

Exam Questions MCPA-Level-1

MuleSoft Certified Platform Architect - Level 1

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



NEW QUESTION 1

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 2

What Mule application can have API policies applied by Anypoint Platform to the endpoint exposed by that Mule application?

- A) A Mule application that accepts requests over HTTP/1.x



- B) A Mule application that accepts JSON requests over TCP but is NOT required to provide a response



- C) A Mute application that accepts JSON requests over WebSocket



- D) A Mule application that accepts gRPC requests over HTTP/2



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

Answer: A

Explanation:

Correct Answer

Option A

>> Anypoint API Manager and API policies are applicable to all types of HTTP/1.x APIs.

>> They are not applicable to WebSocket APIs, HTTP/2 APIs and gRPC APIs

NEW QUESTION 3

What is the most performant out-of-the-box solution in Anypoint Platform to track transaction state in an asynchronously executing long-running process implemented as a Mule application deployed to multiple CloudHub workers?

- A. Redis distributed cache
- B. java.util.WeakHashMap
- C. Persistent Object Store
- D. File-based storage

Answer: C

Explanation:

Correct Answer

Persistent Object Store

>> Redis distributed cache is performant but NOT out-of-the-box solution in Anypoint Platform

>> File-storage is neither performant nor out-of-the-box solution in Anypoint Platform

>> java.util.WeakHashMap needs a completely custom implementation of cache from scratch using Java code and is limited to the JVM where it is running. Which means the state in the cache is not worker aware when running on multiple workers. This type of cache is local to the worker. So, this is neither out-of-the-box nor worker-aware among multiple workers on cloudhub. <https://www.baeldung.com/java-weakhashmap>

>> Persistent Object Store is an out-of-the-box solution provided by Anypoint Platform which is performant as well as worker aware among multiple workers running on CloudHub. <https://docs.mulesoft.com/object-store/>

So, Persistent Object Store is the right answer.

NEW QUESTION 4

What is most likely NOT a characteristic of an integration test for a REST API implementation?

- A. The test needs all source and/or target systems configured and accessible
- B. The test runs immediately after the Mule application has been compiled and packaged
- C. The test is triggered by an external HTTP request
- D. The test prepares a known request payload and validates the response payload

Answer: B

Explanation:

Correct Answer

The test runs immediately after the Mule application has been compiled and packaged

>> Integration tests are the last layer of tests we need to add to be fully covered.

>> These tests actually run against Mule running with your full configuration in place and are tested from external source as they work in PROD.

>> These tests exercise the application as a whole with actual transports enabled. So, external systems are affected when these tests run.

So, these tests do NOT run immediately after the Mule application has been compiled and packaged.

FYI... Unit Tests are the one that run immediately after the Mule application has been compiled and packaged.

NEW QUESTION 5

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 ones 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 for 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 6

Say, there is a legacy CRM system called CRM-Z which is offering below functions:

- * 1. Customer creation
- * 2. Amend details of an existing customer
- * 3. Retrieve details of a customer
- * 4. Suspend a customer

A. Implement a system API named customerManagement which has all the functionalities wrapped in it asvarious operations/resources

B. Implement different system APIs named createCustomer, amendCustomer, retrieveCustomer and suspendCustomer as they are modular and has seperation of concerns

C. Implement different system APIs named createCustomerInCRMZ, amendCustomerInCRMZ, retrieveCustomerFromCRMZ and suspendCustomerInCRMZ as they are modular and has seperation of concerns

Answer: B

Explanation:

Correct Answer

Implement different system APIs named createCustomer, amendCustomer, retrieveCustomer and suspendCustomer as they are modular and has seperation of concerns

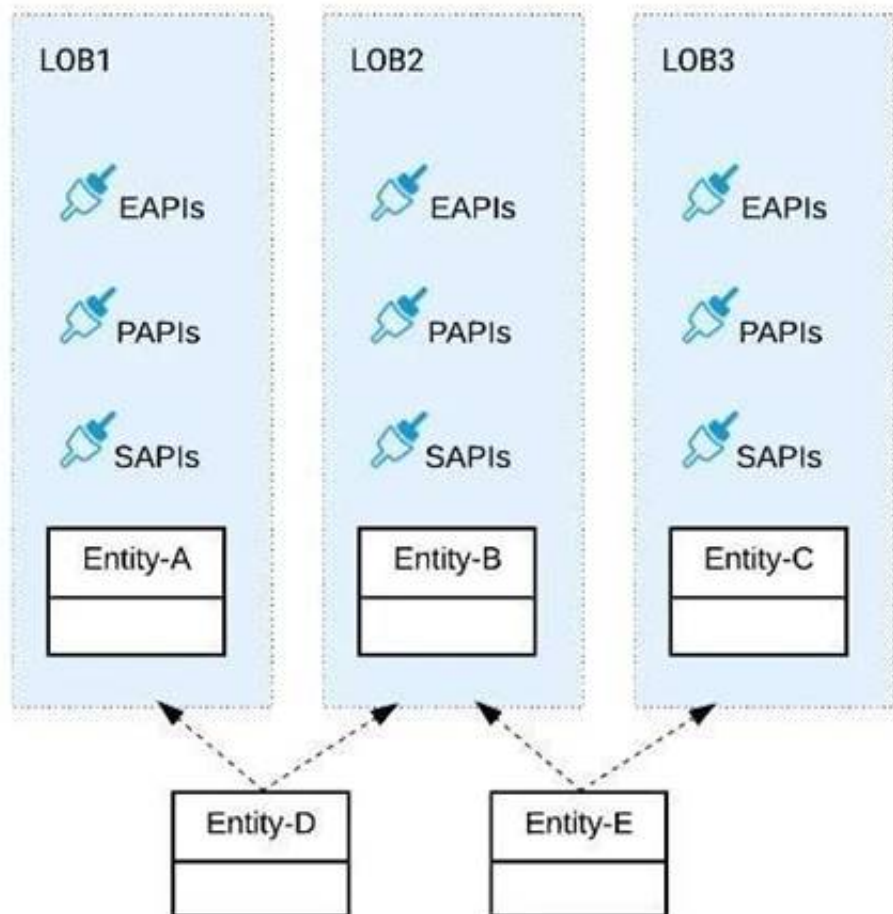
>> It is quite normal to have a single API and different Verb + Resource combinations. However, this fits well for an Experience API or a Process API but not a best architecture style for System APIs. So, option with just one customerManagement API is not the best choice here.

>> The option with APIs in createCustomerInCRMZ format is next close choice w.r.t modularization and less maintenance but the naming of APIs is directly coupled with the legacy system. A better foreseen approach would be to name your APIs by abstracting the backend system names as it allows seamless replacement/migration of any backend system anytime. So, this is not the correct choice too.

>> createCustomer, amendCustomer, retrieveCustomer and suspendCustomer is the right approach and is the best fit compared to other options as they are both modular and same time got the names decoupled from backend system and it has covered all requirements a System API needs.

NEW QUESTION 7

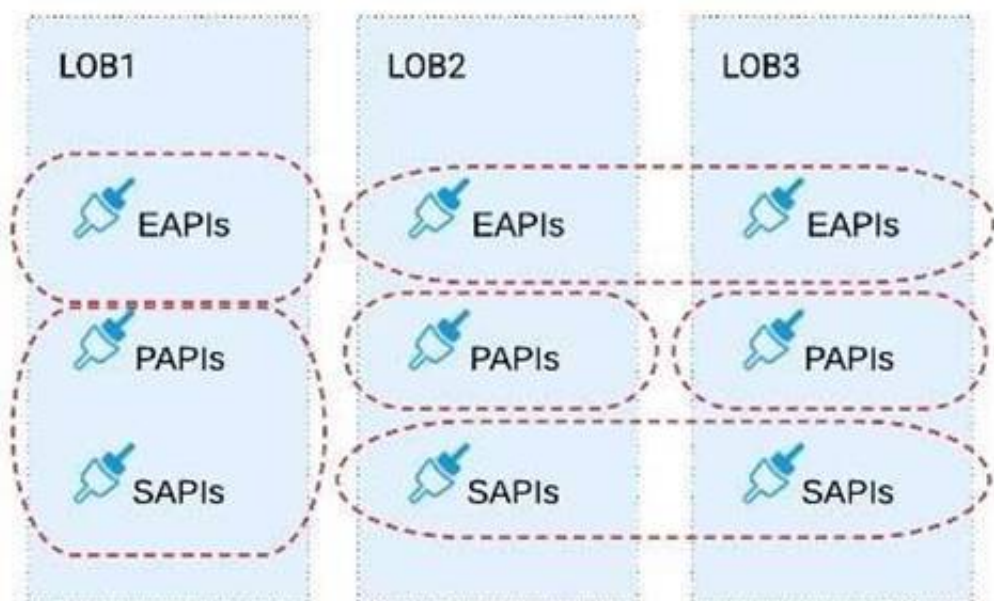
Refer to the exhibit.



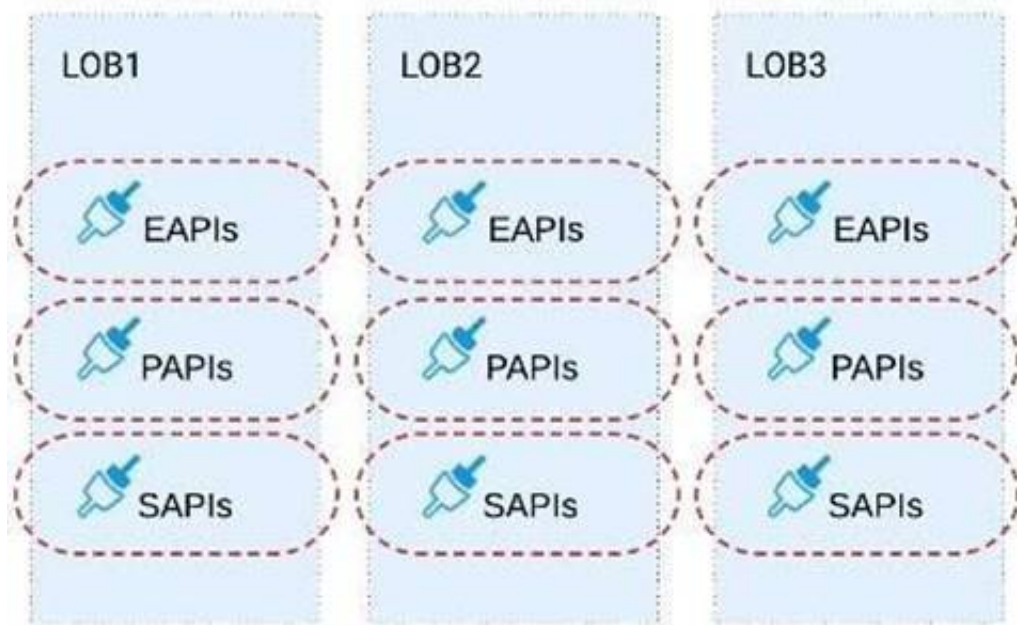
Three business processes need to be implemented, and the implementations need to communicate with several different SaaS applications. These processes are owned by separate (siloe) LOBs and are mainly independent of each other, but do share a few business entities. Each LOB has one development team and 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?

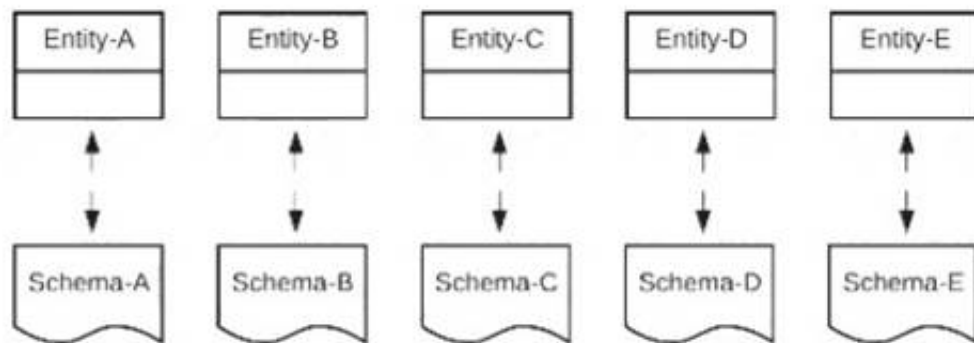
A) Build several Bounded Context Data Models that align with coherent parts of the business processes and the definitions of associated business entities



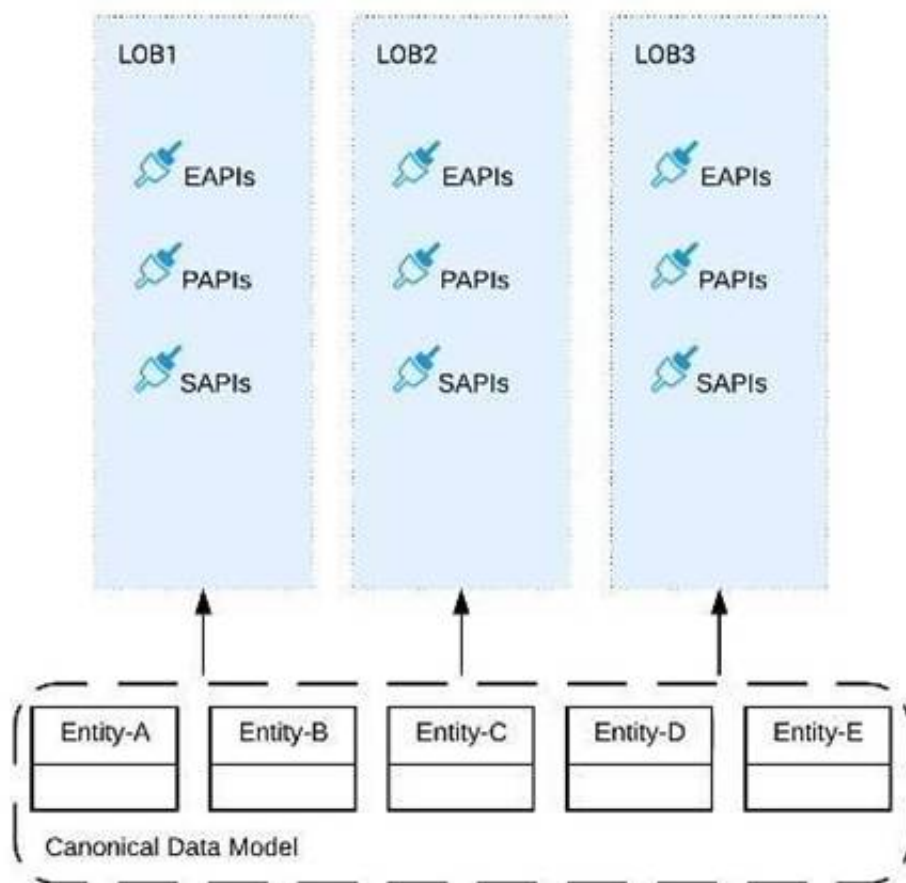
B) Build distinct data models for each API to follow established micro-services and Agile API-centric practices



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:

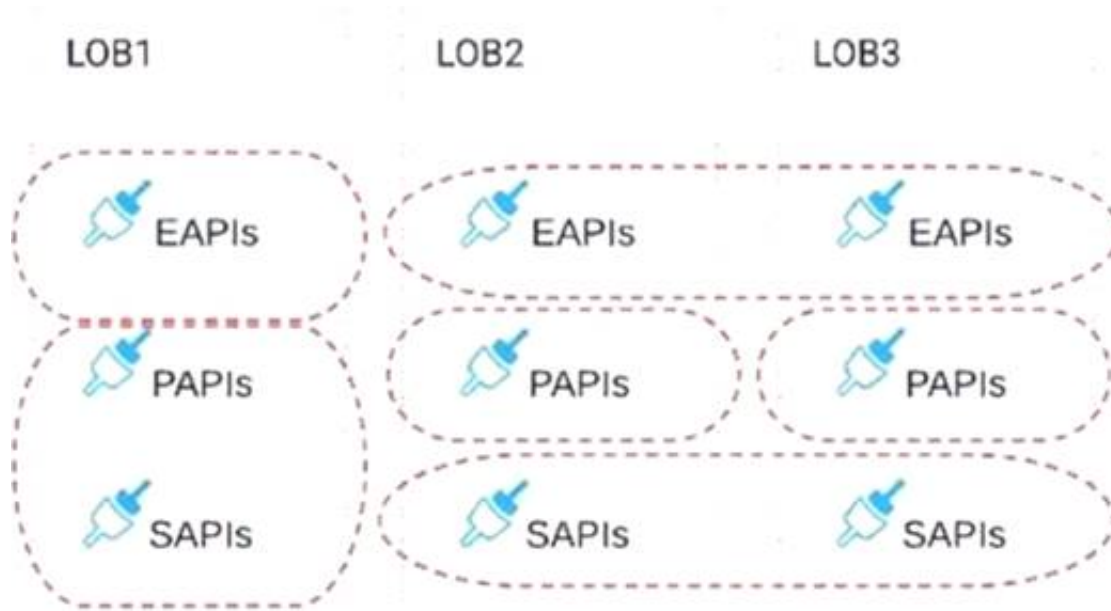
Correct Answer

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 8

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 9

Which of the below, when used together, makes the IT Operational Model effective?

- A. Create reusable assets, Do marketing on the created assets across organization, Arrange time to time LOB reviews to ensure assets are being consumed or not
- B. Create reusable assets, Make them discoverable so that LOB teams can self-serve and browse the APIs, Get active feedback and usage metrics
- C. Create reusable assets, make them discoverable so that LOB teams can self-serve and browse the APIs

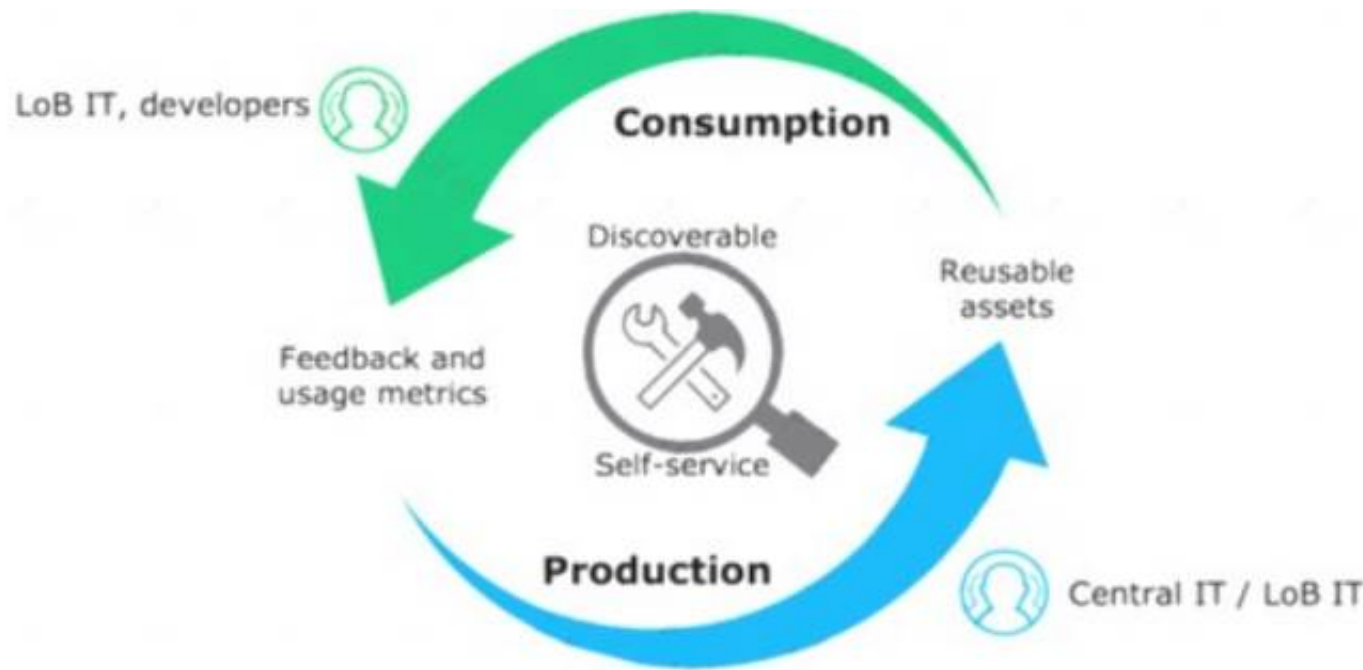
Answer: C

Explanation:

Correct Answer

Create reusable assets, Make them discoverable so that LOB teams can self-serve and browse the APIs, Get active feedback and usage metrics.

***** Diagram, arrow Description automatically generated



NEW QUESTION 10

How can the application of a rate limiting API policy be accurately reflected in the RAML definition of an API?

- A. By refining the resource definitions by adding a description of the rate limiting policy behavior
- B. By refining the request definitions by adding a remaining Requests query parameter with description, type, and example
- C. By refining the response definitions by adding the out-of-the-box Anypoint Platform rate-limit-enforcement securityScheme with description, type, and example
- D. By refining the response definitions by adding the x-ratelimit-* response headers with description, type, and example

Answer: D

Explanation:

Correct Answer

By refining the response definitions by adding the x-ratelimit-* response headers with description, type, and example

Response Headers

The following access-limiting policies return headers having information about the current state of the request:

- X-Ratelimit-Remaining: The amount of available quota.
- X-Ratelimit-Limit: The maximum available requests per window.
- X-Ratelimit-Reset: The remaining time, in milliseconds, until a new window starts.

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.

References:

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

NEW QUESTION 10

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 14

In an organization, the InfoSec team is investigating Anypoint Platform related data traffic.

From where does most of the data available to Anypoint Platform for monitoring and alerting originate?

- A. From the Mule runtime or the API implementation, depending on the deployment model
- B. From various components of Anypoint Platform, such as the Shared Load Balancer, VPC, and Mule runtimes
- C. From the Mule runtime or the API Manager, depending on the type of data
- D. From the Mule runtime irrespective of the deployment model

Answer: D

Explanation:

Correct Answer

From the Mule runtime irrespective of the deployment model

>> Monitoring and Alerting metrics are always originated from Mule Runtimes irrespective of the deployment model.

>> It may seem that some metrics (Runtime Manager) are originated from Mule Runtime and some are (API Invocations/ API Analytics) from API Manager.

However, this is realistically NOT TRUE. The reason is, API manager is just a management tool for API instances but all policies upon applying on APIs eventually gets executed on Mule Runtimes only (Either Embedded or API Proxy).

>> Similarly all API Implementations also run on Mule Runtimes.

So, most of the data required for monitoring and alerts are originated from Mule Runtimes only irrespective of whether the deployment model is MuleSoft-hosted or Customer-hosted or Hybrid.

NEW QUESTION 19

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

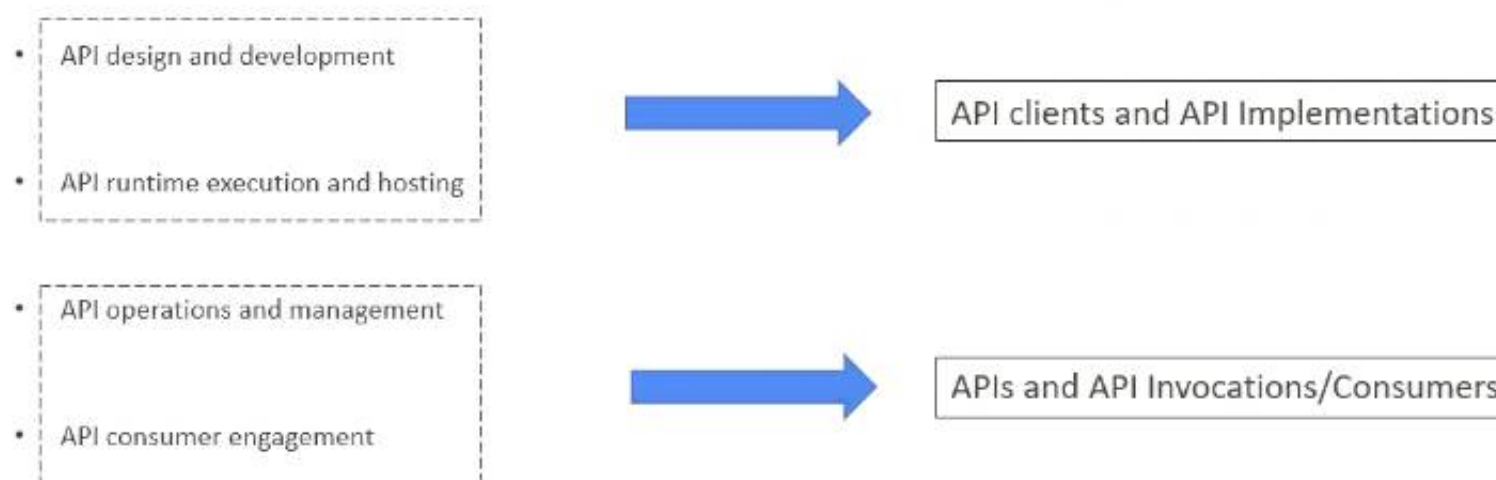
>> API Design and Development - Anypoint Studio, Anypoint Design Center, Anypoint Connectors

>> 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



NEW QUESTION 22

When must an API implementation be deployed to an Anypoint VPC?

- A. When the API Implementation must invoke publicly exposed services that are deployed outside of CloudHub in a customer- managed AWS instance
- B. When the API implementation must be accessible within a subnet of a restricted customer-hosted network that does not allow public access
- C. When the API implementation must be deployed to a production AWS VPC using the Mule Maven plugin
- D. When the API Implementation must write to a persistent Object Store

Answer: A

NEW QUESTION 23

What Mule application deployment scenario requires using Anypoint Platform Private Cloud Edition or Anypoint Platform for Pivotal Cloud Foundry?

- A. When it is required to make ALL applications highly available across multiple data centers
- B. When it is required that ALL APIs are private and NOT exposed to the public cloud
- C. When regulatory requirements mandate on-premises processing of EVERY data item, including meta-data
- D. When ALL backend systems in the application network are deployed in the organization's intranet

Answer: C

Explanation:

Correct Answer

When regulatory requirements mandate on-premises processing of EVERY data item, including meta-data.

We need NOT require to use Anypoint Platform PCE or PCF for the below. So these options are OUT.

>> We can make ALL applications highly available across multiple data centers using CloudHub too.

>> We can use Anypoint VPN and tunneling from CloudHub to connect to ALL backend systems in the application network that are deployed in the organization's intranet.

>> We can use Anypoint VPC and Firewall Rules to make ALL APIs private and NOT exposed to the public cloud.

Only valid reason in the given options that requires to use Anypoint Platform PCE/ PCF is - When regulatory requirements mandate on-premises processing of EVERY data item, including meta-data.

NEW QUESTION 27

An API implementation is being designed that must invoke an Order API, which is known to repeatedly experience downtime.

For this reason, a fallback API is to be called when the Order API is unavailable.

What approach to designing the invocation of the fallback API provides the best resilience?

- A. Search Anypoint Exchange for a suitable existing fallback API, and then implement invocations to this fallback API in addition to the Order API
- B. Create a separate entry for the Order API in API Manager, and then invoke this API as a fallback API if the primary Order API is unavailable
- C. Redirect client requests through an HTTP 307 Temporary Redirect status code to the fallback API whenever the Order API is unavailable
- D. Set an option in the HTTP Requester component that invokes the Order API to instead invoke a fallback API whenever an HTTP 4xx or 5xx response status code is returned from the Order API

Answer: A

Explanation:

Correct Answer

Search Anypoint exchange for a suitable existing fallback API, and then implement invocations to this fallback API in addition to the order API

>> It is not ideal and good approach, until unless there is a pre-approved agreement with the API clients that they will receive a HTTP 3xx temporary redirect status code and they have to implement fallback logic their side to call another API.

>> Creating separate entry of same Order API in API manager would just create an another instance of it on top of same API implementation. So, it does NO GOOD by using clone of same API as a fallback API. Fallback API should be ideally a different API implementation that is not same as primary one.

>> There is NO option currently provided by Anypoint HTTP Connector that allows us to invoke a fallback API when we receive certain HTTP status codes in response.

The only statement TRUE in the given options is to Search Anypoint exchange for a suitable existing fallback API, and then implement invocations to this fallback API in addition to the order API.

NEW QUESTION 29

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 30

What correctly characterizes unit tests of Mule applications?

- A. They test the validity of input and output of source and target systems
- B. They must be run in a unit testing environment with dedicated Mule runtimes for the environment
- C. They must be triggered by an external client tool or event source
- D. They are typically written using MUnit to run in an embedded Mule runtime that does not require external connectivity

Answer: D

Explanation:

Correct Answer

They are typically written using MUnit to run in an embedded Mule runtime that does not require external connectivity.

Below TWO are characteristics of Integration Tests but NOT unit tests:

>> They test the validity of input and output of source and target systems.

>> They must be triggered by an external client tool or event source.

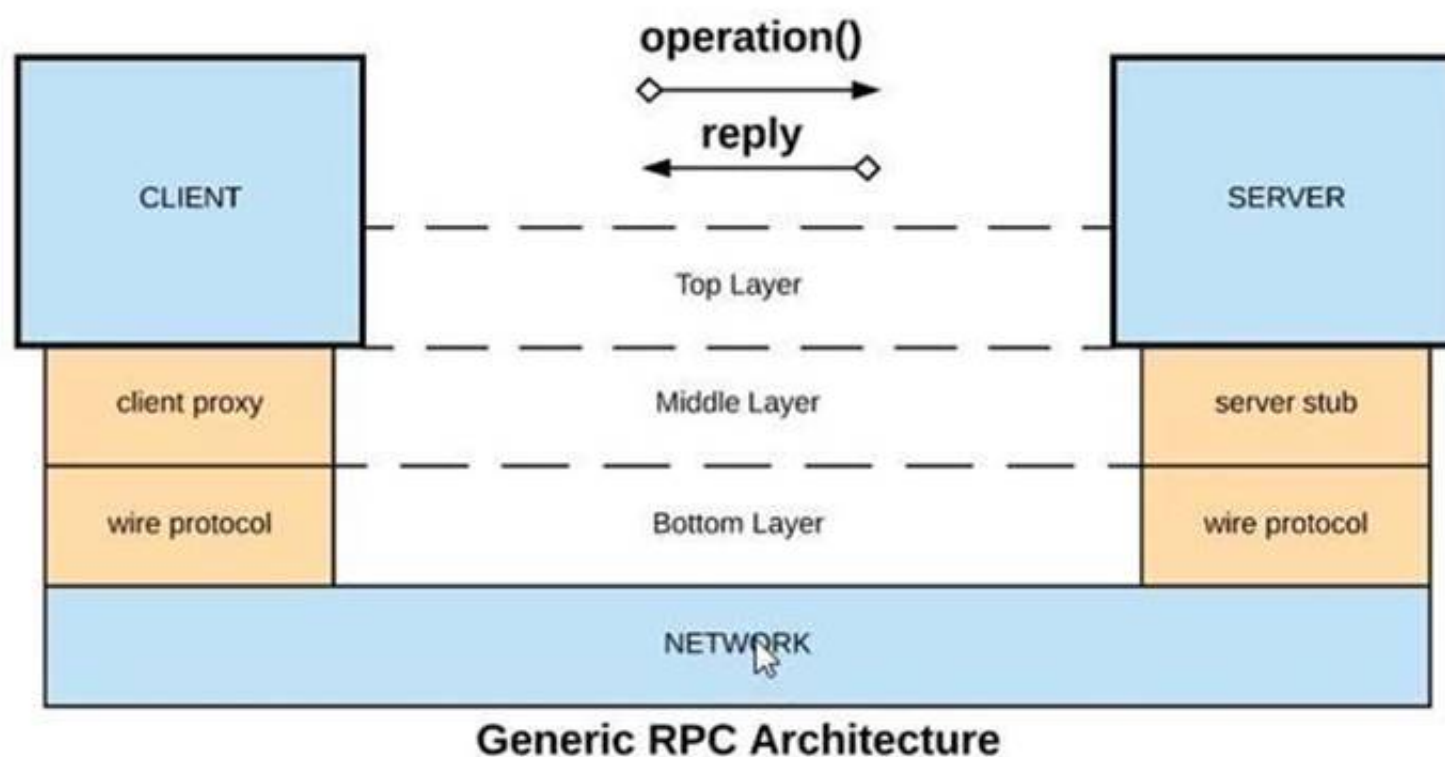
It is NOT TRUE that Unit Tests must be run in a unit testing environment with dedicated Mule runtimes for the environment.

MuleSoft offers MUnit for writing Unit Tests and they run in an embedded Mule Runtime without needing any separate/ dedicated Runtimes to execute them. They also do NOT need any external connectivity as MUnit supports mocking via stubs.

<https://dzone.com/articles/munit-framework>

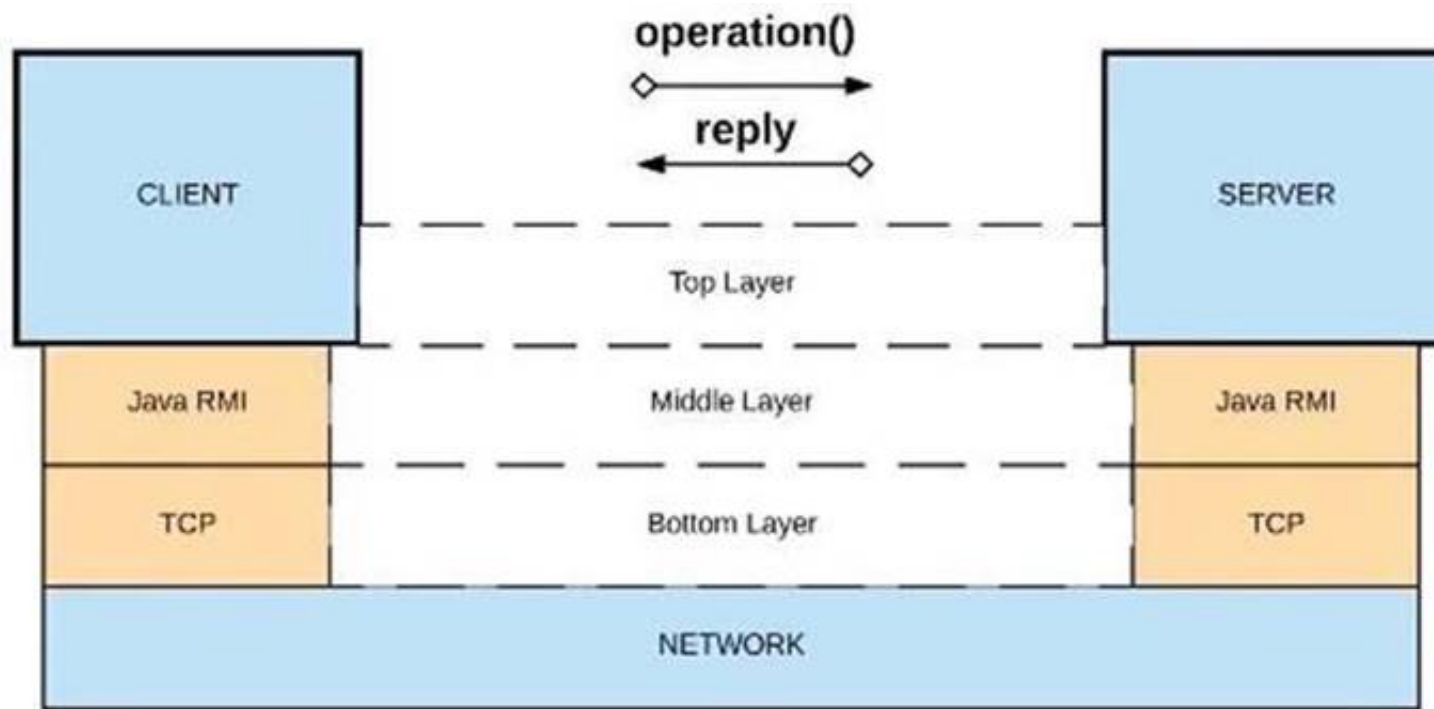
NEW QUESTION 34

Refer to the exhibit.

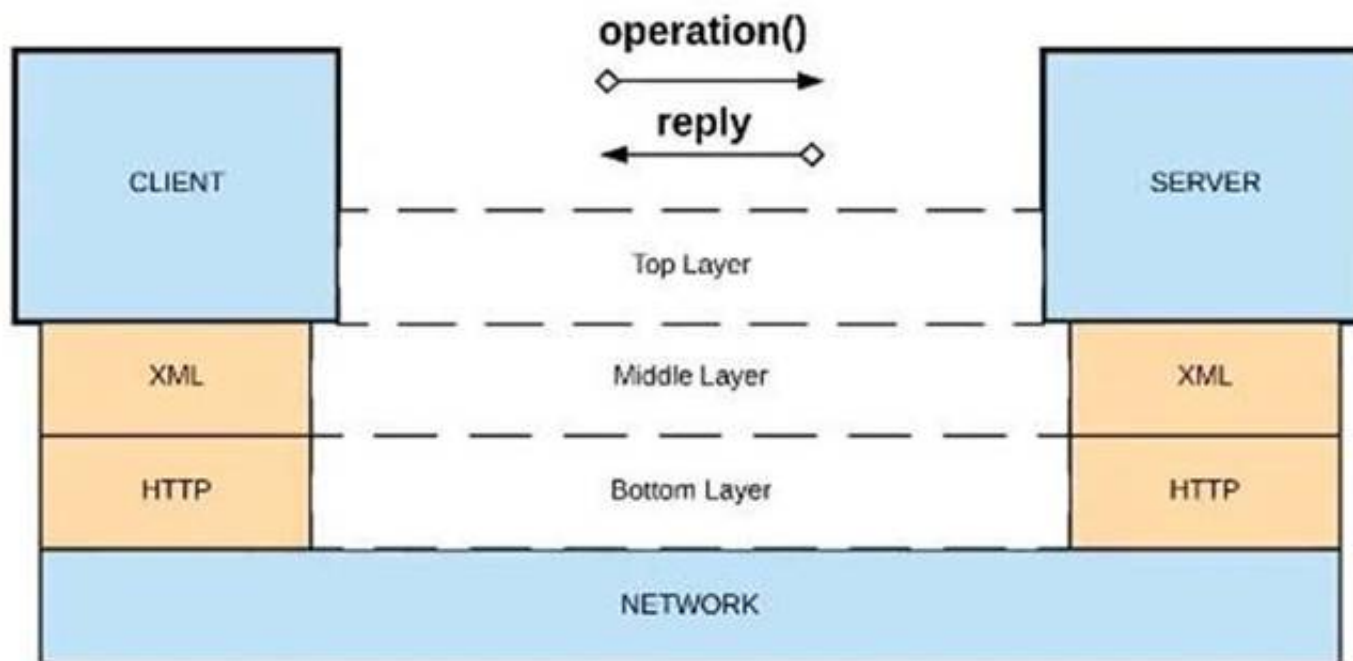


What is a valid API in the sense of API-led connectivity and application networks?

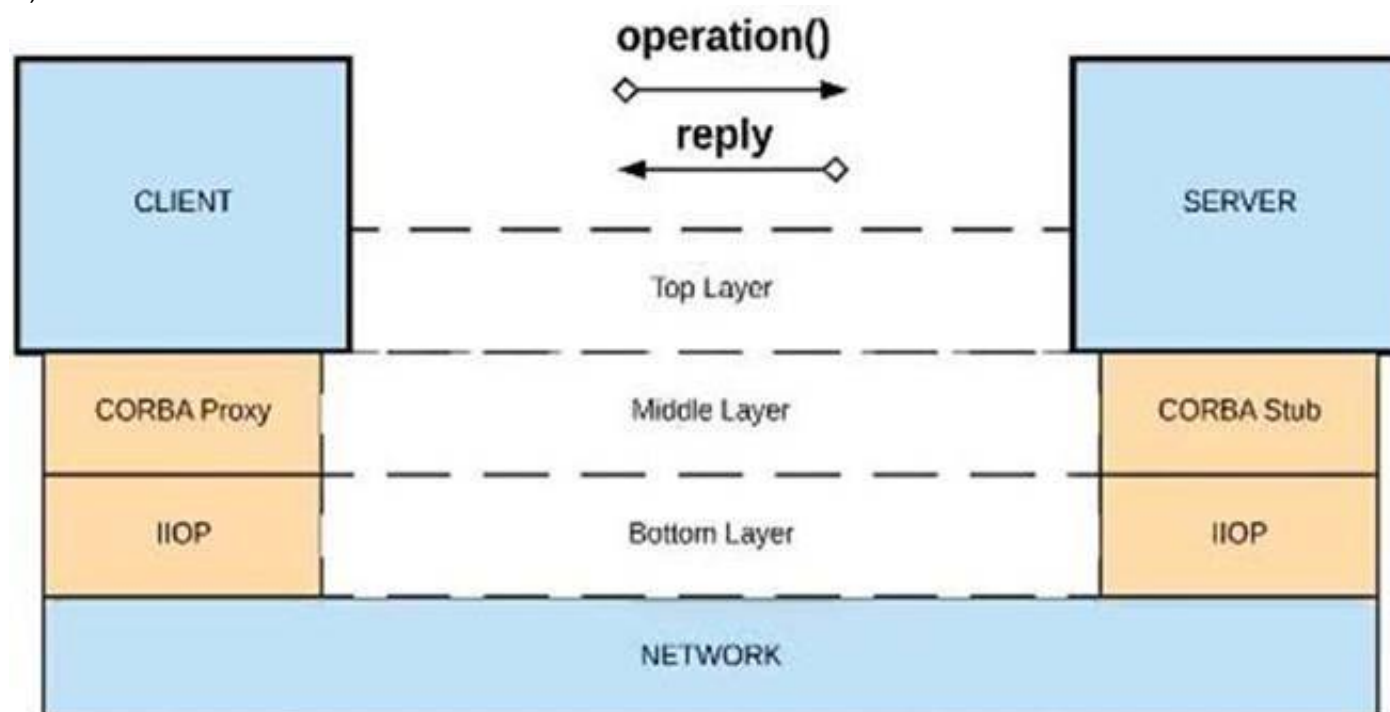
- A) Java RMI over TCP



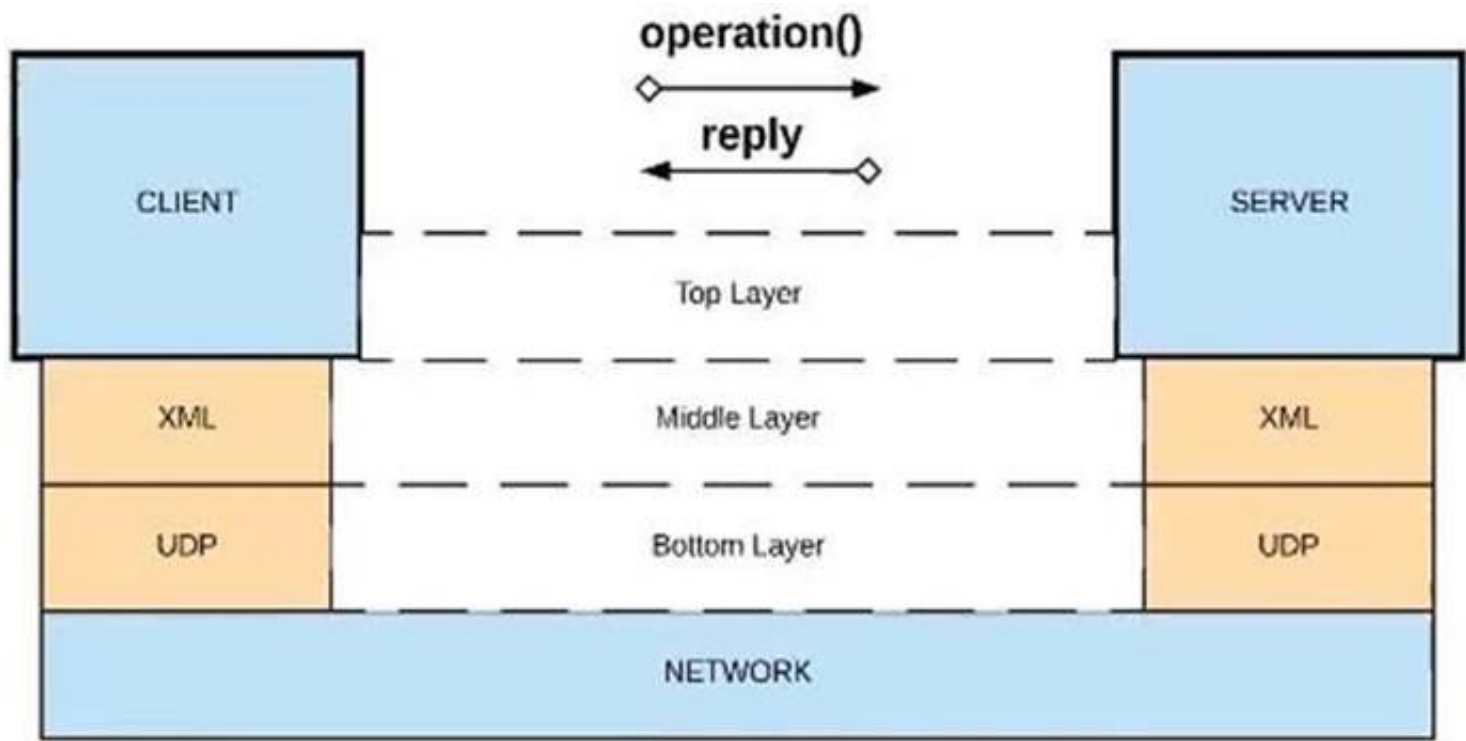
B) Java RMI over TCP



C) CORBA over IIOP



D) XML over UDP



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

Answer: D

Explanation:

\Correct Answer
XML over HTTP

>> API-led connectivity and Application Networks urge to have the APIs on HTTP based protocols for building most effective APIs and networks on top of them.
>> The HTTP based APIs allow the platform to apply various varieties of policies to address many NFRs
>> The HTTP based APIs also allow to implement many standard and effective implementation patterns that adhere to HTTP based w3c rules.
Bottom of Form Top of Form

NEW QUESTION 35

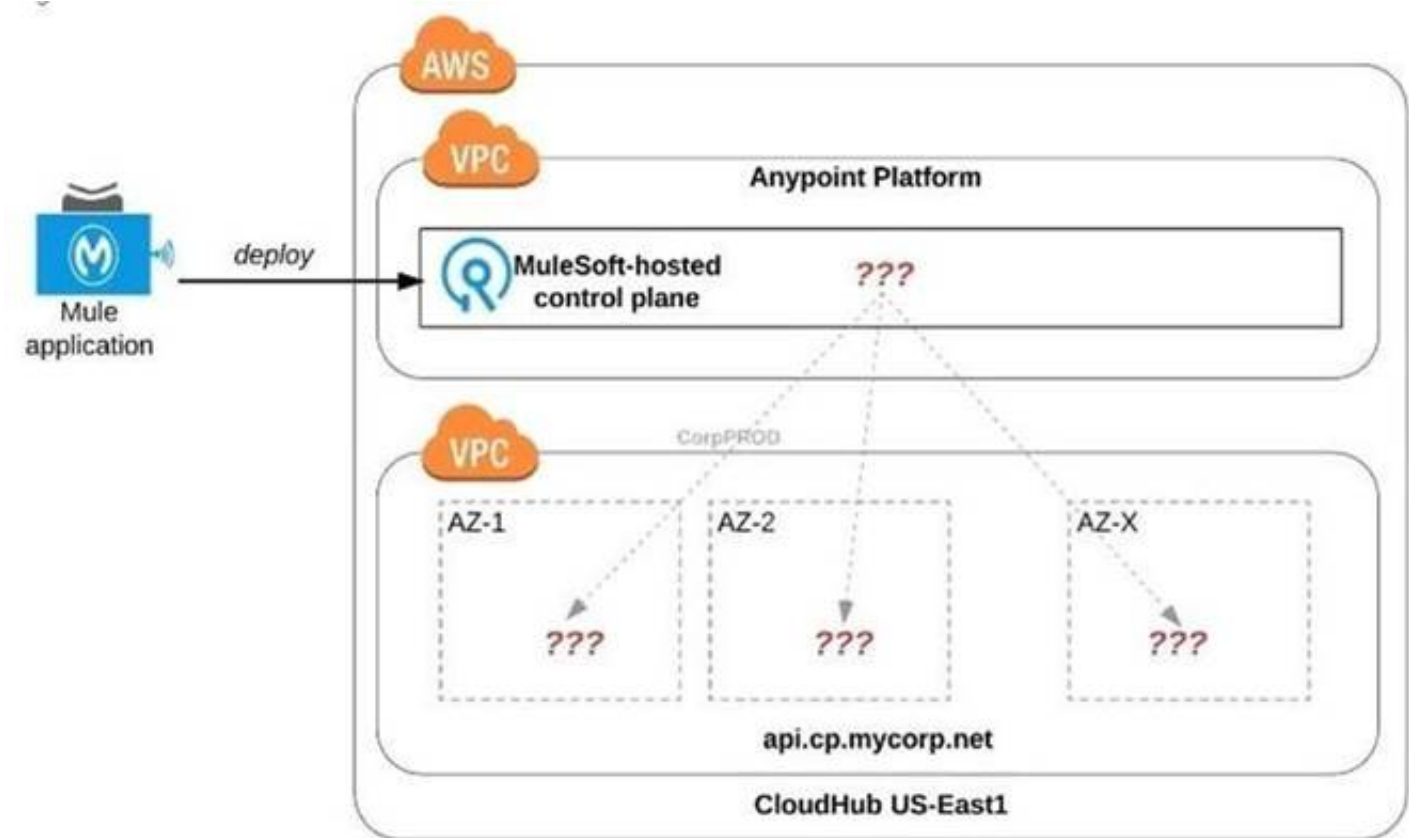
An API has been updated in Anypoint Exchange by its API producer from version 3.1.1 to 3.2.0 following accepted semantic versioning practices and the changes have been communicated via the API's public portal.
The API endpoint does NOT change in the new version.
How should the developer of an API client respond to this change?

- A. The update should be identified as a project risk and full regression testing of the functionality that uses this API should be run
- B. The API producer should be contacted to understand the change to existing functionality
- C. The API producer should be requested to run the old version in parallel with the new one
- D. The API client code ONLY needs to be changed if it needs to take advantage of new features

Answer: D

NEW QUESTION 40

Refer to the exhibit.



An organization uses one specific CloudHub (AWS) region for all CloudHub deployments.

How are CloudHub workers assigned to availability zones (AZs) when the organization's Mule applications are deployed to CloudHub in that region?

- A. Workers belonging to a given environment are assigned to the same AZ within that region
- B. AZs are selected as part of the Mule application's deployment configuration
- C. Workers are randomly distributed across available AZs within that region
- D. An AZ is randomly selected for a Mule application, and all the Mule application's CloudHub workers are assigned to that one AZ

Answer: D

Explanation:

Correct Answer

Workers are randomly distributed across available AZs within that region.

>> Currently, we only have control to choose which AWS Region to choose but there is no control at all using any configurations or deployment options to decide what Availability Zone (AZ) to assign to what worker.

>> There are No

fixed or implicit rules on platform too w.r.t assignment of AZ to workers based on environment or application.

>> They are completely assigned in random. However, cloudhub definitely ensures that HA is achieved by assigning the workers to more than one AZ so that all workers are not assigned to same AZ for same application.

NEW QUESTION 43

What is a key performance indicator (KPI) that measures the success of a typical C4E that is immediately apparent in responses from the Anypoint Platform APIs?

- A. The number of production outage incidents reported in the last 24 hours
- B. The number of API implementations that have a publicly accessible HTTP endpoint and are being managed by Anypoint Platform
- C. The fraction of API implementations deployed manually relative to those deployed using a CI/CD tool
- D. The number of API specifications in RAML or OAS format published to Anypoint Exchange

Answer: D

Explanation:

Correct Answer

The number of API specifications in RAML or OAS format published to Anypoint Exchange

>> The success of C4E always depends on their contribution to the number of reusable assets that they have helped to build and publish to Anypoint Exchange.

>> It is NOT due to any factors w.r.t # of outages, Manual vs CI/CD deployments or Publicly accessible HTTP endpoints

>> Anypoint Platform APIs help us to quickly run and get the number of published RAML/OAS assets to Anypoint Exchange. This clearly depicts how successful a C4E team is based on number of returned assets in the response.

NEW QUESTION 44

An organization wants to make sure only known partners can invoke the organization's APIs. To achieve this security goal, the organization wants to enforce a Client ID Enforcement policy in API Manager so that only registered partner applications can invoke the organization's APIs. In what type of API implementation does MuleSoft recommend adding an API proxy to enforce the Client ID Enforcement policy, rather than embedding the policy directly in the application's JVM?

- A. A Mule 3 application using APIkit
- B. A Mule 3 or Mule 4 application modified with custom Java code
- C. A Mule 4 application with an API specification
- D. A Non-Mule application

Answer: D

Explanation:

Correct Answer

A Non-Mule application

>> All type of Mule applications (Mule 3/ Mule 4/ with APIkit/ with Custom Java Code etc) running on Mule Runtimes support the Embedded Policy Enforcement on them.

>> The only option that cannot have or does not support embedded policy enforcement and must have API Proxy is for Non-Mule Applications.

So, Non-Mule application is the right answer.

NEW QUESTION 45

What API policy would be LEAST LIKELY used when designing an Experience API that is intended to work with a consumer mobile phone or tablet application?

- A. OAuth 2.0 access token enforcement
- B. Client ID enforcement
- C. JSON threat protection
- D. IPwhitelist

Answer: D

Explanation:

Correct Answer

IP whitelist

>> OAuth 2.0 access token and Client ID enforcement policies are VERY common to apply on Experience APIs as API consumers need to register and access the APIs using one of these mechanisms

>> JSON threat protection is also VERY common policy to apply on Experience APIs to prevent bad or suspicious payloads hitting the API implementations.

>> IP whitelisting policy is usually very common in Process and System APIs to only whitelist the IP range inside the local VPC. But also applied occasionally on some experience APIs where the End User/ API Consumers are FIXED.

>> When we know the API consumers upfront who are going to access certain Experience APIs, then we can request for static IPs from such consumers and whitelist them to prevent anyone else hitting the API.
 However, the experience API given in the question/ scenario is intended to work with a consumer mobile phone or tablet application. Which means, there is no way we can know all possible IPs that are to be whitelisted as mobile phones and tablets can so many in number and any device in the city/state/country/globe.
 So, It is very LEAST LIKELY to apply IP Whitelisting on such Experience APIs whose consumers are typically Mobile Phones or Tablets.

NEW QUESTION 50

The responses to some HTTP requests can be cached depending on the HTTP verb used in the request. According to the HTTP specification, for what HTTP verbs is this safe to do?

- A. PUT, POST, DELETE
- B. GET, HEAD, POST
- C. GET, PUT, OPTIONS
- D. GET, OPTIONS, HEAD

Answer: D

Explanation:

Correct Answer
 GET, OPTIONS, HEAD

APIs use HTTP-based protocols: cached HTTP responses from previous HTTP requests may potentially be returned if the same HTTP request is seen again.

Safe HTTP methods are ones that do not alter the state of the underlying resource. That is, the *HTTP responses to requests using safe HTTP methods may be cached.*

The HTTP standard requires the following HTTP methods on any resource to be safe:

- GET
- HEAD
- OPTIONS

Safety must be honored by REST APIs (but not by non-REST APIs like SOAP APIs): It is the *responsibility of every API implementation to implement GET, HEAD or OPTIONS methods such that they never change the state of a resource.*

<http://restcookbook.com/HTTP%20Methods/idempotency/>

NEW QUESTION 54

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 56

What should be ensured before sharing an API through a public Anypoint Exchange portal?

- A. The visibility level of the API instances of that API that need to be publicly accessible should be set to public visibility
- B. The users needing access to the API should be added to the appropriate role in Anypoint Platform
- C. The API should be functional with at least an initial implementation deployed and accessible for users to interact with

D. The API should be secured using one of the supported authentication/authorization mechanisms to ensure that data is not compromised

Answer: A

Explanation:



Correct Answer

The visibility level of the API instances of that API that need to be publicly accessible should be set to public visibility.

NEW QUESTION 58

Which layer in the API-led connectivity focuses on unlocking key systems, legacy systems, data sources etc and exposes the functionality?

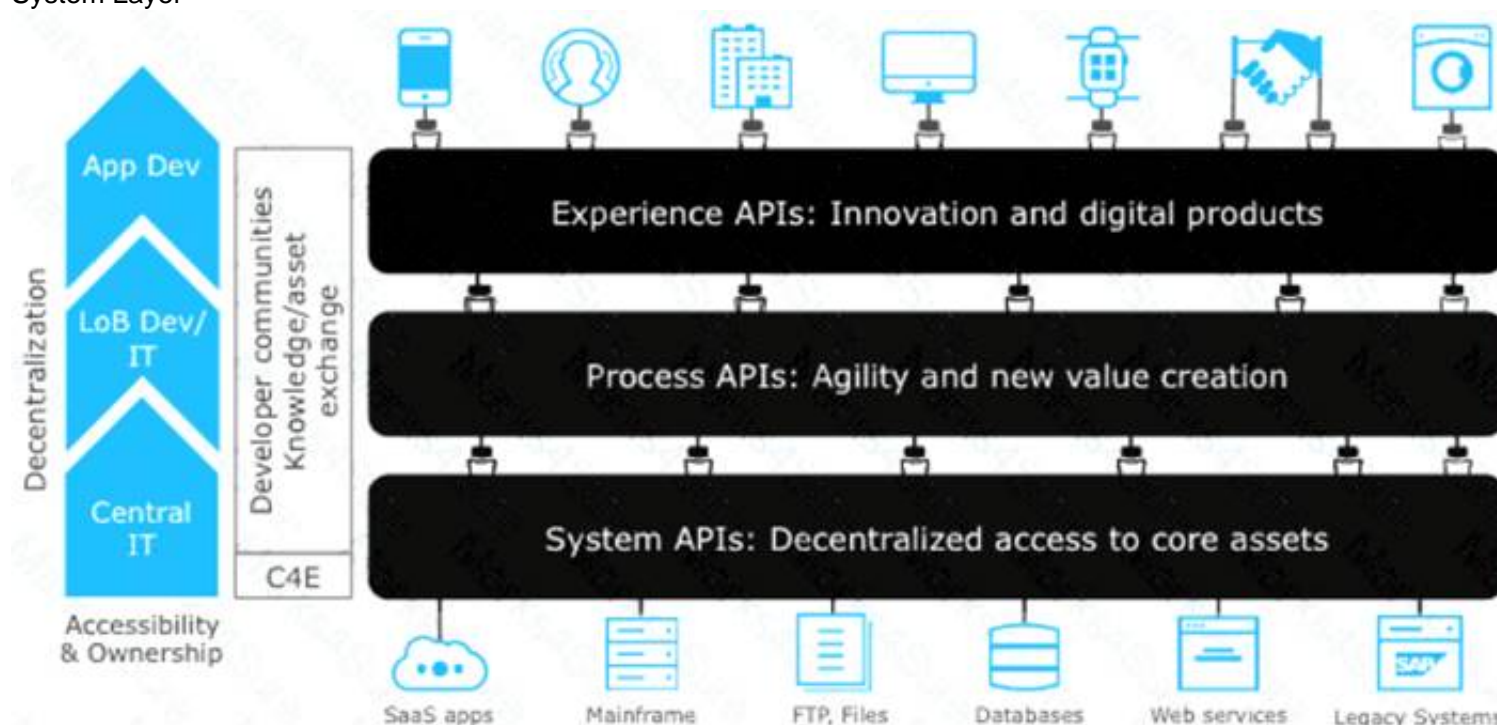
- A. Experience Layer
- B. Process Layer
- C. System Layer

Answer: C

Explanation:

Correct Answer

System Layer



The APIs used in an API-led approach to connectivity fall into three categories:

System APIs – these usually access the core systems of record and provide a means of insulating the user from the complexity or any changes to the underlying systems. Once built, many users, can access data without any need to learn the underlying systems and can reuse these APIs in multiple projects.

Process APIs – These APIs interact with and shape data within a single system or across systems (breaking down data silos) and are created here without a dependence on the source systems from which that data originates, as well as the target channels through which that data is delivered.

Experience APIs – Experience APIs are the means by which data can be reconfigured so that it is most easily consumed by its intended audience, all from a common data source, rather than setting up separate point-to-point integrations for each channel. An Experience API is usually created with API-first design principles where the API is designed for the specific user experience in mind.

NEW QUESTION 62

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 63

What is true about API implementations when dealing with legal regulations that require all data processing to be performed within a certain jurisdiction (such as in the USA or the EU)?

- A. They must avoid using the Object Store as it depends on services deployed ONLY to the US East region
- B. They must use a Jurisdiction-local external messaging system such as Active MQ rather than Anypoint MQ
- C. They must be deployed to Anypoint Platform runtime planes that are managed by Anypoint Platform control planes, with both planes in the same Jurisdiction
- D. They must ensure ALL data is encrypted both in transit and at rest

Answer: C

Explanation:

Correct Answer

They must be deployed to Anypoint Platform runtime planes that are managed by Anypoint Platform control planes, with both planes in the same Jurisdiction.

>> As per legal regulations, all data processing to be performed within a certain jurisdiction. Meaning, the data in USA should reside within USA and should not go out. Same way, the data in EU should reside within EU and should not go out.

>> So, just encrypting the data in transit and at rest does not help to be compliant with the rules. We need to make sure that data does not go out too.

>> The data that we are talking here is not just about the messages that are published to Anypoint MQ. It includes the apps running, transaction states, application logs, events, metric info and any other metadata. So, just replacing Anypoint MQ with a locally hosted ActiveMQ does NOT help.

>> The data that we are talking here is not just about the key/value pairs that are stored in Object Store. It includes the messages published, apps running, transaction states, application logs, events, metric info and any other metadata. So, just avoiding using Object Store does NOT help.

>> The only option left and also the right option in the given choices is to deploy application on runtime and control planes that are both within the jurisdiction.

NEW QUESTION 65

An Anypoint Platform organization has been configured with an external identity provider (IdP) for identity management and client management. What credentials or token must be provided to Anypoint CLI to execute commands against the Anypoint Platform APIs?

- A. The credentials provided by the IdP for identity management
- B. The credentials provided by the IdP for client management
- C. An OAuth 2.0 token generated using the credentials provided by the IdP for client management
- D. An OAuth 2.0 token generated using the credentials provided by the IdP for identity management

Answer: A

Explanation:

Correct Answer

The credentials provided by the IdP for identity management

NEW QUESTION 69

An organization is deploying their new implementation of the OrderStatus System API to multiple workers in CloudHub. This API fronts the organization's on-premises Order Management System, which is accessed by the API implementation over an IPsec tunnel.

What type of error typically does NOT result in a service outage of the OrderStatus System API?

- A. A CloudHub worker fails with an out-of-memory exception
- B. API Manager has an extended outage during the initial deployment of the API implementation
- C. The AWS region goes offline with a major network failure to the relevant AWS data centers
- D. The Order Management System is Inaccessible due to a network outage in the organization's on-premises data center

Answer: A

Explanation:

Correct Answer

A CloudHub worker fails with an out-of-memory exception.

>> An AWS Region itself going down will definitely result in an outage as it does not matter how many workers are assigned to the Mule App as all of those in that region will go down. This is a complete downtime and outage.

>> Extended outage of API manager during initial deployment of API implementation will of course cause issues in proper application startup itself as the API Autodiscovery might fail or API policy templates and policies may not be downloaded to embed at the time of application startup etc... there are many reasons that could cause issues.

>> A network outage on-premises would of course cause the Order Management System not accessible and it does not matter how many workers are assigned to the app they all will fail and cause outage for sure.

The only option that does NOT result in a service outage is if a CloudHub worker fails with an out-of-memory exception. Even if a worker fails and goes down, there are still other workers to handle the requests and keep the API UP and Running. So, this is the right answer.

NEW QUESTION 71

A company requires Mule applications deployed to CloudHub to be isolated between non-production and production environments. This is so Mule applications deployed to non-production environments can only access backend systems running in their customer-hosted non-production environment, and so Mule applications deployed to production environments can only access backend systems running in their customer-hosted production environment. How does MuleSoft recommend modifying Mule applications, configuring environments, or changing infrastructure to support this type of per-environment isolation between Mule applications and backend systems?

- A. Modify properties of Mule applications deployed to the production Anypoint Platform environments to prevent access from non-production Mule applications
- B. Configure firewall rules in the infrastructure inside each customer-hosted environment so that only IP addresses from the corresponding Anypoint Platform environments are allowed to communicate with corresponding backend systems
- C. Create non-production and production environments in different Anypoint Platform business groups
- D. Create separate Anypoint VPCs for non-production and production environments, then configure connections to the backend systems in the corresponding customer-hosted environments

Answer: D

Explanation:

Correct Answer

Create separate Anypoint VPCs for non-production and production environments, then configure connections to the backend systems in the corresponding customer-hosted environments.

>> Creating different Business Groups does NOT make any difference w.r.t accessing the non-prod and prod customer-hosted environments. Still they will be accessing from both Business Groups unless process network restrictions are put in place.

>> We need to modify or couple the Mule Application Implementations with the environment. In fact, we should never implements application coupled with environments by binding them in the properties. Only basic things like endpoint URL etc should be bundled in properties but not environment level access restrictions.

>> IP addresses on CloudHub are dynamic until unless a special static addresses are assigned. So it is not possible to setup firewall rules in customer-hosted infrastrcture. More over, even if static IP addresses are assigned, there could be 100s of applications running on cloudhub and setting up rules for all of them would be a hectic task, non-maintainable and definitely got a good practice.

>> Thbeest practice recommended

by MulesoftIn(fact any cloud provider), is to have your Anypoint VPCs

seperated for Prod and Non-Prod and perform the VPC peering or VPN tunneling for these Anypoint VPCs to respective Prod and Non-Prod customer-hosted environment networks.

NEW QUESTION 74

What is a typical result of using a fine-grained rather than a coarse-grained API deployment model to implement a given business process?

- A. A decrease in the number of connections within the application network supporting the business process
- B. A higher number of discoverable API-related assets in the application network
- C. A better response time for the end user as a result of the APIs being smaller in scope and complexity
- D. An overall tower usage of resources because each fine-grained API consumes less resources

Answer: B

Explanation:

Correct Answer

A higher number of discoverable API-related assets in the application network.

>> We do NOT get faster response times in fine-grained approach when compared to coarse-grained approach.

>> In fact, we get faster response times from a network having coarse-grained APIs compared to a network having fine-grained APIs model. The reasons are below.

Fine-grained approach:

* 1. will have more APIs compared to coarse-grained

* 2. So, more orchestration needs to be done to achieve a functionality in business process.

* 3. Which means, lots of API calls to be made. So, more connections will needs to be established. So, obviously more hops, more network i/o, more number of integration points compared to coarse-grained approach where fewer APIs with bulk functionality embedded in them.

* 4. That is why, because of all these extra hops and added latencies, fine-grained approach will have bit more response times compared to coarse-grained.

* 5. Not only added latencies and connections, there will be more resources used up in fine-grained approach due to more number of APIs.

That's why, fine-grained APIs are good in a way to expose more number of resuable assets in your network and make them discoverable. However, needs more maintenance, taking care of integration points, connections, resources with a little compromise w.r.t network hops and response times.

NEW QUESTION 78

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 80

A system API has a guaranteed SLA of 100 ms per request. The system API is deployed to a primary environment as well as to a disaster recovery (DR) environment, with different DNS names in each environment. An upstream process API invokes the system API and the main goal of this process API is to

respond to client requests in the least possible time. In what order should the system APIs be invoked, and what changes should be made in order to speed up the response time for requests from the process API?

- A. In parallel, invoke the system API deployed to the primary environment and the system API deployed to the DR environment, and ONLY use the first response
- B. In parallel, invoke the system API deployed to the primary environment and the system API deployed to the DR environment using a scatter-gather configured with a timeout, and then merge the responses
- C. Invoke the system API deployed to the primary environment, and if it fails, invoke the system API deployed to the DR environment
- D. Invoke ONLY the system API deployed to the primary environment, and add timeout and retry logic to avoid intermittent failures

Answer: A

Explanation:

Correct Answer

In parallel, invoke the system API deployed to the primary environment and the system API deployed to the DR environment, and ONLY use the first response.

>> The API requirement in the given scenario is to respond in least possible time.

>> The option that is suggesting to first try the API in primary environment and then fallback to API in DR environment would result in successful response but NOT in least possible time. So, this is NOT a right choice of implementation for given requirement.

>> Another option that is suggesting to ONLY invoke API in primary environment and to add timeout and retries may also result in successful response upon retries but NOT in least possible time. So, this is also NOT a right choice of implementation for given requirement.

>> One more option that is suggesting to invoke API in primary environment and API in DR environment in parallel using Scatter-Gather would result in wrong API response as it would return merged results and moreover, Scatter-Gather does things in parallel which is true but still completes its scope only on finishing all routes inside it. So again, NOT a right choice of implementation for given requirement

The Correct choice is to invoke the API in primary environment and the API in DR environment parallelly, and using ONLY the first response received from one of them.

NEW QUESTION 81

A system API is deployed to a primary environment as well as to a disaster recovery (DR) environment, with different DNS names in each environment. A process API is a client to the system API and is being rate limited by the system API, with different limits in each of the environments. The system API's DR environment provides only 20% of the rate limiting offered by the primary environment. What is the best API fault-tolerant invocation strategy to reduce overall errors in the process API, given these conditions and constraints?

- A. Invoke the system API deployed to the primary environment; add timeout and retry logic to the process API to avoid intermittent failures; if it still fails, invoke the system API deployed to the DR environment
- B. Invoke the system API deployed to the primary environment; add retry logic to the process API to handle intermittent failures by invoking the system API deployed to the DR environment
- C. In parallel, invoke the system API deployed to the primary environment and the system API deployed to the DR environment; add timeout and retry logic to the process API to avoid intermittent failures; add logic to the process API to combine the results
- D. Invoke the system API deployed to the primary environment; add timeout and retry logic to the process API to avoid intermittent failures; if it still fails, invoke a copy of the process API deployed to the DR environment

Answer: A

Explanation:

Correct Answer

Invoke the system API deployed to the primary environment; add timeout and retry logic to the process API to avoid intermittent failures; if it still fails, invoke the system API deployed to the DR environment

There is one important consideration to be noted in the question which is - System API in DR environment provides only 20% of the rate limiting offered by the primary environment. So, comparatively, very less calls will be allowed into the DR environment API opposed to its primary environment. With this in mind, let's analyse what is the right and best fault-tolerant invocation strategy.

* 1. Invoking both the system APIs in parallel is definitely NOT a feasible approach because of the 20% limitation we have on DR environment. Calling in parallel every time would easily and quickly exhaust the rate limits on DR environment and may not give chance to genuine intermittent error scenarios to let in during the time of need.

* 2. Another option given is suggesting to add timeout and retry logic to process API while invoking primary environment's system API. This is good so far.

However, when all retries failed, the option is suggesting to invoke the copy of process API on DR environment which is not right or recommended. Only system API is the one to be considered for fallback and not the whole process API. Process APIs usually have lot of heavy orchestration calling many other APIs which we do not want to repeat again by calling DR's process API. So this option is NOT right.

* 3. One more option given is suggesting to add the retry (no timeout) logic to process API to directly retry on DR environment's system API instead of retrying the primary environment system API first. This is not at all a proper fallback. A proper fallback should occur only after all retries are performed and exhausted on Primary environment first. But here, the option is suggesting to directly retry fallback API on first failure itself without trying main API. So, this option is NOT right too.

This leaves us one option which is right and best fit.

- Invoke the system API deployed to the primary environment
- Add Timeout and Retry logic on it in process API
- If it fails even after all retries, then invoke the system API deployed to the DR environment.

NEW QUESTION 82

.....

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