# 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

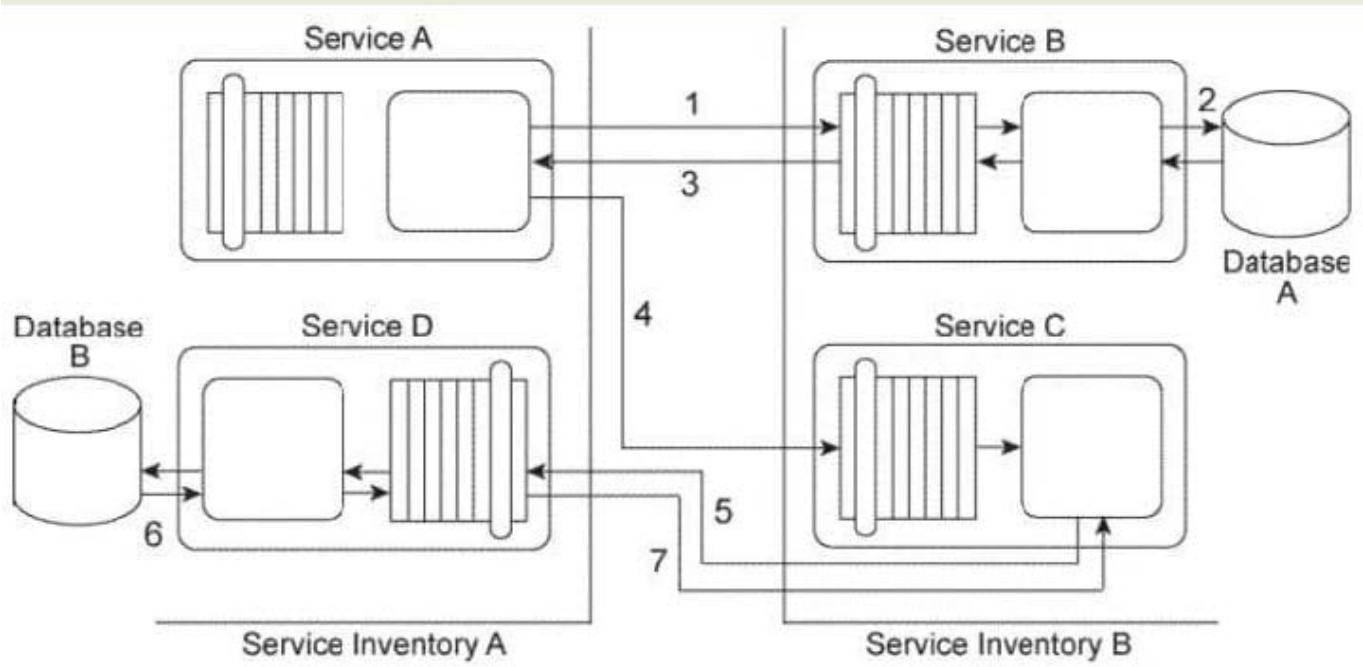
**QUESTION 1**

Service Consumer A invokes Service A (1). The logic within Service A is required to retrieve three independent data values from Services B, C, and D and to then return these data values back to Service Consumer A.

To accomplish this, Service A begins by sending a request message to Service B (2). After receiving a response message with the first data value from Service B, Service A sends a request message to Service C (3). After it receives a response message with the second data value from Service C, Service A then sends a request message to Service D (4). Upon receiving a response message with the third data value from Service D. Service A finally sends its own response message (containing all three collected data values) back to Service Consumer A.

Service Consumer A and Service A reside in Service Inventory A. Service B and Service C reside in Service Inventory B. Service D is a public service that can be openly accessed via the World Wide Web. The service is also available for purchase so that it can be deployed independently within IT enterprises.

Due to the rigorous application of the Service Abstraction principle within Service Inventory B, the only information that is made available about Service B and Service C are the published service contracts. For Service D, the service contract plus a Service Level Agreement (SLA) are made available. The SLA indicates that Service D has a planned outage every night from 11 PM to midnight.




You are an architect with a project team building services for Service Inventory A . You are told that the owners of Service Inventory A and Service Inventory B are not generally cooperative or communicative. Cross-inventory service composition is tolerated, but not directly supported. As a result, no SLAs for Service B and Service C are available and you have no knowledge about how available these services are. Based on the service contracts you can determine that the services in Service Inventory B use different data models and a different transport protocol than the services in Service Inventory A. Furthermore, recent testing results have shown that the performance of Service D is highly unpredictable due to the heavy amount of concurrent access it receives from service consumers from other organizations. You are also told that there is a concern about how long Service Consumer A will need to remain stateful while waiting for a response from Service A . What steps can be taken to solve these problems?

A. The Event-Driven Messaging pattern is applied so that a subscriber-publisher relationship is established between Service Consumer A and Service A . This gives Service A the flexibility to provide its response to Service Consumer A whenever it is able to collect the three data values without having to require that Service Consumer A remain stateful.

The Asynchronous Queuing pattern is applied so that a central messaging queue is positioned between Service A and Service B and between Service A and Service C . The Data Model Transformation and Protocol Bridging patterns are applied to enable communication between Service A and Service B and between Service A and Service C . The Redundant Implementation pattern is applied so that a copy of Service D is brought in- house and made part of Service Inventory A.

B. The Asynchronous Queuing pattern is applied so that a central messaging queue is positioned between Service A and Service B and between Service A and Service C and so that a separate messaging queue is positioned between Service A and Service Consumer

C. The Data Model Transformation and Protocol Bridging patterns are applied to enable communication between Service A and Service B and between Service A and Service C . The Redundant Implementation pattern is applied so that a copy of Service D is brought in- house for fail-over purposes. The Legacy Wrapper pattern is further applied to wrap Service D with a standardized service contract that is in compliance with the design standards used in Service Inventory

A. This wrapper utility service first attempts to access the external service, but if that service is

unavailable it will access the redundant internal service instead.

D. The Reliable Messaging pattern is applied so that a system of acknowledgements is established between Service Consumer A and Service A . This gives Service A the flexibility to provide Service Consumer A with acknowledgements that indicate that the processing steps that are occurring between Service A and Service B, Service C, and Service D are progressing. The Asynchronous Queuing pattern is applied so that a central messaging queue is positioned between Service A and Service B and between Service A and Service C and between Service A and Service D . The Data Model Transformation and Protocol Bridging patterns are applied to enable communication between Service A and Service B and between Service A and Service C .

E. None of the above.

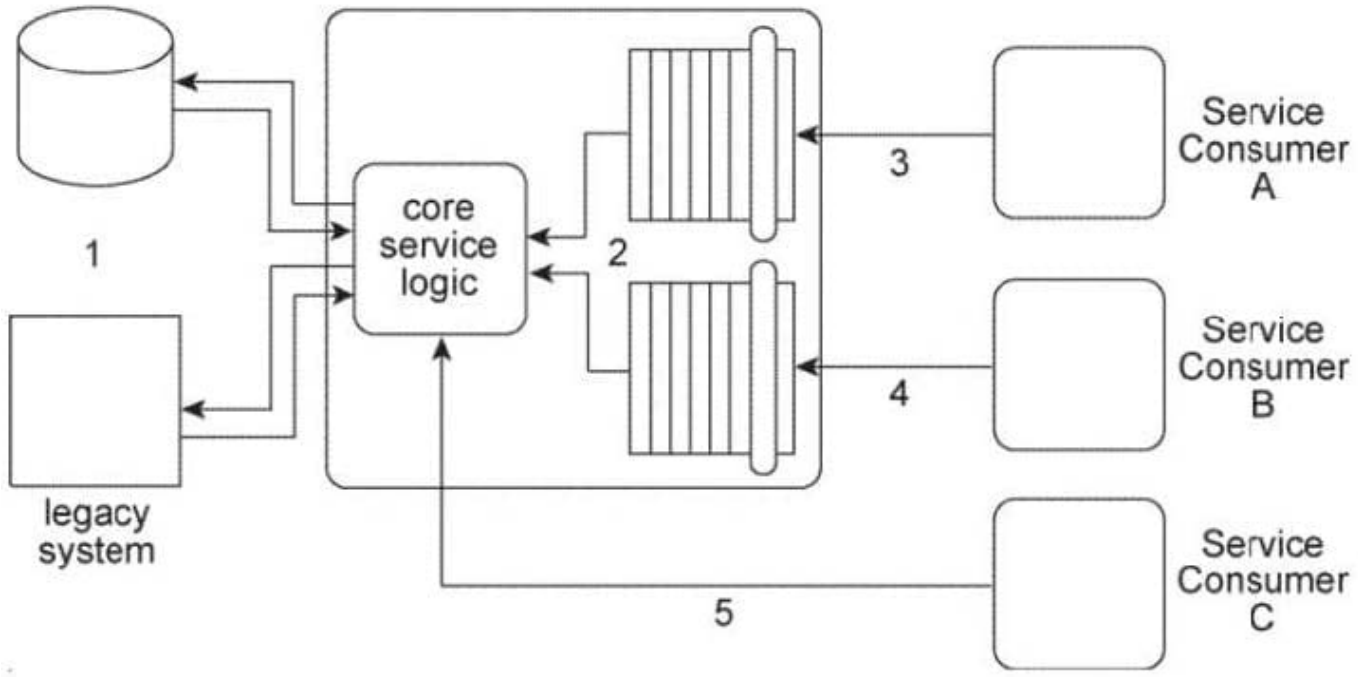
Correct Answer: B

**QUESTION 2**

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

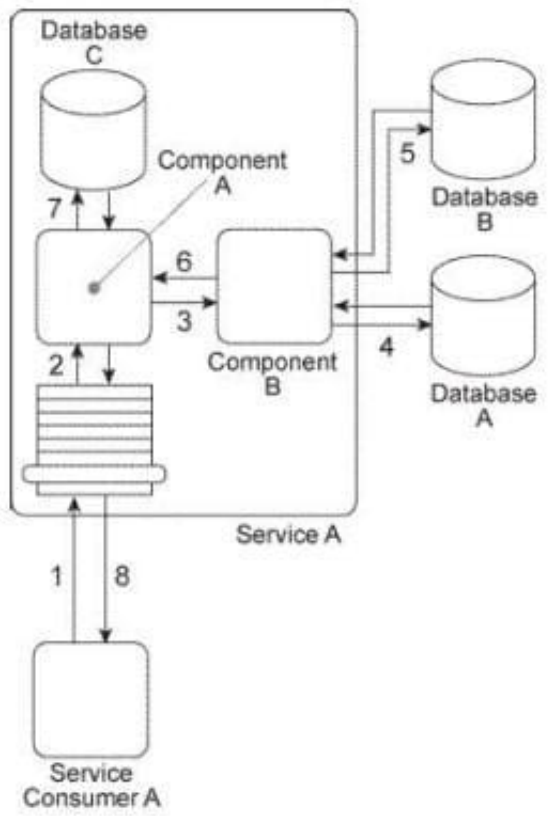
**QUESTION 3**

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**

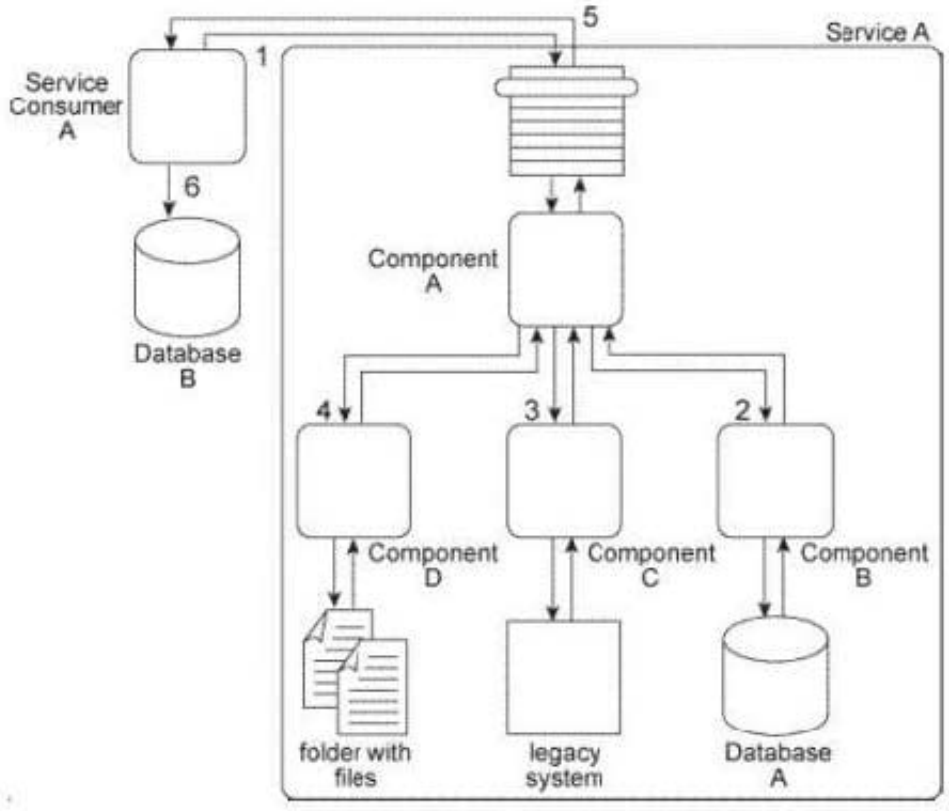
When Service A receives a message from Service Consumer A(1),the message is processed by Component A. This component first invokes Component B (2), which uses values from the message to query Database A in order to retrieve additional data. Component B then returns the additional data to Component A.

Component A then invokes Component C (3), which interacts with the API of a legacy system to retrieve a new data value. Component C then returns the data value back to Component A.

Next, Component A sends some of the data it has accumulated to Component D (4), which writes the data to a text file that is placed in a specific folder. Component D then waits until this file is imported into a different system via a regularly scheduled batch import. Upon completion of the import, Component D returns a success or failure code back to Component A.

Component A finally sends a response to Service Consumer A (5) containing all of the data collected so far and Service Consumer A writes all of the data to Database B (6).

Components A, B, C. and D belong to the Service A service architecture. Database A, the legacy system, and the file folders are shared resources within the IT enterprise.




Service A is a task service that completes an entire business task on its own without having to compose other services. However, you have received many complaints about the reliability of Service A . Specifically, it has three problems. First, when Component B accesses Database A, it may not receive a response for several minutes when the database is being accessed by other applications in the IT enterprise. Secondly, the legacy system accessed by Component C frequently crashes and therefore becomes unavailable for extended periods of time. Third, for Component D to respond to Component A, it must first wait for the batch import of the files to occur. This can take several minutes during which Service Consumer A remains stateful and consumes excessive memory. What steps can be taken to address these three problems?

A. The Legacy Wrapper pattern can be applied so that Component B is separated to wrap the shared database, thereby allowing Component A to interact with this new service instead of directly interacting with the database. The Legacy Wrapper pattern can be applied again so that Component C is separated into a separate service that acts as a wrapper of the legacy system API. Component D can then be separated into a separate service and the Event-Driven Messaging pattern can be applied to establish a publisher- subscriber relationship between this new service and Component A and between Service A and Service Consumer A. The interaction between Service Consumer A and Component A is then redesigned so that Component A issues a message back to Service Consumer A when the event related to the batch import is triggered.

B. The Service Data Replication pattern can be applied so that Component B can access a replicated database instead of having to access the shared Database A directly. The Legacy Wrapper pattern can be applied so that Component C is separated into a separate service that acts as a wrapper of the legacy system API. Next, the Reliable Messaging pattern can be applied so that acknowledgements are issued from the new wrapper service to Component A, thereby enabling notifying Component A during times when the legacy system is unavailable. Finally, Component D is separated into a separate service and the Event-Driven Messaging pattern is applied to establish a publisher-subscriber relationship between this new service and Component A. The interaction between Service Consumer A and Component A is then redesigned so that Component A first interacts with Component B and the new wrapper service. Service A then issues a final message back to Service Consumer A.

C. The Service Data Replication pattern can be applied so that Component B can access a replicated database instead

of having to access the shared Database A directly. The Legacy Wrapper pattern can be applied so that Component C is separated into a separate service that acts as a wrapper of the legacy system API. Next, the Asynchronous Queuing pattern can be applied so that a messaging queue is positioned between Component A and the new wrapper service, thereby enabling communication during times when the legacy system is unavailable. Finally, Component D is separated into a new service and the Event-Driven Messaging pattern is applied to establish a publisher-subscriber relationship between this service and Component A and between Service A and Service Consumer A. The interaction logic is redesigned as follows: Component A interacts with Component B, the new wrapper service, and then issues a request to the new event-driven service. Upon receiving a response triggered by the event related to the batch import, Service A responds to Service Consumer A.

D. None of the above.

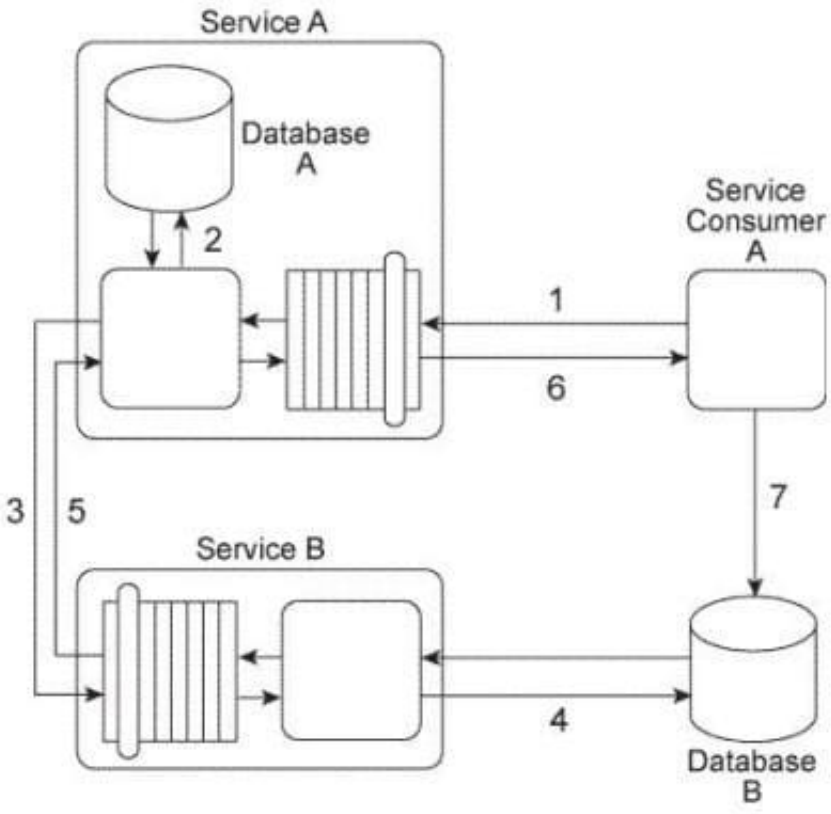
Correct Answer: C

**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.



There are two problems with this service composition architecture that you are asked to address: First, both Service Consumer A and Service B need to transform the business document data from an XML format to a proprietary Comma Separated Value (CSV) in order to write the data to Database B. This has led to redundant data format transformation logic that has been difficult to keep in synch when Database B changes. Secondly, Service A is an entity service that is being reused by several other service compositions. It has lately developed reliability problems that have caused the service to become unavailable for extended periods. What steps can be taken to solve these problems?

A. The Legacy Wrapper pattern can be applied so that data access to Database B is separated into a new wrapper utility service. This way, the Data Format Transformation pattern only needs to be applied within the logic of this new service which will expose a standardized contract that both Service Consumer A and Service B can access. The Asynchronous Queuing pattern can be applied so that messaging queues are established between Service Consumer A and Service A and between Service A and Service B . The Service Autonomy principle can be further applied to Service A in order to establish a more isolated and reliable surrounding infrastructure.

B. The Legacy Wrapper pattern can be applied so that data access to Database B is separated into a new wrapper utility service. This way, the Data Format Transformation pattern only needs to be applied within the logic of this new service which will expose a standardized contract that both Service Consumer A and Service B can access. The Reliable Messaging pattern can be applied so that acknowledgements are passed between Service Consumer A and Service A and between Service A and Service B . The Service Composability principle can be further applied to Service A in order to optimize its service architecture for improved participation in multiple service compositions.

C. The service composition can be redesigned with the application of the Contract Centralization pattern so that instead of writing the business document to Database B, Service Consumer A sends the business document to Service B instead. This way, Service B would provide the only location where data format transformation logic for Database B needs to be carried out, which further supports the application of the Service Reusability principle. The Reliable Messaging pattern can be applied so that acknowledgements are passed between Service Consumer A and Service A and between Service A and Service B . The Service Composability principle can be further applied to Service A in order to optimize its service architecture for improved participation in multiple service compositions.

D. None of the above.

Correct Answer: A

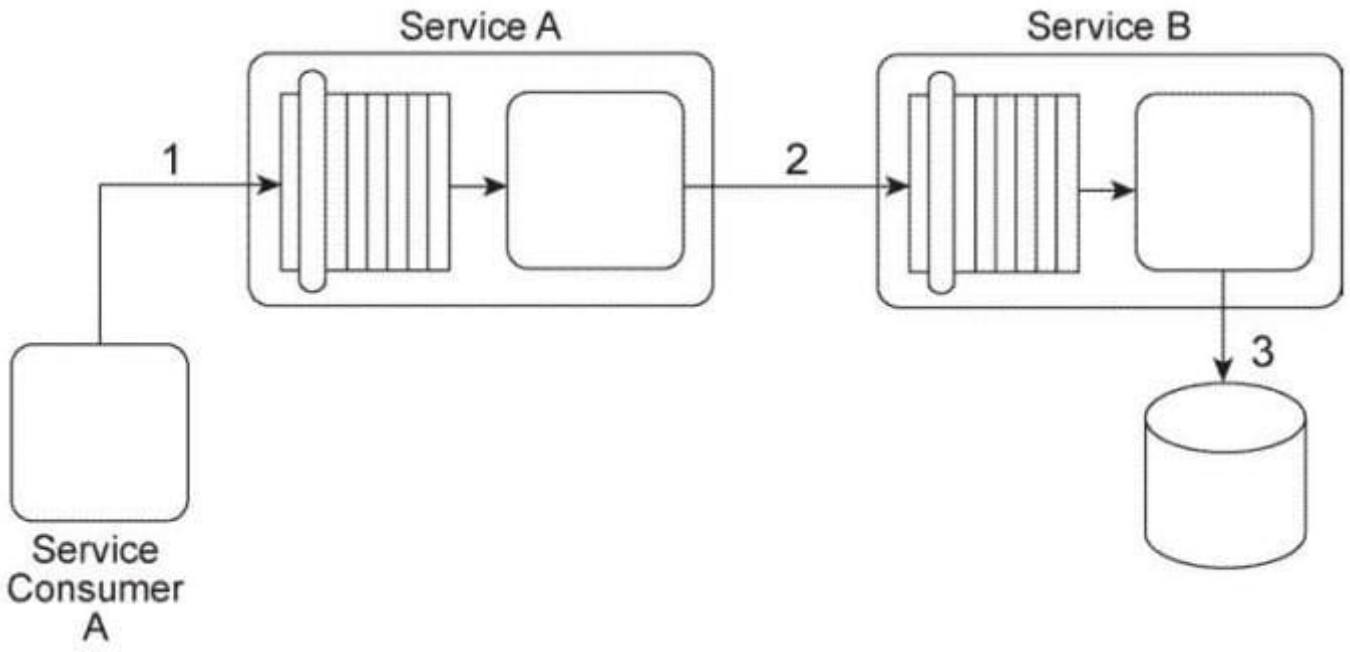
**QUESTION 6**

Service A is an entity service with a functional context dedicated to invoice-related processing. Service B is a utility service that provides generic data access to a database.

In this service composition architecture, Service Consumer A sends a SOAP message containing an invoice XML document to Service A(1). Service A then sends the invoice XML document to Service B (2), which then writes the invoice document to a database.

The data model used by Service Consumer A to represent the invoice document is based on XML Schema

A. The service contract of Service A is designed to accept invoice documents based on XML Schema B. The service contract for Service B is designed to accept invoice documents based on XML Schema A. The database to which Service B needs to write the invoice record only accepts entire business documents in Comma Separated Value (CSV) format.




Due to the incompatibility of XML schemas used by the services, the sending of the invoice document from Service Consumer A through to Service B cannot be accomplished using the services as they currently exist. Assuming that the Contract Centralization and Logic Centralization patterns are being applied, what steps can be taken to enable the sending of the invoice document from Service Consumer A to the database without adding logic that will increase the runtime performance of the service composition?

A. The Data Model Transformation pattern can be applied so that the invoice document sent by Service Consumer A is transformed into an invoice document that is compliant with the XML Schema B used by Service A . The Data Model Transformation pattern can be applied again to ensure that the invoice document sent by Service A is compliant with XML Schema A used by Service B.

B. The service composition can be redesigned so that Service Consumer A sends the invoice document directly to Service B. Because Service Consumer A and Service B use XML Schema A, the need for transformation logic is avoided. This naturally applies the Service Loose Coupling principle because Service Consumer A is not required to send the invoice document in a format that is compliant with the database used by Service B.

C. The Standardized Service Contract principle can be applied to the service contract of Service A so that it is redesigned to use XML Schema A. This would make it capable of receiving the invoice document from Service Consumer A and sending the invoice document to Service B without the need to further apply the Data Model Transformation pattern.

D. None of the above.

Correct Answer: C

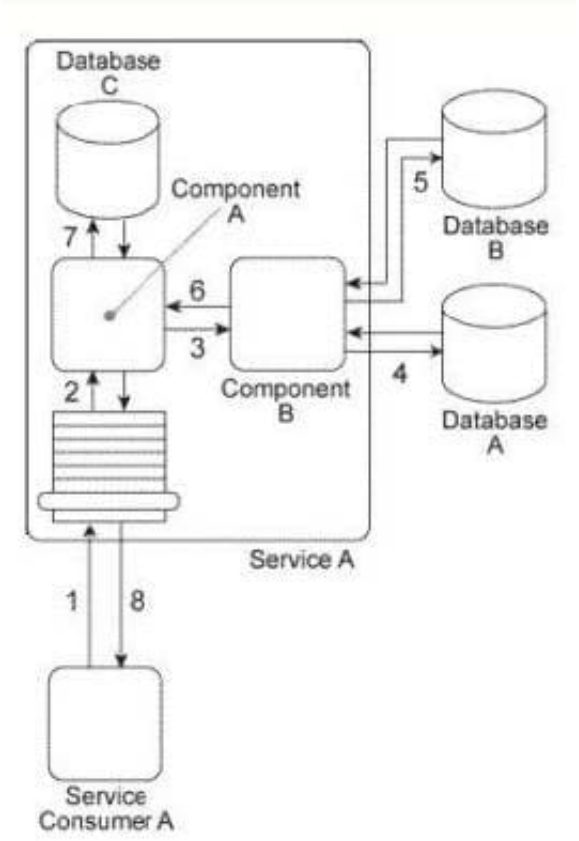
**QUESTION 7**

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.

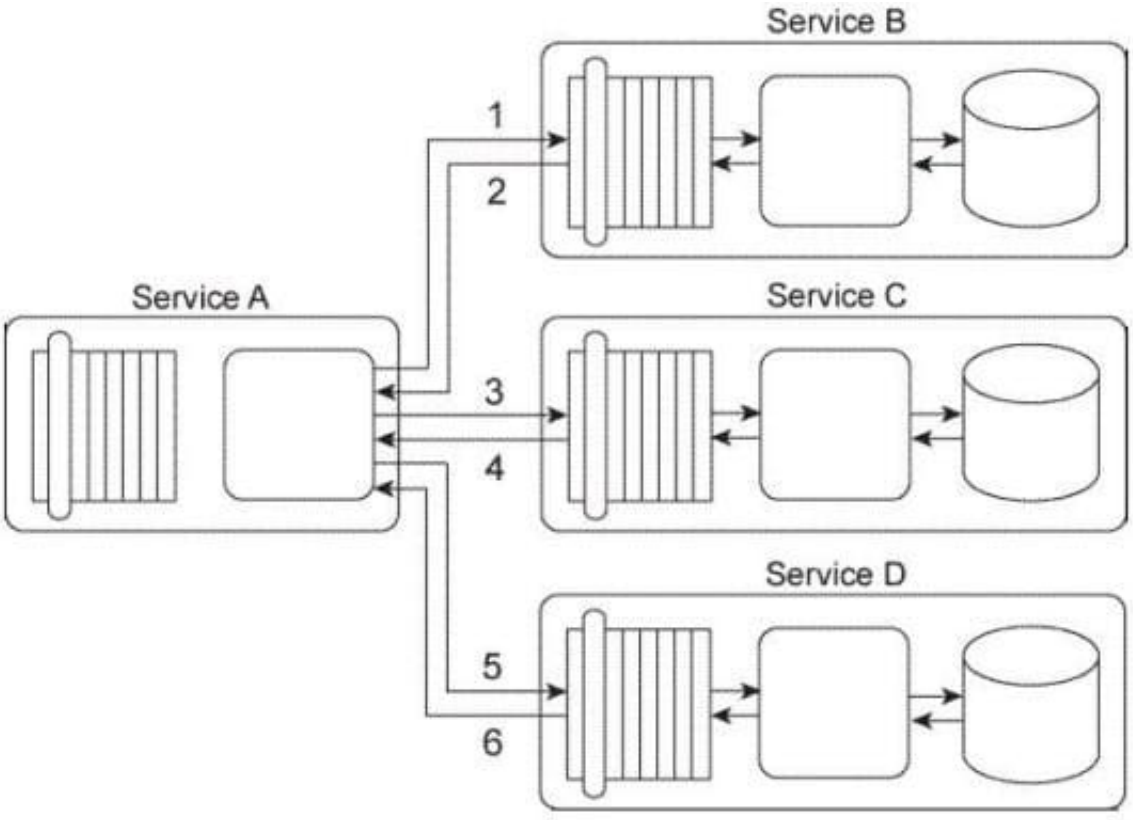
E. None of the above.

Correct Answer: B

**QUESTION 8**

Service A is 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 9**

Upon reviewing these requirements it becomes evident to you that the Orchestration compound pattern will need to be applied. However, there are additional requirements that need to be fulfilled. To build this service composition architecture, which patterns that is not associated with the Orchestration compound pattern need to also be applied? (Be sure to choose only those patterns that relate directly to the requirements described above. Patterns associated with the Orchestration compound pattern include both the required or core patterns that are part of the basic compound pattern and the optional patterns that can extend the basic compound pattern.)

A. Atomic Service Transaction

B. Compensating Service Transaction

C. Data Format Transformation

D. Data Model Transformation

E. Event-Driven Messaging

F. Intermediate Routing

G. Policy Centralization

H. Process Centralization

I. Protocol Bridging

J. Redundant Implementation

K. Reliable Messaging

L. Service Data Replication

M. State Repository

Correct Answer: CL

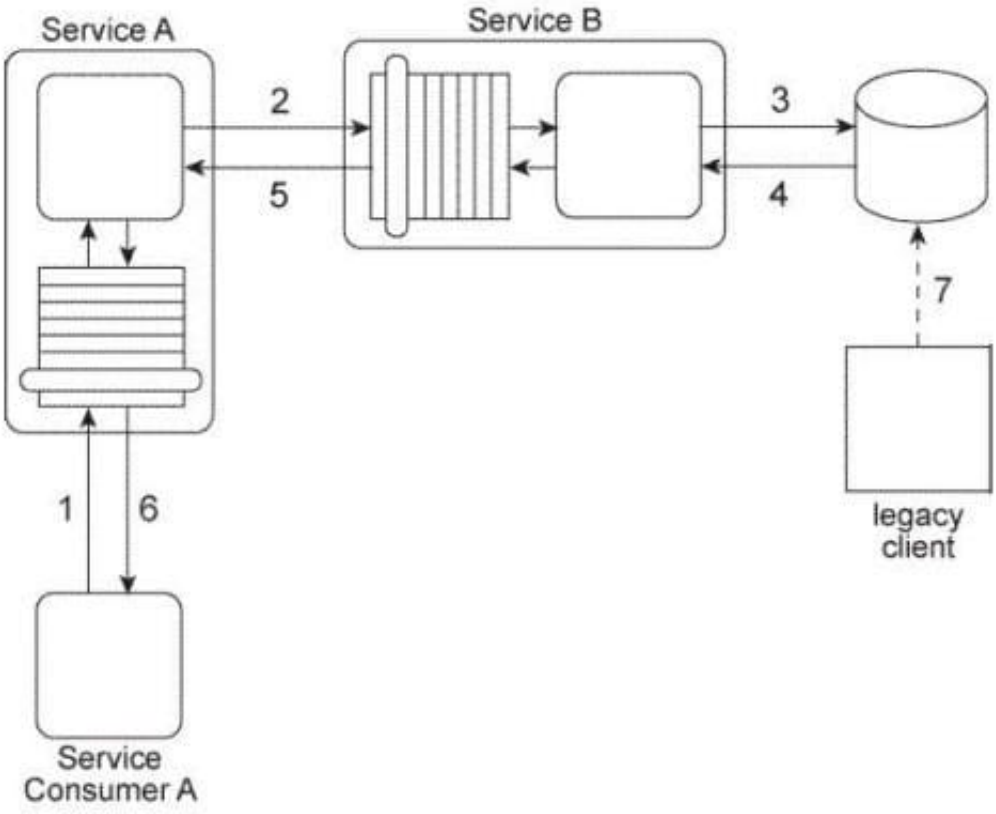
**QUESTION 10**

Service A is an entity service that provides a Get capability that returns a data value that is frequently changed.

Service Consumer A invokes Service A in order to request this data value (1). For Service A to carry out this request, it must invoke Service B (2), a utility service that interacts (3.4) with the database in which the data value is stored. Regardless of whether the data value changed, Service B returns the latest value to Service A (5), and Service A returns the latest value to Service Consumer A (6).

The data value is changed when the legacy client program updates the database (7) When this change happens is not predictable. Note also that Service A and Service B are not always available at the same time.

Any time the data value changes. Service Consumer A needs to receive it as soon as possible. Therefore, Service Consumer A initiates the message exchange shown in the Figure several times a day. When it receives the same data value as before, the response from Service A is ignored. When Service A provides an updated data value, Service Consumer A can process it to carry out its task.




The current service composition architecture is using up too many resources due to the repeated invocation of Service A by Service Consumer A and the resulting message exchanges that occur with each invocation. What steps can be taken to solve this problem?

A. The Event-Driven Messaging pattern can be applied by establishing a subscriber- publisher relationship between Service A and Service B . This way, every time the data value is updated, an event is triggered and Service B, acting as the publisher, can notify Service A, which acts as the subscriber. The Asynchronous Queuing pattern can be applied between Service A and Service B so that the event notification message sent out by Service B will be received by Service A, even when Service A is unavailable.

B. The Event-Driven Messaging pattern can be applied by establishing a subscriber- publisher relationship between Service Consumer A and Service A . This way, every time the data value is updated, an event is triggered and Service A, acting as the publisher, can notify Service Consumer A, which acts as the subscriber. The Asynchronous Queuing pattern can be applied between Service Consumer A and Service A so that the event notification message sent out by Service A will be

received by Service Consumer A, even when Service Consumer A is unavailable.

C. The Asynchronous Queuing pattern can be applied so that messaging queues are established between Service A and Service B and between Service Consumer A and Service A . This way, messages are never lost due to the unavailability of Service A or Service B.

D. None of the above.

Correct Answer: D