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.geekcert.com/s90-09.html

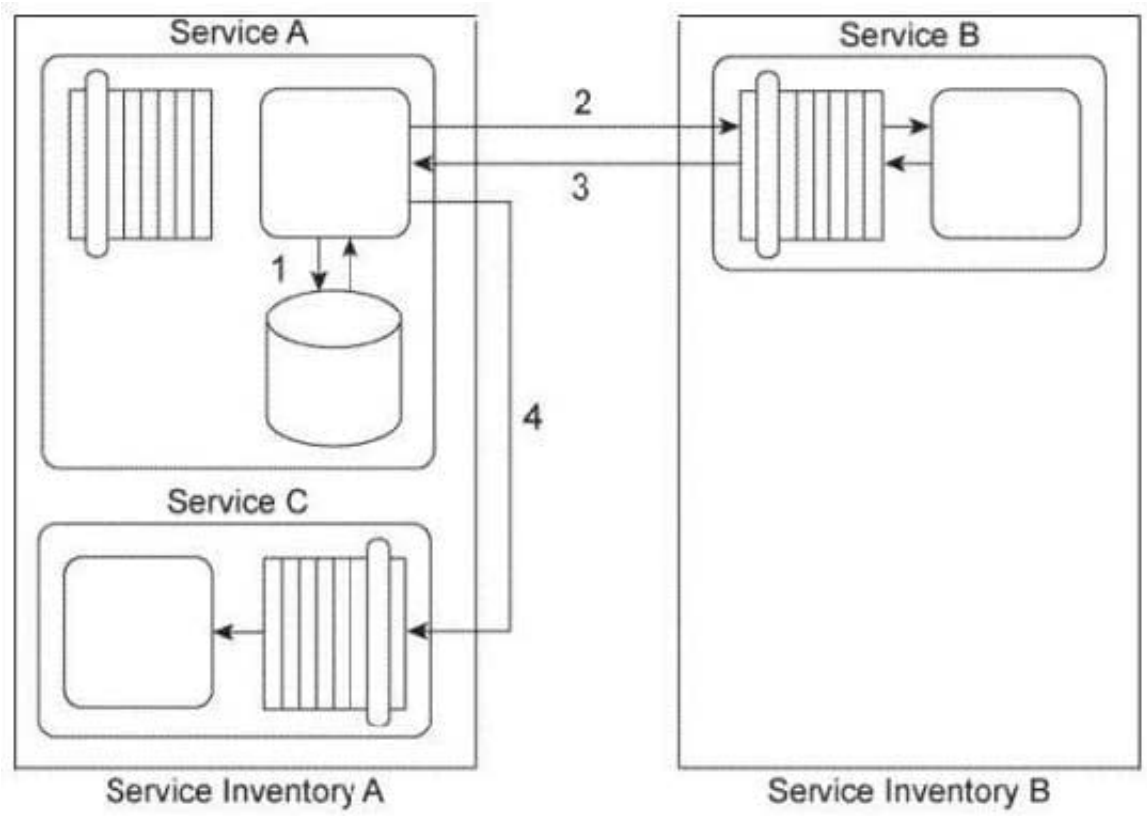
100% Passing Guarantee
100% Money Back Assurance

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

QUESTION 1

Service A is a task service that 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.

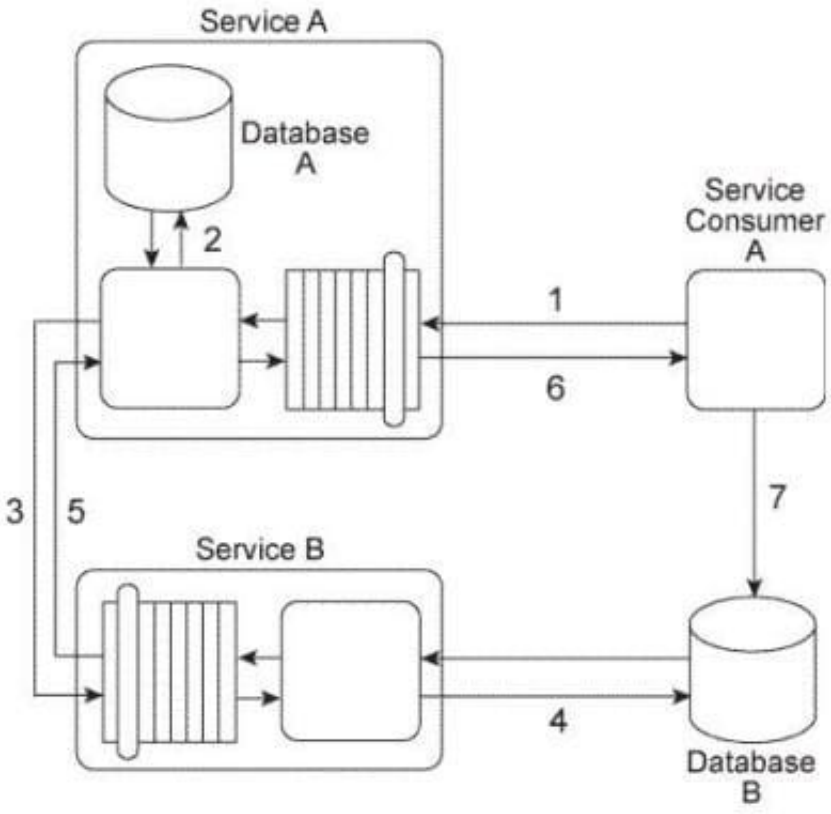
D. None of the above.

Correct Answer: B

QUESTION 2

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.

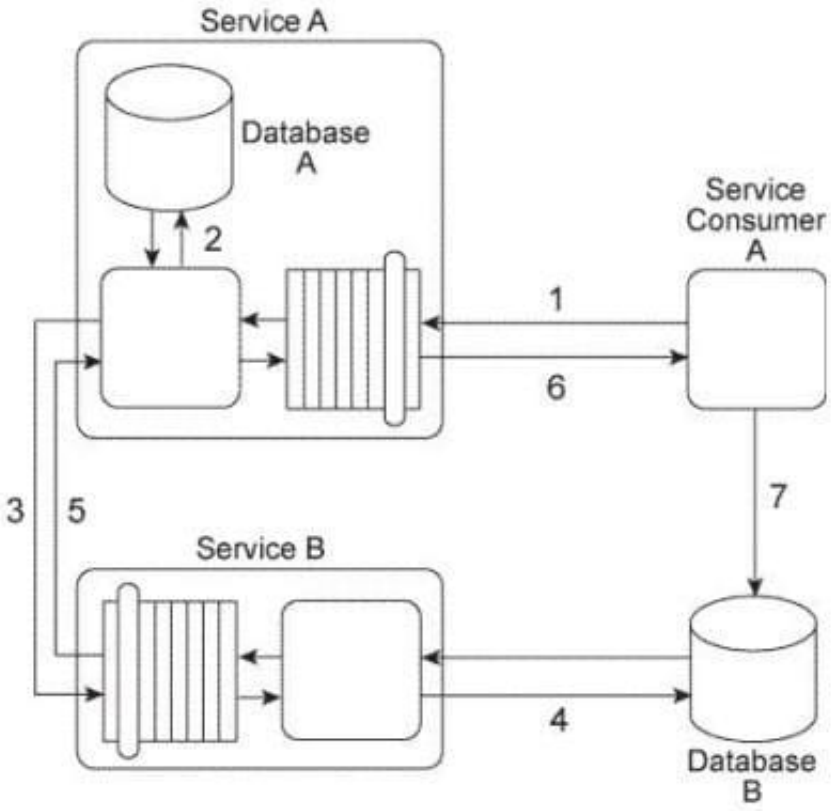
Correct Answer: D

QUESTION 3

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

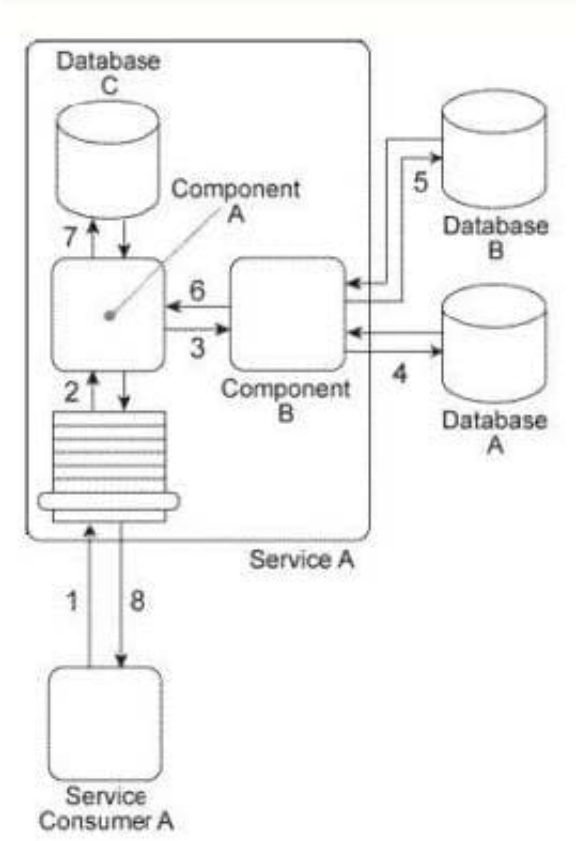
QUESTION 4

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 .

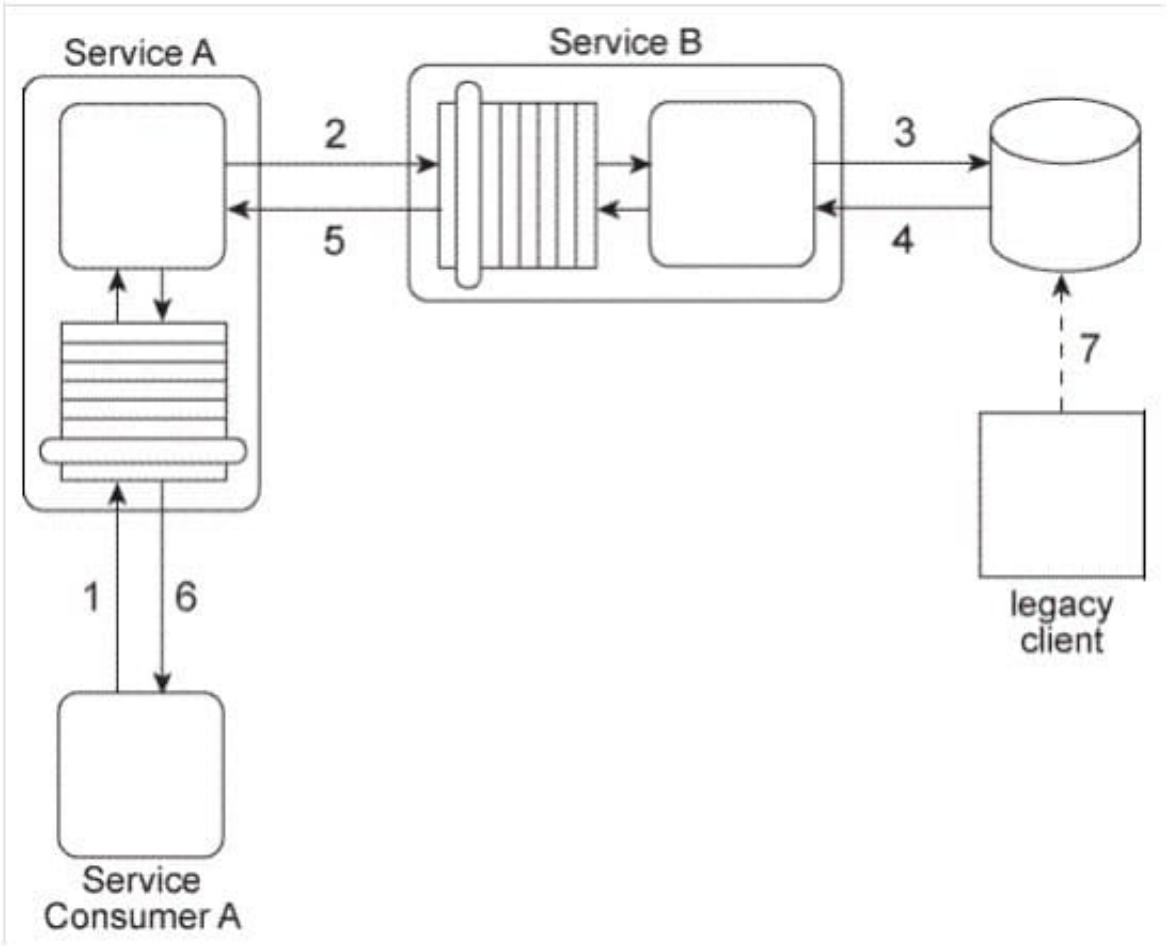
D. The Service Statelessness principle can be applied together with the State Repository pattern in order to establish a state database that Component A can defer the business document data to while it waits for a response from Component B. The Service Autonomy principle can be applied together with 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 5

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