DUMPSARENA

Salesforce Certified Mobile Solutions Architecture Designer

Salesforce Mobile-Solutions-Architecture-Designer

Version Demo

Total Demo Questions: 17

Total Premium Questions: 173

Buy Premium PDF

https://dumpsarena.co

sales@dumpsarena.co

sales@dumpsarena.co
dumpsarena.co

Topic Break Down

Topic	No. of Questions
Topic 1, Mobile Strategy	1
Topic 2, Mobile App Development	14
Topic 3, Mobile Security	24
Topic 4, Mobile Integration	129
Topic 5, Mobile App Distribution	1
Topic 6, Mix Questions	4
Total	173

QUESTION NO: 1

Universal Containers is building a managed package to distribute on the AppExchange. As part of the solution they would like to include authentication information (username/password) inside of the package for web service calls made from the package Universal containers web services. A Salesforce security review has flagged this as a security violation and the architect must decide how best to protect these credentials. Which two methods should the architect consider in order to protect these credentials? Choose 2 answers

A. Utilize named credentials to store the username/password of the web service end post.

B. Utilize a custom object with an encrypted text field to store the username/password of the web service end point.

C. Utilize protected custom settings to store the username/password of the web service end point.

D. Store the username/password directly in the Apex class that will be obfuscated in the managed package.

ANSWER: A C

Explanation:

“Utilize named credentials to store the username/password of the web service end post.” is appropriate because Named Credentials are Salesforce’s purpose-built mechanism for external-callout authentication. They allow Apex callouts to reference a credential configuration rather than embedding a username or password in code. Salesforce securely manages the authentication details and supports endpoint configuration and authentication protocols in a declarative, maintainable model. This reduces the chance that secrets are exposed through source code, logs, package metadata, or changes made during upgrades.

“Utilize protected custom settings to store the username/password of the web service end point.” is also appropriate for a managed-package use case. Protected custom settings can hold package configuration data that is available to the managed package’s code while being shielded from subscriber-org Apex code and subscriber users. This enables the package publisher to retain sensitive integration configuration without exposing the values as ordinary package configuration or hardcoded literals. The package should retrieve the values only when required by its integration logic and should avoid logging them. Salesforce documentation describes Named Credentials as the secure model for authenticated callouts and documents protected settings as a package protection mechanism. See Salesforce Named Credentials documentation and Salesforce packaging component documentation.

QUESTION NO: 2

Universal Containers has chosen Salesforce Wave as their Analytics Platform. There is a requirement to join data from multiple systems(including Salesforce) tobe displayed in a single Wave Lens. What should the Architect recommend?

A. Use an ETL tool to load the data into Salesforce, upserts to ensure that the data in properly joined.

B. Use Data flow to load Salesforce data, and an ETL tool to load other data sets.

C. Use an ETL tool to join multiple sources and load them into a single data set.

D. Use data flow to load Salesforce data,and lightning connect to access the other data sets in real time.

ANSWER: C

Explanation:

Use an ETL tool to join multiple sources and load them into a single data set. A Wave (now CRM Analytics) lens queries a dataset, so data that must be analyzed together should be prepared at a common grain and represented in a dataset that contains the required joined fields. An external ETL process is well suited to extract data from Salesforce and the other source systems, standardize identifiers and formats, perform the cross-system joins, and load the resulting denormalized analytics dataset into CRM Analytics. This approach gives the lens a consistent, analytics-ready source and avoids requiring the lens itself to reconcile data from separate operational systems at query time. The integration process can also be scheduled to refresh the dataset according to the business’s latency requirements, while applying data-quality controls before analytics users consume the information. Salesforce documents datasets as the fundamental data structures used by CRM Analytics for analysis, and its data-integration guidance describes bringing external data into CRM Analytics for

reporting and exploration. See Salesforce CRM Analytics dataset resources and Salesforce CRM Analytics Data Integration Basics.

QUESTION NO: 3

Universal Containers is developing a native mobile application that accesses Salesforce on behalf of individual users. The security team requires that the application must not embed a client secret and must use an OAuth flow designed for public clients. Which two approaches should the Architect recommend? Choose 2 answers

A. Use the OAuth 2.0 authorization code flow with PKCE.

B. Use a connected app configured for mobile OAuth access.

C. Embed the connected app consumer secret in the mobile application.

D. Use the OAuth 2.0 username-password flow from the mobile application.

ANSWER: A B

Explanation:

Use the OAuth 2.0 authorization code flow with PKCE. is appropriate for a native mobile application because PKCE adds a proof key to the authorization-code exchange. The app creates a code verifier and code challenge, and Salesforce validates the verifier when the authorization code is exchanged for tokens. This mitigates authorization-code interception without requiring the mobile application to protect a confidential client secret.

Use a connected app configured for mobile OAuth access. is also required because the connected app defines the mobile application's OAuth configuration, such as its callback URL, permitted scopes, and token policies. The application uses the connected app consumer key as its public client identifier while Salesforce manages the user authentication and authorization process. The username-password flow is not appropriate because it requires the application to collect and handle user credentials directly. Salesforce documents mobile OAuth use in the Mobile SDK OAuth overview and PKCE in the Salesforce OAuth authorization flow documentation.

QUESTION NO: 4

Universal containers has complex data transformation, error handling and process automation requirements as part of their integration strategy. What technology should an Architectrecommend in order to minimize Salesforce code customizations?

A. Data Loader

B. Canvas

C. Process Builder

D. Middleware

ANSWER: D

Explanation:

Middleware is the appropriate recommendation because it provides an integration layer designed to handle complex transformations, orchestration, routing, retries, centralized error handling, monitoring, and process automation outside Salesforce. Rather than implementing these concerns through custom Apex, triggers, or bespoke point-to-point integrations, an integration platform can transform source data into the format required by Salesforce, coordinate calls across multiple systems, and apply consistent failure-handling policies. This minimizes custom code in the Salesforce org and keeps integration logic more maintainable, reusable, and scalable as connected applications evolve.

Salesforce’s integration guidance describes middleware as a way to mediate between systems and support capabilities such as data transformation, protocol conversion, routing, and orchestration. This pattern is especially valuable when requirements extend beyond a simple data import or a single synchronous API call. A middleware solution can also decouple Salesforce from individual back-end systems, reducing the impact of changes in either system and allowing operations

teams to manage integration errors without requiring Salesforce deployments. See Salesforce's Integration Patterns guidance and MuleSoft's overview of middleware for further details.

QUESTION NO: 5

Universal Containers manages a catalog of overone million products that it makes available to its customers. The master product catalog is stored and managed in their ERP application with frequent updates made to the product catalog by their sourcing team. The sourcing team may update attributes suchas price, general catalog availability, and the product description. When the sourcing team makes an update that change must go into effect during the next business day and there may be thousands of changes made over the course of the day. What integration pattern would you recommend to best manage this scenario?

A. Write a custom web service to accept product catalog changes from ERP.

B. Use the streaming API to receive product changes in real time from ERP.

C. Write an outbound message to send productchanges in real time from ERP.

D. Build a scheduled ETL job to sync products on a nightly basis from ERP.

ANSWER: D

Explanation:

Build a scheduled ETL job to sync products on a nightly basis from ERP. The catalog's system of record is the ERP application, and the stated service-level requirement is that changes become effective by the next business day rather than immediately. A scheduled batch integration is therefore an appropriate fit: it can collect the day's changed product records, transform and validate attribute values, and load the resulting delta into Salesforce during an agreed integration window. This approach is especially suitable for a catalog exceeding one million products and potentially thousands of daily updates because it is designed for high-volume processing without requiring a separate transaction for every edit. The ETL process should use a stable external identifier from the ERP and perform upserts, so it creates new products when needed and updates existing products reliably. It should also record processing outcomes, support retries for failed records, and reconcile the source and target counts after each run. Salesforce's bulk data-processing guidance supports asynchronous, high-volume loading, while its integration-pattern guidance identifies batch synchronization as the appropriate style when data does not require real-time propagation. See Salesforce Bulk API 2.0 Developer Guide and Salesforce Integration Patterns.

QUESTION NO: 6

Universal Containers would like to integrate Salesforce to their Accounting system. Salesforce must notify the accounting system for every new account that has been created. The security team will not allow Salesforce to integrate directly to the accounting system due to potential security issues. Which three stages should the Architect consider to reduce the security concerns for this Integration use case? Choose 3 answers

A. Terminate the SSL connection at a reverse proxy in the DMZ which establishes trust in the connection using mutual SSL.

B. Enable WS-security for the web services made between Salesforce and the accounting system.

C. Whitelist the Salesforce IP range on the firewall to ensure only Salesforce-originated traffic can penetrate the network.

D. Utilize an Enterprise Service Bus to ensure Accounting system credentials are not stored within Salesforce.

E. Enable platform encryption in the Salesforce org to ensure network communication to the Accounting system is encrypted.

ANSWER: A C D

Explanation:

"Terminate the SSL connection at a reverse proxy in the DMZ which establishes trust in the connection using mutual SSL," "Whitelist the Salesforce IP range on the firewall to ensure only Salesforce-originated traffic can penetrate the network," and

“Utilize an Enterprise Service Bus to ensure Accounting system credentials are not stored within Salesforce” provide complementary controls for a securely segmented integration. A reverse proxy located in a demilitarized zone creates a controlled boundary between the public cloud and internal services. Mutual TLS allows both endpoints to authenticate through certificates, rather than relying only on a server certificate. Firewall allowlisting further restricts inbound access at the network boundary to Salesforce-published outbound address ranges, reducing the possible sources that can reach the proxy. An enterprise service bus or other middleware layer keeps the accounting system off the direct Salesforce connection path and centralizes routing, transformation, credential storage, and access controls. Salesforce recommends using middleware when integration requirements call for mediation or when direct connectivity is not appropriate. Salesforce publishes the IP ranges that customers can use for network allowlisting, though these ranges must be maintained as Salesforce updates them. See Salesforce Network Access documentation and Salesforce Architect guidance on middleware.

QUESTION NO: 7

Universal Containers has a SOAP-based integration that runs nightly to update the Product(Product2) object in Salesforce with updated product availability for over 500,000 products. The source system is a green-screen ERP that must be takenoffline to produce nightly production reports, such as the inventory availability report used for this integration. The integration is performing very slowly and does not complete within the allocated four-hour time slot. What three recommendations might aTechnical Architect make to resolve this issue? Choose 3 answers

A. Use outbound Messaging to notify Salesforce promptly when productavailability changes in the source system.

B. Store Salesforce Product IDs in the source system to use updates rather than External ID-based UPSERT calls.

C. Pre-process the data to avoid the need for workflow rules or triggers

D. Use Bulk API to update or upsert records more efficiently.

E. Contact Salesforce support to request that they turn off record locking on theProduct2 object.

ANSWER: B C D

Explanation:

Storing Salesforce Product IDs in the source system allows the integration to target existing Product2 records directly. This avoids repeatedly resolving external identifiers and enables an update-oriented load pattern, which is appropriate when the process is refreshing availability for products already represented in Salesforce. Pre-processing the extract before it is sent to Salesforce reduces the amount of work performed during the limited ERP reporting window. For example, the integration can remove unchanged rows, validate values, consolidate duplicate updates, and prepare data so that unnecessary downstream automation is not invoked. This is especially important at a volume exceeding 500,000 records, because trigger and workflow processing is performed in addition to the database update itself. Finally, Bulk API is designed for asynchronous, high-volume data loads and processes records in batches rather than through individual SOAP requests. Using Bulk API for updates, or upserts where an external key is genuinely required, substantially reduces API-request overhead and provides a scalable mechanism for monitoring batch results and retrying failed records. Salesforce documents Bulk API as its asynchronous interface for loading or deleting large data sets in Bulk API Developer Guide. Its guidance on bulk processing also explains why automation must be designed to efficiently handle collections of records; see Bulk Apex Trigger Best Practices.

QUESTION NO: 8

Universal Containers has a SOAP-based integration that runs nightly to update the Product(Product2) object in Salesforce with updated product availability for over 500,000 products. The source system is a green-screen ERP that must be takenoffline to produce nightly production reports, such as the inventory availability report used for this integration. The integration is performing very slowly and does not complete within the allocated four-hour time slot. What three recommendations might aTechnical Architect make to resolve this issue? Choose 3 answers

A. Use outbound Messaging to notify Salesforce promptly when product availability changes in the source system.

B. Store the Salesforce Product ID in the source system to eliminate the needfor External IDs and UPSERT API calls.

C. Pre-process the data to avoid the need for workflow rules or triggers

D. Use the Bulk API UPDATE or UPSERT records more efficiently.

E. Contact Salesforce support to request that they turn off record locking on theProduct2 object.

ANSWER: B C D

Explanation:

For a nightly load of more than 500,000 existing Product2 records, storing the Salesforce Product ID in the source system allows the integration to send direct update operations keyed by Salesforce record ID. This avoids the matching work associated with external-ID upserts and makes the source-to-target relationship deterministic. The integration should also use the Bulk API UPDATE or UPSERT records more efficiently. Bulk API is designed for high-volume asynchronous data processing, accepts large batches of records, and is substantially better suited than making large numbers of SOAP API calls within a constrained overnight processing window. When Salesforce IDs are stored, UPDATE is the preferred bulk operation for known existing products. Finally, pre-processing the data to avoid the need for workflow rules or triggers reduces per-record automation overhead during the load. Calculating availability changes, validating source data, and excluding unchanged products before submission can greatly reduce both the number of records processed and the work performed in each transaction. Together, direct ID-based updates, asynchronous bulk processing, and minimizing automation create a scalable approach that helps the job finish while the ERP reporting extract is available. Salesforce documents Bulk API as its high-volume asynchronous data-loading interface and provides additional guidance on load performance and automation impact. See Salesforce Bulk API Developer Guide and Salesforce data load performance guidance.

QUESTION NO: 9

Universal containers usesa legacy system to receive and handle Level 1 service requests, and Salesforce service Cloud for Level2 requests and above, Cases will be pushed from the legacy system to Service Cloud by a nightly batch process. Once the cases are closed in SF, the caseneeds to be updated in the legacy system as soon as possible. How should the Technical Architect recommend that case status be updated in the legacy system?

A. Use Apex callout to send case status from Salesforce to the legacy system.

B. use Outbound messages to send status updates from Salesforce to the legacy system.

C. Use a middleware tool to pull case status from Salesforce and push to the legacy system at regular intervals.

D. Write an Apex web service returning case status, to be called from the legacy system.

ANSWER: B

Explanation:

use Outbound messages to send status updates from Salesforce to the legacy system. is the appropriate recommendation because an outbound message can be configured in a workflow rule to fire when a Case reaches a closed status. Salesforce then sends a SOAP notification containing the selected Case fields to a listener endpoint exposed by the legacy system. This supports the requirement to notify the legacy system as soon as the Case is closed, rather than waiting for another scheduled extract or polling interval.

Outbound Messaging is also designed for reliable asynchronous delivery. If the receiving endpoint is temporarily unavailable or does not acknowledge receipt successfully, Salesforce retries delivery according to its retry mechanism. The legacy application can use the Case ID and status included in the message to locate and update its corresponding service request, and it should make its processing idempotent because notifications can be retried. This declarative integration pattern is particularly suitable when Salesforce is notifying an external SOAP-capable system about a record change without requiring custom Apex integration code. See Salesforce's Outbound Messaging documentation and Outbound Messaging setup guidance.

QUESTION NO: 10

Universal Containers has a trigger on the Order object to update the parentAcount with the date and time of the last closed Opportunity. An integration that inserts orders for the high-volume customers is failing periodically, with no obvious pattern to the timing of failures. What could be the cause of this issue?

A. The trigger is failing Unit Tests that access the new data.

B. API limits being limited.

C. Data skew is causing record locking issues on the Oder Share object.

D. Record locking contention on the parent Account.

ANSWER: D

Explanation:

Record locking contention on the parent Account. is the likely cause because each Order insert invokes automation that updates its parent Account. Updating an Account requires Salesforce to obtain a lock on that Account record for the duration of the transaction. When an integration inserts many Orders concurrently for high-volume customers, multiple transactions can attempt to update the same parent Account at nearly the same time. One transaction obtains the lock, while another may wait; if it cannot obtain the lock before Salesforce's lock timeout, the operation fails with a locking error such as `UNABLE_TO_LOCK_ROW`.

This behavior can appear intermittent because it depends on transaction timing, batch concurrency, and how long the transaction holding the Account lock takes to complete. It is especially common when many child records share a single parent, which concentrates updates and locking activity on that parent. Salesforce recommends reducing concurrent updates to shared parent records, keeping transactions efficient, and designing integrations to retry transient lock failures with appropriate backoff. See Salesforce's Record Locking and Concurrency guidance and its Batch Apex documentation for transaction and batch-processing considerations.

QUESTION NO: 11

Universal containers(UC) leverages the standard opportunity and opportunity product objects tomanage their orders in Salesforce. When a deal is closed, all opportunity information, including products and billing contacts, must be send to their ERP application for order fulfillment. As UC has an "express shipping" guarantee, leadership would like order information sent to ERP as quickly as possible after the deal is closed? How should an Architect fulfill this requirement?

A. Write a nightly batch job to send customer information to ERP.

B. Write a visualforce page to send order information to ERP.

C. Write an Opportunity trigger to send order information to ERP.

D. Write an outbound message to send order information to ERP.

ANSWER: C

Explanation:

Write an Opportunity trigger to send order information to ERP. An Opportunity trigger can detect the transaction in which a deal changes to Closed Won and initiate near-real-time integration processing immediately after that business event. The implementation can query the related Opportunity Products and billing-contact data, construct the complete order payload required by the ERP, and enqueue asynchronous processing for the outbound callout. This meets the express-shipping objective because processing begins as part of the close event rather than waiting for a scheduled interval or a user-driven interaction.

Salesforce recommends using asynchronous Apex, such as Queueable Apex, for callouts initiated from trigger-based business logic. The trigger should remain bulk-safe: it should identify all newly closed opportunities in the transaction, pass their IDs to a queueable job, and let that job retrieve related data and make the ERP callout. This design supports scalable processing while keeping the Opportunity save transaction responsive. Queueable jobs that implement `Database.AllowsCallouts` are specifically designed to perform external HTTP callouts asynchronously. See Salesforce's Queueable Apex documentation and Apex Triggers documentation.

QUESTION NO: 12

Universal Containers sells its products online using a system built on Force.com sites. The orders are captures and processed in Salesforce. the company uses an external marketing system and would like to make use of the customer data captured in Salesforce.The marketing system has REST API that can be used to push data into it. Which three options should a Technical Architect consider that do not require building custom web services on the marketing system? Choose 3 answers

A. Write a custom Apex web service, Which will be called from the marketing system to retrieve customer data.

B. Use Apex callout to send customer data from Salesforce to the marketing system

C. Use a middleware tool to pull customer data from Salesforce and push it to the marketing system on adaily basis.

D. Builda custom java application using the Enterprise WSDL to pull data from Salesforce and push it to the marketing system.

E. Use outbound messages to send customer data from Salesforce to the marketing system.

ANSWER: B C D

Explanation:

Use Apex callout to send customer data from Salesforce to the marketing system is appropriate because Salesforce can invoke the marketing platform's existing REST endpoints directly. An Apex integration can serialize the relevant customer information, authenticate to the external service with a named credential, and submit it through an HTTP request without requiring the marketing platform to expose any additional custom service.

Use a middleware tool to pull customer data from Salesforce and push it to the marketing system on adaily basis is also a valid integration pattern. Middleware can authenticate to Salesforce through its APIs, extract the required customer records, transform or map fields if necessary, and invoke the marketing system's supplied REST API. Scheduled or batch synchronization is especially suitable when immediate propagation is not a business requirement.

Builda custom java application using the Enterprise WSDL to pull data from Salesforce and push it to the marketing system is another viable option. A Java integration application can use Salesforce's SOAP-based Enterprise API to query Salesforce data and then act as a REST client to submit that data to the marketing system's existing API. In each case, Salesforce or the integration layer calls an already available marketing REST endpoint, rather than requiring a custom web-service endpoint on that system. See Salesforce Apex Callouts and Salesforce API Quick Start.

QUESTION NO: 13

Universal Containers has a batch integration that runs every five minutes toload Shipment records related to existing orders that have been updated in the previous five minutes. the integration is not reporting any errors, but some Shipment records are not being loaded. What could be the problem?

A. Error reporting is notenabledin Salesforce.

B. The integration takes more than five minutes to run.

C. The integration is causing UC to exceedits API limits

D. The Integration cannot find the parent orders for some Shipments.

ANSWER: B

Explanation:

The integration takes more than five minutes to run. A polling integration that selects records updated only within the prior five-minute interval depends on its processing cycle completing within that same interval. If a run lasts longer than five minutes, the next query can begin after part of the relevant update period has already elapsed. Shipment-related updates that occur during the uncovered interval can then fall outside the next run's fixed five-minute `LastModifiedDate` query window, resulting in records that are silently missed even though the integration itself reports no errors.

A resilient incremental-load design uses a stored high-water mark, such as the timestamp of the last successfully processed record, rather than assuming a scheduled interval equals the actual processing duration. It should also use a small overlapping time window and deduplicate records by Salesforce ID, allowing the process to safely re-read boundary records while preventing gaps caused by slow executions, delays, retries, or clock differences. Salesforce documents `SystemModstamp` and `LastModifiedDate` as fields that can support record-change queries, and recommends considering selectivity and indexing when querying large data volumes. See Salesforce System Fields and Salesforce REST API Resources.

QUESTION NO: 14

Universal Containers sells its products online using a system built on Force.com sites. The orders are captures and processed in Salesforce. the company uses an external marketing system and would like to make use of the customer data captured in Salesforce.The marketing system has REST API that can be used to push data into it. Which three options should a Technical Architect consider that do not require building custom web services on the marketing system? Choose 3 answers

A. Write a custom Apex web service, Which will be called from the marketing system to retrieve customer data.

B. Use Apex callout to send customer data from Salesforce to the marketing system

C. Use a middleware tool to pull customer data from Salesforce and push it to the marketing system on adaily basis.

D. Build a custom java application using the Enterprise WSDL to pull data from Salesforce and push it to the marketing system.

E. Use outbound messages to send customer data from Salesforce to the marketing system.

ANSWER: B C D

Explanation:

“Use Apex callout to send customer data from Salesforce to the marketing system” is appropriate because Salesforce can act as an HTTP client and invoke the marketing platform’s existing REST endpoints. The integration can be implemented with named credentials for endpoint and authentication management, then triggered synchronously or asynchronously according to the order-processing design. Salesforce documents the supported HTTP request and response patterns for Apex callouts at Apex Developer Guide: Callouts.

“Use a middleware tool to pull customer data from Salesforce and push it to the marketing system on adaily basis.” is also suitable. Middleware can authenticate to Salesforce, query the required customer and order data through Salesforce APIs, transform or batch it if needed, and submit it to the marketing system’s REST API. This provides a decoupled scheduled synchronization model without requiring a service hosted by the marketing system.

“Build a custom java application using the Enterprise WSDL to pull data from Salesforce and push it to the marketing system.” likewise uses a client-side integration application. The Java application can consume Salesforce SOAP API operations generated from the Enterprise WSDL, retrieve the needed records, and make outbound REST requests to the marketing platform. Salesforce’s API documentation describes the Enterprise WSDL and SOAP API integration model at Generate a WSDL File for Your Org.

QUESTION NO: 15

Which two approaches should an Integration Architect recommend to allow access to on-premise systems by Salesforce? Choose 2 answers

A. Place the systems in aDMZ.

B. Whitelist Salesforce IPs on the firewall.

C. Utilize two-way(mutual) SSL

D. Whitelist the corporate IPS in Salesforce.

ANSWER: B C

Explanation:

Whitelist Salesforce IPs on the firewall is appropriate because an on-premises firewall can restrict inbound integration traffic to Salesforce's documented outbound IP address ranges. This permits Salesforce callouts to reach the required endpoint while maintaining a network control that blocks untrusted sources. The architect should use the published ranges applicable to the organization's Salesforce infrastructure and maintain the allowlist as Salesforce documentation changes.

Utilize two-way(mutual) SSL adds strong endpoint authentication to the connection. With mutual TLS, the on-premises service validates a client certificate presented by Salesforce, rather than relying only on network location or a shared secret. Salesforce can use a certificate configured for the outbound authentication flow, and the service can trust the issuing certificate authority or the specific client certificate. Combining an IP allowlist with mutual TLS provides layered protection: the firewall limits which network sources can connect, while certificate-based authentication verifies that the caller is an authorized Salesforce integration. This is especially useful when exposing a narrowly scoped API endpoint for Salesforce callouts to internal services. See Salesforce guidance on Salesforce IP address ranges and the developer documentation for mutual authentication with certificates.

QUESTION NO: 16

Universal Containers is hearing complaints from users that recently released changes arebreaking existing functionality. What type of testing program should a Technical Architect implement to reduce or eliminate this complaint?

A. Performance Testing

B. Unit TestingC .Regression Testing

C. User Acceptance Testing

D. Regression Testing

ANSWER: D

Explanation:

Regression Testing is the appropriate program because its purpose is to verify that previously working functionality continues to work after a change, such as a new feature, configuration update, integration change, or platform release. A Technical Architect should establish a repeatable regression suite covering the organization's critical business processes, core user journeys, integrations, security-sensitive actions, and high-risk customizations. The suite should be executed before releases and after relevant changes in lower environments, with results reviewed as a release-quality gate. Automated tests are especially valuable because they can be run consistently and frequently, while targeted manual regression scripts can cover end-to-end processes that are difficult to automate. Salesforce's testing guidance emphasizes using tests to validate expected behavior and protect application quality as code changes. A well-maintained regression program therefore directly detects unintended side effects before users encounter them in production. See the Salesforce Apex Testing documentation and the Salesforce Trailhead module on Apex testing.

QUESTION NO: 17

Universal Containers wishes to move data between 3 back office systems: ERP, financial and a legacy home-grown shipping system that will be replaced 3 months after integration is scheduled to be complete. What integration pattern should an architect recommend to ensure minimal throwaway code?

A. point-to-point

B. Web Mashup

C. MiddleWare

D. Apex batch Processing

ANSWER: C

Explanation:

MiddleWare is the appropriate integration pattern because it decouples the ERP and financial systems from the short-lived, home-grown shipping application. A middleware layer can expose stable, canonical APIs or messages to the participating applications while it handles system-specific transformations, routing, protocol differences, orchestration, and error handling. When the shipping system is replaced, the integration team primarily updates or replaces that system's middleware adapter and mappings. The contracts used by the ERP and financial systems can remain stable, which substantially limits the code and configuration that must be discarded or rewritten.

This is particularly valuable when an endpoint has a known, near-term replacement date. The initial effort produces reusable integration capabilities rather than embedding shipping-system assumptions throughout direct integrations. Salesforce integration guidance describes middleware as a way to centralize integration logic and connect Salesforce and external systems through reusable services. Its architecture guidance also emphasizes using API-led, loosely coupled integrations to isolate changes and improve maintainability over time. See Salesforce Integration Patterns and Salesforce API Basics.