

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



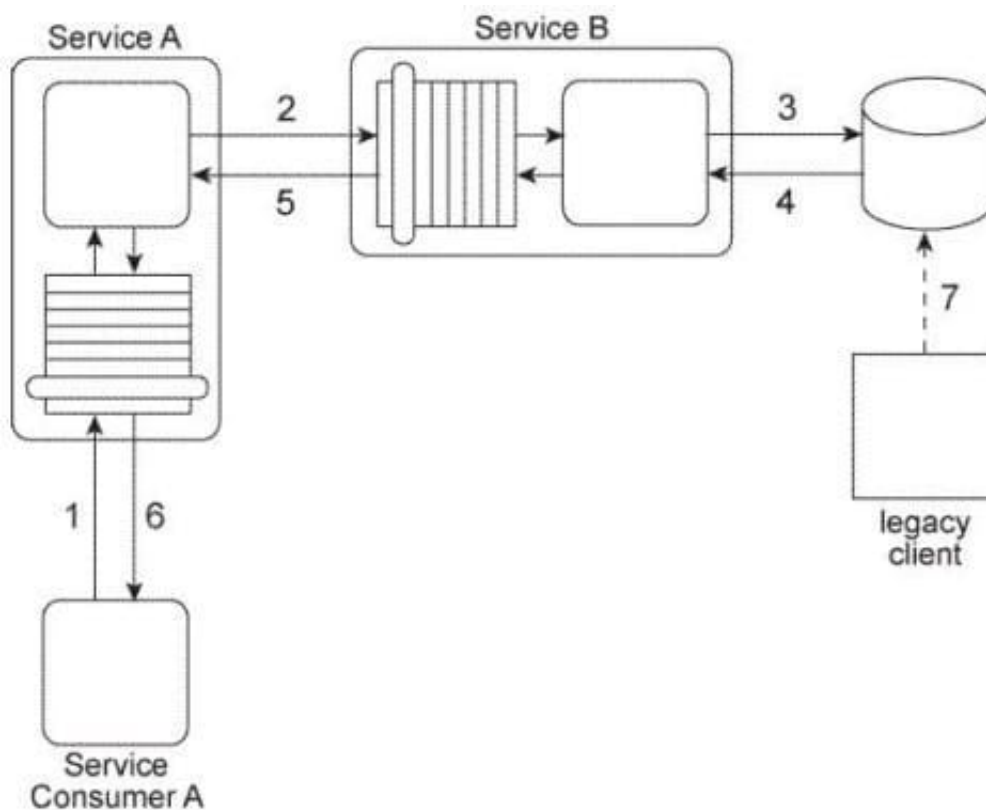
QUESTION 1

Service A is an entity service that provides a 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.



Because Service A and Service B are not always available at the same times, messages are getting lost and several invocation attempts by Service Consumer A fail. What steps can be taken to solve this problem?

- A. 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 .
- B. The Asynchronous Queuing pattern can be applied so that a messaging queue is established between Service A and Service B . This way, messages are never lost due to the unavailability of Service A or Service B . The Service Agent pattern can be further applied to establish a service agent that makes a log entry and issues a notification when re-transmission attempts by the messaging queue exceeds a pre-determined quantity.

C. The Asynchronous Queuing pattern can be applied so that a messaging queue is established between Service Consumer A and Service A. This way, messages are never lost due to the unavailability of Service A or Service B. The Service Agent pattern can be further applied to establish a service agent that makes a log entry each time a runtime exception occurs.

D. None of the above.

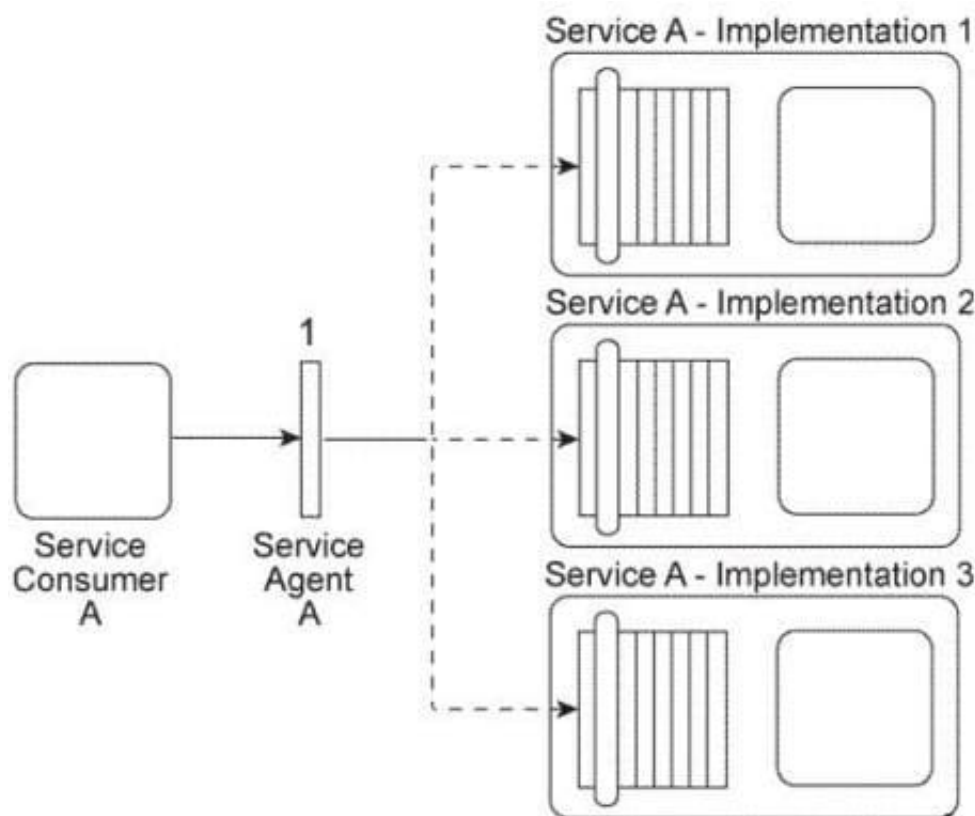
Correct Answer: A

QUESTION 2

Service Consumer A sends a message to Service A. There are currently three duplicate implementations of Service A (Implementation 1, Implementation 2, Implementation 3).

The message sent by Service Consumer A is intercepted by Service Agent A (1), which determines at runtime which implementation of Service A to forward the message to.

All three implementations of Service A reside on the same physical server.



You are told that despite the fact that duplicate implementations of Service A exist, performance is still poor at times. Also, you are informed that a new service capability will soon need to be added to Service A that will introduce functionality that will require access to a shared database that is used by many other clients and applications in the IT enterprise. This is expected to add further performance demands on Service A. How can this service architecture be changed to improve performance in preparation for the addition of the new service capability?

A. The Standardized Service Contract principle is applied to ensure that the new service capability extends the existing service contract in a manner that is compliant with current design standards. The Redundant Implementation pattern is

applied to establish separate implementations of Service A that include duplicate databases with copies of the data that Service A requires from the shared database.

B. The Service Autonomy principle is applied to further isolate the individual implementations of Service A by separating them onto different physical servers. When the new service capability is added, the Service Data Replication pattern is applied to give each implementation of Service A its own copy of the data it requires from the shared database.

C. The Service Loose Coupling principle is applied together with the Standardized Service Contract principle to ensure that Service Consumer A is not indirectly coupled to the shared database after the new service capability is added to the service contract. The Legacy Wrapper pattern can be applied to establish a new utility service that will provide standardized data access service capabilities for the shared database.

D. None of the above.

Correct Answer: B

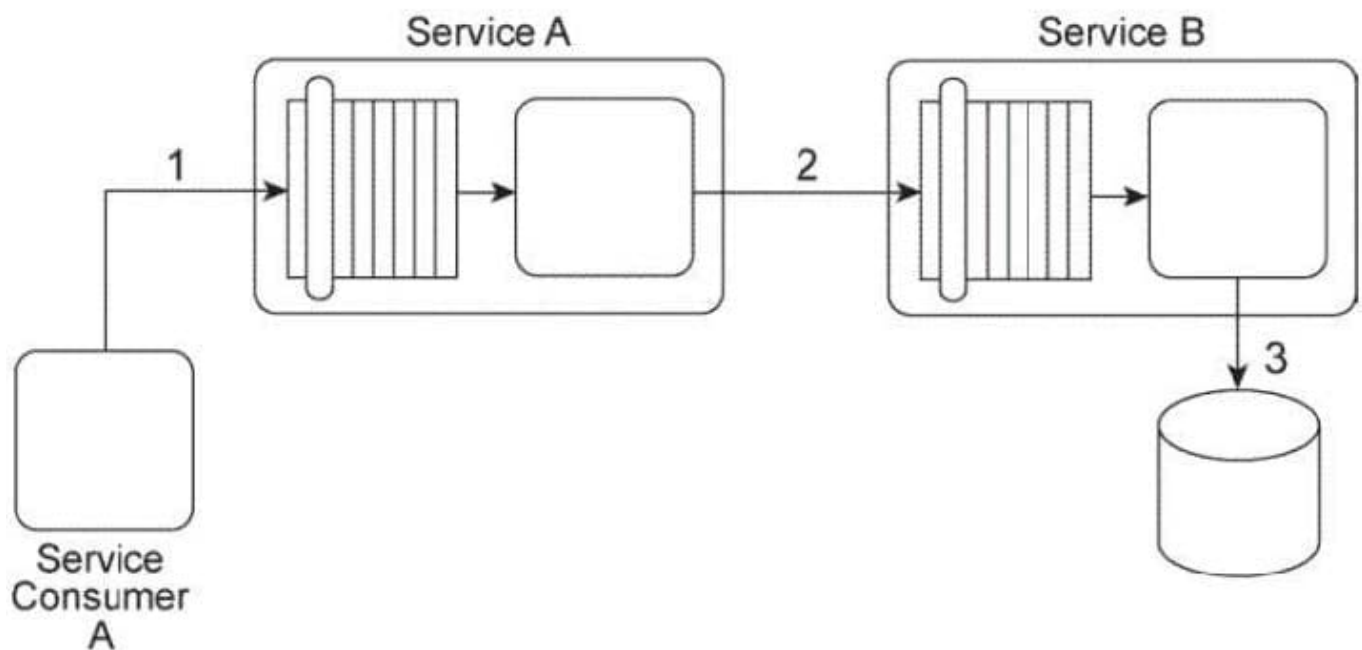
QUESTION 3

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.

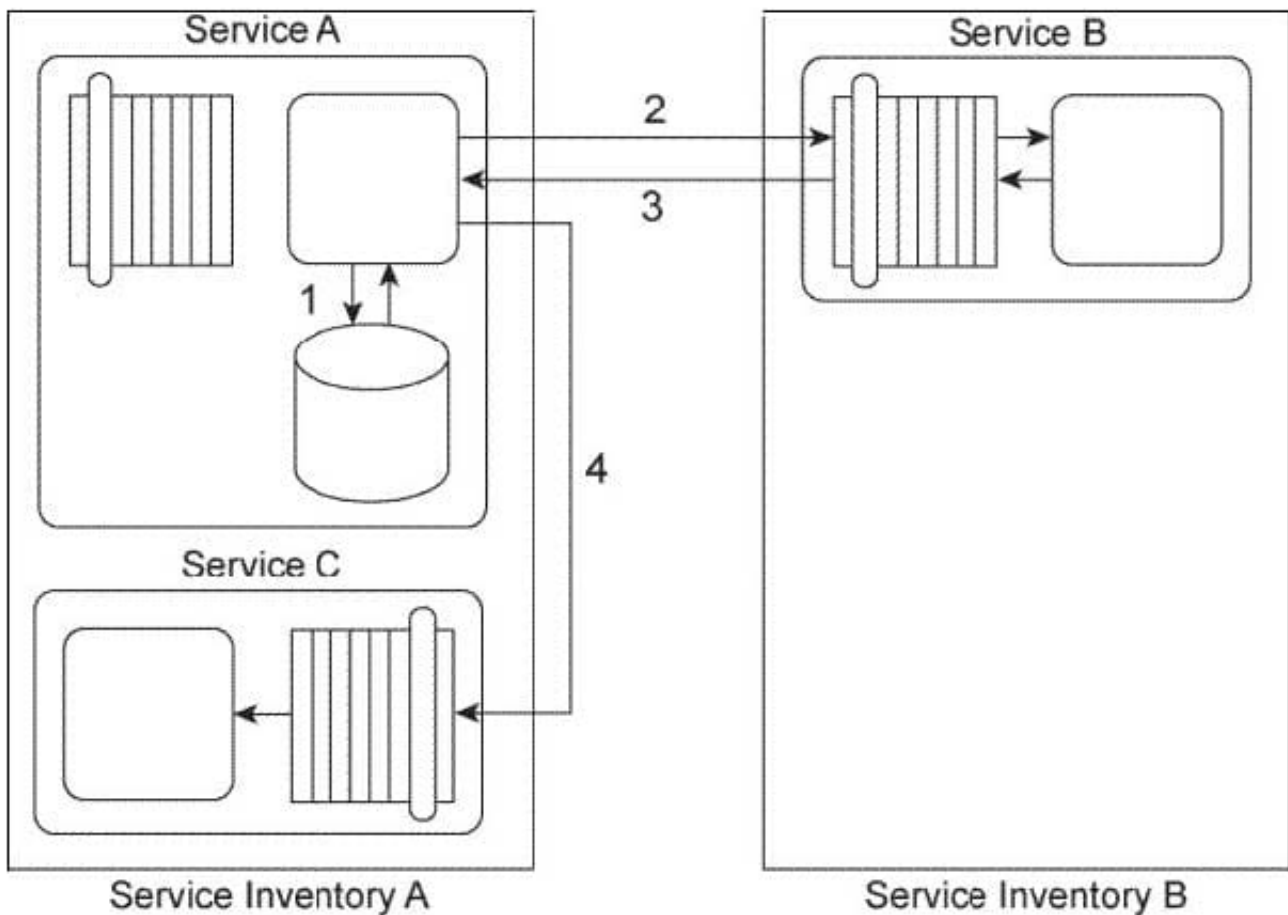
Correct Answer: B

QUESTION 4

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 were designed with service contracts based on different design standards than the services in Service Inventory B. As a result, Service A and Service B use different data models to represent the data they need to exchange. Therefore, Service A and Service B cannot currently communicate. Furthermore, Service C is an agnostic service that is heavily accessed by many concurrent service consumers. Service C frequently reaches its usage thresholds during which it is not available and messages sent to it are not received. How can this service composition architecture be changed to avoid these problems?

A. The Data Model Transformation pattern can be applied by establishing an intermediate processing layer between Service A and Service B that can transform a message from one data model to another at runtime. The Intermediate Routing and Service Agent patterns can be applied so that when Service B sends a response message, a service agent can intercept the message and, based on its contents, either forward the message to Service A or route the message to Service C. The Service Autonomy principle can be further applied to Service C together with the Redundant Implementation pattern to help establish a more reliable and scalable service architecture.

B. The Data Model Transformation pattern can be applied by establishing an intermediate processing layer between Service A and Service B that can transform a message from one data model to another at runtime. The Asynchronous Queuing pattern can be applied to establish an intermediate queue between Service A and Service C so that when Service A needs to send a message to Service C, the queue will store the message and retransmit it to Service C until it is successfully delivered. The Service Autonomy principle can be further applied to Service C together with the Redundant Implementation pattern to help establish a more reliable and scalable service architecture.

C. The Data Model Transformation pattern can be applied by establishing an intermediate processing layer between Service A and Service B that can transform a message from one data model to another at runtime. The Intermediate Routing and Service Agent patterns can be applied so that when Service B sends a response message, a service agent can intercept the message and, based on its contents, either forward the message to Service A or route the message to Service C. The Service Statelessness principle can be applied with the help of the State Repository pattern so that

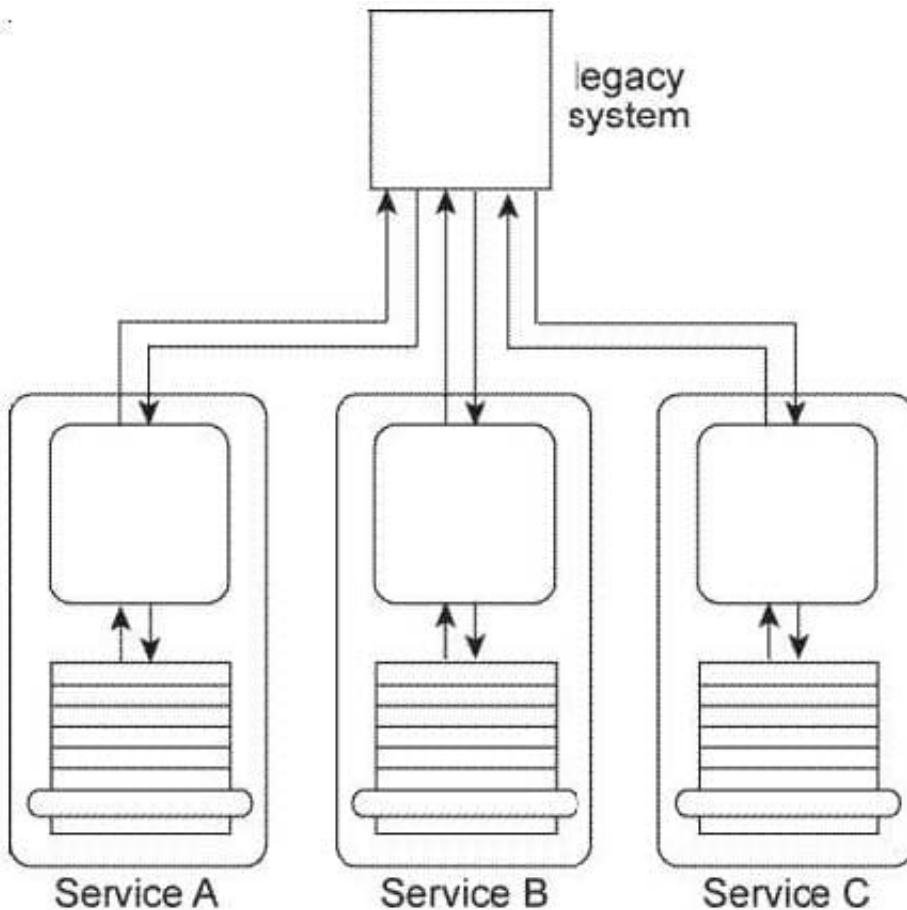
Service A can write the code value data to a state database while it is waiting for Service B to respond.

D. None of the above.

Correct Answer: B

QUESTION 5

Service A, Service B, and Service C are each designed to access the same shared legacy system. The service contracts for Service A, Service B, and Service C are standardized and decoupled from the underlying service logic. Service A and Service B are agnostic services that are frequently reused by different service compositions. Service C is a non-agnostic task service that requires access to the legacy system in order to retrieve business rules required for the service to make runtime decisions that determine its service composition logic. The legacy system uses a proprietary file format that Services A, B, and C need to convert to and from.



Service A is an agnostic utility service that is used by other services to gain access to the legacy system. Services B and C were not designed to access the legacy system via Service A because the Service A service contract was derived from the legacy system API and is therefore not standardized and exhibits negative contract-to-implementation coupling. You are told that additional services need to be created, all of which need access to the legacy system. You are also told that the legacy system may be replaced in the near future. What steps can be taken to ensure that the replacement of the legacy system has a minimal impact on Services B and C and any future services that are designed to rely upon it?

A. The Service Abstraction, Service Reusability, and Service Autonomy principles need to be applied in order to support

the application of the Official Endpoint pattern to Service A . This would position Service A as the official utility service through which the legacy system can be accessed. Service B will need to be redesigned to access Service A instead of accessing the legacy

system directly. Due to the dependency on business rules embedded within the legacy system the

option of applying the Rules Centralization pattern is not available. Service C will therefore need to

continue accessing the legacy system directly.

B. The Standardized Service Contract and Service Loose Coupling principles can be applied in order to establish a standardized service contract for Service A that will eliminate its negative contract coupling. Service B will need to be redesigned to access Service A instead of accessing the legacy system directly. Due to the dependency on business rules embedded within the legacy system the option of applying the Rules Centralization pattern is not available. Service C will therefore need to continue accessing the legacy system directly.

C. The Legacy Wrapper pattern can be applied together with the Standardized Service Contract principle in order to establish a standardized service contract for Service A that will eliminate its negative contract coupling. The Official Endpoint pattern can then be applied to position Service A as the official utility service through which the legacy system can be accessed. Services B and C will need to be redesigned to access Service A instead of accessing the legacy system directly.

D. None of the above.

Correct Answer: C

[S90.09 VCE Dumps](#)

[S90.09 Practice Test](#)

[S90.09 Exam Questions](#)