DUMPSARENA

Salesforce Certified MuleSoft Platform Architect 1

Salesforce MuleSoft-Platform-Architect-I

Version Demo

Total Demo Questions: 17

Total Premium Questions: 177

Buy Premium PDF

https://dumpsarena.co

sales@dumpsarena.co

sales@dumpsarena.co
dumpsarena.co

Topic Break Down

Topic	No. of Questions
Topic 1, Anypoint Platform Architecture	45
Topic 2, Application Network	39
Topic 3, Integration Architecture	28
Topic 4, Security	28
Topic 5, Operations	37
Total	177

QUESTION NO: 1

4 Production environment is running on a dedicated Virtual Private Cloud (VPC) on CloudHub 1,0, and the security team guidelines clearly state no traffic on HTTP.

Which two options support these security guidelines?

Choose 2 answers

A. Configure the HTTPS protocol in HTTP listener in the Mule application

B. Create a custom policy to apply to outgoing and incoming HTTP requests to control access to a configured API endpoint

C. Remove the HTTP (port 8081) entry from the VPC firewall rules.

D. Configure the IP Blocklist policy to control access to a configured API endpoint from either a single IP address or a range of IP addresses.

E. Add an HTTP (port 8081) entry to the VPC firewall rules.

ANSWER: A C

Explanation:

Configuring the HTTPS protocol in HTTP listener in the Mule application ensures that the application listener is configured for TLS-protected communication. HTTPS encrypts the request and response in transit and authenticates the endpoint through its certificate, satisfying the requirement that application traffic use a secure HTTP transport. The listener must be configured with the appropriate TLS context, including a keystore and certificate, so that clients connect using HTTPS rather than unencrypted HTTP.

Removing the HTTP (port 8081) entry from the VPC firewall rules provides network-level enforcement of the same requirement. CloudHub VPC firewall rules control which inbound TCP ports can reach application workers. By not allowing the application's HTTP port, the VPC prevents inbound HTTP connections from reaching the worker, while HTTPS traffic can be permitted on the required secure listener port. Using HTTPS at the listener together with a restrictive VPC firewall creates defense in depth: the application is configured to serve secure traffic and the network perimeter does not expose the insecure HTTP path.

See MuleSoft's HTTP Listener configuration reference and VPC firewall rules documentation.

QUESTION NO: 2

An organization requires several APIs to be secured with OAuth 2.0, and PingFederate has been identified as the identity provider for API client authorization, The

PingFederate Client Provider is configured in access management, and the PingFederate OAuth 2.0 Token Enforcement policy is configured for the API instances required by the

organization. The API instances reside in two business groups (Group A and Group B) within the Master Organization (Master Org).

What should be done to allow API consumers to access the API instances?

A. The API administrator should configure the correct client discovery URL in both child business groups, and the API consumer should request access to the API in Ping Identity

B. The API administrator should grant access to the API consumers by creating contracts in the relevant API instances in API Manager

C. The APL consumer should create a client application and request access to the APT in Anypoint Exchange, and the API administrator should approve the request

D. The APT consumer should create a client application and request access to the API in Ping Identity, and the organization's Ping Identity workflow will grant access

ANSWER: C

Explanation:

The API consumer should create a client application and request access to the API in Anypoint Exchange, and the API administrator should approve the request. This follows Anypoint Platform's API access model: a consumer application is created in Anypoint Platform, then a request for API access creates a client-application contract for the relevant API instance. The API owner or administrator approves that request, granting the application the authorized relationship needed to call the managed API.

With PingFederate configured as the client provider and the PingFederate OAuth 2.0 Token Enforcement policy applied, PingFederate supplies the OAuth 2.0 authorization and token capabilities for the client application. Anypoint Exchange remains the consumer-facing place to discover an API, create or select an application, and request access. The approval process ensures that API access is governed for each API instance, including instances located in different business groups. After approval, the consumer can obtain and present the required OAuth token when invoking the API according to the configured PingFederate integration.

See MuleSoft documentation on requesting access to an API from Exchange and configuring PingFederate as a client provider.

QUESTION NO: 3

What is a best practice when building System APIs?

A. Document the API using an easily consumable asset like a RAML definition

B. Model all API resources and methods to closely mimic the operations of the backend system

C. Build an Enterprise Data Model (Canonical Data Model) for each backend system and apply it to System APIs

D. Expose to API clients all technical details of the API implementation's interaction wifch the backend system

ANSWER: B

Explanation:

Model all API resources and methods to closely mimic the operations of the backend system is the best practice because a System API is the reusable, stable interface to a specific system of record or backend application. In MuleSoft's API-led connectivity approach, this layer encapsulates the backend system's connectivity and exposes its core data and operations in a form that downstream APIs can reuse. Designing resources and methods around the backend's meaningful capabilities—for example, retrieving customer records, creating an order, or updating an account—makes the API intuitive for teams that need to access that system and keeps backend-specific integration logic contained in one place. This approach also creates a clear separation of concerns: System APIs provide access to underlying systems, while Process APIs orchestrate and transform data across systems, and Experience APIs tailor data for a particular channel or consumer. A well-defined System API therefore makes backend access easier to govern, reuse, secure, and evolve without requiring every consuming application to create its own direct integration. MuleSoft describes these responsibilities in its API-led connectivity guidance and learning material: MuleSoft API-led connectivity documentation and Salesforce Trailhead: API-led connectivity.

QUESTION NO: 4

A team is planning to enhance an Experience API specification, and they are following API-led connectivity design principles.

What is their motivation for enhancing the API?

A. The primary API consumer wants certain kinds of endpoints changed from the Center for Enablement standard to the consumer system standard

B. The underlying System API is updated to provide more detailed data for several heavily used resources

C. An IP Allowlist policy is being added to the API instances in the Development and Staging environments

D. A Canonical Data Model is being adopted that impacts several types of data included in the API

ANSWER: A

Explanation:

The primary API consumer wants certain kinds of endpoints changed from the Center for Enablement standard to the consumer system standard is the appropriate motivation for enhancing an Experience API. In API-led connectivity, the Experience layer is intentionally consumer-oriented: it exposes data and capabilities in a form that best supports the needs, terminology, interaction patterns, and user experience of a particular consuming application or channel. This layer is the appropriate place to tailor an API contract without forcing every consumer to adopt the same representation or exposing implementation details from lower layers.

Consequently, when the consuming system has a justified need for differently shaped or named endpoints, the team can evolve the Experience API specification to provide that consumer-appropriate interface. The change should still be governed through the organization's API lifecycle practices, including contract review, versioning where needed, and clear documentation, so existing consumers have a predictable migration path. This preserves the separation of concerns that API-led connectivity is intended to provide: consumer-facing contracts can evolve independently while reusable lower-layer APIs remain decoupled from a particular channel's requirements. MuleSoft describes Experience APIs as APIs designed around the needs of a specific consumer or experience in its API-led connectivity documentation. See also the API-led connectivity architecture learning module.

QUESTION NO: 5

Once an API Implementation is ready and the API is registered on API Manager, who should request the access to the API on Anypoint Exchange?

A. None

B. Both

C. API Client

D. API Consumer

ANSWER: D

Explanation:

API Consumer is correct because the consumer represents the person, team, or application-owning organization that intends to use an API product. In Anypoint Exchange, a consumer requests access to a managed API asset, typically by creating a client application and requesting the appropriate Service Level Agreement (SLA) tier. After the API owner approves the request when approval is required, Anypoint Platform provisions client credentials, such as a client ID and client secret. The consuming application then uses those credentials when it calls the protected API.

This distinction supports API governance and traceability: API owners can identify who is using an API, apply an SLA and associated policies, and manage or revoke the consumer application's credentials independently of the code that makes requests. An API client is the technical software component that sends requests, but access registration and credential lifecycle management are performed in the context of the API consumer's client application in Exchange. MuleSoft documents the access-request workflow and client-application credential management in its Request Access to an API guide and explains how client applications obtain credentials in Client Applications.

QUESTION NO: 6

A retail company is using an Order API to accept new orders. The Order API uses a JMS queue to submit orders to a backend order management service. The normal load for orders is being handled using two (2) CloudHub workers, each configured with 0.2 vCore. The CPU load of each CloudHub worker normally runs well below 70%. However, several times during the year the Order API gets four times (4x) the average number of orders. This causes the CloudHub worker CPU load to exceed 90% and the order submission time to exceed 30 seconds. The cause, however, is NOT the backend order management service, which still responds fast enough to meet the response SLA for the Order API. What is the MOST

resource-efficient way to configure the Mule application's CloudHub deployment to help the company cope with this performance challenge?

A. Permanently increase the size of each of the two (2) CloudHub workers by at least four times (4x) to one (1) vCore

B. Use a vertical CloudHub autoscaling policy that triggers on CPU utilization greater than 70%

C. Permanently increase the number of CloudHub workers by four times (4x) to eight (8) CloudHub workers

D. Use a horizontal CloudHub autoscaling policy that triggers on CPU utilization greater than 70%

ANSWER: D

Explanation:

Use a horizontal CloudHub autoscaling policy that triggers on CPU utilization greater than 70% because the performance issue is intermittent, originates in the deployed application tier, and is indicated directly by sustained worker CPU pressure. During ordinary demand, the existing two small workers have ample capacity. Adding workers only when CPU rises above the selected threshold supplies additional application instances to receive load-balanced requests and submit orders concurrently to the JMS queue. This increases request-processing throughput while avoiding the recurring cost of permanently provisioning capacity sized for infrequent fourfold peaks.

Horizontal scaling is especially appropriate where the downstream order-management service is not the bottleneck: additional application workers can improve the rate at which incoming orders are accepted and queued without overwhelming a slow dependency identified in the scenario. The application must remain suitable for multiple replicas, including avoiding reliance on worker-local state and ensuring any shared resources are handled safely. MuleSoft documents worker sizing and scaling behavior for CloudHub deployments, including configuration of replica counts and autoscaling in supported deployment models. See CloudHub architecture and Deploying applications to CloudHub.

QUESTION NO: 7

What best explains the use of auto-discovery in API implementations?

A. It makes API Manager aware of API implementations and hence enables it to enforce policies

B. It enables Anypoint Studio to discover API definitions configured in Anypoint Platform

C. It enables Anypoint Exchange to discover assets and makes them available for reuse

D. It enables Anypoint Analytics to gain insight into the usage of APIs

ANSWER: A

Explanation:

It makes API Manager aware of API implementations and hence enables it to enforce policies. Auto-discovery links a deployed Mule application to a specific API instance that is managed in API Manager. A developer configures the Mule application with the API's unique autodiscovery identifier; when the application starts, the Mule runtime registers the implementation with Anypoint Platform. This association lets API Manager apply and manage policies for the running API implementation, such as client ID enforcement, rate limiting, IP allowlists, and authentication-related controls. It also establishes the management connection needed for API governance and operational visibility of that implementation. Auto-discovery is therefore a core mechanism for centrally managing an API implementation without embedding each API Manager policy configuration directly in the application's source code. The application must be deployed to a compatible Mule runtime and have the required connectivity and permissions to communicate with Anypoint Platform. For implementation details and prerequisites, see MuleSoft's About API Autodiscovery and Configure API Autodiscovery documentation.

QUESTION NO: 8

An IT Security Compliance Auditor is assessing which nonfunctional requirements (NFRs) are already being implemented to meet security measures.

* The Web API has Rate-Limiting SLA

* Basic Authentication - LDAP

* JSON Threat Protection

* TP Allowlist policies applied

Which two NFRs-are enforced?

A. The API invocations are coming from a known subnet range

B. Username/password supported to validate login credentials

C. Sensitive data is masked to prevent compromising critical information

D. The API is protected against XML invocation attacks

E. Performance expectations are to be allowed up to 1,000 requests per second

ANSWER: A B

Explanation:

The API invocations are coming from a known subnet range is enforced by the applied IP allowlist policy. An IP allowlist establishes a network-access control at the API gateway boundary: requests are accepted only when their source address matches configured trusted IP addresses, CIDR ranges, or subnets. This directly implements the stated security requirement that API traffic originate from a known network range, reducing exposure to requests from unapproved locations.

Username/password supported to validate login credentials is enforced by Basic Authentication configured with LDAP. Basic authentication requires a client to provide credentials, while LDAP provides the directory against which those credentials are validated. Together, this implements identity authentication for callers before access to the protected API is granted. It is an authentication control, distinct from the network-origin restriction delivered by the allowlist.

These two controls are explicitly supported by the described policy set and represent independently enforceable security NFRs: trusted source-network access and directory-backed credential validation. MuleSoft documents the available API gateway policies, including authentication and traffic/security controls, in its included policies reference. Configuration and behavior of the network restriction are documented in the IP allowlist policy documentation.

QUESTION NO: 9

A retail company with thousands of stores has an API to receive data about purchases and insert it into a single database. Each individual store sends a batch of purchase data to the API about every 30 minutes. The API implementation uses a database bulk insert command to submit all the purchase data to a database using a custom JDBC driver provided by a data analytics solution provider. The API implementation is deployed to a single CloudHub worker. The JDBC driver processes the data into a set of several temporary disk files on the CloudHub worker, and then the data is sent to an analytics engine using a proprietary protocol. This process usually takes less than a few minutes. Sometimes a request fails. In this case, the logs show a message from the JDBC driver indicating an out-of-file-space message. When the request is resubmitted, it is successful. What is the best way to try to resolve this throughput issue?

A. se a CloudHub autoscaling policy to add CloudHub workers

B. Use a CloudHub autoscaling policy to increase the size of the CloudHub worker

C. Increase the size of the CloudHub worker(s)

D. Increase the number of CloudHub workers

ANSWER: C

Explanation:

Increasing the size of the CloudHub worker(s) is the appropriate first action because the observed failure is explicitly an out-of-file-space condition generated while the JDBC driver creates temporary files on the worker's local disk. A larger worker provides greater allocated resources, including more available filesystem capacity, allowing a bulk operation to retain its required temporary data through completion. The fact that a resubmission succeeds is consistent with temporary files from prior processing having been released, so the operation later has enough free local storage. The application is currently deployed to one worker, and the immediate limiting resource is the local disk space used during the driver's staging process. Worker sizing should therefore be reviewed alongside the largest expected batch size and the possibility of overlapping store uploads, because concurrent bulk requests can increase temporary-storage demand. MuleSoft documents CloudHub worker resource allocation and deployment sizing in its CloudHub architecture documentation. After resizing, monitor worker-level resource use and application logs to validate that free disk capacity is sufficient during peak ingestion windows. See CloudHub architecture and Deploying Applications to CloudHub.

QUESTION NO: 10

4 Production environment is running on a dedicated Virtual Private Cloud (VPC) on CloudHub 1,0, and the security team guidelines clearly state no traffic on HTTP.

Which two options support these security guidelines?

Choose 2 answers

A. Configure the HTTPS protocol in HTTP listener in the Mule application

B. Create a custom policy to apply to outgoing and incoming HTTP requests to control access to a configured API endpoint

C. Remove the entry from the VPC firewall rule

D. Configure the IP Blocklist policy to control access to a configured API endpoint from either a single IP address or a range of IP addresses.

E. Add the entry in the VPC firewall rule.

ANSWER: A C

Explanation:

Configuring the HTTPS protocol in the Mule application's HTTP Listener ensures that the application endpoint accepts TLS-protected HTTPS connections. TLS encrypts data in transit and provides server authentication, which meets the requirement that production traffic must not use clear-text HTTP. The listener must be configured with an appropriate TLS context, including the certificate and key material needed to establish secure connections. MuleSoft documents HTTPS listener configuration in the HTTP Listener reference.

Removing the HTTP-specific entry from the VPC firewall rule complements the application-level HTTPS configuration by preventing network traffic from reaching the HTTP listener port. CloudHub VPC firewall rules control which inbound connections can reach deployed workers. Removing the rule that permits HTTP means that an HTTP path is not inadvertently left reachable through the VPC, while the required HTTPS listener port can remain explicitly allowed. This is defense in depth: the application is configured to serve HTTPS, and the network policy denies the insecure transport. MuleSoft's VPC firewall rules documentation describes how inbound access is controlled for applications in a VPC.

QUESTION NO: 11

An organization uses an external identity provider to issue OAuth 2.0 access tokens to partner applications. The organization wants to protect a Partner API managed in Anypoint Platform, while retaining the external identity provider as the authority that authenticates the partner applications.

Which two actions support this design?

Choose 2 answers

A. Configure the external identity provider as the client provider for Anypoint Platform

B. Apply an OAuth 2.0 or OpenID Connect access token enforcement policy to the Partner API

C. Apply only a Client ID Enforcement policy and accept any bearer token from partner applications

D. Configure the external identity provider only as an SSO provider for Anypoint Platform users

E. Store each partner user's password in the Partner API configuration properties

ANSWER: A B

Explanation:

The external identity provider must be configured as the client provider so that Anypoint Platform can use it for client-management integration. The API must also enforce OAuth access tokens issued by that trusted provider, using an appropriate OAuth or OpenID Connect token enforcement policy.

Client ID Enforcement alone identifies a registered client application but does not validate an externally issued OAuth access token. Configuring the identity provider only for Anypoint Platform user sign-in does not automatically protect API invocations.

QUESTION NO: 12

Which three tools automate the deployment of Mule applications?

Choose 3 answers

A. Runtime Manager

B. Anypoint Platform CLI

C. Platform APIs

D. Anypoint Studio

E. Mule Maven plugin

F. API Community Manager

ANSWER: B C E

Explanation:

Anypoint Platform CLI, Platform APIs, and the Mule Maven plugin are appropriate tools for automating Mule application deployments because each can be invoked noninteractively from scripts and continuous integration/continuous delivery pipelines. The Anypoint Platform CLI provides commands for managing Anypoint Platform resources and deploying applications, allowing a pipeline to authenticate, execute deployment operations, and report outcomes without requiring a user to operate the web interface. MuleSoft documents the CLI as a command-line tool for working with Anypoint Platform: Anypoint CLI documentation.

Platform APIs provide programmatic HTTP access to Anypoint Platform capabilities. A delivery process can call the relevant Runtime Manager or CloudHub endpoints to deploy and manage an application as part of an automated release workflow. This enables integration with external build servers, release orchestration tools, and custom deployment processes; see the CloudHub API documentation.

The Mule Maven plugin integrates packaging and deployment tasks into Maven builds. Because Maven goals can run in a build agent, the plugin supports repeatable deployments using versioned project configuration and pipeline-managed credentials. MuleSoft describes its deployment configuration and goals in the Mule Maven plugin reference.

QUESTION NO: 13

An API implementation is deployed to CloudHub.

What conditions can be alerted on using the default Anypoint Platform functionality, where the alert conditions depend on the end-to-end request processing of the API implementation?

A. When the API is invoked by an unrecognized API client

B. When a particular API client invokes the API too often within a given time period

C. When the response time of API invocations exceeds a threshold

D. When the API receives a very high number of API invocations

ANSWER: C

Explanation:

When the response time of API invocations exceeds a threshold is correct because response time is measured across the processing of an API request through to the production of its response. This makes it an end-to-end operational condition: Anypoint Platform must observe the completed invocation and calculate how long the API implementation took to respond before it can determine whether the configured threshold has been breached.

In Anypoint API Manager, an API alert can be configured with a response-time condition and threshold. When a request's observed response time meets the alert criteria, the platform can notify the configured recipients or connected systems. This supports proactive monitoring of user-visible API performance, including slow downstream dependencies, overloaded application workers, inefficient processing, and other latency that occurs during execution of the deployed implementation. The alert is based on the runtime behavior of requests handled by the API, rather than solely on gateway-level identification or counting of incoming requests. See the MuleSoft documentation for API Manager alerts and the CloudHub guidance on monitoring applications.

QUESTION NO: 14

An organization exposes a Partner Pricing API to several partner applications. Each partner must be identifiable through its own registered client application, and premium partners must be allowed a higher request rate than standard partners.

Which two API Manager controls should be configured?

Choose 2 answers

A. Apply a Client ID enforcement policy to the Partner Pricing API

B. Apply an SLA-based rate limiting policy and assign partner client applications to the appropriate SLA tiers

C. Apply a CORS policy that defines a separate allowed origin for each partner tier

D. Apply a JSON Threat Protection policy with a larger payload limit for premium partners

E. Create a separate Anypoint Platform business group for every partner application

ANSWER: A B

Explanation:

Client ID enforcement requires callers to identify the registered client application associated with the request. This creates the client-application identity needed to govern consumption and trace usage to a specific partner application.

An SLA-based rate limiting policy applies the request limits associated with the SLA tier assigned to that client application. The API owner can assign premium and standard partner applications to different tiers without deploying separate APIs or manually maintaining individual limits in the implementation.

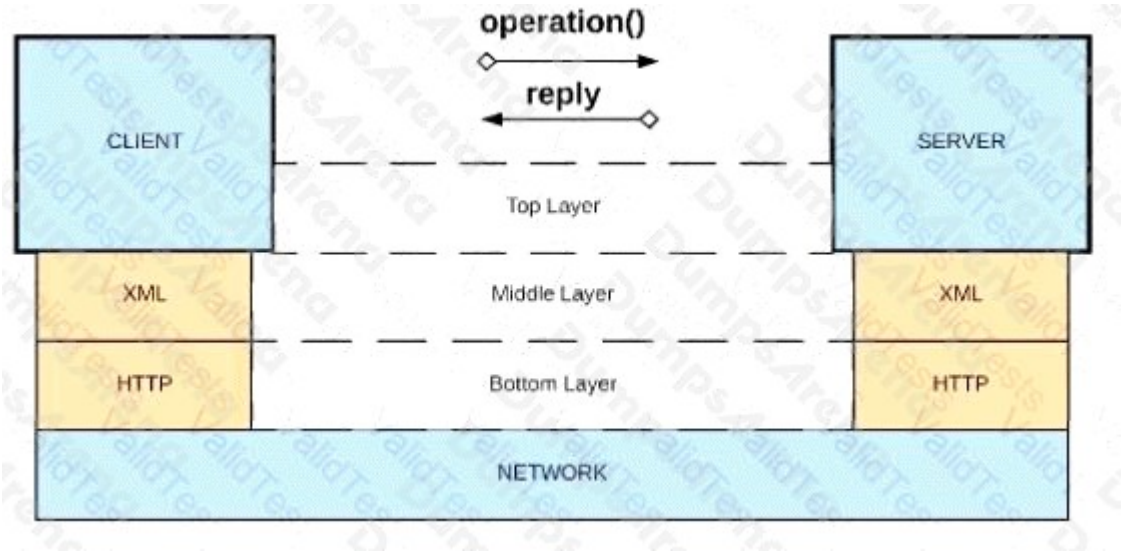
QUESTION NO: 15

Refer to the exhibit.

What is a valid API in the sense of API-led connectivity and application networks?

A) Java RMI over TCP

B) Java RMI over TCP



C) CORBA over HOP

D) XML over UDP

A. Option A

B. Option B

C. Option C

D. Option D

E. XML over HTTP

ANSWER: E

Explanation:

XML over HTTP is the valid choice because HTTP provides the broadly interoperable, standards-based transport expected for reusable APIs in an API-led connectivity and application-network architecture. An API is a contract that enables independent consumers to discover, invoke, secure, and reuse a capability without being coupled to its underlying implementation. HTTP supports this model through a well-understood request/response interaction style, uniform methods, status codes, headers, media types, and widespread tooling across languages and platforms. XML can be used as the message representation while HTTP provides the transport semantics needed by consumers and API gateways.

In MuleSoft, HTTP-based APIs can be managed consistently through API Manager and gateway capabilities. This enables application of cross-cutting controls such as client access enforcement, traffic management, and other policies at the API boundary, rather than embedding those concerns separately in every consumer or implementation. That governance and reuse model is central to building an application network from composable APIs. MuleSoft describes API-led connectivity as organizing reusable capabilities into layers, and its API management documentation describes applying policies to managed API instances. See MuleSoft: What is API-led connectivity? and MuleSoft API Manager policies.

QUESTION NO: 16

An existing Quoting API is defined in RAML and used by REST clients for interacting with the quoting engine. Currently there is a resource defined in the RAML that allows the creation of quotes; however, a new requirement was just received to allow for the updating of existing quotes.

Which two actions need to be taken to facilitate this change so it can be processed?

Choose 2 answers

A. Update the API implementation to accommodate the new update request

B. Remove the old client applications and create new client applications to account for the changes

C. Update the RAML with new method details for the update request

D. Deprecate existing versions of the API in Exchange

E. Add a new API policy to API Manager to allow access to the updated endpoint

ANSWER: A C

Explanation:

Update the RAML with new method details for the update request is required because the RAML contract is the authoritative API specification shared by API consumers and implementation teams. The contract must declare the resource and appropriate update operation, commonly `PUT` for replacement or `PATCH` for partial modification, along with request and response schemas, expected status codes, and any relevant URI parameters. Publishing the revised specification makes the intended capability discoverable and allows consumers to understand how to invoke it.

Update the API implementation to accommodate the new update request is also required because the API must actually route, validate, process, and persist the update operation described by the revised RAML. This includes implementing the operation's flow, validating the incoming quote data, invoking the appropriate quoting-engine service or data store, and returning responses that conform to the API contract. Keeping the implementation aligned with the RAML preserves a contract-first API lifecycle and ensures that the deployed endpoint delivers the newly documented behavior. MuleSoft documents RAML as a language for describing RESTful APIs and their methods, resources, requests, and responses: Design API specifications. For implementation generated or guided from an API specification, see Scaffold an application from an API specification.

QUESTION NO: 17

A Mule application consumes customer-update messages from Anypoint MQ. A downstream CRM system can be temporarily unavailable, and the business requires that a message not be acknowledged until the CRM update has completed successfully. Duplicate delivery is possible after failures.

Which two design decisions best support reliable processing?

Choose 2 answers

A. Acknowledge the Anypoint MQ message only after the CRM update succeeds

B. Design the CRM update processing to be idempotent by using a stable message or business identifier

C. Acknowledge the message before invoking the CRM system to maximize consumer throughput

D. Maintain processed-message identifiers only in an in-memory variable in each worker

E. Disable message redelivery so that a failed CRM invocation cannot produce a duplicate

ANSWER: A B

Explanation:

The consumer should acknowledge a message only after the CRM update completes successfully. If processing fails before acknowledgement, the message can be redelivered according to the queue configuration rather than being treated as successfully processed.

The CRM update processing must also be idempotent, using a stable message or business identifier to recognize repeated delivery. This is necessary because at-least-once delivery can result in duplicates when a failure occurs after the CRM operation but before message acknowledgement. Acknowledging before the CRM update risks losing the update, while an in-memory duplicate list is not reliable across restarts or multiple workers.