Free Salesforce Certified MuleSoft Platform Architect (Mule-Arch-201) Practice Questions

Shared by Avila on 13-08-2026

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

A developer from the Central IT team has created an initial version of the RAML definition in Design Center for an OAuth 2.0-protected System API and published it

to Exchange. Another developer from LoB IT discovered the System API in Exchange and would like to leverage it in the Process API.

What is the MuleSoft-recommended approach for Process API to invoke the System API?

Options:

A- The Process API needs to import an CAuth 2.0 module from Exchange first and update it with OAuth 2.0 credentials before the System API can be invoked
B- The Process API uses property YAML files to store the System API URLs and uses the HTTP Request Connector to invoke the Systerm API
C- The Process APL uses the REST Connect Connector autogenerated in Exchange for the System API
D- The Process API manually updates the Process API POM file to include the System API as a dependency

Answer:

C

Explanation:

In MuleSoft's ecosystem, when a Process API needs to consume a System API (published to Exchange and protected by OAuth 2.0), the recommended approach is to utilize the REST Connect Connector. Here's how it aligns with best practices:

Automated Connector Generation:

When a RAML or OAS specification is published in Exchange, MuleSoft automatically generates a REST Connect Connector for that API. This connector simplifies integration as it abstracts the complexity of making HTTP requests and handling OAuth authentication.

Streamlined Integration:

The Process API can import this generated connector from Exchange and configure OAuth credentials, streamlining secure access to the System API without manual HTTP setup.

Why Option C is Correct:

Using the REST Connect Connector directly leverages MuleSoft's automated tooling, minimizing

manual configurations and ensuring a more maintainable integration.

of Incorrect Options:

Option A (importing an OAuth module) is unnecessary; OAuth is handled within the connector's configuration.

Option B (property YAML files with HTTP requests) involves manual setup, which is more error-prone and not recommended.

Option D (manually updating POM file) does not directly aid in invoking an API through Exchange.

Reference For more information on using REST Connect Connectors and OAuth integration in MuleSoft, refer to the MuleSoft documentation on API Management and Connectors.

Question 2

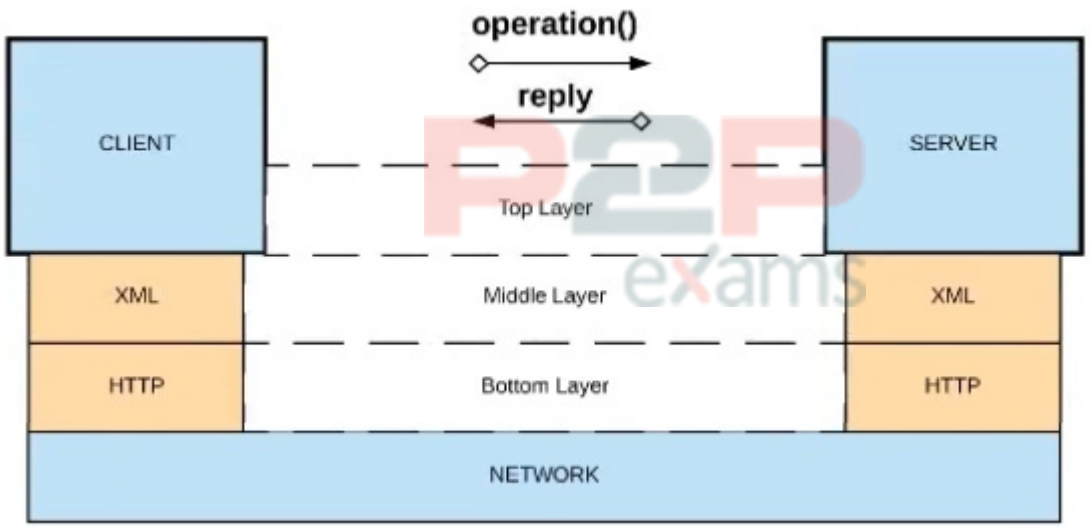
Question Type: MultipleChoice

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

Options:

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

Answer:

D

Explanation:

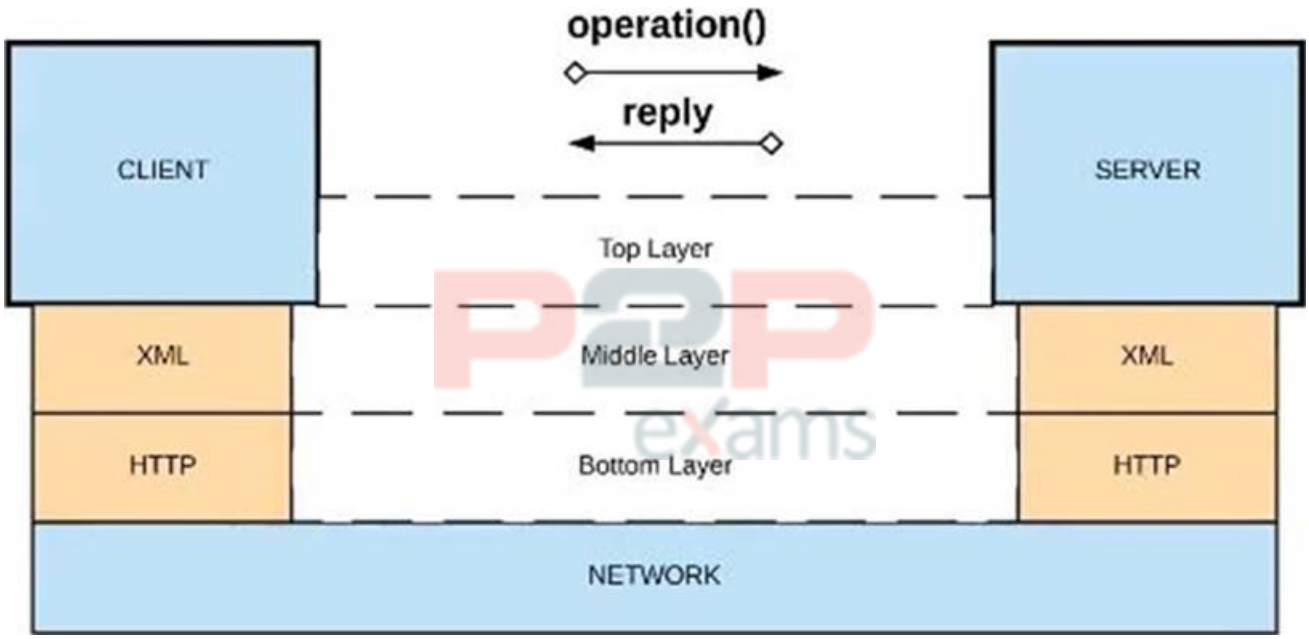
Correct Answe r:XML over HTTP

****

>>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.



Bottom of Form

Top of Form

Question 3

Question Type: MultipleChoice

A circuit breaker strategy is planned in order to meet the goal of improved response time and demand on a downstream API.

* Circuit Open: More than 10 errors per minute for three minutes

* Circuit Half-Open: One error per minute

* Circuit Closed: Less than one error per minute for five minutes

Out of several proposals from the engineering team, which option will meet this goal?

Options:

A- Create a custom policy that implements the circuit breaker and includes policy template expressions for the required settings
B- Create Anypoint Monitoring alerts for Circuit Open/Closed configurations, and then implement a retry strategy for Circuit Half-Open configuration
C- Add the Circuit Breaker policy to the API instance, and configure the required settings
D- Implement the strategy in a Mule application, and provide the settings in the YAML configuration

Answer:

C

Explanation:

Understanding Circuit Breaker Policy:

A circuit breaker is a design pattern used to detect failures and prevent an application from continually trying to execute a failing operation. In this case, it will help improve response time and reduce demand on the downstream API.

The specified configuration includes conditions for opening, half-opening, and closing the circuit based on error rates over time:

Circuit Open: Triggered if there are more than 10 errors per minute for three consecutive minutes.

Circuit Half-Open: The circuit transitions to half-open if there is one error per minute.

Circuit Closed: The circuit closes if the error rate is less than one error per minute for five minutes.

Evaluating the Options:

Option A: Creating a custom policy with template expressions could work, but it would require custom development. Since the Anypoint Platform already has a Circuit Breaker policy available, this would be a less efficient and more complex solution.

Option B: Anypoint Monitoring alerts can be used for monitoring the API, but they do not provide circuit-breaking functionality. Additionally, implementing a retry strategy for the half-open state is not sufficient to achieve the required circuit breaker behavior.

Option C (Correct Answer): Adding the Circuit Breaker policy to the API instance on Anypoint Platform allows you to set up circuit-breaking conditions directly. This approach uses the built-in Circuit Breaker policy, where you can configure parameters such as error thresholds and time intervals to match the requirements. This solution is efficient, reliable, and leverages Anypoint's out-of-the-box capabilities.

Option D: Implementing the strategy within a Mule application with a YAML configuration could be complex and less manageable. Additionally, it does not leverage Anypoint Platform's built-in Circuit Breaker policy, which is more suited to this scenario.

Conclusion:

Option C is the correct choice, as it leverages Anypoint Platform's Circuit Breaker policy. This solution allows for configuring thresholds and time intervals as specified, improving response time and reducing demand on the downstream API while utilizing Anypoint's managed policy feature.

Refer to MuleSoft's documentation on implementing the Circuit Breaker policy in API Manager for detailed configuration guidance.

Question 4

Question Type: MultipleChoice

What is true about API implementations when dealing with legal regulations that require all data processing to be performed within a certain jurisdiction (such as in the USA or the EU)?

Options:

A- They must avoid using the Object Store as it depends on services deployed ONLY to the US East region
B- They must use a Jurisdiction-local external messaging system such as Active MQ rather than Anypoint MQ
C- They must te deployed to Anypoint Platform runtime planes that are managed by Anypoint Platform control planes, with both planes in the same Jurisdiction
D- They must ensure ALL data is encrypted both in transit and at rest

Answer:

C

Explanation:

Correct Answe r: They must be deployed to Anypoint Platform runtime planes that are managed by Anypoint Platform control planes, with both planes in the same Jurisdiction.

****

>> As per legal regulations, all data processing to be performed within a certain jurisdiction. Meaning, the data in USA should reside within USA and should not go out. Same way, the data in EU should reside within EU and should not go out.

>> So, just encrypting the data in transit and at rest does not help to be compliant with the rules. We need to make sure that data does not go out too.

>> The data that we are talking here is not just about the messages that are published to Anypoint MQ. It includes the apps running, transaction states, application logs, events, metric info and any other metadata. So, just replacing Anypoint MQ with a locally hosted ActiveMQ does NOT help.

>> The data that we are talking here is not just about the key/value pairs that are stored in Object Store. It includes the messages published, apps running, transaction states, application logs, events, metric info and any other metadata. So, just avoiding using Object Store does NOT help.

>> The only option left and also the right option in the given choices is to deploy application on runtime and control planes that are both within the jurisdiction.

Question 5

Question Type: MultipleChoice

Refer to the exhibit.

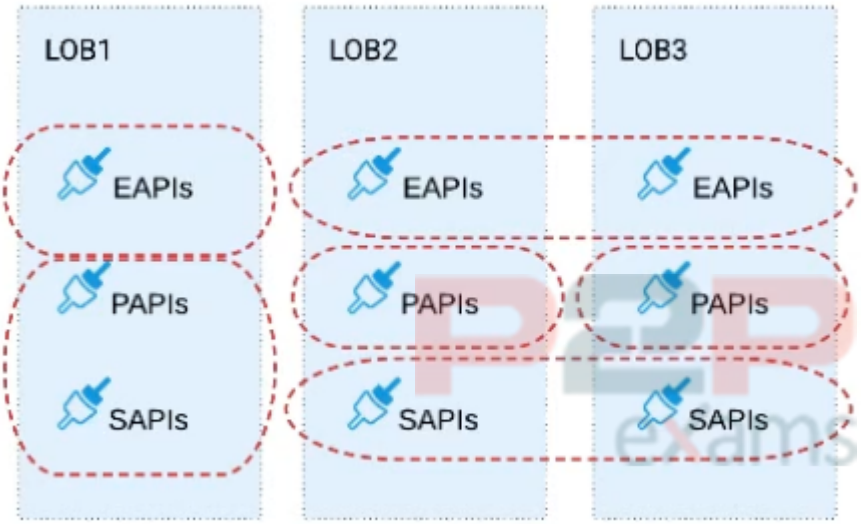
Three business processes need to be implemented, and the implementations need to communicate with several different SaaS applications.

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

Options:

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

Answer:

A

Explanation:

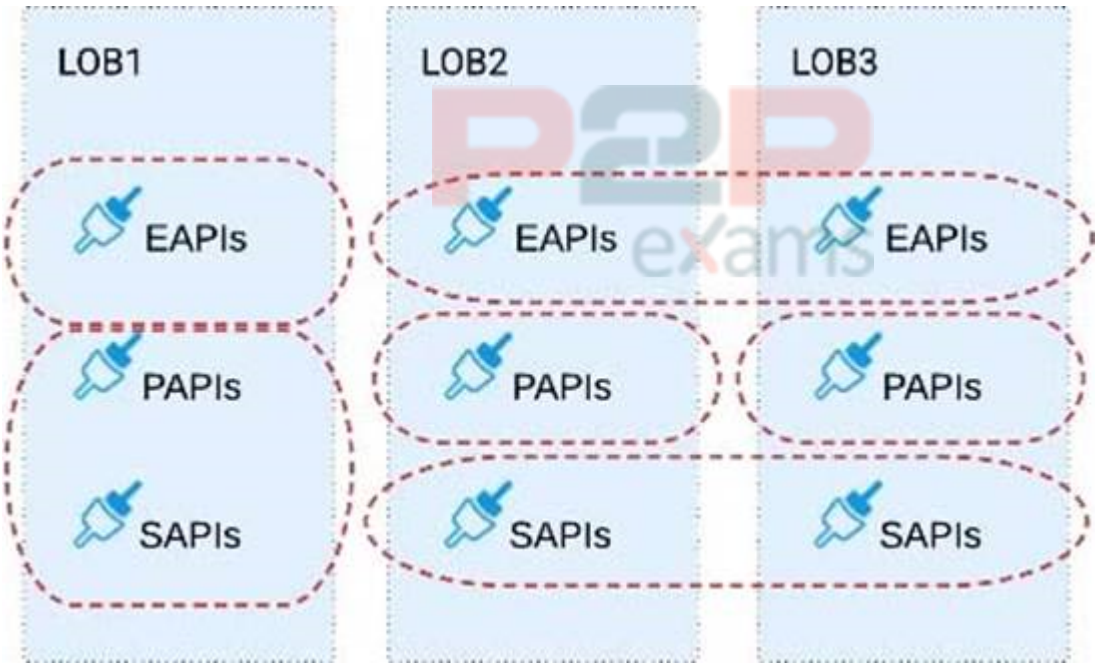
Correct Answe r:Build several Bounded Context Data Models that align with coherent parts of the business processes and the definitions of associated business entities.

****

>>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.



Question 6

Question Type: MultipleChoice

A large lending company has developed an API to unlock data from a database server and web server. The API has been deployed to Anypoint Virtual Private Cloud

(VPC) on CloudHub 1.0.

The database server and web server are in the customer's secure network and are not accessible through the public internet. The database server is in the customer's AWS

VPC, whereas the web server is in the customer's on-premises corporate data center.

How can access be enabled for the API to connect with the database server and the web server?

Options:

A- Set up VPC peering with AWS VPC and a VPN tunnel to the customer's on-premises corporate data center
B- Set up VPC peering with AWS VPC and the customer's on-premises corporate data center
C- Setup a transit gateway to the customer's on-premises corporate data center through AWS VPC
D- Set up VPC peering with the customer's on-premises corporate data center and a VPN tunnel to AWS VPC

Answer:

A

Explanation:

Scenario Overview:

The API resides in Anypoint Virtual Private Cloud (VPC) on CloudHub 1.0, where it requires connectivity to both an AWS-hosted database server and an on-premises web server.

Both servers are isolated from the public internet: the database server is within the customer's AWS VPC, and the web server is within the customer's on-premises corporate data center.

Connectivity Requirements:

To connect to the AWS database server from the API in Anypoint VPC, VPC peering is ideal. This would allow a private network connection between the MuleSoft Anypoint VPC and the customer's AWS VPC, enabling secure, direct access to the database.

To connect to the on-premises web server, a VPN tunnel is suitable. This would establish a secure, encrypted connection from the Anypoint VPC to the customer's corporate data center, allowing secure data flow between the API and the on-premises web server.

Analysis of Options:

Option A (Correct Answer): Setting up VPC peering with AWS VPC enables private network connectivity with the database server, while a VPN tunnel to the on-premises data center allows secure access to the web server. This combination meets the requirements for secure, controlled access to both resources.

Option B: VPC peering alone would not suffice because it does not support a connection from the Anypoint VPC directly to an on-premises network. A VPN is necessary for on-premises access.

Option C: Setting up a transit gateway would provide connectivity within AWS but would not enable direct connectivity from CloudHub to the on-premises network.

Option D: VPC peering with the on-premises network is not possible because VPC peering is typically used to connect two VPCs, not a VPC with an on-premises network.

Conclusion:

Option A is the correct choice, as it provides a complete solution by using VPC peering for AWS VPC connectivity and a VPN tunnel for secure on-premises connectivity. This setup aligns with Anypoint Platform best practices for connecting Anypoint VPCs to both AWS-hosted and on-premises systems, ensuring secure, controlled access to both the database and web server.

For more detailed reference, MuleSoft documentation on Anypoint VPC peering and VPN connectivity provides additional context on best practices for setting up these connections within a hybrid network infrastructure.

Question 7

Question Type: MultipleChoice

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?

Options:

A- 443
B- 8081
C- 8091
D- 8082

Answer:

D

Explanation:

Correct Answe r: 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.

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-Another-Cloudhub-Application

https://help.mulesoft.com/s/question/0D52T00004mXXULSA4/multiple-http-listerners-on-cloudhub-one-with-port-9090

P2P exams

P2P exams

To Get Premium Files for MuleSoft-Platform-Architect-I Visit

https://www.p2pexams.com/products/mulesoft-platform-architect-i

For More Free Questions Visit

https://www.p2pexams.com/salesforce/pdf/mulesoft-platform-architect-i

