

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.leads4pass.com/s90-09.html>

100% Passing Guarantee
100% Money Back Assurance

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

-  **Instant Download** After Purchase
-  **100% Money Back** Guarantee
-  **365 Days** Free Update
-  **800,000+** Satisfied Customers



QUESTION 1

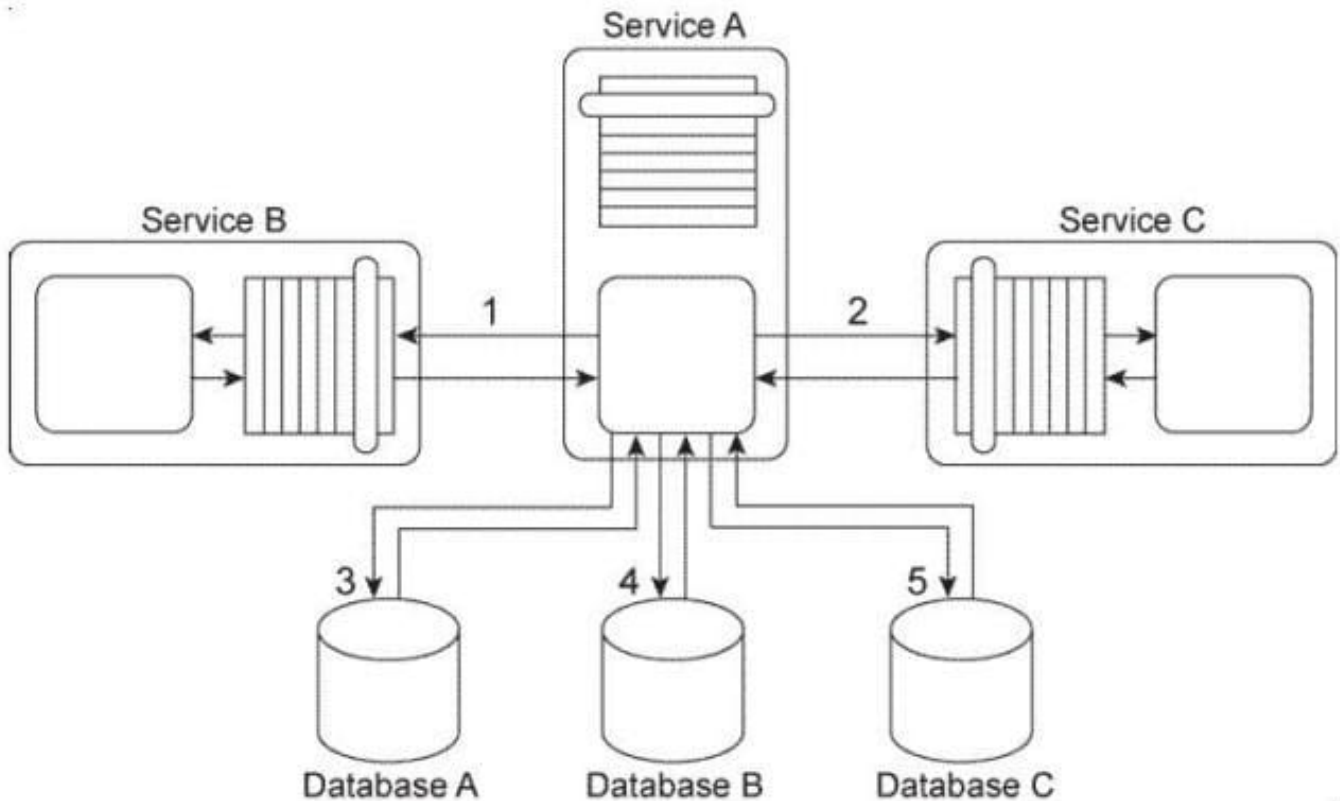
Service A is an entity service that provides a set of generic and reusable service capabilities. In order to carry out the functionality of any one of its service capabilities, Service A is required to compose Service B

(1) and Service C (2) and Service A is required to access Database A (3), Database B (4), and Database C (5). These three databases are shared by other applications within the IT enterprise.

All of service capabilities provided by Service A are synchronous, which means that for each request a service consumer makes. Service A is required to issue a response message after all of the processing has completed.

Depending on the nature of the service consumer request, Service A may be required to hold data it receives in memory until its underlying processing completes. This includes data it may receive from either Service A or Service B or from any of the three shared databases.

Service A is one of many entity services that reside in a highly normalized service inventory. Because Service A provides agnostic logic, it is heavily reused and is currently part of many service compositions.



You are told that Service A has recently become unstable and unreliable. The problem has been traced to two issues with the current service architecture. First, Service B, which is also an entity service, is being increasingly reused and has itself become unstable and unreliable. When Service B fails, the failure is carried over to Service A . Secondly, shared Database B has a complex data model. Some of the queries issued by Service A to shared Database B can take a very long time to complete. What steps can be taken to solve these problems without compromising the normalization of the service inventory?

A. The Redundant Implementation pattern can be applied to Service A, thereby making duplicate deployments of the service available. This way, when one implementation of Service A is too busy, another implementation can be accessed by service consumers instead. The Service Data Replication pattern can be applied to establish a dedicated

database that contains an exact copy of the data from shared Database B that is required by Service A .

B. The Redundant Implementation pattern can be applied to Service B, thereby making duplicate deployments of the service available. This way, when one implementation of Service B is too busy, another implementation can be accessed by Service A instead. The Service Data Replication pattern can be applied to establish a dedicated database that contains an exact copy of the data from shared Database B that is required by Service A .

C. The Redundant Implementation pattern can be applied to Service B, thereby making duplicate deployments of the service available. This way, when one implementation of Service B is too busy, another implementation can be accessed by Service A instead. The Service Data Replication pattern can be applied to establish a dedicated database that contains a copy of the data from shared Database B that is required by Service A . The replicated database is designed with an optimized data model in order to improve query execution performance.

D. None of the above.

Correct Answer: C

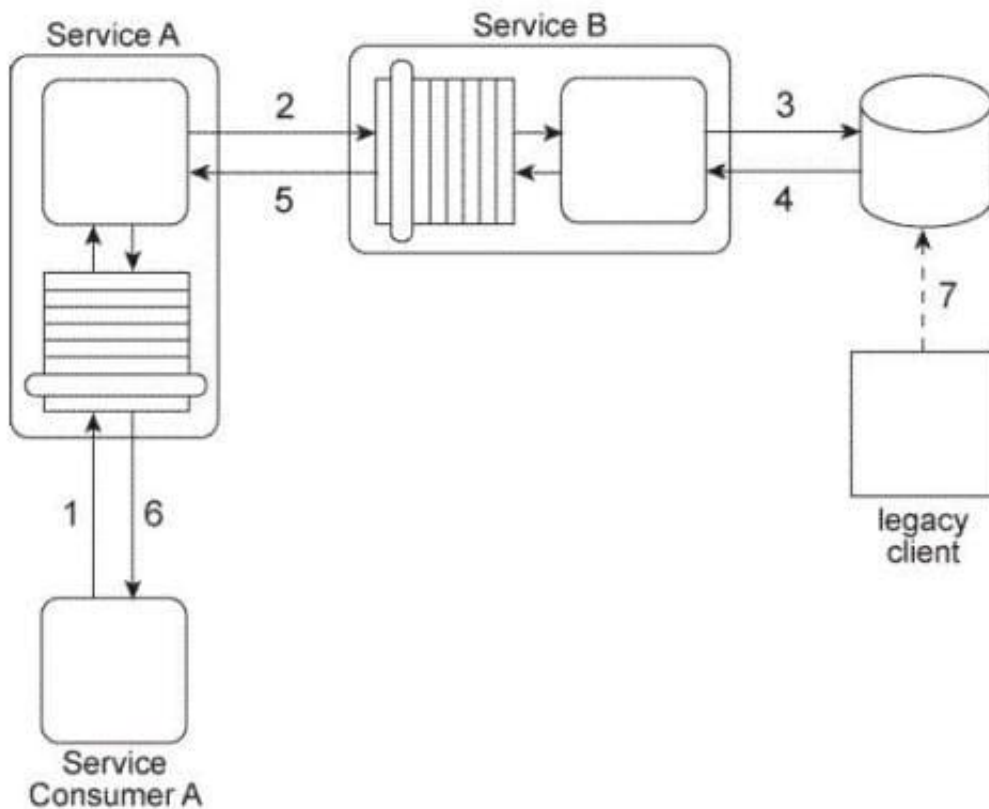
QUESTION 2

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

QUESTION 3

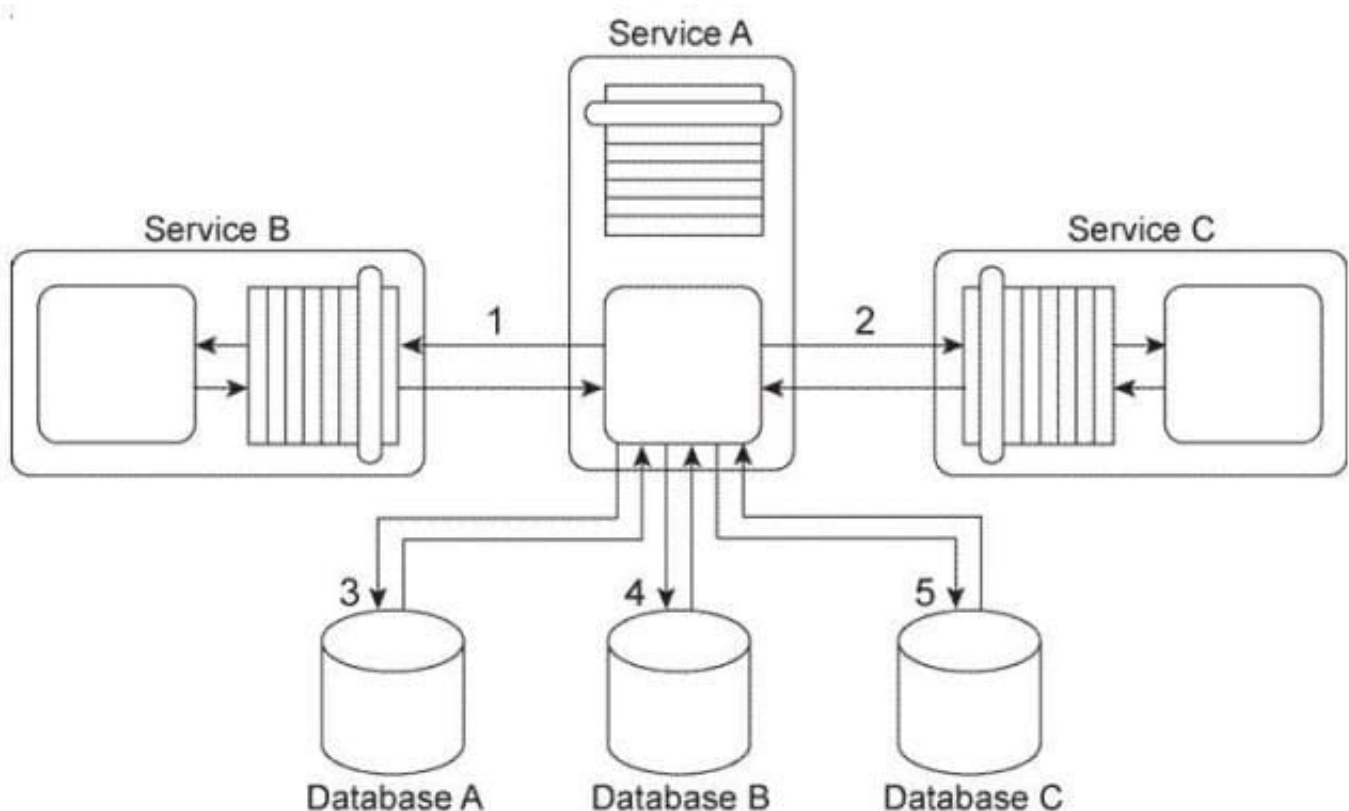
Service A is an entity service that provides a set of generic and reusable service capabilities. In order to carry out the functionality of any one of its service capabilities, Service A is required to compose Service B

(1) and Service C (2) and Service A is required to access Database A (3), Database B (4), and Database C (5). These three databases are shared by other applications within the IT enterprise.

All of service capabilities provided by Service A are synchronous, which means that for each request a service consumer makes. Service A is required to issue a response message after all of the processing has completed.

Depending on the nature of the service consumer request, Service A may be required to hold data it receives in memory until its underlying processing completes. This includes data it may receive from either Service A or Service B or from any of the three shared databases.

Service A is one of many entity services that reside in a highly normalized service inventory. Because Service A provides agnostic logic, it is heavily reused and is currently part of many service compositions.



You are told that Service A has recently become unstable and unreliable and several of the service consumers that access it have had to raise runtime exceptions due to these problems. What steps can be taken to solve these problems without compromising the normalization of the service inventory?

A. The Service Autonomy principle can be applied to increase the physical isolation of Service A and to reduce dependencies Service A has on external resources. In support of this, the Service Data Replication pattern can be applied in order to establish a dedicated database that contains replicated data from shared Databases A, B, and C . Furthermore, the Redundant Implementation pattern can be applied so that the logic Service A requires from Services B and C can be redundantly placed inside of Service A . This way, Service A avoids having to separately compose Services B and C

B. The Service Statelessness principle can be applied with the help of the State Repository pattern in order to establish a state database that Service A can use to defer state data it may be required to hold for extended periods. The Service Autonomy principle can also be applied in order to increase the physical isolation of Service A and to reduce

dependencies Service A has on external resources. In support of this, the Service Data Replication pattern can be applied in order to establish a dedicated database that contains replicated data from shared Databases A, B, and C.

C. The Service Loose Coupling and Standardized Service Contract principles can be applied by introducing a separate utility service that provides centralized data access to the Databases A, B, and C, and exposes a standardized service contract that can be used by Service A. This will prevent Service A from direct dependencies on the shared databases in case any of them are replaced in the future. By following this approach, the Legacy Wrapper pattern is effectively applied via the introduction of the new utility service.

D. None of the above.

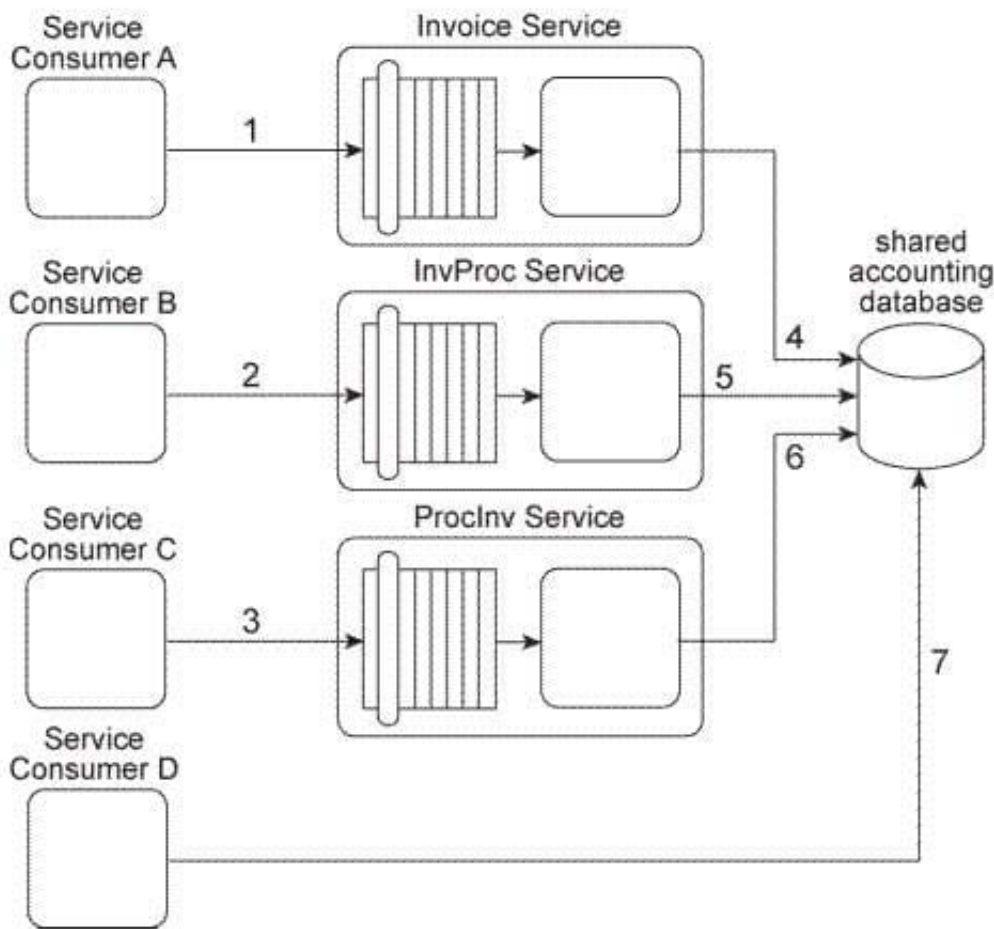
Correct Answer: B

QUESTION 4

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 5

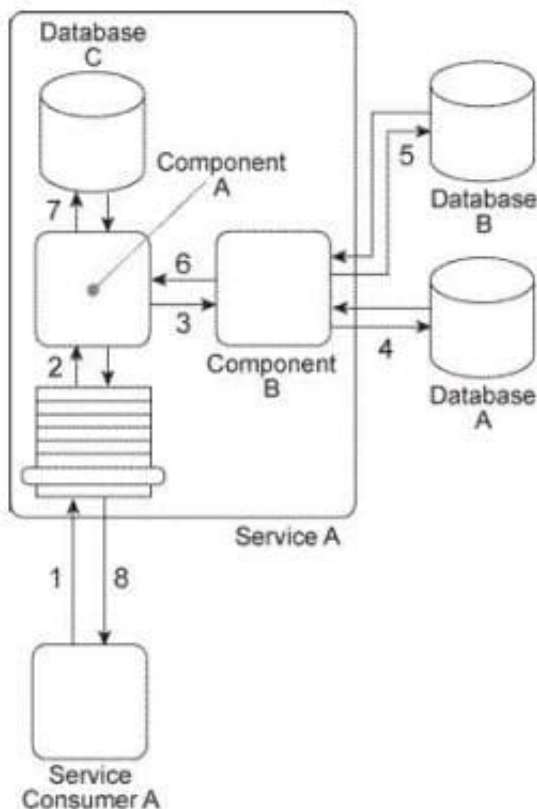
Service Consumer A sends Service A a message containing a business document (1). The business document is

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: 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

[S90.09 PDF Dumps](#)

[S90.09 VCE Dumps](#)

[S90.09 Study Guide](#)