



# S90.09<sup>Q&As</sup>

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

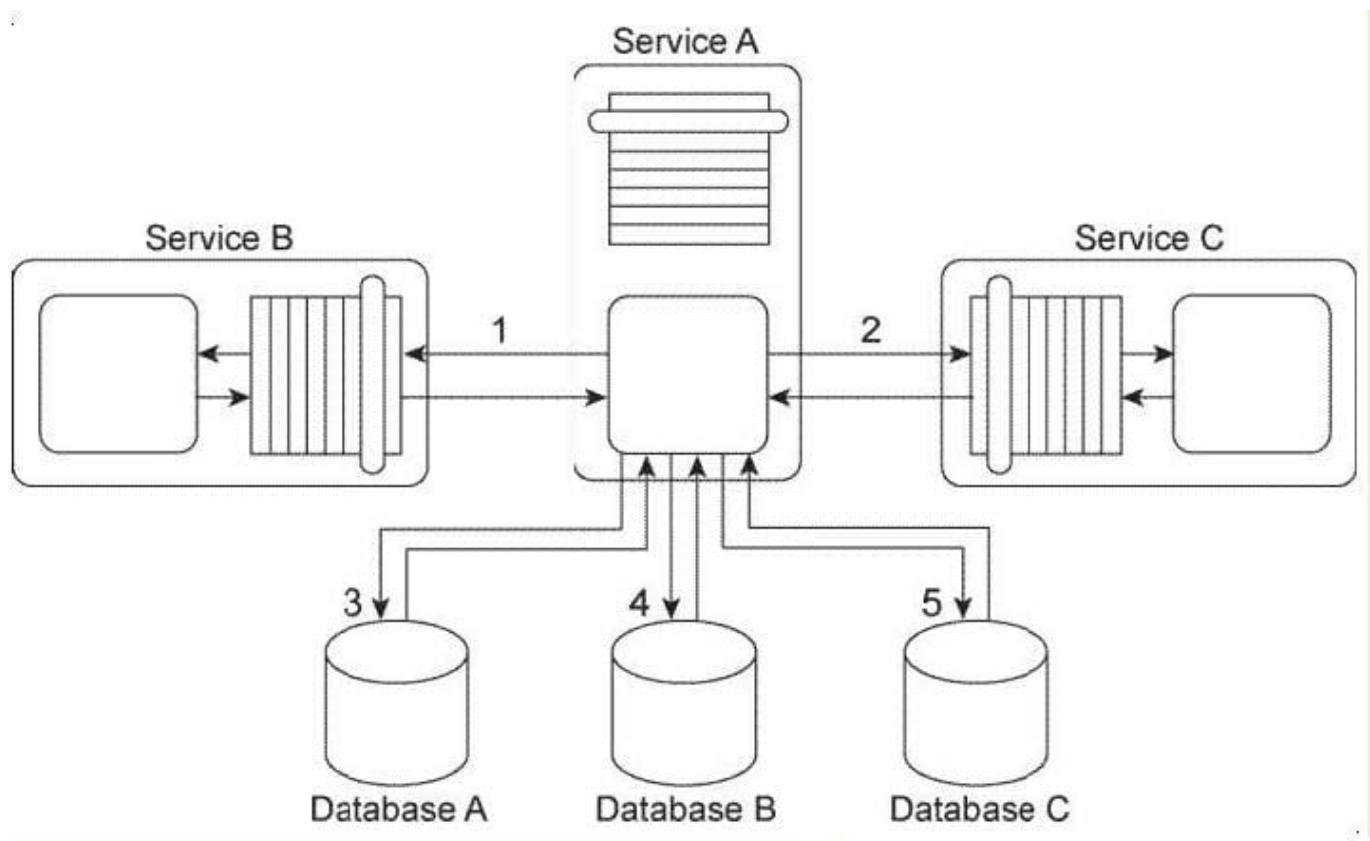
Following Questions and Answers are all new published by SOA Official Exam Center

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



**QUESTION 1**

You are told that in this service composition architecture, all four services are exchanging invoice-related data in an XML format. The services in Service Inventory A are standardized to use a specific XML schema for invoice data. Design standards were not applied to the service contracts used in Service Inventory B, which means that each service uses a different XML schema for the same kind of data. Database A and Database B can only accept data in the Comma Separated Value (CSV) format and therefore cannot accept XML formatted data. 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 A and Service C, and between Service C and Service D . The Data Format Transformation pattern can be applied so that data format transformation logic is positioned between the Service B logic and Database A and between the Service D logic and Database B.

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

the Service B logic and Database A 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 C . The Protocol Bridging pattern can be applied so that protocol bridging logic is positioned between Service A and Service B and between the Service C and Service D . The Data Format Transformation pattern can be applied so that data format transformation logic is positioned between the Service B logic and Database A and between the Service D logic and Database B.

D. None of the above.

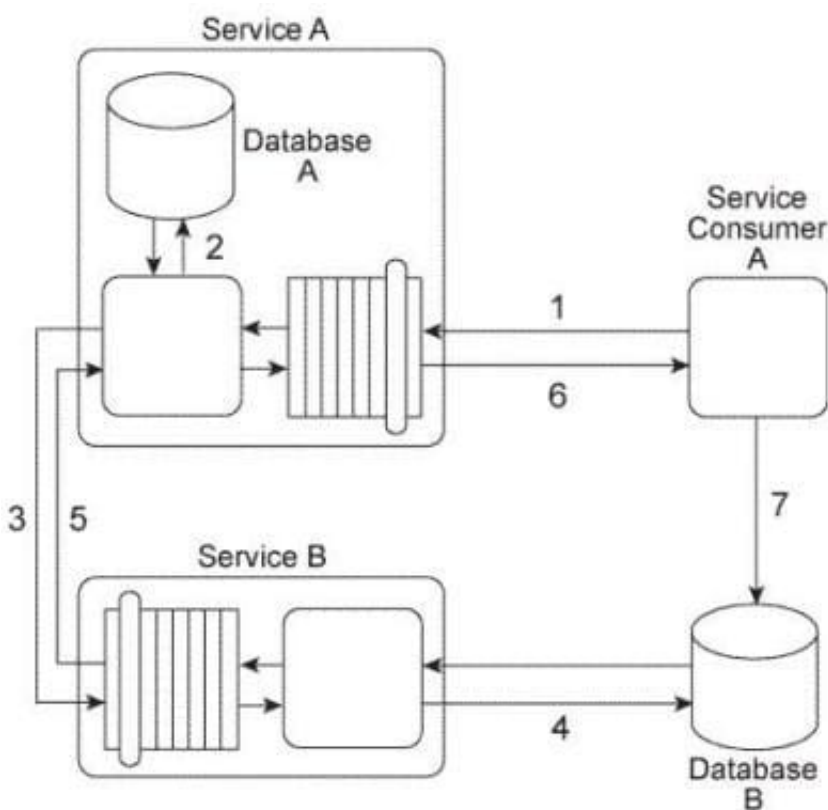


Correct Answer: A

## 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

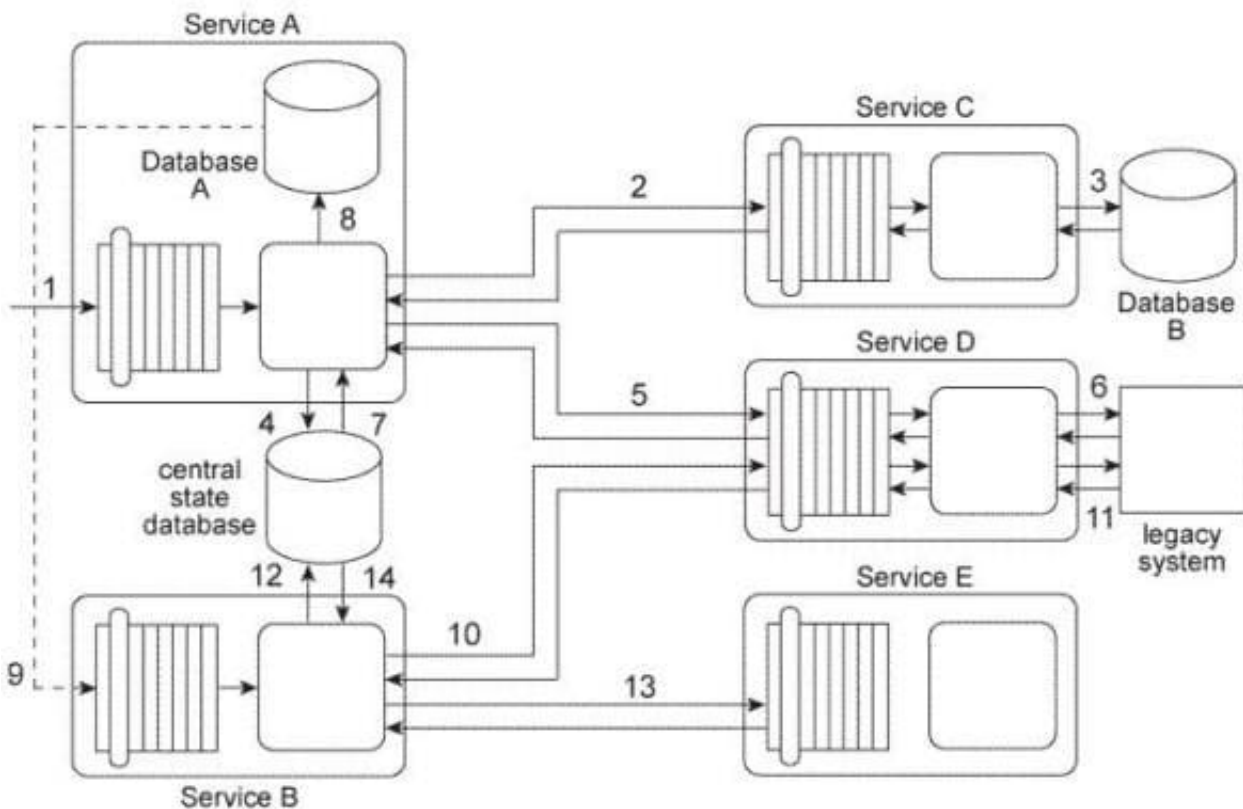
### QUESTION 3

Service A is an orchestrated task service that is invoked by a separate composition initiator (1) and then sends a request message to Service C (2). Service C queries Database B to retrieve a large data record

(3) and provides this data in a response message that is sent back to Service A. Service A temporarily stores this data in a central state database (4) and then sends a request message to Service D (5), which accesses a legacy system API to retrieve a data value (6). Service D then sends this data value in a response message back to Service A. The data in the state database is subsequently retrieved by Service A (7) and merged with the newly received data value. This combined data is written to Database A (8), which triggers an event that results in the invocation of Service B (9).

Service B is an orchestrated task service that sends a request message to Service D (10), which accesses a legacy system API to retrieve a data value (11) and then sends this data value in a response message back to Service B. Service B temporarily stores this data in a central state database (12) and then sends a request message to Service E (13), which performs a runtime calculation and then responds with the calculated data value back to Service B. The data in the state database is then retrieved by Service B (14) and merged with the calculated data value. Service B then uses the merged data to complete its business task.

The following specific problems and requirements exist:





Upon reviewing these requirements it becomes evident to you that the Enterprise Service Bus 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 Enterprise Service Bus 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 Enterprise Service Bus 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: HLM

---

#### QUESTION 4

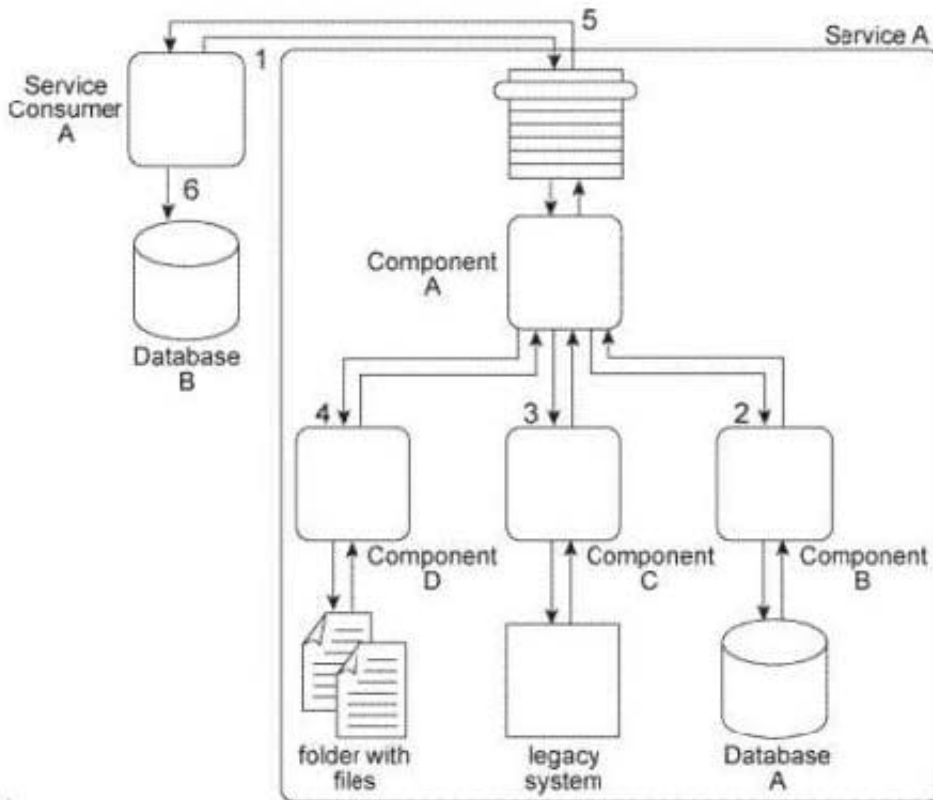
When Service A receives a message from Service Consumer A(1),the message is processed by Component A. This component first invokes Component B (2), which uses values from the message to query Database A in order to retrieve additional data. Component B then returns the additional data to Component A.

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



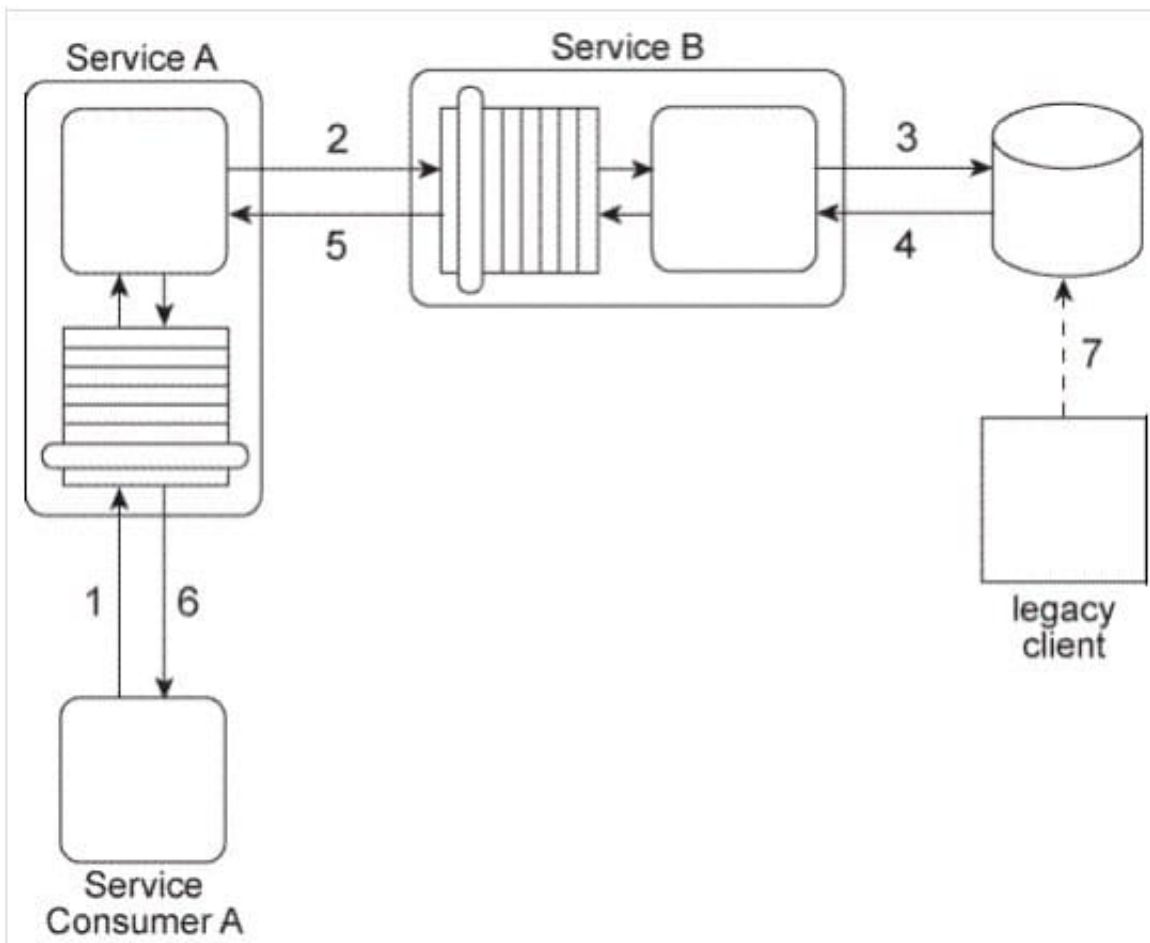
of having to access the shared Database A directly. The Legacy Wrapper pattern can be applied so that Component C is separated into a separate service that acts as a wrapper of the legacy system API. Next, the Asynchronous Queuing pattern can be applied so that a messaging queue is positioned between 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 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

[Latest S90.09 Dumps](#)

[S90.09 VCE Dumps](#)

[S90.09 Exam Questions](#)