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

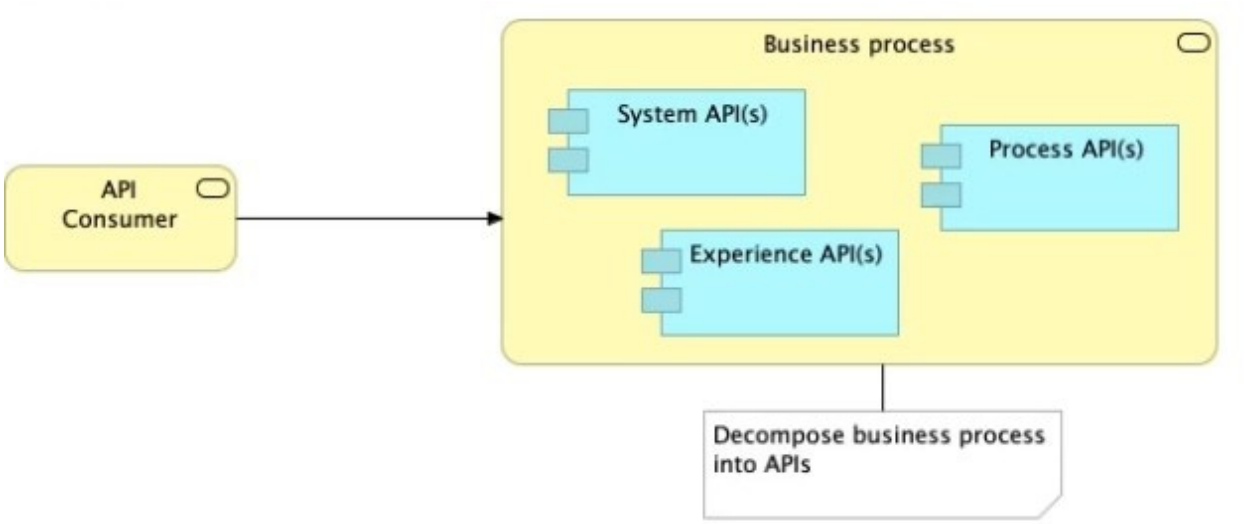
100% Passing Guarantee
100% Money Back Assurance

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

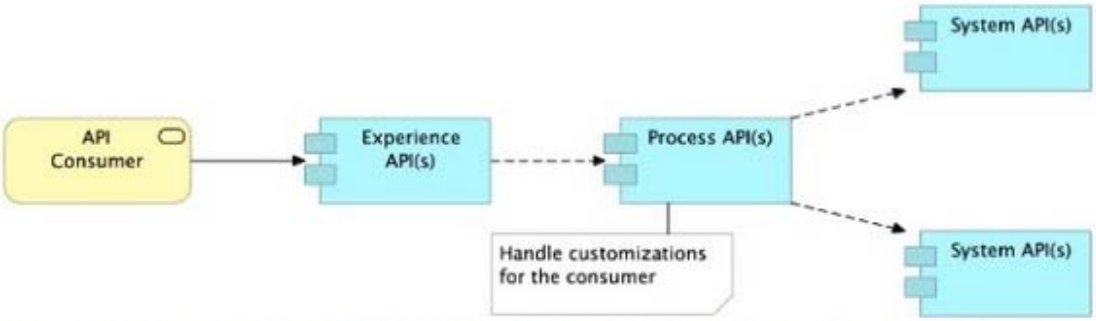
QUESTION 1

Refer to the exhibit.

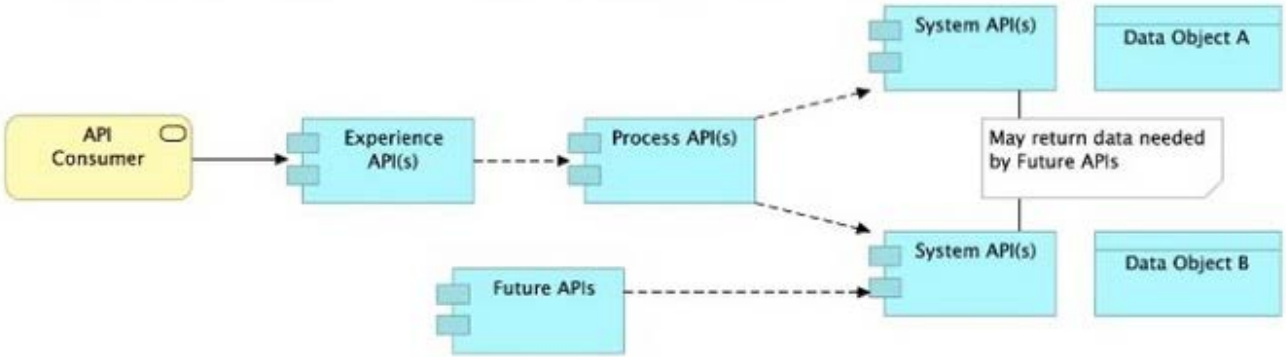


What is the best way to decompose one end-to-end business process into a collaboration of Experience, Process, and System APIs?

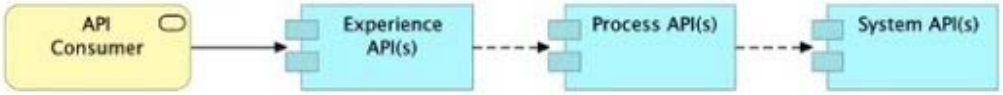
A. Handle customizations for the end-user application at the Process API level rather than the Experience API level



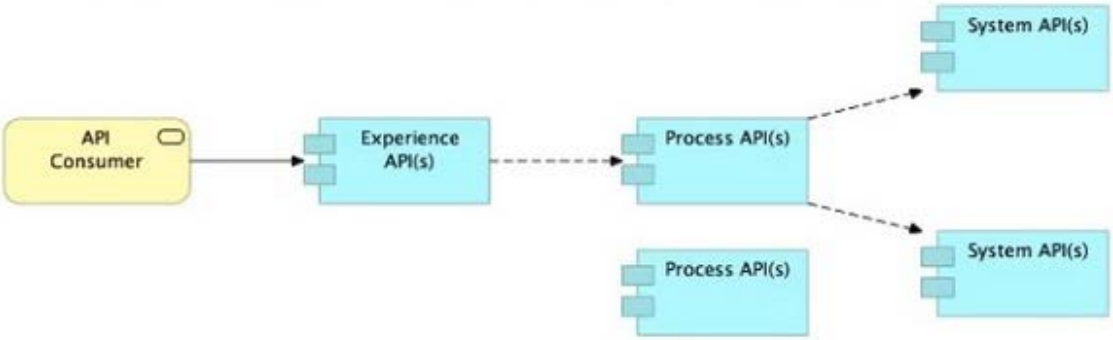
B. Allow System APIs to return data that is NOT currently required by the identified Process or Experience APIs



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

Correct Answer: B

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

****

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

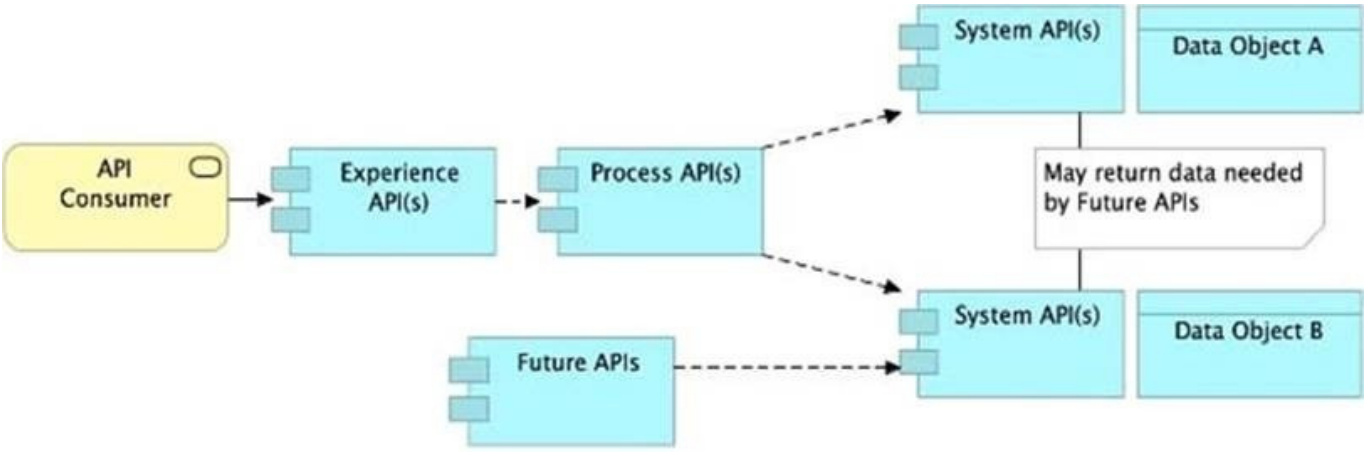
>> We should use tiered approach but NOT always by creating exactly one API for each of the 3 layers. Experience APIs might be one but Process APIs and System APIs are often more than one. System APIs for sure will be more than one

all the time as they are the smallest modular APIs built in front of end systems. >> Process APIs can call System APIs as well as other Process APIs. There is no such anti-design pattern in API-Led connectivity saying Process APIs should

not call other Process APIs.

So, the right answer in the given set of options that makes sense as per API-Led connectivity principles is to allow System APIs to return data that is NOT currently required by the identified Process or Experience APIs. This way, some future

Process APIs can make use of that data from System APIs and we need NOT touch the System layer APIs again and again.



QUESTION 2

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

Correct Answer: B

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.

QUESTION 3

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

A. The assignment of each HTTP request to a particular CloudHub worker

B. The logging configuration that enables log entries to be visible in Runtime Manager

C. The SSL certificates used by the API implementation to expose HTTPS endpoints

D. The number of DNS entries allocated to the API implementation

Correct Answer: C

The SSL certificates used by the API implementation to expose HTTPS endpoints

****

>> The assignment of each HTTP request to a particular CloudHub worker is taken care by Anypoint Platform itself. We need not manage it explicitly in the API implementation and in fact we CANNOT manage it in the API implementation. >>

The logging configuration that enables log entries to be visible in Runtime Manager is ALWAYS managed in the API implementation and NOT just for SLB. So this is not something we do EXCLUSIVELY when using SLB.

>> We DO NOT manage the number of DNS entries allocated to the API implementation inside the code. Anypoint Platform takes care of this.

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

QUESTION 4

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

Correct Answer: C

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.

QUESTION 5

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 API\\'s 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 update should be identified as a project risk and full regression testing of the functionality that uses this API should be run

B. The API producer should be contacted to understand the change to existing functionality

C. The API producer should be requested to run the old version in parallel with the new one

D. The API client code ONLY needs to be changed if it needs to take advantage of new features

Correct Answer: D

Reference: https://docs.mulesoft.com/exchange/to-change-raml-version