

Exam Questions MCPA-Level-1

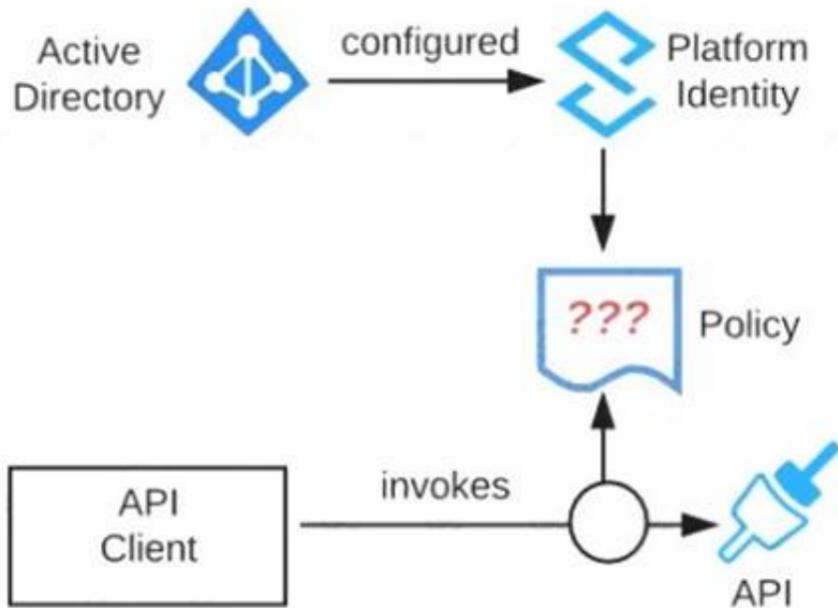
MuleSoft Certified Platform Architect - Level 1

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

True or False. We should always make sure that the APIs being designed and developed are self-servable even if it needs more man-day effort and resources.

- A. FALSE
- B. TRUE

Answer: B

Explanation:

Correct Answer

TRUE

>> As per MuleSoft proposed IT Operating Model, designing APIs and making sure that they are discoverable and self-servable is VERY VERY IMPORTANT and decides the success of an API and its application network.

NEW QUESTION 3

A company has created a successful enterprise data model (EDM). The company is committed to building an application network by adopting modern APIs as a core enabler of the company's IT operating model. At what API tiers (experience, process, system) should the company require reusing the EDM when designing modern API data models?

- A. At the experience and process tiers
- B. At the experience and system tiers
- C. At the process and system tiers
- D. At the experience, process, and system tiers

Answer: C

Explanation:

Correct Answer

At the process and system tiers

>> Experience Layer APIs are modeled and designed exclusively for the end user's experience. So, the data models of experience layer vary based on the nature and type of such API consumer. For example, Mobile consumers will need light-weight data models to transfer with ease on the wire, where as web-based consumers will need detailed data models to render most of the info on web pages, so on. So, enterprise data models fit for the purpose of canonical models but not of good use for experience APIs.

>> That is why, EDMs should be used extensively in process and system tiers but NOT in experience tier.

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

Which of the following best fits the definition of API-led connectivity?

- A. API-led connectivity is not just an architecture or technology but also a way to organize people and processes for efficient IT delivery in the organization
- B. API-led connectivity is a 3-layered architecture covering Experience, Process and System layers
- C. API-led connectivity is a technology which enabled us to implement Experience, Process and System layer based APIs

Answer: A

Explanation:

Correct Answer

API-led connectivity is not just an architecture or technology but also a way to organize people and processes for efficient IT delivery in the organization.

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 as various operations/resources
- B. Implement different system APIs named createCustomer, amendCustomer, retrieveCustomer and suspendCustomer as they are modular and has separation of concerns
- C. Implement different system APIs named createCustomerInCRMZ, amendCustomerInCRMZ, retrieveCustomerFromCRMZ and suspendCustomerInCRMZ as they are modular and has separation of concerns

Answer: B

Explanation:

Correct Answer

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

What are the major benefits of MuleSoft proposed IT Operating Model?

- A. * 1. Decrease the IT delivery gap* 2. Meet various business demands without increasing the IT capacity* 3. Focus on creation of reusable assets first
- B. Upon finishing creation of all the possible assets then inform the LOBs in the organization to start using them
- C. * 1. Decrease the IT delivery gap* 2. Meet various business demands by increasing the IT capacity and forming various IT departments* 3. Make consumption of assets at the rate of production
- D. * 1. Decrease the IT delivery gap* 2. Meet various business demands without increasing the IT capacity* 3. Make consumption of assets at the rate of production

Answer: C

Explanation:

Correct Answer

- * 1. Decrease the IT delivery gap
 - * 2. Meet various business demands without increasing the IT capacity
 - * 3. Make consumption of assets at the rate of production.
- *****

NEW QUESTION 8

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:

- o X-Ratelimit-Remaining: The amount of available quota.
- o X-Ratelimit-Limit: The maximum available requests per window.
- o 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 9

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 with 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 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 10

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 day 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 14

An organization is implementing a Quote of the Day API that caches today's quote.

What scenario can use the GitHub Object Store via the Object Store connector to persist the cache's state?

- A. When there are three CloudHub deployments of the API implementation to three separate CloudHub regions that must share the cache state
- B. When there are two CloudHub deployments of the API implementation by two Anypoint Platform business groups to the same CloudHub region that must share the cache state
- C. When there is one deployment of the API implementation to CloudHub and an on-premise deployment to a customer-hosted Mule runtime that must share the cache state
- D. When there is one CloudHub deployment of the API implementation to three CloudHub workers that must share the cache state

Answer: D

Explanation:

Correct Answer

When there is one CloudHub deployment of the API implementation to three CloudHub workers that must share the cache state.

***** Key details in the scenario:

>> Use the CloudHub Object Store via the Object Store connector Considering above details:

>> CloudHub Object Stores have one-to-one relationship with CloudHub Mule Applications.

>> We CANNOT use an application's CloudHub Object Store to be shared among multiple Mule applications running in different Regions or Business Groups or Customer-hosted Mule Runtimes by using Object Store connector.

>> If it is really necessary and very badly needed, then Anypoint Platform supports a way by allowing access to CloudHub Object Store of another application using Object Store REST API. But NOT using Object Store connector.

So, the only scenario where we can use the CloudHub Object Store via the Object Store connector to persist the cache's state is when there is one CloudHub deployment of the API implementation to multiple CloudHub workers that must share the cache state.

NEW QUESTION 17

A Mule application exposes an HTTPS endpoint and is deployed to three CloudHub workers that do not use static IP addresses. The Mule application expects a high volume of client requests in short time periods. What is the most cost-effective infrastructure component that should be used to serve the high volume of client requests?

- A. A customer-hosted load balancer
- B. The CloudHub shared load balancer
- C. An API proxy
- D. Runtime Manager autoscaling

Answer: B

Explanation:

Correct Answer

The CloudHub shared load balancer

***** The scenario in this question can be split as below:

- >> There are 3 CloudHub workers (So, there are already good number of workers to handle high volume of requests)
- >> The workers are not using static IP addresses (So, one CANNOT use customer load-balancing solutions without static IPs)
- >> Looking for most cost-effective component to load balance the client requests among the workers. Based on the above details given in the scenario:
- >> Runtime autoscaling is NOT at all cost-effective as it incurs extra cost. Most over, there are already 3 workers running which is a good number.
- >> We cannot go for a customer-hosted load balancer as it is also NOT most cost-effective (needs custom load balancer to maintain and licensing) and same time the Mule App is not having Static IP Addresses which limits from going with custom load balancing.
- >> An API Proxy is irrelevant there as it has no role to play w.r.t handling high volumes or load balancing. So, the only right option to go with and fits the purpose of scenario being most cost-effective is - using a CloudHub Shared Load Balancer.

NEW QUESTION 21

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 thr 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 ✕	▼
	<input checked="" type="checkbox"/> 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 23

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 24

An API experiences a high rate of client requests (TPS) vwth small message payloads. How can usage limits be imposed on the API based on the type of client

application?

- A. Use an SLA-based rate limiting policy and assign a client application to a matching SLA tier based on its type
- B. Use a spike control policy that limits the number of requests for each client application type
- C. Use a cross-origin resource sharing (CORS) policy to limit resource sharing between client applications, configured by the client application type
- D. Use a rate limiting policy and a client ID enforcement policy, each configured by the client application type

Answer: A

Explanation:

Correct Answer

Use an SLA-based rate limiting policy and assign a client application to a matching SLA tier based on its type.

>> SLA tiers will come into play whenever any limits to be imposed on APIs based on client type

NEW QUESTION 26

Traffic is routed through an API proxy to an API implementation. The API proxy is managed by API Manager and the API implementation is deployed to a CloudHub VPC using Runtime Manager. API policies have been applied to this API. In this deployment scenario, at what point are the API policies enforced on incoming API client requests?

- A. At the API proxy
- B. At the API implementation
- C. At both the API proxy and the API implementation
- D. At a MuleSoft-hosted load balancer

Answer: A

Explanation:

Correct Answer

At the API proxy

>> API Policies can be enforced at two places in Mule platform.

>> One - As an Embedded Policy enforcement in the same Mule Runtime where API implementation is running.

>> Two - On an API Proxy sitting in front of the Mule Runtime where API implementation is running.

>> As the deployment scenario in the question has API Proxy involved, the policies will be enforced at the API Proxy.

NEW QUESTION 27

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 30

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 31

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.

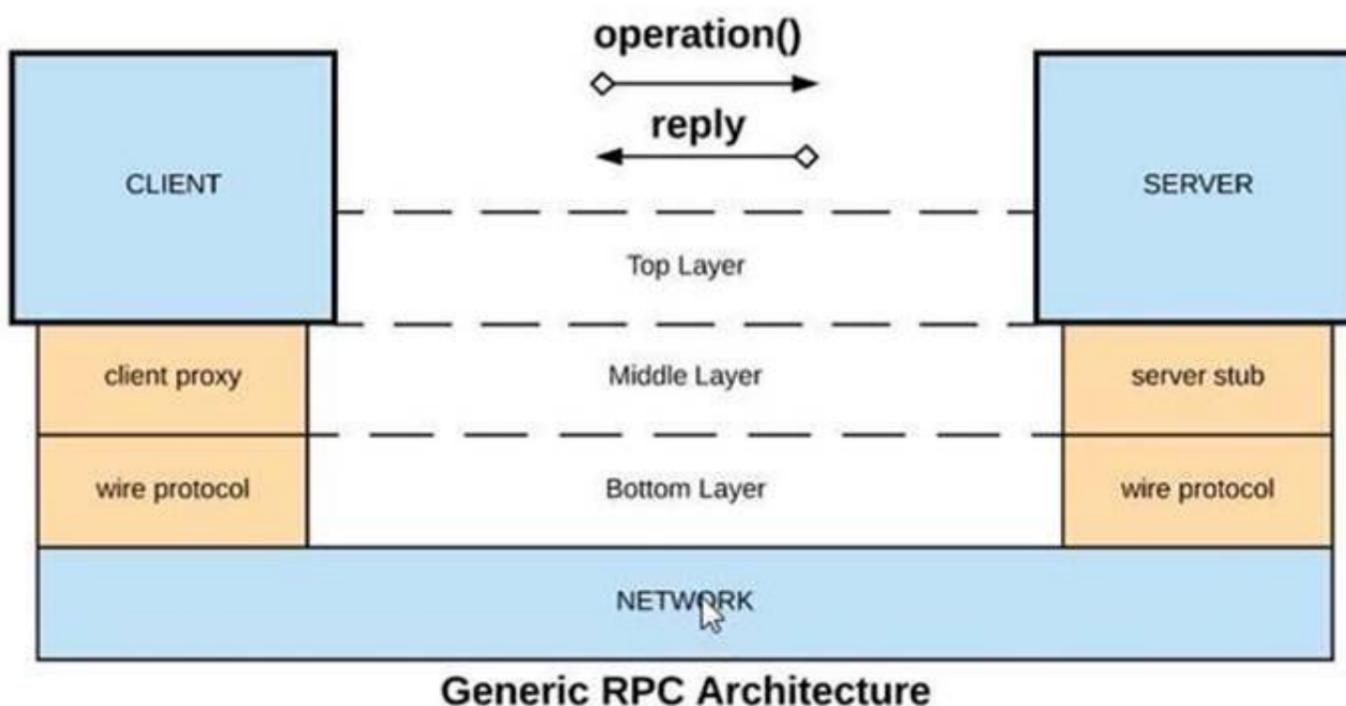
For example, a user of your API may receive a response that includes these headers:

```
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 32

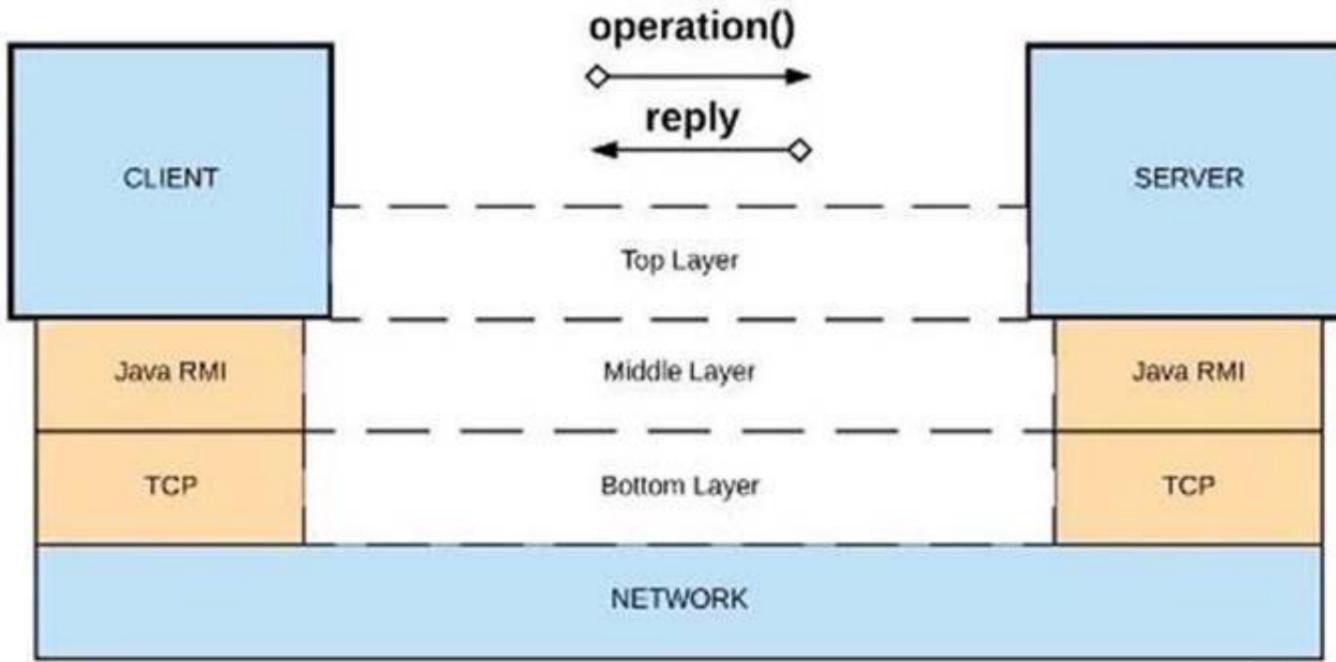
Refer to the exhibit.



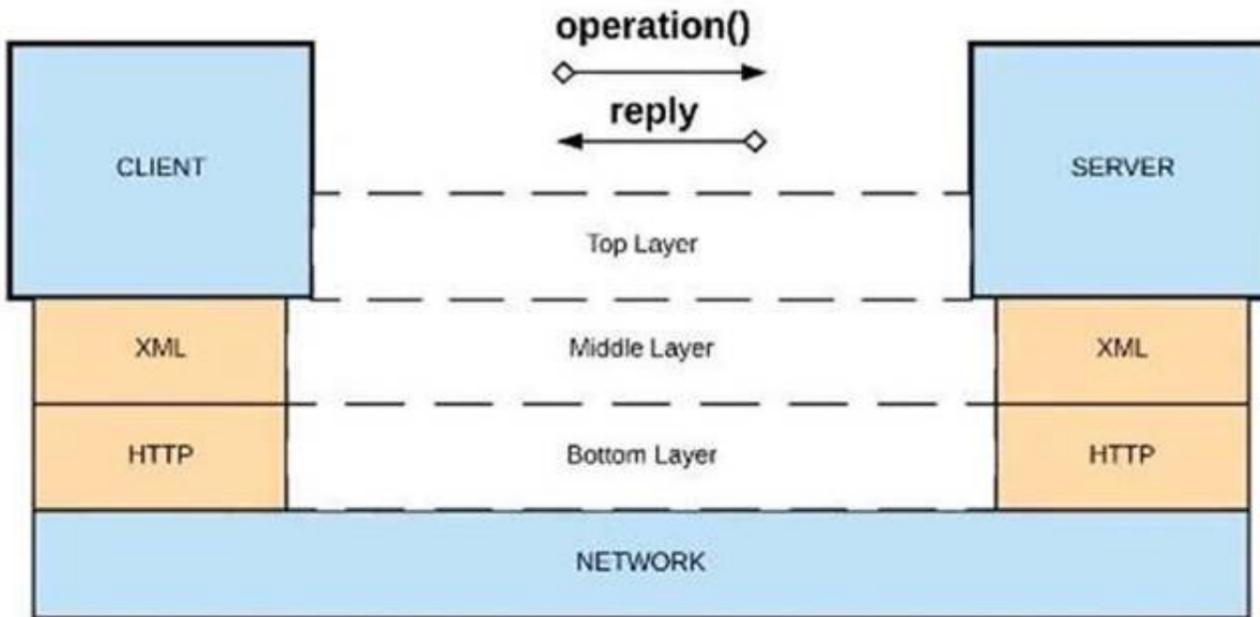
Generic RPC Architecture

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

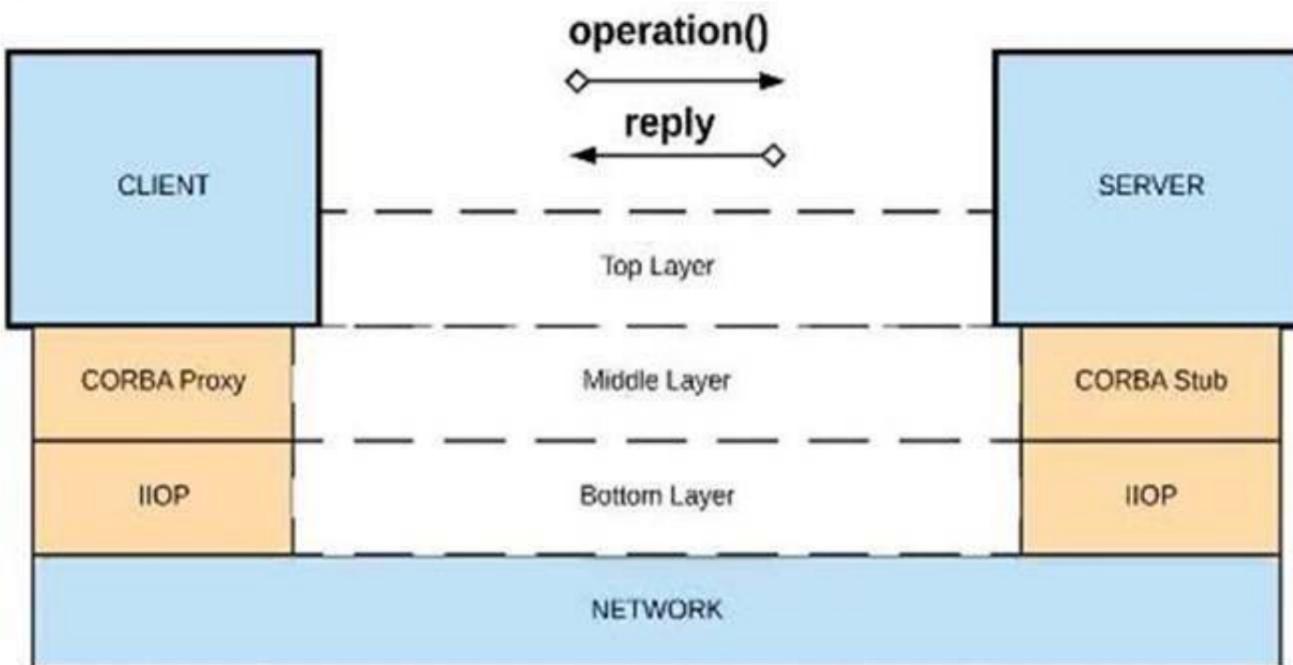
- A) Java RMI over TCP



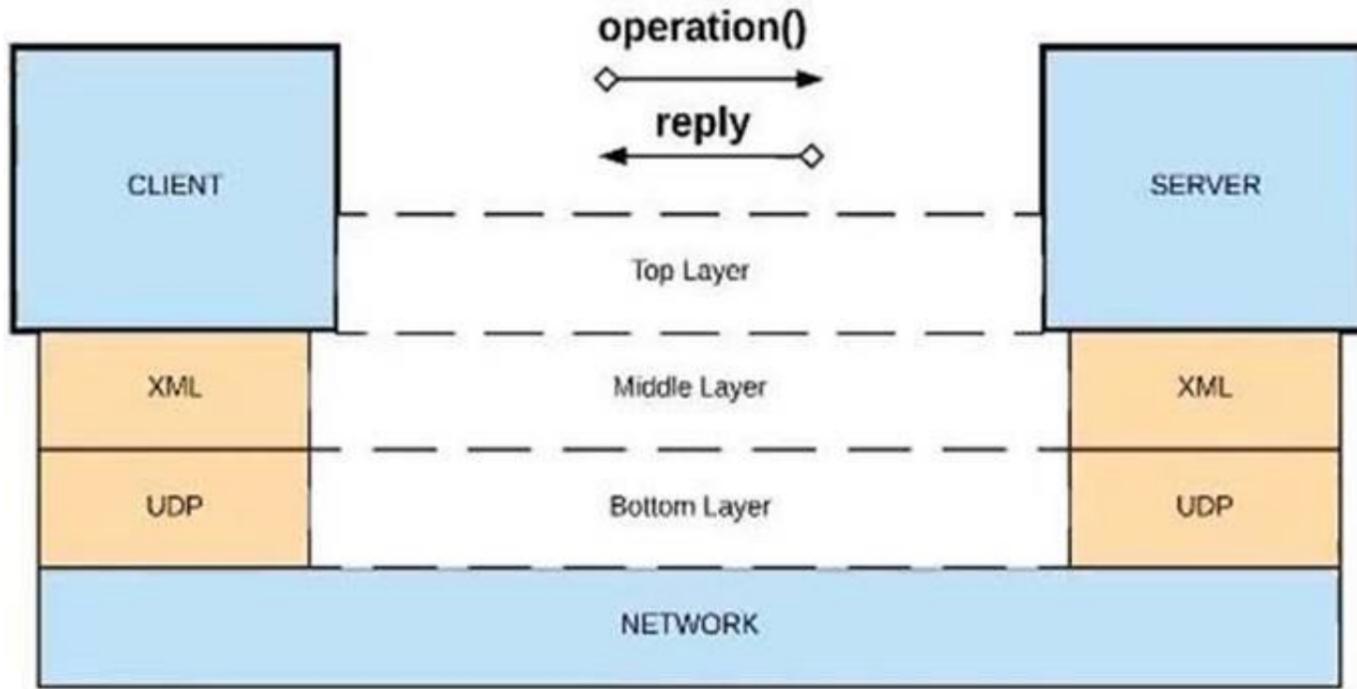
B) Java RMI over TCP



C) CORBA over IIOP



D) XML over UDP



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

Answer: D

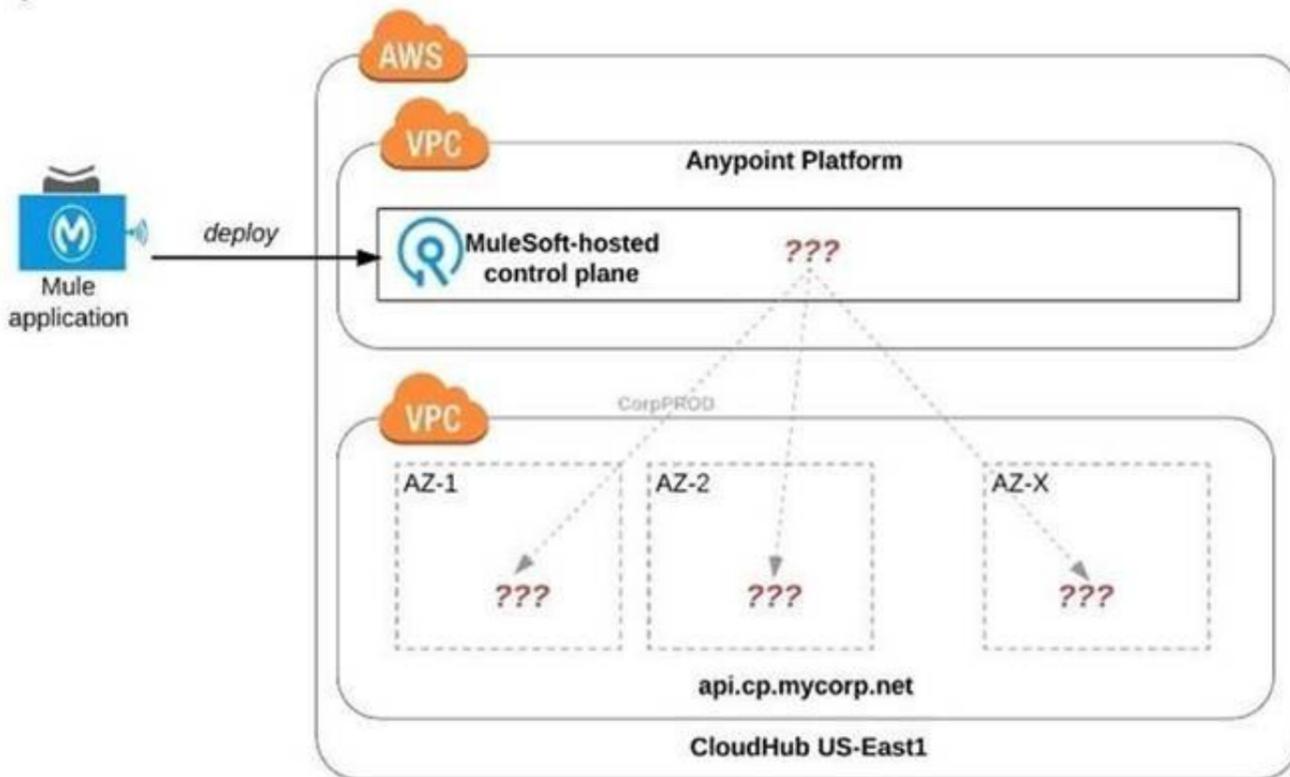
Explanation:

\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

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 40

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 42

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 helps 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 45

What CANNOT be effectively enforced using an API policy in Anypoint Platform?

- A. Guarding against Denial of Service attacks
- B. Maintaining tamper-proof credentials between APIs
- C. Logging HTTP requests and responses
- D. Backend system overloading

Answer: A

Explanation:

Correct Answer

Guarding against Denial of Service attacks

>> Backend system overloading can be handled by enforcing "Spike Control Policy"

>> Logging HTTP requests and responses can be done by enforcing "Message Logging Policy"

>> Credentials can be tamper-proofed using "Security" and "Compliance" Policies

However, unfortunately, there is no proper way currently on Anypoint Platform to guard against DOS attacks.

NEW QUESTION 48

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

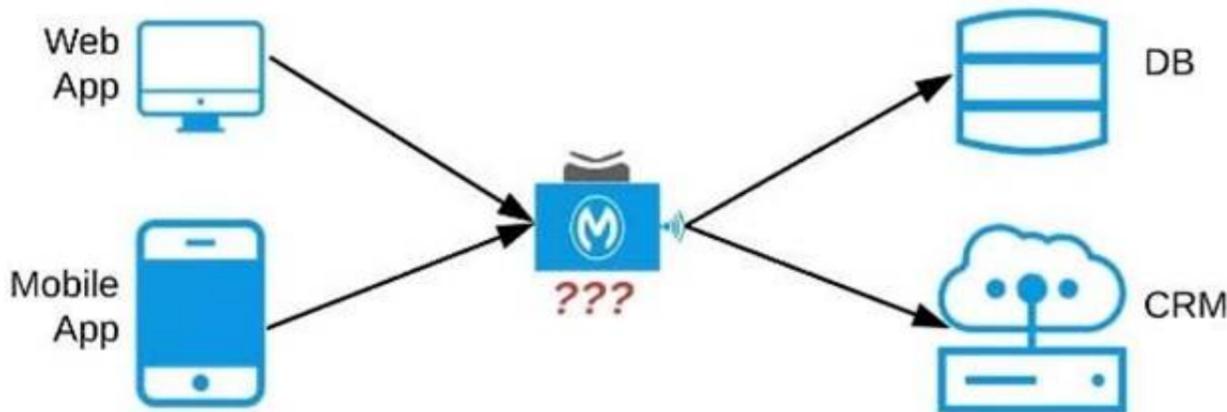
***** Key details to take from the given scenario:

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

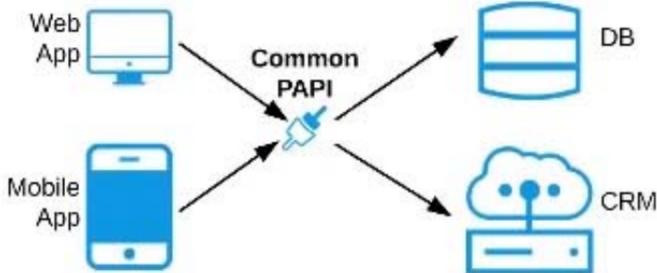
>> This API calls 3 downstream APIs sequentially and all these are of similar complexity.
 >> The first downstream API is offering median SLA of 100ms, 80th percentile: 500ms; 95th percentile: 1000ms.
 Based on the above details:
 >> We can rule out the option which is suggesting to set 50ms timeout. 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 51

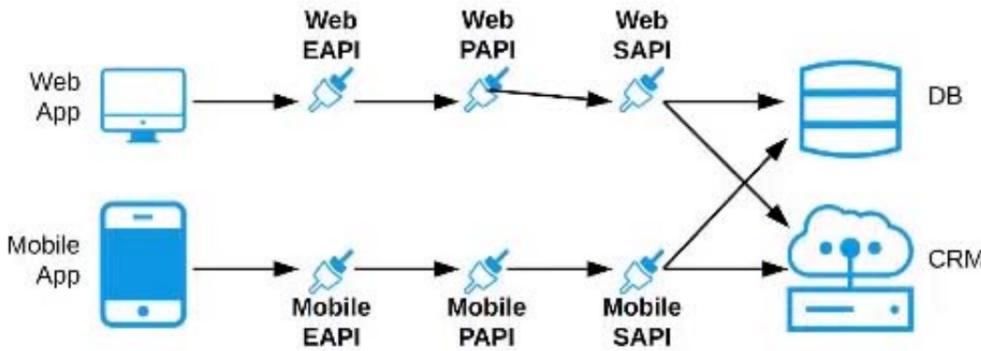
Refer to the exhibit. An organization needs to enable access to their customer data from both a mobile app and a web application, which each need access to common fields as well as certain unique fields.
 The data is available partially in a database and partially in a 3rd-party CRM system.
 What APIs should be created to best fit these design requirements?



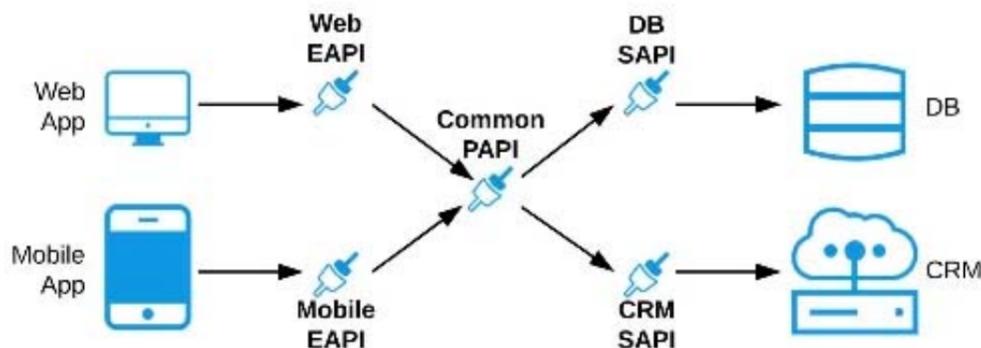
A) A Process API that contains the data required by both the web and mobile apps, allowing these applications to invoke it directly and access the data they need thereby providing the flexibility to add more fields in the future without needing API changes



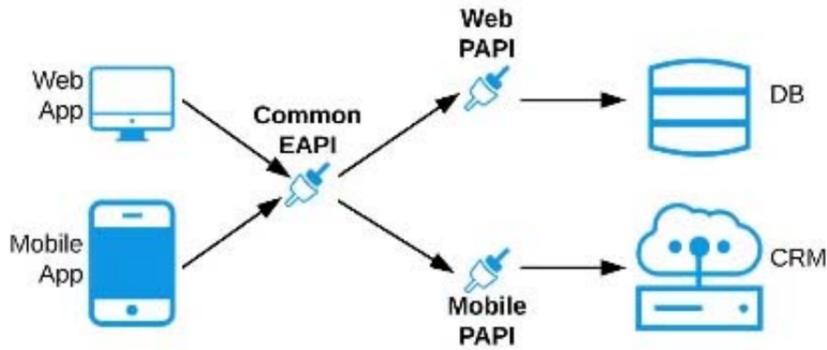
B) One set of APIs (Experience API, Process API, and System API) for the web app, and another set for the mobile app



C) Separate Experience APIs for the mobile and web app, but a common Process API that invokes separate System APIs created for the database and CRM system



D) A common Experience API used by both the web and mobile apps, but separate Process APIs for the web and mobile apps that interact with the database and the CRM System



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

Answer: C

Explanation:

Correct Answer

Separate Experience APIs for the mobile and web app, but a common Process API that invokes separate System APIs created for the database and CRM system
 ***** As per MuleSoft's API-led connectivity:

- >> Experience APIs should be built as per each consumer needs and their experience.
- >> Process APIs should contain all the orchestration logic to achieve the business functionality.
- >> System APIs should be built for each backend system to unlock their data.

NEW QUESTION 53

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

- A. None
- B. Both
- C. API Client
- D. API Consumer

Answer: D

Explanation:

Correct Answer

API Consumer

- >> API clients are piece of code or programs that use the client credentials of API consumer but does not directly interact with Anypoint Exchange to get the access
 - >> API consumer is the one who should get registered and request access to API and then API client needs to use those client credentials to hit the APIs
- So, API consumer is the one who needs to request access on the API from Anypoint Exchange

NEW QUESTION 57

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

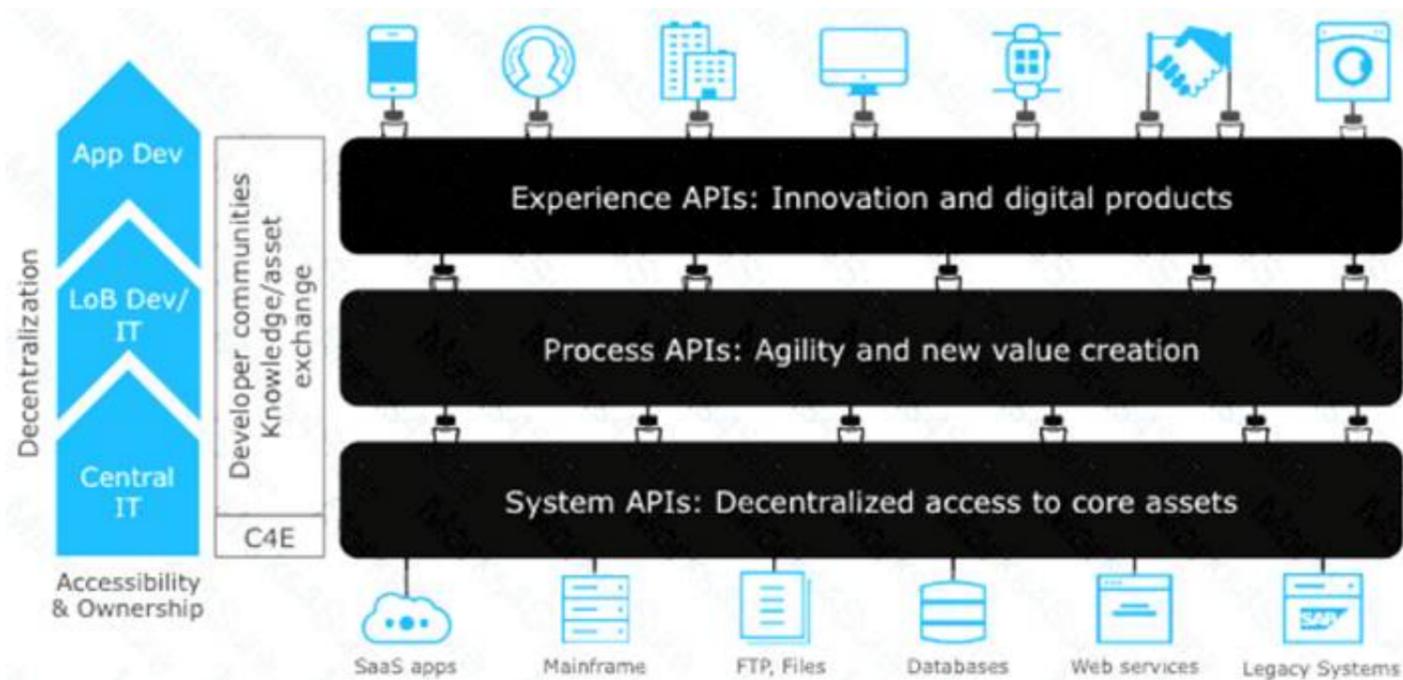
- A. Experience Layer
- B. Process Layer
- C. System Layer

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 59

A System API is designed to retrieve data from a backend system that has scalability challenges. What API policy can best safeguard the backend system?

- A. IPwhitelist
- B. SLA-based rate limiting
- C. Auth 2 token enforcement
- D. Client ID enforcement

Answer: B

Explanation:

Correct Answer

SLA-based rate limiting

>> Client Id enforcement policy is a "Compliance" related NFR and does not help in maintaining the "Quality of Service (QoS)". It CANNOT and NOT meant for protecting the backend systems from scalability challenges.

>> IP Whitelisting and OAuth 2.0 token enforcement are "Security" related NFRs and again does not help in maintaining the "Quality of Service (QoS)". They CANNOT and are NOT meant for protecting the backend systems from scalability challenges.

Rate Limiting, Rate Limiting-SLA, Throttling, Spike Control are the policies that are "Quality of Service (QoS)" related NFRs and are meant to help in protecting the backend systems from getting overloaded.

<https://dzone.com/articles/how-to-secure-apis>

NEW QUESTION 62

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 65

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 70

An organization has several APIs that accept JSON data over HTTP POST. The APIs are all publicly available and are associated with several mobile applications and web applications.

The organization does NOT want to use any authentication or compliance policies for these APIs, but at the same time, is worried that some bad actor could send payloads that could somehow compromise the applications or servers running the API implementations.

What out-of-the-box Anypoint Platform policy can address exposure to this threat?

- A. Shut out bad actors by using HTTPS mutual authentication for all API invocations
- B. Apply an IP blacklist policy to all APIs; the blacklist will include all bad actors
- C. Apply a Header injection and removal policy that detects the malicious data before it is used
- D. Apply a JSON threat protection policy to all APIs to detect potential threat vectors

Answer: D

Explanation:

Correct Answer

Apply a JSON threat protection policy to all APIs to detect potential threat vectors

>> Usually, if the APIs are designed and developed for specific consumers (known consumers/customers) then we would IP Whitelist the same to ensure that traffic only comes from them.

>> However, as this scenario states that the APIs are publicly available and being used by so many mobile and web applications, it is NOT possible to identify and blacklist all possible bad actors.

>> So, JSON threat protection policy is the best chance to prevent any bad JSON payloads from such bad actors.

NEW QUESTION 74

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 77

What is the main change to the IT operating model that MuleSoft recommends to organizations to improve innovation and clock speed?

- A. Drive consumption as much as production of assets; this enables developers to discover and reuse assets from other projects and encourages standardization
- B. Expose assets using a Master Data Management (MDM) system; this standardizes projects and enables developers to quickly discover and reuse assets from other projects
- C. Implement SOA for reusable APIs to focus on production over consumption; this standardizes on XML and WSDL formats to speed up decision making
- D. Create a lean and agile organization that makes many small decisions everyday; this speeds up decision making and enables each line of business to take ownership of its projects

Answer: A

Explanation:

Correct Answer

Drive consumption as much as production of assets; this enables developers to discover and reuse assets from other projects and encourages standardization

>> The main motto of the new IT Operating Model that MuleSoft recommends and made popular is to change the way that they are delivered from a production model to a production + consumption model, which is done through an API strategy called API-led connectivity.
 >> The assets built should also be discoverable and self-serveable for reusability across LOBs and organization.
 >> MuleSoft's IT operating model does not talk about SDLC model (Agile/ Lean etc) or MDM at all. So, options suggesting these are not valid.
 References:
<https://blogs.mulesoft.com/biz/connectivity/what-is-a-center-for-enablement-c4e/> <https://www.mulesoft.com/resources/api/secret-to-managing-it-projects>

NEW QUESTION 81

A company has started to create an application network and is now planning to implement a Center for Enablement (C4E) organizational model. What key factor would lead the company to decide upon a federated rather than a centralized C4E?

- A. When there are a large number of existing common assets shared by development teams
- B. When various teams responsible for creating APIs are new to integration and hence need extensive training
- C. When development is already organized into several independent initiatives or groups
- D. When the majority of the applications in the application network are cloud based

Answer: C

Explanation:

Correct Answer

When development is already organized into several independent initiatives or groups

>> It would require lot of process effort in an organization to have a single C4E team coordinating with multiple already organized development teams which are into several independent initiatives. A single C4E works well with different teams having at least a common initiative. So, in this scenario, federated C4E works well instead of centralized C4E.

NEW QUESTION 86

The application network is recomposable: it is built for change because it "bends but does not break"

- A. TRUE
- B. FALSE

Answer: A

Explanation:

>> Application Network is a disposable architecture.
 >> Which means, it can be altered without disturbing entire architecture and its components.
 >> It bends as per requirements or design changes but does not break

NEW QUESTION 87

An organization makes a strategic decision to move towards an IT operating model that emphasizes consumption of reusable IT assets using modern APIs (as defined by MuleSoft).
 What best describes each modern API in relation to this new IT operating model?

- A. Each modern API has its own software development lifecycle, which reduces the need for documentation and automation
- B. Each modern API must be treated like a product and designed for a particular target audience (for instance, mobile app developers)
- C. Each modern API must be easy to consume, so should avoid complex authentication mechanisms such as SAML or JWT
- D. Each modern API must be REST and HTTP based

Answer: B

Explanation:

Correct Answers

* 1. Each modern API must be treated like a product and designed for a particular target audience (for instance mobile app developers)

Bottom of Form Top of Form

NEW QUESTION 88

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

- A. set a CloudHub autoscaling policy to add CloudHub workers
- B. Use a CloudHub autoscaling policy to increase the size of the CloudHub worker
- C. Increase the size of the CloudHub worker(s)
- D. Increase the number of CloudHub workers

Answer: D

Explanation:

Correct Answer

Increase the size of the CloudHub worker(s)

The key details that we can take out from the given scenario are:

- >> API implementation uses a database bulk insert command to submit all the purchase data to a database
 - >> JDBC driver processes the data into a set of several temporary disk files on the CloudHub worker
 - >> Sometimes a request fails and the logs show a message indicating an out-of-file-space message Based on above details:
 - >> Both auto-scaling options does NOT help because we cannot set auto-scaling rules based on error messages. Auto-scaling rules are kicked-off based on CPU/Memory usages and not due to some given error or disk space issues.
 - >> Increasing the number of CloudHub workers also does NOT help here because the reason for the failure is not due to performance aspects w.r.t CPU or Memory. It is due to disk-space.
 - >> Moreover, the API is doing bulk insert to submit the received batch data. Which means, all data is handled by ONE worker only at a time. So, the disk space issue should be tackled on "per worker" basis. Having multiple workers does not help as the batch may still fail on any worker when disk is out of space on that particular worker.
- Therefore, the right way to deal this issue and resolve this is to increase the vCore size of the worker so that a new worker with more disk space will be provisioned.

NEW QUESTION 90

What Anypoint Platform Capabilities listed below fall under APIs and API Invocations/Consumers category? Select TWO.

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

Answer: D

Explanation:

Correct Answers: API Operations and Management and 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
- Bottom of Form Top of Form

NEW QUESTION 94

What is true about where an API policy is defined in Anypoint Platform and how it is then applied to API instances?

- A. The API policy is defined in Runtime Manager as part of the API deployment to a Mule runtime, and then ONLY applied to the specific API Instance
- B. The API policy is defined in API Manager for a specific API Instance, and then ONLY applied to the specific API instance
- C. The API policy is defined in API Manager and then automatically applied to ALL API instances
- D. The API policy is defined in API Manager, and then applied to ALL API instances in the specified environment

Answer: B

Explanation:

Correct Answer

The API policy is defined in API Manager for a specific API instance, and then ONLY applied to the specific API instance.

- >> Once our API specifications are ready and published to Exchange, we need to visit API Manager and register an API instance for each API.
- >> API Manager is the place where management of API aspects takes place like addressing NFRs by enforcing policies on them.
- >> We can create multiple instances for a same API and manage them differently for different purposes.
- >> One instance can have a set of API policies applied and another instance of same API can have different set of policies applied for some other purpose.
- >> These APIs and their instances are defined PER environment basis. So, one need to manage them separately in each environment.
- >> We can ensure that same configuration of API instances (SLAs, Policies etc..) gets promoted when promoting to higher environments using platform feature. But this is optional only. Still one can change them per environment basis if they have to.
- >> Runtime Manager is the place to manage API Implementations and their Mule Runtimes but NOT APIs itself. Though API policies gets executed in Mule Runtimes, We CANNOT enforce API policies in Runtime Manager. We would need to do that via API Manager only for a cherry picked instance in an environment. So, based on these facts, right statement in the given choices is - "The API policy is defined in API Manager for a specific API instance, and then ONLY applied to the specific API instance".

NEW QUESTION 98

.....

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