MuleSoft Integration Architect I

1. Mule applications need to be deployed to CloudHub so they can access on-premises database systems. These systems store sensitive and hence tightly protected data, so are not accessible over the internet. What network architecture supports this requirement?

A. An Anypoint VPC connected to the on-premises network using an IPsec tunnel or AWS DirectConnect, plus matching firewall rules in the VPC and on-premises network

B. Static IP addresses for the Mule applications deployed to the CloudHub Shared Worker Cloud, plus matching firewall rules and IPwhitelisting in the on-premises network

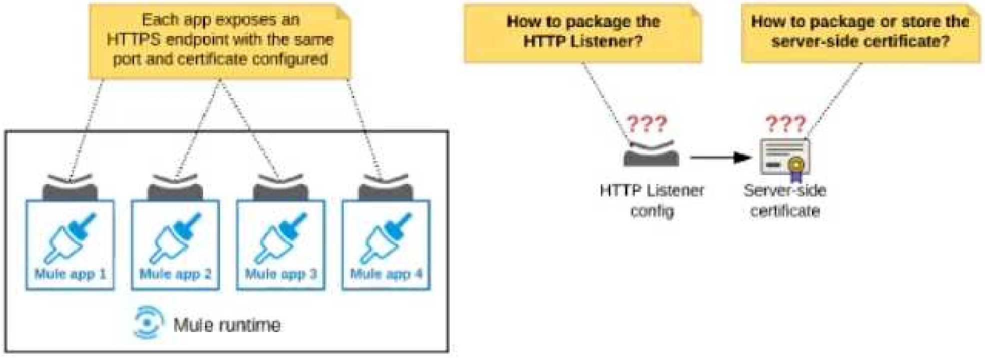
C. An Anypoint VPC with one Dedicated Load Balancer fronting each on-premises database system, plus matching IP whitelisting in the load balancer and firewall rules in the VPC and on-premises network

D. Relocation of the database systems to a DMZ in the on-premises network, with Mule applications deployed to the CloudHub Shared Worker Cloud connecting only to the DMZ

Answer(s): A

---

2. Refer to the exhibit.


Each app exposes an HTTPS endpoint with the same port and certificate configured
How to package the HTTP Listener?
How to package or store the server-side certificate?
???
???
HTTP Listener config
Server-side certificate
Mule app 1
Mule app 2
Mule app 3
Mule app 4
Mule runtime

An organization deploys multiple Mule applications to the same customer -hosted Mule runtime. Many of these Mule applications must expose an HTTPS endpoint on the same port using a server- side certificate that rotates often.
What is the most effective way to package the HTTP Listener and package or store the server-side certificate when deploying these Mule applications, so the disruption caused by certificate rotation is minimized?

A. Package the HTTPS Listener configuration in a Mule DOMAIN project, referencing it from all Mule applications that need to expose an HTTPS endpoint Package the server-side certificate in ALL MuleAPPLICATIONS that need to expose an HTTPS endpoint

B. Package the HTTPS Listener configuration in a Mule DOMAIN project, referencing it from all Mule applications that need to expose an HTTPS endpoint. Store the server-side certificate in a shared filesystem location in the Mule runtime's classpath, OUTSIDE the Mule DOMAIN or any Mule APPLICATION

C. Package an HTTPS Listener configuration In all Mule APPLICATIONS that need to expose an HTTPS endpoint Package the server-side certificate in a NEW Mule DOMAIN project

D. Package the HTTPS Listener configuration in a Mule DOMAIN project, referencing It from all Mule applications that need to expose an HTTPS endpoint. Package the server-side certificate in the SAME Mule DOMAIN project Go to Set

Answer(s): B

---

3. An API client is implemented as a Mule application that includes an HTTP Request operation using a default configuration. The HTTP Request operation invokes an external API that follows standard HTTP status code conventions, which causes the HTTP Request operation to return a 4xx status code.
What is a possible cause of this status code response?

A. An error occurred inside the external API implementation when processing the HTTP request that was received from the outbound HTTP Request operation of the Mule application

B. The external API reported that the API implementation has moved to a different external endpoint

C. The HTTP response cannot be interpreted by the HTTP Request operation of the Mule application after it was received from the external API

D. The external API reported an error with the HTTP request that was received from the outbound HTTP Request operation of the Mule application

Answer(s): D

---

4. An XA transaction Is being configured that involves a JMS connector listening for Incoming JMS messages.
What is the meaning of the timeout attribute of the XA transaction, and what happens after the timeout expires?

A. The time that is allowed to pass between committing the transaction and the completion of the Mule flow After the timeout, flow processing triggers an error

B. The time that Is allowed to pass between receiving JMS messages on the same JMS connection After the timeout, a new JMS connection Is established

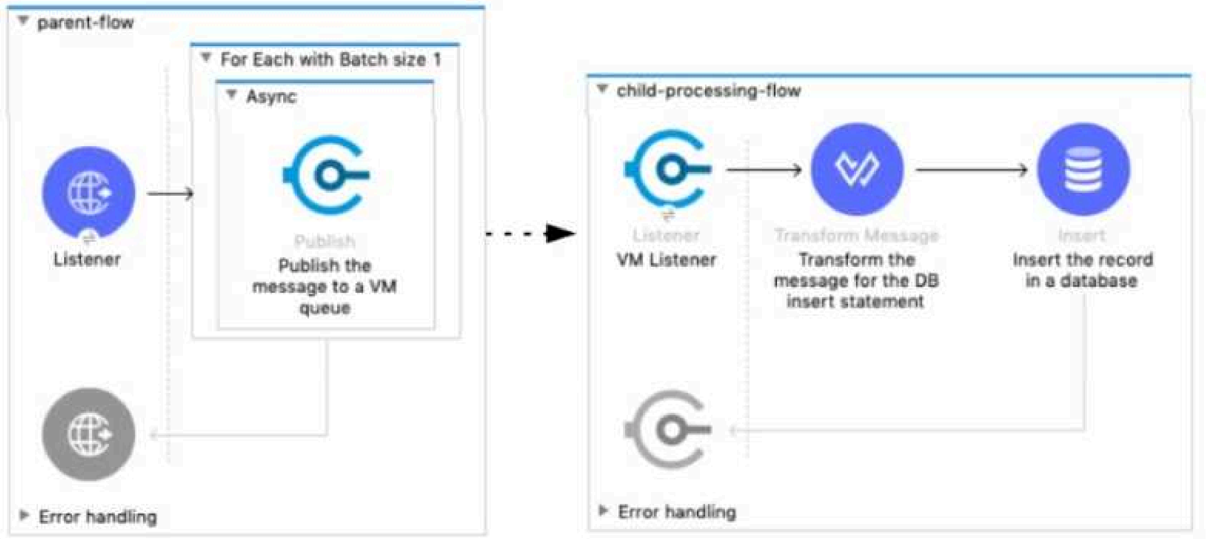
C. The time that Is allowed to pass without the transaction being ended explicitly After the timeout, the transaction Is forcefully rolled-back

D. The time that Is allowed to pass for state JMS consumer threads to be destroyed After the timeout, a new JMS consumer thread is created

Answer(s): C

---

5. Refer to the exhibit.


parent-flow
For Each with Batch size 1
Async
Listener
Publish
Publish the message to a VM queue
Error handling
child-processing-flow
Listener
VM Listener
Transform Message
Transform the message for the DB insert statement
Insert
Insert the record in a database
Error handling

A Mule 4 application has a parent flow that breaks up a JSON array payload into 200 separate items, then sends each item one at a time inside an Async scope to a VM queue. A second flow to process orders has a VM Listener on the same VM queue. The rest of this flow processes each received item by writing the item to a database.

This Mule application is deployed to four CloudHub workers with persistent queues enabled.

What message processing guarantees are provided by the VM queue and the CloudHub workers, and how are VM messages routed among the CloudHub workers for each invocation of the parent flow under normal operating conditions where all the CloudHub workers remain online?

A. EACH item VM message is processed AT MOST ONCE by ONE CloudHub worker, with workers chosen in a deterministic round-robin fashion Each of the four CloudHub workers can be expected to process 1/4 of the Item VM messages (about 50 items)

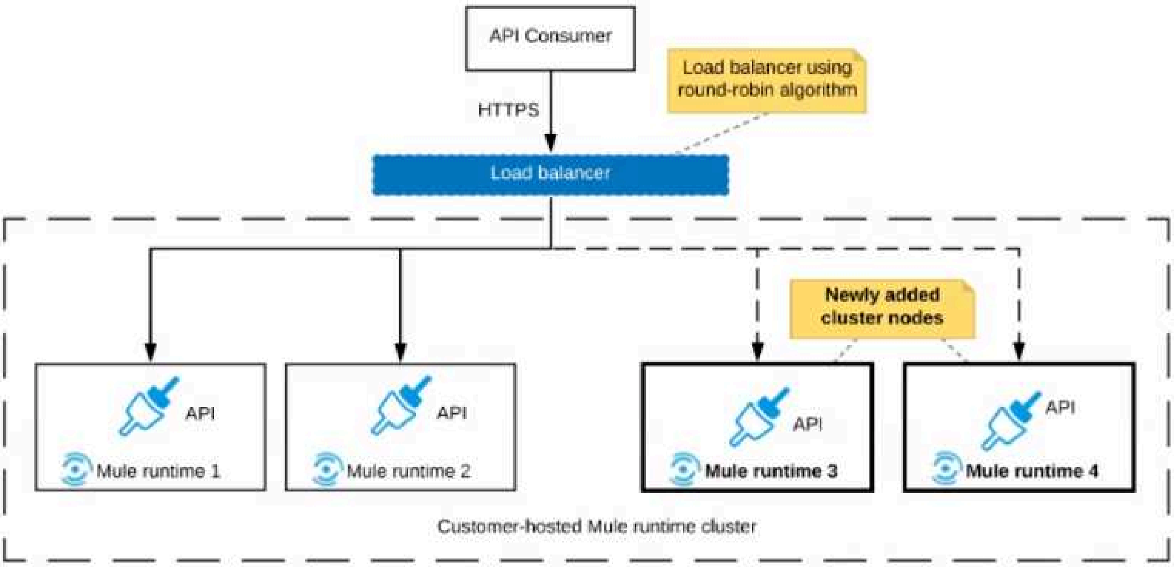
B. EACH item VM message is processed AT LEAST ONCE by ONE ARBITRARY CloudHub worker Each of the four CloudHub workers can be expected to process some item VM messages

C. ALL Item VM messages are processed AT LEAST ONCE by the SAME CloudHub worker where the parent flow was invokedThis one CloudHub worker processes ALL 200 item VM messages

D. ALL item VM messages are processed AT MOST ONCE by ONE ARBITRARY CloudHub worker This one CloudHub worker processes ALL 200 item VM messages

Answer(s): B

6. Refer to the exhibit.


API Consumer
Load balancer using round-robin algorithm
HTTPS
Load balancer
Newly added cluster nodes
API
Mule runtime 1
API
Mule runtime 2
API
Mule runtime 3
API
Mule runtime 4
Customer-hosted Mule runtime cluster

An organization uses a 2-node Mute runtime cluster to host one stateless API implementation. The API is accessed over HTTPS through a load balancer that uses round-robin for load distribution. Two additional nodes have been added to the cluster and the load balancer has been configured to recognize the new nodes with no other change to the load balancer.

What average performance change is guaranteed to happen, assuming all cluster nodes are fully operational?

A. 50% reduction in the response time of the API

B. 100% increase in the throughput of the API

C. 50% reduction In the JVM heap memory consumed by each node

D. 50% reduction In the number of requests being received by each node

Answer(s): D

---

7. An integration Mule application is deployed to a customer-hosted multi-node Mule 4 runtime duster. The Mule application uses a Listener operation of a JMS connector to receive incoming messages from a JMS queue.

How are the messages consumed by the Mule application?

A. Depending on the JMS provider's configuration, either all messages are consumed by ONLY the primary cluster node or else ALL messages are consumed by ALL cluster nodes

B. Regardless of the Listener operation configuration, all messages are consumed by ALL cluster nodes

C. Depending on the Listener operation configuration, either all messages are consumed by ONLY the primary cluster node or else EACH message is consumed by ANY ONE cluster node

D. Regardless of the Listener operation configuration, all messages are consumed by ONLY the primary cluster node

Answer(s): C

---

8. An Integration Mule application is being designed to synchronize customer data between two systems. One system is an IBM Mainframe and the other system is a Salesforce Marketing Cloud (CRM) instance. Both systems have been deployed in their typical configurations, and are to be invoked using the native protocols provided by Salesforce and IBM.
What interface technologies are the most straightforward and appropriate to use in this Mute application to interact with these systems, assuming that Anypoint Connectors exist that implement these interface technologies?

A. IBM: DB access CRM: gRPC

B. IBM: REST CRM:REST

C. IBM: Active MQ CRM: REST

D. IBM: CICS CRM: SOAP

Answer(s): D

---

9. What is required before an API implemented using the components of Anypoint Platform can be managed and governed (by applying API policies) on Anypoint Platform?

A. The API must be published to Anypoint Exchange and a corresponding API instance ID must be obtained from API Manager to be used in the API implementation

B. The API implementation source code must be committed to a source control management system (such as GitHub)

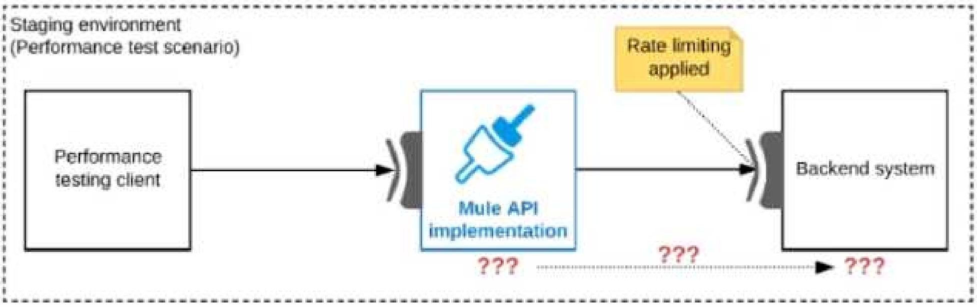
C. A RAML definition of the API must be created in API designer so it can then be published to Anypoint Exchange

D. The API must be shared with the potential developers through an API portal so API consumers can interact with the API

Answer(s): A

---

10. Refer to the exhibit.



One of the backend systems invoked by an API implementation enforces rate limits on the number of requests a particular client can make. Both the backend system and the API implementation are deployed to several non-production environments in addition to production. Rate limiting of the backend system applies to all non-production environments. The production environment, however, does NOT have any rate limiting.
What is the most effective approach to conduct performance tests of the API implementation in a staging (non-production) environment?

A. Create a mocking service that replicates the backend system's production performance characteristics. Then configure the API implementation to use the mocking service and conduct the performance tests

B. Use MUnit to simulate standard responses from the backend system then conduct performance tests to identify other bottlenecks in the system

C. Include logic within the API implementation that bypasses invocations of the backend system in a performance test situation. Instead invoking local stubs that replicate typical backend system responses then conduct performance tests using this API Implementation

D. Conduct scaled-down performance tests in the staging environment against the rate limited backend system then upscale performance results to full production scale

Answer(s): A

---

11. An API has been unit tested and is ready for integration testing. The API is governed by a Client ID Enforcement policy in all environments.
What must the testing team do before they can start integration testing the API in the Staging environment?

A. They must access the API portal and create an API notebook using the Client ID and Client Secret supplied by the API portal in the Staging environment

B. They must request access to the API instance in the Staging environment and obtain a Client ID and Client Secret to be used for testing the API

C. They must be assigned as an API version owner of the API in the Staging environment

D. They must request access to the Staging environment and obtain the Client ID and Client Secret for that environment to be used for testing the API

Answer(s): B

---

12. What requires configuration of both a key store and a trust store for an HTTP Listener?

A. Support for TLS mutual (two-way) authentication with HTTP clients

B. Encryption of requests to both subdomains and API resource endpoints fhttPs://aDi.customer.com/ and https://customer.com/api)

C. Encryption of both HTTP request and HTTP response bodies for all HTTP clients

D. Encryption of both HTTP request header and HTTP request body for all HTTP clients

Answer(s): A

---

13. A retailer is designing a data exchange interface to be used by its suppliers. The interface must support secure communication over the public internet. The interface must also work with a wide variety of programming languages and IT systems used by suppliers.
What are suitable interface technologies for this data exchange that are secure, cross-platform, and internet friendly, assuming that Anypoint Connectors exist for these interface technologies?

A. EDJFACT XML over SFTP JSON/REST over HTTPS

B. SOAP over HTTPS HOP over TLS gRPC over HTTPS

C. XML over ActiveMQ XML over SFTP XML/REST over HTTPS

D. CSV over FTP YAML over TLS JSON over HTTPS

Answer(s): C

---

14. An organization currently uses a multi-node Mule runtime deployment model within their datacenter, so each Mule runtime hosts several Mule applications. The organization is planning to transition to a deployment model based on Docker containers in a Kubernetes cluster. The organization has already created a standard Docker image containing a Mule runtime and all required dependencies (including a JVM), but excluding the Mule application itself.
What is an expected outcome of this transition to container-based Mule application deployments?

A. Required redesign of Mule applications to follow microservice architecture principles

B. Required migration to the Docker and Kubernetes-based Anypoint Platform - Private Cloud Edition

C. Required change to the URL endpoints used by clients to send requests to the Mule applications

D. Guaranteed consistency of execution environments across all deployments of a Mule application

Answer(s): A

---

15. A team would like to create a project skeleton that developers can use as a starting point when creating API Implementations with Anypoint Studio. This skeleton should help drive consistent use of best practices within the team.
What type of Anypoint Exchange artifact(s) should be added to Anypoint Exchange to publish the project skeleton?

A. A custom asset with the default API implementation

B. A RAML archetype and reusable trait definitions to be reused across API implementations

C. An example of an API implementation following best practices

D. a Mule application template with the key components and minimal integration logic

Answer(s): D

---

16. What aspect of logging is only possible for Mule applications deployed to customer-hosted Mule runtimes, but NOT for Mule applications deployed to CloudHub?

A. To send Mule application log entries to Splunk

B. To change tog4j2 tog levels in Anypoint Runtime Manager without having to restart the Mule application

C. To log certain messages to a custom log category

D. To directly reference one shared and customized log4j2.xml file from multiple Mule applications

Answer(s): D

---

17. What is true about the network connections when a Mule application uses a JMS connector to interact with a JMS provider (message broker)?

A. To complete sending a JMS message, the JMS connector must establish a network connection with the JMS message recipient

B. To receive messages into the Mule application, the JMS provider initiates a network connection to the JMS connector and pushes messages along this connection

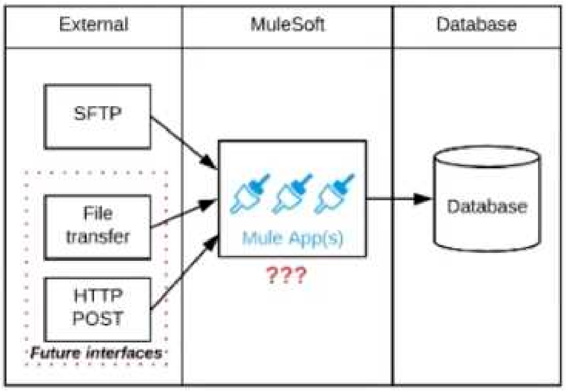
C. The JMS connector supports both sending and receiving of JMS messages over the protocol determined by the JMS provider

D. The AMQP protocol can be used by the JMS connector to portably establish connections to various types of JMS providers

Answer(s): C

---

18. Refer to the exhibit.



A business process involves the receipt of a file from an external vendor over SFTP. The file needs to be parsed and its content processed, validated, and ultimately persisted to a database. The delivery mechanism is expected to change in the future as more vendors send similar files using other mechanisms such as file transfer or HTTP POST.

What is the most effective way to design for these requirements in order to minimize the impact of future change?

A. Use a MuleSoft Scatter-Gather and a MuleSoft Batch Job to handle the different files coming from different sources

B. Create a Process API to receive the file and process it using a MuleSoft Batch Job while delegating the data save process to a System API

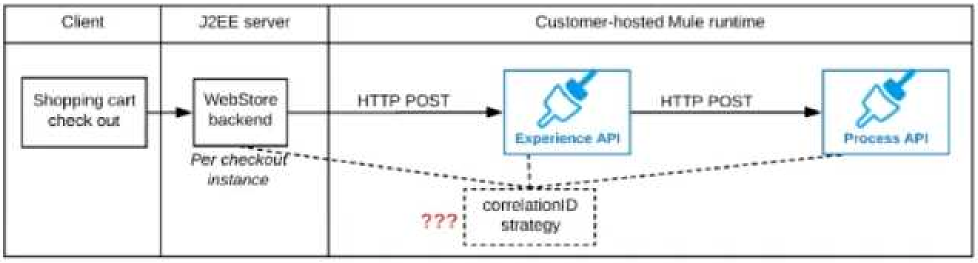
C. Create an API that receives the file and invokes a Process API with the data contained In the file, then have the Process API process the data using a MuleSoft Batch Job and other System APIs as needed

D. Use a composite data source so files can be retrieved from various sources and delivered to a MuleSoft Batch Job for processing

Answer(s): C

---

19. Refer to the exhibit.

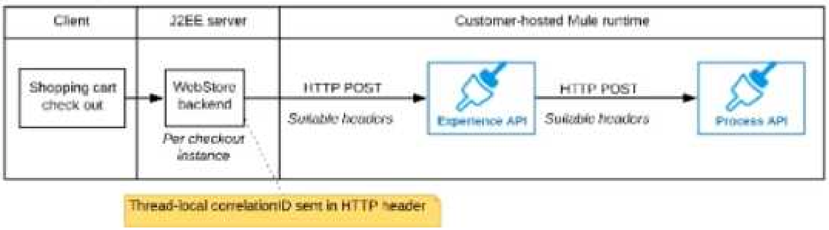


A shopping cart checkout process consists of a web store backend sending a sequence of API invocations to an Experience API, which in turn invokes a Process API. All API invocations are over HTTPS POST. The Java web store backend executes in a Java EE application server, while all API implementations are Mule applications executing in a customer -hosted Mule runtime. End-to-end correlation of all HTTP requests and responses belonging to each individual checkout Instance is required. This is to be done through a common correlation ID, so that all log entries written by the web store backend, Experience API implementation, and Process API implementation include the same correlation ID for all requests and responses belonging to the same checkout instance.

What is the most efficient way (using the least amount of custom coding or configuration) for the web store backend and the implementations of the Experience API and Process API to participate in end-to-end correlation of the API invocations for each checkout instance? A)

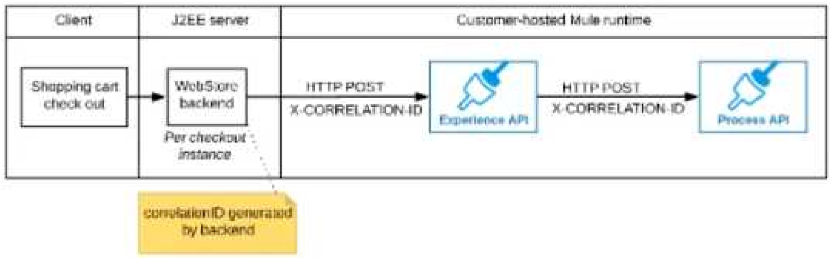
The web store backend, being a Java EE application, automatically makes use of the thread-local correlation ID generated by the Java EE application server and automatically transmits that to the Experience API using HTTP-standard headers

No special code or configuration is included in the web store backend, Experience API, and Process API implementations to generate and manage the correlation ID



B)

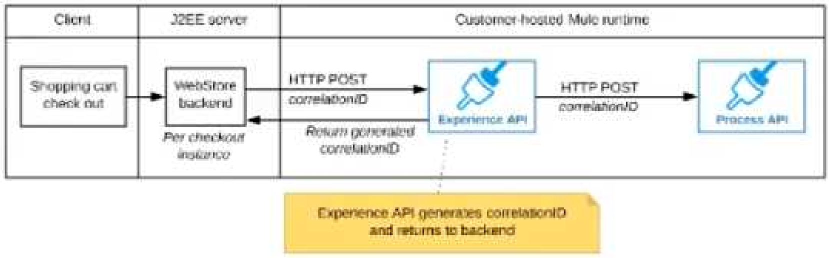
The web store backend generates a new correlation ID value at the start of checkout and sets it on the X-CORRELATlON-lt HTTP request header In each API invocation belonging to that checkout No special code or configuration is included in the Experience API and Process API implementations to generate and manage the correlation ID



C)

The Experience API implementation generates a correlation ID for each incoming HTTP request and passes it to the web store backend in the HTTP response, which includes it in all subsequent API invocations to the Experience API.

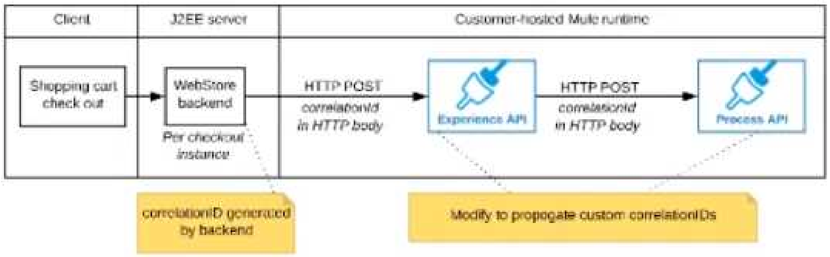
The Experience API implementation must be coded to also propagate the correlation ID to the Process API in a suitable HTTP request header



D)

The web store backend sends a correlation ID value in the HTTP request body In the way required by the Experience API

The Experience API and Process API implementations must be coded to receive the custom correlation ID In the HTTP requests and propagate It in suitable HTTP request headers



A. Option A

B. Option B

C. Option C

D. Option D

Answer(s): B

---

20. What operation can be performed through a JMX agent enabled in a Mule application?

A. View object store entries

B. Replay an unsuccessful message

C. Set a particular tog4J2 log level to TRACE

D. Deploy a Mule application

Answer(s): A