

MuleSoft.MCPA-Level-1-Maintenance.v2023-04-10.q33

Exam Code:	MCPA-Level-1-Maintenance
Exam Name:	MuleSoft Certified Platform Architect - Level 1 MAINTENANCE
Certification Provider:	MuleSoft
Free Question Number:	33
Version:	v2023-04-10
# of views:	842
# of Questions views:	330
https://www.freepdfdumps.com/MuleSoft.MCPA-Level-1-Maintenance.v2023-04-10.q33.html	

NEW QUESTION: 1

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: (SHOW 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: 2

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: (SHOW 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: 3

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 (LEAVE A REPLY)

GET, OPTIONS, HEAD

APIs use HTTP-based protocols: cached HTTP responses from previous HTTP requests may potentially be returned if the same HTTP request is seen again.

Safe HTTP methods are ones that do not alter the state of the underlying resource. That is, the *HTTP responses to requests using safe HTTP methods may be cached*.

The HTTP standard requires the following HTTP methods on any resource to be safe:

- GET
- HEAD
- OPTIONS

Safety must be honored by REST APIs (but not by non-REST APIs like SOAP APIs): It is the *responsibility of every API implementation* to implement GET, HEAD or OPTIONS methods such that they never change the state of a resource.

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

NEW QUESTION: 4

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: ([SHOW 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 that 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: 5

An Anypoint Platform organization has been configured with an external identity provider (IdP) for identity management and client management. What credentials or token must be provided to Anypoint CLI to execute commands against the Anypoint Platform APIs?

- A.** The credentials provided by the IdP for identity management
- B.** The credentials provided by the IdP for client management
- C.** An OAuth 2.0 token generated using the credentials provided by the IdP for client management
- D.** An OAuth 2.0 token generated using the credentials provided by the IdP for identity management

Answer: ([SHOW ANSWER](#))

The credentials provided by the IdP for identity management

NEW QUESTION: 6

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 (LEAVE A REPLY)

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: 7

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

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

Answer: A (LEAVE A REPLY)

Guarding against Denial of Service attacks

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

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

>> Credentials can be tamper-proofed using "Security" and "Compliance" Policies However, unfortunately, there is no proper way currently on Anypoint Platform to guard against DOS attacks.

NEW QUESTION: 8

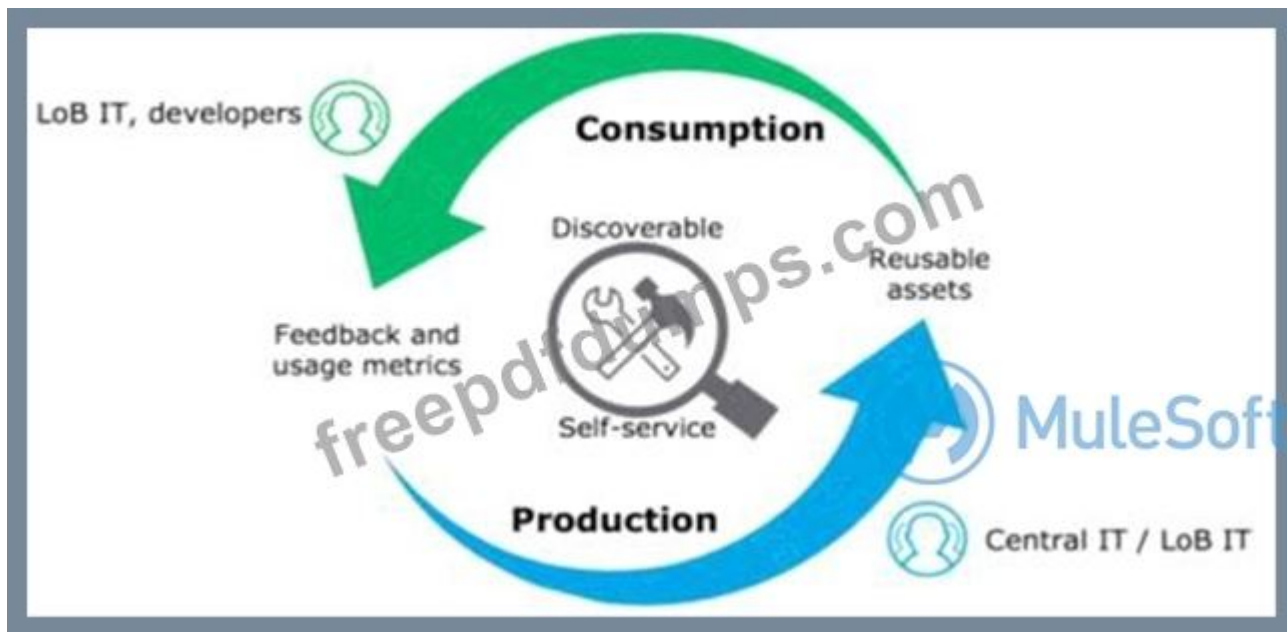
Which of the below, when used together, makes the IT Operational Model effective?

- A. Create reusable assets, Do marketing on the created assets across organization, Arrange time to time LOB reviews to ensure assets are being consumed or not
- B. Create reusable assets, Make them discoverable so that LOB teams can self-serve and browse the APIs, Get active feedback and usage metrics
- C. Create reusable assets, make them discoverable so that LOB teams can self-serve and browse the APIs

Answer: C (LEAVE A REPLY)

Create reusable assets, Make them discoverable so that LOB teams can self-serve and browse the APIs, Get active feedback and usage metrics.

Diagram, arrow Description automatically generated



NEW QUESTION: 9

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 ([LEAVE A REPLY](#))

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

Response Headers

The following access-limiting policies return headers having information about the current state of the request:

- X-Ratelimit-Remaining: The amount of available quota.
- X-Ratelimit-Limit: The maximum available requests per window.
- X-Ratelimit-Reset: The remaining time, in milliseconds, until a new window starts.

Response Headers

Three headers are included in request responses that inform users about the SLA restrictions and inform them when nearing the threshold. When the SLA enforces multiple policies that limit request throughput, a single set of headers pertaining to the most restrictive of the policies provides this information.

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

```
X-RateLimit-Limit: 20
X-RateLimit-Remaining: 14
X-RateLimit-Reset: 19100
```

Within the next 19100 milliseconds, only 14 more requests are allowed by the SLA, which is set to allow 20 within this time-window.

References:

<https://docs.mulesoft.com/api-manager/2.x/rate-limiting-and-throttling#response-headers>

<https://docs.mulesoft.com/api-manager/2.x/rate-limiting-and-throttling-sla-based-policies#response-headers>

NEW QUESTION: 10

What is true about automating interactions with Anypoint Platform using tools such as Anypoint Platform REST APIs, Anypoint CU, or the Mule Maven plugin?

- A.** Access to Anypoint Platform APIs and Anypoint CU can be controlled separately through the roles and permissions in Anypoint Platform, so that specific users can get access to Anypoint CLI while others get access to the platform APIs
- B.** Anypoint Platform APIs can ONLY automate interactions with CloudHub, while the Mule Maven plugin is required for deployment to customer-hosted Mule runtimes
- C.** By default, the Anypoint CLI and Mule Maven plugin are NOT included in the Mule runtime, so are NOT available to be used by deployed Mule applications
- D.** API policies can be applied to the Anypoint Platform APIs so that ONLY certain LOBs have access to specific functions

Answer: (SHOW ANSWER)

By default, the Anypoint CLI and Mule Maven plugin are NOT included in the Mule runtime, so are NOT available to be used by deployed Mule applications

>> We CANNOT apply API policies to the Anypoint Platform APIs like we can do on our custom written API instances. So, option suggesting this is FALSE.

>> Anypoint Platform APIs can be used for automating interactions with both CloudHub and customer-hosted Mule runtimes. Not JUST the CloudHub. So, option opposing this is FALSE.

>> Mule Maven plugin is NOT mandatory for deployment to customer-hosted Mule runtimes. It just helps your CI/CD to have smoother automation. But not a compulsory requirement to deploy. So, option opposing this is FALSE.

>> We DO NOT have any such special roles and permissions on the platform to separately control access for some users to have Anypoint CLI and others to have Anypoint Platform APIs. With proper general roles/permissions (API Owner, Cloudhub Admin etc..), one can use any of the options (Anypoint CLI or Platform APIs). So, option suggesting this is FALSE.

Only TRUE statement given in the choices is that - Anypoint CLI and Mule Maven plugin are NOT included in the Mule runtime, so are NOT available to be used by deployed Mule applications. Maven is part of Studio or you can use other Maven installation for development. CLI is convenience only. It is one of many ways how to install app to the runtime. These are definitely NOT part of anything except your process of deployment or automation.

NEW QUESTION: 11

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: ([SHOW ANSWER](#))

Unit tests (Whitebox)

NEW QUESTION: 12

Due to a limitation in the backend system, a system API can only handle up to 500 requests per second. What is the best type of API policy to apply to the system API to avoid overloading the backend system?

- A. Rate limiting
- B. HTTP caching
- C. Rate limiting - SLA based
- D. Spike control

Answer: ([SHOW ANSWER](#))

Spike control

>> First things first, HTTP Caching policy is for purposes different than avoiding the backend system from overloading. So this is OUT.

>> Rate Limiting and Throttling/ Spike Control policies are designed to limit API access, but have different intentions.

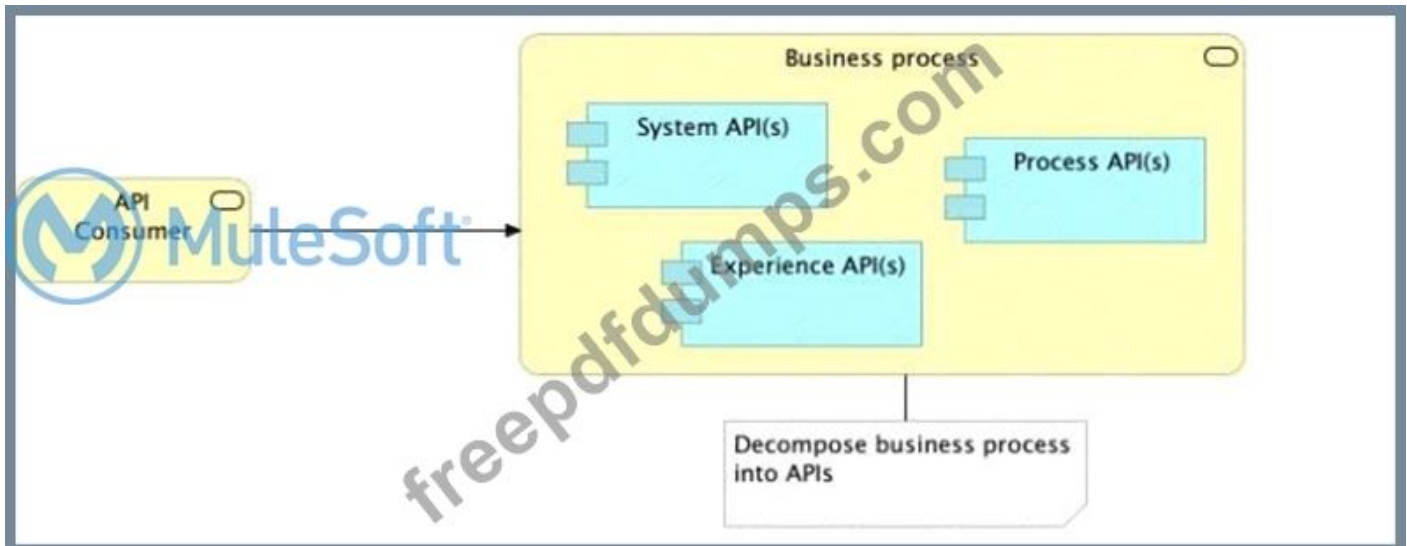
>> Rate limiting protects an API by applying a hard limit on its access.

>> Throttling/ Spike Control shapes API access by smoothing spikes in traffic.

That is why, Spike Control is the right option.

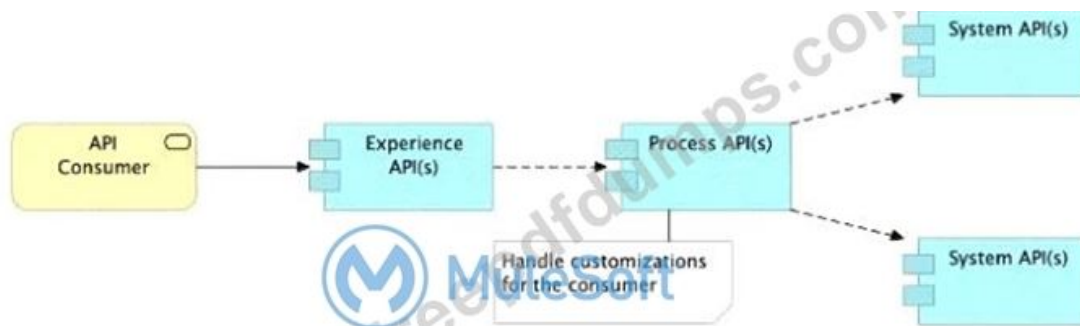
NEW QUESTION: 13

Refer to the exhibit.

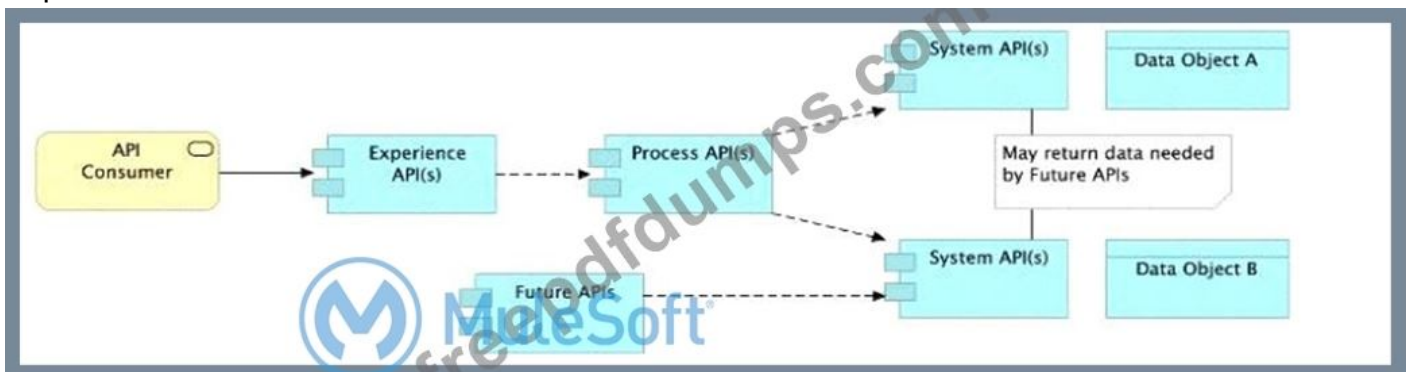


What is the best way to decompose one end-to-end business process into a collaboration of Experience, Process, and System APIs?

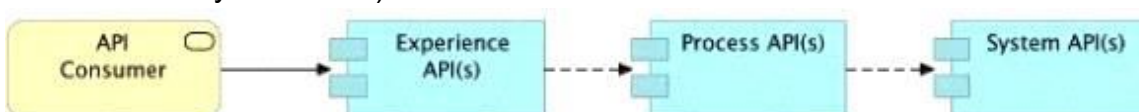
A) Handle customizations for the end-user application at the Process API level rather than the Experience API level



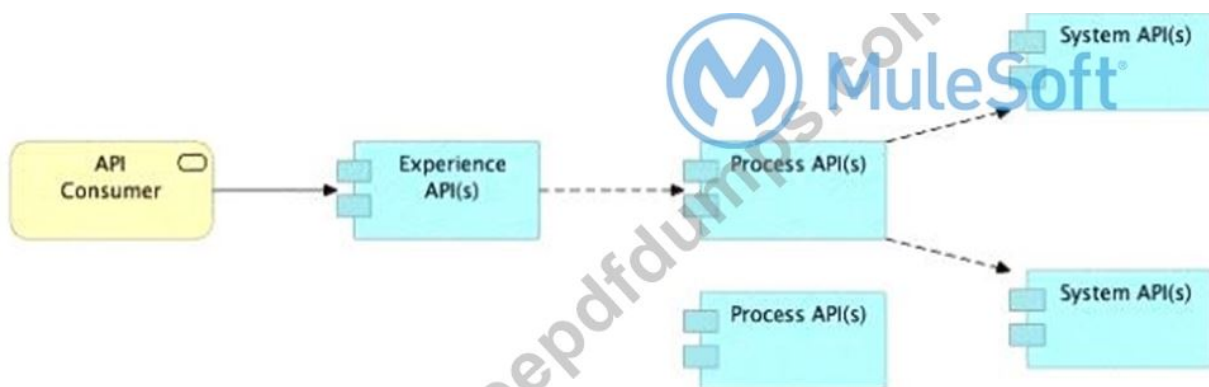
B) Allow System APIs to return data that is NOT currently required by the identified Process or Experience APIs



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 (LEAVE A REPLY)

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

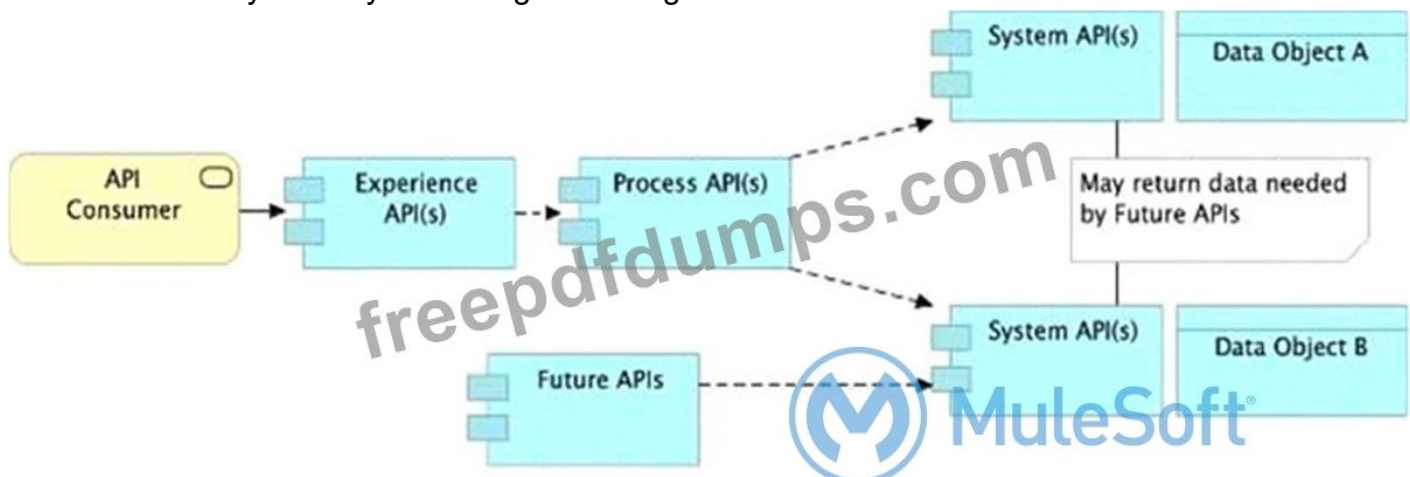
>> All customizations for the end-user application should be handled in "Experience API" only. Not in Process API

>> 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: 14

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 (LEAVE A REPLY)

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%.

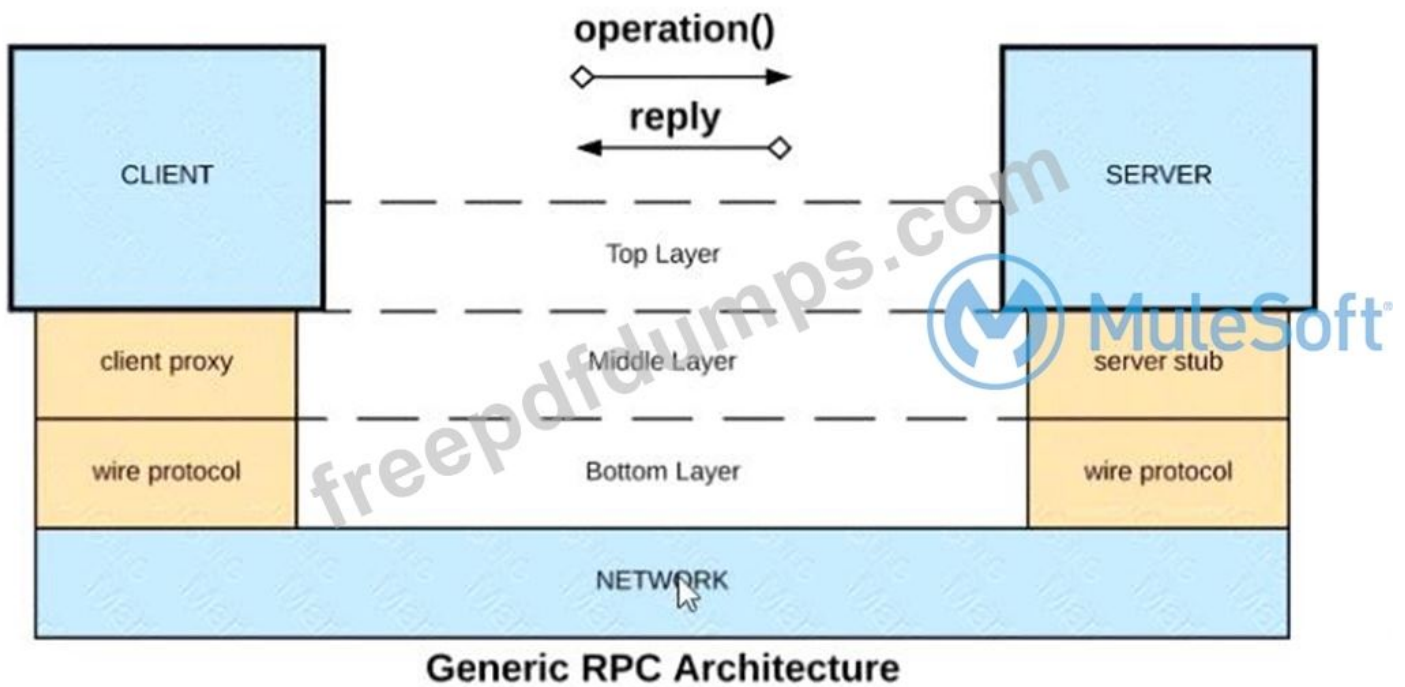
>> However, If we go with Vertical Scaling, then from Order Submission Rate perspective, as the application is still being load balanced with two workers only, there may not be much improvement in the incoming request processing rate and order submission rate to JMS queue. The throughput would be same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

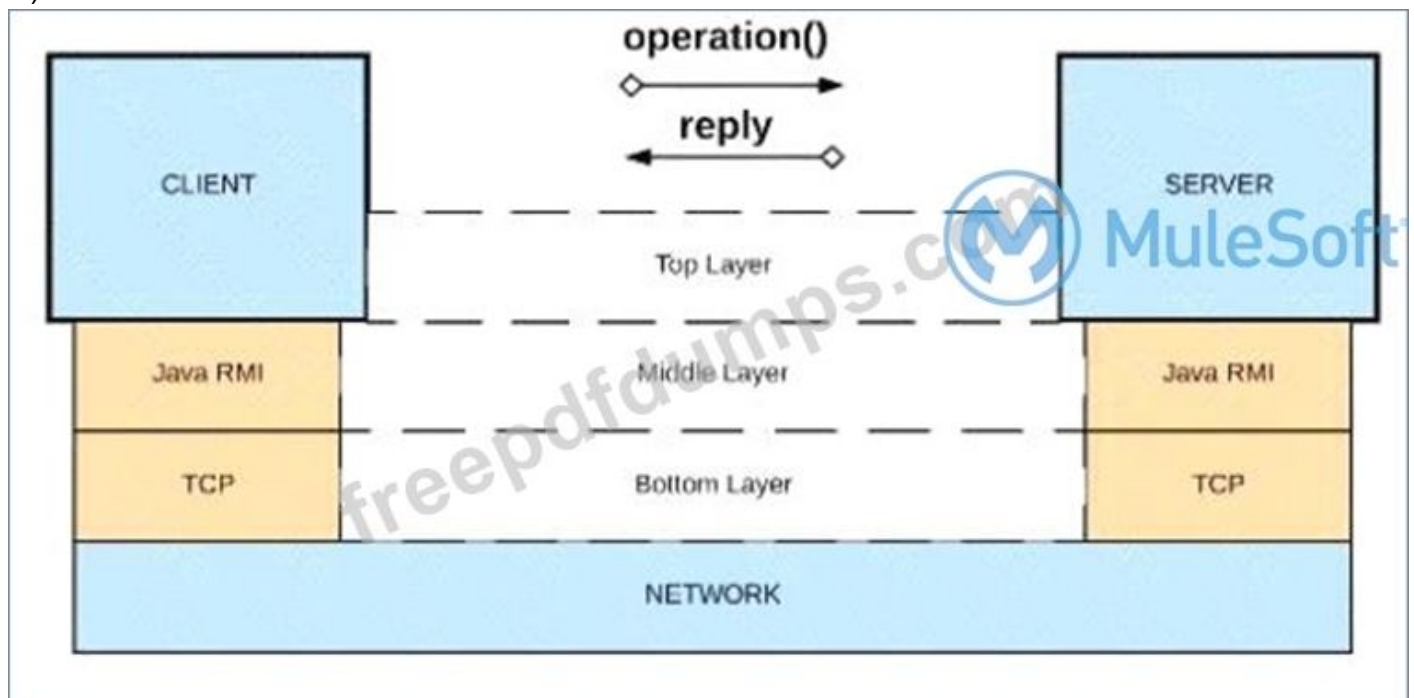
NEW QUESTION: 15

Refer to the exhibit.

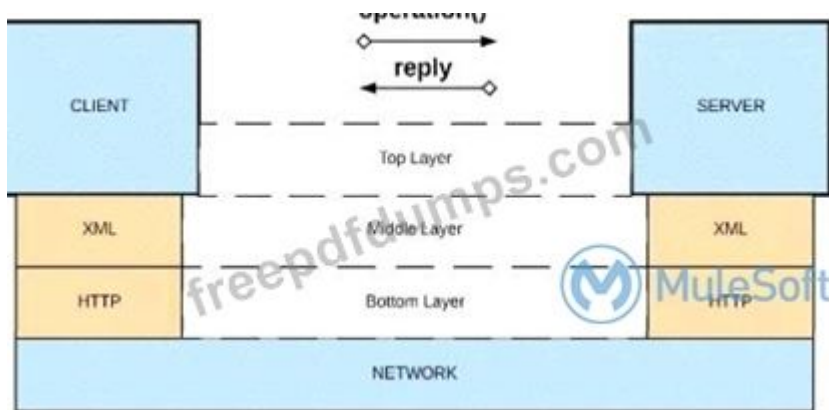


What is a valid API in the sense of API-led connectivity and application networks?

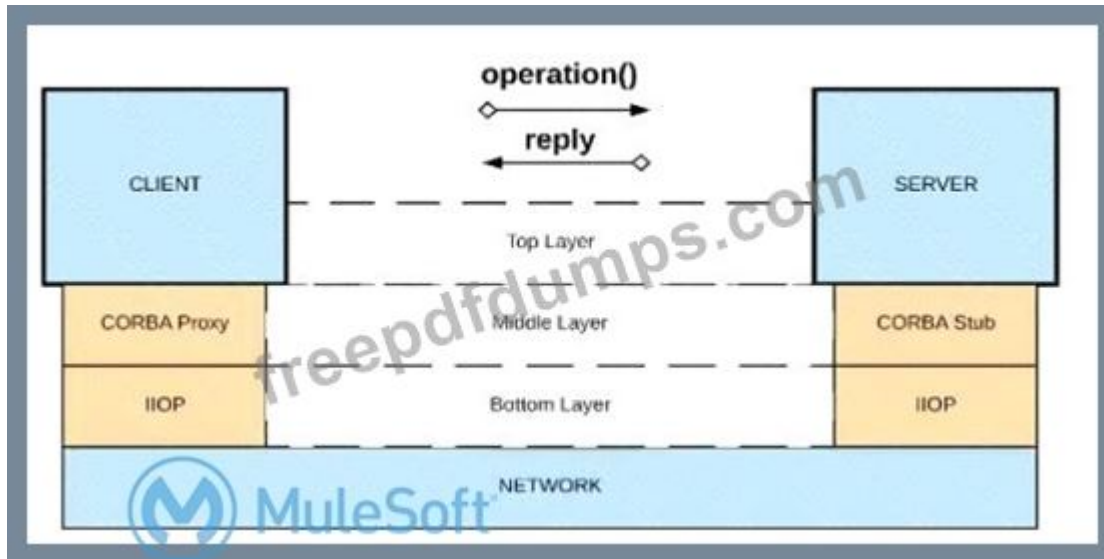
A) Java RMI over TCP



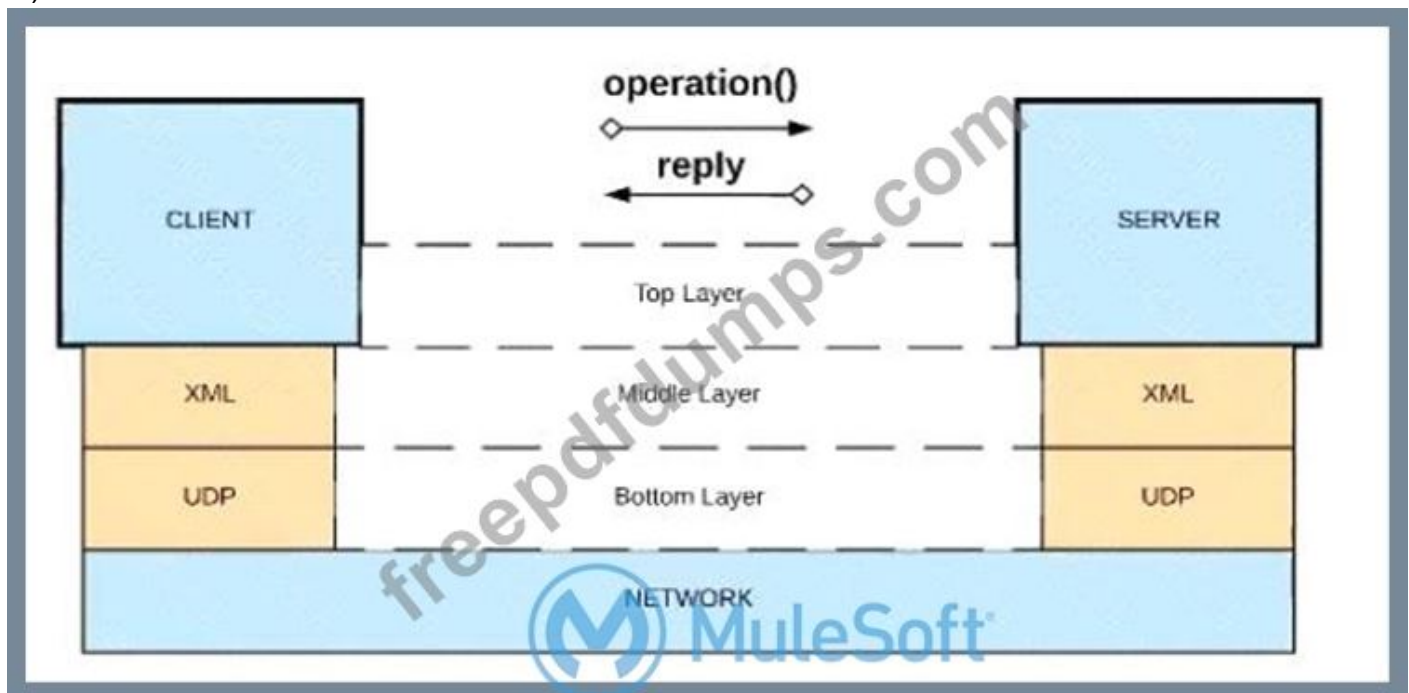
B) Java RMI over TCP



C) CORBA over HOP



D) XML over UDP



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

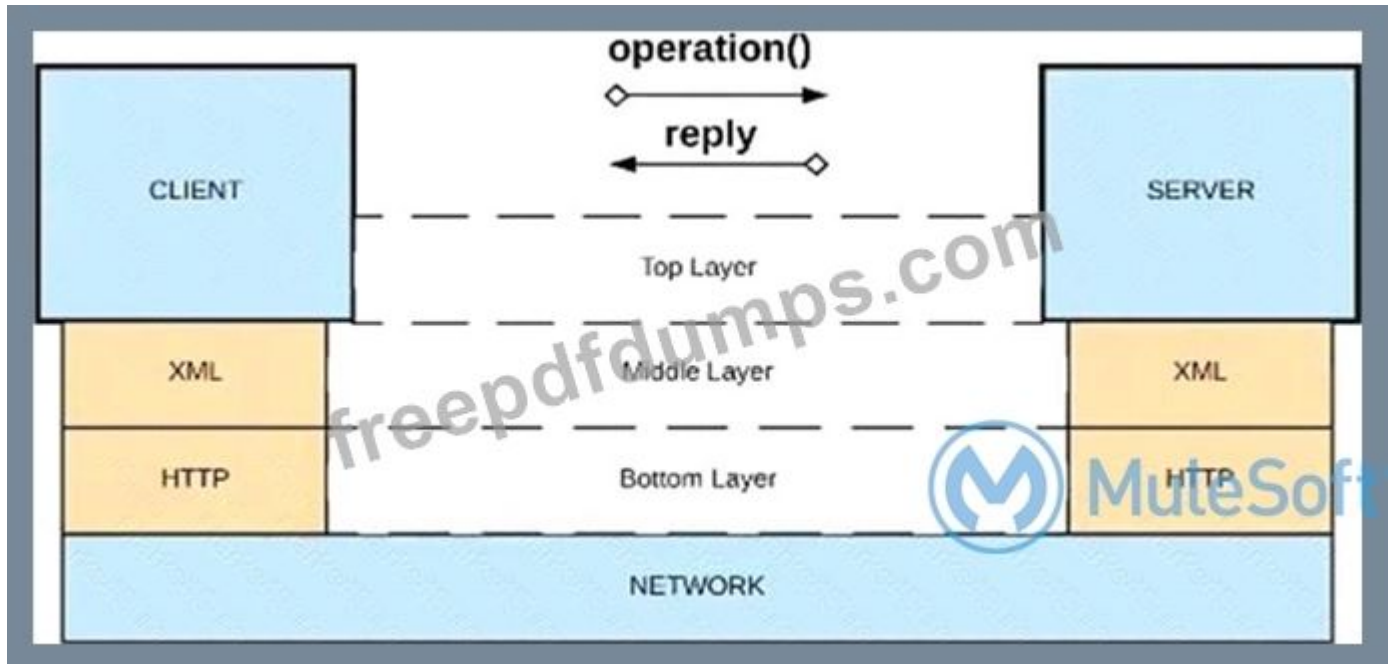
Answer: D ([LEAVE A REPLY](#))

XML over HTTP

>> API-led connectivity and Application Networks urge to have the APIs on HTTP based protocols for building most effective APIs and networks on top of them.

>> The HTTP based APIs allow the platform to apply various varieties of policies to address many NFRs

>> The HTTP based APIs also allow to implement many standard and effective implementation patterns that adhere to HTTP based w3c rules.



Bottom of Form

Top of Form

NEW QUESTION: 16

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

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

Answer: ([SHOW ANSWER](#))

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

>> The main motto of the new IT Operating Model that MuleSoft recommends and made popular is to change the way that they are delivered from a production model to a production + consumption model, which is done through an API strategy called API-led connectivity.

>> The assets built should also be discoverable and self-serveable for reusability across LOBs and organization.

>> MuleSoft's IT operating model does not talk about SDLC model (Agile/ Lean etc) or MDM at all. So, options suggesting these are not valid.

References:

<https://blogs.mulesoft.com/biz/connectivity/what-is-a-center-for-enablement-c4e/>

<https://www.mulesoft.com/resources/api/secret-to-managing-it-projects>

Valid MCPA-Level-1-Maintenance Dumps shared by Actual4test.com for Helping Passing MCPA-Level-1-Maintenance Exam! Actual4test.com now offer the **newest MCPA-Level-1-Maintenance exam dumps**, the Actual4test.com MCPA-Level-1-Maintenance exam **questions have been updated** and **answers have been corrected** get the **newest** Actual4test.com MCPA-Level-1-Maintenance dumps with Test Engine here:
https://www.actual4test.com/MCPA-Level-1-Maintenance_examcollection.html (81 Q&As Dumps, **30%OFF Special Discount: Freepdfdumps**)

NEW QUESTION: 17

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

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

Answer: A (LEAVE A REPLY)

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

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

>> Company requires both data and metadata to be resided within the corporate firewall

>> Company would like to go with customer-hosted infrastructure.

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

Application data can be controlled inside firewall by having Mule Runtimes on customer hosted runtime plane. But if we go with Mulsoft-hosted/ Cloud-based control plane, the control plane required atleast some minimum level of metadata to be sent outside the corporate firewall. As the customer requirement is pretty clear about the data and metadata both to be within the corporate firewall, even though customer wants to move to production as quickly as possible, unfortunately due to the nature of their security requirements, they have no other option but to go with manually provisioned customer-hosted runtime plane and customer-hosted control plane.

NEW QUESTION: 18

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

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

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

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

Answer: B ([LEAVE A REPLY](#))

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

Key details to take from the given scenario:

- >> Upstream API's designed SLA is 500ms (median). Lets ignore maximum SLA response times.
- >> This API calls 3 downstream APIs sequentially and all these are of similar complexity.
- >> The first downstream API is offering median SLA of 100ms, 80th percentile: 500ms; 95th percentile: 1000ms.

Based on the above details:

- >> We can rule out the option which is suggesting to set 50ms timeout. Because, if the median SLA itself being offered is 100ms then most of the calls are going to timeout and time gets wasted in retried them and eventually gets exhausted with all retries. Even if some retries gets successful, the remaining time wont leave enough room for 2nd and 3rd downstream APIs to respond within time.
- >> The option suggesting to NOT set a timeout as the invocation of this API is mandatory and so we must wait until it responds is silly. As not setting time out would go against the good implementation pattern and moreover if the first API is not responding within its offered median

SLA 100ms then most probably it would either respond in 500ms (80th percentile) or 1000ms (95th percentile). In BOTH cases, getting a successful response from 1st downstream API does NO GOOD because already by this time the Upstream API SLA of 500 ms is breached. There is no time left to call 2nd and 3rd downstream APIs.

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

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

NEW QUESTION: 19

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 (LEAVE A REPLY)

SLA-based rate limiting

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

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

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

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

NEW QUESTION: 20

In which layer of API-led connectivity, does the business logic orchestration reside?

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

Answer: (SHOW ANSWER)

Process Layer

>> Experience layer is dedicated for enrichment of end user experience. This layer is to meet the needs of different API clients/ consumers.

>> System layer is dedicated to APIs which are modular in nature and implement/ expose various individual functionalities of backend systems

>> Process layer is the place where simple or complex business orchestration logic is written by invoking one or many System layer modular APIs So, Process Layer is the right answer.

NEW QUESTION: 21

An API client calls one method from an existing API implementation. The API implementation is later updated. What change to the API implementation would require the API client's invocation logic to also be updated?

- A. When the data type of the response is changed for the method called by the API client
- B. When a new method is added to the resource used by the API client
- C. When a new required field is added to the method called by the API client
- D. When a child method is added to the method called by the API client

Answer: C ([LEAVE A REPLY](#))

When a new required field is added to the method called by the API client

>> Generally, the logic on API clients need to be updated when the API contract breaks.

>> When a new method or a child method is added to an API , the API client does not break as it can still continue to use its existing method. So these two options are out.

>> We are left for two more where "datatype of the response if changed" and "a new required field is added".

>> Changing the datatype of the response does break the API contract. However, the question is insisting on the "invocation" logic and not about the response handling logic. The API client can still invoke the API successfully and receive the response but the response will have a different datatype for some field.

>> Adding a new required field will break the API's invocation contract. When adding a new required field, the API contract breaks the RAML or API spec agreement that the API client/API consumer and API provider has between them. So this requires the API client invocation logic to also be updated.

NEW QUESTION: 22

What Anypoint Connectors support transactions?

- A. Database, JMS, VM
- B. Database, 3MS, HTTP
- C. Database, JMS, VM, SFTP
- D. Database, VM, File

Answer: A ([LEAVE A REPLY](#))

NEW QUESTION: 23

A Mule application exposes an HTTPS endpoint and is deployed to three CloudHub workers that do not use static IP addresses. The Mule application expects a high volume of client requests in

short time periods. What is the most cost-effective infrastructure component that should be used to serve the high volume of client requests?

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

Answer: B (LEAVE A REPLY)

The CloudHub shared load balancer

The scenario in this question can be split as below:

>> There are 3 CloudHub workers (So, there are already good number of workers to handle high volume of requests)

>> The workers are not using static IP addresses (So, one CANNOT use customer load-balancing solutions without static IPs)

>> Looking for most cost-effective component to load balance the client requests among the workers.

Based on the above details given in the scenario:

>> Runtime autoscaling is NOT at all cost-effective as it incurs extra cost. Most over, there are already 3 workers running which is a good number.

>> We cannot go for a customer-hosted load balancer as it is also NOT most cost-effective (needs custom load balancer to maintain and licensing) and same time the Mule App is not having Static IP Addresses which limits from going with custom load balancing.

>> An API Proxy is irrelevant there as it has no role to play w.r.t handling high volumes or load balancing.

So, the only right option to go with and fits the purpose of scenario being most cost-effective is - using a CloudHub Shared Load Balancer.

NEW QUESTION: 24

Which of the following sequence is correct?

- A. API Client implements logic to call an API >> API Consumer requests access to API >> API Implementation routes the request to >> API
- B. API Consumer requests access to API >> API Client implements logic to call an API >> API routes the request to >> API Implementation
- C. API Consumer implements logic to call an API >> API Client requests access to API >> API Implementation routes the request to >> API
- D. API Client implements logic to call an API >> API Consumer requests access to API >> API routes the request to >> API Implementation

Answer: B (LEAVE A REPLY)

API Consumer requests access to API >> API Client implements logic to call an API >> API routes the request to >> API Implementation

>> API consumer does not implement any logic to invoke APIs. It is just a role. So, the option stating "API Consumer implements logic to call an API" is INVALID.

>> API Implementation does not route any requests. It is a final piece of logic where functionality of target systems is exposed. So, the requests should be routed to the API implementation by some other entity. So, the options stating "API Implementation routes the request to >> API" is INVALID

>> The statements in one of the options are correct but sequence is wrong. The sequence is given as "API Client implements logic to call an API >> API Consumer requests access to API >> API routes the request to

>> API Implementation". Here, the statements in the options are VALID but sequence is WRONG.

>> Right option and sequence is the one where API consumer first requests access to API on Anypoint Exchange and obtains client credentials. API client then writes logic to call an API by using the access client credentials requested by API consumer and the requests will be routed to API implementation via the API which is managed by API Manager.

NEW QUESTION: 25

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: ([SHOW 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: 26

A system API is deployed to a primary environment as well as to a disaster recovery (DR) environment, with different DNS names in each environment. A process API is a client to the system API and is being rate limited by the system API, with different limits in each of the environments. The system API's DR environment provides only 20% of the rate limiting offered by the primary environment. What is the best API fault-tolerant invocation strategy to reduce overall errors in the process API, given these conditions and constraints?

- A.** Invoke the system API deployed to the primary environment; add timeout and retry logic to the process API to avoid intermittent failures; if it still fails, invoke the system API deployed to the DR environment
- B.** Invoke the system API deployed to the primary environment; add retry logic to the process API to handle intermittent failures by invoking the system API deployed to the DR environment
- C.** In parallel, invoke the system API deployed to the primary environment and the system API deployed to the DR environment; add timeout and retry logic to the process API to avoid intermittent failures; add logic to the process API to combine the results
- D.** Invoke the system API deployed to the primary environment; add timeout and retry logic to the process API to avoid intermittent failures; if it still fails, invoke a copy of the process API deployed to the DR environment

Answer: A ([LEAVE A REPLY](#))

Invoke the system API deployed to the primary environment; add timeout and retry logic to the process API to avoid intermittent failures; if it still fails, invoke the system API deployed to the DR environment

There is one important consideration to be noted in the question which is - System API in DR environment provides only 20% of the rate limiting offered by the primary environment. So, comparatively, very less calls will be allowed into the DR environment API opposed to its primary environment. With this in mind, let's analyse what is the right and best fault-tolerant invocation strategy.

1. Invoking both the system APIs in parallel is definitely NOT a feasible approach because of the 20% limitation we have on DR environment. Calling in parallel every time would easily and quickly exhaust the rate limits on DR environment and may not give chance to genuine intermittent error scenarios to let in during the time of need.

2. Another option given is suggesting to add timeout and retry logic to process API while invoking primary environment's system API. This is good so far. However, when all retries failed, the option is suggesting to invoke the copy of process API on DR environment which is not right or recommended. Only system API is the one to be considered for fallback and not the whole process API. Process APIs usually have lot of heavy orchestration calling many other APIs which we do not want to repeat again by calling DR's process API. So this option is NOT right.

3. One more option given is suggesting to add the retry (no timeout) logic to process API to directly retry on DR environment's system API instead of retrying the primary environment system API first. This is not at all a proper fallback. A proper fallback should occur only after all retries are performed and exhausted on Primary environment first. But here, the option is suggesting to directly retry fallback API on first failure itself without trying main API. So, this option is NOT right too.

This leaves us one option which is right and best fit.

- Invoke the system API deployed to the primary environment
- Add Timeout and Retry logic on it in process API
- If it fails even after all retries, then invoke the system API deployed to the DR environment.

NEW QUESTION: 27

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
- D.** Each modern API must be REST and HTTP based

Answer: B ([LEAVE A REPLY](#))

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: 28

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: (SHOW 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: 29

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: (SHOW 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: 30

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 (LEAVE A REPLY)

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-Ano>

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

NEW QUESTION: 31

What is a best practice when building System APIs?

- A. Document the API using an easily consumable asset like a RAML definition
- B. Model all API resources and methods to closely mimic the operations of the backend system
- C. Build an Enterprise Data Model (Canonical Data Model) for each backend system and apply it to System APIs
- D. Expose to API clients all technical details of the API implementation's interaction with the backend system

Answer: B (LEAVE A REPLY)

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

>> There are NO fixed and straight best practices while opting data models for APIs. They are completely contextual and depends on number of factors. Based upon those factors, an enterprise can choose if they have to go with Enterprise Canonical Data Model or Bounded Context Model etc.

>> One should NEVER expose the technical details of API implementation to their API clients. Only the API interface/ RAML is exposed to API clients.

>> It is true that the RAML definitions of APIs should be as detailed as possible and should reflect most of the documentation. However, just that is NOT enough to call your API as best documented API. There should be even more documentation on Anypoint Exchange with API Notebooks etc. to make and create a developer friendly API and repository..

>> The best practice always when creating System APIs is to create their API interfaces by modeling their resources and methods to closely reflect the operations and functionalities of that backend system.

Valid MCPA-Level-1-Maintenance Dumps shared by Actual4test.com for Helping Passing MCPA-Level-1-Maintenance Exam! Actual4test.com now offer the **newest MCPA-Level-1-Maintenance exam dumps**, the Actual4test.com MCPA-Level-1-Maintenance exam **questions have been updated** and **answers have been corrected** get the **newest** Actual4test.com MCPA-Level-1-Maintenance dumps with Test Engine here:

NEW QUESTION: 32

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: (SHOW 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: 33

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: (SHOW 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.

Valid MCPA-Level-1-Maintenance Dumps shared by Actual4test.com for Helping Passing MCPA-Level-1-Maintenance Exam! Actual4test.com now offer the **newest MCPA-Level-1-Maintenance exam dumps**, the Actual4test.com MCPA-Level-1-Maintenance exam **questions have been updated** and **answers have been corrected** get the **newest** Actual4test.com MCPA-Level-1-Maintenance dumps with Test Engine here:
https://www.actual4test.com/MCPA-Level-1-Maintenance_examcollection.html (81 Q&As Dumps, **30%OFF** Special Discount: **Freepdfdumps**)