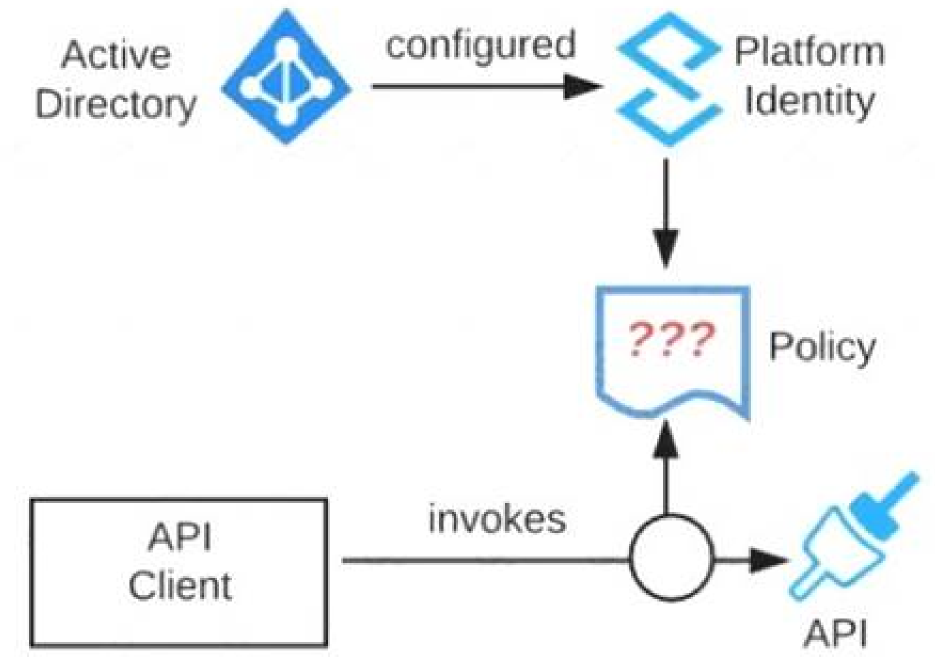
Exam Questions MCPA-Level-1

MuleSoft Certified Platform Architect - Level 1

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

NEW QUESTION 1
Refer to the exhibit. An organization is running a Mule standalone runtime and has configured Active Directory as the Anypoint Platform external Identity Provider. The organization does not have budget for other system components.



What policy should be applied to all instances of APIs in the organization to most 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 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
How are an API implementation, API client, and API consumer combined to invoke and process an API?

A. The API consumer creates an API implementation, which receives API invocations from an API such that they are processed for an API client
B. The API client creates an API consumer, which receives API invocations from an API such that they are processed for an API implementation
C. The ApI consumer creates an API client, which sends API invocations to an API such that they are processed by an API implementation
D. The ApI client creates an API consumer, which sends API invocations to an API such that they are processed by an API implementation

Answer: C

Explanation:

Correct Answer
The API consumer creates an API client, which sends API invocations to an API such that they are processed by an API implementation
*** Terminology:
>> API Client - It is a piece of code or program the is written to invoke an API
>> API Consumer - An owner/entity who owns the API Client. API Consumers write API clients.
>> API - The provider of the API functionality. Typically an API Instance on API Manager where they are managed and operated.
>> API Implementation - The actual piece of code written by API provider where the functionality of the API is implemented. Typically, these are Mule Applications running on Runtime Manager.

NEW QUESTION 3
What is the most performant out-of-the-box solution in Anypoint Platform to track transaction state in an asynchronously executing long-running process implemented as a Mule application deployed to multiple CloudHub workers?

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

Answer: C

Explanation:
Correct Answer
Persistent Object Store
****
>> Redis distributed cache is performant but NOT out-of-the-box solution in Anypoint Platform
>> File-storage is neither performant nor out-of-the-box solution in Anypoint Platform

>> java.util.WeakHashMap needs a completely custom implementation of cache from scratch using Java code and is limited to the JVM where it is running. Which means the state in the cache is not worker aware when running on multiple workers. This type of cache is local to the worker. So, this is neither out-of-the-box nor worker-aware among multiple workers on cloudhub. https://www.baeldung.com/java-weakhashmap
>> Persistent Object Store is an out-of-the-box solution provided by Anypoint Platform which is performant as well as worker aware among multiple workers running on CloudHub. https://docs.mulesoft.com/object-store/
So, Persistent Object Store is the right answer.

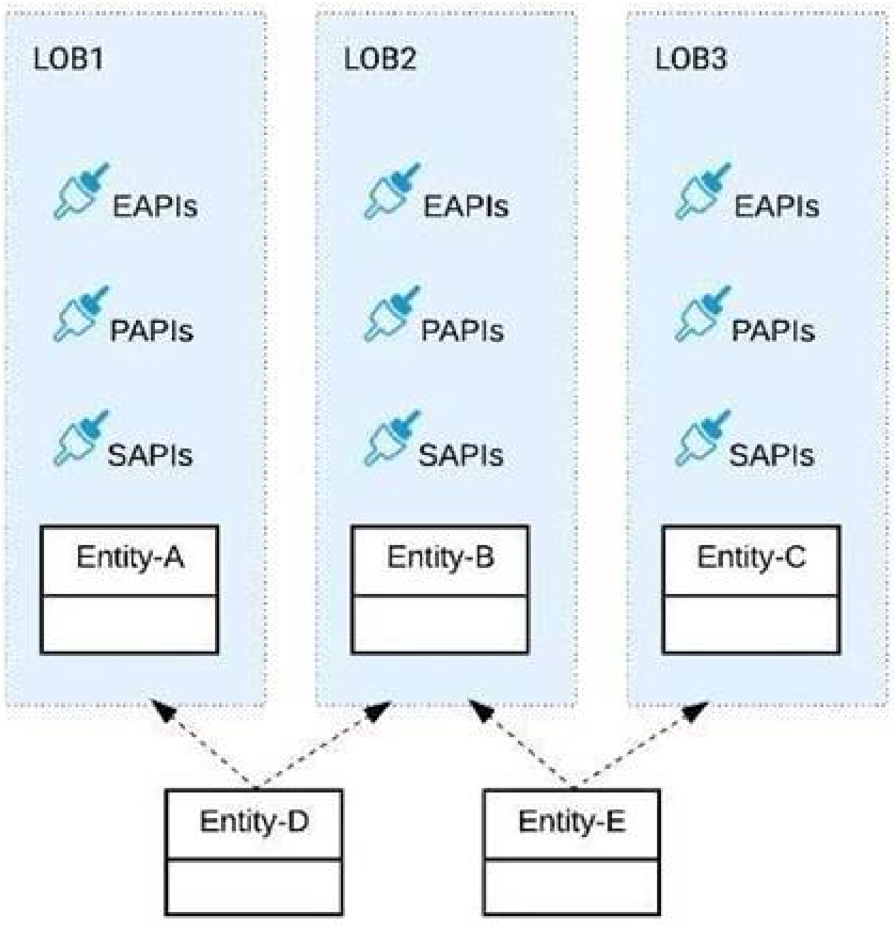
NEW QUESTION 4
Say, there is a legacy CRM system called CRM-Z which is offering below functions:
* 1. Customer creation
* 2. Amend details of an existing customer
* 3. Retrieve details of a customer
* 4. Suspend a customer

A. Implement a system API named customerManagement which has all the functionalities wrapped in it asvarious operations/resources
B. Implement different system APIs named createCustomer, amendCustomer, retrieveCustomer and suspendCustomer as they are modular and has seperation of concerns
C. Implement different system APIs named createCustomerInCRMZ, amendCustomerInCRMZ, retrieveCustomerFromCRMZ and suspendCustomerInCRMZ as they are modular and has seperation of concerns

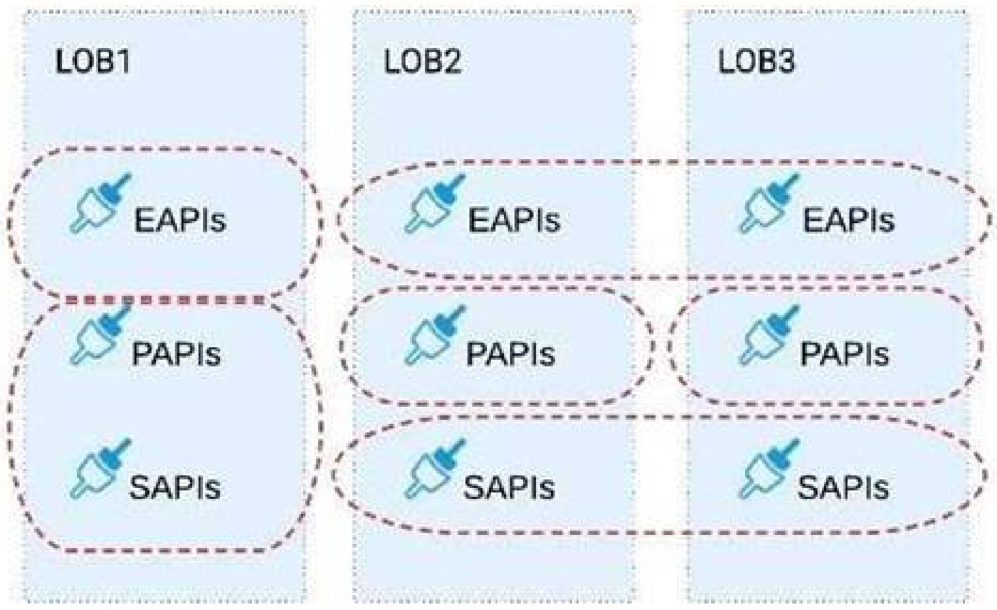
Answer: B

Explanation:
Correct Answer
Implement different system APIs named createCustomer, amendCustomer, retrieveCustomer and suspendCustomer as they are modular and has seperation of concerns
****
>> It is quite normal to have a single API and different Verb + Resource combinations. However, this fits well for an Experience API or a Process API but not a best architecture style for System APIs. So, option with just one customerManagement API is not the best choice here.
>> The option with APIs in createCustomerInCRMZ format is next close choice w.r.t modularization and less maintenance but the naming of APIs is directly coupled with the legacy system. A better foreseen approach would be to name your APIs by abstracting the backend system names as it allows seamless replacement/migration of any backend system anytime. So, this is not the correct choice too.
>> createCustomer, amendCustomer, retrieveCustomer and suspendCustomer is the right approach and is the best fit compared to other options as they are both modular and same time got the names decoupled from backend system and it has covered all requirements a System API needs.

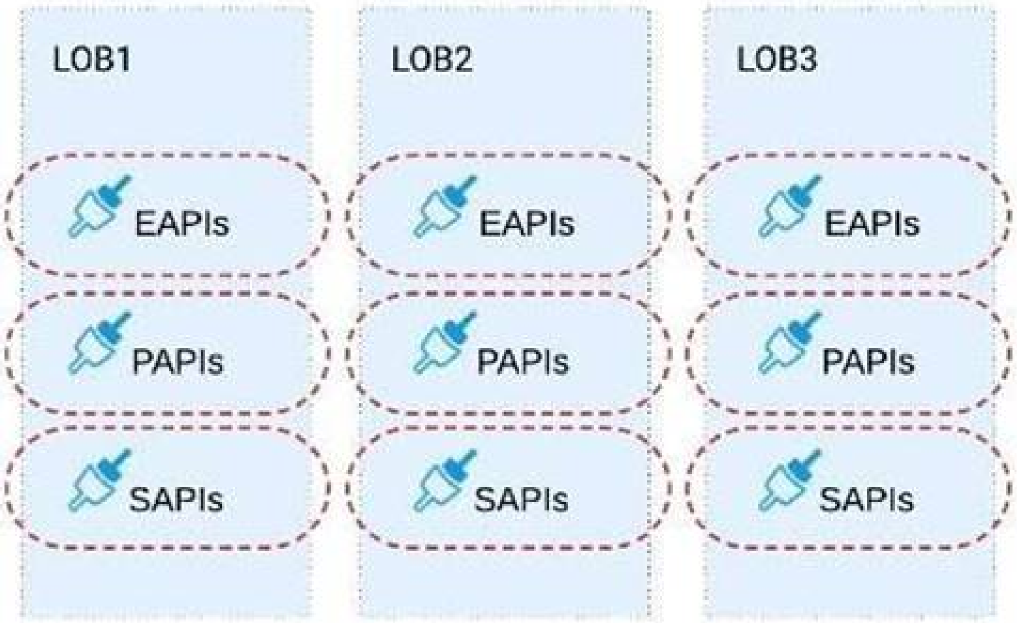
NEW QUESTION 5
Refer to the exhibit.



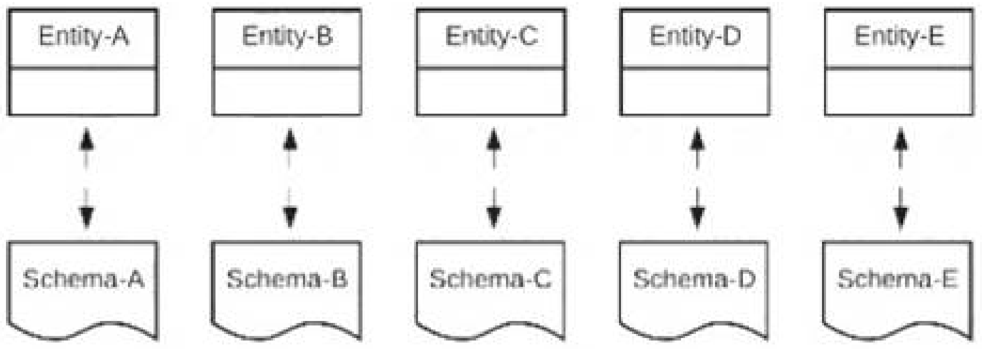
Three business processes need to be implemented, and the implementations need to communicate with several different SaaS applications.
These processes are owned by separate (siloed) LOBs and are mainly independent of each other, but do share a few business entities. Each LOB has one development team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



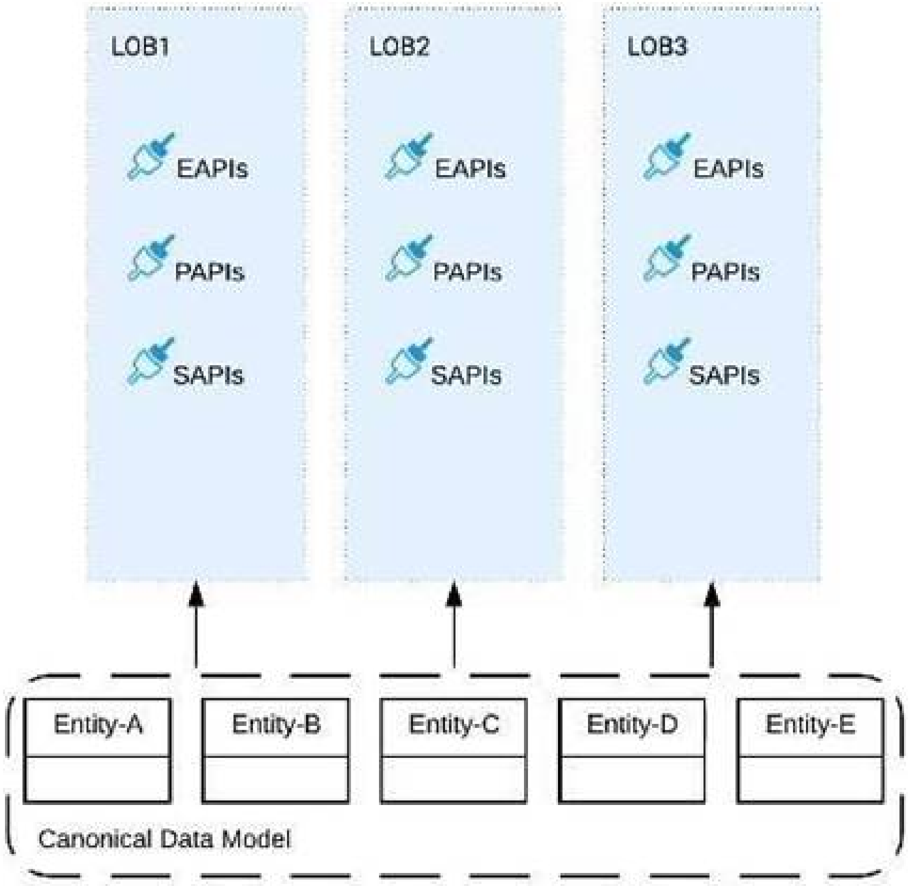
B) Build distinct data models for each API to follow established micro-services and Agile API-centric practices



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



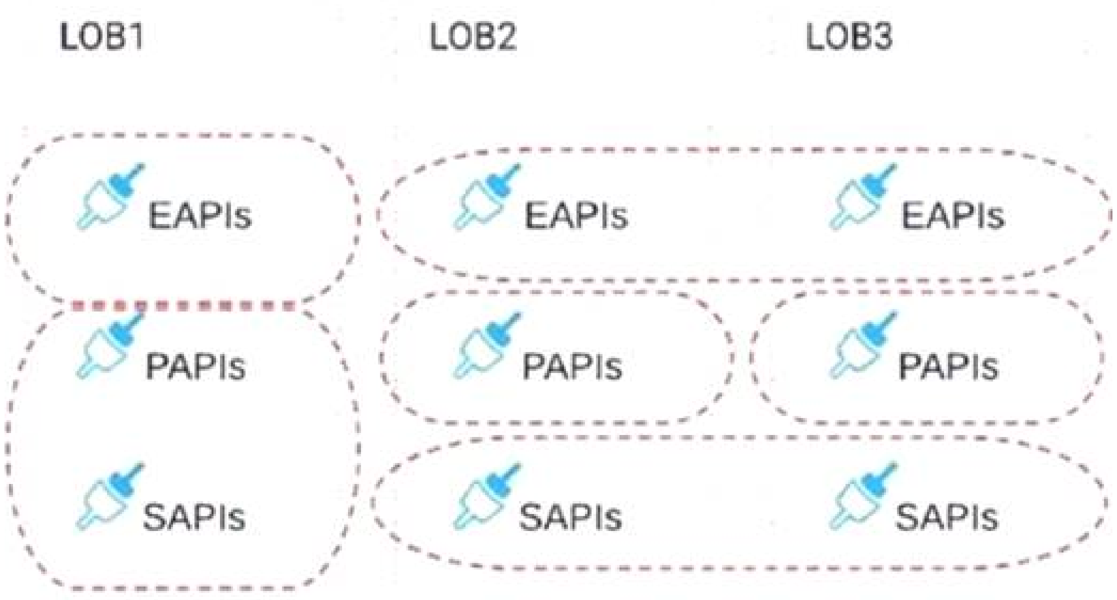
A. Option A

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

Answer: A

Explanation:

Correct Answer
Build several Bounded Context Data Models that align with coherent parts of the business processes and the definitions of associated business entities.
****
>> The options w.r.t building API data models using XML schema/ Agile API-centric practices are irrelevant to the scenario given in the question. So these two are INVALID.
>> Building EDM (Enterprise Data Model) is not feasible or right fit for this scenario as the teams and LOBs work in silo and they all have different initiatives, budget etc.. Building EDM needs intensive coordination among all the team which evidently seems not possible in this scenario.
So, the right fit for this scenario is to build several Bounded Context Data Models that align with coherent parts of the business processes and the definitions of associated business entities.



NEW QUESTION 6
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 firs
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 7
A company uses a hybrid Anypoint Platform deployment model that combines the EU control plane with customer-hosted Mule runtimes. After successfully testing a Mule API implementation in the Staging environment, the Mule API implementation is set with environment-specific properties and must be promoted to the Production environment. What is a way that MuleSoft recommends to configure the Mule API implementation and automate its promotion to the Production environment?

A. Bundle properties files for each environment into the Mule API implementation's deployable archive, then promote the Mule API implementation to the Production environment using Anypoint CLI or the Anypoint Platform REST APIsB.
B. Modify the Mule API implementation's properties in the API Manager Properties tab, then promote the Mule API implementation to the Production environment using API Manager
C. Modify the Mule API implementation's properties in Anypoint Exchange, then promote the Mule API implementation to the Production environment using Runtime Manager
D. Use an API policy to change properties in the Mule API implementation deployed to the Staging environment and another API policy to deploy the Mule API implementation to the Production environment

Answer: A

Explanation:

Correct Answer
Bundle properties files for each environment into the Mule API implementation's deployable archive, then promote the Mule API implementation to the Production environment using Anypoint CLI or the Anypoint Platform REST APIs
****
>> Anypoint Exchange is for asset discovery and documentation. It has got no provision to modify the properties of Mule API implementations at all.

>> API Manager is for managing API instances, their contracts, policies and SLAs. It has also got no provision to modify the properties of API implementations.
>> API policies are to address Non-functional requirements of APIs and has again got no provision to modify the properties of API implementations.
So, the right way and recommended way to do this as part of development practice is to bundle properties files for each environment into the Mule API implementation and just point and refer to respective file per environment.

NEW QUESTION 8
An API implementation is updated. When must the RAML definition of the API also be updated?

A. When the API implementation changes the structure of the request or response messages
B. When the API implementation changes from interacting with a legacy backend system deployedon-premises to a modern, cloud-based (SaaS) system
C. When the API implementation is migrated from an older to a newer version of the Mule runtime
D. When the API implementation is optimized to improve its average response time

Answer: A

Explanation:
Correct Answer
When the API implementation changes the structure of the request or response messages
****
>> RAML definition usually needs to be touched only when there are changes in the request/response schemas or in any traits on API.
>> It need not be modified for any internal changes in API implementation like performance tuning, backend system migrations etc..

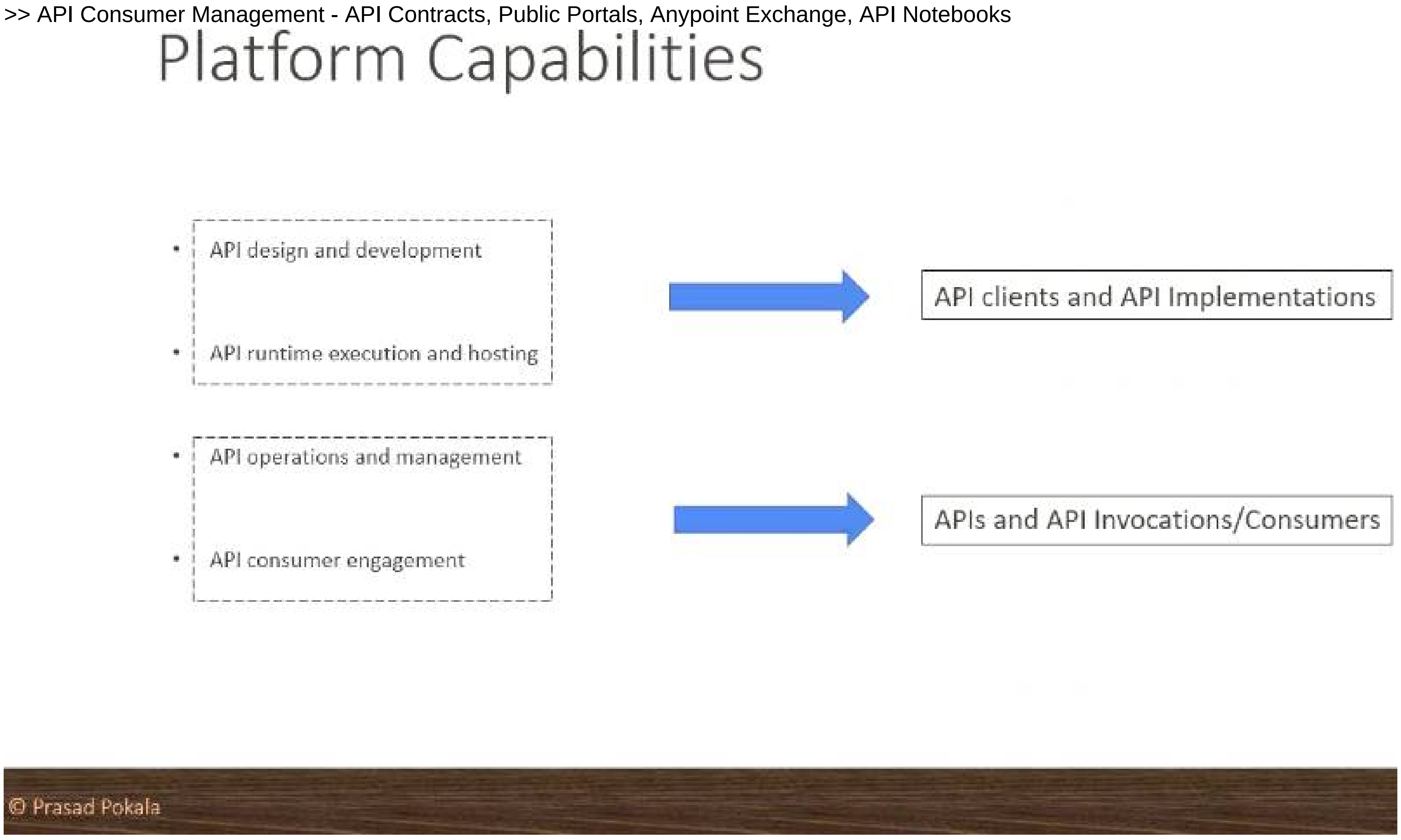
NEW QUESTION 9
What are 4 important Platform Capabilities offered by Anypoint Platform?

A. API Versioning, API Runtime Execution and Hosting, API Invocation, API Consumer Engagement
B. API Design and Development, API Runtime Execution and Hosting, API Versioning, API Deprecation
C. API Design and Development, API Runtime Execution and Hosting, API Operations and Management, API Consumer Engagement
D. API Design and Development, API Deprecation, API Versioning, API Consumer Engagement

Answer: C

Explanation:

Correct Answer
API Design and Development, API Runtime Execution and Hosting, API Operations and Management, API Consumer Engagement
****
>> API Design and Development - Anypoint Studio, Anypoint Design Center, Anypoint Connectors
>> API Runtime Execution and Hosting - Mule Runtimes, CloudHub, Runtime Services
>> API Operations and Management - Anypoint API Manager, Anypoint Exchange
>> API Consumer Management - API Contracts, Public Portals, Anypoint Exchange, API Notebooks


Platform Capabilities

- API design and development
- API runtime execution and hosting

API clients and API Implementations

- API operations and management
- API consumer engagement

APIs and API Invocations/Consumers

© Prasad Pokala

NEW QUESTION 10
What Mule application deployment scenario requires using Anypoint Platform Private Cloud Edition or Anypoint Platform for Pivotal Cloud Foundry?

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

Answer: C

Explanation:
Correct Answer
When regulatory requirements mandate on-premises processing of EVERY data item, including meta-data.
****
We need NOT require to use Anypoint Platform PCE or PCF for the below. So these options are OUT.
>> We can make ALL applications highly available across multiple data centers using CloudHub too.
>> We can use Anypoint VPN and tunneling from CloudHub to connect to ALL backend systems in the application network that are deployed in the organization's intranet.
>> We can use Anypoint VPC and Firewall Rules to make ALL APIs private and NOT exposed to the public cloud.
Only valid reason in the given options that requires to use Anypoint Platform PCE/ PCF is - When regulatory requirements mandate on-premises processing of EVERY data item, including meta-data.

NEW QUESTION 10
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 15
When could the API data model of a System API reasonably mimic the data model exposed by the corresponding backend system, with minimal improvements over the backend system's data model?

A. When there is an existing Enterprise Data Model widely used across the organization
B. When the System API can be assigned to a bounded context with a corresponding data model
C. When a pragmatic approach with only limited isolation from the backend system is deemed appropriate
D. When the corresponding backend system is expected to be replaced in the near future

Answer: C

Explanation:

Correct Answer
When a pragmatic approach with only limited isolation from the backend system is deemed appropriate.
*** General guidance w.r.t choosing Data Models:
>> If an Enterprise Data Model is in use then the API data model of System APIs should make use of data types from that Enterprise Data Model and the corresponding API implementation should translate between these data types from the Enterprise Data Model and the native data model of the backend system.
>> If no Enterprise Data Model is in use then each System API should be assigned to a Bounded Context, the API data model of System APIs should make use of data types from the corresponding Bounded Context Data Model and the corresponding API implementation should translate between these data types from the Bounded Context Data Model and the native data model of the backend system. In this scenario, the data types in the Bounded Context Data Model are defined purely in terms of their business characteristics and are typically not related to the native data model of the backend system. In other words, the translation effort may be significant.
>> If no Enterprise Data Model is in use, and the definition of a clean Bounded Context Data Model is considered too much effort, then the API data model of System APIs should make use of data types that approximately mirror those from the backend system, same semantics and naming as backend system, lightly sanitized, expose all fields needed for the given System API's functionality, but not significantly more and making good use of REST conventions.
The latter approach, i.e., exposing in System APIs an API data model that basically mirrors that of the backend system, does not provide satisfactory isolation from backend systems through the System API tier on its own. In particular, it will typically not be possible to "swap out" a backend system without significantly changing all System APIs in front of that backend system and therefore the API implementations of all Process APIs that depend on those System APIs! This is so because it is not desirable to prolong the life of a previous backend system's data model in the form of the API data model of System APIs that now front a new backend system. The API data models of System APIs following this approach must therefore change when the backend system is replaced.
On the other hand:
>> It is a very pragmatic approach that adds comparatively little overhead over accessing the backend system directly
>> Isolates API clients from intricacies of the backend system outside the data model (protocol, authentication, connection pooling, network address, …)
>> Allows the usual API policies to be applied to System APIs
>> Makes the API data model for interacting with the backend system explicit and visible, by exposing it in the RAML definitions of the System APIs
>> Further isolation from the backend system data model does occur in the API implementations of the Process API tier

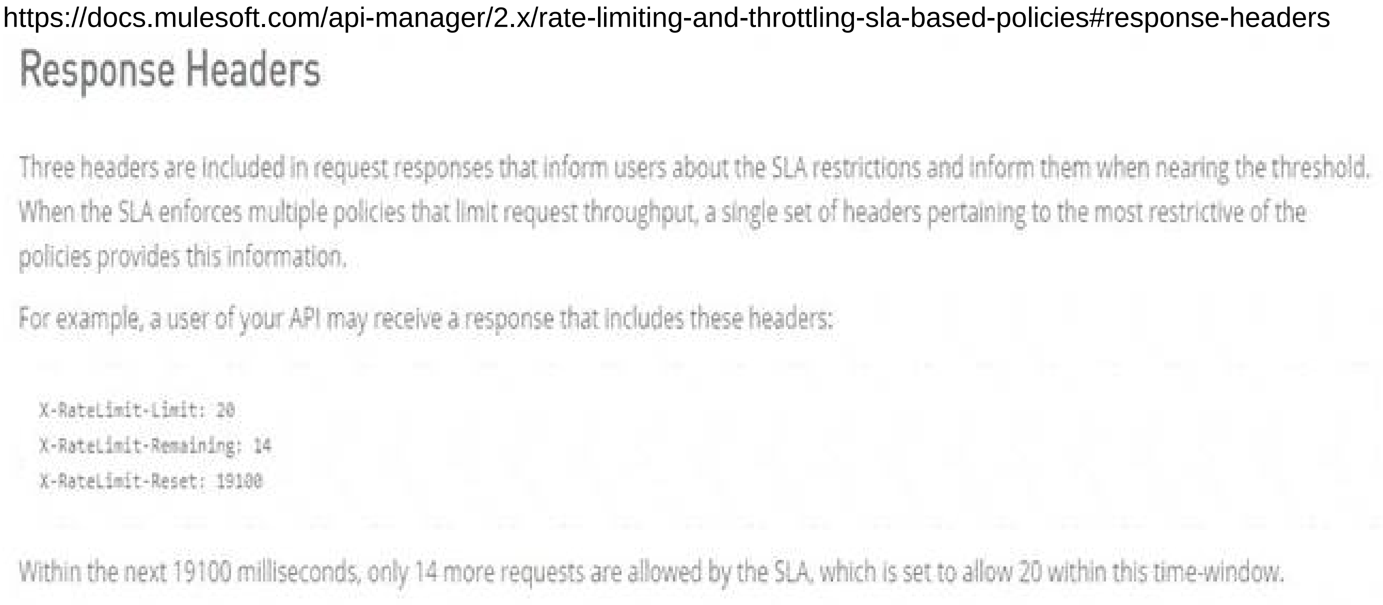
NEW QUESTION 20
Question 10: Skipped
An API implementation returns three X-RateLimit-* HTTP response headers to a requesting API client. What type of information do these response headers indicate to the API client?

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 21
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 Answer
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.

NEW QUESTION 23
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 24
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.

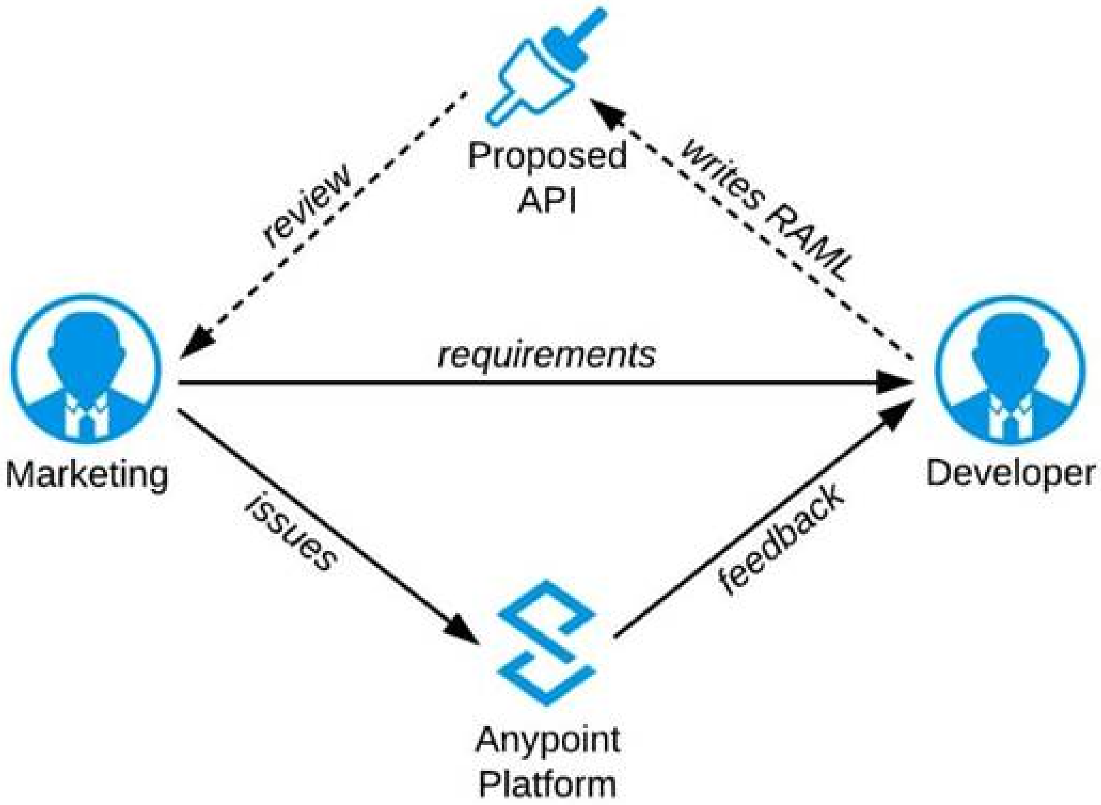
NEW QUESTION 27
What is a key performance indicator (KPI) that measures the success of a typical C4E that is immediately apparent in responses from the Anypoint Platform APIs?

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

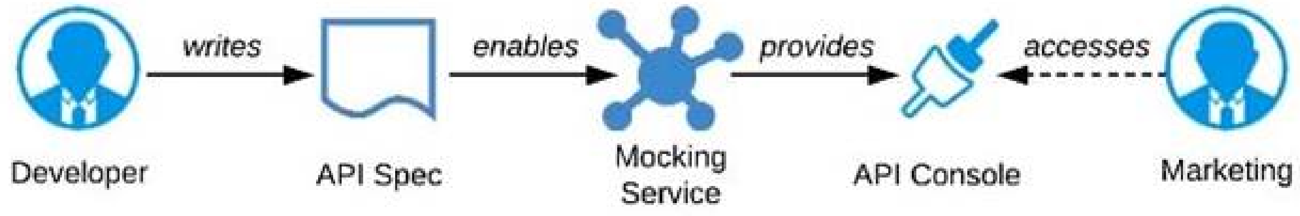
Answer: D

Explanation:
Correct Answer
The number of API specifications in RAML or OAS format published to Anypoint Exchange
****
>> The success of C4E always depends on their contribution to the number of reusable assets that they have helped to build and publish to Anypoint Exchange.
>> It is NOT due to any factors w.r.t # of outages, Manual vs CI/CD deployments or Publicly accessible HTTP endpoints
>> Anypoint Platform APIs helps us to quickly run and get the number of published RAML/OAS assets to Anypoint Exchange. This clearly depicts how successful a C4E team is based on number of returned assets in the response.

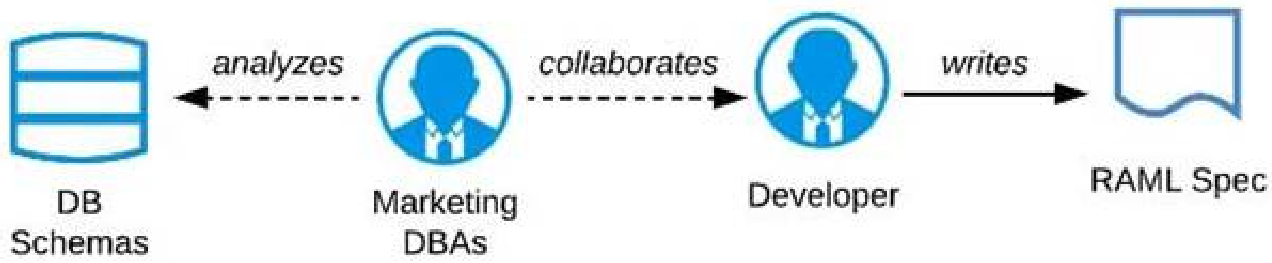
NEW QUESTION 32
Refer to the exhibit.



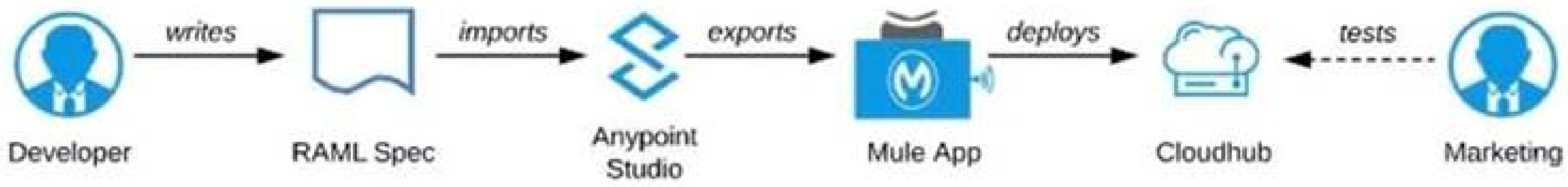
A RAML definition has been proposed for a new Promotions Process API, and has been published to
Anypoint Exchange.
The Marketing Department, who will be an important consumer of the Promotions API, has important requirements and expectations that must be met.
What is the most effective way to use Anypoint 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



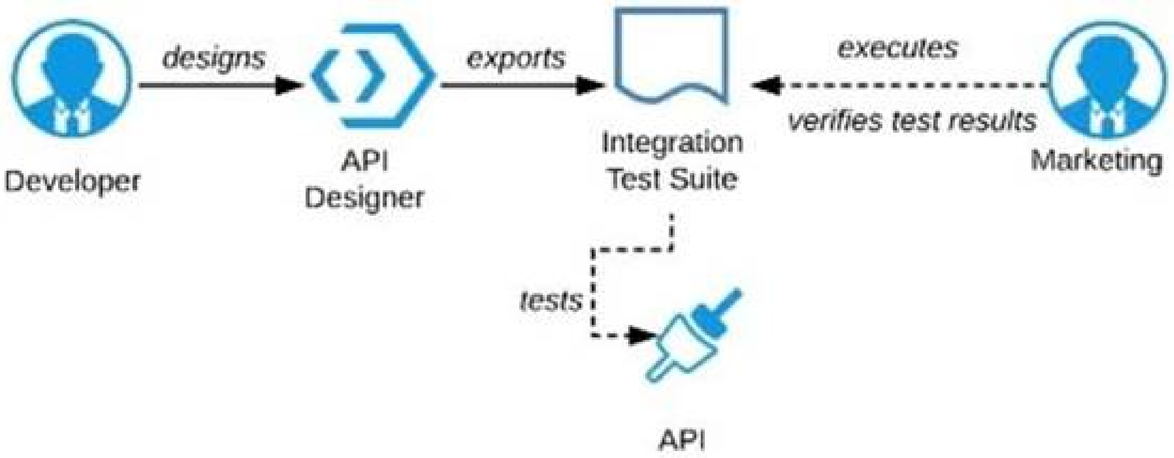
B) Organize a design workshop with the DBAs of the Marketing Department in which the database schema of the Marketing IT systems is translated into RAML



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 Answer
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.

NEW QUESTION 33
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 35
The implementation of a Process API must change.
What is a valid approach that minimizes the impact of this change on API clients?

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

Answer: C

Explanation:
Correct Answer
Implement required changes to the Process API implementation so that, whenever possible, the Process API's RAML definition remains unchanged.
** Key requirement in the question is:
>> Approach that minimizes the impact of this change on API clients Based on above:

>> Updating the RAML definition would possibly impact the API clients if the changes require any thing mandatory from client side. So, one should try to avoid doing that until really necessary.
>> Implementing the changes as a completely different API and then redirectly the clients with 3xx status code is really upsetting design and heavily impacts the API clients.
>> Organisations and IT cannot simply postpone the changes required until all API consumers acknowledge they are ready to migrate to a new Process API or API version. This is unrealistic and not possible.
The best way to handle the changes always is to implement required changes to the API implementations so that, whenever possible, the API's RAML definition remains unchanged.

NEW QUESTION 40
A system API has a guaranteed SLA of 100 ms per request. The system API is deployed to a primary environment as well as to a disaster recovery (DR) environment, with different DNS names in each environment. An upstream process API invokes the system API and the main goal of this process API is to respond to client requests in the least possible time. In what order should the system APIs be invoked, and what changes should be made in order to speed up the response time for requests from the process API?

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

Answer: A

Explanation:
Correct Answer
In parallel, invoke the system API deployed to the primary environment and the system API deployed to the DR environment, and ONLY use the first response.
****
>> The API requirement in the given scenario is to respond in least possible time.
>> The option that is suggesting to first try the API in primary environment and then fallback to API in DR environment would result in successful response but NOT in least possible time. So, this is NOT a right choice of implementation for given requirement.
>> Another option that is suggesting to ONLY invoke API in primary environment and to add timeout and retries may also result in successful response upon retries but NOT in least possible time. So, this is also NOT a right choice of implementation for given requirement.
>> One more option that is suggesting to invoke API in primary environment and API in DR environment in parallel using Scatter-Gather would result in wrong API response as it would return merged results and moreover, Scatter-Gather does things in parallel which is true but still completes its scope only on finishing all routes inside it. So again, NOT a right choice of implementation for given requirement
The Correct choice is to invoke the API in primary environment and the API in DR environment parallelly, and using ONLY the first response received from one of them.

NEW QUESTION 43
......