S90.09 $^{\text{Q\&As}}$

SOA Design & Architecture Lab

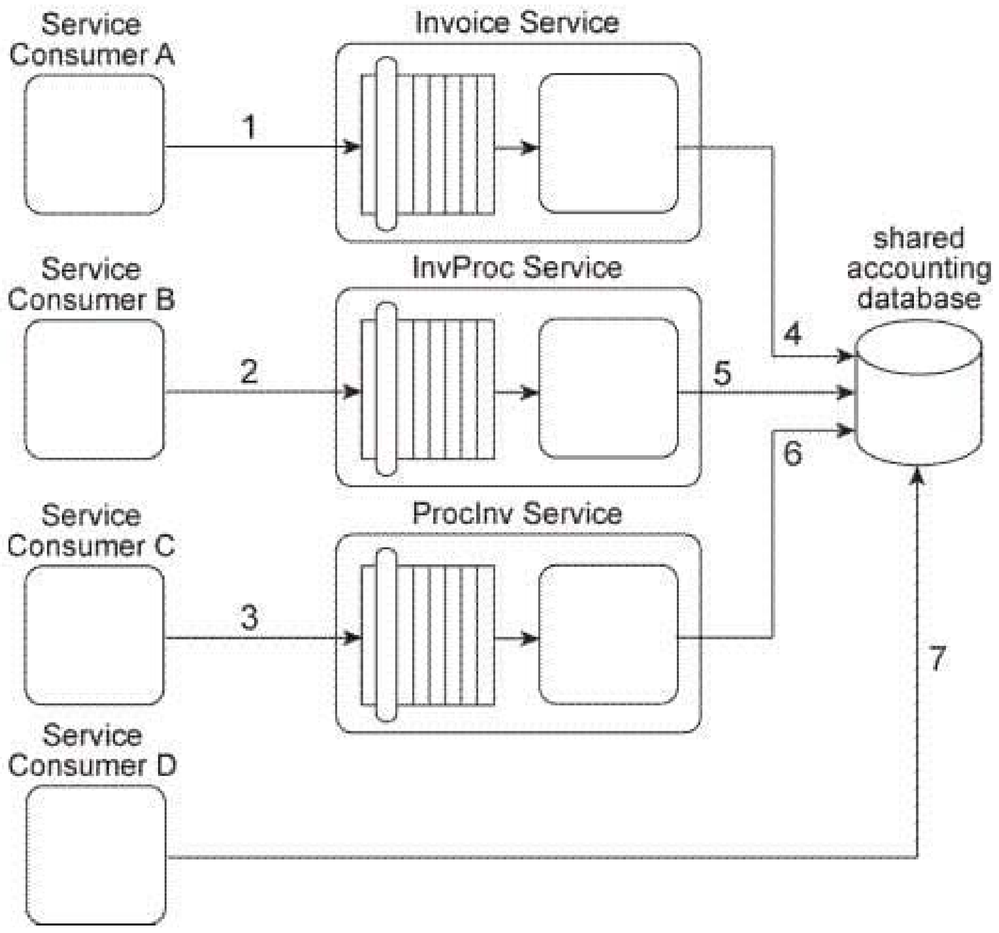
Pass SOA S90.09 Exam with 100% Guarantee

QUESTION 1

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.)



A project team recently proclaimed that it has successfully applied the Contract Centralization pattern to the service inventory in which the Invoice service, InvProc service, and Proclnv service reside. Upon reviewing the previously described architecture you have doubts that this is true. After voicing your doubts to a manager, you are asked to provide specific evidence as to why the Contract Centralization is not currently fully applied. Which of the following statements provides this evidence?

A. The Contract Centralization pattern is not fully applied to the Invoice, InvProc, and Proclnv services because they are being accessed by different service consumers and because they have redundant logic that introduces denormalization into the service inventory.

B. The Contract Centralization pattern is not fully applied because Service Consumer D is accessing the shared accounting database directly.

C. The Contract Centralization pattern is not fully applied because none of the invoice- related services are carrying out data access logic via a centralized and standardized invoice service. This is primarily because the Standardized Service Contract principle was not consistently applied during the delivery processes of the individual services.

D. None of the above.

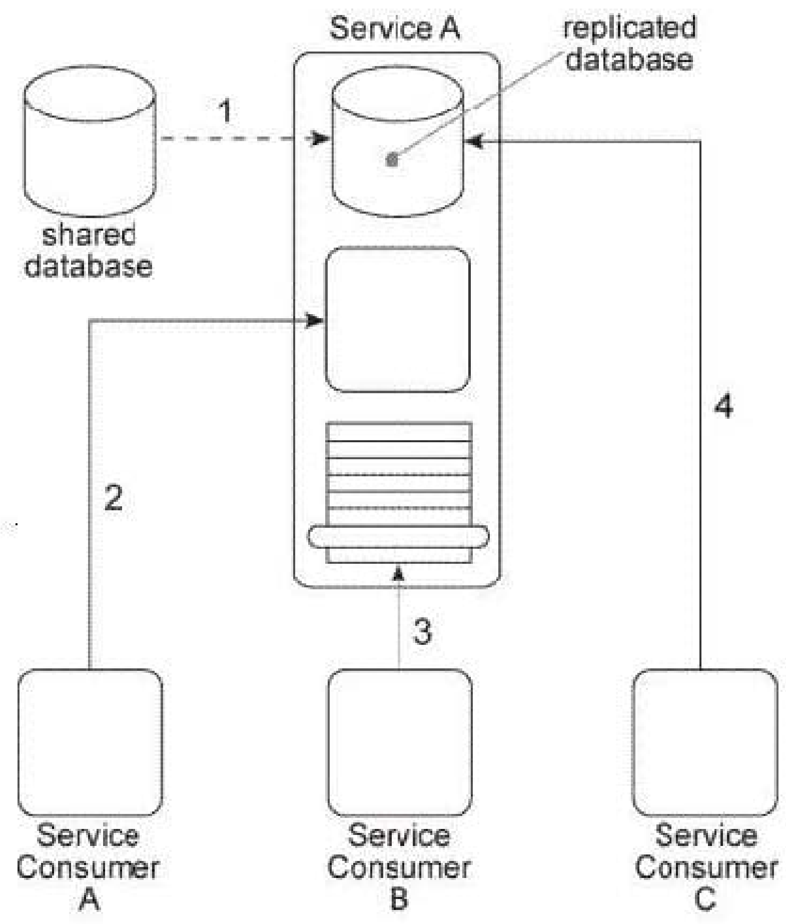
Correct Answer: B

QUESTION 2

Service A is a utility service that provides generic data access logic to a database that contains data that is periodically replicated from a shared database (1). Because the Standardized Service Contract principle was applied to the design of Service A, its service contract has been fully standardized.

The service architecture of Service A is being accessed by three service consumers. Service Consumer A accesses a component that is part of the Service A implementation by invoking it directly (2). Service Consumer B invokes Service A by accessing its service contract (3). Service Consumer C directly accesses the replicated database that is part of the Service A implementation (4).

You\\'ve been told that the shared database will soon be replaced with a new database product that will have new data models and new replication technology. How can the Service A architecture be changed to avoid negative impacts that may result from the replacement of the database and to establish a service architecture in which negative forms of coupling can be avoided in the future?



A. The Contract Centralization pattern can be applied to force all service consumers to access the Service A architecture via its published service contract. This will prevent negative forms of coupling that could lead to problems when the database is replaced. The Service Abstraction principle can then be applied to hide underlying service implementation details so that future service consumers cannot be designed to access any part of the underlying service implementation.

B. The Contract Centralization pattern can be applied to force Service Consumer C to access the Service A architecture via its published service contract. This will prevent Service Consumer A from being negatively impacted when the database is replaced in the future.

C. The Standardized Service Contract principle can be applied to force Service Consumer B to comply to the standardized service contract of Service A. As a result, the coupling between Service Consumer B and Service A is reduced. The Logic Centralization pattern can then be applied to position the logic provided by Service A as a primary access point for the database. As a result, the component within the Service A architecture abstracts the proprietary details of the database, thereby shielding Service Consumer A (and any future service consumers) from changes made to the database.

D. None of the above.

Correct Answer: A

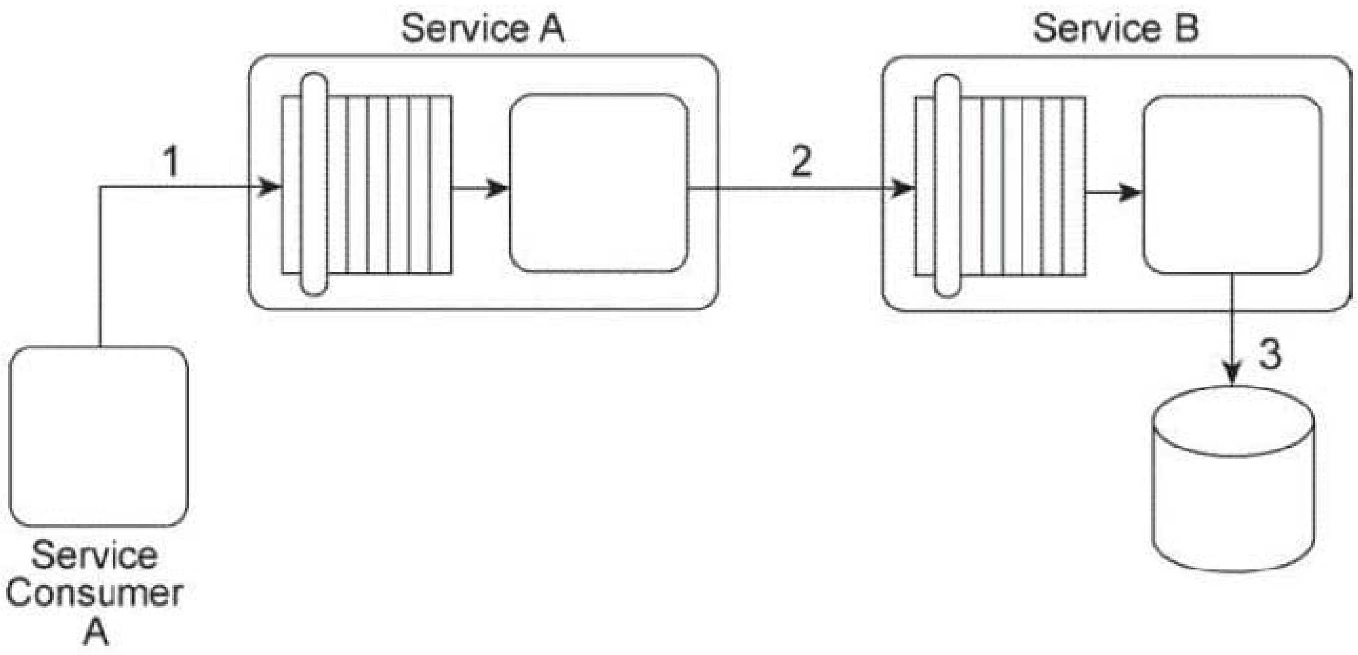
QUESTION 3

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 the 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 pattern is being applied and that the Logic Centralization is not 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 requirements of the service composition?

A. Service Consumer A can be redesigned to use XML Schema B so that the SOAP message it sends is compliant with the service contract of Service A . The Data Model Transformation pattern can then be applied to transform the SOAP message sent by Service A so that it conforms to the XML Schema A used by Service B. The Standardized Service Contract principle must then be applied to Service B and Service Consumer A so that the invoice XML document is optimized to avoid unnecessary validation.

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. Service Consumer A can be redesigned to write the invoice document directly to the database. This reduces performance requirements by avoiding the involvement of Service A and Service B . It further supports the application of the Service Abstraction principle by ensuring that Service Consumer A hides the details of the data access logic required to write to the database.

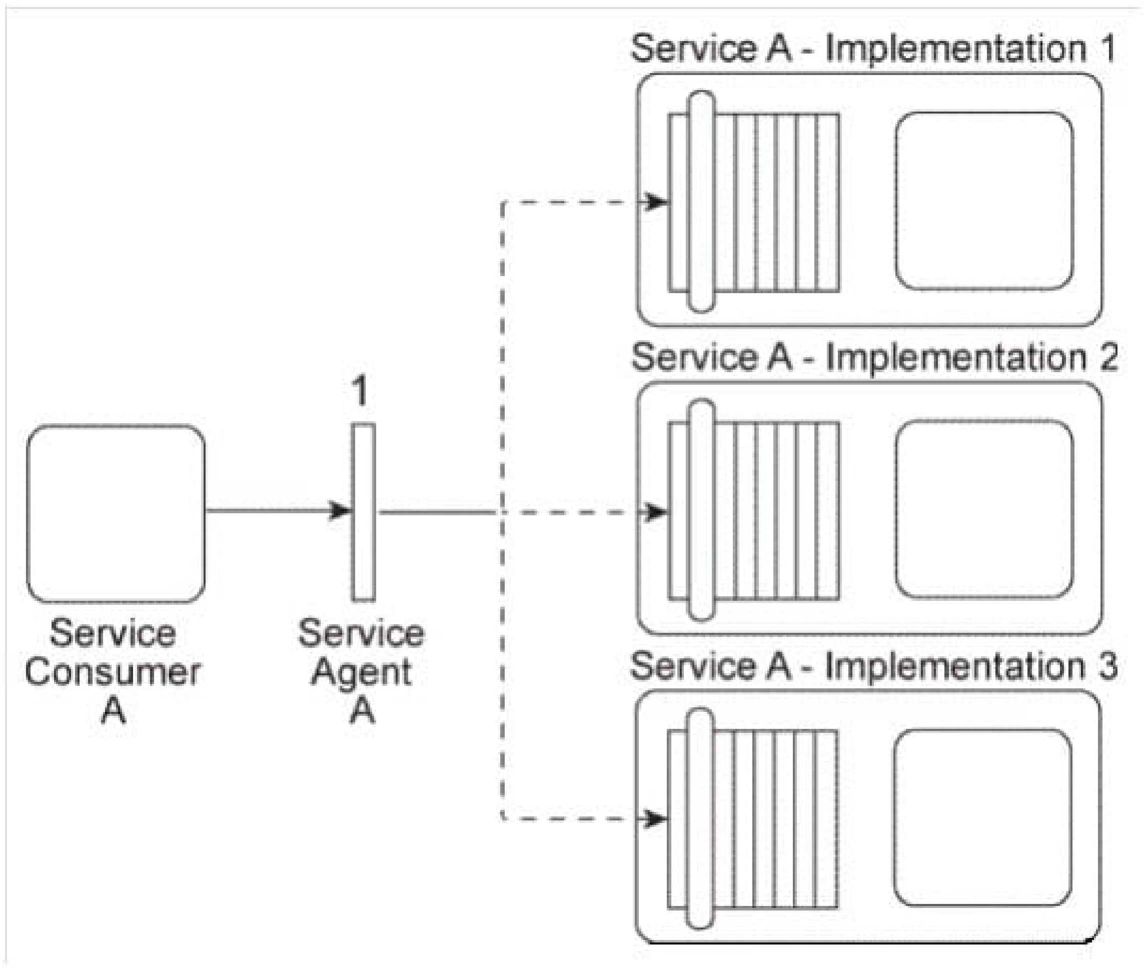
D. None of the above.

Correct Answer: B

QUESTION 4

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 .


Service A - Implementation 1
Service A - Implementation 2
Service A - Implementation 3
1
Service Consumer A
Service Agent A

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

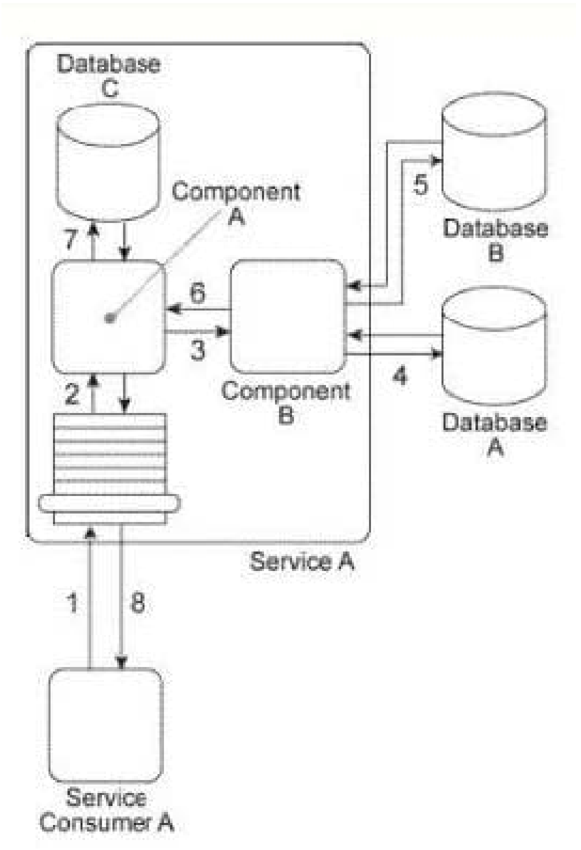
QUESTION 5

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