S90.09^Q&As

SOA Design & Architecture Lab

Pass SOA S90.09 Exam with 100% Guarantee

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

https://www.pass4itsure.com/s90-09.html

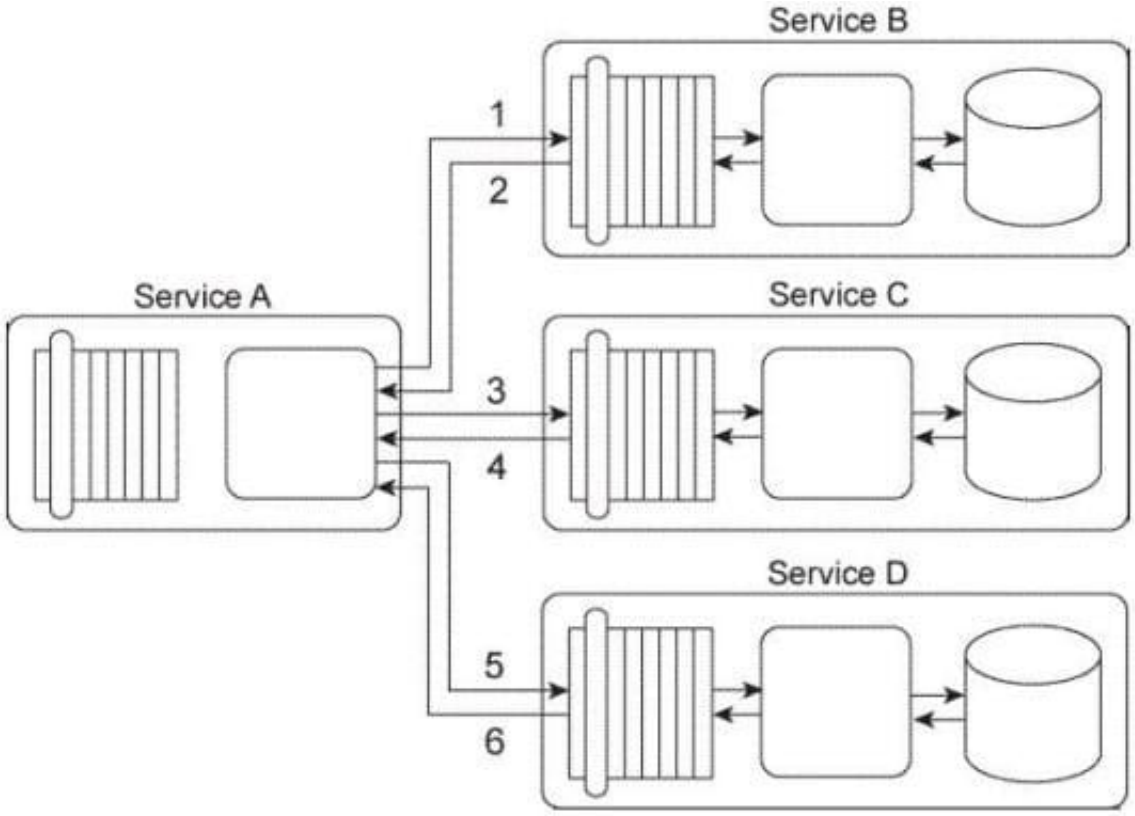
100% Passing Guarantee
100% Money Back Assurance

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

QUESTION 1

Service A is a task service that is required to carry out a series of updates to a set of databases in order to complete a task. To perform the database updates Service A must interact with three other services, each of which provides standardized data access capabilities.

Service A sends its first update request message to Service B (1), which then responds with a message containing a success or failure code (2). Service A then sends its second update request message to Service C (3), which also responds with a message containing a success or failure code (4). Finally, Service A sends a request message to Service D (5), which responds with its own message containing a success or failure code (6).



You\\'ve been asked to change this service composition architecture in order to fulfill a set of new requirements: First, if the database update performed by Service B fails, then it must be logged by Service

A. Secondly, if the database update performed by Service C fails, then a notification e-mail must be sent out to a human administrator. Third, if the database update performed by either Service C or Service D fails, then both of these updates must be reversed so that the respective databases are restored back to their original states. What steps can be taken to fulfill these requirements?

A. Service A is updated to perform a logging routine when Service A receives a response message from Service B containing a failure code. Service A is further updated to send an e-mail notification to a human administrator if Service A receives a response message from Service C containing a failure code. The Atomic Service Transaction pattern is applied so that Services A, C, and D are encompassed in the scope of a transaction that will guarantee that if the database updates performed by either Service C or Service D fails, then both updates will be rolled back.

B. The Compensating Service Transaction pattern is applied to Service B so that it invokes exception handling logic that logs failed database updates before responding with a failure code back to Service

A . Similarly, the Compensating Service Transaction pattern is applied to Service C so that it issues an e-mail notification to a human administrator when a database update fails. The Atomic Service Transaction pattern is applied so that Services A, C, and D are encompassed in the scope of a transaction that will guarantee that if the database updates performed by either Service C or Service D fails, then both updates will be rolled back. The Service Autonomy principle is further applied to Service A to ensure that it remains consistently available to carry out this sequence of actions.

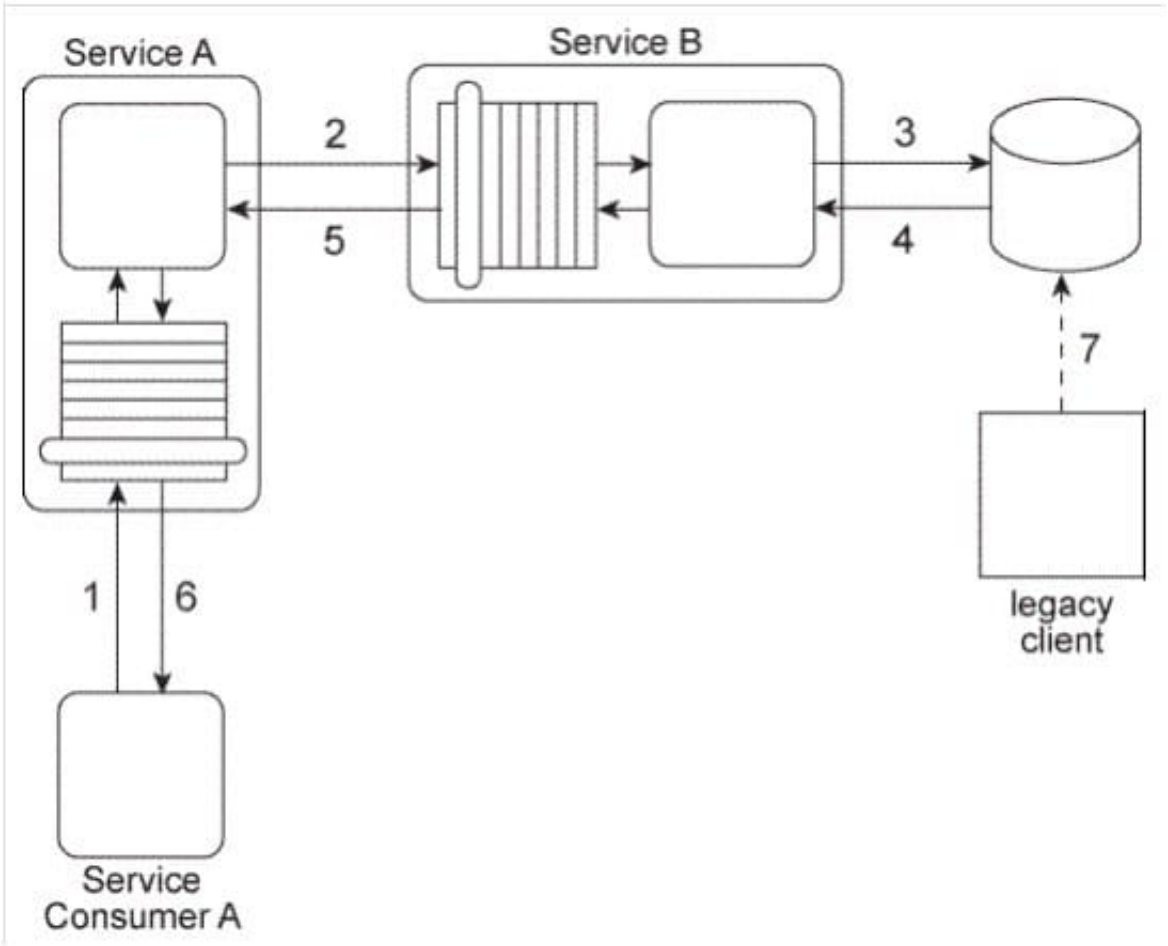
C. The Atomic Service Transaction pattern is applied so that Services A, C, and D are encompassed in the scope of a transaction that will guarantee that if the database updates performed by either Service C or Service D fails, then both updates will be rolled back. The Compensating Service Transaction pattern is then applied to all services so that the scope of the compensating transaction includes the scope of the atomic transaction. The compensating exception logic that is added to Service D automatically invokes Service B to log the failure condition and Service C to issue the e-mail notification to the human administrator. This way, it is guaranteed that the compensating logic is always executed together with the atomic transaction logic.

D. None of the above.

Correct Answer: A

QUESTION 2

Service A has become increasingly difficult to maintain. Its core service logic has become bloated and convoluted because it has been updated numerous times during which additional functionality was added to interact with the database and the legacy system and to support interaction with Service Consumers A and B (via the two service contracts) as well as interaction directly with Service Consumer C.



What steps can be taken to solve these problems and to prevent them from happening again in the future?

A. The Service Facade pattern can be applied to position a Facade component between the core service logic and the implementation resources (the database and the legacy system) and to also position a Facade component between the two service contracts and Service Consumers A and

B. The Official Endpoint pattern can be applied to limit access to Service A to one of its two published service contracts. The Service Loose Coupling principle can be applied so that Service Consumer C does not negatively couple itself directly to the core service logic of Service A . B. The Service Facade pattern can be applied to position a Facade component between the core service logic and the implementation resources (the database and the legacy system) and to position a faade component between the core service logic and the two service contracts. The Contract Centralization pattern can be applied to limit access to Service A to one of its two published service contracts. The Service Abstraction principle can be applied to hide the implementation details of Service A from service consumers.

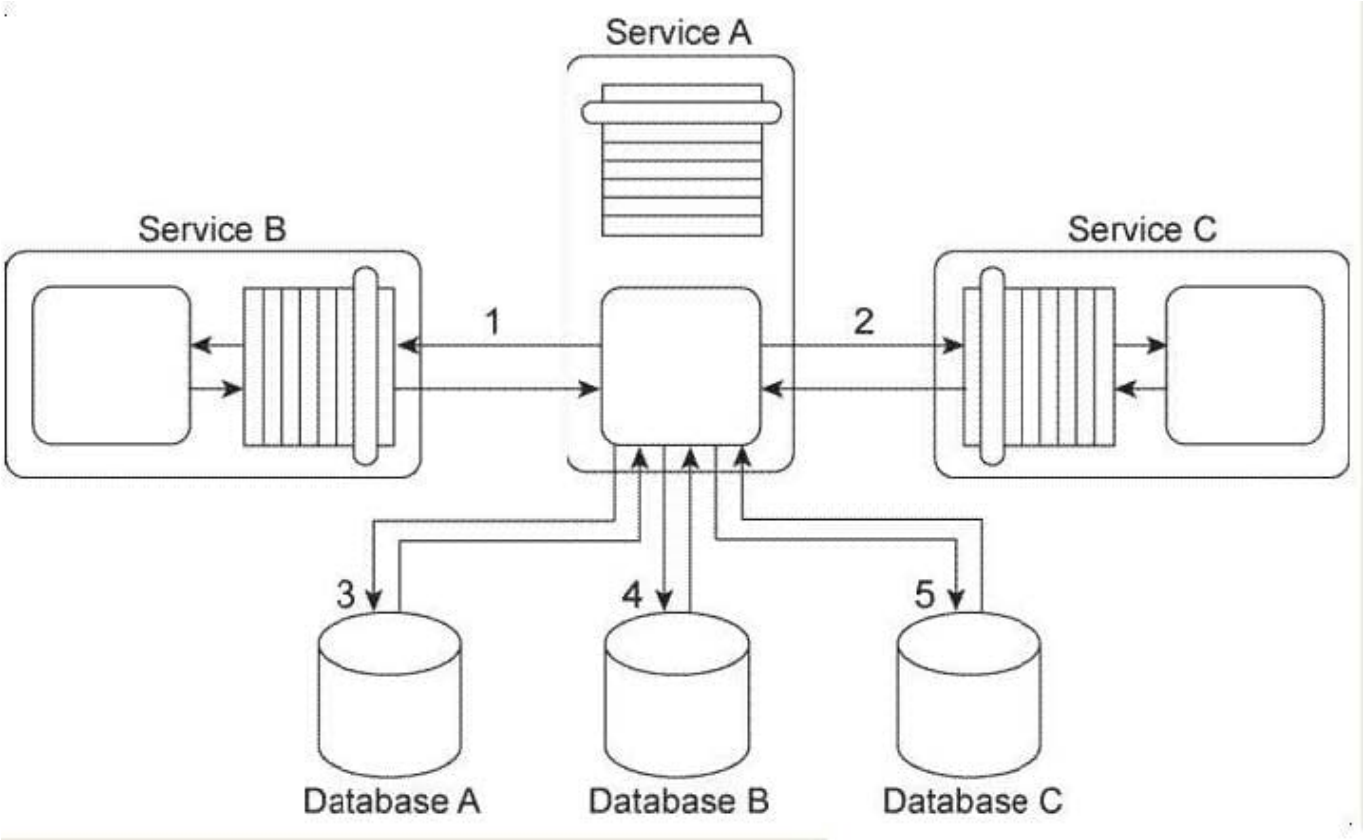
C. The Service Faade pattern can be applied to position a Facade component between the core service logic and the two service contracts. The Contract Centralization pattern can be applied to limit access to Service A to one of its two published service contracts. The Service Loose Coupling principle can be applied so that Service Consumer C does not negatively couple itself directly to the core service logic of Service A .

D. None of the above.

Correct Answer: B

QUESTION 3

You are told that in this service composition architecture, all four services are exchanging invoice-related data in an XML format. The services in Service Inventory A are standardized to use a specific XML schema for invoice data. Design standards were not applied to the service contracts used in Service Inventory B, which means that each service uses a different XML schema for the same kind of data. Database A and Database B can only accept data in the Comma Separated Value (CSV) format and therefore cannot accept XML formatted data. What steps can be taken to enable the planned data exchange between these four services?



A. The Data Model Transformation pattern can be applied so that data model transformation logic is positioned between Service A and Service B, between Service A and Service C, and between Service C and Service D . The Data Format Transformation pattern can be applied so that data format transformation logic is positioned between the Service B logic and Database A and between the Service D logic and Database B.

B. The Data Model Transformation pattern can be applied so that data model transformation logic is positioned between Service A and Service C and between Service C and Service D . The Data Format Transformation pattern can be applied so that data format transformation logic is positioned between

the Service B logic and Database A and between the Service D logic and Database B.

C. The Data Model Transformation pattern can be applied so that data model transformation logic is positioned between Service A and Service C . The Protocol Bridging pattern can be applied so that protocol bridging logic is positioned between Service A and Service B and between the Service C and Service D . The Data Format Transformation pattern can be applied so that data format transformation logic is positioned between the Service B logic and Database A and between the Service D logic and Database B.

D. None of the above.

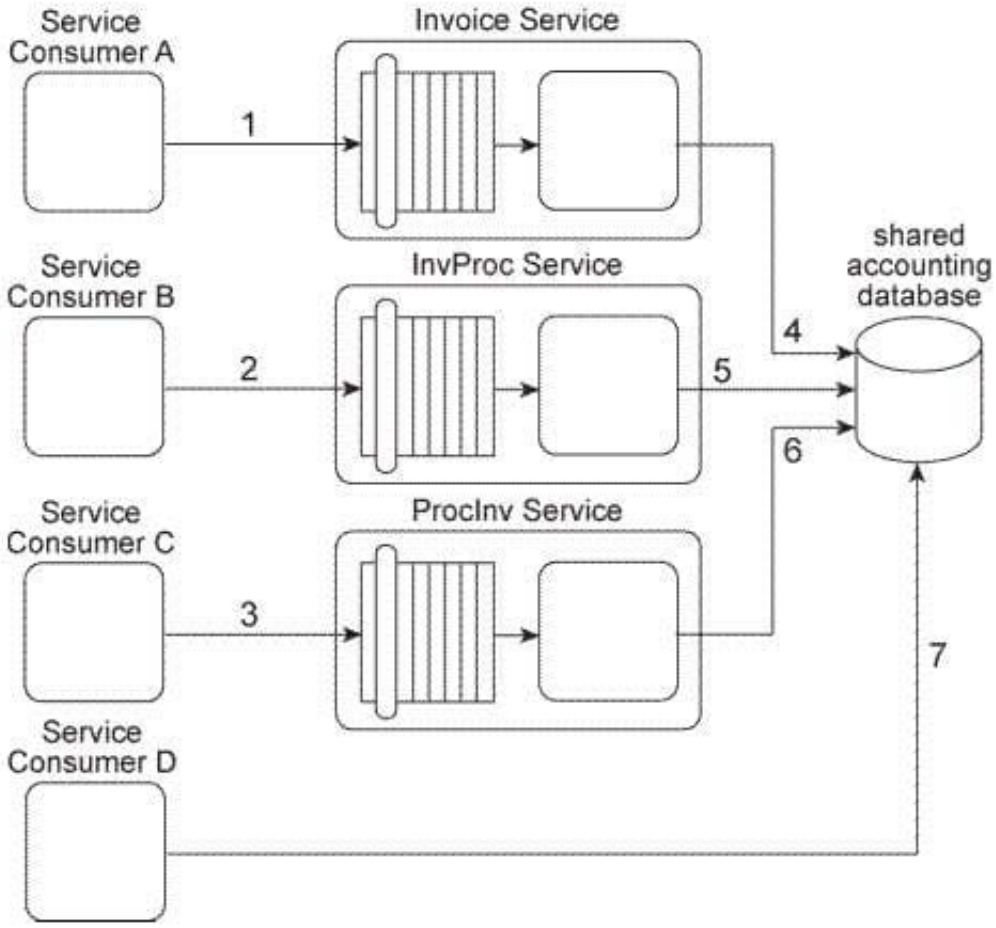
Correct Answer: A

QUESTION 4

Our service inventory contains the following three services that provide invoice-related data access capabilities: Invoice, InvProc, and Proclnv. These services were created at different times by different project teams and were not required to comply to any design standards. Therefore each of these services has a different data model for representing invoice data.

Currently each of these three services has one service consumer: Service Consumer A accesses the Invoice service(1). Service Consumer B (2) accesses the InvProc service, and Service Consumer C (3) accesses the Proclnv service. Each service consumer invokes a data access capability of an invoice-related service, requiring that service to interact with the shared accounting database that is used by all invoice-related services (4, 5, 6).

Additionally, Service Consumer D was designed to access invoice data from the shared accounting database directly (7), (Within the context of this architecture. Service Consumer D is labeled as a service consumer because it is accessing a resource that is related to the illustrated service architectures.)



Assuming that the Invoice service, InvProc service, and ProcInv service are part of the same service inventory, what steps would be required to fully apply the Official Endpoint pattern?

A. One of the invoice-related services needs to be chosen as the official service providing invoice data access capabilities. Service Consumers A, B, and C then need to be redesigned to only access the chosen invoice-related service. Because Service Consumer D does not rely on an invoice-related service, it is not affected by the Official Endpoint pattern and can continue to access the accounting database directly. The Service Abstraction principle can be further applied to hide the existence of the shared accounting database and other implementation details from current and future service consumers.

B. One of the invoice-related services needs to be chosen as the official service providing invoice data access capabilities. Service Consumers A, B, and C then need to be redesigned to only access the chosen invoice-related service. Service Consumer D also needs to be redesigned to not access the shared accounting database directly, but to also perform its data access by interacting with the official invoice-related service. The Service Abstraction principle can be further applied to hide the existence of the shared accounting database and other implementation details from current and future service consumers.

C. Because Service Consumers A, B, and C are already carrying out their data access via published contracts, they are not affected by the Official Endpoint pattern. Service Consumer D needs to be redesigned to not access the shared accounting database directly, but to perform its data access by interacting with the official invoice-related service. The Service Abstraction principle can be further applied to hide the existence of the shared accounting database and other implementation details from current and future service consumers.

D. None of the above.

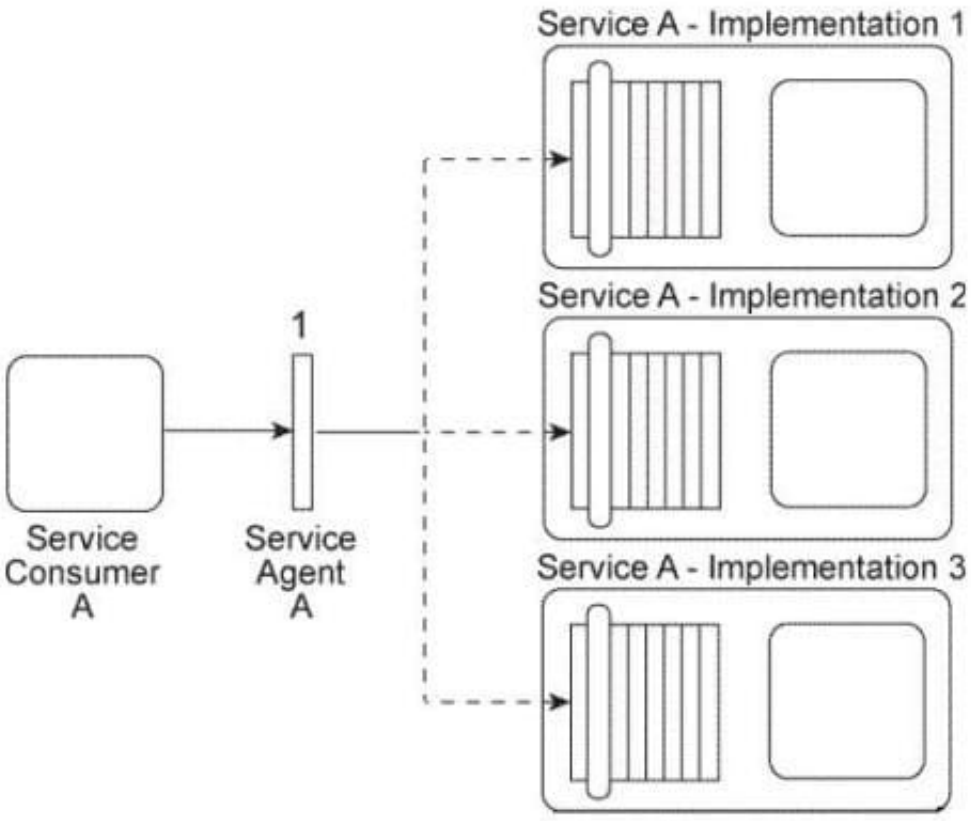
Correct Answer: B

QUESTION 5

Service Consumer A sends a message to Service A. There are currently three duplicate implementations of Service A (Implementation 1, Implementation 2, Implementation 3).

The message sent by Service Consumer A is intercepted by Service Agent A (1), which determines at runtime which implementation of Service A to forward the message to.

All three implementations of Service A reside on the same physical server.



You are told that after Service A was deployed, each of its three implementations was claimed by a different IT department, which means each implementation of Service A has a different owner. You are informed that a new service capability will soon need to be added to Service A . This service capability will introduce new business logic specific to Service A as well as logic required to access a shared database. What steps can be taken to ensure that the service owners will each add the service capability in a consistent manner to their respective implementations of Service A?

A. The Contract Centralization pattern can be applied so that when the new service capability is added, the Service A service contract will become the primary contact point for Service A . This will avoid Service Consumer A or any other potential service consumer from being designed to access the shared database directly. The Service Abstraction principle can be applied to further hide the implementation details so that Service Consumer A and other service consumers are unaware of the fact that the shared database is being accessed.

B. The Legacy Wrapper pattern can be applied to establish a new wrapper utility service that will provide standardized data access service capabilities for the shared database. This will avoid Service A from having to access the shared database directly and will further support the application of the Service Loose Coupling principle between Service A and the new utility service. By abstracting the data access logic into the wrapper service, there is no need to add the new service capability to each implementation of Service A .

C. The Standardized Service Contract principle is applied to ensure that the new service capability is consistently added to the service contract of each implementation and that it extends the existing Service A service contract in a manner that is compliant with current design standards. The Service Loose Coupling principle is applied to ensure that the new service capability remains decoupled from the underlying logic and implementation so that Service Consumer A does not become indirectly coupled to any new logic or to the shared database.

D. None of the above.

Correct Answer: C

QUESTION 6

Currently, due to the increasing amount of concurrent access by service consumers, the runtime performance of both the Client and Vendor services has worsened and has therefore reduced their effectiveness as service composition members. Additionally, a review of the logic of both services has revealed that some of the business rules used by the Client and Vendor services are actually the same. What steps can be taken to improve performance and reduce redundant business rule logic?

A. The Rules Centralization pattern can be applied by extracting the business rule logic from the Client and Vendor services and placing it into a new Rules service, thereby reducing the redundancy of business rules logic. The Redundant Implementation pattern can then be applied to establish a scalable Rules service that is capable of supporting concurrent access from many service consumers.

B. The Redundant Implementation pattern can be applied to the Client and Vendor services, thereby establishing duplicate service implementations that can be accessed when a service reaches its runtime usage threshold. The Intermediate Routing pattern can be further applied to provide load balancing logic that can, at runtime, determine which of the redundant service implementations is the least busy for a given service consumer request.

C. The Rules Centralization pattern can be applied to isolate business rules logic into a central and reusable Rules service. Additionally, the Service Abstraction principle can be applied to hide the implementation details of new the Rules service.

D. None of the above.

Correct Answer: A

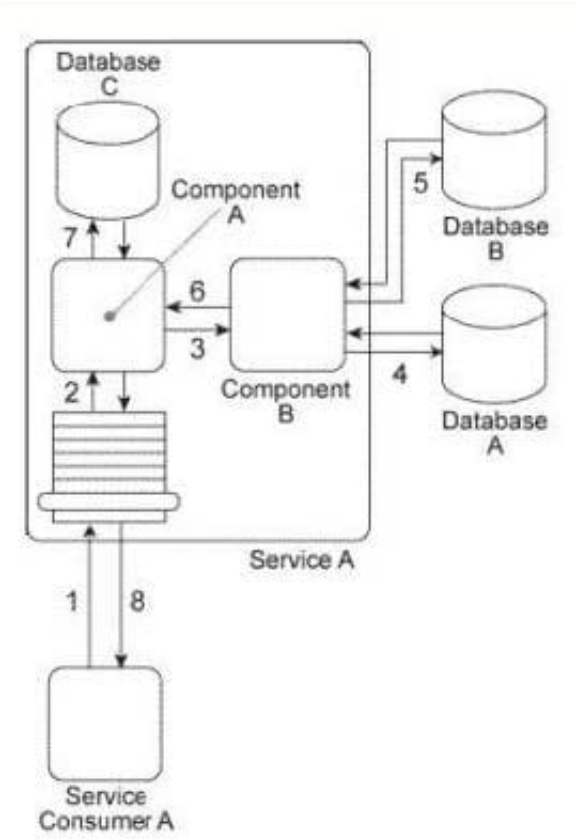
QUESTION 7

Service Consumer A sends Service A a message containing a business document (1). The business document is received by Component A, which keeps the business document in memory and forwards a copy to Component B (3). Component B first writes portions of the business document to Database A (4).

Component B writes the entire business document to Database B and then uses some of the data values from the business document as query parameters to retrieve new data from Database B (5).

Next, Component B returns the new data back to Component A (6), which merges it together with the original business document it has been keeping in memory and then writes the combined data to Database C (7). The Service A service capability invoked by Service Consumer A requires a synchronous request-response data exchange. Therefore, based on the outcome of the last database update, Service A returns a message with a success or failure code back to Service Consumer A (8).

Databases A and B are shared and Database C is dedicated to the Service A service architecture.



There are several problems with this architecture: The business document that Component A is required to keep in memory (while it waits for Component B to complete its processing) can be very large. Especially when Service A is concurrently invoked by multiple service consumers, the amount of runtime resources it uses to keep this data in memory can decrease the overall performance of all service instances. Additionally, because Database A is a shared database that sometimes takes a long time to respond to Component B, Service A can take a long time to respond back to Service Consumer A . Currently, Service Consumer A will wait for a response for up to 30 seconds after which it will assume the request to Service A has failed and any subsequent response messages from Service A will be rejected. What steps can be taken to solve these problems?

A. The Service Statelessness principle can be applied together with the State Repository pattern in order to extend Database C so that it also becomes a state database allowing Component A to temporarily defer the business document data while it waits for a response from Component B. The Service Autonomy principle is applied together with

the Legacy Wrapper pattern to isolate Database A so that it is encapsulated by a separate wrapper utility service. The Compensating Service Transaction pattern is applied so that if the response time of Service A exceeds 30 seconds, a notification is sent to a human administrator to raise awareness of the fact that the eventual response of Service A will be rejected by Service Consumer A.

B. The Service Statelessness principle can be applied together with the State Repository pattern in order to establish a state database that Component A can defer the business document data to while it waits for a response from Component B. The Service Autonomy principle can be applied together with the Service Data Replication pattern to establish a dedicated replicated database for Component B to access instead of the shared Database

C. The Asynchronous Queuing pattern can be applied to establish a messaging queue between Service Consumer A and Service A so that Service Consumer A does not need to remain stateful while it waits for a response from Service A .

D. The Service Statelessness principle can be applied together with the State Repository pattern in order to establish a state database that Component A can defer the business document data to while it waits for a response from Component B. The Service Autonomy principle can be applied together with Service Abstraction principle, the Legacy Wrapper pattern, and the Service Facade pattern in order to isolate Database A so that it is encapsulated by a separate wrapper utility service and to hide the Database A implementation from Service A and to position a Facade component between Component B and the new wrapper service. This Facade component will be responsible for compensating the unpredictable behavior of Database A.

E. None of the above.

Correct Answer: B

QUESTION 8

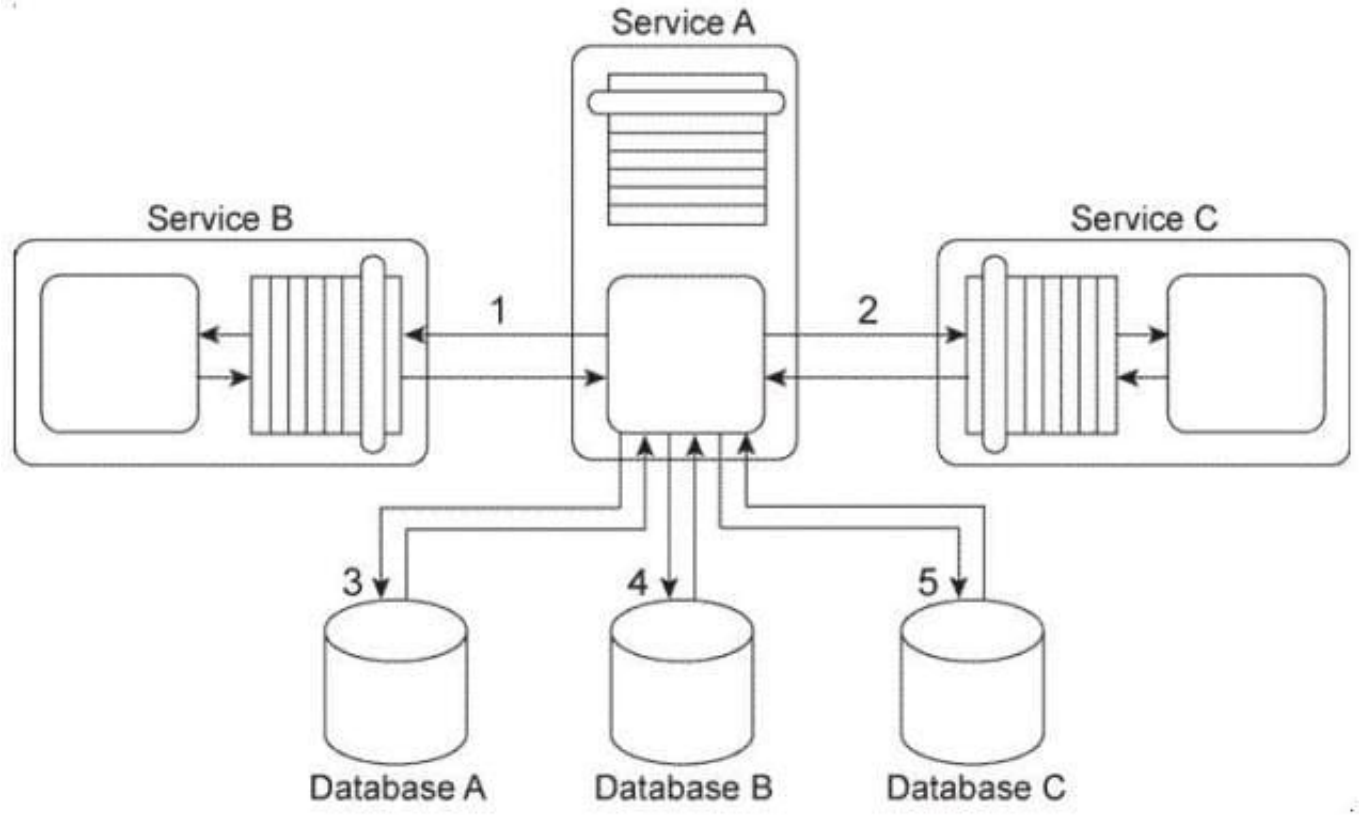
Service A is an entity service that provides a set of generic and reusable service capabilities. In order to carry out the functionality of any one of its service capabilities, Service A is required to compose Service B

(1) and Service C (2) and Service A is required to access Database A (3), Database B (4), and Database C (5). These three databases are shared by other applications within the IT enterprise.

All of service capabilities provided by Service A are synchronous, which means that for each request a service consumer makes. Service A is required to issue a response message after all of the processing has completed.

Depending on the nature of the service consumer request, Service A may be required to hold data it receives in memory until its underlying processing completes. This includes data it may receive from either Service A or Service B or from any of the three shared databases.

Service A is one of many entity services that reside in a highly normalized service inventory. Because Service A provides agnostic logic, it is heavily reused and is currently part of many service compositions.



You are told that Service A has recently become unstable and unreliable and several of the service consumers that access it have had to raise runtime exceptions due to these problems. What steps can be taken to solve these problems without compromising the normalization of the service inventory?

A. The Service Autonomy principle can be applied to increase the physical isolation of Service A and to reduce dependencies Service A has on external resources. In support of this, the Service Data Replication pattern can be applied in order to establish a dedicated database that contains replicated data from shared Databases A, B, and C . Furthermore, the Redundant Implementation pattern can be applied so that the logic Service A requires from Services B and C can be redundantly placed inside of Service A . This way, Service A avoids having to separately compose Services B and C

B. The Service Statelessness principle can be applied with the help of the State Repository pattern in order to establish a state database that Service A can use to defer state data it may be required to hold for extended periods. The Service Autonomy principle can also be applied in order to increase the physical isolation of Service A and to reduce dependencies Service A has on external resources. In support of this, the Service Data Replication pattern can be applied in order to establish a dedicated database that contains replicated data from shared Databases A, B, and C.

C. The Service Loose Coupling and Standardized Service Contract principles can be applied by introducing a separate utility service that provides centralized data access to the Databases A, B, and C, and exposes a standardized service contract that can be used by Service A . This will prevent Service A from direct dependencies on the shared databases in case any of them are replaced in the future. By following this approach, the Legacy Wrapper pattern is effectively applied via the introduction of the new utility service.

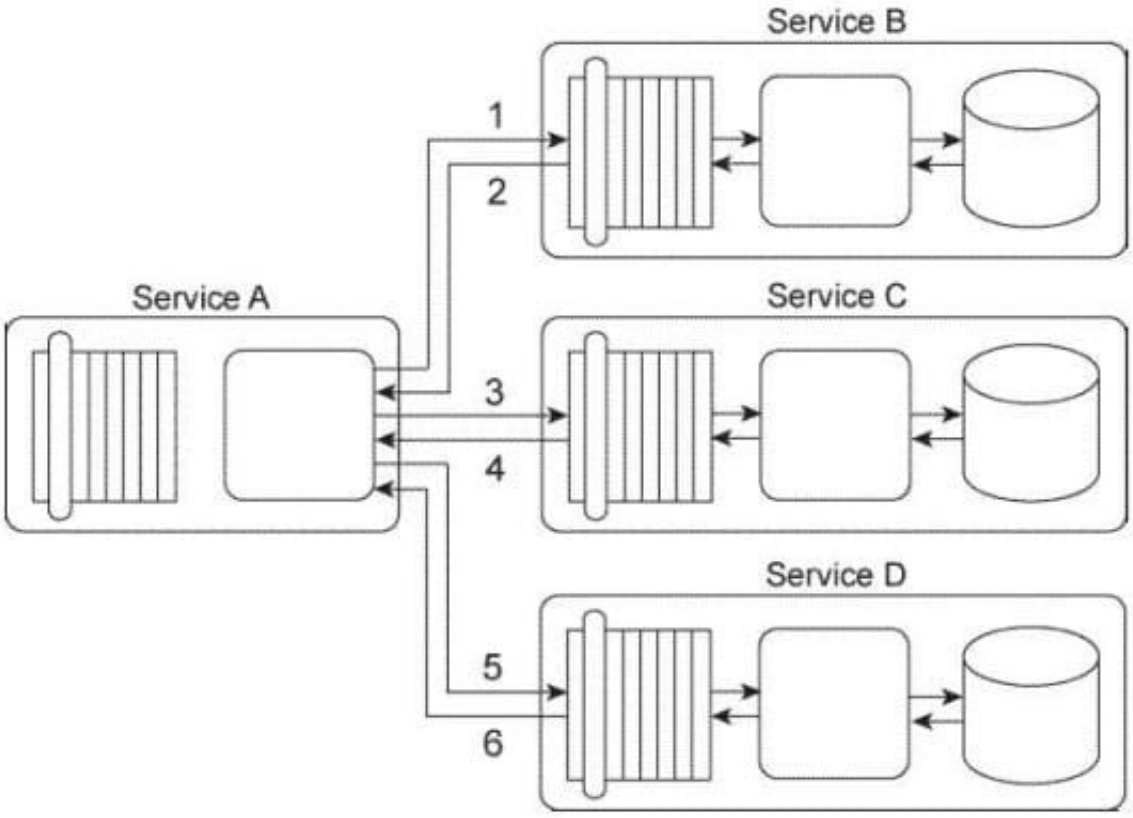
D. None of the above.

Correct Answer: B

QUESTION 9

Service A is a task service that is required to carry out a series of updates to a set of databases in order to complete a task. To perform the database updates Service A must interact with three other services, each of which provides standardized data access capabilities.

Service A sends its first update request message to Service B (1), which then responds with a message containing a success or failure code (2). Service A then sends its second update request message to Service C (3), which also responds with a message containing a success or failure code (4). Finally, Service A sends a request message to Service D (5), which responds with its own message containing a success or failure code (6).



You\\'ve been given a requirement that all database updates must either be completed successfully or not at all. This means that if any of the three response messages received by Service A contain a failure code, all of the updates carried out until that point must be reversed. Note that if Service A does not receive a response message back from Services B, C, or D, it must assume that a failure has occurred. How can this service composition architecture be changed to fulfill these requirements?

A. The Reliable Messaging pattern can be applied to guarantee the delivery of positive or negative acknowledgements. This way, Service A will always be informed of whether a failure condition has occurred with any of the database updates performed by Services B, C, and D. Furthermore, the Service Loose Coupling principle can be applied to ensure that the request and response messages exchanged by the services do not contain any implementation details that would indirectly couple Service A to any of the databases.

B. The Atomic Service Transaction pattern can be applied individually to Services B, C, and D so that each of these services performs its own database update within the scope of an atomic transaction. If anyone update fails, that change can be rolled back on that database. Furthermore, the Service Loose Coupling principle can be applied to ensure that Service A is kept out of the scope of the atomic transaction so that it is not negatively coupled to the proprietary database technologies that are required to enable the atomic transaction functionality.

C. The Compensating Service Transaction can be applied to Service A so that when any one response message containing a failure code is received by Service A, it can invoke exception handling logic that will log the failed database

updates. The Service Loose Coupling principle can be further applied to ensure that Services B, C, or D are not indirectly coupled to the exception handling logic, especially if Service A requires additional access to Services B, C, or D in order to collect more information for logging purposes.

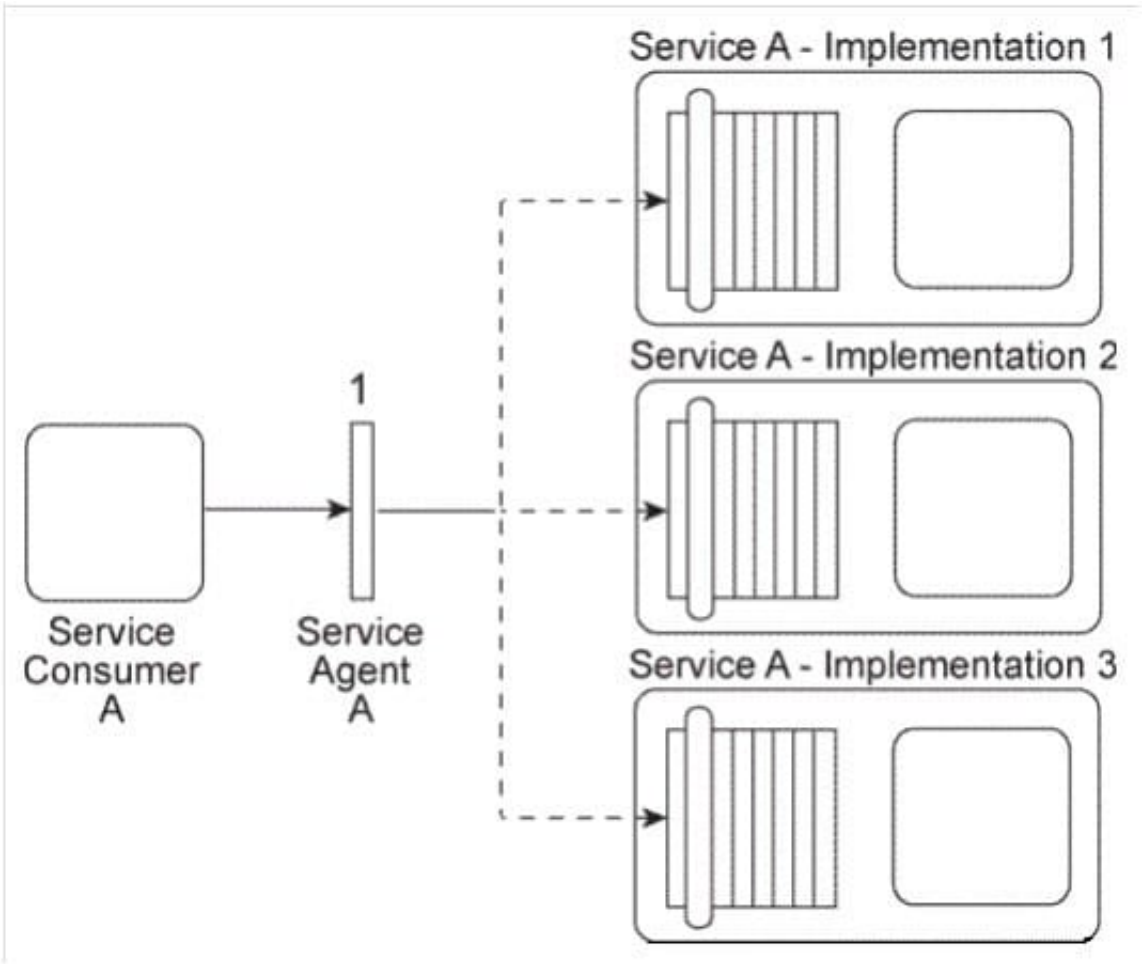
D. None of the above.

Correct Answer: D

QUESTION 10

It has been confirmed that Policy A and Policy B are, in fact, the same policy and that the security credential check performed by Service Agent B also needs to be carried out on messages sent to Service

B .



How can this service composition architecture be changed to reduce the redundancy of policy content and fulfill the new security requirement?

A. The Policy Centralization pattern can be applied so that Policy A and Policy B are combined into the same policy. The policy enforcement logic is removed from Service Agent C and Service Agent A is then used to enforce the policy for messages sent to Service A and Service B . Service Agent B can be used to perform the security credential check for Service A and Service B .

B. The Policy Centralization pattern can be applied so that Policy A and Policy B are combined into the same policy.

The Service Agent pattern is then applied to introduce a new service agent (called Service Agent D) which carries out the validation and enforcement of Policy A and Policy B. Service Agent B can be moved so that it performs the security credential check for Service B, but not for Service A .

C. The Policy Centralization pattern can be applied so that Service Agent A is changed to enforce the policy for messages sent to Service A and Service B and to perform the security credential check for Service A and Service B .

D. None of the above.

Correct Answer: A