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

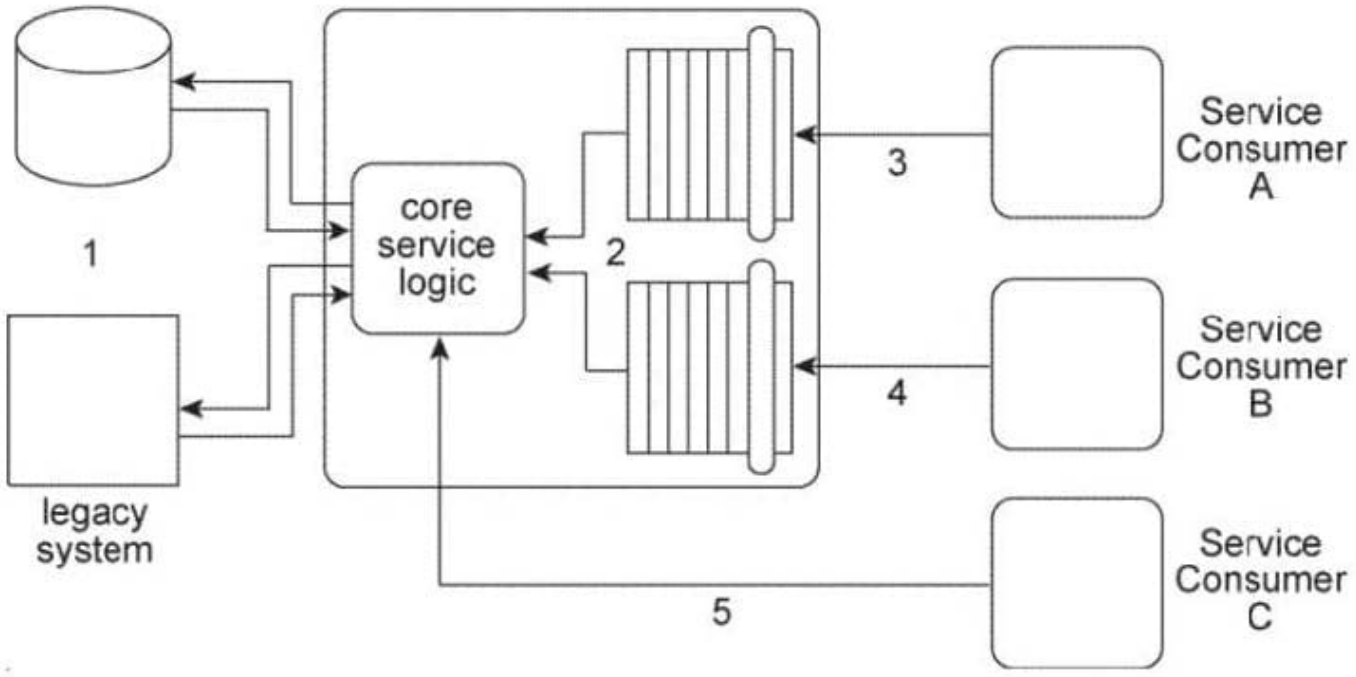
Following Questions and Answers are all new published by SOA Official Exam Center

QUESTION 1

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.

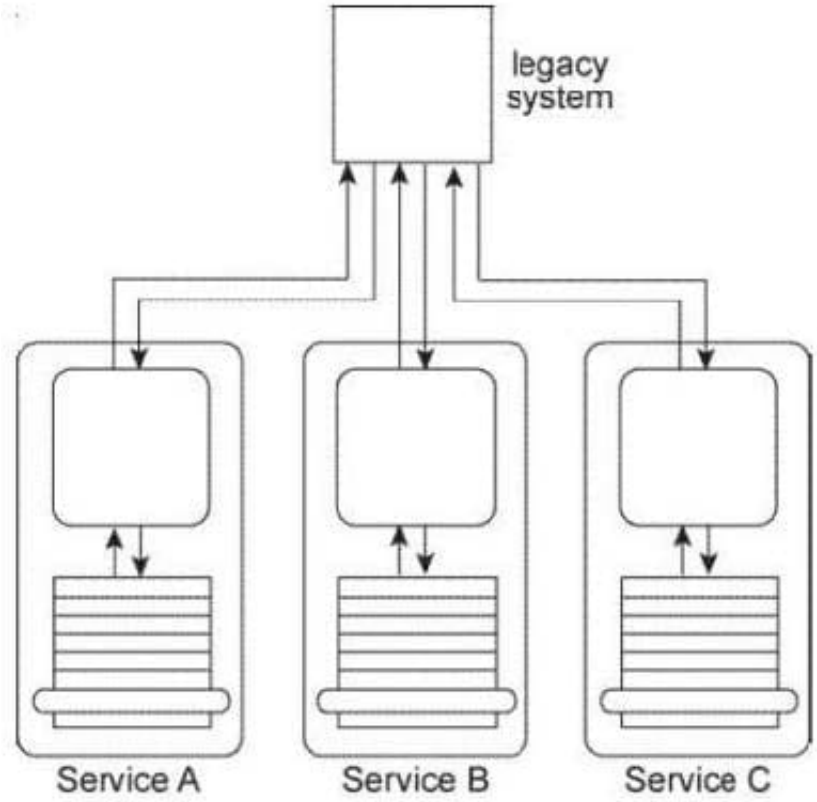
C. The Service Facade pattern can be applied to position Facade components between the core service logic and the two service contracts. These Facade components will be designed to regulate the behavior of Service A . The Contract Centralization pattern can also be applied to force Service Consumer C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

D. None of the above.

Correct Answer: C

QUESTION 2

Service A. Service B. and Service C are each designed to access the same shared legacy system. The service contracts for Service A, Service B, and Service C are standardized and decoupled from the underlying service logic. Service A and Service B are agnostic services that are frequently reused by different service compositions. Service C is a non- agnostic task service that requires access to the legacy system in order to retrieve business rules required for the service to make runtime decisions that determine its service composition logic. The legacy system uses a proprietary file format that Services A, B, and C need to convert to and from.



You are told that additional services need to be created, all of which need access to the legacy system. You are also told that the legacy system may be replaced in the near future. What steps can be taken to ensure that the replacement of the legacy system has a minimal impact on Services A, B, and C and any future services that are designed to rely upon it?

A. The Legacy Wrapper pattern can be applied together with the Standardized Service Contract principle to position a standardized service contract between the legacy system and any services that require access to it. This effectively establishes a new utility service dedicated to the encapsulation of the legacy system. When the legacy system is

replaced, the utility service can keep its standardized service contract. To build the utility service, the Data Format Transformation pattern is applied to convert between the proprietary legacy system file format and the XML format used in the standardized service contract.

B. The Legacy Wrapper pattern can be applied together with the Official Endpoint pattern so that the Service A service contract is positioned as the sole access point for the legacy system. The Data Format Transformation pattern is applied to enable the conversion between the proprietary legacy system file format and the XML format used in the Service A service contract. Finally, the Contract Centralization pattern is applied so that Service A is forced to only access the legacy system via its

published standardized service contract.

C. The Legacy Wrapper pattern can be applied together with the Data Format Transformation pattern and the Standardized Service Contract principle in order to establish an intermediate layer of standardized transformation logic that is positioned between the legacy system and Services A, B, and C. This way, if the legacy system is replaced, the services will not be affected because of the abstraction established by the standardized transformation layer.

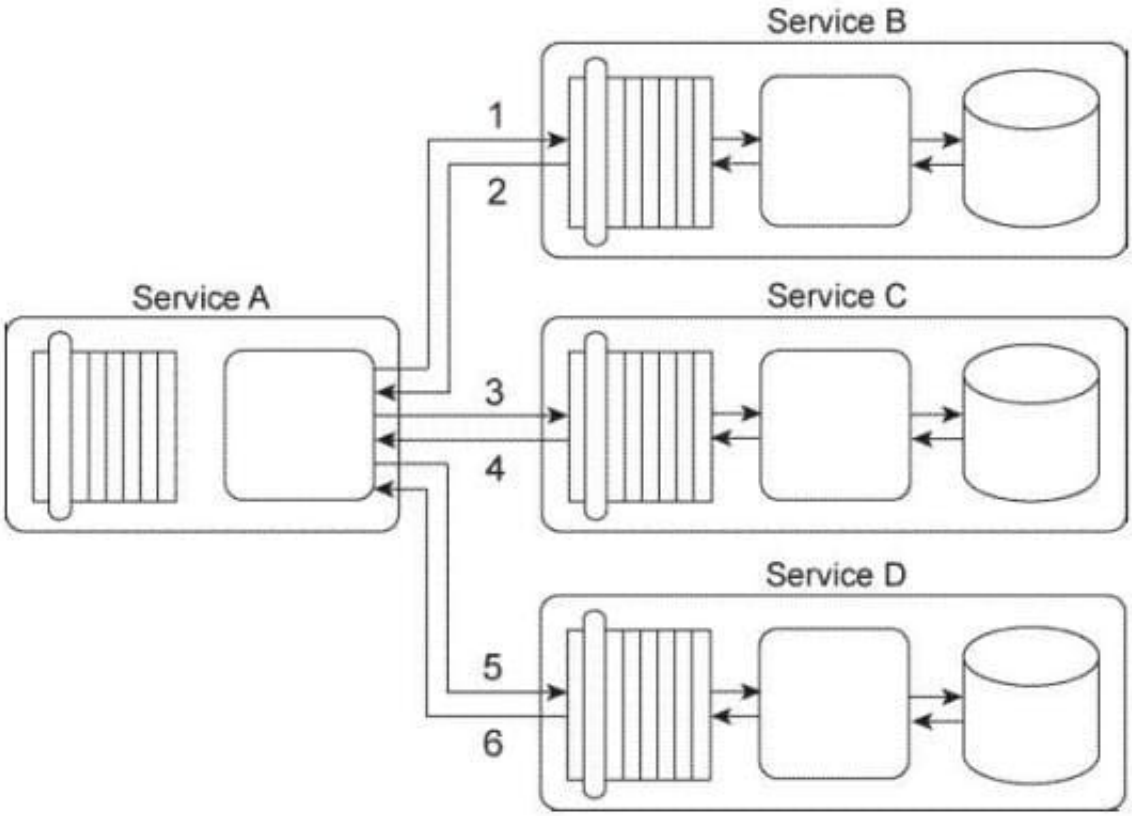
D. None of the above.

Correct Answer: A

QUESTION 3

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.

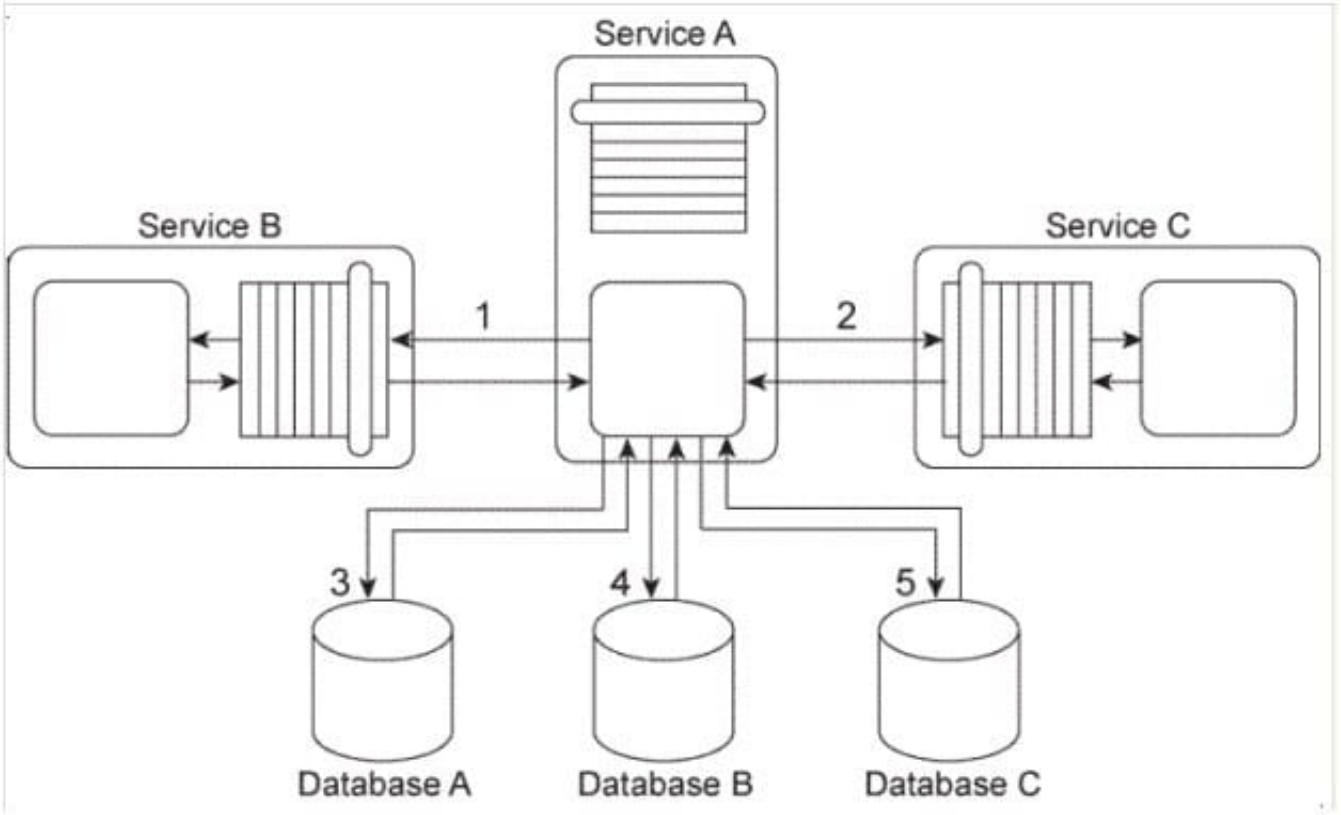
Correct Answer: D

QUESTION 4

Service A sends a message to Service B (1). After Service B writes the message contents to Database A

(2) it issues a response message back to Service A (3). Service A then sends a message to Service C (4). Upon receiving this message, Service C sends a message to Service D (5), which then writes the message contents to Database B (6) and issues a response message back to Service C (7).

Service A and Service D are in Service Inventory A. Service B and Service C are in Service Inventory B.



You are told that in this service composition architecture, all four services are exchanging invoice-related data in an XML format. However, the services in Service Inventory A are standardized to use a different XML schema for invoice data than the services in Service Inventory B. Also, Database A can only accept data in the Comma Separated Value (CSV) format and therefore cannot accept XML formatted data. Database B only accepts XML formatted data. However, it is a legacy database that uses a proprietary XML schema to represent invoice data that is different from the XML schema used by services in Service Inventory A or Service Inventory B. 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 C and Service D, and between the Service D logic and Database B. The Data Format Transformation pattern can be applied so that data format transformation logic is positioned between Service A and Service C, and between the Service B

logic and Database A.

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

D. None of the above.

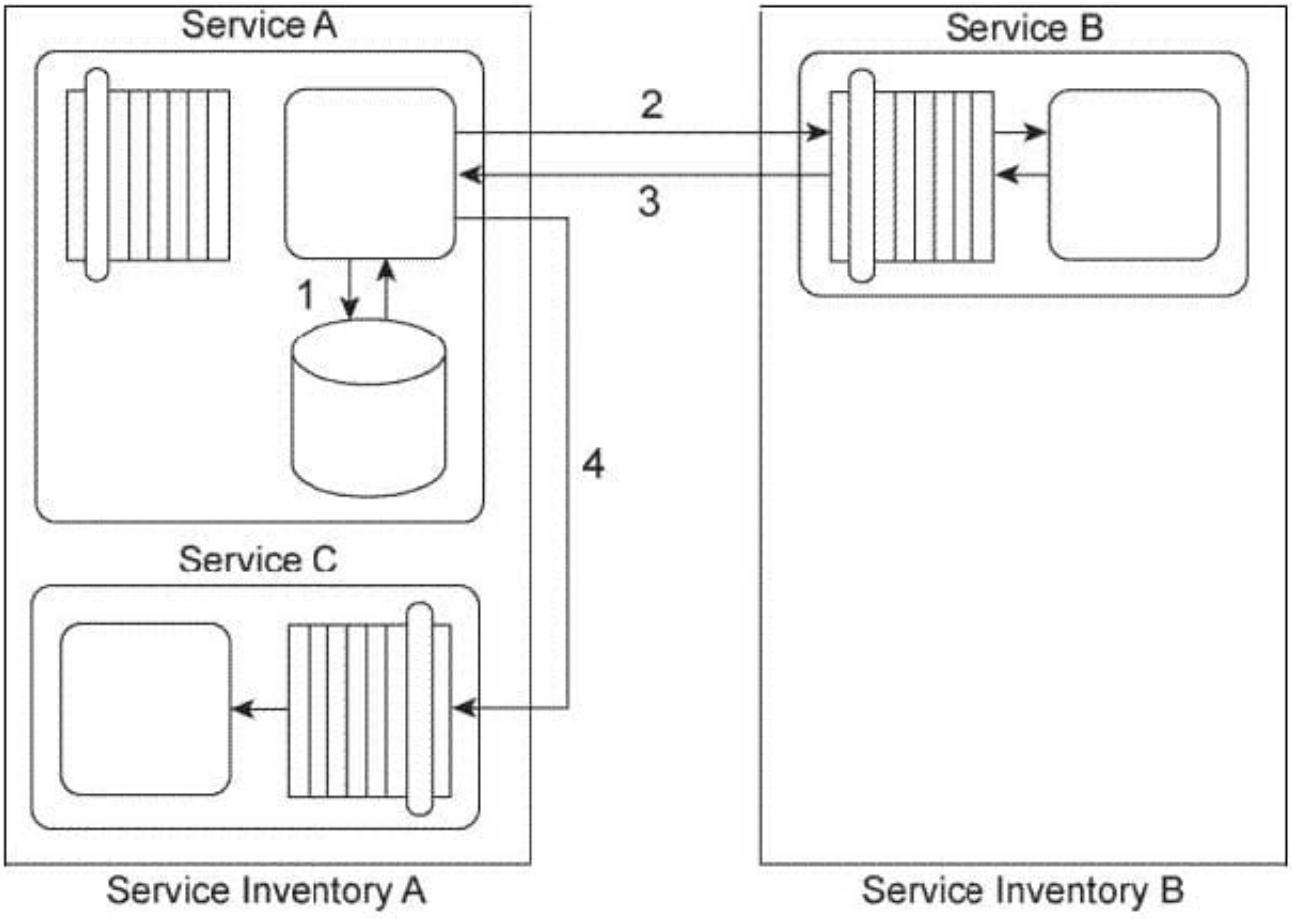
Correct Answer: C

QUESTION 5

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 were designed with service contracts based on different design standards than the services in Service Inventory B. As a result, Service A and Service B use different data models to represent the data they need to exchange. Therefore, Service A and Service B cannot currently communicate. Furthermore, Service C is an agnostic service that is heavily accessed by many concurrent service consumers. Service C frequently reaches its usage thresholds during which it is not available and messages sent to it are not received. How can this service composition architecture be changed to avoid these problems?

A. The Data Model Transformation pattern can be applied by establishing an intermediate processing layer between Service A and Service B that can transform a message from one data model to another at runtime. The Intermediate Routing and Service Agent patterns can be applied so that when Service B sends a response message, a service agent can intercept the message and, based on its contents, either forward the message to Service A or route the message to Service C . The Service Autonomy principle can be further applied to Service C together with the Redundant Implementation pattern to help establish a more reliable and scalable service architecture.

B. The Data Model Transformation pattern can be applied by establishing an intermediate processing layer between Service A and Service B that can transform a message from one data model to another at runtime. The Asynchronous Queuing pattern can be applied to establish an intermediate queue between Service A and Service C so that when Service A needs to send a message to Service C, the queue will store the message and retransmit it to Service C until it is successfully delivered. The Service Autonomy principle can be further applied to Service C together with the Redundant Implementation pattern to help establish a more reliable and scalable service architecture.

C. The Data Model Transformation pattern can be applied by establishing an intermediate processing layer between Service A and Service B that can transform a message from one data model to another at runtime. The Intermediate Routing and Service Agent patterns can be applied so that when Service B sends a response message, a service agent can intercept the message and, based on its contents, either forward the message to Service A or route the message to Service C . 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.

D. None of the above.

Correct Answer: B