



# MuleSoft

## Exam Questions MCPA-Level-1

MuleSoft Certified Platform Architect - Level 1

## About ExamBible

*[Your Partner of IT Exam](#)*

## Found in 1998

ExamBible is a company specialized on providing high quality IT exam practice study materials, especially Cisco CCNA, CCDA, CCNP, CCIE, Checkpoint CCSE, CompTIA A+, Network+ certification practice exams and so on. We guarantee that the candidates will not only pass any IT exam at the first attempt but also get profound understanding about the certificates they have got. There are so many alike companies in this industry, however, ExamBible has its unique advantages that other companies could not achieve.

## Our Advances

### \* 99.9% Uptime

All examinations will be up to date.

### \* 24/7 Quality Support

We will provide service round the clock.

### \* 100% Pass Rate

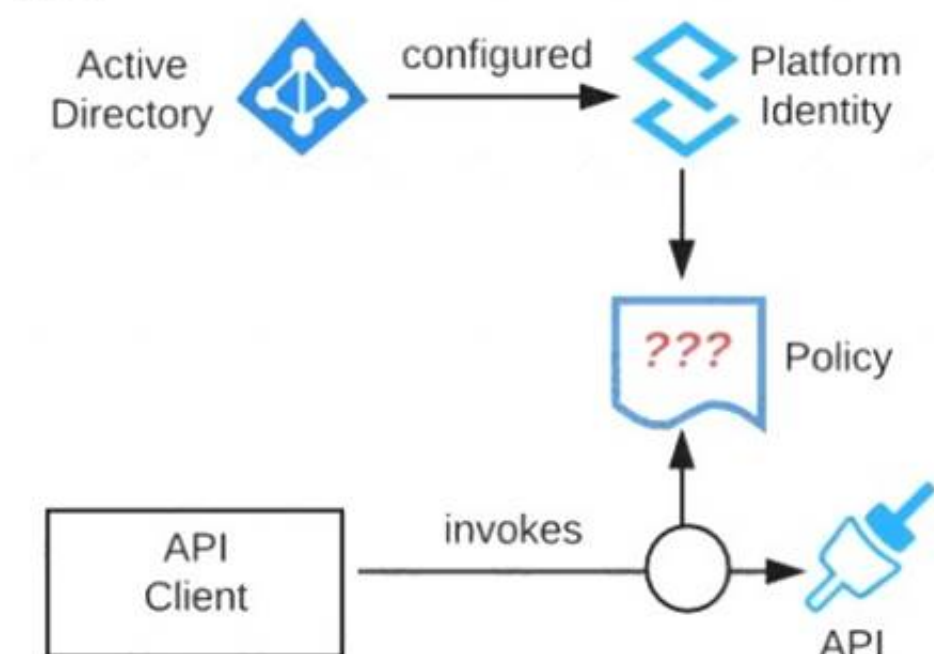
Our guarantee that you will pass the exam.

### \* Unique Gurantee

If you do not pass the exam at the first time, we will not only arrange FULL REFUND for you, but also provide you another exam of your claim, ABSOLUTELY FREE!

### 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

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

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

**Answer: B**

#### Explanation:

Correct Answer

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

\*\*\*\*\*

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

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

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

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

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

#### NEW QUESTION 4

A retail company is using an Order API to accept new orders. The Order API uses a JMS queue to submit orders to a backend order management service. The normal load for orders is being handled using two (2) CloudHub workers, each configured with 0.2 vCore. The CPU load of each CloudHub worker normally runs well below 70%. However, several times during the year the Order API gets four times (4x) the average number of orders. This causes the CloudHub worker CPU load to exceed 90% and the order submission time to exceed 30 seconds. The cause, however, is NOT the backend order management service, which still responds fast enough to meet the response SLA for the Order API. What is the MOST resource-efficient way to configure the Mule application's CloudHub deployment to help the company cope with this performance challenge?

- A. Permanently increase the size of each of the two (2) CloudHub workers by at least four times (4x) to one(1) vCore
- B. Use a vertical CloudHub autoscaling policy that triggers on CPU utilization greater than 70%
- C. Permanently increase the number of CloudHub workers by four times (4x) to eight (8) CloudHub workers
- D. Use a horizontal CloudHub autoscaling policy that triggers on CPU utilization greater than 70%

**Answer: D**

#### Explanation:

Correct Answer

Use a horizontal CloudHub autoscaling policy that triggers on CPU utilization greater than 70%

\*\*\*\*\*

The scenario in the question is very clearly stating that the usual traffic in the year is pretty well handled by the existing worker configuration with CPU running well below 70%. The problem occurs only "sometimes" occasionally when there is spike in the number of orders coming in.

So, based on above, We neither need to permanently increase the size of each worker nor need to permanently increase the number of workers. This is unnecessary as other than those "occasional" times the resources are idle and wasted.

We have two options left now. Either to use horizontal Cloudhub autoscaling policy to automatically increase the number of workers or to use vertical Cloudhub autoscaling policy to automatically increase the vCore size of each worker.

Here, we need to take two things into consideration:

\* 1. CPU

\* 2. Order Submission Rate to JMS Queue

>> From CPU perspective, both the options (horizontal and vertical scaling) solves the issue. Both helps to bring down the usage below 90%.

>> However, If we go with Vertical Scaling, then from Order Submission Rate perspective, as the application is still being load balanced with two workers only, there may not be much improvement in the incoming request processing rate and order submission rate to JMS queue. The throughput would be same as before. Only CPU utilization comes down.

>> But, if we go with Horizontal Scaling, it will spawn new workers and adds extra hand to increase the throughput as more workers are being load balanced now. This way we can address both CPU and Order Submission rate.

Hence, Horizontal CloudHub Autoscaling policy is the right and best answer.

#### NEW QUESTION 5

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

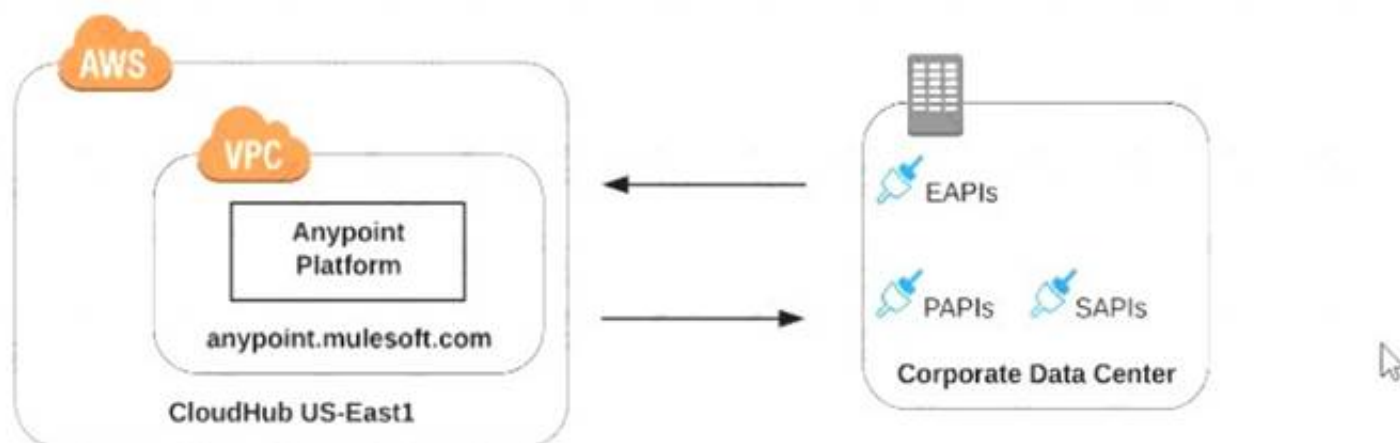
\*\*\*\*\*

>> 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 6

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 Answer

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

◦ **Load balancing**

Load balancing is not provided for hybrid deployments. You can manage load balancing with the tools connected to your on-premises resources.

>> 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.

References:

<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->

<https://help.mulesoft.com/s/article/On-premise-Runtimes-Appear-Disconnected-in-Runtime-Manager-May-29th>

**On-Premise Runtimes Disconnected From US Control Plane - June 18th 2018**

🕒 Jun 19, 2018 - RCA

**Content**

**Impacted Platforms      Impacted Duration**

Anypoint Runtime Manager / On-Prem Runtimes	During this time frame, on-prem runtimes appeared disconnected from the US Anypoint Control Plane:  June 18, 2018 10:35 AM PST to June 18, 2018 11:12 AM PST
---	--

**Incident Description**

On-premises applications weren't able to connect to Anypoint Runtime Manager during the length of the incident, which made on-premises runtimes to throw errors in their logs because they received network disconnect messages from the control plane. Other than generating the log as mentioned above entries, on-premises runtimes and applications were not impacted.

=====

**Runtime Manager cannot manage On-Prem Applications and Servers from US Control Plane - June 25th 2019**

🕒 Jul 3, 2019 - RCA

**Content**

**Incident Summary**

Between 2:51 p.m. PT June 25th and 12:41 a.m. PT June 26th, customers were not able to manage their On-Prem applications and servers. The availability of running applications and runtimes were not impacted.

**Impacted Platforms      Impact Duration**

US-Prod	9 hours and 50 minutes
---------	------------------------

=====



## On-premise Runtimes Appear Disconnected in Runtime Manager - May 29th 2018

🕒 Jun 2, 2018 - RCA

### Content

Impacted Platforms	Impacted Duration
Anypoint Runtime Manager / On-Prem Runtimes	During this time frame, on-prem runtimes appeared disconnected from the US Anypoint Control Plane: Tuesday, May 29, 2018, 3:35 AM PDT to 4:27 AM PDT

### Incident Description

During the incident time frame, managed Runtimes running on-premises disconnected from the US Anypoint Platform Control Plane and may have encountered recurrent re-connection errors. Customers were unable to manage applications running on those runtimes or register new ones during this time. Runtimes and Applications continued to operate without impact.

### 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

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

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

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

**Answer: C**

#### Explanation:

Correct Answer

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

\*\*\*\*\*

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

### NEW QUESTION 9

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

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 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 10

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 Answer 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.

References:

<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-An>

<https://help.mulesoft.com/s/question/0D52T00004mXXULSA4/multiple-http-listeners-on-cloudhub-one-with-p>

#### NEW QUESTION 14

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 18

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

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

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

A. Search Anypoint Exchange for a suitable existing fallback API, and then implement invocations to this fallback API in addition to the Order API

B. Create a separate entry for the Order API in API Manager, and then invoke this API as a fallback API if the primary Order API is unavailable

- C. Redirect client requests through an HTTP 307 Temporary Redirect status code to the fallback API whenever the Order API is unavailable  
D. Set an option in the HTTP Requester component that invokes the Order API to instead invoke a fallback API whenever an HTTP 4xx or 5xx response status code is returned from the Order API

**Answer:** A

**Explanation:**

Correct Answer

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

\*\*\*\*\*

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

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

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

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

#### NEW QUESTION 21

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 22

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 25

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 27

What correctly characterizes unit tests of Mule applications?

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

**Answer: D**

#### Explanation:

Correct Answer

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

\*\*\*\*\*

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

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

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

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

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

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

### NEW QUESTION 32

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 Answer

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.

### NEW QUESTION 34

An organization wants MuleSoft-hosted runtime plane features (such as HTTP load balancing, zero downtime, and horizontal and vertical scaling) in its Azure environment. What runtime plane minimizes the organization's effort to achieve these features?

- A. Anypoint Runtime Fabric
- B. Anypoint Platform for Pivotal Cloud Foundry
- C. CloudHub
- D. A hybrid combination of customer-hosted and MuleSoft-hosted Mule runtimes

**Answer: A**

#### Explanation:

Correct Answer

Anypoint Runtime Fabric

\*\*\*\*\*

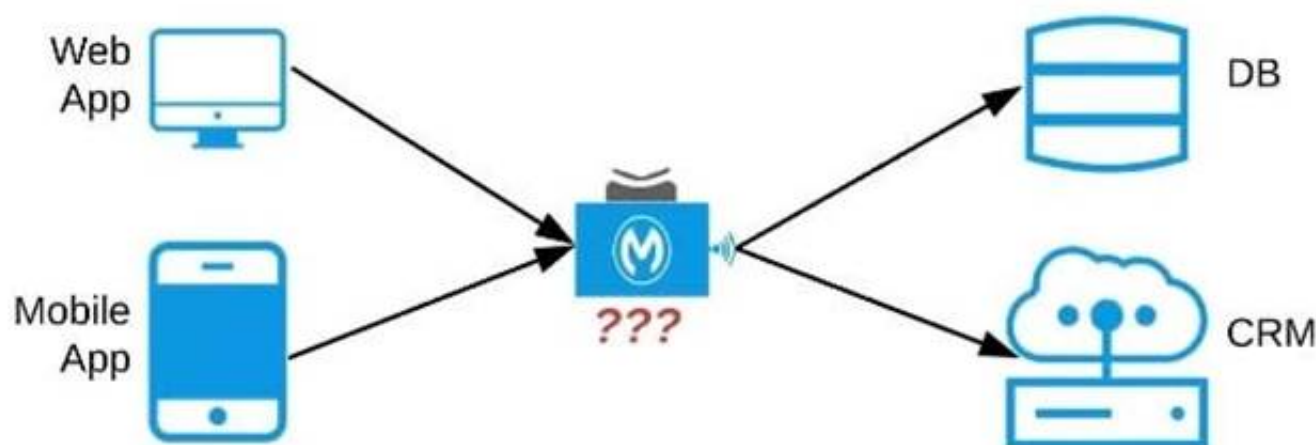
>> When a customer is already having an Azure environment, It is not at all an ideal approach to go with hybrid model having some Mule Runtimes hosted on Azure and some on MuleSoft. This is unnecessary and useless.  
 >> CloudHub is a Mulesoft-hosted Runtime plane and is on AWS. We cannot customize to point CloudHub to customer's Azure environment.  
 >> Anypoint Platform for Pivotal Cloud Foundry is specifically for infrastructure provided by Pivotal Cloud Foundry  
 >> Anypoint Runtime Fabric is right answer as it is a container service that automates the deployment and orchestration of Mule applications and API gateways. Runtime Fabric runs within a customer-managed infrastructure on AWS, Azure, virtual machines (VMs), and bare-metal servers.  
 -Some of the capabilities of Anypoint Runtime Fabric include:  
 -Isolation between applications by running a separate Mule runtime per application.  
 -Ability to run multiple versions of Mule runtime on the same set of resources.  
 -Scaling applications across multiple replicas.  
 -Automated application fail-over.  
 -Application management with Anypoint Runtime Manager.

### NEW QUESTION 37

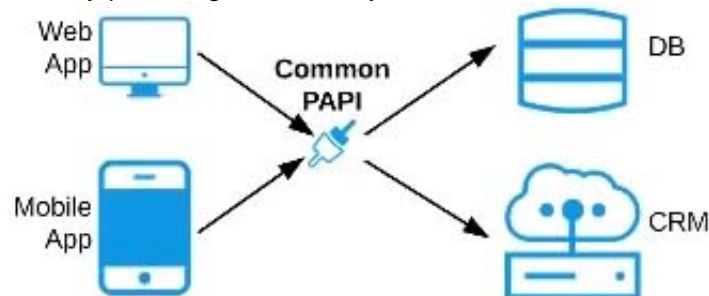
Refer to the exhibit. An organization needs to enable access to their customer data from both a mobile app and a web application, which each need access to common fields as well as certain unique fields.

The data is available partially in a database and partially in a 3rd-party CRM system.

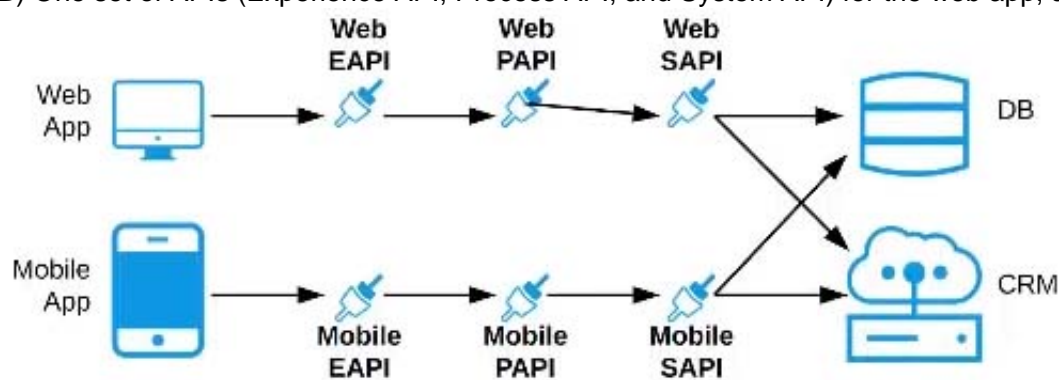
What APIs should be created to best fit these design requirements?



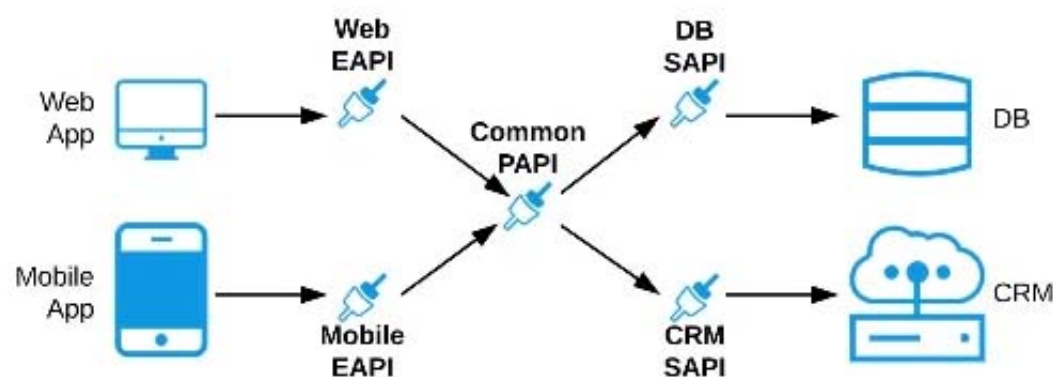
A) A Process API that contains the data required by both the web and mobile apps, allowing these applications to invoke it directly and access the data they need thereby providing the flexibility to add more fields in the future without needing API changes



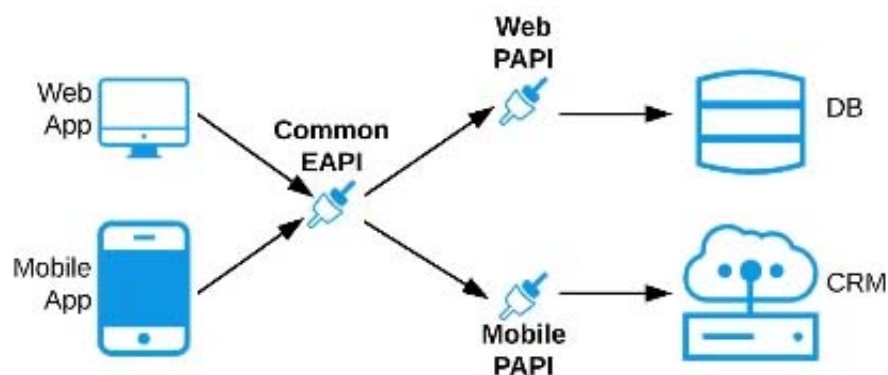
B) One set of APIs (Experience API, Process API, and System API) for the web app, and another set for the mobile app



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 40

What is typically NOT a function of the APIs created within the framework called API-led connectivity?

- A. They provide an additional layer of resilience on top of the underlying backend system, thereby insulating clients from extended failure of these systems.
- B. They allow for innovation at the user Interface level by consuming the underlying assets without being aware of how data is being extracted from backend systems.
- C. They reduce the dependency on the underlying backend systems by helping unlock data from backend systems in a reusable and consumable way.
- D. They can compose data from various sources and combine them with orchestration logic to create higher level value.

**Answer: A**

**Explanation:**

Correct Answer

They provide an additional layer of resilience on top of the underlying backend system, thereby insulating clients from extended failure of these systems.

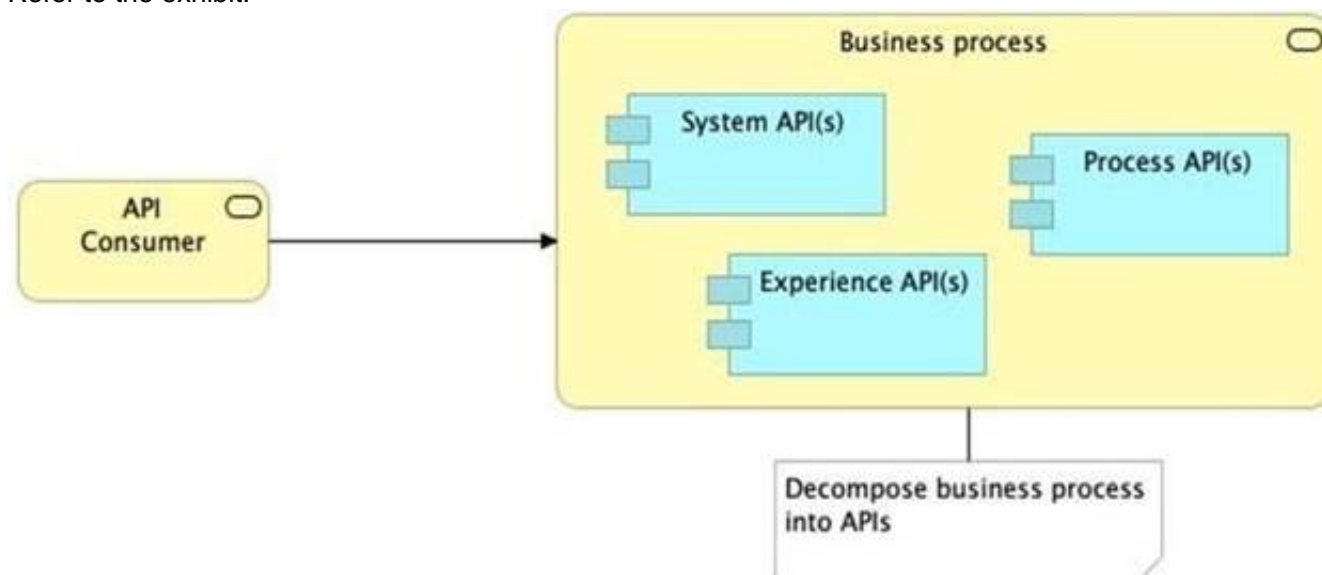
\*\*\*\*\* In API-led connectivity,

- >> Experience APIs - allow for innovation at the user interface level by consuming the underlying assets without being aware of how data is being extracted from backend systems.
  - >> Process APIs - compose data from various sources and combine them with orchestration logic to create higher level value
  - >> System APIs - reduce the dependency on the underlying backend systems by helping unlock data from backend systems in a reusable and consumable way.
- However, they NEVER promise that they provide an additional layer of resilience on top of the underlying backend system, thereby insulating clients from extended failure of these systems.

<https://dzone.com/articles/api-led-connectivity-with-mule>

#### NEW QUESTION 44

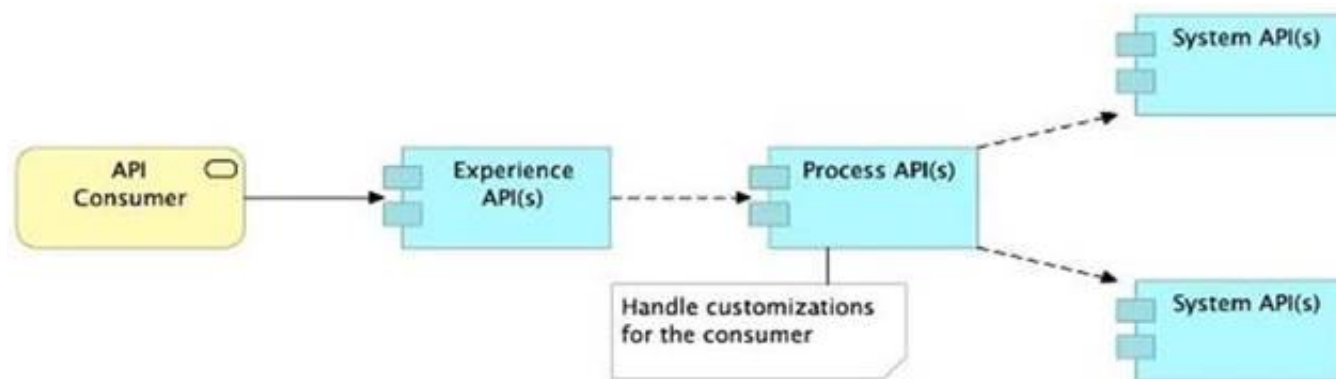
Refer to the exhibit.



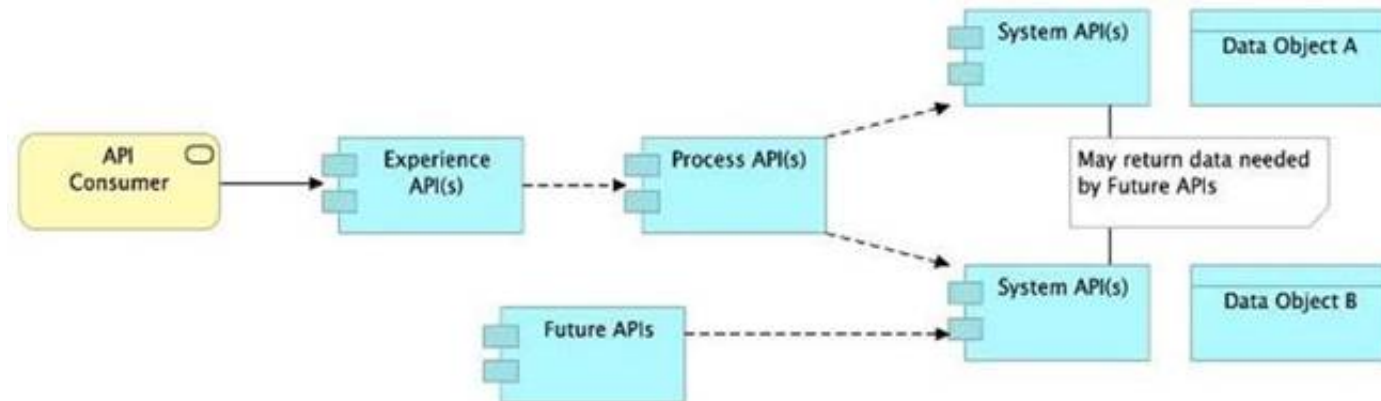
What is the best way to decompose one end-to-end business process into a collaboration of Experience, Process, and System APIs?

- A) Handle customizations for the end-user application at the Process API level rather than the Experience API level

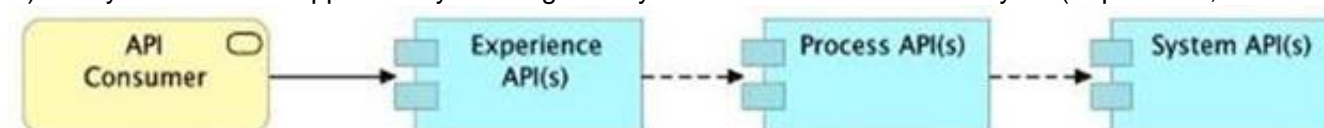




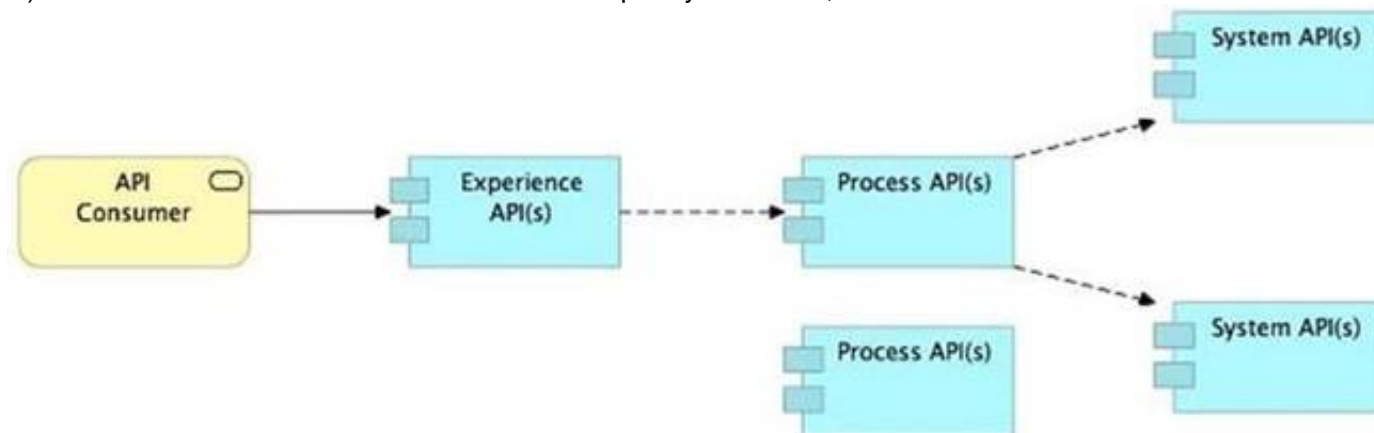
B) Allow System APIs to return data that is NOT currently required by the identified Process or Experience APIs



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 Answer

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.

#### NEW QUESTION 47

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 50

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 51

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 54

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

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

**Answer: C**

#### NEW QUESTION 58

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 61

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. se 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 64

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 69

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

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

**Answer:** B

**Explanation:**

Correct Answer

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

\*\*\*\*\*

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

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

Fine-grained approach:

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

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

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

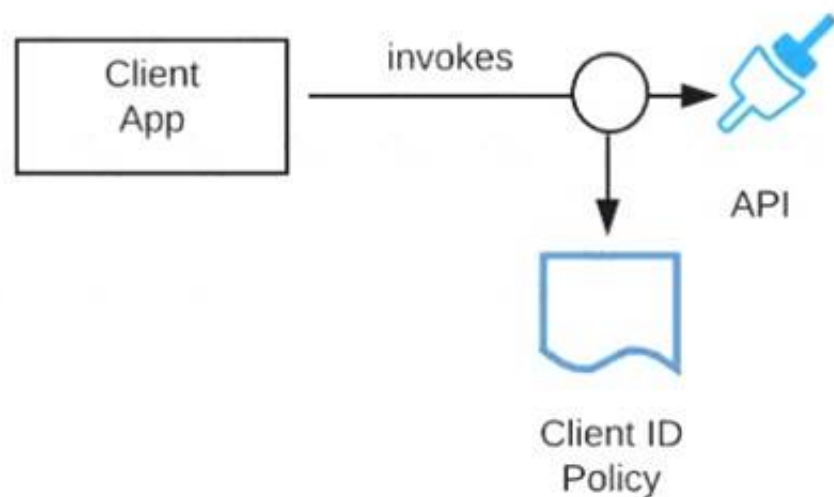
\* 4. That is why, because of all these extra hops and added latencies, fine-grained approach will have bit more response times compared to coarse-grained.

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

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

#### NEW QUESTION 70

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 Answer

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 Enforcement 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.

References:

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>

#### NEW QUESTION 74

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

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

Answer: C

#### Explanation:

Correct Answer

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

\*\*\*\*\*

>> Canonical models are not an option here as the organization has already put in efforts and created bounded-context models for Experience and Process APIs.

>> Anti-corruption layers for ALL APIs is unnecessary and invalid because it is mentioned that experience and process APIs share same bounded-context model.

It is just the System layer APIs that need to choose their approach now.

>> So, having an anti-corruption layer just between the process and system layers will work well. Also to speed up the approach, system APIs can mimic the backend system data model.

#### NEW QUESTION 78

.....



## Relate Links

**100% Pass Your MCPA-Level-1 Exam with ExamBible Prep Materials**

<https://www.exambible.com/MCPA-Level-1-exam/>

## Contact us

We are proud of our high-quality customer service, which serves you around the clock 24/7.

Viste - <https://www.exambible.com/>