

Exam Questions MCPA-Level-1

MuleSoft Certified Platform Architect - Level 1

<https://www.2passeasy.com/dumps/MCPA-Level-1/>



NEW QUESTION 1

How are an API implementation, API client, and API consumer combined to invoke and process an API?

- A. The API consumer creates an API implementation, which receives API invocations from an API such that they are processed for an API client
- B. The API client creates an API consumer, which receives API invocations from an API such that they are processed for an API implementation
- C. The API consumer creates an API client, which sends API invocations to an API such that they are processed by an API implementation
- D. The API client creates an API consumer, which sends API invocations to an API such that they are processed by an API implementation

Answer: C

Explanation:

Correct Answer

The API consumer creates an API client, which sends API invocations to an API such that they are processed by an API implementation

***** Terminology:

>> API Client - It is a piece of code or program that is written to invoke an API

>> API Consumer - An owner/entity who owns the API Client. API Consumers write API clients.

>> API - The provider of the API functionality. Typically an API Instance on API Manager where they are managed and operated.

>> API Implementation - The actual piece of code written by API provider where the functionality of the API is implemented. Typically, these are Mule Applications running on Runtime Manager.

NEW QUESTION 2

What best describes the Fully Qualified Domain Names (FQDNs), also known as DNS entries, created when a Mule application is deployed to the CloudHub Shared Worker Cloud?

- A. A fixed number of FQDNs are created, IRRESPECTIVE of the environment and VPC design
- B. The FQDNs are determined by the application name chosen, IRRESPECTIVE of the region
- C. The FQDNs are determined by the application name, but can be modified by an administrator after deployment
- D. The FQDNs are determined by both the application name and the Anypoint Platform organization

Answer: B

Explanation:

Correct Answer

The FQDNs are determined by the application name chosen, IRRESPECTIVE of the region

>> When deploying applications to Shared Worker Cloud, the FQDN are always determined by application name chosen.

>> It does NOT matter what region the app is being deployed to.

>> Although it is fact and true that the generated FQDN will have the region included in it (Ex:

exp-salesorder-api.au-s1.cloudhub.io), it does NOT mean that the same name can be used when deploying to another CloudHub region.

>> Application name should be universally unique irrespective of Region and Organization and solely determines the FQDN for Shared Load Balancers.

NEW QUESTION 3

What condition requires using a CloudHub Dedicated Load Balancer?

- A. When cross-region load balancing is required between separate deployments of the same Mule application
- B. When custom DNS names are required for API implementations deployed to customer-hosted Mule runtimes
- C. When API invocations across multiple CloudHub workers must be load balanced
- D. When server-side load-balanced TLS mutual authentication is required between API implementations and API clients

Answer: D

Explanation:

Correct Answer

When server-side load-balanced TLS mutual authentication is required between API implementations and API clients

Fact/ Memory Tip: Although there are many benefits of CloudHub Dedicated Load balancer, TWO important things that should come to one's mind for considering it are:

>> Having URL endpoints with Custom DNS names on CloudHub deployed apps

>> Configuring custom certificates for both HTTPS and Two-way (Mutual) authentication. Coming to the options provided for this question:

>> We CANNOT use DLB to perform cross-region load balancing between separate deployments of the same Mule application.

>> We can have mapping rules to have more than one DLB URL pointing to same Mule app. But viceversa (More than one Mule app having same DLB URL) is NOT POSSIBLE

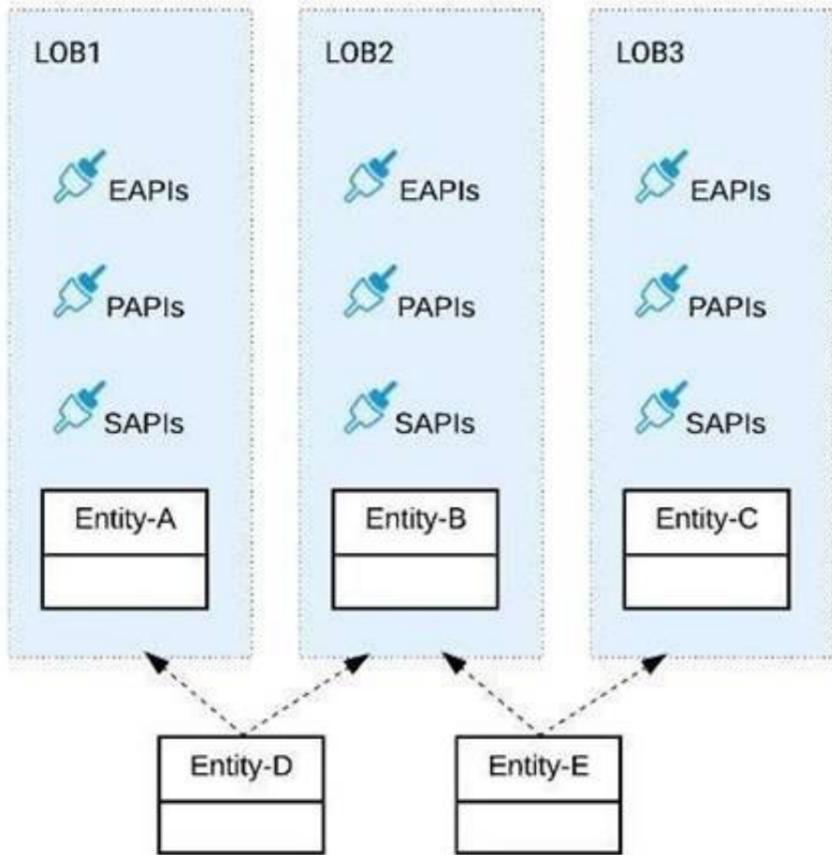
>> It is true that DLB helps to setup custom DNS names for Cloudhub deployed Mule apps but NOT true for apps deployed to Customer-hosted Mule Runtimes.

>> It is true that we can load balance API invocations across multiple CloudHub workers using DLB but it is NOT A MUST. We can achieve the same (load balancing) using SLB (Shared Load Balancer) too. We DO NOT necessarily require DLB to achieve it.

So the only right option that fits the scenario and requires us to use DLB is when TLS mutual authentication is required between API implementations and API clients.

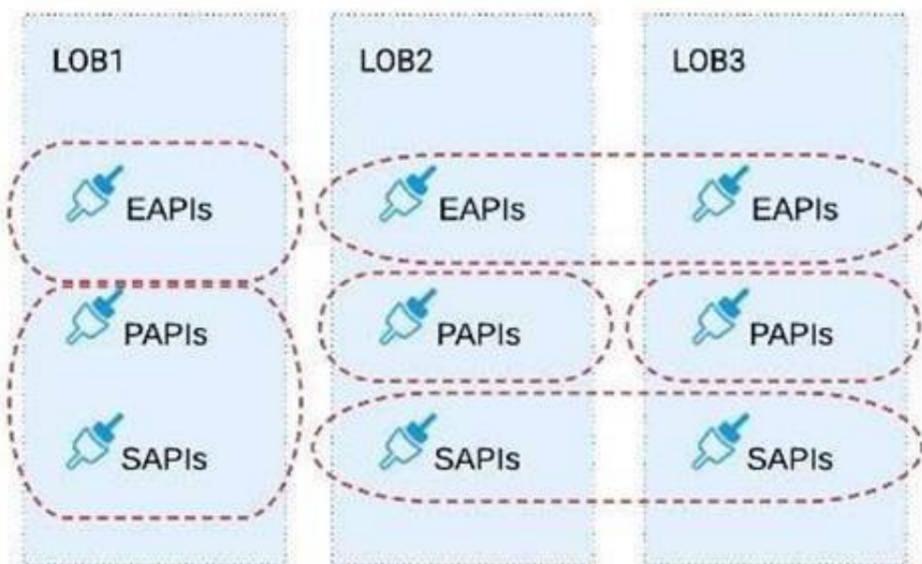
NEW QUESTION 4

Refer to the exhibit.

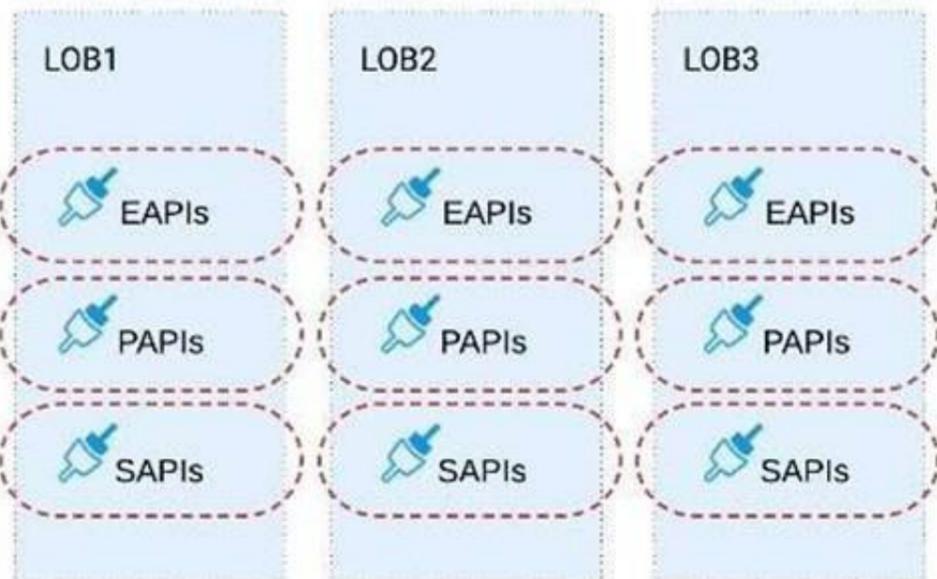


Three business processes need to be implemented, and the implementations need to communicate with several different SaaS applications. These processes are owned by separate (siloe) LOBs and are mainly independent of each other, but do share a few business entities. Each LOB has one development team and their own budget
 In this organizational context, what is the most effective approach to choose the API data models for the APIs that will implement these business processes with minimal redundancy of the data models?

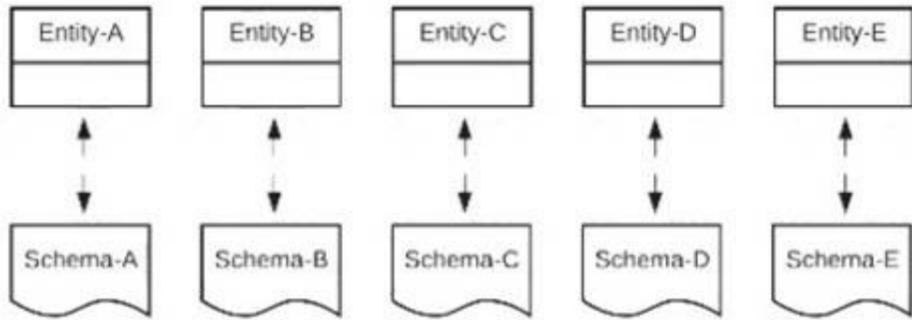
A) Build several Bounded Context Data Models that align with coherent parts of the business processes and the definitions of associated business entities



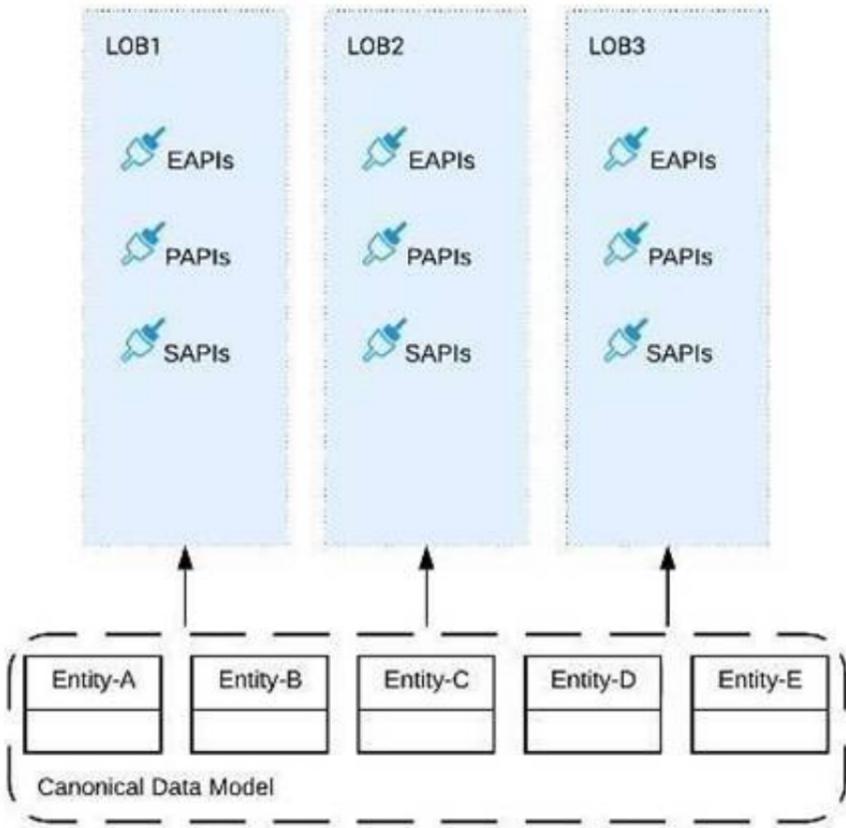
B) Build distinct data models for each API to follow established micro-services and Agile API-centric practices



C) Build all API data models using XML schema to drive consistency and reuse across the organization



D) Build one centralized Canonical Data Model (Enterprise Data Model) that unifies all the data types from all three business processes, ensuring the data model is consistent and non-redundant



- A. Option A
- B. Option B
- C. Option C
- D. Option D

Answer: A

Explanation:

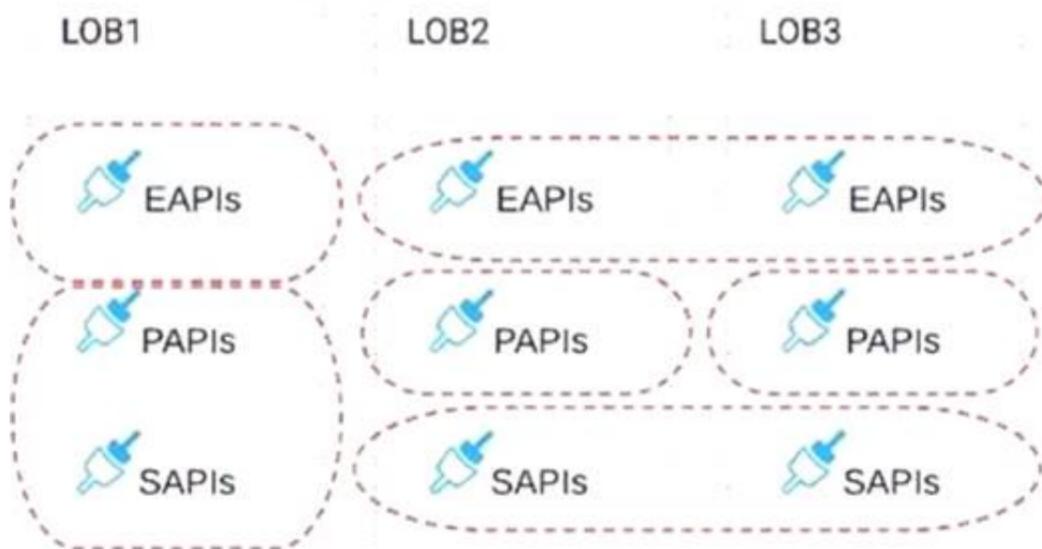
Correct Answer

Build several Bounded Context Data Models that align with coherent parts of the business processes and the definitions of associated business entities.

>> The options w.r.t building API data models using XML schema/ Agile API-centric practices are irrelevant to the scenario given in the question. So these two are INVALID.

>> Building EDM (Enterprise Data Model) is not feasible or right fit for this scenario as the teams and LOBs work in silo and they all have different initiatives, budget etc.. Building EDM needs intensive coordination among all the team which evidently seems not possible in this scenario.

So, the right fit for this scenario is to build several Bounded Context Data Models that align with coherent parts of the business processes and the definitions of associated business entities.



NEW QUESTION 5

A company uses a hybrid Anypoint Platform deployment model that combines the EU control plane with customer-hosted Mule runtimes. After successfully testing a Mule API implementation in the Staging environment, the Mule API implementation is set with environment-specific properties and must be promoted to the

Production environment. What is a way that MuleSoft recommends to configure the Mule API implementation and automate its promotion to the Production environment?

- A. Bundle properties files for each environment into the Mule API implementation's deployable archive, then promote the Mule API implementation to the Production environment using Anypoint CLI or the Anypoint Platform REST APIs.
- B. Modify the Mule API implementation's properties in the API Manager Properties tab, then promote the Mule API implementation to the Production environment using API Manager
- C. Modify the Mule API implementation's properties in Anypoint Exchange, then promote the Mule API implementation to the Production environment using Runtime Manager
- D. Use an API policy to change properties in the Mule API implementation deployed to the Staging environment and another API policy to deploy the Mule API implementation to the Production environment

Answer: A

Explanation:

Correct Answer

Bundle properties files for each environment into the Mule API implementation's deployable archive, then promote the Mule API implementation to the Production environment using Anypoint CLI or the Anypoint Platform REST APIs

>> Anypoint Exchange is for asset discovery and documentation. It has got no provision to modify the properties of Mule API implementations at all.
>> API Manager is for managing API instances, their contracts, policies and SLAs. It has also got no provision to modify the properties of API implementations.
>> API policies are to address Non-functional requirements of APIs and has again got no provision to modify the properties of API implementations.
So, the right way and recommended way to do this as part of development practice is to bundle properties files for each environment into the Mule API implementation and just point and refer to respective file per environment.

NEW QUESTION 6

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:

Correct Answer

When the response time of API invocations exceeds a threshold

>> Alerts can be setup for all the given options using the default Anypoint Platform functionality
>> However, the question insists on an alert whose conditions depend on the end-to-end request processing of the API implementation.
>> Alert w.r.t "Response Times" is the only one which requires end-to-end request processing of API implementation in order to determine if the threshold is exceeded or not.

NEW QUESTION 7

An API implementation is updated. When must the RAML definition of the API also be updated?

- A. When the API implementation changes the structure of the request or response messages
- B. When the API implementation changes from interacting with a legacy backend system deployed on-premises to a modern, cloud-based (SaaS) system
- C. When the API implementation is migrated from an older to a newer version of the Mule runtime
- D. When the API implementation is optimized to improve its average response time

Answer: A

Explanation:

Correct Answer

When the API implementation changes the structure of the request or response messages

>> RAML definition usually needs to be touched only when there are changes in the request/response schemas or in any traits on API.
>> It need not be modified for any internal changes in API implementation like performance tuning, backend system migrations etc..

NEW QUESTION 8

What are 4 important Platform Capabilities offered by Anypoint Platform?

- A. API Versioning, API Runtime Execution and Hosting, API Invocation, API Consumer Engagement
- B. API Design and Development, API Runtime Execution and Hosting, API Versioning, API Deprecation
- C. API Design and Development, API Runtime Execution and Hosting, API Operations and Management, API Consumer Engagement
- D. API Design and Development, API Deprecation, API Versioning, API Consumer Engagement

Answer: C

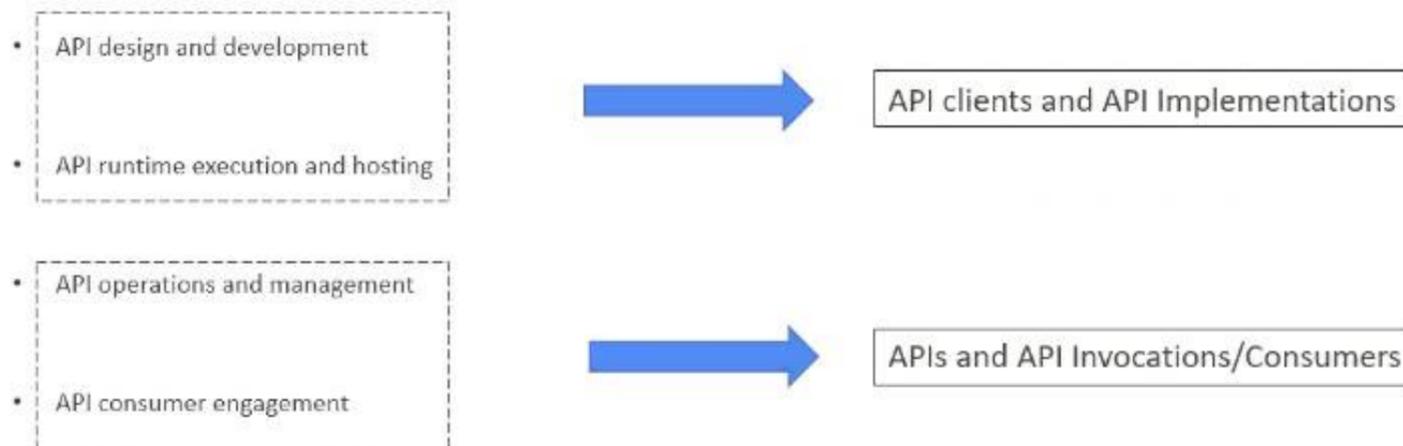
Explanation:

Correct Answer

API Design and Development, API Runtime Execution and Hosting, API Operations and Management, API Consumer Engagement

- >> API Design and Development - Anypoint Studio, Anypoint Design Center, Anypoint Connectors
- >> API Runtime Execution and Hosting - Mule Runtimes, CloudHub, Runtime Services
- >> API Operations and Management - Anypoint API Manager, Anypoint Exchange
- >> API Consumer Management - API Contracts, Public Portals, Anypoint Exchange, API Notebooks

Platform Capabilities



© Prasad Pokala

NEW QUESTION 9

What is true about the technology architecture of Anypoint VPCs?

- A. The private IP address range of an Anypoint VPC is automatically chosen by CloudHub
- B. Traffic between Mule applications deployed to an Anypoint VPC and on-premises systems can stay within a private network
- C. Each CloudHub environment requires a separate Anypoint VPC
- D. VPC peering can be used to link the underlying AWS VPC to an on-premises (non AWS) private network

Answer: B

Explanation:

Correct Answer

Traffic between Mule applications deployed to an Anypoint VPC and on-premises systems can stay within a private network

>> The private IP address range of an Anypoint VPC is NOT automatically chosen by CloudHub. It is chosen by us at the time of creating VPC using the CIDR blocks.

CIDR Block: The size of the Anypoint VPC in Classless Inter-Domain Routing (CIDR) notation.

For example, if you set it to 10.111.0.0/24, the Anypoint VPC is granted 256 IP addresses from 10.111.0.0 to 10.111.0.255.

Ideally, the CIDR Blocks you choose for the Anypoint VPC come from a private IP space, and should not overlap with any other Anypoint VPC's CIDR Blocks, or any CIDR Blocks in use in your corporate network.

← Create VPC

[Learn more about VPCs](#)

General information

Name: vpc1

Region: US East (N. Virginia)

CIDR Block: 10.0.0.0/16

Environments: Design

Set as default VPC

Business Groups: MyBusinessGroup (MyOrg)

that each CloudHub environment requires a separate Anypoint VPC. Once an Anypoint VPC is created, we can choose a same VPC by multiple environments. However, it is generally a best and recommended practice to always have separate Anypoint VPCs for Non-Prod and Prod environments. >> We use Anypoint VPN to link the underlying AWS VPC to an on-premises (non AWS) private network. NOT VPC Peering.

NEW QUESTION 10

When using CloudHub with the Shared Load Balancer, what is managed EXCLUSIVELY by the API implementation (the Mule application) and NOT by Anypoint Platform?

- A. The assignment of each HTTP request to a particular CloudHub worker
- B. The logging configuration that enables log entries to be visible in Runtime Manager
- C. The SSL certificates used by the API implementation to expose HTTPS endpoints
- D. The number of DNS entries allocated to the API implementation

Answer: C

Explanation:

Correct Answer

The SSL certificates used by the API implementation to expose HTTPS endpoints

>> The assignment of each HTTP request to a particular CloudHub worker is taken care by Anypoint Platform itself. We need not manage it explicitly in the API implementation and in fact we CANNOT manage it in the API implementation.
 >> The logging configuration that enables log entries to be visible in Runtime Manager is ALWAYS managed in the API implementation and NOT just for SLB. So this is not something we do EXCLUSIVELY when using SLB.
 >> We DO NOT manage the number of DNS entries allocated to the API implementation inside the code. Anypoint Platform takes care of this.
 It is the SSL certificates used by the API implementation to expose HTTPS endpoints that is to be managed EXCLUSIVELY by the API implementation. Anypoint Platform does NOT do this when using SLBs.

NEW QUESTION 10

A set of tests must be performed prior to deploying API implementations to a staging environment. Due to data security and access restrictions, untested APIs cannot be granted access to the backend systems, so instead mocked data must be used for these tests. The amount of available mocked data and its contents is sufficient to entirely test the API implementations with no active connections to the backend systems. What type of tests should be used to incorporate this mocked data?

- A. Integration tests
- B. Performance tests
- C. Functional tests (Blackbox)
- D. Unit tests (Whitebox)

Answer: D

Explanation:

Correct Answer

Unit tests (Whitebox)

NEW QUESTION 13

When could the API data model of a System API reasonably mimic the data model exposed by the corresponding backend system, with minimal improvements over the backend system's data model?

- A. When there is an existing Enterprise Data Model widely used across the organization
- B. When the System API can be assigned to a bounded context with a corresponding data model
- C. When a pragmatic approach with only limited isolation from the backend system is deemed appropriate
- D. When the corresponding backend system is expected to be replaced in the near future

Answer: C

Explanation:

Correct Answer

When a pragmatic approach with only limited isolation from the backend system is deemed appropriate.

***** General guidance w.r.t choosing Data Models:

- >> If an Enterprise Data Model is in use then the API data model of System APIs should make use of data types from that Enterprise Data Model and the corresponding API implementation should translate between these data types from the Enterprise Data Model and the native data model of the backend system.
 - >> If no Enterprise Data Model is in use then each System API should be assigned to a Bounded Context, the API data model of System APIs should make use of data types from the corresponding Bounded Context Data Model and the corresponding API implementation should translate between these data types from the Bounded Context Data Model and the native data model of the backend system. In this scenario, the data types in the Bounded Context Data Model are defined purely in terms of their business characteristics and are typically not related to the native data model of the backend system. In other words, the translation effort may be significant.
 - >> If no Enterprise Data Model is in use, and the definition of a clean Bounded Context Data Model is considered too much effort, then the API data model of System APIs should make use of data types that approximately mirror those from the backend system, same semantics and naming as backend system, lightly sanitized, expose all fields needed for the given System API's functionality, but not significantly more and making good use of REST conventions. The latter approach, i.e., exposing in System APIs an API data model that basically mirrors that of the backend system, does not provide satisfactory isolation from backend systems through the System API tier on its own. In particular, it will typically not be possible to "swap out" a backend system without significantly changing all System APIs in front of that backend system and therefore the API implementations of all Process APIs that depend on those System APIs! This is so because it is not desirable to prolong the life of a previous backend system's data model in the form of the API data model of System APIs that now front a new backend system. The API data models of System APIs following this approach must therefore change when the backend system is replaced.
- On the other hand:
- >> It is a very pragmatic approach that adds comparatively little overhead over accessing the backend system directly
 - >> Isolates API clients from intricacies of the backend system outside the data model (protocol, authentication, connection pooling, network address, ...)
 - >> Allows the usual API policies to be applied to System APIs
 - >> Makes the API data model for interacting with the backend system explicit and visible, by exposing it in the RAML definitions of the System APIs
 - >> Further isolation from the backend system data model does occur in the API implementations of the Process API tier

NEW QUESTION 18

Question 10: Skipped

An API implementation returns three X-RateLimit-* HTTP response headers to a requesting API client. What type of information do these response headers indicate to the API client?

- A. The error codes that result from throttling
- B. A correlation ID that should be sent in the next request
- C. The HTTP response size
- D. The remaining capacity allowed by the API implementation

Answer: D

Explanation:

Correct Answer

The remaining capacity allowed by the API implementation.

>> Reference:

<https://docs.mulesoft.com/api-manager/2.x/rate-limiting-and-throttling-sla-based-policies#response-headers>

Response Headers

Three headers are included in request responses that inform users about the SLA restrictions and inform them when nearing the threshold.

When the SLA enforces multiple policies that limit request throughput, a single set of headers pertaining to the most restrictive of the policies provides this information.

For example, a user of your API may receive a response that includes these headers:

```
X-RateLimit-Limit: 20
X-RateLimit-Remaining: 14
X-RateLimit-Reset: 19100
```

Within the next 19100 milliseconds, only 14 more requests are allowed by the SLA, which is set to allow 20 within this time-window.

NEW QUESTION 20

An Order API must be designed that contains significant amounts of integration logic and involves the invocation of the Product API.

The power relationship between Order API and Product API is one of "Customer/Supplier", because the Product API is used heavily throughout the organization and is developed by a dedicated development team located in the office of the CTO.

What strategy should be used to deal with the API data model of the Product API within the Order API?

- A. Convince the development team of the Product API to adopt the API data model of the Order API such that the integration logic of the Order API can work with one consistent internal data model
- B. Work with the API data types of the Product API directly when implementing the integration logic of the Order API such that the Order API uses the same

(unchanged) data types as the Product API

C. Implement an anti-corruption layer in the Order API that transforms the Product API data model into internal data types of the Order API

D. Start an organization-wide data modeling initiative that will result in an Enterprise Data Model that will then be used in both the Product API and the Order API

Answer: C

Explanation:

Correct Answer

Convince the development team of the product API to adopt the API data model of the Order API such that integration logic of the Order API can work with one consistent internal data model

***** Key details to note from the given scenario:

>> Power relationship between Order API and Product API is customer/supplier

So, as per below rules of "Power Relationships", the caller (in this case Order API) would request for features to the called (Product API team) and the Product API team would need to accommodate those requests.

NEW QUESTION 25

When designing an upstream API and its implementation, the development team has been advised to NOT set timeouts when invoking a downstream API, because that downstream API has no SLA that can be relied upon. This is the only downstream API dependency of that upstream API.

Assume the downstream API runs uninterrupted without crashing. What is the impact of this advice?

A. An SLA for the upstream API CANNOT be provided

B. The invocation of the downstream API will run to completion without timing out

C. A default timeout of 500 ms will automatically be applied by the Mule runtime in which the upstream API implementation executes

D. A load-dependent timeout of less than 1000 ms will be applied by the Mule runtime in which the downstream API implementation executes

Answer: A

Explanation:

Correct Answer

An SLA for the upstream API CANNOT be provided.

>> First thing first, the default HTTP response timeout for HTTP connector is 10000 ms (10 seconds). NOT 500 ms.

>> Mule runtime does NOT apply any such "load-dependent" timeouts. There is no such behavior currently in Mule.

>> As there is default 10000 ms time out for HTTP connector, we CANNOT always guarantee that the invocation of the downstream API will run to completion without timing out due to its unreliable SLA times. If the response time crosses 10 seconds then the request may time out.

The main impact due to this is that a proper SLA for the upstream API CANNOT be provided.

NEW QUESTION 26

Mule applications that implement a number of REST APIs are deployed to their own subnet that is inaccessible from outside the organization.

External business-partners need to access these APIs, which are only allowed to be invoked from a separate subnet dedicated to partners - called Partner-subnet.

This subnet is accessible from the public internet, which allows these external partners to reach it.

Anypoint Platform and Mule runtimes are already deployed in Partner-subnet. These Mule runtimes can already access the APIs.

What is the most resource-efficient solution to comply with these requirements, while having the least impact on other applications that are currently using the APIs?

A. Implement (or generate) an API proxy Mule application for each of the APIs, then deploy the API proxies to the Mule runtimes

B. Redeploy the API implementations to the same servers running the Mule runtimes

C. Add an additional endpoint to each API for partner-enablement consumption

D. Duplicate the APIs as Mule applications, then deploy them to the Mule runtimes

Answer: A

NEW QUESTION 31

What is a key requirement when using an external Identity Provider for Client Management in Anypoint Platform?

A. Single sign-on is required to sign in to Anypoint Platform

B. The application network must include System APIs that interact with the Identity Provider

C. To invoke OAuth 2.0-protected APIs managed by Anypoint Platform, API clients must submit access tokens issued by that same Identity Provider

D. APIs managed by Anypoint Platform must be protected by SAML 2.0 policies

Answer: C

Explanation:

<https://www.folkstalk.com/2019/11/mulesoft-integration-and-platform.html>

Correct Answer

To invoke OAuth 2.0-protected APIs managed by Anypoint Platform, API clients must submit access tokens issued by that same Identity Provider

>> It is NOT necessary that single sign-on is required to sign in to Anypoint Platform because we are using an external Identity Provider for Client Management

>> It is NOT necessary that all APIs managed by Anypoint Platform must be protected by SAML 2.0 policies because we are using an external Identity Provider for Client Management

>> Not TRUE that the application network must include System APIs that interact with the Identity Provider because we are using an external Identity Provider for Client Management

Only TRUE statement in the given options is - "To invoke OAuth 2.0-protected APIs managed by Anypoint Platform, API clients must submit access tokens issued by that same Identity Provider"

References:

<https://docs.mulesoft.com/api-manager/2.x/external-oauth-2.0-token-validation-policy> <https://blogs.mulesoft.com/dev/api-dev/api-security-ways-to-authenticate-and-authorize/>

NEW QUESTION 33

A REST API is being designed to implement a Mule application.
What standard interface definition language can be used to define REST APIs?

- A. Web Service Definition Language(WSDL)
- B. OpenAPI Specification (OAS)
- C. YAML
- D. AsyncAPI Specification

Answer: B

NEW QUESTION 37

Version 3.0.1 of a REST API implementation represents time values in PST time using ISO 8601 hh:mm:ss format. The API implementation needs to be changed to instead represent time values in CEST time using ISO 8601 hh:mm:ss format. When following the semver.org semantic versioning specification, what version should be assigned to the updated API implementation?

- A. 3.0.2
- B. 4.0.0
- C. 3.1.0
- D. 3.0.1

Answer: B

Explanation:

Correct Answer 4.0.0

***** As per semver.org semantic versioning specification:

Given a version number MAJOR.MINOR.PATCH, increment the:

- MAJOR version when you make incompatible API changes.
- MINOR version when you add functionality in a backwards compatible manner.
- PATCH version when you make backwards compatible bug fixes.

As per the scenario given in the question, the API implementation is completely changing its behavior. Although the format of the time is still being maintained as hh:mm:ss and there is no change in schema w.r.t format, the API will start functioning different after this change as the times are going to come completely different.

Example: Before the change, say, time is going as 09:00:00 representing the PST. Now on, after the change, the same time will go as 18:00:00 as Central European Summer Time is 9 hours ahead of Pacific Time.

>> This may lead to some uncertain behavior on API clients depending on how they are handling the times in the API response. All the API clients need to be informed that the API functionality is going to change and will return in CEST format. So, this considered as a MAJOR change and the version of API for this new change would be 4.0.0

NEW QUESTION 41

An organization has implemented a Customer Address API to retrieve customer address information. This API has been deployed to multiple environments and has been configured to enforce client IDs everywhere.

A developer is writing a client application to allow a user to update their address. The developer has found the Customer Address API in Anypoint Exchange and wants to use it in their client application.

What step of gaining access to the API can be performed automatically by Anypoint Platform?

- A. Approve the client application request for the chosen SLA tier
- B. Request access to the appropriate API Instances deployed to multiple environments using the client application's credentials
- C. Modify the client application to call the API using the client application's credentials
- D. Create a new application in Anypoint Exchange for requesting access to the API

Answer: A

Explanation:

Correct Answer

Approve the client application request for the chosen SLA tier

>> Only approving the client application request for the chosen SLA tier can be automated

>> Rest of the provided options are not valid

NEW QUESTION 46

An organization has created an API-led architecture that uses various API layers to integrate mobile clients with a backend system. The backend system consists of a number of specialized components and can be accessed via a REST API. The process and experience APIs share the same bounded-context model that is different from the backend data model. What additional canonical models, bounded-context models, or anti-corruption layers are best added to this architecture to help process data consumed from the backend system?

- A. Create a bounded-context model for every layer and overlap them when the boundary contexts overlap, letting API developers know about the differences between upstream and downstream data models
- B. Create a canonical model that combines the backend and API-led models to simplify and unify data models, and minimize data transformations.
- C. Create a bounded-context model for the system layer to closely match the backend data model, and add an anti-corruption layer to let the different bounded contexts cooperate across the system and process layers
- D. Create an anti-corruption layer for every API to perform transformation for every data model to match each other, and let data simply travel between APIs to avoid the complexity and overhead of building canonical models

Answer: C

Explanation:

Correct Answer

Create a bounded-context model for the system layer to closely match the backend data model, and add an anti-corruption layer to let the different bounded contexts cooperate across the system and process layers

>> Canonical models are not an option here as the organization has already put in efforts and created bounded-context models for Experience and Process APIs.
>> Anti-corruption layers for ALL APIs is unnecessary and invalid because it is mentioned that experience and process APIs share same bounded-context model. It is just the System layer APIs that need to choose their approach now.
>> So, having an anti-corruption layer just between the process and system layers will work well. Also to speed up the approach, system APIs can mimic the backend system data model.

NEW QUESTION 51

.....

THANKS FOR TRYING THE DEMO OF OUR PRODUCT

Visit Our Site to Purchase the Full Set of Actual MCPA-Level-1 Exam Questions With Answers.

We Also Provide Practice Exam Software That Simulates Real Exam Environment And Has Many Self-Assessment Features. Order the MCPA-Level-1 Product From:

<https://www.2passeasy.com/dumps/MCPA-Level-1/>

Money Back Guarantee

MCPA-Level-1 Practice Exam Features:

- * MCPA-Level-1 Questions and Answers Updated Frequently
- * MCPA-Level-1 Practice Questions Verified by Expert Senior Certified Staff
- * MCPA-Level-1 Most Realistic Questions that Guarantee you a Pass on Your FirstTry
- * MCPA-Level-1 Practice Test Questions in Multiple Choice Formats and Updatesfor 1 Year