they can completely delete their Latin American customer's personal data on demand.

Which three requirements should the integration architect consider?

Choose 3 answers

- A. Manual steps and procedures that may be necessary.
- B. Impact of deleted records on system functionality.
- C. Ability to delete personal data in every system.
- D. Feasibility to restore deleted records when needed.
- E. Ability to provide a 360-degree view of the customer.

ANSWER: B C E

Explanation:

“Impact of deleted records on system functionality,” “Ability to delete personal data in every system,” and “Ability to provide a 360-degree view of the customer” are the key requirements for a complete, scalable deletion capability. A deletion request must be evaluated as an end-to-end business process, not merely as a Salesforce record deletion. The architect needs to understand whether removing a customer’s identifying data affects order history, service cases, marketing suppression, authentication, integrations, referential integrity, reporting, or downstream processes. This enables the organization to design compliant deletion or anonymization behavior while preserving only information that is legitimately required.

The organization also needs a reliable way to locate and act on personal data across every repository, including the legacy mainframe and each Salesforce cloud. A unified, 360-degree customer view is essential to correlate identities and records held under different identifiers, so that a verified request reaches all relevant data stores and no duplicate profile is missed. Given the projected scale and ongoing growth, these capabilities should be designed for repeatable orchestration, auditability, and operational monitoring. Salesforce describes privacy programs and data-subject rights considerations at [Salesforce Privacy](#); its privacy learning material further covers responsible handling of personal data at [Salesforce Trailhead: Data Protection and Privacy](#).

QUESTION NO: 29

Northern Trail Outfitters uses a custom Java application to display code coverage and test results for all of its enterprise applications and plans to include Salesforce as well. Which Salesforce API should an integration architect use to meet the requirement?

- A. Analytics REST API
- B. Metadata API
- C. Tooling API

ANSWER: C

Explanation:

The **Tooling API** is designed for building development tools that need programmatic access to Salesforce's implementation and testing information. A custom Java quality-dashboard application can use its REST resources and SOQL querying capabilities to retrieve Apex test execution data from `ApexTestResult`, including outcomes, error details, run times, and the test queue context. It can also retrieve code-coverage information through objects such as `ApexCodeCoverage` and `ApexCodeCoverageAggregate`. This gives the external application the technical metrics needed to consolidate Salesforce test health and coverage alongside the organization's other enterprise applications.

Tooling API is intended to expose metadata and developer-oriented artifacts in a manner suitable for IDEs, build systems, test automation, and similar SDLC integrations. The Java application can authenticate to Salesforce, execute Tooling API queries, and periodically collect the current test and coverage records for presentation in its centralized interface. Salesforce documents the Tooling API as an API for interacting with Salesforce metadata and developer tooling, and its object reference specifically includes the Apex testing and code-coverage objects required by this use case. See [Tooling API Developer Guide](#) and [Tooling API Object Reference](#).

QUESTION NO: 30

Universal Containers is migrating Accounts and Contacts from an ERP application into Salesforce. The source files contain an ERP customer number for every Account and an ERP customer number on each Contact that identifies its parent Account.

Which two recommendations should an Integration Architect make to preserve the Account-to-Contact relationship during the migration?

Choose 2 answers

- A. Create an External ID field on Account for the ERP customer number.
- B. Use an external ID relationship reference for Contact.AccountId.
- C. Load Contacts first and populate AccountId with the ERP customer number directly.
- D. Create a separate Account record for each Contact during the load.

ANSWER: A B

Explanation:

Create an External ID field on Account for the ERP customer number so Salesforce has a durable, indexed cross-system identifier for each parent record. Accounts can be loaded or upserted using this source-system key.

Use an external ID relationship reference for Contact.AccountId when loading Contacts. The integration can identify the parent Account by its ERP customer number instead of requiring Salesforce Account IDs to be written back to the ERP before the Contact load. This makes the migration more resilient and simplifies repeatable data loads. Salesforce documents parent relationship references using external IDs in the [REST API Relationship Fields documentation](#).

QUESTION NO: 31

A large B2C customer is planning to implement Salesforce CRM to become a Customer centric enterprise. Below, is their current system landscape diagram.

The goals for implementing Salesforce follows:

1. Develop a 360 view of customer
2. Leverage Salesforce capabilities for Marketing, Sales and Service processes
3. 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?

Choose 3 answers

- A. Order Management System
- B. Case Management System
- C. Sales Activity System
- D. Email Marketing System
- E. Quoting System

ANSWER: B C D

Explanation:

Case Management System, Sales Activity System, and Email Marketing System can be retired because their core business capabilities are directly aligned to the stated Salesforce implementation scope: service, sales, and marketing. Salesforce Service Cloud provides centralized case management, allowing agents to create, assign, prioritize, track, and resolve customer issues while maintaining a unified customer record. This supports the objective of creating a 360-degree customer view by connecting service interactions to the same customer data used elsewhere in CRM. Salesforce's case-management capabilities are described in [Salesforce Help: Cases](#).

Sales Cloud supports the tracking and management of sales activities, including tasks, events, customer interactions, opportunities, and related account/contact information. Consolidating sales activities in Salesforce makes those interactions available as part of the shared customer profile. For marketing, Salesforce Marketing Cloud Engagement supports email creation, audience targeting, journey orchestration, delivery, and engagement measurement, enabling the organization to move email-marketing execution into Salesforce. This is documented in [Salesforce Help: Marketing Cloud Engagement overview](#). Retiring these three specialized systems reduces duplicate customer data and helps establish Salesforce as the central platform for marketing, sales, and service engagement.

QUESTION NO: 32

Universal Containers (UC) is a large printing company that sells advertisement banners. The company works with third-party agents on banner initial design concepts. The design files are stored in an on-premise file store that can be accessed by UC internal users and the third party agencies. UC would like to collaborate with the 3rd party agencies on the design files and allow them to be able to view the design files in the community.

The conceptual design files size is 2.5 GB.

Which solution should an integration architect recommend?

- A. Create a lightning component with a Request and Reply integration pattern to allow the community users to download the design files.
- B. Configure Files Connect with an external data source so community users can access the on-premises design files without storing them in Salesforce.
- C. Create a custom object to store the file location URL, when community user clicks on the file URL, redirect the user to the on-prem system file location.
- D. Use Salesforce Files to link the files to Salesforce records and display the record and the files in the community.

ANSWER: B

Explanation:

Configure Files Connect with an external data source so community users can access the on-premises design files without storing them in Salesforce. This approach keeps the 2.5 GB conceptual-design files in the existing file repository while making them discoverable and accessible through Salesforce, including the community experience when the external data source and user permissions are configured appropriately. Files Connect is intended to bridge Salesforce with externally stored files, allowing users to work from Salesforce without requiring the files to be copied into Salesforce Files. That is particularly important here because the files exceed the maximum size supported for an individual Salesforce file upload. The solution also avoids creating a custom download mechanism or exposing unmanaged direct URLs; instead, the integration can use the external repository's authentication and authorization model alongside Salesforce access controls. The architect should ensure that the selected external repository is supported, configure the external data source and Files Connect permissions, and validate that community users have the required access to the external content. Salesforce documents Files Connect as the capability for accessing external file systems from Salesforce: [Files Connect](#). File upload constraints are documented in [Salesforce Files Limits](#).

QUESTION NO: 33

A company has an external system that processes and tracks orders. Sales reps manage their leads and opportunity pipeline in Salesforce. In the current state, the two systems are disconnected and processing orders requires a lot of manual entry on sales rep part. This creates delays in processing orders and incomplete data due to manual entry.

As a part of modernization efforts the company decided to integrate Salesforce and the order management system. The following technical requirements were identified:

1. Orders need to be created in real time from salesforce
2. Minimal customization*, and code should be written due to a tight timeline and lack of developer resources
3. Sales reps need to be able to see order history and be able to see most up to date information on current order status.
4. Managers need to be able to run reports in Salesforce to see daily and monthly order volumes and fulfillment timelines.
5. The legacy system is hosted on premise and is currently connected to the Enterprise Service Bus (ESB). The ESB is flexible enough to provide any methods and connection types needed by salesforce team.
6. There are 1000 sales reps. Each user processes/creates on average 15 orders per shift. Most of the orders contain 20-30 line items.

How should an integration architect integrate the two systems based on the technical requirements and system constraints?

- A. Use Salesforce external object and OData connector.
- B. Use Salesforce custom object, custom REST API and ETL.
- C. Use Salesforce standard object, REST API and ETL.
- D. Use Salesforce big object, SOAP API and Dataloader.

ANSWER: C

Explanation:

Use Salesforce standard object, REST API and ETL. Salesforce's standard Order and OrderItem data model provides an appropriate operational representation for orders and their line items, allowing sales users to access order history and enabling native Salesforce reports and dashboards for volume and fulfillment metrics. Persisting the required order information in Salesforce is important because managers require reporting there, rather than merely viewing remote data.

The ESB can expose a RESTful endpoint to receive a real-time order submission when a sales rep creates an order. This supports prompt order creation while isolating Salesforce from the on-premises system's implementation details. Salesforce's REST API is a resource-oriented interface suited to exchanging standard object records with the ESB and order-management platform. The ESB can also return status changes through the integration layer so Salesforce order records remain current.

ETL is appropriate for initial historical-order loading and for scheduled or bulk reconciliation workloads. It reduces the need to build custom data-loading code while handling the expected order and line-item volume efficiently. This combination uses supported standard capabilities, minimizes customization, and separates real-time operational processing from bulk synchronization. See Salesforce's [Order object reference](#) and [REST API overview](#).

QUESTION NO: 34

A large consumer goods manufacturer operating in multiple countries is planning to implement Salesforce for their Sales and Support operations globally. They have the following security requirements:

1. Internal users from each country have to be authenticated with their local active directory.
2. Customers can create their own login or use Google login.
3. Partners have to be authenticated through a central system which is to be determined.
4. Internal users will have access to the central ERP with their credentials maintained in the ERP.
5. Additional internal systems will be integrated with Salesforce for Sales and Support business processes.

Which three requirements should the integration architect evaluate while designing the integration needs of this project?

Choose 3 answers

- A. Evaluate Salesforce solution for customers and for partners, using thirdparty solution for internal users.
- B. Assess security requirements for internal systems and decide Integration methods that support the requirements.
- C. Evaluate the build of a custom authentication mechanism for users in each country and support for customers and partners.
- D. Consider Third party Single Sign On solution supporting all user authentication including customer and partner.
- E. Evaluate Salesforce native authentication mechanism for all users including customers and partners.

ANSWER: A B D

Explanation:

“Assess security requirements for internal systems and decide Integration methods that support the requirements” is essential because the architecture must account for authentication, authorization, identity propagation, and secure API access across the ERP and the other integrated internal applications. This assessment determines whether the selected interfaces and middleware can meet enterprise security and compliance needs.

“Evaluate Salesforce solution for customers and for partners, using thirdparty solution for internal users” is an appropriate evaluation because these populations have different identity requirements. Salesforce Experience Cloud can support externally facing customer and partner identity experiences, while enterprise identity services can federate internal users from country-specific directory services. The architect should assess the appropriate division of responsibility among Salesforce and the enterprise identity platform.

“Consider Third party Single Sign On solution supporting all user authentication including customer and partner” should also be evaluated as a possible centralized identity architecture. An identity provider or identity broker can federate local directories, support social identity providers such as Google, and provide a centrally governed authentication approach for partner users. Salesforce supports federated authentication using standards such as SAML, allowing an external identity provider to authenticate users before they access Salesforce. See [Salesforce SAML single sign-on documentation](#) and [Salesforce social sign-on documentation](#).