

Salesforce.Integration-Architect.v2024-02-05.q51

Exam Code:	Integration-Architect
Exam Name:	Salesforce Certified Integration Architect
Certification Provider:	Salesforce
Free Question Number:	51
Version:	v2024-02-05
# of views:	958
# of Questions views:	510
https://www.freepdfdumps.com/Salesforce.Integration-Architect.v2024-02-05.q51.html	

NEW QUESTION: 1

Sales representatives at Universal Containers (UC) use Salesforce Sales Cloud as their primary CRM. UC owns a legacy homegrown application that stores a copy of customer data as well. Sales representatives may edit or update Contact records in Salesforce if there is a change.

Both Salesforce and the homegrown application should be kept synchronized for consistency. UC has these requirements:

1. When a Contact record in Salesforce is updated, the external homegrown application should be
2. The synchronization should be event driven.
3. The integration should be asynchronous.

Which option should an architect recommend to satisfy the requirements?

- A.** Leverage Platform Events to publish a custom event message containing changes to the Contact object.
- B.** Leverage Change Data Capture to track changes to the Contact object and write a CometD subscriber on the homegrown application.
- C.** Write an Apex Trigger with the @future annotation.
- D.** Use an ETL tool to keep Salesforce and the homegrown application in sync on a regular cadence.

Answer: B (LEAVE A REPLY)

Explanation

Leverage Change Data Capture to track changes to the Contact object and write a CometD subscriber on the homegrown application. This solution meets the requirements of keeping Salesforce and the homegrown application synchronized for consistency, using event-driven and asynchronous integration. Change Data Capture is a feature that allows you to receive near real-time notifications of changes to Salesforce records, such as inserts, updates, deletes, and undeletes. You can use Change Data Capture to track changes to the Contact object, and publish them as events to an event bus. CometD is a

protocol that allows you to subscribe to events from the event bus using a streaming API. You can write a CometD subscriber on the homegrown application, and use it to receive the events and update the customer data accordingly.

References: Certification - Integration Architect - Trailhead, [Change Data Capture Developer Guide],
[Streaming API Developer Guide]

NEW QUESTION: 2

NorthernTrail Outfitters needs to send order and line items directly to an existing finance application webservice when an order is fulfilled. It is critical that each order reach the finance application exactly once for accurate invoicing.

What solution should an architect propose?

- A. Trigger invokes Queueable Apex method, with custom error handling process.
- B. Trigger makes @future Apex method, with custom error handling process.
- C. Button press invokes synchronous callout, with user handling retries in case of error
- D. Outbound Messaging, which will automatically handle error retries to the service.

Answer: (SHOW ANSWER)

Explanation

Trigger invokes Queueable Apex method, with custom error handling process. Queueable Apex allows you to run asynchronous jobs that can make callouts to external web services. You can use custom error handling logic to handle any failures and retry the callouts if necessary. You can also use Database.Stateful interface to maintain state across transactions and ensure that each order is sent exactly once. This solution meets the requirements of sending order and line items directly to an existing finance application web service when an order is fulfilled, and ensuring that each order reaches the finance application exactly once for accurate invoicing. References: Certification - Integration Architect - Trailhead, [Queueable Apex], [Making a Web Service Callout from a Queueable Apex Job]

NEW QUESTION: 3

A company wants to standardize exception tracking, handling, and analytics. Given the following actions:

1. Build a companywide logging service hosted on a middleware platform
2. Create case object records for exceptions-based thresholds
3. Change all their Apex Loggers to publish Application Exceptions as custom Platform Events.

Which two specifications should the integration architect include in the logging service architecture?

Choose 2 answers

- A. Receive Application Events through Change Data Capture (CDC).
- B. Create Salesforce Cases using the Salesforce REST, SOAP or Bulk API.

- C. Create Salesforce Cases conditionally using automatic Case creation rules.
- D. Subscribe to the Application Exceptions using the Salesforce Streaming API.

Answer: ([SHOW ANSWER](#))

Explanation

Creating Salesforce Cases using the Salesforce REST, SOAP or Bulk API is a solution that can allow the logging service to create case records in Salesforce based on the exceptions received from the middleware platform. This way, the logging service can use the standard APIs to interact with Salesforce and leverage the features and functionality of the case object. Creating Salesforce Cases conditionally using automatic Case creation rules is a solution that can allow the logging service to create case records in Salesforce based on predefined criteria and actions. This way, the logging service can reduce the number of API calls and avoid creating unnecessary or duplicate cases. Receiving Application Events through Change Data Capture (CDC) is not a solution that is related to this requirement, as CDC is used for capturing data changes in Salesforce and sending them to external systems, not for receiving data from external systems. Subscribing to the Application Exceptions using the Salesforce Streaming API is also not a solution that is related to this requirement, as Streaming API is used for subscribing to events or notifications from Salesforce, not for sending data to Salesforce. Reference: Salesforce Integration Architecture Designer Resource Guide, page 25-26

NEW QUESTION: 4

A company's security assessment noted vulnerabilities on the un managed packages in their Salesforce orgs, notably secrets that are easily accessible and in plain text, such as usernames, passwords, and OAuth tokens used in callouts from Salesforce.

Which two persistence mechanisms should an integration architect require to be used to ensure that secrets are protected from deliberate or inadvertent exposure?

Choose 2 answers

- A. Encrypted Custom Fields
- B. Named Credentials
- C. Protected Custom Metadata Types
- D. Protected Custom Settings

Answer: ([SHOW ANSWER](#))

Explanation

Named Credentials and Protected Custom Settings are two persistence mechanisms that can be used to ensure that secrets are protected from deliberate or inadvertent exposure. Named Credentials allow you to specify the URL of a callout endpoint and its required authentication parameters in one definition. Salesforce manages all the authentication for Apex callouts that specify a named credential as the callout endpoint, and you don't have to add more authentication logic in your Apex code. Named Credentials can be defined to provide a secure and convenient way of setting up authenticated callouts, and they can also be used in Lightning components, Visualforce pages, and flows¹.

Protected Custom Settings are a type of custom settings that store application-specific data that is hidden from subscribers. They are only accessible by the managed package that created them, and they can be used to store secrets such as encryption keys, passwords, or tokens. Protected Custom Settings can be accessed by Apex code, formulas, or validation rules within the same namespace as the settings².

Encrypted Custom Fields are not a suitable mechanism for storing secrets, because they are not designed to prevent unauthorized access to sensitive data. Encrypted Custom Fields allow you to encrypt text fields using a standard encryption scheme. The encrypted data is masked in reports, list views, and search results, but it can still be viewed by users who have the "View Encrypted Data" permission. Encrypted Custom Fields are intended to protect data from unauthorized access by users within your organization, not from external threats or malicious code³.

Protected Custom Metadata Types are another type of custom metadata types that store application-specific data that is hidden from subscribers. They are similar to Protected Custom Settings, but they have some advantages such as being deployable using change sets or Metadata API, being accessible by SOQL queries, and being able to reference other metadata types or settings. However, Protected Custom Metadata Types cannot be used to store secrets, because they do not support encryption or masking of sensitive data. Protected Custom Metadata Types are intended to store configuration data that is specific to your managed package, not secrets that need to be secured.

Therefore, the correct answer is B and D, because Named Credentials and Protected Custom Settings are the only persistence mechanisms that can be used to securely store secrets in Salesforce.

References: 1: Named Credentials | Apex Developer Guide | Salesforce Developers 2: Custom Settings | Apex Developer Guide | Salesforce Developers 3: Encrypted Fields | Salesforce Help : [Protected Custom Metadata Types | ISVforce Guide | Salesforce Developers]

NEW QUESTION: 5

A conglomerate is designing a Lightning Web Component (LWC) to display transactions aggregated from different sources. Their current system landscape is as follows:

1. Transactions are created at any time through their various on-premise and cloud-based systems.
2. All necessary transactions are replicated to a custom Transaction object in Salesforce. It is updated periodically so it only has a subset of the necessary transactions between updates.
3. Middleware supports publish-subscribe interactions and provides RESTful Enterprise APIs that can retrieve transactions from on-premise and cloud-based systems.

The company wants to address a usability concern regarding incomplete data displayed on the LWC component. What should the Integration Architect specify so the LWC will be able to display all the required transactions?

- A.** Let the Lightning Data Service with an wire adapter display new values when the custom object records change.
- B.** Use the Continuation class to call the Enterprise APIs and then process the response in a callback method.
- C.** Call the Enterprise APIs directly from the LWC's JavaScript code and redisplay the LWC on receipt of the API response.
- D.** Publish a Platform Event, have the middleware subscribe and update the custom object on receipt of Platform Event.

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 6

Northern Trail Outfitters is seeking to improve the performance and security of outbound integrations from Salesforce to on-premise servers.

What should the Architect consider before recommending a solution?

- A.** External gateway products in use
- B.** A Default gateway restrictions
- C.** Considerations for using Deterministic Encryption
- D.** ShieldPlatform Encryption Limitations

Answer: ([SHOW ANSWER](#))

Explanation

Option A is correct because external gateway products in use can affect the performance and security of outbound integrations from Salesforce to on-premise servers. External gateway products are software or hardware devices that act as intermediaries between Salesforce and the on-premise servers, such as firewalls, proxies, load balancers, or VPNs. They can have different configurations, features, and limitations that can impact the speed, reliability, and security of the data transmission. For example, some external gateway products may require authentication, encryption, or compression of the data, which can add overhead and latency to the integration. Some external gateway products may also have bandwidth or throughput limits, which can affect the scalability and availability of the integration.

Therefore, the architect should consider the external gateway products in use before recommending a solution.

Option B is incorrect because a default gateway restriction is not a factor that can affect the performance and security of outbound integrations from Salesforce to on-premise servers. A default gateway restriction is a feature that allows administrators to restrict outbound requests from Salesforce to a specific IP address or domain name. This can help prevent unauthorized or malicious requests from Salesforce to external systems. However, this feature does not affect the performance or security of the outbound requests themselves, as it only acts as a filter for the destination of the requests.

Option C is incorrect because considerations for using deterministic encryption are not relevant for outbound integrations from Salesforce to on-premise servers. Deterministic

encryption is a type of encryption that produces the same ciphertext for the same plaintext input. This can help preserve some functionality and performance of encrypted data in Salesforce, such as filtering, sorting, and indexing.

However, deterministic encryption is not applicable for outbound integrations from Salesforce to on-premise servers, as it is only supported for custom fields and not for standard fields or attachments.

Moreover, deterministic encryption does not affect the security of the data transmission itself, as it only encrypts the data at rest in Salesforce.

Option D is incorrect because Shield Platform Encryption limitations are not relevant for outbound integrations from Salesforce to on-premise servers. Shield Platform Encryption is a feature that allows administrators to encrypt sensitive data at rest in Salesforce using AES 256-bit encryption. This can help protect data from unauthorized access or theft.

However, Shield Platform Encryption limitations are not relevant for outbound integrations from Salesforce to on-premise servers, as they only affect the functionality and performance of encrypted data in Salesforce, such as searching, reporting, or validation rules. Shield Platform Encryption does not affect the security of the data transmission itself, as it only encrypts the data at rest in Salesforce.

References: Salesforce Integration Patterns and Practices : Salesforce Integration Guide : Restrict Outbound Requests with a Default Gateway : Deterministic Encryption : Shield Platform Encryption Considerations :

Shield Platform Encryption : Shield Platform Encryption Architecture

NEW QUESTION: 7

Northern Trail Outfitters needs to make synchronous callouts "available to promise" services to query product availability and reserve inventory during customer checkout process.

Which two considerations should an integration architect make when building a scalable integration solution?

Choose 2 answers

- A.** The number batch jobs that can run concurrently.
- B.** How many concurrent service calls are being placed.
- C.** The typical and worst-case historical response times.
- D.** The maximum query cursors open per user on the service.

Answer: B,C (LEAVE A REPLY)

Explanation

The typical and worst-case historical response times, and how many concurrent service calls are being placed are two considerations that an integration architect should make when building a scalable integration solution for synchronous callouts to "available to promise" services. These two factors can affect the performance, reliability, and availability of the integration solution, as well as the user experience of the customer checkout process. The integration architect should design the solution to handle high volumes of

service calls, optimize the response times, handle errors and timeouts, and avoid hitting governor limits or service quotas.

References: Certification - Integration Architect - Trailhead, [Callout Limits and Limitations], [Integration Patterns and Practices]

NEW QUESTION: 8

A company's cloud-based single page application consolidates data local to the application with data from on premise and 3rd party systems. The diagram below typifies the application's combined use of synchronous and asynchronous calls.

The company wants to use the average response time of its application's user interface as a basis for certain alerts. For this purpose, the following occurs:

1. Log every call's start and finish date and time to a central analytics data store.
2. Compute response time uniformly as the difference between the start and finish date and time - A to H in the diagram.

Which computation represents the end-to-end response time from the user's perspective?

- A. Sum of A to H
- B. Sum of A to F
- C. Sum of A, G, and H
- D. Sum of A and H

Answer: D (LEAVE A REPLY)

Explanation

The end-to-end response time from the user's perspective is the time elapsed between the user's request and the user's receipt of the final response. In the diagram, this corresponds to the sum of A and H, which are the durations of the synchronous calls from the user interface to the cloud-based application and back. The other durations (B to G) are either internal to the cloud-based application or asynchronous calls that do not affect the user's perception of response time. Therefore, the correct answer is D, because it represents the sum of A and H.

NEW QUESTION: 9

A call center manager uses a custom dashboard to track Case related metrics. The manager wants a component to display the number of closed Cases in real time.

What type of event should be used to meet this requirement?

- A. Push Topic Event
- B. Change Data Capture Event
- C. Platform Event
- D. Generic Event

Answer: B (LEAVE A REPLY)

Explanation

Change Data Capture Event is the best option to meet this requirement. Change Data Capture Event is a type of streaming event that notifies subscribers of changes to Salesforce records, such as creation, update, delete, and undelete operations¹. By subscribing to Change Data Capture Event for the Case object, the dashboard component can receive real-time updates on the number of closed Cases. Push Topic Event is another type of streaming event that notifies subscribers of changes to Salesforce records that match a SOQL query². However, Push Topic Event has some limitations, such as not supporting all SOQL features and not capturing delete and undelete operations³. Platform Event is a type of streaming event that delivers custom notifications within the Salesforce platform or from external sources⁴. Platform Event is not suitable for this requirement because it is not tied to Salesforce records and requires custom logic to publish and subscribe. Generic Event is a type of streaming event that sends custom JSON notifications to subscribers without a predefined schema⁵. Generic Event is not suitable for this requirement because it is not tied to Salesforce records and requires custom logic to publish and subscribe.

NEW QUESTION: 10

A company needs to integrate a legacy on premise application that can only support SOAP API. After the Integration Architect has evaluated the requirements and volume, they determined that the Fire and Forget integration pattern will be most appropriate for sending data from Salesforce to the external application and getting response back in a strongly typed format.

Which integration capabilities should be used to integrate the two systems?

- A.** Outbound Message for Salesforce to Legacy System direction and SOAP API using Enterprise WSDL for the communication back from legacy system to salesforce.
- B.** Platform Events for Salesforce to Legacy System direction and SOAP API using Partner WSDL for the communication back from legacy system to salesforce.
- C.** Platform Events for Salesforce to Legacy System direction and SOAP API using Enterprise WSDL for the communication back from legacy system to salesforce.
- D.** Outbound Message for Salesforce to Legacy System direction and SOAP API using Partner WSDL for the communication back from legacy system to salesforce.

Answer: A (LEAVE A REPLY)

Explanation

Outbound Message is a SOAP-based notification service that sends information about changes in Salesforce to a specified endpoint. It is a type of Remote Process Invocation-Fire and Forget integration pattern, which means that Salesforce does not wait for a response from the external system. Outbound Message can be configured using workflow rules or process builder, and it can include a subset of fields of the object that triggered the message. Outbound Message requires the external system to expose a SOAP web service that conforms to the WSDL generated by Salesforce¹.

SOAP API is a way to access Salesforce data and functionality using SOAP (Simple Object Access Protocol), which is a standard XML-based protocol for exchanging structured data over the web. SOAP API can be used for both inbound and outbound integration, and it supports two types of WSDL (Web Service Description Language) files: Enterprise and Partner. Enterprise WSDL is a strongly typed WSDL file that is specific to an organization's Salesforce configuration. It can be used when the client application knows the objects and fields it needs to access at compile time. Partner WSDL is a loosely typed WSDL file that is generic for any organization. It can be used when the client application needs to access data dynamically at run time².

Therefore, the correct answer is A, because Outbound Message is suitable for Fire and Forget integration from Salesforce to the legacy system, and SOAP API using Enterprise WSDL is suitable for getting response back in a strongly typed format from the legacy system to Salesforce.

References: 1: Remote Process Invocation-Fire and Forget | Integration Patterns and Practices | Salesforce Developers 2: SOAP API Developer Guide | SOAP API Basics | Salesforce Developers

NEW QUESTION: 11

Which WSDL should an architect consider when creating an integration that might be used for more than one Salesforce organization and different metadata?

- A. Corporate WSDL
- B. Partner WSDL
- C. SOAP API WSDL
- D. Enterprise WSDL

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

Explanation

The Partner WSDL is the best option for creating an integration that might be used for more than one Salesforce organization and different metadata. The Partner WSDL is loosely typed and can reflect against any configuration of Salesforce. It is static and does not change if modifications are made to an organization's Salesforce configuration.

Therefore, it is more flexible and adaptable than the Enterprise WSDL, which is strongly typed and bound to a specific configuration of Salesforce¹ References: Differences between Salesforce provided WSDL files

NEW QUESTION: 12

An Integration Architect has built a Salesforce application that integrates multiple systems and keeps them synchronized via Platform Events.

What is taking place if events are only being published?

- A. The platform events are published immediately before the Apex transaction completes.
- B. The platform events are published after the Apex transaction completes.
- C. The platform events has a trigger in Apex.

D. The platform events are being published from Apex.

Answer: B (LEAVE A REPLY)

Explanation

Option B is correct because platform events are published after the Apex transaction completes.

Platform events are asynchronous messages that are published and consumed by different processes or systems. When an Apex transaction publishes a platform event, the event is not sent immediately, but rather added to a temporary queue. The event is only published and delivered to the subscribers after the Apex transaction commits successfully. This ensures that the event data is consistent with the database state and that the event is not lost in case of a rollback.

Option A is incorrect because platform events are not published immediately before the Apex transaction completes. As explained above, platform events are published after the Apex transaction commits successfully. Publishing events before the transaction completes can cause data inconsistency and event loss issues.

Option C is incorrect because platform events can have a trigger in Apex, but this is not related to the question of when events are published. A platform event trigger is a special type of Apex trigger that executes when a platform event message is received by Salesforce. A platform event trigger can perform logic based on the event data, such as updating records, sending notifications, or calling external services. A platform event trigger does not affect the publishing of events, but rather the consumption of events.

Option D is incorrect because platform events can be published from Apex, but this does not answer the question of when events are published. Platform events can be published from various sources, such as Apex code, Process Builder, Flows, or API calls. Publishing events from Apex requires creating an instance of a platform event object and using the EventBus.publish method to add it to the temporary queue. The actual publishing of events happens after the Apex transaction completes, regardless of the source of the events.

References: Publish and Subscribe with Platform Events: Publish Platform Events :
Subscribe to Platform Events : Write a Platform Event Trigger : Create and Publish
Platform Events in Apex : EventBus Class

NEW QUESTION: 13

Which three considerations should an Integration Architect consider when recommending Platform Event as a Integration solution?

Choose 3 answers

A. Inability to query event messages using SOQL

B. Subscribe to an AssetToken Event stream to monitor OAuth2.0 authentication activity.

C

C. Inability to create a Lightning record page for platform events.

D. When you delete an event definition, it's permanently removed and can't be restored.

E. You can use Event Monitoring to track user activity, such as logins and running reports.

Answer: ([SHOW ANSWER](#))

Explanation

These are three considerations that an Integration Architect should consider when recommending Platform Event as an Integration solution. Platform Events are a type of event-driven architecture that allows you to publish and subscribe to custom events in Salesforce and external systems. Platform Events have some limitations and features that an Integration Architect should be aware of, such as:

Inability to query event messages using SOQL. Platform Events are not stored in Salesforce database, but in an event bus that retains them for 24 hours. You cannot use SOQL to query event messages, but you can use the EventLogFile object or the Event Monitoring feature to access them.

Inability to create a Lightning record page for platform events. Platform Events are not standard or custom objects, but a special type of sObject that can be inserted into the event bus. You cannot create a Lightning record page for platform events, but you can use Apex triggers, Lightning web components, or CometD clients to subscribe to them.

When you delete an event definition, it's permanently removed and can't be restored. An event definition is a metadata type that defines the schema and properties of a platform event message. You can create or modify event definitions in Salesforce Setup or using Metadata API. However, if you delete an event definition, it's permanently removed from your org and can't be restored. You also lose access to any event messages that use that event definition.

References: Certification - Integration Architect - Trailhead, [Platform Events Developer Guide]

NEW QUESTION: 14

Universal Containers (UC) is a leading provider of management training globally, UC requested students course registration data generated from the Salesforce student community to be synced with the learning management system (LMS). Any update to the course registration data needs to be reflected in the LMS.

Which integration mechanism should be used to meet the requirement?

- A.** Change Data Capture (CDC)
- B.** Platform Event
- C.** Streaming API
- D.** Outbound Message

Answer: ([SHOW ANSWER](#))

Explanation

Change Data Capture (CDC) is the best option to meet the requirement. CDC is a type of streaming event that notifies subscribers of changes to Salesforce records, such as creation, update, delete, and undelete operations.

By subscribing to CDC events for the course registration data, the LMS can receive real-time updates and sync with Salesforce. Streaming API is another type of streaming event

that notifies subscribers of changes to Salesforce records that match a SOQL query. However, Streaming API has some limitations, such as not supporting all SOQL features and not capturing delete and undelete operations. Platform Event is a type of streaming event that delivers custom notifications within the Salesforce platform or from external sources.

Platform Event is not suitable for this requirement because it is not tied to Salesforce records and requires custom logic to publish and subscribe. Outbound Message is a workflow action that sends SOAP messages with field values to designated endpoints. Outbound Message is not suitable for this requirement because it does not support complex data types, such as sObjects or collections.

NEW QUESTION: 15

A company needs to be able to send data from Salesforce to a homegrown system behind a corporate firewall.

The data needs to be pushed only one way and doesn't need to be sent in real time. The average volume is 2 million records per day.

What should an integration architect consider when choosing the right option in building the integration between the external system and Salesforce?

- A.** Due to high volume of records, number of concurrent requests can hit the limit for the REST API call to external system.
- B.** Due to high volume of records, a third-party integration tool is required to stage records off platform.
- C.** Due to high volume of records, the external system will need to use a BULK API Rest endpoint to connect to salesforce.
- D.** Due to high volume of records, salesforce will need to make a REST API call to external system.

Answer: ([SHOW ANSWER](#))

Explanation

Using a third party integration tool to stage records off platform is a solution that can handle the high volume of data and avoid hitting the API limits for the REST API call to the external system. A third party integration tool can also provide features such as data transformation, error handling, and logging. Due to high volume of records, number of concurrent requests can hit the limit for the REST API call to external system is not a solution, but a problem that needs to be addressed. Due to high volume of records, the external system will need to use a BULK API Rest endpoint to connect to Salesforce is not a solution, as the requirement is to send data from Salesforce to the external system, not vice versa. Due to high volume of records, Salesforce will need to make a REST API call to external system is not a solution, as it does not address how to handle the high volume of data and avoid hitting the API limits. Reference: Salesforce Integration Architecture Designer Resource Guide, page 18-19

NEW QUESTION: 16

Universal Containers (UC) uses Salesforce to track the following customer data:

1. Leads,
2. Contacts
3. Accounts
4. Cases

Salesforce is considered to be the system of record for the customer. In addition to Salesforce, customer data exists in an Enterprise Resource Planning (ERP) system, ticketing system, and enterprise data lake. Each of these additional systems have their own unique identifier. UC plans on using middleware to integrate Salesforce with the external systems.

UC has a requirement to update the proper external system with record changes in Salesforce and vice versa.

Which two solutions should an Integration Architect recommend to handle this requirement?

Choose 2 answers

- A.** Locally cache external ID'S at the middleware layer and design business logic to map updates between systems.
- B.** Store unique identifiers in an External ID field in Salesforce and use this to update the proper records across systems.
- C.** Use Change Data Capture to update downstream systems accordingly when a record changes.
- D.** Design an MDM solution that maps external ID's to the Salesforce record ID.

Answer: C,D (LEAVE A REPLY)

Explanation

Using Change Data Capture (CDC) to update downstream systems accordingly when a record changes is a solution that can handle this requirement by capturing data changes in Salesforce and sending them to external systems via a publish-subscribe model. This way, the external systems can receive near real-time updates from Salesforce and synchronize their data accordingly. Designing an MDM solution that maps external ID's to the Salesforce record ID is a solution that can handle this requirement by creating a master data hub that stores and manages the unique identifiers of each system and their relationships. This way, the MDM solution can ensure data quality, consistency, and accuracy across systems. Locally caching external ID's at the middleware layer and designing business logic to map updates between systems is not a good solution because it can introduce performance and scalability issues, as well as increase the complexity and maintenance cost of the middleware layer. Storing unique identifiers in an External ID field in Salesforce and using this to update the proper records across systems is not enough to handle this requirement, as it does not address how to update Salesforce with record changes from external systems. Reference: Salesforce Integration Architecture Designer Resource Guide, page 27-28

Valid Integration-Architect Dumps shared by Actual4test.com for Helping Passing Integration-Architect Exam! Actual4test.com now offer the **newest Integration-Architect exam dumps**, the Actual4test.com Integration-Architect exam **questions have been updated** and **answers have been corrected** get the **newest** Actual4test.com Integration-Architect dumps with Test Engine here:
https://www.actual4test.com/Integration-Architect_examcollection.html (130 Q&As Dumps, **30%OFF Special Discount: Freepdfdumps**)

NEW QUESTION: 17

A customer is migrating from an old legacy system to Salesforce. As part of the modernization effort, they would like to integrate all existing systems that currently work with their legacy application with Salesforce.

Which three constraints and pain-points should an integration architect consider when choosing the integration pattern/mechanism?

Choose 3 answers

- A. System types - APIs, File systems, Email
- B. Reporting and usability requirements
- C. Multi-language and multi-currency requirement
- D. Error handling mechanisms
- E. Data Volume and Processing volume

Answer: A,D,E (LEAVE A REPLY)

Explanation

The system types, the error handling mechanisms, and the data volume and processing volume are three constraints and pain-points that an integration architect should consider when choosing the integration pattern/mechanism. The system types determine what kind of interfaces and protocols are available or required for the integration, such as APIs, file systems, email, etc. The error handling mechanisms ensure that the integration can handle any failures or exceptions gracefully and provide appropriate logging and notification. The data volume and processing volume affect the performance and scalability of the integration, as well as the choice of synchronous or asynchronous methods. Reference: Salesforce Integration Architecture Designer Resource Guide, page 17

NEW QUESTION: 18

Northern Trail Outfitters submits orders to the manufacturing system web-service.

Recently, the system has experienced outages that keeping service unavailable for several days.

What solution should an architect recommend to handle errors during these types of service outages?

- A.** Use middleware queuing and buffering to insulate Salesforce from system outages.
- B.** A Use Platform Event replayId and custom scheduled Apex process to retrieve missed events.
- C.** Use @future jobId and custom scheduled apex process to retry failed service calls.
- D.** Use Outbound Messaging to automatically retry failed service calls.

Answer: ([SHOW ANSWER](#))

Explanation

Using middleware queuing and buffering is a solution that can handle errors during service outages by storing the messages in a queue until the service is available again. This way, Salesforce does not lose any data or encounter any failures when the service is down. Using @future jobId and custom scheduled apex process to retry failed service calls is not a good solution because it can consume a lot of governor limits and create a lot of duplicate records. Using Outbound Messaging to automatically retry failed service calls is also not a good solution because it has a limited number of retries and a fixed retry interval, which may not be sufficient for long service outages. Using Platform Event replayId and custom scheduled Apex process to retrieve missed events is not applicable for this scenario because Platform Events are used for event-driven integration, not for web-service integration. Reference: Salesforce Integration Architecture Designer Resource Guide, page 29-30

NEW QUESTION: 19

Northern Trail Outfitters needs to present shipping costs and estimated delivery times to their customers.

Shipping services used vary by region, and have similar but distinct service request parameters.

Which integration component capability should be used?

- A.** Enterprise Service Bus to determine which shipping service to use, and transform requests to the necessary format.
- B.** Outbound Messaging to request costs and delivery times from Shipper delivery services with automated error retry.
- C.** APEX REST Service to implement routing logic to the various shipping service.
- D.** Enterprise Service Bus user interface to collect shipper-specific form data.

Answer: ([SHOW ANSWER](#))

Explanation

Using an Enterprise Service Bus (ESB) to determine which shipping service to use, and transform requests to the necessary format is a good solution because it can provide routing, transformation, mediation, and orchestration capabilities for integrating different services. An ESB can also abstract the complexity and heterogeneity of the services from the client application, which simplifies the integration. Using Outbound Messaging to request costs and delivery times from Shipper delivery services with automated error retry is not a good solution because Outbound Messaging is a Salesforce-specific feature that

uses SOAP web services, which may not be compatible with all shipping services. Using APEX REST Service to implement routing logic to the various shipping service is also not a good solution because it can introduce performance and scalability issues, as well as increase the maintenance cost and complexity of the code. Using Enterprise Service Bus user interface to collect shipper-specific form data is not a valid option because an ESB does not have a user interface component. Reference: Salesforce Integration Architecture Designer Resource Guide, page 16-17

NEW QUESTION: 20

Northern Trail Outfitters wants to improve the quality of call-outs from Salesforce to their REST APIs. For this purpose, they will require all API clients/consumers to adhere to RESTAPI Markup Language (RAML) specifications that include field-level definition of every API request and response payload. RAML specs serve as interface contracts that Apex REST API Clients can rely on.

Which two design specifications should the Integration Architect include in the integration architecture to ensure that ApexREST API Clients unit tests confirm adherence to the RAML specs?

Choose 2 answers

- A. Call the Apex REST API Clients in a test context to get the mock response.
- B. Require the Apex REST API Clients to implement the HttpCalloutMock.
- C. Call the HttpCalloutMock implementation from the Apex REST AP Clients.
- D. Implement HttpCalloutMock to return responses per RAML specification.

Answer: A,D (LEAVE A REPLY)

Explanation

The HttpCalloutMock interface allows testing HTTP callouts by returning a predefined response in a test context. By implementing HttpCalloutMock to return responses per RAML specification, the Apex REST API Clients unit tests can confirm that the API requests and responses match the expected format and values.

Calling the Apex REST API Clients in a test context to get the mock response is also necessary to verify the adherence to the RAML specs. Calling the HttpCalloutMock implementation from the Apex REST API Clients or requiring the Apex REST API Clients to implement the HttpCalloutMock are not valid options because the HttpCalloutMock interface is implemented by a separate class that is passed as a parameter to the Test.setMock method2 References: 1: Idempotent REST APIs 2: Testing HTTP Callouts by Implementing the HttpCalloutMock Interface

NEW QUESTION: 21

Northern Trail Outfitters has a requirement to encrypt few of widely used standard fields. They also want to be able to use these fields in workflow rules.

Which security solution should an Integration Architect recommend to fulfill the business use case?

- A. Cryptography Class
- B. Data Masking
- C. Classic Encryption
- D. Platform Shield Encryption

Answer: ([SHOW ANSWER](#))

Explanation

Option D is correct because Platform Shield Encryption allows you to encrypt standard fields and use them in workflow rules. Platform Shield Encryption uses AES 256-bit encryption to protect sensitive data at rest, while preserving platform functionality¹² Option A is incorrect because Cryptography Class is a set of Apex methods for creating digests, message authentication codes, signatures, and encryption. It does not provide a way to encrypt standard fields or use them in workflow rules³ Option B is incorrect because Data Masking is a feature that replaces sensitive data in sandboxes with dummy data. It does not encrypt data in production or allow the use of encrypted data in workflow rules⁴⁵ Option C is incorrect because Classic Encryption is a feature that encrypts custom fields with 128-bit AES encryption. It does not support standard fields or workflow rules⁶⁷

References: 1: Salesforce Shield - Data Monitoring & End to End Encryption 2: How Shield Platform Encryption Works 3: Crypto Class 4: Data Masking - Anonymize Sensitive Data 5: Secure Your Sandbox Data with Salesforce Data Mask 6: What's the Difference Between Classic Encryption and Shield Platform Encryption? 7: Salesforce Classic vs. Salesforce Shield Platform Encryption: Which One Do You Need?

NEW QUESTION: 22

Which two approaches will require the least amount of development effort?

Choose 2 answers

- A. Configure named credentials in the source org.
- B. Use the Composite REST API to aggregate multiple leads in a single call.
- C. Use the tooling API with Process Builder to insert leads in real time.
- D. Call the Salesforce REST API to insert the lead into the target system.

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

The two approaches that will require the least amount of development effort are configuring named credentials in the source org and using the Composite REST API to aggregate multiple leads in a single call. Named credentials are a type of metadata that store authentication information for accessing external services, such as the target Salesforce org. By using named credentials, you can simplify the code for making callouts and avoid hardcoding credentials or tokens. The Composite REST API is a resource that allows you to execute multiple REST API requests in a single call. You can use the Composite REST API to create, update, or delete up to 25 records in one request. This can reduce the number of API calls and improve performance.

References: [Named Credentials], [Composite Resources]

NEW QUESTION: 23

Northern Trail Outfitters wants to use Salesforce as a front end for creating accounts using the lead-to-opportunity process.

1. An order is created in Salesforce when the opportunity is closed and won, but the back-end ERP system is the data master for order,
2. Customer wants to be able to see within Salesforce all the stages of order processing like Order Created, Order Shipped, Order Paid that are within the retention window.

Which two message durability considerations should an Integration Architect make when designing a solution to meet these business requirements?

Choose 2 answers

- A.** When subscribing to Salesforce Event bus, ReplayID is used with a value of -2 to be able to see old and new events.
- B.** High-volume event messages are stored for 24 hours (one day).
- C.** When subscribing to Salesforce Event bus, ReplayID is used with a value of -1 to be able to see new events.
- D.** High-volume event messages are stored for 72 hours (three days).

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

Explanation

The message durability considerations that an Integration Architect should make when designing a solution to meet these business requirements are:

When subscribing to Salesforce Event bus, ReplayID is used with a value of -2 to be able to see old and new events.

High-volume event messages are stored for 72 hours (three days).

A Salesforce Event bus is a messaging service that allows you to publish and subscribe to events in Salesforce or from external sources. ReplayID is a property of an event message that indicates its position in the event stream. By using ReplayID with a value of -2, you can replay all events stored in the retention window, which can help you see the stages of order processing in Salesforce. High-volume event messages are a type of event message that can handle large volumes of data and are optimized for performance and scalability. High-volume event messages are stored for 72 hours (three days) in the event bus, which means they can be replayed within that time frame. The other options are not correct. Using ReplayID with a value of -1 means you can only see new events that are published after your subscription. High-volume event messages are not stored for 24 hours (one day), but for 72 hours (three days).

NEW QUESTION: 24

Service Agents at Northern Trail Outfitters uses Salesforce to manage cases and B2C Commerce for ordering.

Which integration solution should an architect recommend in order for the service agents to see order history from a B2C Commerce system?

- A.** Salesforce B2C Commerce to Service Cloud Connector

- B.** A REST API offered by Commerce Platform
- C.** Mulesoft Anypoint Platform
- D.** REST API offered by Salesforce Platform

Answer: ([SHOW ANSWER](#))

Explanation

Option A is correct because Salesforce B2C Commerce to Service Cloud Connector is an integration solution that allows service agents to see order history from a B2C Commerce system. Salesforce B2C Commerce to Service Cloud Connector is a pre-built package that integrates Salesforce B2C Commerce and Service Cloud using REST APIs and Platform Events. It enables service agents to view, edit, cancel, and refund orders from B2C Commerce within the Service Cloud console. It also supports features such as customer verification, order search, order details, and order history.

Option B is incorrect because a REST API offered by Commerce Platform is not an integration solution that allows service agents to see order history from a B2C Commerce system. A REST API offered by Commerce Platform is a set of web services that expose the functionality and data of the Commerce Platform to external applications. It can be used to create, update, delete, or query resources such as products, catalogs, customers, or orders. However, a REST API offered by Commerce Platform is not a complete integration solution, as it requires additional development, configuration, and maintenance to connect with Service Cloud and display the order history in the Service Cloud console.

Option C is incorrect because Mulesoft Anypoint Platform is not an integration solution that allows service agents to see order history from a B2C Commerce system. Mulesoft Anypoint Platform is a platform that enables developers to build, manage, and monitor integrations and APIs across various systems and applications. It can be used to connect Salesforce B2C Commerce and Service Cloud using various connectors, protocols, and transformations. However, Mulesoft Anypoint Platform is not a pre-built integration solution, as it requires additional development, configuration, and maintenance to connect with Service Cloud and display the order history in the Service Cloud console.

Option D is incorrect because a REST API offered by Salesforce Platform is not an integration solution that allows service agents to see order history from a B2C Commerce system. A REST API offered by Salesforce Platform is a set of web services that expose the functionality and data of the Salesforce Platform to external applications. It can be used to create, update, delete, or query resources such as objects, records, metadata, or Apex classes. However, a REST API offered by Salesforce Platform is not a complete integration solution, as it requires additional development, configuration, and maintenance to connect with B2C Commerce and display the order history in the Service Cloud console.

References: Salesforce B2C Commerce to Service Cloud Connector : Salesforce B2C Commerce to Service Cloud Connector Implementation Guide : Commerce API Explorer : MuleSoft | Integration Platform for Connecting SaaS and Enterprise Applications : REST API Developer Guide

NEW QUESTION: 25

Northern Trail Outfitters' (NTO) Salesforce org usually goes through 8k-10k batches a day to synch data from external sources. NTO's Integration Architect has received requirements for a new custom object, FooBarc, for which 90M records will need to be loaded into the org. Once complete, 20GB (about 30M records) needs to be extracted to an external auditing system.

What should the architect recommend using to meet these requirements in a day?

A. Insert using Bulk API 2.0 and query using REST API.

B. Insert and query using Bulk API 1.0.

C. Insert using Bulk API 1.0 and query using REST API.

D. Insert and query using Bulk API 2.0.

Answer: ([SHOW ANSWER](#))

Answer D is valid because using Bulk API 2.0 for both inserting and querying data can meet the requirements of loading and extracting large volumes of data in a day. Bulk API 2.0 is a RESTful API that allows creating, updating, deleting, or querying millions of records asynchronously by uploading or downloading CSV or JSON files. Bulk API 2.0 has several advantages over Bulk API 1.0, such as:

It does not count against the daily limit of 5,000 batches per rolling 24-hour period.

It supports PK Chunking for queries, which automatically splits large data sets into manageable chunks based on the primary key.

It simplifies the job lifecycle and reduces the number of API calls needed to perform a bulk operation.

Answer A is not valid because using Bulk API 2.0 for inserting data and REST API for querying data is not an optimal solution for handling large volumes of data in a day. REST API is a synchronous API that allows creating, updating, deleting, or querying individual records or small batches of records using HTTP methods. REST API has several limitations and drawbacks for this use case, such as:

It counts against the daily limit of 15,000 API requests per 24-hour period.

It does not support PK Chunking for queries, which means that large data sets may exceed the query timeout or heap size limits.

It requires more API calls and processing time to perform a bulk operation than Bulk API 2.0.

Answer B is not valid because using Bulk API 1.0 for both inserting and querying data is not a feasible solution for handling large volumes of data in a day. Bulk API 1.0 is a SOAP-based API that allows creating, updating, deleting, or querying millions of records asynchronously by uploading or downloading XML or CSV files. Bulk API 1.0 has several limitations and drawbacks compared to Bulk API 2.0, such as:

It counts against the daily limit of 5,000 batches per rolling 24-hour period, which may not be enough to load and extract 90M and 30M records respectively.

It does not support JSON format for data files, which may not be compatible with some external systems or applications.

It requires more API calls and complexity to manage the job lifecycle and handle errors or retries than Bulk API 2.0.

Answer C is not valid because using Bulk API 1.0 for inserting data and REST API for querying data is not a suitable or reliable solution for handling large volumes of data in a day. As explained above, both Bulk API 1.0 and REST API have limitations and drawbacks that may affect the performance, efficiency, and scalability of the integration.

NEW QUESTION: 26

An organization needs to integrate Salesforce with an external system and is considering authentication options. The organization already has implemented SAML, using a third-party Identity Provider for integrations between other systems.

Which use case can leverage the existing SAML integration to connect Salesforce with other internal systems?

- A. Make formula fields with HYPERLINK() to external web servers more secure.
- B. Make Apex SOAP outbound integrations to external web services more secure.
- C. A Make Apex REST outbound integrations to external web services more secure.
- D. Make an API inbound integration from an external Java client more secure.

Answer: B (LEAVE A REPLY)

Explanation

The best use case for leveraging the existing SAML integration to connect Salesforce with other internal systems is to make Apex SOAP outbound integrations to external web services more secure. SAML can be used to authenticate the Salesforce org as the service provider and obtain a session ID from the external system as the identity provider. This session ID can then be used to make SOAP calls to the external web service without exposing any credentials in the Apex code. Option A is not correct because formula fields with HYPERLINK() do not support SAML authentication. Option C is not correct because Apex REST outbound integrations require OAuth or basic authentication, not SAML. Option D is not correct because API inbound integrations from an external Java client require OAuth or basic authentication, not SAML. References:

Named Credentials as Callout Endpoints

SAML SSO with Salesforce as the Service Provider

NEW QUESTION: 27

Northern Trail Outfitters' ERP is integrated with Salesforce and syncs several million contacts per day. To prevent specific data from syncing, the integration uses a SOQL query filtered by sharing hierarchy.

Which two things should an architect do to improve the performance of the integration?

Choose 2 answers

- A. Include non-selective criteria in query filters.
- B. Remove the query filters.
- C. Include selective criteria in query filters.

D. Remove the sharing restrictions.

Answer: ([SHOW ANSWER](#))

Explanation

Option C is correct because including selective criteria in query filters can improve the performance of the integration. Selective criteria are filters that reduce the number of records that need to be scanned by the query optimizer, such as indexed fields, standard indexes, or custom indexes. Selective criteria can help the query run faster and avoid hitting the query timeout limit or the query row limit.

Option D is correct because removing the sharing restrictions can improve the performance of the integration. Sharing restrictions are filters that limit the access to records based on the user's role, profile, or sharing rules. Sharing restrictions can add complexity and overhead to the query execution, as they require additional joins and calculations. Removing the sharing restrictions can simplify the query and reduce the number of records that need to be processed.

Option A is incorrect because including non-selective criteria in query filters can degrade the performance of the integration. Non-selective criteria are filters that do not reduce the number of records that need to be scanned by the query optimizer, such as non-indexed fields, formula fields, or OR conditions. Non-selective criteria can cause the query to run slower and hit the query timeout limit or the query row limit.

Option B is incorrect because removing the query filters can degrade the performance of the integration.

Query filters are conditions that specify which records to retrieve from the database, such as WHERE clauses or LIMIT clauses. Query filters can help the query run faster and avoid retrieving unnecessary or unwanted data. Removing the query filters can increase the number of records that need to be processed and transmitted by the integration.

References: Working with Very Large SOQL Queries: Improve performance with custom indexes using Salesforce Query Plan tool: Querying Data That Respects User

Permissions: How does sharing affect SOQL performance?: SOQL SELECT Syntax : SOQL Best Practices

NEW QUESTION: 28

Northern Trail Outfitters (NTO) has an integration set up between a Salesforce org and a quoting system. NTO would like to show a notification to all sales representatives that use Salesforce anytime the quoting system will be taken down for maintenance.

Which Salesforce API should an Integration Architect use to fulfill this requirement?

A. Connect REST API

B. Tooling API

C. REST API

D. Streaming API

Answer: ([SHOW ANSWER](#))

Explanation

Streaming API is the best option for sending real-time notifications of changes to Salesforce data. Streaming API uses a publish-subscribe model to push relevant data to subscribers without polling. Streaming API supports PushTopic events, which are based on SOQL queries that define the data changes to listen for. The affiliate company can subscribe to a PushTopic event on the NTO Salesforce org and receive notifications whenever the data that matches the query changes. This way, the affiliate company can be informed of any updates to the opportunities in the NTO Salesforce instance.

NEW QUESTION: 29

A large enterprise customer with the following system landscape is planning to implement Salesforce Sales Cloud.

The following business processes need to be supported in Salesforce:

1. Sales Consultants should be able to have access to current inventory.
2. Enterprise Resource Planning System(ERP) is the system of record for pricing information.
3. Quotes should be generated in Salesforce with pricing from ERP.
4. Sales Management uses a Enterprise Business Intelligence (BI) tool to view Sales dashboards.
5. Master Data Management (MDM) is the system of record for customers and prospects.
6. Invoices should be accessible in Salesforce.

Which systems in the landscape should the Integration Consultant consider to be integrated with Salesforce to support the business requirements?

- A.** ERP, MDM, BI tool and Data Warehouse
- B.** ERP, Invoices system, Data Warehouse and BI Tool
- C.** ERP, Inventory, Pricing Engine, Invoices system
- D.** ERP, MDM, Data Warehouse, Invoices system

Answer: C (LEAVE A REPLY)

NEW QUESTION: 30

What is the first thing an Integration Architect should validate if a callout from a LightningWeb Component to an external endpoint is failing?

- A.** The endpoint domain has been added to Cross-Origin Resource Sharing.
- B.** The endpoint URL has been added to Content Security Policies.
- C.** The endpoint URL has added been to an outbound firewall rule.
- D.** The endpoint URL has been added to Remote Site Settings.

Answer: D (LEAVE A REPLY)

Explanation

The first thing an integration architect should validate if a callout from a Lightning Web Component to an external endpoint is failing is the endpoint domain has been added to Cross-Origin Resource Sharing (CORS).

CORS is a mechanism that allows web browsers to make requests to servers on different origins, such as different domains, protocols, or ports. CORS requires the server to send back a special header that indicates whether the browser is allowed to access the resource or not. If the endpoint domain is not added to the CORS whitelist in Salesforce, the browser will block the callout and throw an error. Option B is not correct because Content Security Policies (CSP) are used to control what resources can be loaded on a Visualforce or Lightning page, such as scripts, stylesheets, images, etc. CSP does not affect the callout from a Lightning Web Component to an external endpoint. Option C is not correct because outbound firewall rules are used to restrict the network traffic from Salesforce to external systems. Firewall rules are configured at the network level, not at the Salesforce level. Option D is not correct because Remote Site Settings are used to specify the domains that are allowed for callouts from Apex code, not from Lightning Web Components. References:

Working with CORS and CSP to Call APIs from LWC
[Cross-Origin Resource Sharing (CORS)]

NEW QUESTION: 31

A global financial company sells financial products and services that include the following:

1. Bank Accounts
2. Loans
3. Insurance

The company has a core banking system that is state of the art and is the master system to store financial transactions, financial products and customer information. The core banking system currently processes 10M financial transactions per day. The CTO for the company is considering building a community port so that customers can review their bank account details, update their information and review their account financial transactions.

What should an integration architect recommend as a solution to enable customer community users to view their financial transactions?

- A. Use Salesforce Connect to display the financial transactions as an external object.
- B. Use Salesforce Connect to display the financial transactions as an external object.
- C. Use Salesforce External Service to display financial transactions in a community lightning page.
- D. Use Iframe to display core banking financial transactions data in the customer community.

Answer: (SHOW ANSWER)

Explanation

The integration architect should recommend using Salesforce Connect to display the financial transactions as an external object. Salesforce Connect is a feature that allows you to integrate external data sources with Salesforce and access them in real time via external objects. External objects are similar to custom objects, but they store metadata only and not data. You can use external objects to display data from the core banking

system in the customer community without copying or syncing the data. You can also use standard Salesforce features, such as reports, dashboards, or global search, with external objects. Salesforce External Service is a feature that allows you to import an external schema definition and generate an Apex wrapper class that can invoke an external service. This is useful for integrating complex business processes or workflows with Salesforce, but not for displaying data in a community lightning page. Iframe is a HTML element that allows you to embed another web page within a web page. This is not a recommended solution for displaying data in a customer community, as it can pose security risks, performance issues, or user interface problems.

References: [Salesforce Connect], [Salesforce External Services], [Using Iframes in Lightning Components]

Valid Integration-Architect Dumps shared by Actual4test.com for Helping Passing Integration-Architect Exam! Actual4test.com now offer the **newest Integration-Architect exam dumps**, the Actual4test.com Integration-Architect exam **questions have been updated** and **answers have been corrected** get the **newest**

Actual4test.com Integration-Architect dumps with Test Engine here:

https://www.actual4test.com/Integration-Architect_examcollection.html (130 Q&As

Dumps, **30%OFF** Special Discount: **Freepdfdumps**)

NEW QUESTION: 32

A US business-to-consumer (B2C) company is planning to expand to Latin America. They project an initial Latin American customer base of about one million, and a growth rate of around 10% every year for the next 5 years. They anticipate privacy and data protection requirements similar to those in the European Union to come into effect during this time. Their initial analysis indicates that key personal data is stored in the following systems:

1. Legacy mainframe systems that have remained untouched for years and are due to be decommissioned.
2. Salesforce Commerce Cloud Service Cloud, Marketing Cloud, and Community Cloud.
3. The company's CIO tasked the integration architect with ensuring that they can completely delete their Latin American customer's personal data on demand.

Which three requirements should the integration architect consider?

Choose 3 answers

- A. Manual steps and procedures that may be necessary.
- B. Impact of deleted records on system functionality.
- C. Ability to delete personal data in every system.
- D. Feasibility to restore deleted records when needed.
- E. Ability to provide a 360-degree view of the customer.

Answer: B,C,E (LEAVE A REPLY)

Explanation

The integration architect should consider the impact of deleted records on system functionality, the ability to delete personal data in every system, and the ability to provide a 360-degree view of the customer. These are important requirements for ensuring that the company can comply with the privacy and data protection regulations, as well as deliver a customer-centric service. Option A is not correct because manual steps and procedures are not desirable for deleting personal data on demand. The integration architect should aim for an automated and reliable solution that minimizes human intervention and errors. Option D is not correct because restoring deleted records when needed may violate the privacy and data protection regulations, as well as the customer's consent. The integration architect should ensure that the deletion of personal data is permanent and irreversible.

NEW QUESTION: 33

A company that is a leading provider of training delivers courses to students globally. The company decided to use customer community in order to allow Studer to log in to the community, register for courses and pay course fees. The company has a payment gateway that takes more than 30 seconds to process the payment transaction. Students would like to get the payment result in real-time so in case an error happens, the students can retry the payment process.

What is the recommended integration approach to process payments based on this requirement?

- A. Use platform event to process payment to the payment gateway.
- B. Use continuation to process payment to the payment gateway.
- C. Use change data capture to process payment to the payment gateway.
- D. Use request and reply to make an API call to the payment gateway.

Answer: B (LEAVE A REPLY)

Explanation

The continuation class in Apex allows you to make long-running requests to external web services and process their responses in a callback method. This is useful for scenarios where the response time exceeds the normal Apex limits, such as 10 seconds for synchronous transactions or 120 seconds for asynchronous transactions. By using continuation, you can avoid hitting these limits and provide a better user experience.

References: Callout Limits and Limitations

NEW QUESTION: 34

An Architect is required to integrate with an External Data Source via a Named Credential with an Apex callout due to technical constraints.

How is authentication achieved?

- A. Handle authentication with login flows.
- B. Handle authentication in the code.
- C. Connect via Salesforce Connect.

D. Connect via Communities.

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

Explanation

The authentication is achieved by handling it in the code. A Named Credential is a type of metadata that specifies the URL of a callout endpoint and its required authentication parameters. An Apex callout is a way to invoke an external web service or API from Apex code. When using a Named Credential with an Apex callout, the authentication is handled by the Apex code that invokes the callout. The Apex code can use the `HttpRequest` class to set the endpoint, method, headers, and body of the callout request, and use the `Http` class to send the request and receive the response. The Apex code can also use the `NamedCredential` class to access the properties of the Named Credential, such as the principal type, protocol, and credential type. Handling authentication with login flows is not a valid option because login flows are a way to customize the user login process in Salesforce, not for external web services or APIs. Connecting via Salesforce Connect or Communities is not a valid option because they are not related to Apex callouts or Named Credentials.

NEW QUESTION: 35

A Salesforce customer is planning to roll out Salesforce for all their Sales and Service staff. Senior Management has requested that monitoring is to be in place for Operations to notify any degradation in Salesforce performance.

How should an integration consultant implement monitoring?

- A.** Use Salesforce limits API to capture current API usage and configure alerts for monitoring.
- B.** Use `APIEVENT` to track all user-initiated API calls through SOAP, REST or BULK APIs.
- C.** Identify critical business processes and establish automation to monitor performance against established benchmarks.
- D.** Request Salesforce to monitor the Salesforce instance and notify when there is degradation in performance.

Answer: ([SHOW ANSWER](#)**)**

Explanation

The integration consultant should implement monitoring by identifying critical business processes and establishing automation to monitor performance against established benchmarks. This approach can help detect and diagnose any degradation in Salesforce performance, such as slow response time, high error rate, or low availability. The automation can use various tools and methods, such as platform events, event monitoring, health check, or third-party monitoring services¹²³ References: Proactive Monitoring - Salesforce.com, 17 Free Ways to Monitor Your Salesforce Org | Salesforce Ben, Measure Lightning Experience Performance and Experienced ... - Trailhead

NEW QUESTION: 36

Northern Trail Outfitters (NTO) has hired an Integration Architect to design the integrations between existing systems and a new instance of Salesforce. NTO has the following requirements:

1. Initial load of 2M Accounts, 5.5M Contacts, 4.3M Opportunities, and 45k Products into the new org.
2. Notification of new and updated Accounts and Contacts needs to be sent to 3 external systems.
3. Expose custom business logic to 5 external applications in a highly secure manner.
4. Schedule nightly automated dataflows, recipes and data syncs.

Which set of APIs are recommended in order to meet the requirements?

- A.** Bulk API, Chatter REST API, Apex SOAP API, Tooling API
- B.** Bulk API, Chatter REST API, Apex REST API, Analytics REST API
- C.** Bulk API, Streaming API, Apex REST API, Analytics REST API
- D.** Bulk API, Streaming API, Apex SOAP API, Analytics REST API

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

Explanation

The set of APIs recommended to meet the requirements are Bulk API, Streaming API, Apex REST API, and Analytics REST API. Bulk API is a RESTful API that allows you to quickly load large amounts of data into Salesforce. Bulk API is suitable for the initial load of millions of records into the new org. Streaming API is a type of streaming event that notifies subscribers of changes to Salesforce records that match a SOQL query. Streaming API is suitable for sending notifications of new and updated Accounts and Contacts to external systems. Apex REST API is an API that allows you to expose custom business logic as RESTful web services that can be accessed by external applications. Apex REST API is suitable for exposing custom business logic in a highly secure manner. Analytics REST API is an API that allows you to access analytics features, such as dashboards, lenses, datasets, and dataflows. Analytics REST API is suitable for scheduling nightly automated dataflows, recipes and data syncs. The other sets of APIs are not suitable for meeting all the requirements.

Chatter REST API is an API that allows you to access Chatter feeds and social data such as users, groups, followers, and files. Chatter REST API is not relevant for this scenario.

Apex SOAP API is an API that allows you to expose custom business logic as SOAP web services that can be accessed by external applications.

Apex SOAP API is less preferable than Apex REST API because it requires more bandwidth and processing time due to the XML format. Tooling API is an API that allows you to build custom development tools or apps for Salesforce applications. Tooling API is not relevant for this scenario.

NEW QUESTION: 37

When user clicks Check Preferences as part of a Lightning flow in Salesforce, preferences from an externally hosted RESTful service are to be checked in real-time. The RESTful

service has OpenAPI 2.0 JSON definitions, responding in data types of Boolean and string values.

Which integration pattern and mechanism should be selected to meet the conditions?

A. Fire and Forget: Process-driven platform events publishes events on Salesforce Event Bus.

B. Remote Call-In: Salesforce REST API with REST Composite Resources.

C. Request-Reply: Enhanced External Services invokes a REST API.

D. Data Virtualization: Salesforce Connect map data external REST data in external objects.

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

Explanation

The correct answer is C, Request-Reply: Enhanced External Services invokes a REST API. This is because Enhanced External Services allows you to register an external web service that has an OpenAPI 2.0 specification and use it as an invocable action in a Lightning flow. This way, you can check the preferences from the external service in real-time and get the response in Boolean and string values. The other options are not suitable for this scenario because:

A, Fire and Forget: Process-driven platform events publishes events on Salesforce Event Bus, is a pattern that is used for asynchronous integration, where the sender does not expect a response from the receiver. This is not suitable for checking preferences in real-time.

B, Remote Call-In: Salesforce REST API with REST Composite Resources, is a pattern that is used for integrating external applications with Salesforce by calling the Salesforce REST API. This is not suitable for invoking an external service from a Lightning flow.

D, Data Virtualization: Salesforce Connect map data external REST data in external objects, is a pattern that is used for accessing external data without copying or synchronizing it to Salesforce. This is not suitable for checking preferences in real-time or using them in a Lightning flow.

References:

Enhanced External Services

Get Started with External Services

Registering external services in salesforce

NEW QUESTION: 38

Northern Trail Outfitters (NTO) has recently implemented middleware for orchestration of services across platforms. The ERP system being used requires transactions be captured near real time at a REST endpoint initiated in Salesforce when creating an order object. Additionally, the Salesforce team has limited development resources and requires a low code solution.

Which two options will fulfill the use case requirements?

Choose 2 answers

- A.** Use Remote Process Invocation fire and forget pattern on insert on the order object using Flow Builder.
- B.** Implement a Workflow Rule with Outbound Messaging to send SOAP messages to the designated endpoint.
- C.** Implement Change Data Capture on the order object and leverage the replay Id in the middleware solution.
- D.** Use a process builder to create a Platform Event, selecting the record type as the Platform Event Name on insert of record.

Answer: ([SHOW ANSWER](#))

Explanation

Answer A is valid because the Remote Process Invocation fire and forget pattern allows sending a message from Salesforce to the ERP system without waiting for a response or acknowledgement. This enables near real-time integration and reduces the latency and complexity of the integration. The Flow Builder is a low code tool that can be used to invoke an Apex action that makes the HTTP callout to the REST endpoint on insert of the order object¹² Answer C is valid because the Change Data Capture feature allows capturing changes in Salesforce data and publishing them as events to a streaming channel. The middleware can subscribe to the channel and receive the events that contain the order details. The replay Id is a unique identifier for each event that can be used by the middleware to resume from the last processed event in case of failures or interruptions. This ensures reliable and consistent integration between Salesforce and the ERP system³ Answer B is not valid because the Outbound Messaging feature uses SOAP messages to send notifications to external systems based on workflow rules. This requires configuring a SOAP endpoint and a WSDL file for the ERP system, which may not be compatible with the REST endpoint requirement. Moreover, Outbound Messaging does not guarantee delivery or order of messages, which may affect the accuracy and timeliness of the integration.

Answer D is not valid because the Platform Event feature allows publishing custom events from Salesforce to external systems or vice versa using a publish/subscribe model. However, this requires creating a platform event object and defining its fields and permissions, which may not be a low code solution. Moreover, using the record type as the platform event name may not be a valid or meaningful option, as record types are used to define different business processes or user interfaces for an object.

1: Remote Process Invocation-Fire and Forget 2: Invoke an Apex Class That Makes a Callout 3: Change Data Capture : Replay Events Using Middleware : Outbound Messaging Implementation Guide : Platform Events Developer Guide

NEW QUESTION: 39

A company that is a leading provider of courses and training delivers courses using third party trainers. The trainer for the company has to be verified from 10 different training

accreditation verification agencies before providing training for the company. Each training accreditation agency has its own response time, which could take days to confirm a trainer. The company decided to automate the trainer accreditation verification process by integrating to the agencies' web services.

What is the recommended approach to automate this process?

A. Use salesforce external service to make the call out, Salesforce external service should check the verification agencies until the result is verified, then update the trainer status to "verified".

B. Create a trigger on the trainer record to make a Callout to each verification agencies, write business logic to consolidate the verification then update the trainer status to verified".

C. Make an apex callout using @future annotation to make the call out to all different agencies. The response should update the trainer status to "verified".

D. Use middleware to handle the call out to the 10 different verification services, the middleware will handle the business logic of consolidating the verification result from t 10 services, then make a call-in to salesforce and update the verification status to "verified".

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

Explanation

Answer D is valid because using middleware to handle the call out to the 10 different verification services is a scalable and reliable solution that can handle the complexity and variability of the integration. The middleware can orchestrate the calls to the different web services, consolidate the verification results, and handle any errors or retries. The middleware can then make a call-in to Salesforce and update the verification status to "verified" using an API or a platform event¹² Answer A is not valid because using Salesforce external service to make the call out to the 10 different verification services is not a feasible or efficient solution. Salesforce external service is a feature that allows invoking an external service from a flow and mapping its inputs and outputs to flow variables.

However, this feature requires configuring an Apex action, a named credential, and an external service definition for each web service, which is not a low code solution.

Moreover, this feature does not support checking the verification agencies until the result is verified, as it only invokes the external service once per flow interview³ Answer B is not valid because creating a trigger on the trainer record to make a callout to each verification agency is not a recommended or robust solution. Triggers are Apex code that execute before or after database events, such as insert, update, or delete. However, triggers cannot make callouts directly, as they are part of a database transaction and must complete quickly. To make a callout from a trigger, an asynchronous process such as a future method or a queueable job must be used, which adds complexity and overhead to the integration. Moreover, triggers have limits on the number of callouts and asynchronous calls they can make per transaction, which may affect the scalability and reliability of the integration.

Answer C is not valid because making an Apex callout using @future annotation to make the call out to all different agencies is not a suitable or reliable solution. The @future annotation allows marking a method for execution at a later time when system resources become available. However, this annotation has several limitations and drawbacks, such as:

Future methods cannot return values, so they cannot update the trainer status to "verified" directly.

Future methods have limits on the number of callouts and future calls they can make per execution, which may affect the scalability and reliability of the integration.

Future methods run in their own thread and do not share any static variables or state with other methods, which makes it difficult to consolidate the verification results from different agencies.

Future methods are not guaranteed to execute at a specific time or order, which may affect the timeliness and accuracy of the integration.

1: Orchestration Pattern 2: Remote Process Invocation-Request and Reply 3: External Services : Apex Developer Guide: Triggers : Apex Developer Guide: Using Future Methods

NEW QUESTION: 40

Which two requirements should the Salesforce Community Cloud support for self-registration and SSO?

Choose 2 answers

- A.** SAML SSO and Registration Handler
- B.** OpenId Connect Authentication Provider and Registration Handler
- C.** SAML SSO and just-in-time provisioning
- D.** OpenId Connect Authentication Provider and just-in-time provisioning

Answer: B,C (LEAVE A REPLY)

Explanation

OpenId Connect Authentication Provider and Registration Handler, and SAML SSO and just-in-time provisioning are two requirements that the Salesforce Community Cloud can support for self-registration and SSO. OpenId Connect is a protocol that allows users to authenticate with an external identity provider and access Salesforce resources. A registration handler is a class that implements the Auth.RegistrationHandler interface and defines the logic for creating or updating a user in Salesforce after authentication. SAML SSO is a protocol that allows users to log in to Salesforce with a single click, using an assertion from an identity provider that confirms the user's identity. Just-in-time provisioning is a feature that allows Salesforce to create or update a user account based on the information in the SAML assertion. References: Certification - Integration Architect - Trailhead, [OpenID Connect Authentication Providers], [SAML Single Sign-On (SSO) Service for Salesforce]

NEW QUESTION: 41

Northern Trail Outfitters needs to make synchronous callouts "available to promise" services to query product availability and reserve inventory during customer checkout process.

Which two considerations should an integration architect make when building a scalable integration solution?

Choose 2 answers

- A.** The typical and worst-case historical response times.
- B.** The number batch jobs that can run concurrently.
- C.** How many concurrent service calls are being placed.
- D.** The maximum query cursors open per user on the service.

Answer: ([SHOW ANSWER](#))

Explanation

The typical and worst-case historical response times, and how many concurrent service calls are being placed are two considerations that an integration architect should make when building a scalable integration solution for synchronous callouts to "available to promise" services. These two factors can affect the performance, reliability, and availability of the integration solution, as well as the user experience of the customer checkout process. The integration architect should design the solution to handle high volumes of service calls, optimize the response times, handle errors and timeouts, and avoid hitting governor limits or service quotas.

References: Certification - Integration Architect - Trailhead, [Callout Limits and Limitations], [Integration Patterns and Practices]

NEW QUESTION: 42

Universal Containers is a global financial company that sells financial products and services including, bank accounts, loans, and insurance. UC uses Salesforce Service cloud to service their customer via calls, live chat.

The support agents would open bank accounts on the spot for customers who are inquiring about UC bank accounts.

UC Core banking system is the system of record for bank accounts and all accounts opened in salesforce have to be synced in real-time to the core banking system. Support agents need to inform the customers with the newly created bank account ID which has to be generated from the core banking system.

Which integration pattern is recommended for this use case?

- A.** Use streaming API to generate push topic.
- B.** Use outbound message.
- C.** Use salesforce platform event.
- D.** Use request and reply.

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

Explanation

Using request and reply is the recommended integration pattern for this use case because it allows the support agents to send a request to the core banking system and receive a response with the bank account ID in real-time. This way, the support agents can inform the customers with the newly created bank account ID without any delay or inconsistency. Using streaming API to generate push topic is not a good solution because it is used for event-driven integration, not for web-service integration. Using outbound message is also not a good solution because it is a Salesforce-specific feature that uses SOAP web services, which may not be compatible with the core banking system. Using Salesforce platform event is also not a good solution because it is used for event-driven integration, not for web-service integration. Reference: Salesforce Integration Architecture Designer Resource Guide, page 29-30

NEW QUESTION: 43

Given the diagram below, a Salesforce org, middleware, and Historical data store (with 20million records and growing) exists with connectivity between them

Historical records are archived from Salesforce and moved to Historical Data store (which houses 20M records and growing; fine-tuned to be performant with search queries).

Call center agents use Salesforce, when reviewing occasional special cases, have requested access to view the related historical case items that relate to submit cases.

Which mechanism and patterns are recommended to maximize declarative configuration?

- A.** Use ESB tool with Data Virtualization pattern, expose OData endpoint, and then use Salesforce Connect to consume and display the External Object alongside with the Case object.
- B.** C Use an ESB tool with a fire and forget pattern and then publish a platform event for the requested historical data.
- C.** Use an ESB tool with Request-Reply pattern and then make a real-time Apex callout to the ESB endpoint to fetch and display component related to Case object
- D.** Use an ETL tool with a Batch Data Synchronization pattern to migrate historical data into Salesforce and into a custom object (historical data) related to Case object.

Answer: ([SHOW ANSWER](#))

Explanation

The Data Virtualization pattern allows Salesforce to access external data sources without storing the data in Salesforce. This reduces the data storage and synchronization costs and enables real-time access to the historical data. Salesforce Connect can be used to consume OData endpoints exposed by the ESB tool and display the external objects as related lists or custom components on the Case object

NEW QUESTION: 44

A health care services company maintains a Patient Prescriptions System that has 50+ million records in a secure database. Their customer base and data set growing rapidly. They want to make sure that the following policies are enforced:

1. Identifiable patient prescriptions must exist only in their secure system's database and encrypted at rest.
2. Identifiable patient prescriptions must be made available only to people explicitly authorized in the Patient Prescriptions System assigned nurses and doctors, patient, and people explicitly the patient may authorize.
3. Must be available only to verified and pre-approved people or legal entities.

To enable this, the company provides the following capabilities:

1. One-time use identity tokens for patients, nurses, doctors, and other people that expire within a few minutes.
2. Certificates for legal entities.
- . RESTful services.

The company has a Salesforce Community Cloud portal for patients, nurses, doctors, and other authorized people. A limited number of employees analyze deidentified data in Einstein Analytics.

Which two capabilities should the integration architect require for the Community Cloud portal and Einstein Analytics?

Choose 2 answers

- A.** Identity token data storage
- B.** Bulk load for Einstein Analytics
- C.** Callouts to RESTful services
- D.** Encryption in transit and at rest

Answer: B,C (LEAVE A REPLY)

Explanation

The integration architect should require bulk load for Einstein Analytics and callouts to RESTful services for the Community Cloud portal and Einstein Analytics. Bulk load is a capability that allows loading large volumes of data into Einstein Analytics from external sources, such as the Patient Prescriptions System. This can be done securely by using encryption in transit and at rest, and by masking or removing any identifiable patient information from the data. Callouts to RESTful services are a capability that allows making requests to external web services from Salesforce, such as the Patient Prescriptions System. This can be done securely by using identity tokens or certificates for authentication and authorization, and by using encryption in transit and at rest¹²

References: How to Load Data into Einstein Analytics, Strengthen Your Data's Security with Shield Platform Encryption

NEW QUESTION: 45

Northern Trail Outfitters is planning to create a native employee facing mobile app with the look and feel of Salesforce's Lightning Experience. The mobile app needs to integrate with their Salesforce org.

Which Salesforce API should be used to implement this integration?

- A.** Streaming API

B. REST API

C. Connect REST API

D. User InterfaceAPI

Answer: ([SHOW ANSWER](#))

Explanation

User Interface API should be used to implement this integration. User Interface API is a RESTful API that allows you to create native mobile apps and custom web apps with the look and feel of Salesforce's Lightning Experience. User Interface API provides access to metadata, data, and UI components, such as layouts, actions, picklists, and record types. User Interface API also handles security, performance, and compatibility across different devices and browsers. References: Certification - Integration Architect - Trailhead, [User Interface API Developer Guide]

NEW QUESTION: 46

Northern Trail Outfitters needs to secure an integration with an external Microsoft Azure API Gateway.

What integration security mechanism should be employed?

A. Configure mutual server authentication with two-way SSL using CA issued certificates.

B. Configure a connected app with an authorization endpoint of the API gateway and configure OAuth settings.

C. Use an API only user profile and implement use an external identity provider with federated API access.

D. Implement Salesforce Shield with Encryption at Rest and generate a tenant secret.

Answer: ([SHOW ANSWER](#))

Explanation

The OAuth protocol is a standard way to authorize access to web resources. By configuring a connected app with an authorization endpoint of the API gateway, Salesforce can obtain an access token from the API gateway and use it to invoke the external API securely. This avoids the need to manage certificates or user credentials for authentication2 References: 1: Data Virtualization Pattern 2: OAuth 2.0 Web Server Authentication Flow

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

NEW QUESTION: 47

A company in a heavily regulated industry requires data in legacy systems to be displayed in Salesforce user interfaces (UIs). They are proficient in their cloud-based ETL (extract, transform, load) tools. They expose APIs built on their on-premise middleware to cloud and on-premise applications.

Which two findings about their current state will allow copies of legacy data in Salesforce?

Choose 2 answers

- A. Only on-premise systems are allowed access to legacy systems.
- B. Cloud-based ETL can access Salesforce and supports queues.
- C. On-premise middleware provides APIs to legacy systems data.
- D. Legacy systems can use queues for on-premise integration.

Answer: B,C (LEAVE A REPLY)

Explanation

Option B is a correct finding, because it means that the company can use their cloud-based ETL tools to extract data from the legacy systems via the APIs exposed by the on-premise middleware, transform the data as needed, and load the data into Salesforce using queues. Queues are a way of managing asynchronous operations, such as bulk data loading, by placing them in a queue and executing them when resources are available¹.

Option C is also a correct finding, because it means that the company can expose their legacy data to the cloud via their on-premise middleware APIs. APIs are a way of enabling communication and data exchange between different systems using standard protocols and formats². By providing APIs to access their legacy data, the company can use any cloud-based tool or application that can consume those APIs to retrieve and manipulate the data.

Option A is not a correct finding, because it means that the company cannot access their legacy data from the cloud, which prevents them from using their cloud-based ETL tools or any other cloud-based integration solution. If only on-premise systems are allowed to access the legacy systems, then the company would need to use an on-premise integration tool or application to copy the data to Salesforce.

Option D is not a correct finding, because it does not affect the ability to copy legacy data in Salesforce.

Queues are useful for on-premise integration, but they are not necessary for copying data to Salesforce. The company can use other methods, such as direct API calls or batch processes, to load data into Salesforce without using queues.

References: 1: Queueable Apex | Apex Developer Guide | Salesforce Developers 2: API Basics | SOAP API Developer Guide | Salesforce Developers

NEW QUESTION: 48

A company accepts payment requests 24x7. Once they accept a payment request, their service level agreement (SLA) requires them to make sure each payment request is processed by their Payment System. They track payment requests using a globally unique identifier created at the Data Entry Point. Their simplified flow is as shown in the diagram. They encounter intermittent update errors when two or more processes try to update the same Payment Request record at the same time.

Which two recommendations should an integration architect make to improve their SLA and update conflict handling?

Choose 2 answers

- A.** Middleware should coordinate request delivery and payment processing.
- B.** Data Entry Point and Middleware should automatically retry requests.
- C.** Payment System should process a payment request only once.
- D.** Payment System and Middleware should automatically retry requests.

Answer: A,C (LEAVE A REPLY)

Explanation

Middleware should coordinate request delivery and payment processing, and Payment System should process a payment request only once. This solution ensures that each payment request is delivered to the Payment System in a reliable and consistent manner, and avoids duplicate or conflicting updates to the same Payment Request record.

Middleware can act as a mediator between the Data Entry Point and the Payment System, and implement logic to handle errors, retries, and acknowledgments. Payment System can use the globally unique identifier to check if a payment request has already been processed, and avoid processing it again.

References: Certification - Integration Architect - Trailhead, [Integration Patterns and Practices]

NEW QUESTION: 49

Northern Trail Outfitters (NTO) has an affiliate company that would like immediate notifications of changes to opportunities in the NTO Salesforce instance. The affiliate company has a CometD client available.

Which solution is recommended in order to meet the requirement?

- A.** Create a connected app in the affiliate org and select the "Accept CometD API Requests".
- B.** Implement a polling mechanism in the client that calls the SOAP API get updated method to get the ID values of each updated record.
- C.** Configure External Services to call the subscriber in Apex in the Onchange Trigger event as part of the flow.
- D.** Create a PushTopic update event on the Opportunity Object to allow the subscriber to react to the streaming API.

Answer: D (LEAVE A REPLY)

Explanation

Streaming API is the best option for sending real-time notifications of changes to Salesforce data. Streaming API uses a publish-subscribe model to push relevant data to subscribers without polling. Streaming API supports PushTopic events, which are based on SOQL queries that define the data changes to listen for. The affiliate company can subscribe to a PushTopic event on the NTO Salesforce org and receive notifications whenever the data that matches the query changes. This way, the affiliate company can be informed of any updates to the opportunities in the NTO Salesforce instance.

NEW QUESTION: 50

A developer has been tasked by the integration architect to build a solution based on the Streaming API. The developer has done some research and has found there are different implementations of the events in Salesforce (Push Topic Events, Change Data Capture, Generic Streaming, Platform Events), but is unsure of to proceed with the implementation. The developer asks the system architect for some guidance.

What should the architect consider when making the recommendation?

A. Push Topic Event can define a custom payload. **B** Change Data Capture does not have record access support.

B. Change Data Capture can be published from Apex.

C. Apex triggers can subscribe to Generic Events.

Answer: B (LEAVE A REPLY)

Explanation

Push Topic Events can define a custom payload, which is one of the factors that should be considered when choosing a Streaming API implementation. Push Topic Events are based on SOQL queries that define the data changes to be notified of. You can specify the fields to be returned in the event message by using the SELECT clause in the query. This allows you to customize the payload according to your needs² References: Create a PushTopic

NEW QUESTION: 51

A customer imports data from an external system into Salesforce using Bulk API. These jobs have batch sizes of 2000 and are run in parallel mode. The batch fails frequently with the error "Max CPU time exceeded". A smaller batch size will fix this error.

Which two options should be considered when using a smaller batch size?

Choose 2 answers

A. Smaller batch size may cause record-locking errors.

B. Smaller batch size may increase time required to execute bulk jobs.

C. Smaller batch size may exceed the concurrent API request limits.

D. Smaller batch size can trigger "Too many concurrent batches" error.

Answer: A,B (LEAVE A REPLY)

Explanation

The error "Max CPU time exceeded" occurs when a batch of records takes more than 10 minutes to process in the Bulk API. Using a smaller batch size can reduce the processing

time and avoid this error. However, there are some trade-offs to consider when using a smaller batch size, such as:

Answer A is valid because a smaller batch size may cause record-locking errors if multiple batches try to update the same records or records that are related by a lookup or master-detail relationship. This can result in concurrency issues and data inconsistency¹ Answer B is valid because a smaller batch size may increase the time required to execute bulk jobs, as more batches need to be created and processed sequentially. This can affect the performance and efficiency of the integration¹ Answer C is not valid because a smaller batch size may not exceed the concurrent API request limits, as the Bulk API does not count against the concurrent request limit. The Bulk API has its own limit on the number of batches that can be queued or in progress, which is 5,000 per rolling 24-hour period² Answer D is not valid because a smaller batch size can trigger "Too many concurrent batches" error only if the number of batches exceeds the limit of 5,000 per rolling 24-hour period. This error is not related to the parallel mode of the Bulk API, which allows up to 10 batches to be processed simultaneously

Valid Integration-Architect Dumps shared by Actual4test.com for Helping Passing Integration-Architect Exam! Actual4test.com now offer the **newest Integration-Architect exam dumps**, the Actual4test.com Integration-Architect exam **questions have been updated** and **answers have been corrected** get the **newest** Actual4test.com Integration-Architect dumps with Test Engine here:
https://www.actual4test.com/Integration-Architect_examcollection.html (**130** Q&As Dumps, **30%OFF** Special Discount: **Freepdfdumps**)