# Product Questions: 152
# Version: 11.0

## Question: 1

What API policy would LEAST likely be applied to a Process API?

A. Custom circuit breaker
B. Client ID enforcement
C. Rate limiting
D. JSON threat protection

**Answer: D**

Explanation:

Correct Answe r: JSON threat protection
*****************************************
Fact: Technically, there are no restrictions on what policy can be applied in what layer. Any policy can be applied on any layer API. However, context should also be considered properly before blindly applying the policies on APIs.
That is why, this question asked for a policy that would LEAST likely be applied to a Process API.
From the given options:
>> All policies except "JSON threat protection" can be applied without hesitation to the APIs in Process tier.
>> JSON threat protection policy ideally fits for experience APIs to prevent suspicious JSON payload coming from external API clients. This covers more of a security aspect by trying to avoid possibly malicious and harmful JSON payloads from external clients calling experience APIs.
As external API clients are NEVER allowed to call Process APIs directly and also these kind of malicious and harmful JSON payloads are always stopped at experience API layer only using this policy, it is LEAST LIKELY that this same policy is again applied on Process Layer API.
Reference: https://docs.mulesoft.com/api-manager/2.x/policy-mule3-provided-policies

## Question: 2

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 Answe r: 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.
Reference: https://help.mulesoft.com/s/question/0D52T00004mXSTUSA4/how-should-a-company-measure-c4e-success

## Question: 3

An organization is implementing a Quote of the Day API that caches today's quote.
What scenario can use the GoudHub 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 anottV 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 Answe r: 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.

## Question: 4

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 Answe r: 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 questio n :
>> 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 vicevera (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 to 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.

Reference: https://docs.mulesoft.com/runtime-manager/cloudhub-dedicated-load-balancer

## Question: 5

What do the API invocation metrics provided by Anypoint Platform provide?

A. ROI metrics from APIs that can be directly shared with business users
B. Measurements of the effectiveness of the application network based on the level of reuse
C. Data on past API invocations to help identify anomalies and usage patterns across various APIs
D. Proactive identification of likely future policy violations that exceed a given threat threshold

**Answer: C**

Explanation:

Correct Answe r: Data on past API invocations to help identify anomalies and usage patterns across various APIs
*****************************************
API Invocation metrics provided by Anypoint Platform:
>> Does NOT provide any Return Of Investment (ROI) related information. So the option suggesting it is OUT.
>> Does NOT provide any information w.r.t how APIs are reused, whether there is effective usage of APIs or not etc...
>> Does NOT prodive any prediction information as such to help us proactively identify any future policy violations.
So, the kind of data/information we can get from such metrics is on past API invocations to help identify anomalies and usage patterns across various APIs.
Reference:
https://usermanual.wiki/Document/APAAppNetstudentManual02may2018.991784750.pdf

## Question: 6

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

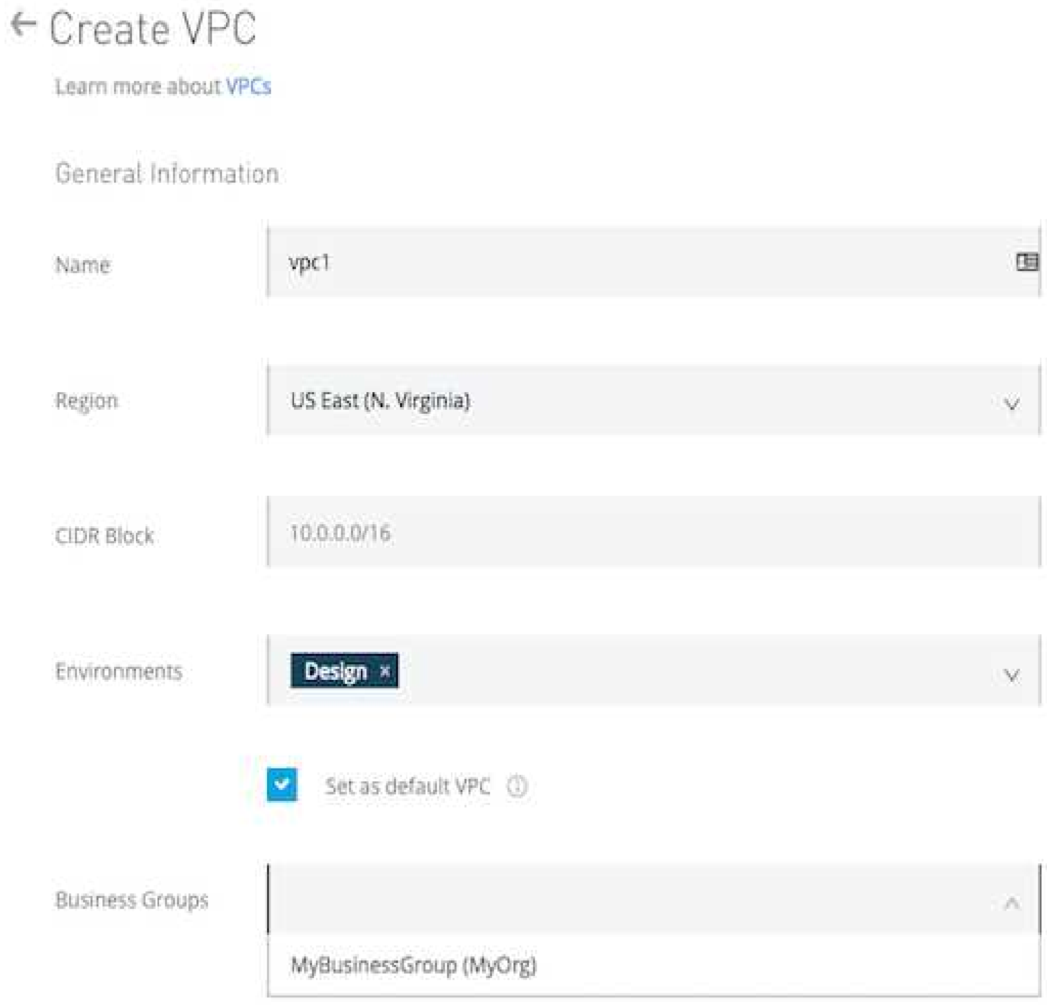
**Answer: B**

Explanation:

Correct Answe r: 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.



| | |
|---|---|
| Name | vpc1 |
| Region | US East (N. Virginia) |
| CIDR Block | 10.0.0.0/16 |
| Environments | Design × |
| | ✓ Set as default VPC |
| Business Groups | MyBusinessGroup (MyOrg) |

that each CloudHub environment requires a separate Anypoint VPC. Once an Anypoint VPC is created, we can choose a same VPC by multiple environments. However, it is generally a best and recommended practice to always have seperate 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.
Reference: https://docs.mulesoft.com/runtime-manager/vpn-about
Only true statement in the given choices is that the traffic between Mule applications deployed to an Anypoint VPC and on-premises systems can stay within a private network.
https://docs.mulesoft.com/runtime-manager/vpc-connectivity-methods-concept

## Question: 7

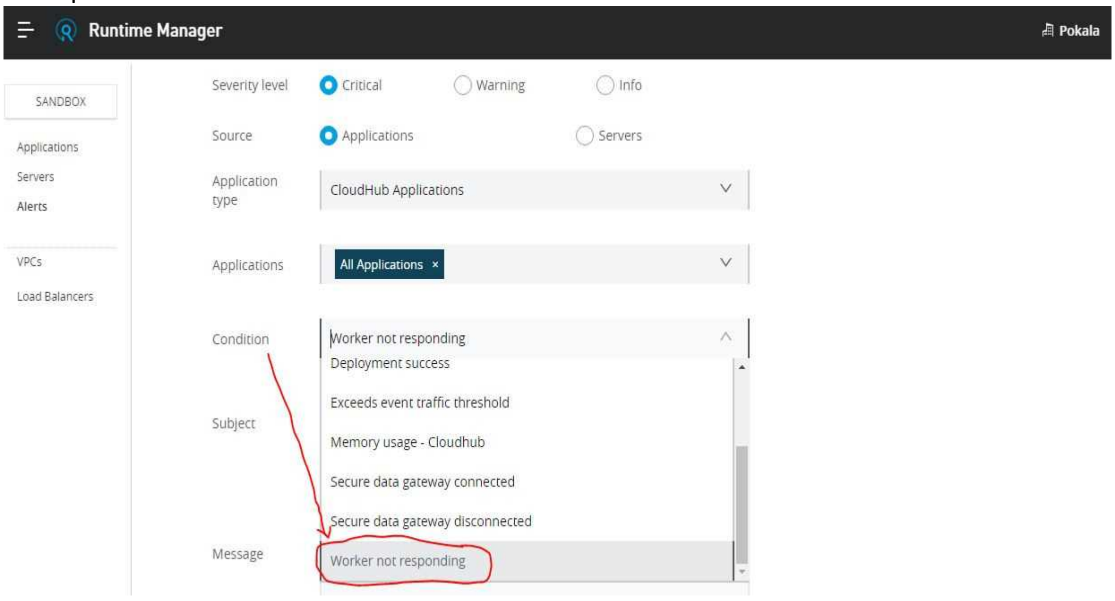
An API implementation is deployed on a single worker on CloudHub and invoked by external API clients (outside of CloudHub). How can an alert be set up that is guaranteed to trigger AS SOON AS that API implementation stops responding to API invocations?

A. Implement a heartbeat/health check within the API and invoke it from outside the Anypoint Platform and alert when the heartbeat does not respond
B. Configure a "worker not responding" alert in Anypoint Runtime Manager
C. Handle API invocation exceptions within the calling API client and raise an alert from that API client when the API Is unavailable
D. Create an alert for when the API receives no requests within a specified time period

**Answer: B**

Explanation:

Correct Answe r: Configure a “Worker not responding” alert in Anypoint Runtime Manager.
*****************************************
>> All the options eventually helps to generate the alert required when the application stops responding.
>> However, handling exceptions within calling API and then raising alert from API client is inappropriate and silly. There could be many API clients invoking the API implementation and it is not ideal to have this setup consistently in all of them. Not a realistic way to do.
>> Implementing a health check/ heartbeat with in the API and calling from outside to detmine the health sounds OK but needs extra setup for it and same time there are very good chances of generating false alarms when there are any intermittent network issues between external tool calling the health check API on API implementation. The API implementation itself may not have any issues but due to some other factors some false alarms may go out.
>> Creating an alert in API Manager when the API receives no requests within a specified time period would actually generate realistic alerts but even here some false alarms may go out when there are genuinely no requests from API clients.
The best and right way to achieve this requirement is to setup an alert on Runtime Manager with a condition "Worker not responding". This would generate an alert AS SOON AS the workers become unresponsive.



Severity level: Critical [✓] Warning Info
Source: Applications [✓] Servers
Application type: CloudHub Applications
Applications: All Applications ×
Condition: Worker not responding
Subject
Message

Bottom of Form
Top of Form

## Question: 8

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

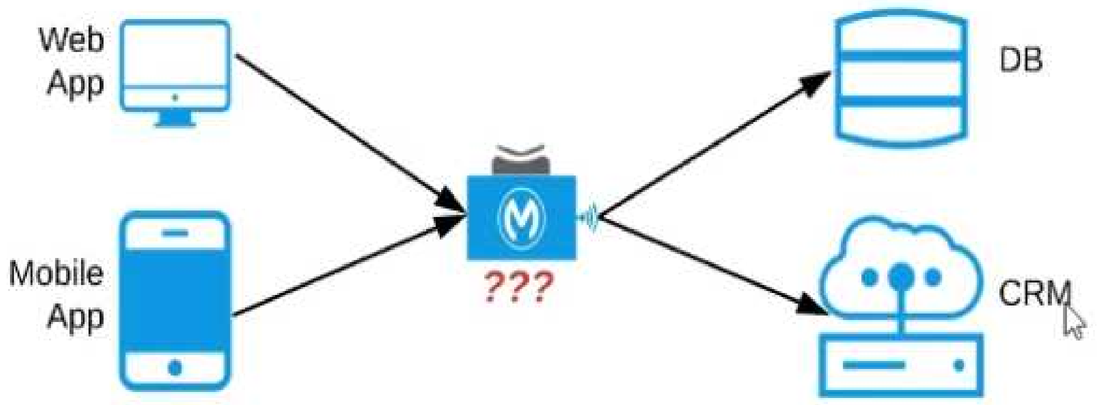
**Answer: C**

Explanation:

Correct Answe r: 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.

## Question: 9

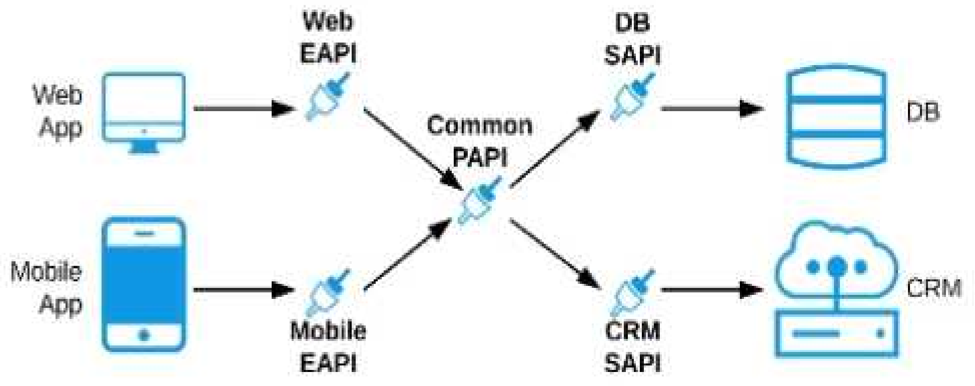
Refer to the exhibit. An organization needs to enable access to their customer data from both a mobile app and a web application, which each need access to common fields as well as certain unique fields.
The data is available partially in a database and partially in a 3rd-party CRM system.
What APIs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

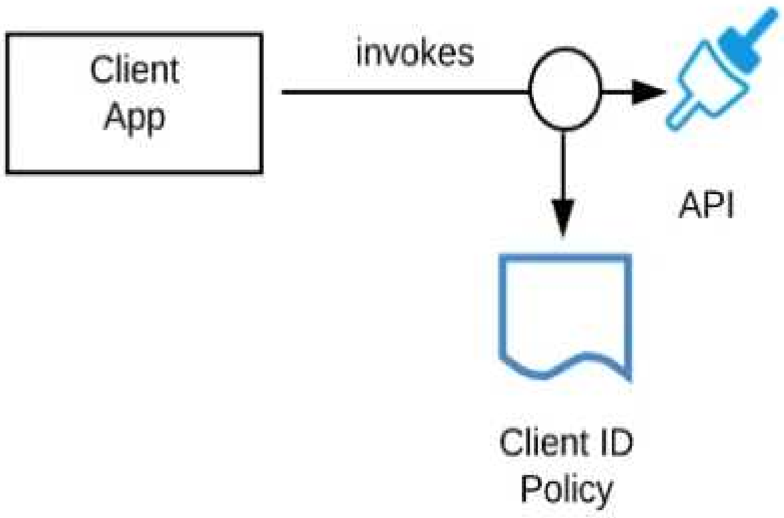
Explanation:

Correct Answe r: 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.
Reference: https://blogs.mulesoft.com/dev/api-dev/what-is-api-led-connectivity/

## Question: 10

Refer to the exhibit.



A developer is building a client application to invoke an API deployed to the STAGING environment that is governed by a client ID enforcement policy.
What is required to successfully invoke the API?

A. The client ID and secret for the Anypoint Platform account owning the API in the STAGING environment
B. The client ID and secret for the Anypoint Platform account's STAGING environment
C. The client ID and secret obtained from Anypoint Exchange for the API instance in the STAGING environment
D. A valid OAuth token obtained from Anypoint Platform and its associated client ID and secret

**Answer: C**

Explanation:

Correct Answe r: The client ID and secret obtained from Anypoint Exchange for the API instance in the STAGING environment
*****************************************
>> We CANNOT use the client ID and secret of Anypoint Platform account or any individual environments for accessing the APIs
>> As the type of policy that is enforced on the API in question is "Client ID Enforcment Policy", OAuth token based access won't work.
Right way to access the API is to use the client ID and secret obtained from Anypoint Exchange for

the API instance in a particular environment we want to work on.
Reference:
Managing API instance Contracts on API Manager
https://docs.mulesoft.com/api-manager/1.x/request-access-to-api-task
https://docs.mulesoft.com/exchange/to-request-access
https://docs.mulesoft.com/api-manager/2.x/policy-mule3-client-id-based-policies

## Question: 11

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 Answe r: 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 seems 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 fron Mule Runtimes only irrespective of whether the deployment model is MuleSoft-hosted or Customer-hosted or Hybrid.

## Question: 12

When designing an upstream API and its implementation, the development team has been advised to NOT set timeouts when invoking a downstream API, because that downstream API has no SLA that can be relied upon. This is the only downstream API dependency of that upstream API.
Assume the downstream API runs uninterrupted without crashing. What is the impact of this advice?

A. An SLA for the upstream API CANNOT be provided
B. The invocation of the downstream API will run to completion without timing out
C. A default timeout of 500 ms will automatically be applied by the Mule runtime in which the upstream API implementation executes

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

**Answer: A**

Explanation:

Correct Answe r: An SLA for the upstream API CANNOT be provided.
*****************************************
>> First thing first, the default HTTP response timeout for HTTP connector is 10000 ms (10 seconds). NOT 500 ms.
>> Mule runtime does NOT apply any such "load-dependent" timeouts. There is no such behavior currently in Mule.
>> As there is default 10000 ms time out for HTTP connector, we CANNOT always guarantee that the invocation of the downstream API will run to completion without timing out due to its unreliable SLA times. If the response time crosses 10 seconds then the request may time out.
The main impact due to this is that a proper SLA for the upstream API CANNOT be provided.
Reference: https://docs.mulesoft.com/http-connector/1.5/http-documentation#parameters-3

## Question: 13

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

A. It makes API Manager aware of API implementations and hence enables it to enforce policies
B. It enables Anypoint Studio to discover API definitions configured in Anypoint Platform
C. It enables Anypoint Exchange to discover assets and makes them available for reuse
D. It enables Anypoint Analytics to gain insight into the usage of APIs

**Answer: A**

Explanation:

Correct Answe r: It makes API Manager aware of API implementations and hence enables it to enforce policies.
*****************************************
>> API Autodiscovery is a mechanism that manages an API from API Manager by pairing the deployed application to an API created on the platform.
>> API Management includes tracking, enforcing policies if you apply any, and reporting API analytics.
>> Critical to the Autodiscovery process is identifying the API by providing the API name and version.
Reference:
https://docs.mulesoft.com/api-manager/2.x/api-auto-discovery-new-concept
https://docs.mulesoft.com/api-manager/1.x/api-auto-discovery
https://docs.mulesoft.com/api-manager/2.x/api-auto-discovery-new-concept

## Question: 14

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**

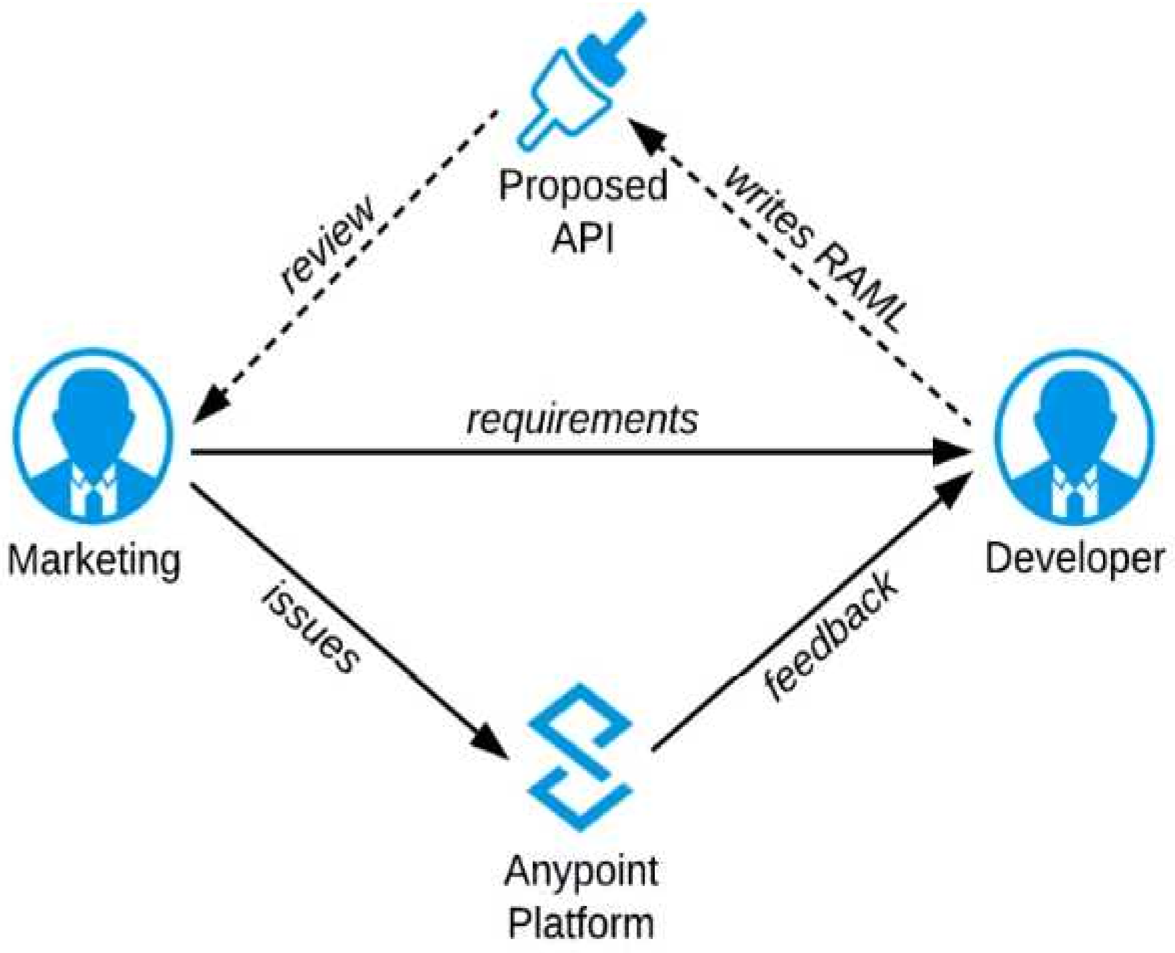
Explanation:

Correct Answe r: The visibility level of the API instances of that API that need to be publicly accessible should be set to public visibility.
*****************************************

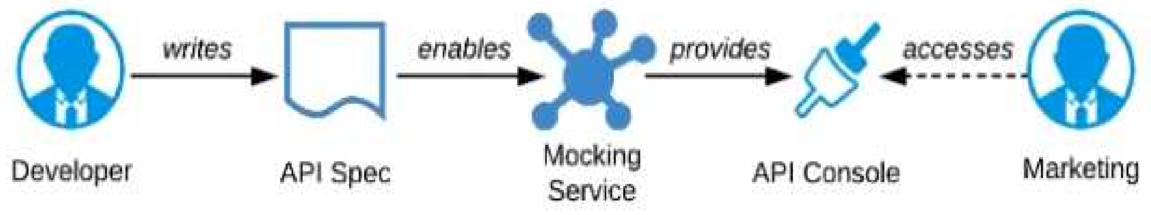
Reference: https://docs.mulesoft.com/exchange/to-share-api-asset-to-portal
https://docs.mulesoft.com/exchange/to-share-api-asset-to-portal

## Question: 15

Refer to the exhibit.



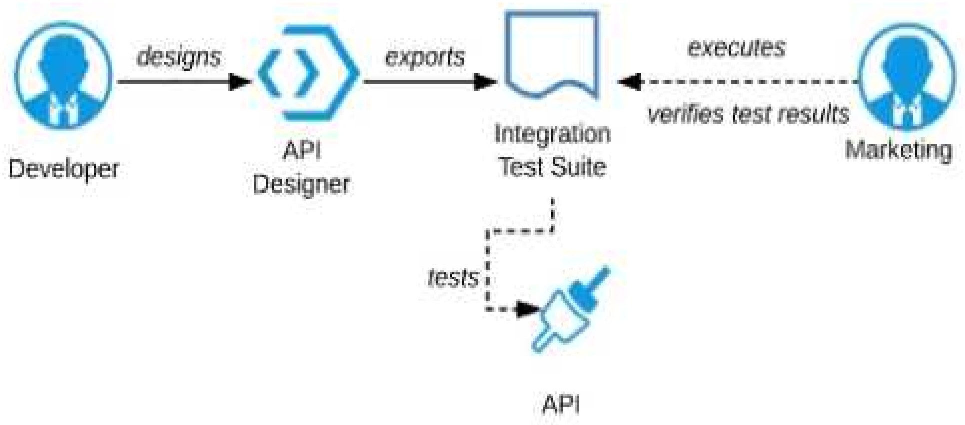
A RAML definition has been proposed for a new Promotions Process API, and has been published to Anypoint Exchange.
The Marketing Department, who will be an important consumer of the Promotions API, has important requirements and expectations that must be met.
What is the most effective way to use Anypoint Platform features to involve the Marketing Department in this early API design phase?
A) Ask the Marketing Department to interact with a mocking implementation of the API using the automatically generated API Console



B) Organize a design workshop with the DBAs of the Marketing Department in which the database schema of the Marketing IT systems is translated into RAML

C) Use Anypoint Studio to Implement the API as a Mule application, then deploy that API implementation to CloudHub and ask the Marketing Department to interact with it

D) Export an integration test suite from API designer and have the Marketing Department execute the tests In that suite to ensure they pass



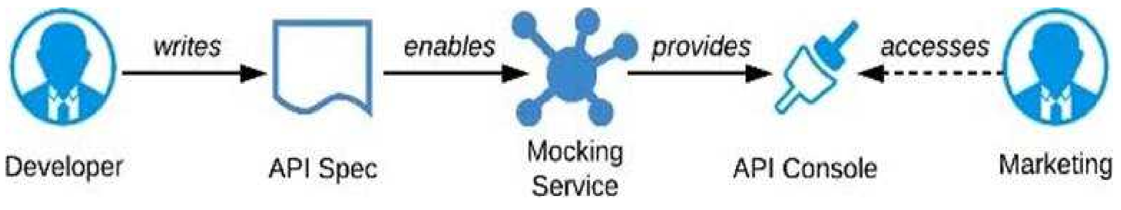
A. Option A
B. Option B
C. Option C
D. Option D

**Answer: A**

Explanation:

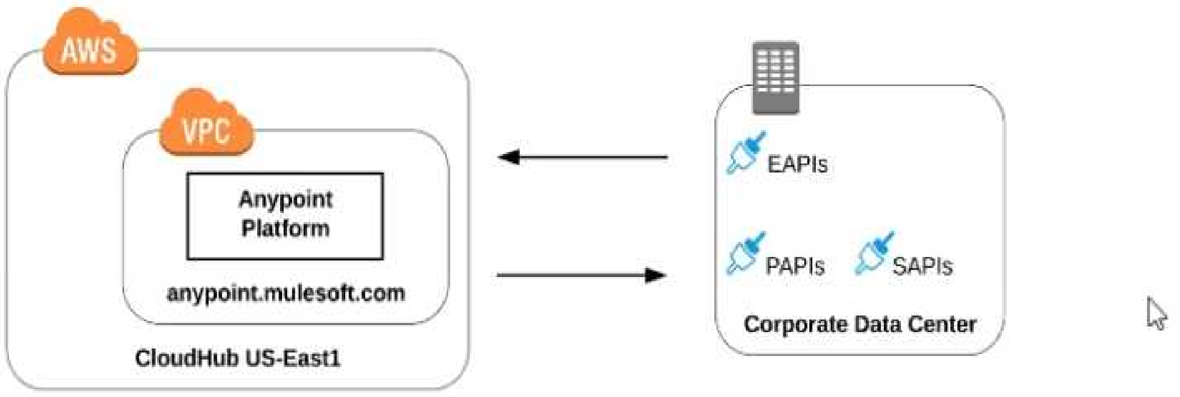
Correct Answe r: Ask the Marketing Department to interact with a mocking implementation of the API using the automatically generated API Console.
*****************************************
As per MuleSoft's IT Operating Model:
>> API consumers need NOT wait until the full API implementation is ready.
>> NO technical test-suites needs to be shared with end users to interact with APIs.
>> Anypoint Platform offers a mocking capability on all the published API specifications to Anypoint Exchange which also will be rich in documentation covering all details of API functionalities and working nature.
>> No needs of arranging days of workshops with end users for feedback.

API consumers can use Anypoint Exchange features on the platform and interact with the API using its mocking feature. The feedback can be shared quickly on the same to incorporate any changes.



## Question: 16

Refer to the exhibit.



what is true when using customer-hosted Mule runtimes with the MuleSoft-hosted Anypoint Platform control plane (hybrid deployment)?

A. Anypoint Runtime Manager initiates a network connection to a Mule runtime in order to deploy Mule applications
B. The MuleSoft-hosted Shared Load Balancer can be used to load balance API invocations to the Mule runtimes
C. API implementations can run successfully in customer-hosted Mule runtimes, even when they are unable to communicate with the control plane
D. Anypoint Runtime Manager automatically ensures HA in the control plane by creating a new Mule runtime instance in case of a node failure

**Answer: C**

Explanation:

Correct Answe r: API implementations can run successfully in customer-hosted Mule runtimes, even when they are unable to communicate with the control plane.
*****************************************
>> We CANNOT use Shared Load balancer to load balance APIs on customer hosted runtimes

>> For Hybrid deployment models, the on-premises are first connected to Runtime Manager using Runtime Manager agent. So, the connection is initiated first from On-premises to Runtime Manager. Then all control can be done from Runtime Manager.
>> Anypoint Runtime Manager CANNOT ensure automatic HA. Clusters/Server Groups etc should be configured before hand.

Only TRUE statement in the given choices is, API implementations can run successfully in customer-hosted Mule runtimes, even when they are unable to communicate with the control plane. There are several references below to justify this statement.

Reference:
https://docs.mulesoft.com/runtime-manager/deployment-strategies#hybrid-deployments
https://help.mulesoft.com/s/article/On-Premise-Runtimes-Disconnected-From-US-Control-Plane-June-18th-2018
https://help.mulesoft.com/s/article/Runtime-Manager-cannot-manage-On-Prem-Applications-and-

Servers-from-US-Control-Plane-June-25th-2019
https://help.mulesoft.com/s/article/On-premise-Runtimes-Appear-Disconnected-in-Runtime-Manager-May-29th-2018

===========================

===========================

## Question: 17

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 Answe r: SLA-based rate limiting
*****************************************
>> Client Id enforement 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

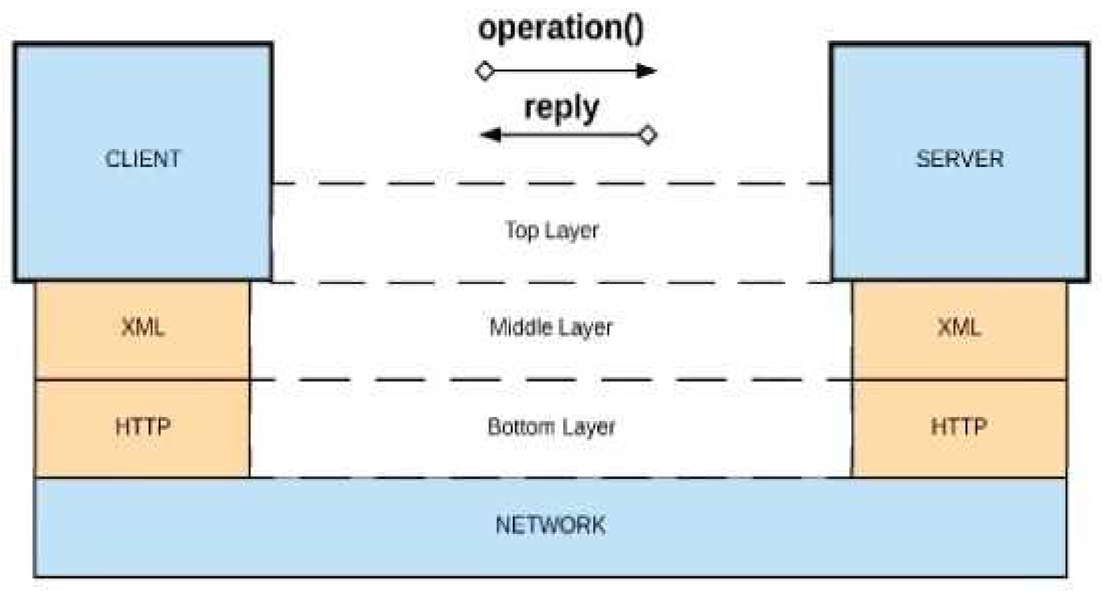
## Question: 18

Refer to the exhibit.

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

A) Java RMI over TCP

B) Java RMI over TCP



C) CORBA over HOP

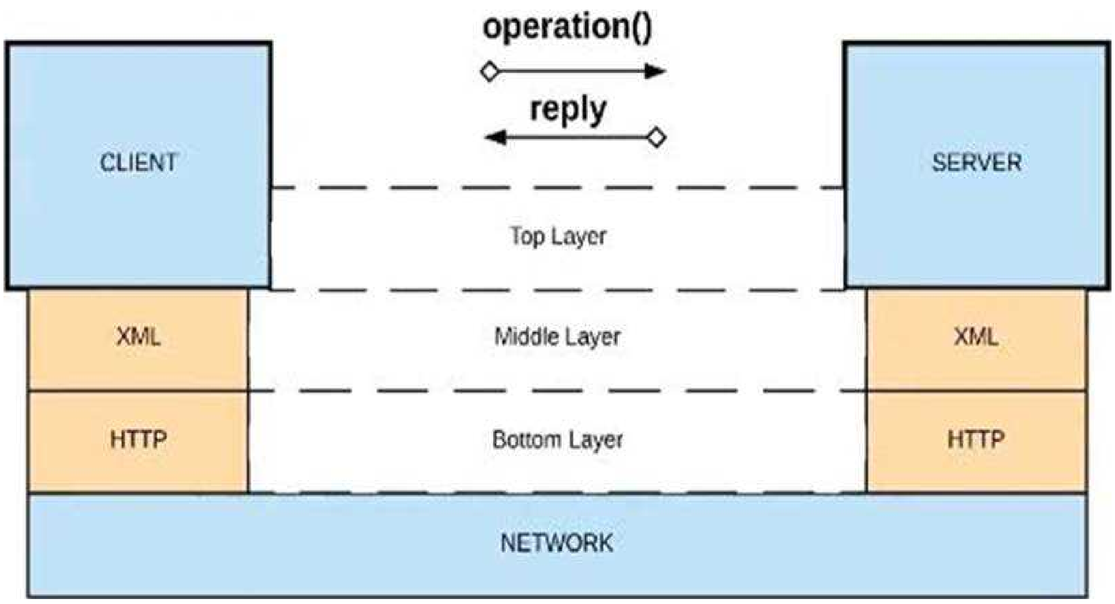
D) XML over UDP

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

**Answer: D**

Explanation:

Correct Answe r: 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 varities 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.

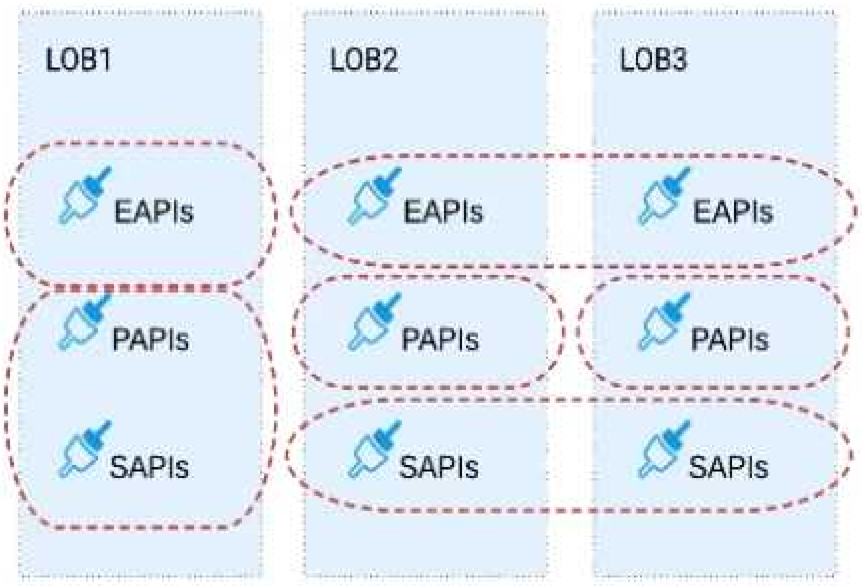


Bottom of Form
Top of Form

## Question: 19

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 (siloed) 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

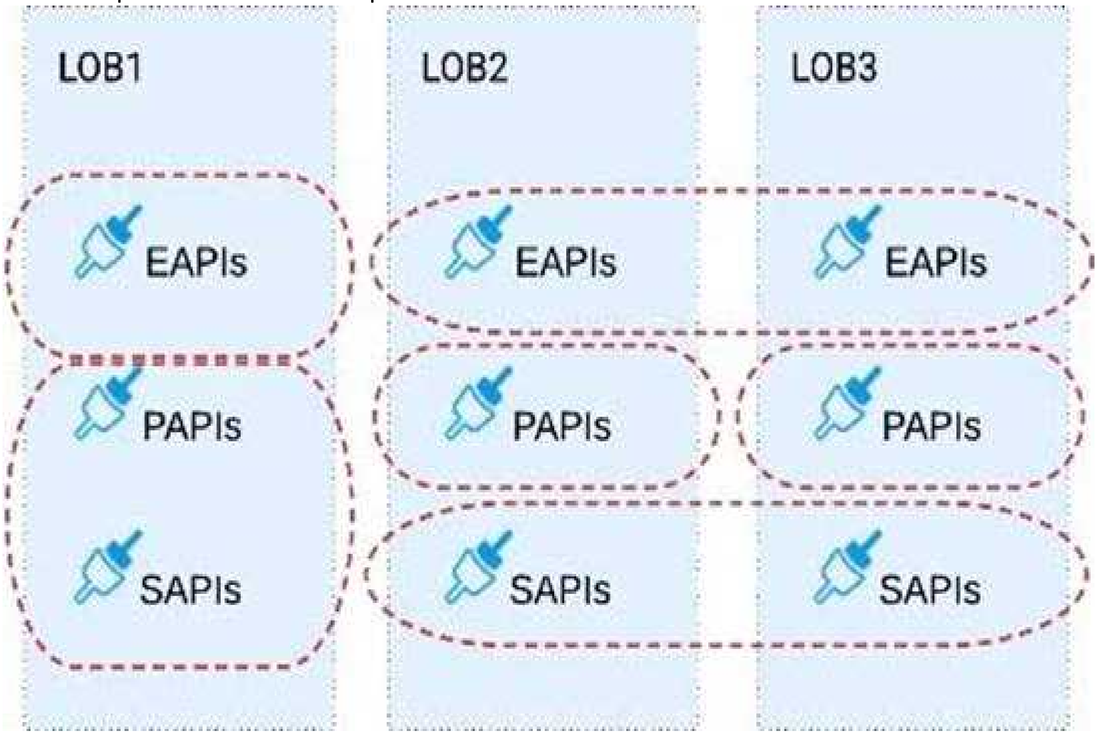
D) Build one centralized Canonical Data Model (Enterprise Data Model) that unifies all the data types from all three business processes, ensuring the data model is consistent and non-redundant

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

**Answer: A**

Explanation:

Correct Answe r: 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.



## Question: 20

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 Answe r: 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.

## Question: 21

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

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

**Answer: C**

Explanation:

Correct Answe r: The SSL certificates used by the API implementation to expose HTTPS endpoints
*****************************************
>> The assignment of each HTTP request to a particular CloudHub worker is taken care by Anypoint Platform itself. We need not manage it explicitly in the API implementation and in fact we CANNOT manage it in the API implementation.

>> The logging configuration that enables log entries to be visible in Runtime Manager is ALWAYS managed in the API implementation and NOT just for SLB. So this is not something we do EXCLUSIVELY when using SLB.
>> We DO NOT manage the number of DNS entries allocated to the API implementation inside the code. Anypoint Platform takes care of this.

It is the SSL certificates used by the API implementation to expose HTTPS endpoints that is to be managed EXCLUSIVELY by the API implementation. Anypoint Platform does NOT do this when using SLBs.

## Question: 22

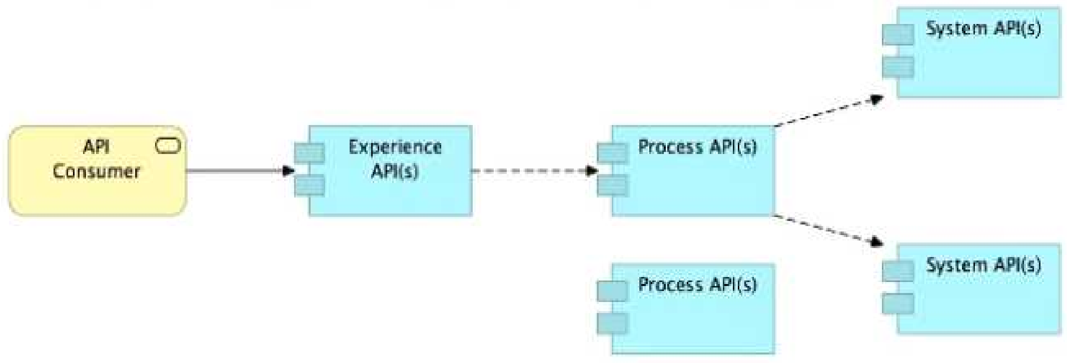
Refer to the exhibit.

What is the best way to decompose one end-to-end business process into a collaboration of Experience, Process, and System APIs?
A) Handle customizations for the end-user application at the Process API level rather than the Experience API level

B) Allow System APIs to return data that is NOT currently required by the identified Process or Experience APIs

C) Always use a tiered approach by creating exactly one API for each of the 3 layers (Experience, Process and System APIs)

D) Use a Process API to orchestrate calls to multiple System APIs, but NOT to other Process APIs

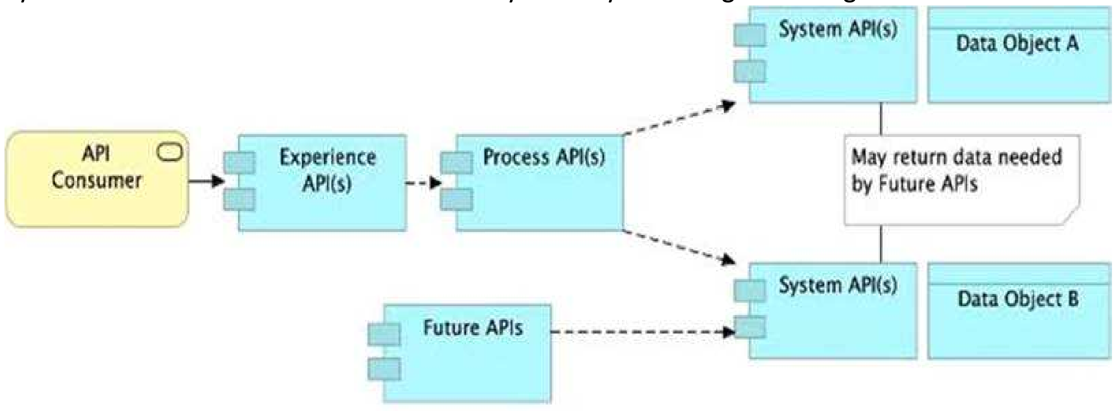


A. Option A
B. Option B
C. Option C
D. Option D

**Answer: B**

Explanation:

Correct Answe r: Allow System APIs to return data that is NOT currently required by the identified Process or Experience APIs.
*****************************************
>> All customizations for the end-user application should be handled in "Experience API" only. Not in Process API
>> We should use tiered approach but NOT always by creating exactly one API for each of the 3 layers. Experience APIs might be one but Process APIs and System APIs are often more than one. System APIs for sure will be more than one all the time as they are the smallest modular APIs built in front of end systems.
>> Process APIs can call System APIs as well as other Process APIs. There is no such anti-design pattern in API-Led connectivity saying Process APIs should not call other Process APIs.
So, the right answer in the given set of options that makes sense as per API-Led connectivity principles is to allow System APIs to return data that is NOT currently required by the identified Process or Experience APIs. This way, some future Process APIs can make use of that data from System APIs and we need NOT touch the System layer APIs again and again.



## Question: 23

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 Answe r: 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 seperately 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".
Reference: https://docs.mulesoft.com/api-manager/2.x/latest-overview-concept

## Question: 24

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 Answe r: 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.
Reference: https://docs.mulesoft.com/api-manager/2.x/using-api-alerts

## Question: 25

A Mule application exposes an HTTPS endpoint and is deployed to the CloudHub Shared Worker Cloud. All traffic to that Mule application must stay inside the AWS VPC.
To what TCP port do API invocations to that Mule application need to be sent?

A. 443
B. 8081
C. 8091
D. 8082

**Answer: D**

Explanation:

Correct Answe r: 8082
*****************************************
>> 8091 and 8092 ports are to be used when keeping your HTTP and HTTPS app private to the LOCAL VPC respectively.
>> Above TWO ports are not for Shared AWS VPC/ Shared Worker Cloud.
>> 8081 is to be used when exposing your HTTP endpoint app to the internet through Shared LB
>> 8082 is to be used when exposing your HTTPS endpoint app to the internet through Shared LB

So, API invocations should be sent to port 8082 when calling this HTTPS based app.
Reference:
https://docs.mulesoft.com/runtime-manager/cloudhub-networking-guide
https://help.mulesoft.com/s/article/Configure-Cloudhub-Application-to-Send-a-HTTPS-Request-Directly-to-Another-Cloudhub-Application
[https://help.mulesoft.com/s/question/0D52T00004mXXULSA4/multiple-http-listerners-on-cloudhub-one-with-port-9090](https://help.mulesoft.com/s/question/0D52T00004mXXULSA4/multiple-http-listerners-on-cloudhub-one-with-port-9090)

## Question: 26

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 Answe r: 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"
Reference:
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/

## Question: 27

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 Answe r: GET, OPTIONS, HEAD

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

## Question: 28

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 Answe r: 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.

## Question: 29

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

Reference:
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

## Question: 30

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 Answe r: 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.

## Question: 31

An API experiences a high rate of client requests (TPS) vwth small message paytoads. 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 Answe r: 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
Reference: https://docs.mulesoft.com/api-manager/2.x/rate-limiting-and-throttling-sla-based-

policies

## Question: 32

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 Answe r: 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
Reference: https://docs.mulesoft.com/exchange/to-use-api-notebook

Bottom of Form
Top of Form

## Question: 33

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 effecuvelyKestrict 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 Answe r: 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.
Reference: https://docs.mulesoft.com/api-manager/2.x/basic-authentication-ldap-concept

## Question: 34

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 Answe r: 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 completly 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