

MuleSoft

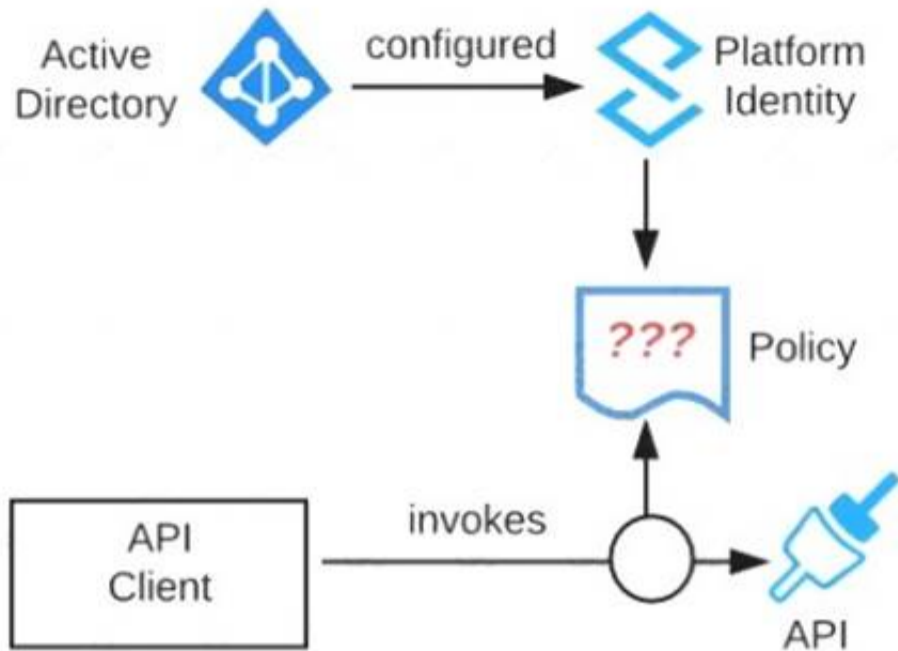
Exam Questions MCPA-Level-1

MuleSoft Certified Platform Architect - Level 1



NEW QUESTION 1

Refer to the exhibit. An organization is running a Mule standalone runtime and has configured Active Directory as the Anypoint Platform external Identity Provider. The organization does not have budget for other system components.



What policy should be applied to all instances of APIs in the organization to most effectively restrict access to a specific group of internal users?

- A. Apply a basic authentication - LDAP policy; the internal Active Directory will be configured as the LDAP source for authenticating users
- B. Apply a client ID enforcement policy; the specific group of users will configure their client applications to use their specific client credentials
- C. Apply an IP whitelist policy; only the specific users' workstations will be in the whitelist
- D. Apply an OAuth 2.0 access token enforcement policy; the internal Active Directory will be configured as the OAuth server

Answer: A

Explanation:

Correct Answer

Apply a basic authentication - LDAP policy; the internal Active Directory will be configured as the LDAP source for authenticating users.

>> IP Whitelisting does NOT fit for this purpose. Moreover, the users workstations may not necessarily have static IPs in the network.

>> OAuth 2.0 enforcement requires a client provider which isn't in the organizations system components.

>> It is not an effective approach to let every user create separate client credentials and configure those for their usage.

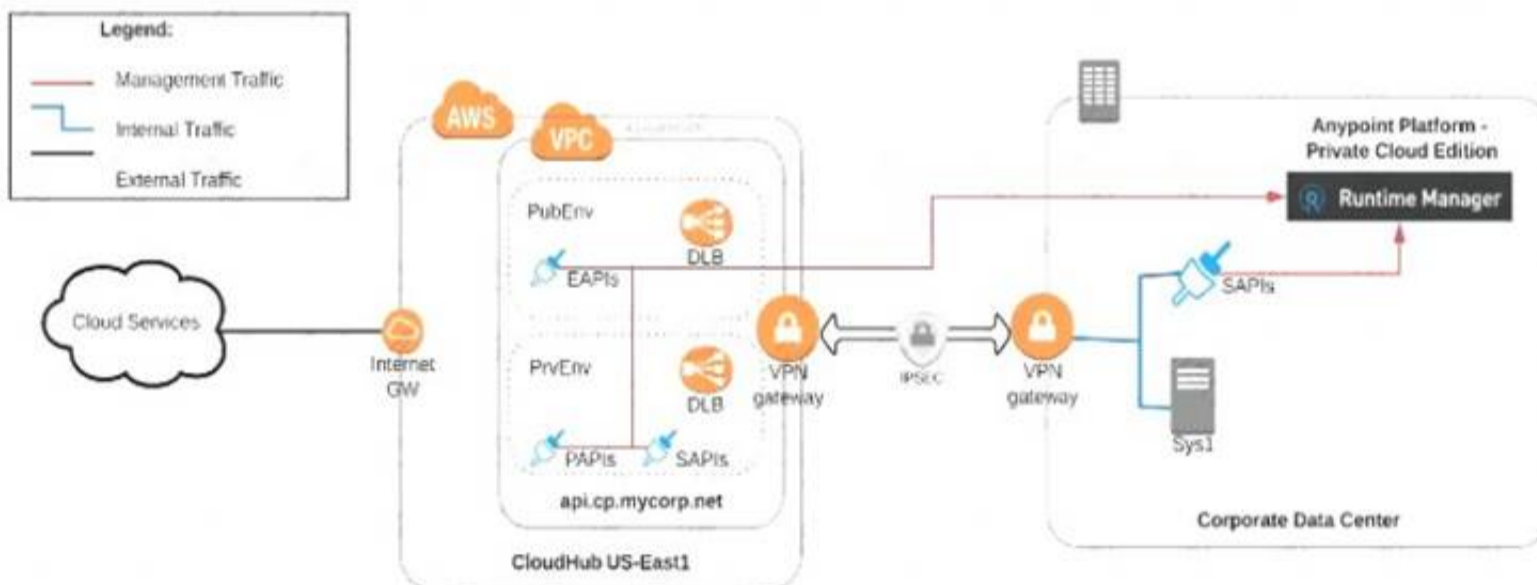
The effective way it to apply a basic authentication - LDAP policy and the internal Active Directory will be configured as the LDAP source for authenticating users.

NEW QUESTION 2

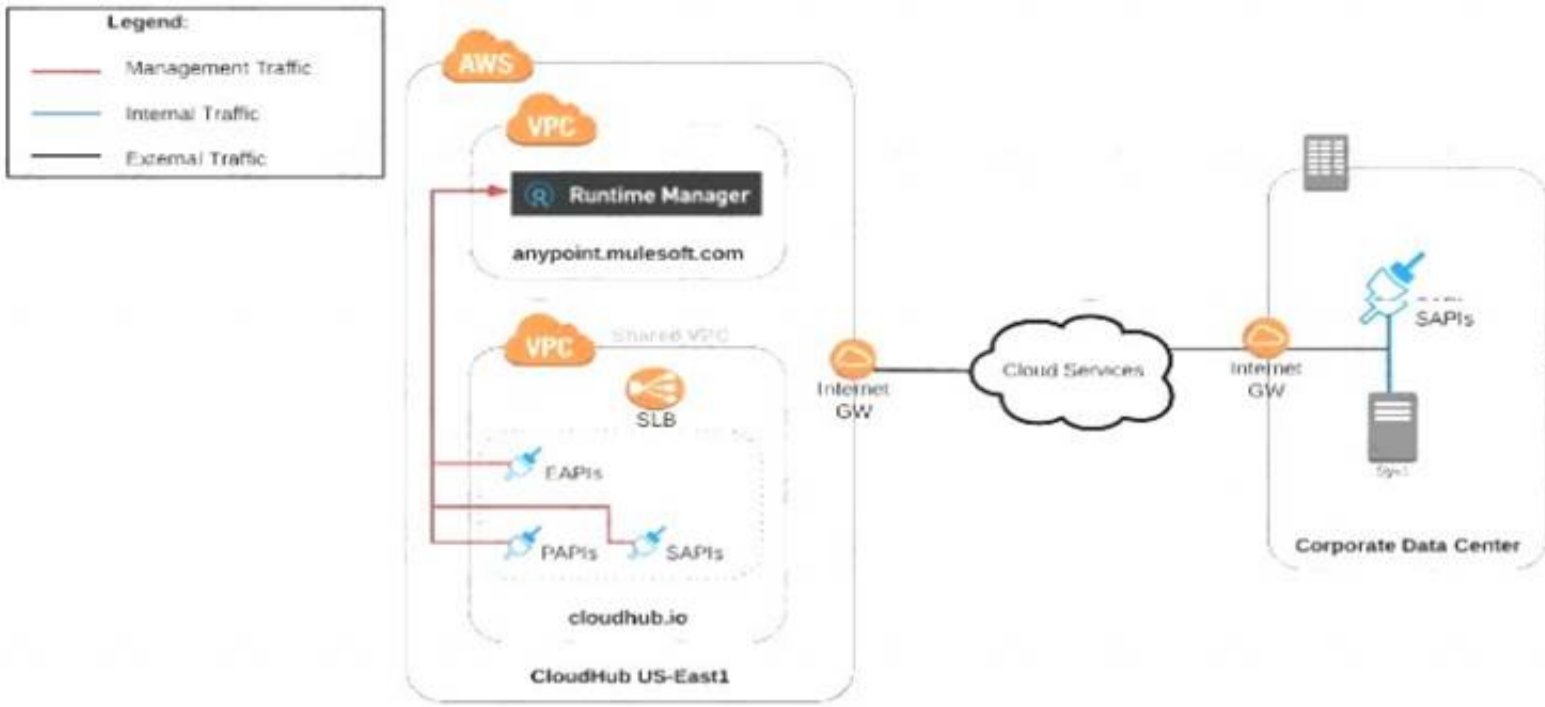
An organization uses various cloud-based SaaS systems and multiple on-premises systems. The on-premises systems are an important part of the organization's application network and can only be accessed from within the organization's intranet.

What is the best way to configure and use Anypoint Platform to support integrations with both the cloud-based SaaS systems and on-premises systems?

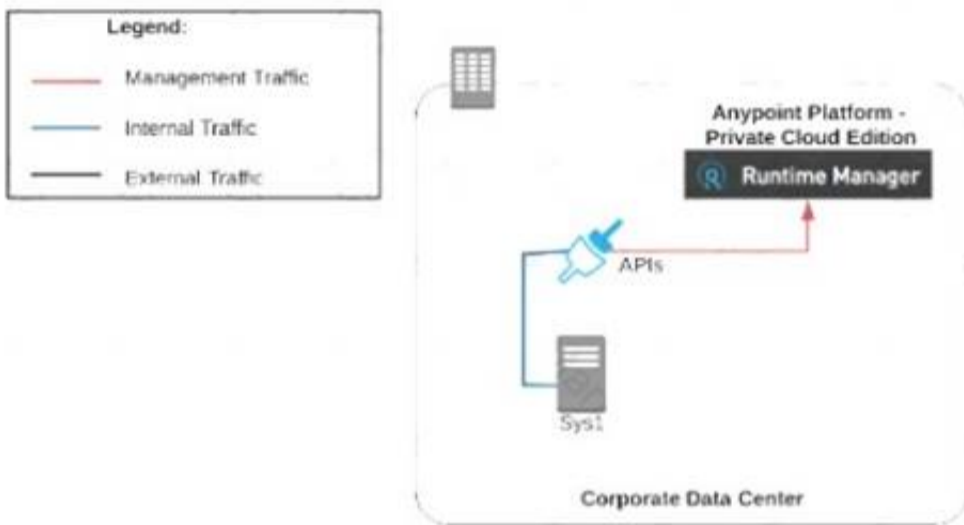
- A) Use CloudHub-deployed Mule runtimes in an Anypoint VPC managed by Anypoint Platform Private Cloud Edition control plane



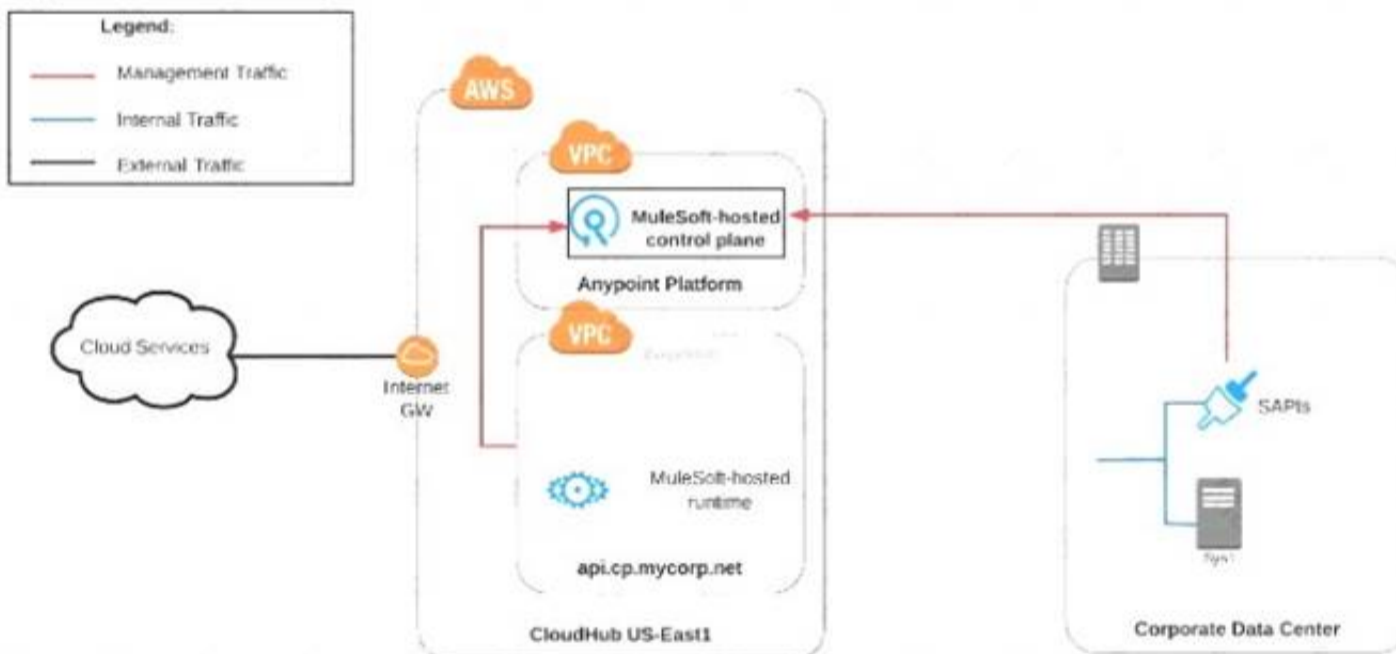
- B) Use CloudHub-deployed Mule runtimes in the shared worker cloud managed by the MuleSoft-hosted Anypoint Platform control plane



C) Use an on-premises installation of Mule runtimes that are completely isolated with NO external network access, managed by the Anypoint Platform Private Cloud Edition control plane



D) Use a combination of Cloud Hub-deployed and manually provisioned on-premises Mule runtimes managed by the MuleSoft-hosted Anypoint Platform control plane



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

Answer: B

Explanation:

Correct Answer

Use a combination of CloudHub-deployed and manually provisioned on-premises Mule runtimes managed by the MuleSoft-hosted Platform control plane.

***** Key details to be taken from the given scenario:

- >> Organization uses BOTH cloud-based and on-premises systems
- >> On-premises systems can only be accessed from within the organization's intranet Let us evaluate the given choices based on above key details:
- >> CloudHub-deployed Mule runtimes can ONLY be controlled using MuleSoft-hosted control plane. We CANNOT use Private Cloud Edition's control plane to control CloudHub Mule Runtimes. So, option suggesting this is INVALID
- >> Using CloudHub-deployed Mule runtimes in the shared worker cloud managed by the MuleSoft-hosted Anypoint Platform is completely IRRELEVANT to given scenario and silly choice. So, option suggesting this is INVALID
- >> Using an on-premises installation of Mule runtimes that are completely isolated with NO external network access, managed by the Anypoint Platform Private Cloud Edition control plane would work for On-premises integrations. However, with NO external access, integrations cannot be done to SaaS-based apps.

Moreover CloudHub-hosted apps are best-fit for integrating with SaaS-based applications. So, option suggesting this is BEST WAY. The best way to configure and use Anypoint Platform to support these mixed/hybrid integrations is to use a combination of CloudHub-deployed and manually provisioned on-premises Mule runtimes managed by the MuleSoft-hosted Platform control plane.

NEW QUESTION 3

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 4

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 5

A code-centric API documentation environment should allow API consumers to investigate and execute API client source code that demonstrates invoking one or more APIs as part of representative scenarios.

What is the most effective way to provide this type of code-centric API documentation environment using Anypoint Platform?

- A. Enable mocking services for each of the relevant APIs and expose them via their Anypoint Exchange entry
- B. Ensure the APIs are well documented through their Anypoint Exchange entries and API Consoles and share these pages with all API consumers
- C. Create API Notebooks and include them in the relevant Anypoint Exchange entries
- D. Make relevant APIs discoverable via an Anypoint Exchange entry

Answer: C

Explanation:

Correct Answer

Create API Notebooks and Include them in the relevant Anypoint exchange entries

>> API Notebooks are the one on Anypoint Platform that enable us to provide code-centric API documentation

NEW QUESTION 6

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:

Correct Answer

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

- >> There are NO fixed and straight best practices while opting data models for APIs. They are completely contextual and depends on number of factors. Based upon those factors, an enterprise can choose if they have to go with Enterprise Canonical Data Model or Bounded Context Model etc.
- >> One should NEVER expose the technical details of API implementation to their API clients. Only the API interface/ RAML is exposed to API clients.
- >> It is true that the RAML definitions of APIs should be as detailed as possible and should reflect most of the documentation. However, just that is NOT enough to call your API as best documented API. There should be even more documentation on Anypoint Exchange with API Notebooks etc. to make and create a developer friendly API and repository..
- >> The best practice always when creating System APIs is to create their API interfaces by modeling their resources and methods to closely reflect the operations and functionalities of that backend system.

NEW QUESTION 7

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 8

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 9

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 10

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.

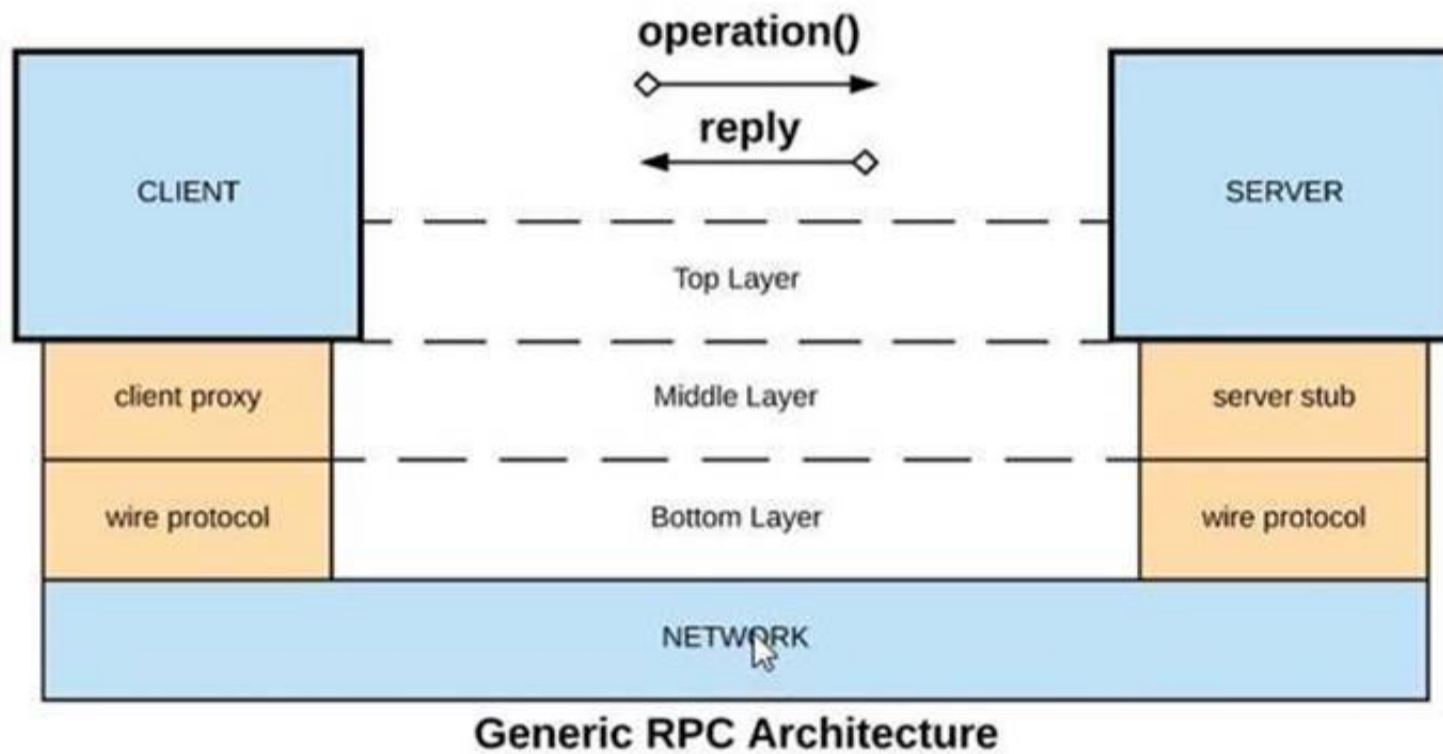
For example, a user of your API may receive a response that includes these headers:

```
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 11

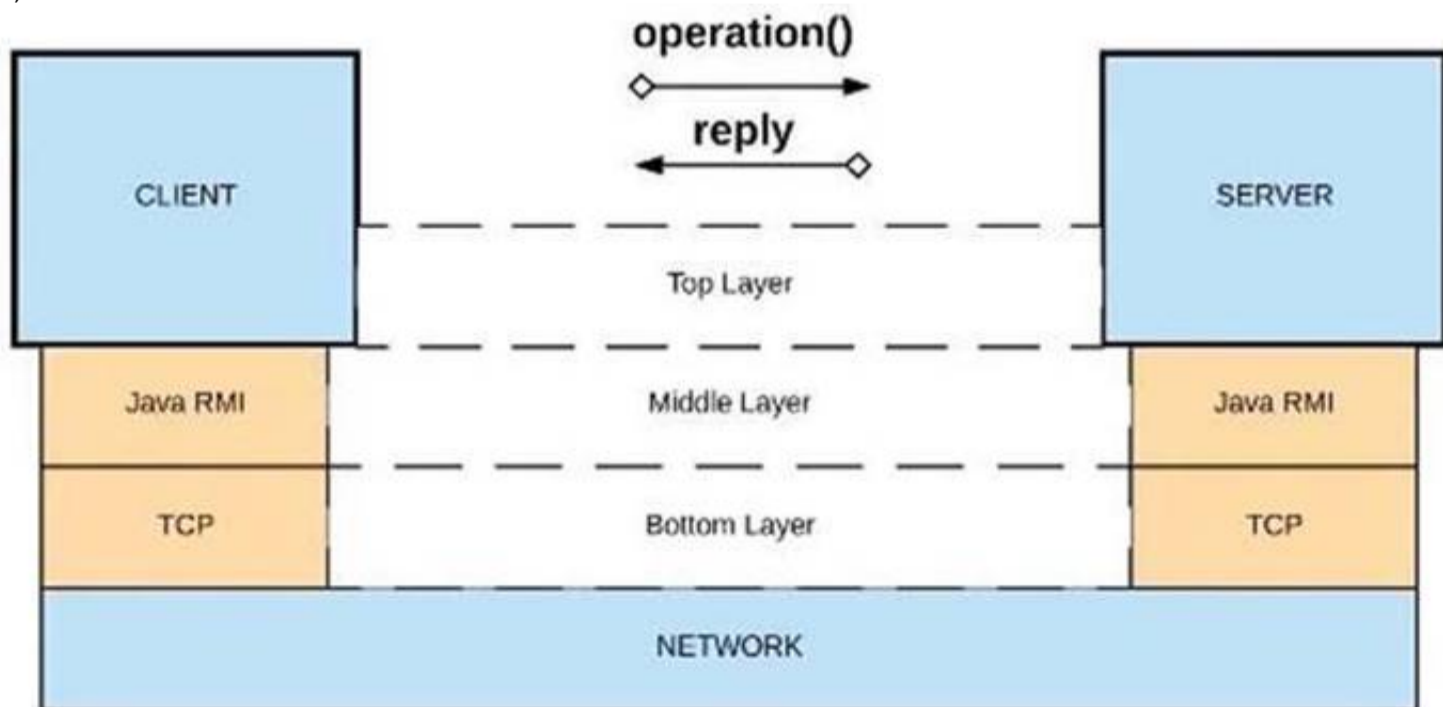
Refer to the exhibit.



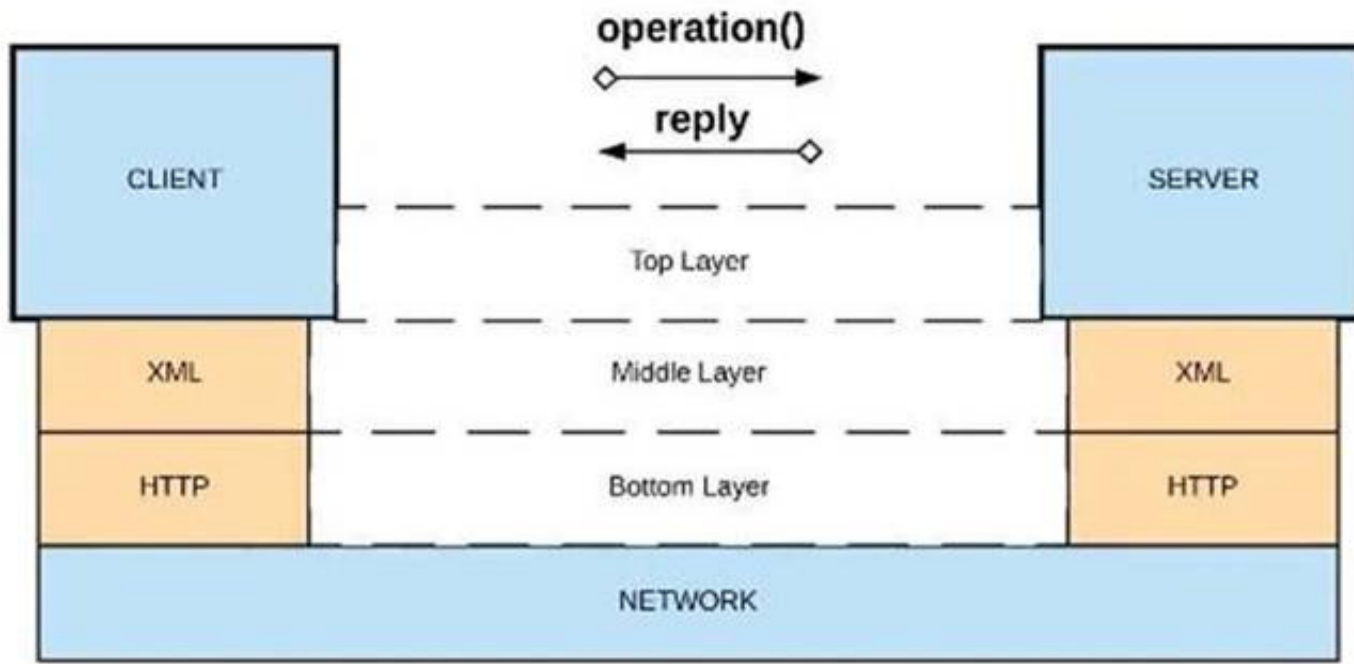
Generic RPC Architecture

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

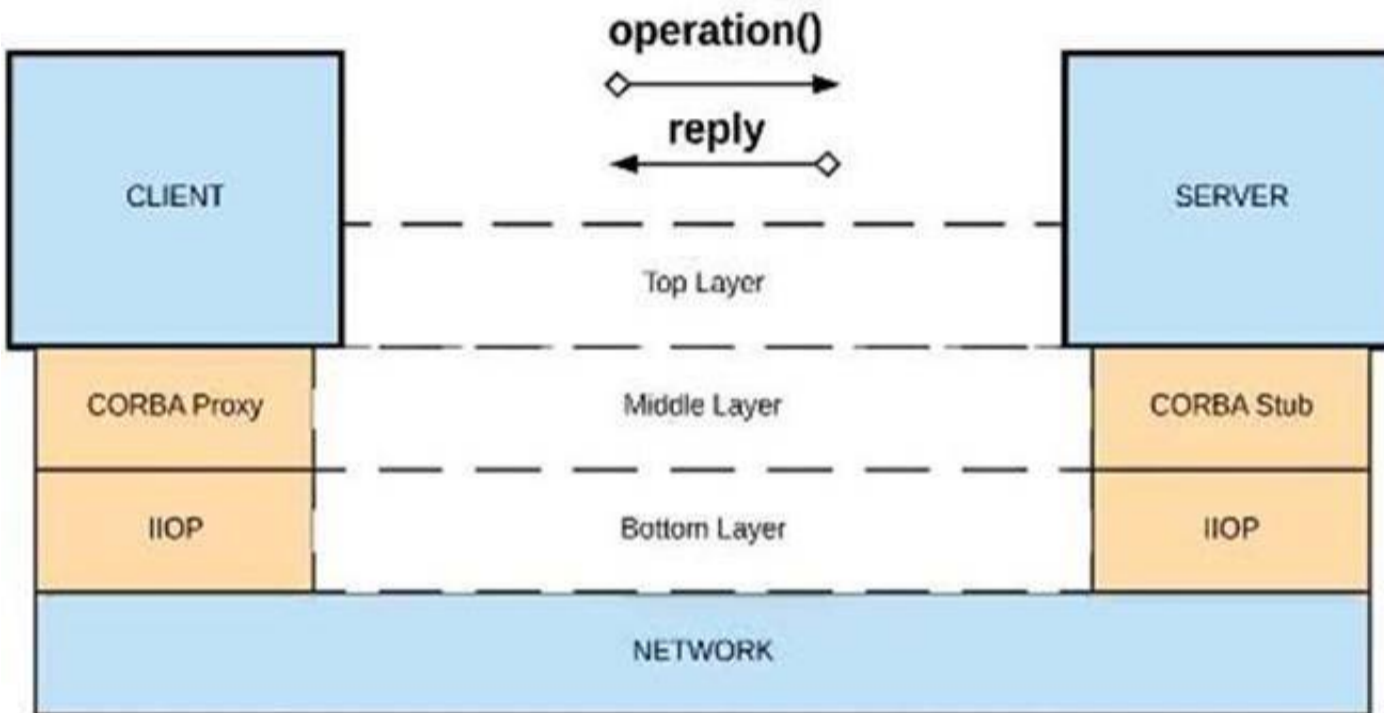
A) Java RMI over TCP



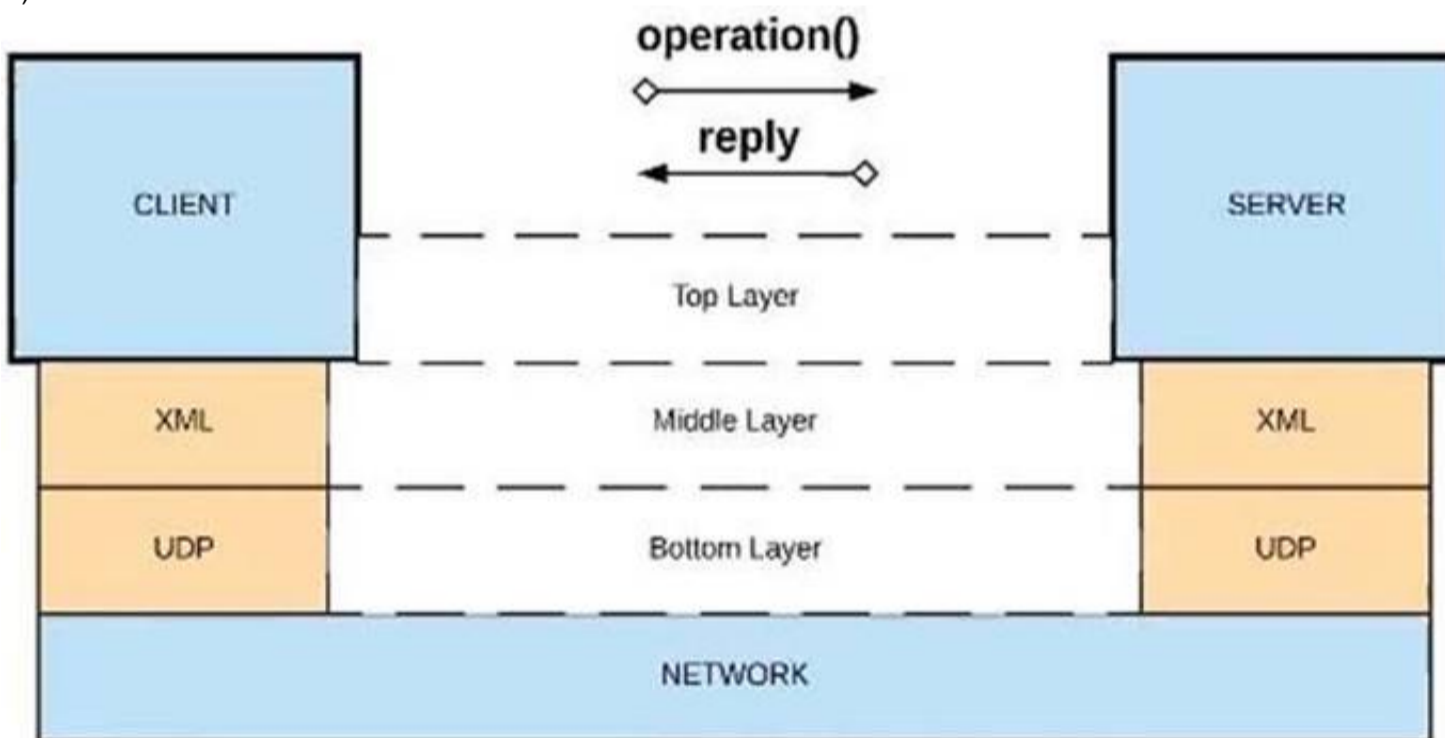
B) Java RMI over TCP



C) CORBA over IIOB



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 12

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 13

A new upstream API is being designed to offer an SLA of 500 ms median and 800 ms maximum (99th percentile) response time. The corresponding API implementation needs to sequentially invoke 3 downstream APIs of very similar complexity.

The first of these downstream APIs offers the following SLA for its response time: median: 100 ms, 80th percentile: 500 ms, 95th percentile: 1000 ms.

If possible, how can a timeout be set in the upstream API for the invocation of the first downstream API to meet the new upstream API's desired SLA?

- A. Set a timeout of 50 ms; this times out more invocations of that API but gives additional room for retries
- B. Set a timeout of 100 ms; that leaves 400 ms for the other two downstream APIs to complete
- C. No timeout is possible to meet the upstream API's desired SLA; a different SLA must be negotiated with the first downstream API or invoke an alternative API
- D. Do not set a timeout; the invocation of this API is mandatory and so we must wait until it responds

Answer: B

Explanation:

Correct Answer

Set a timeout of 100ms; that leaves 400ms for other two downstream APIs to complete

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

>> Upstream API's designed SLA is 500ms (median). Lets ignore maximum SLA response times.

>> This API calls 3 downstream APIs sequentially and all these are of similar complexity.

>> The first downstream API is offering median SLA of 100ms, 80th percentile: 500ms; 95th percentile: 1000ms.

Based on the above details:

>> We can rule out the option which is suggesting to set 50ms timeout. Because, if the median SLA itself being offered is 100ms then most of the calls are going to timeout and time gets wasted in retried them and eventually gets exhausted with all retries. Even if some retries gets successful, the remaining time wont leave enough room for 2nd and 3rd downstream APIs to respond within time.

>> The option suggesting to NOT set a timeout as the invocation of this API is mandatory and so we must wait until it responds is silly. As not setting time out would go against the good implementation pattern and moreover if the first API is not responding within its offered median SLA 100ms then most probably it would either respond in 500ms (80th percentile) or 1000ms (95th percentile). In BOTH cases, getting a successful response from 1st downstream API does NO GOOD because already by this time the Upstream API SLA of 500 ms is breached. There is no time left to call 2nd and 3rd downstream APIs.

>> It is NOT true that no timeout is possible to meet the upstream APIs desired SLA.

As 1st downstream API is offering its median SLA of 100ms, it means MOST of the time we would get the responses within that time. So, setting a timeout of 100ms would be ideal for MOST calls as it leaves enough room of 400ms for remaining 2 downstream API calls.

NEW QUESTION 14

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 15

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 16

Select the correct Owner-Layer combinations from below options

- A. * 1. App Developers owns and focuses on Experience Layer APIs* 2. Central IT owns and focuses on Process Layer APIs* 3. LOB IT owns and focuses on System Layer APIs
- B. * 1. Central IT owns and focuses on Experience Layer APIs* 2. LOB IT owns and focuses on Process Layer APIs* 3. App Developers owns and focuses on System Layer APIs
- C. * 1. App Developers owns and focuses on Experience Layer APIs* 2. LOB IT owns and focuses on Process Layer APIs* 3. Central IT owns and focuses on System Layer APIs

Answer: C

Explanation:

Correct Answer

- * 1. App Developers owns and focuses on Experience Layer APIs
- * 2. LOB IT owns and focuses on Process Layer APIs
- * 3. Central IT owns and focuses on System Layer APIs

References:

<https://blogs.mulesoft.com/biz/api/experience-api-ownership/> <https://blogs.mulesoft.com/biz/api/process-api-ownership/> <https://blogs.mulesoft.com/biz/api/system-api-ownership/>

NEW QUESTION 17

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 APIs 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 API producer should be requested to run the old version in parallel with the new one
- B. The API producer should be contacted to understand the change to existing functionality
- C. The API client code only needs to be changed if it needs to take advantage of the new features
- D. The API clients need to update the code on their side and need to do full regression

Answer: C

NEW QUESTION 18

A company wants to move its Mule API implementations into production as quickly as possible. To protect access to all Mule application data and metadata, the company requires that all Mule applications be deployed to the company's customer-hosted infrastructure within the corporate firewall. What combination of runtime plane and control plane options meets these project lifecycle goals?

- A. Manually provisioned customer-hosted runtime plane and customer-hosted control plane
- B. MuleSoft-hosted runtime plane and customer-hosted control plane
- C. Manually provisioned customer-hosted runtime plane and MuleSoft-hosted control plane
- D. iPaaS provisioned customer-hosted runtime plane and MuleSoft-hosted control plane

Answer: A

Explanation:

Correct Answer

Manually provisioned customer-hosted runtime plane and customer-hosted control plane

There are two key factors that are to be taken into consideration from the scenario given in the question.

>> Company requires both data and metadata to be resided within the corporate firewall
>> Company would like to go with customer-hosted infrastructure.

Any deployment model that is to deal with the cloud directly or indirectly (Mulesoft-hosted or Customer's own cloud like Azure, AWS) will have to share atleast the metadata.

Application data can be controlled inside firewall by having Mule Runtimes on customer hosted runtime plane. But if we go with Mulesoft-hosted/ Cloud-based control plane, the control plane required atleast some minimum level of metadata to be sent outside the corporate firewall.

As the customer requirement is pretty clear about the data and metadata both to be within the corporate firewall, even though customer wants to move to production as quickly as possible, unfortunately due to the nature of their security requirements, they have no other option but to go with manually provisioned customer-hosted runtime plane and customer-hosted control plane.

NEW QUESTION 19

The implementation of a Process API must change.

What is a valid approach that minimizes the impact of this change on API clients?

- A. Update the RAML definition of the current Process API and notify API client developers by sending them links to the updated RAML definition
- B. Postpone changes until API consumers acknowledge they are ready to migrate to a new Process API or API version
- C. Implement required changes to the Process API implementation so that whenever possible, the Process API's RAML definition remains unchanged
- D. Implement the Process API changes in a new API implementation, and have the old API implementation return an HTTP status code 301 - Moved Permanently to inform API clients they should be calling the new API implementation

Answer: C

Explanation:

Correct Answer

Implement required changes to the Process API implementation so that, whenever possible, the Process API's RAML definition remains unchanged.

***** Key requirement in the question is:

>> Approach that minimizes the impact of this change on API clients Based on above:

>> Updating the RAML definition would possibly impact the API clients if the changes require any thing mandatory from client side. So, one should try to avoid doing that until really necessary.

>> Implementing the changes as a completely different API and then redirectly the clients with 3xx status code is really upsetting design and heavily impacts the API clients.

>> Organisations and IT cannot simply postpone the changes required until all API consumers acknowledge they are ready to migrate to a new Process API or API version. This is unrealistic and not possible.

The best way to handle the changes always is to implement required changes to the API implementations so that, whenever possible, the API's RAML definition remains unchanged.

NEW QUESTION 20

.....

Thank You for Trying Our Product

We offer two products:

1st - We have Practice Tests Software with Actual Exam Questions

2nd - Questions and Answers in PDF Format

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

100% Actual & Verified — Instant Download, Please Click
[Order The MCPA-Level-1 Practice Test Here](#)