S90.09 Q&As

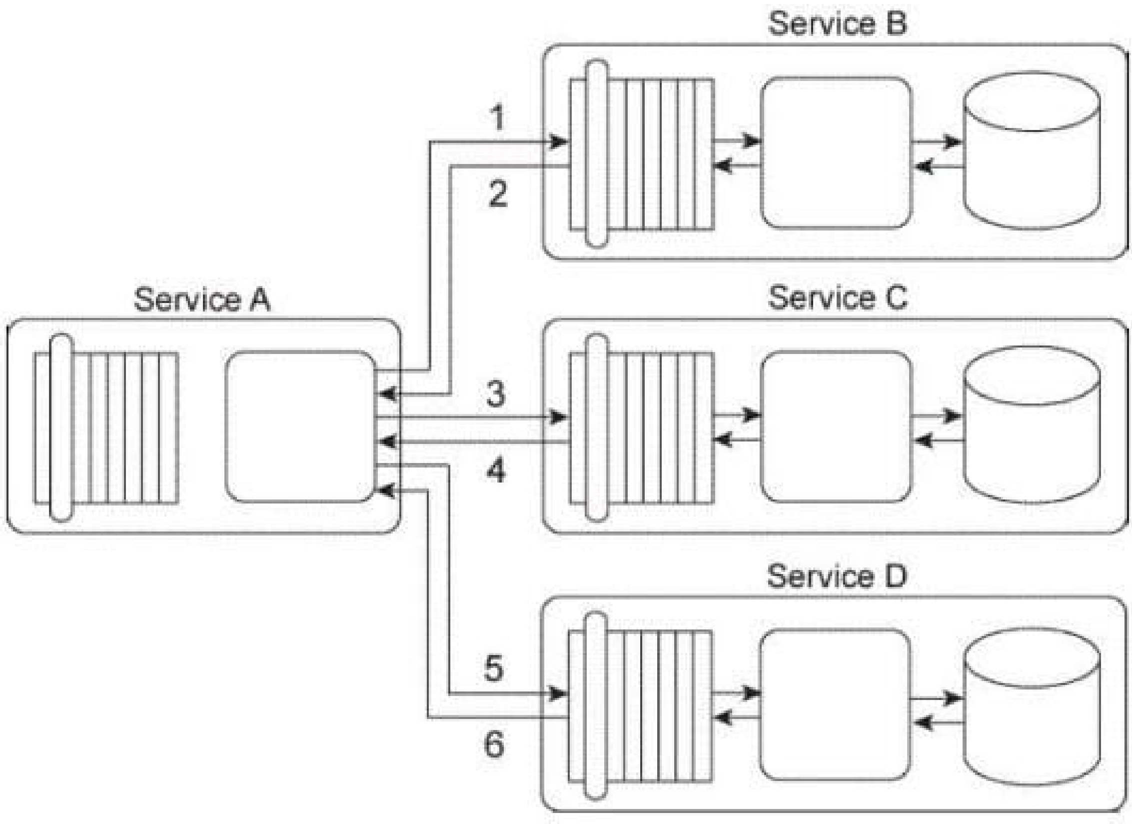
SOA Design & Architecture Lab

Pass SOA S90.09 Exam with 100% Guarantee

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.

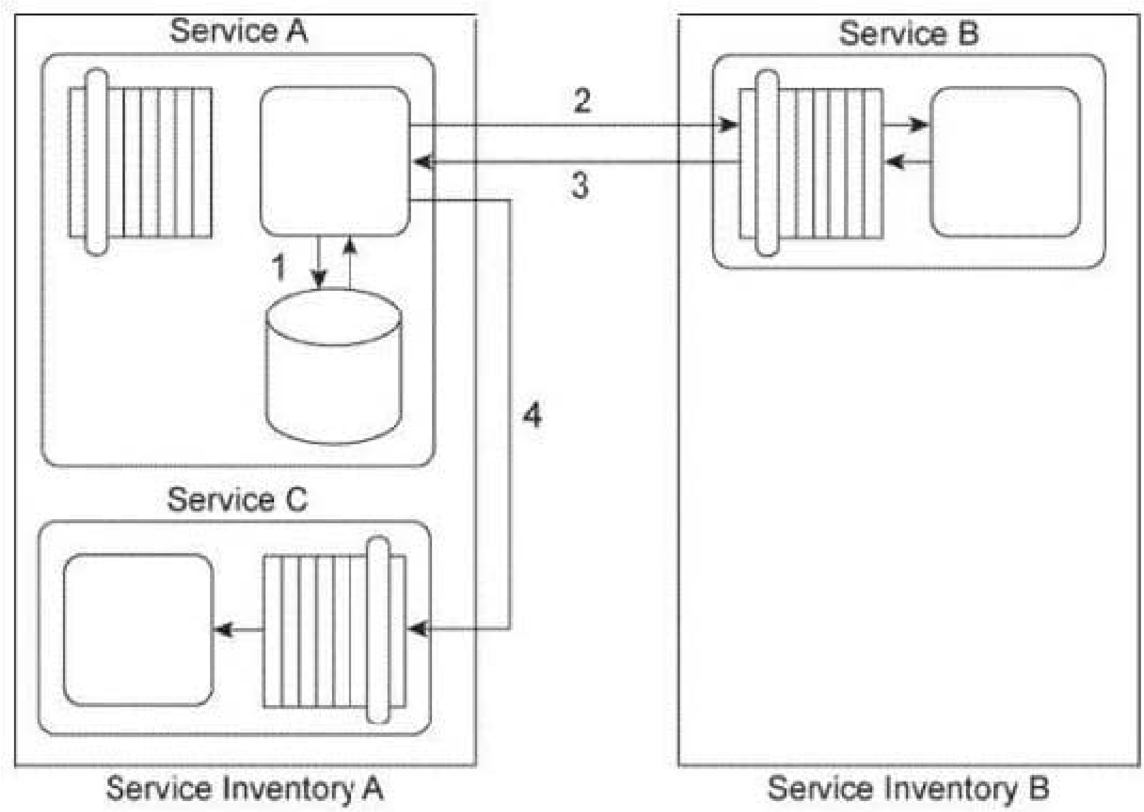
D. None of the above.

Correct Answer: A

QUESTION 2

Service A is a task service that sends Service B a message (2) requesting that Service B return data back to Service A in a response message (3). Depending on the response received. Service A may be required to send a message to Service C (4) for which it requires no response. Before it contacts Service B, Service A must first retrieve a list of code values from its own database (1) and then place this data into its own memory. If it turns out that it must send a message to Service C, then Service A must combine the data it receives from Service B with the data from the code value list in order to create the message it sends to Service C. If Service A is not required to invoke Service C, it can complete its task by discarding the code values.

Service A and Service C reside in Service Inventory A. Service B resides in Service Inventory B.



You are told that the services in Service Inventory A are all SOAP-based Web services designed to exchange SOAP 1.1 messages and the services in Service Inventory B are SOAP-based Web services designed to exchange SOAP 1.2 messages. Therefore, Service A and Service B cannot currently communicate. Furthermore, you are told that Service B needs to access a shared database in order to retrieve the data required by Service A. The response time of the database can sometimes be lengthy, which would cause Service A to consume too much resources while it is waiting and keeping the code values in memory. How can this service composition architecture be changed to avoid these problems?

A. The Protocol Bridging pattern can be applied by establishing an intermediate processing layer between Service A and Service B that can convert SOAP 1.1 messages to SOAP 1.2 messages and vice versa. The Service Data Replication pattern can be applied to Service B so that it is given a dedicated database with its own copy of the data it needs to access. The Service Normalization pattern can then be applied to ensure that the data within the replicated database is normalized with the shared database it is receiving replicated data from.

B. The Protocol Bridging pattern can be applied by establishing an intermediate processing layer between Service A and Service B that can convert SOAP 1.1 messages to SOAP 1.2 messages and vice versa. The Service Statelessness principle can be applied with the help of the State Repository pattern so that Service A can write the code value data to a state database while it is waiting for Service B to respond.

C. The Protocol Bridging pattern can be applied by establishing an intermediate processing layer between Service A and Service B that can convert SOAP 1.1 messages to SOAP 1.2 messages and vice versa. The Intermediate Routing pattern can be applied to dynamically determine whether Service A should send a message to Service C. The Service Autonomy principle can be applied to Service A to further increase its behavioral predictability by reducing the amount of memory it is required to consume.

D. None of the above.

Correct Answer: B

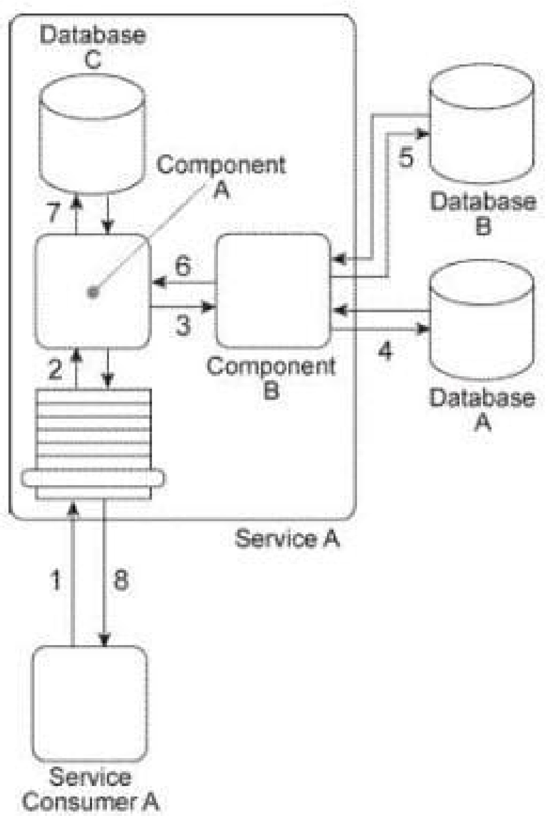
QUESTION 3

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: First, the response time of Database A is often poor, resulting in Component B taking too much time to provide a response to Component A. This results in Component A consuming too many runtime resources while it holds the business document in memory and it also causes unreasonable delays in responding to Service Consumer A. Additionally, Database B is being replaced with a different database product that supports a proprietary file format. This will disable the current interaction between Component B and the new Database B. What steps can be taken to solve these problems?

A. The State Repository pattern is applied so that Component A can defer the business document data to a state database while it waits for a response from Component B. The Service Data Replication pattern is applied so that Component B can interact with a database that is replicated from the shared Database A. This will improve performance and reliability that will affect both Component A and Service Consumer A. Finally, the Legacy Wrapper pattern is applied so that Database B is wrapped in a standardized contract. This will establish a new wrapper utility service that will allow Database B to be replaced with a different database product without affecting Service A . Furthermore, the Data Format

Transformation pattern can be applied within the new wrapper utility service to enable it to convert to

and from the new proprietary file format.

B. The State Repository pattern is applied so that Component A can defer the business document data to a state database while it waits for a response from Component B. The Asynchronous Queuing pattern can be applied so that a messaging queue is established between Service Consumer A and Service A, thereby guaranteeing delivery and avoiding Service Consumer A from being tied up too long waiting for Service A to respond. Finally, the Data Format Transformation pattern can be applied to enable Component B to convert to and from the new proprietary file format introduced by the database product that is replacing Database B.

C. The Legacy Wrapper pattern is applied so that Database B is wrapped in a standardized contract. This will establish a new wrapper utility service that will allow Database B to be replaced with a different database product without affecting Service A . The Data Format Transformation pattern can be applied within the new wrapper utility service to enable it to convert to and from the new proprietary file format. The Service Data Replication pattern is applied so that Component B can interact with a database that is replicated from the shared Database B, regardless of what database product is used to replace Database B. The Service Abstraction principle can be further applied to hide the implementation details, including the changes mentioned in this solution, from Service Consumer A.

D. None of the above.

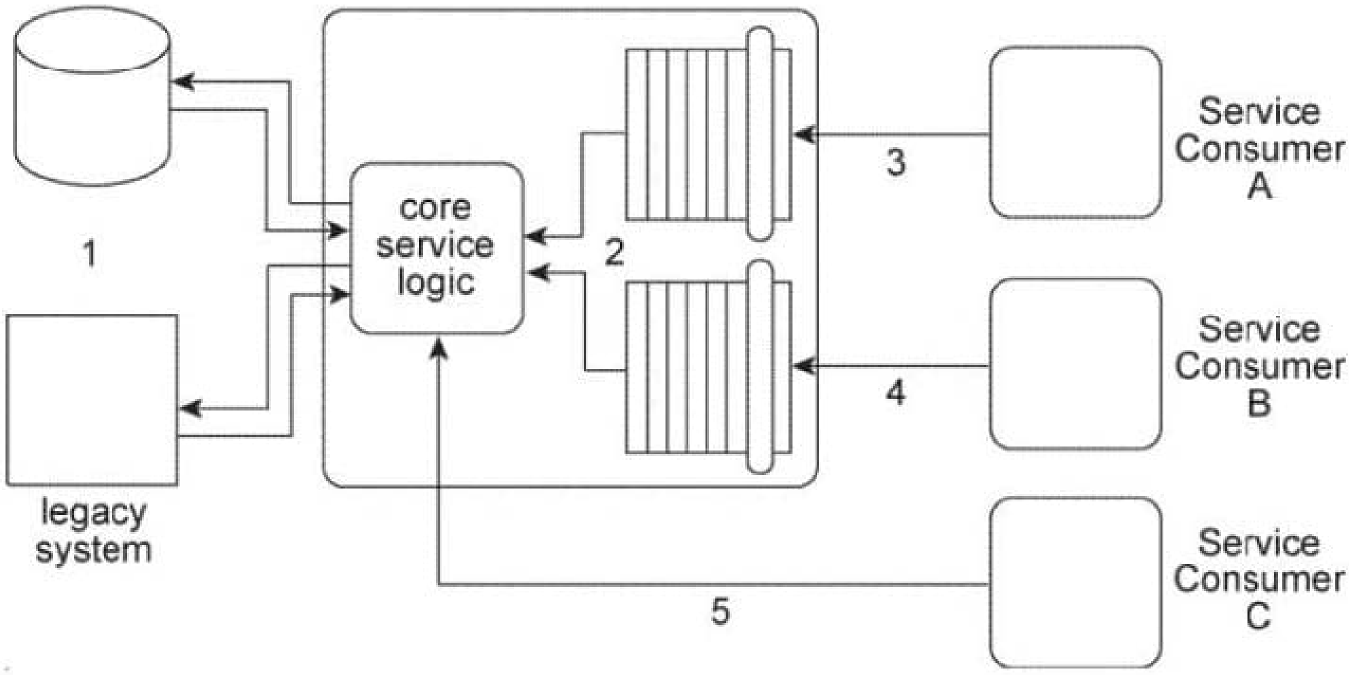
Correct Answer: A

QUESTION 4

The architecture for Service A displayed in the Figure shows how the core logic of Service A has expanded over time to connect to a database and a proprietary legacy system (1) and to support two separate service contracts (2) that are accessed by different service consumers.

The service contracts are fully decoupled from the service logic. The service logic is therefore coupled to the service contracts and to the underlying implementation resources (the database and the legacy system).

Service A currently has three service consumers. Service Consumer A and Service Consumer B access Service A\\'s two service contracts (3, 4). Service Consumer C bypasses the service contracts and accesses the service logic directly (5).



You are told that the database and legacy system that are currently being used by Service A are being replaced with different products. The two service contracts are completely decoupled from the core service logic, but there is still a concern that the introduction of the new products will cause the core service logic to behave differently than before. What steps can be taken to change the Service A architecture in preparation for the introduction of the new products so that the impact on Service Consumers A, B, and C is minimized?

A. The Service Abstraction principle can be applied to hide the implementation details from the core service logic of Service A, thereby shielding this logic from changes to the implementation. In support of this, the Service Facade pattern can be applied to position Facade components between the core service logic and Service Consumers A and B. These Facade components will be designed to regulate the behavior of Service A . The Contract Centralization pattern can be applied to force Service Consumer C to access Service A via one of its existing service contracts.

B. A third service contract can be added together with the application of the Contract Centralization pattern. This will force Service Consumer C to access Service A via the new service contract. The Service Facade pattern can be applied to position a Facade component between the new service contract and Service Consumer C in order to regulate the behavior of Service A . The Service Abstraction principle can be applied to hide the implementation details of Service A so that no future

service consumers are designed to access any of Service A\\'s underlying resources directly.

C. The Service Facade pattern can be applied to position Facade components between the core service logic and the two service contracts. These Facade components will be designed to regulate the behavior of Service A . The Contract Centralization pattern can also be applied to force Service Consumer C to access Service A via one of its existing service contracts.

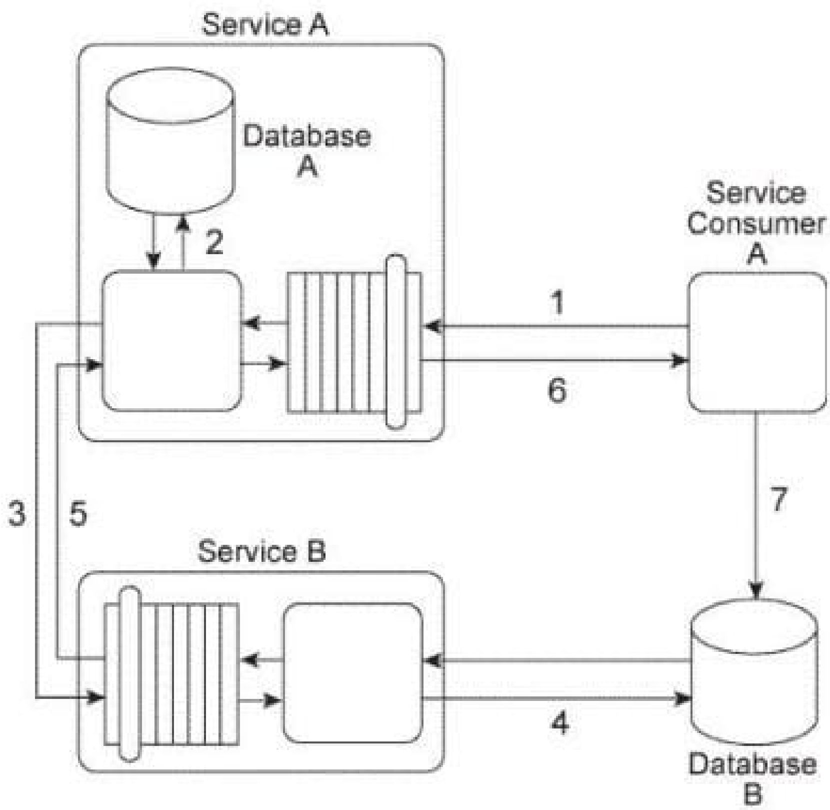
D. None of the above.

Correct Answer: C

QUESTION 5

Service Consumer A sends a message with a business document to Service A (1), which writes the business document to Database A (2). Service A then forwards the business document to Service B (3), which writes the business document to Database B (4).

Service B then responds to Service A with a message containing a failure or success code (5) after which Service A responds to Service Consumer A with a message containing a failure or success code (6). Upon receiving the message, Service Consumer A updates a log table in Database B (7). The log entry is comprised of the entire business document. Database A is dedicated to the Service A service architecture and Database B is a shared database.



You are told that the database updates performed by Service A and Service B must be either both successful or they cannot happen at all. The database update performed by Service Consumer A must happen after it is given the outcome of the database updates performed by Service A and Service B . Given that Service Consumer A must also update Database B as part of this service composition architecture, how is it possible to fulfill these requirements?

A. The State Repository pattern can be applied so that Service A writes the business document data to a separate state database until it receives a response message from Service B . If the response message contains a success code, Service A writes the business document to Database A. If the response contains a failure code, Service A discards the data that was written to the state database.

B. The Service Data Replication pattern can be applied to Service Consumer A and Service B so that separate dedicated databases can be established allowing Service Consumer A to make updates independently of Service B . Service A is simply redesigned to not write the business document to Database A until after it receives a message containing a success code from Service B.

C. The Atomic Service Transaction pattern can be applied to encompass Service A, Service B and Service Consumer A. This will guarantee that all of the actions performed by the service composition participants will either be successful or will be rolled back if anyone is not successful.

D. None of the above.

Correct Answer: D