VCE & PDF
GeekCert.com

MCPA-LEVEL-1-MAINTENANCE^{Q&As}

MuleSoft Certified Platform Architect - Level 1 MAINTENANCE

Pass Mulesoft MCPA-LEVEL-1-MAINTENANCE Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.geekcert.com/mcpa-level-1-maintenance.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Mulesoft Official Exam Center

⚙ **Instant Download** After Purchase

⚙ **100% Money Back** Guarantee

⚙ **365 Days** Free Update

⚙ **800,000+** Satisfied Customers

QUESTION 1

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

Correct Answer: C

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

---

QUESTION 2

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

Correct Answer: 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

****

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, comparitively, very less calls will be allowed into the

[MCPA-LEVEL-1-MAINTENANCE PDF Dumps](#) | [MCPA-LEVEL-1-MAINTENANCE Exam Questions](#) | [MCPA-LEVEL-1-MAINTENANCE Braindumps](#)

2 / 8

DR

environment API opposed to its primary environment. With this in mind, lets 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.

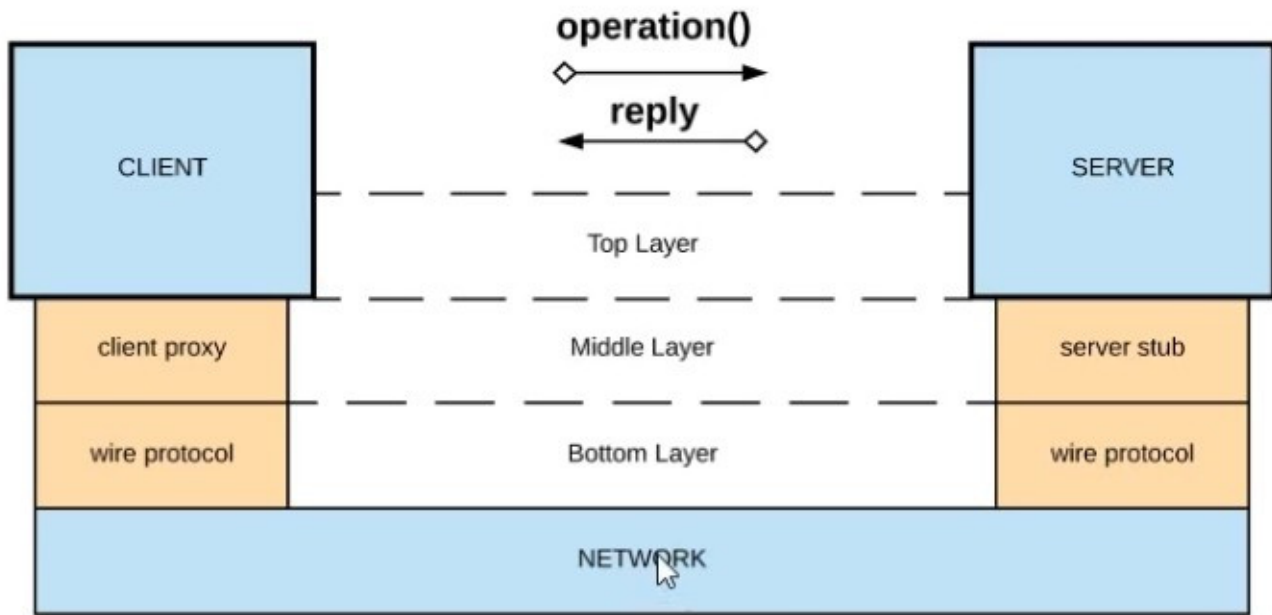-Invoke the system API deployed to the primary environment

-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.

QUESTION 3

Refer to the exhibit.

[MCPA-LEVEL-1-MAINTENANCE PDF Dumps](#) | [MCPA-LEVEL-1-MAINTENANCE Exam Questions](#) | [MCPA-LEVEL-1-MAINTENANCE Braindumps](#)

3 / 8

operation()

reply

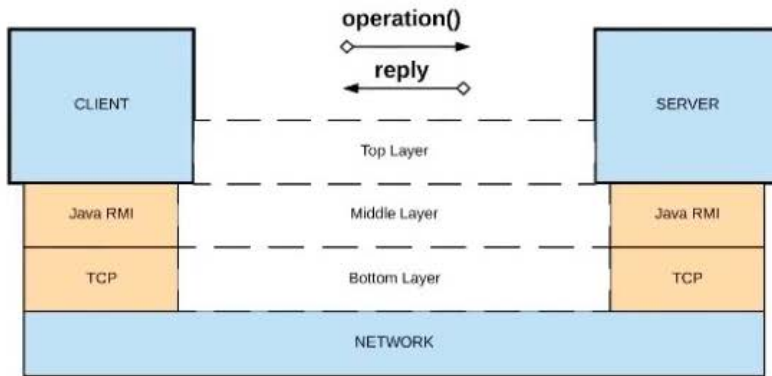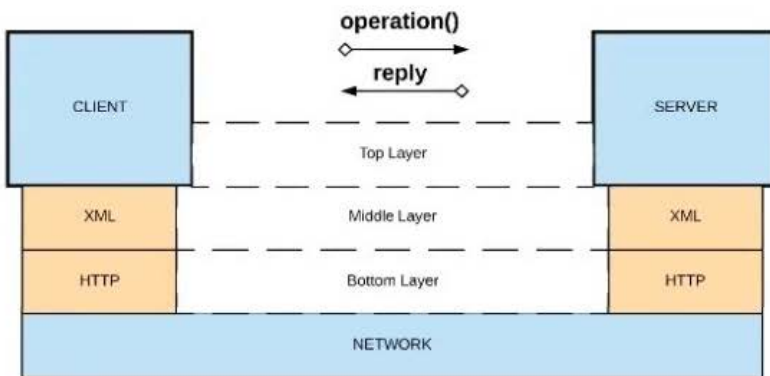| CLIENT			SERVER
		Top Layer	
client proxy		Middle Layer	server stub
wire protocol		Bottom Layer	wire protocol
		NETWORK	

Generic RPC Architecture

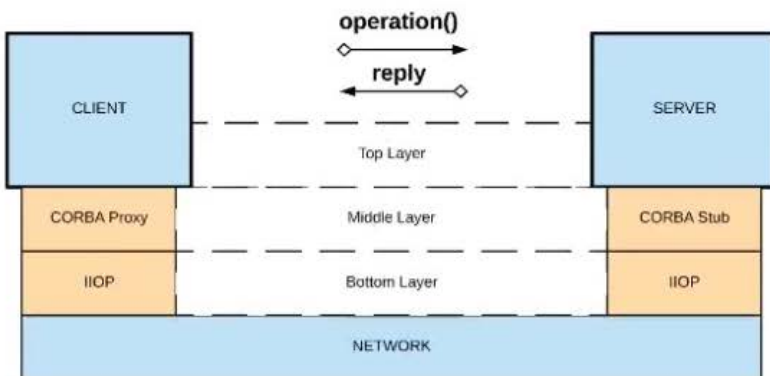What is a valid API in the sense of API-led connectivity and application networks?

A. Java RMI over TCP

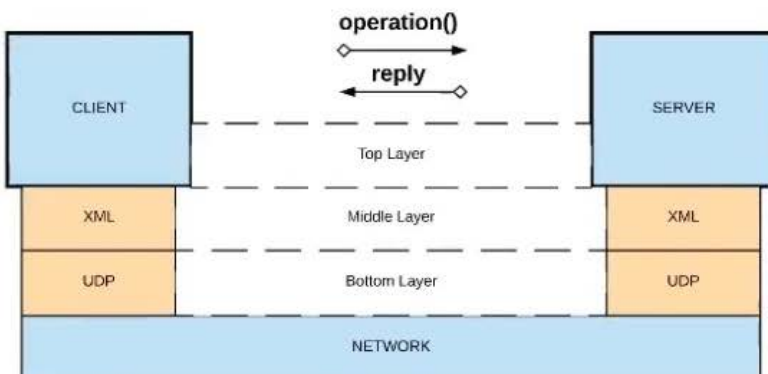B. Java RMI over TCP

C. CORBA over HOP

D. XML over UDP

A. Option A

B. Option B

C. Option C

D. Option D

Correct Answer: D

XML over HTTP *** >> API-led connectivity and Application Networks urge to have the APIs on HTTP based protocols for building most effective APIs and networks on top of them. >> The HTTP based APIs allow the platform to apply various varities of policies to address many NFRs >> The HTTP based APIs also allow to implement many standard and effective implementation patterns that adhere to HTTP based w3c rules.
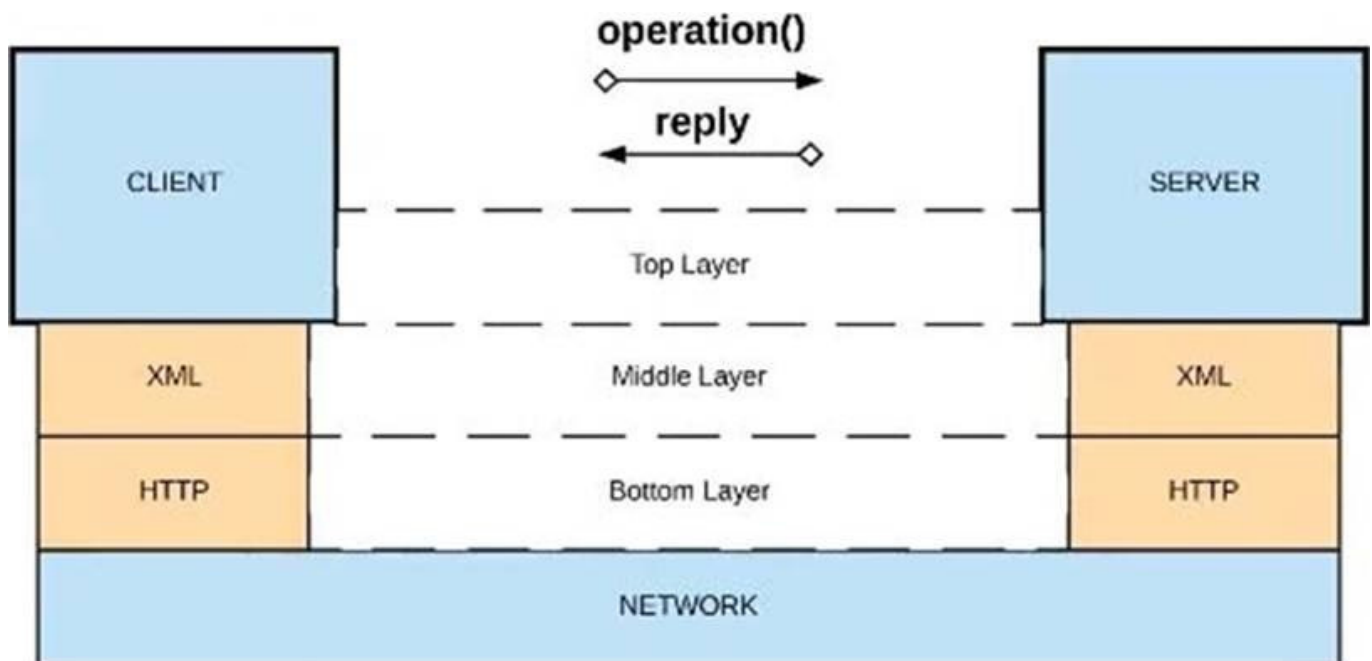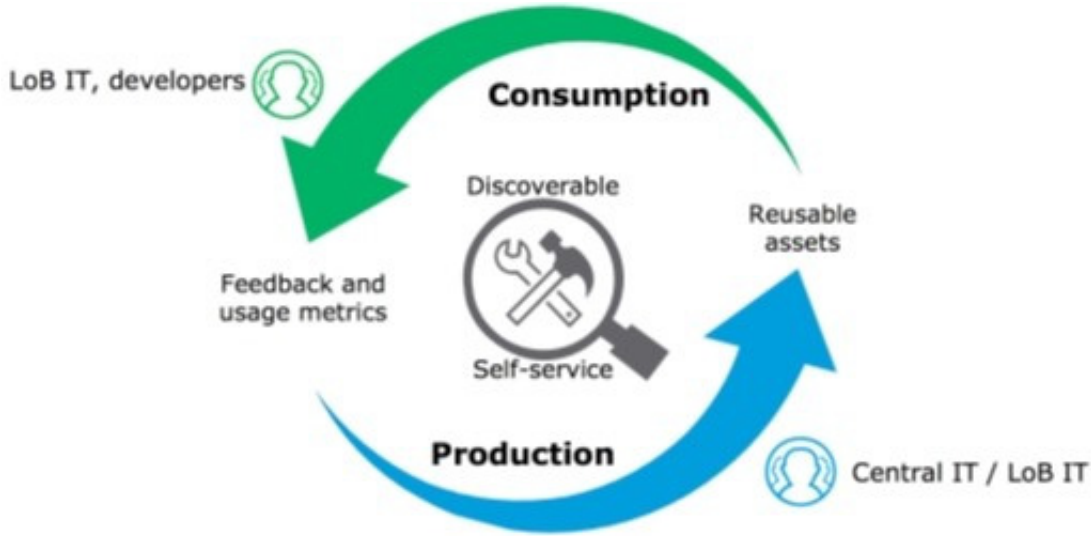

QUESTION 4

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 resusable assets, make them discoverable so that LOB teams can self-serve and browse the APIs

Correct Answer: C

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

QUESTION 5

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

Correct Answer: D

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

MCPA-
LEVEL-1-MAINTENANCE
PDF Dumps

MCPA-
LEVEL-1-MAINTENANCE
Exam Questions

MCPA-
LEVEL-1-MAINTENANCE
Braindumps