

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.pass2lead.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 an architect with a project team building services for Service Inventory A . You are told that no SLAs for Service B and Service C are available. You cannot determine how available these services will be, but it has been confirmed that both of these services support atomic transactions and the issuance of positive and negative acknowledgements. However, you also find out that the services in Service Inventory B use different data models than the services in Service Inventory A. Furthermore, recent testing results have shown that the performance of Service D is steady and reliable. However, Service D uses a different transport protocol than the services in Service Inventory A. The response time of Service A is not a primary concern, but Service Consumer A does need to be able to issue request messages to Service A 24 hours a day without disruption. What steps can be taken to fulfill these requirements?

A. The Event-Driven Messaging pattern is applied so that a subscriber-publisher relationship is established between Service Consumer A and Service A . This gives Service A the flexibility to provide its response to Service Consumer A whenever it is able to collect the three data values without having to require that Service Consumer A remain stateful. The Asynchronous Queuing pattern is applied so that a central messaging queue is positioned between Service A and Service B and between Service A and Service C . The Data Model Transformation and Protocol Bridging patterns are applied to enable communication between Service A and Service B and between Service A and Service C . The Service Autonomy principle is further applied to Service A in order to improve its overall runtime behavioral predictability.

B. The Reliable Messaging pattern is applied so that a system of acknowledgements is established between Service Consumer A and Service A . This gives Service A the flexibility to provide Service Consumer A with acknowledgements that indicate that the processing steps that are occurring between Service A and Service B, Service C, and Service D are progressing. The Asynchronous Queuing pattern is applied so that a central messaging queue is positioned between Service A and Service B and between Service A and Service C and between Service A and Service D . The Redundant Implementation pattern is applied so that a copy of Service D is brought in-Upon reviewing these requirements it becomes D with a standardized service contract that is in compliance with the design standards used in Service Inventory A.

C. The Asynchronous Queuing pattern is applied so that a central messaging queue is positioned between Service A and Service B and between Service A and Service C and between Service A and Service D and so that a separate messaging queue is positioned between Service A and Service Consumer A. The Data Model Transformation pattern is applied to enable communication between Service A and Service B and between Service A and Service C . The Protocol Bridging pattern is applied to enable communication between Service A and Service D .

D. None of the above.

Correct Answer: C

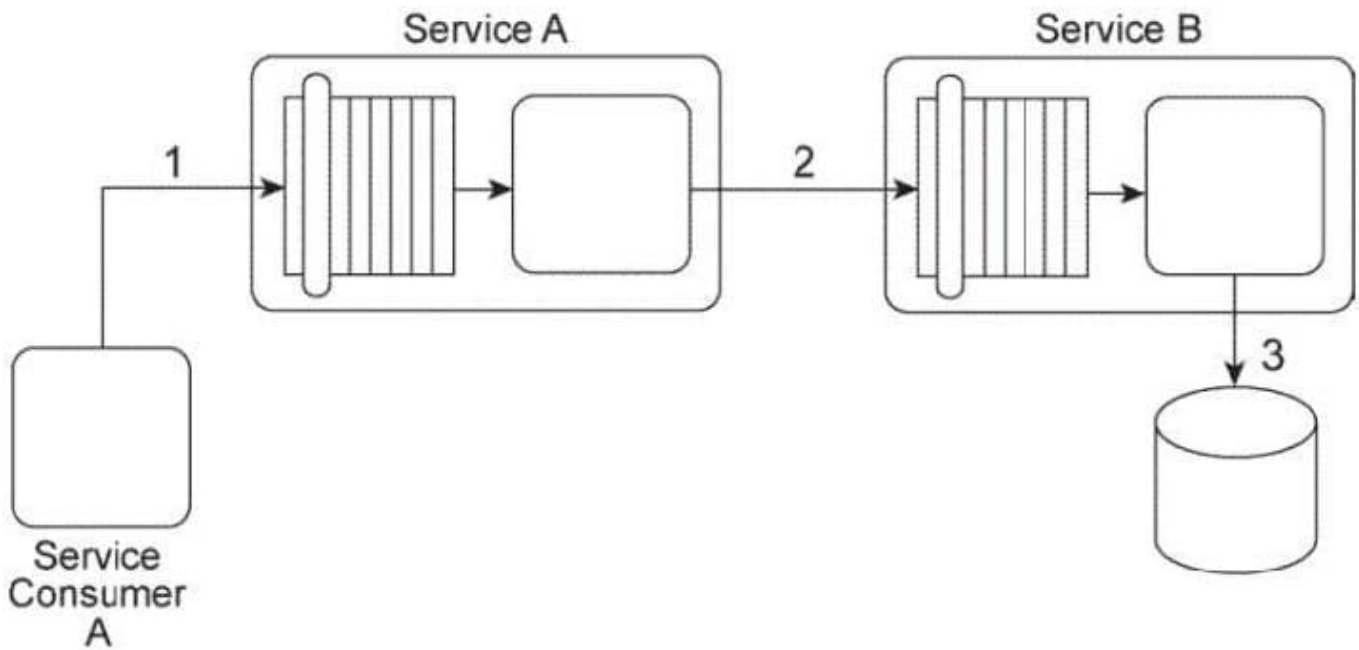
QUESTION 2

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 3

Upon reviewing these requirements it becomes evident to you that the Orchestration 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 Orchestration 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 Orchestration 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: CL

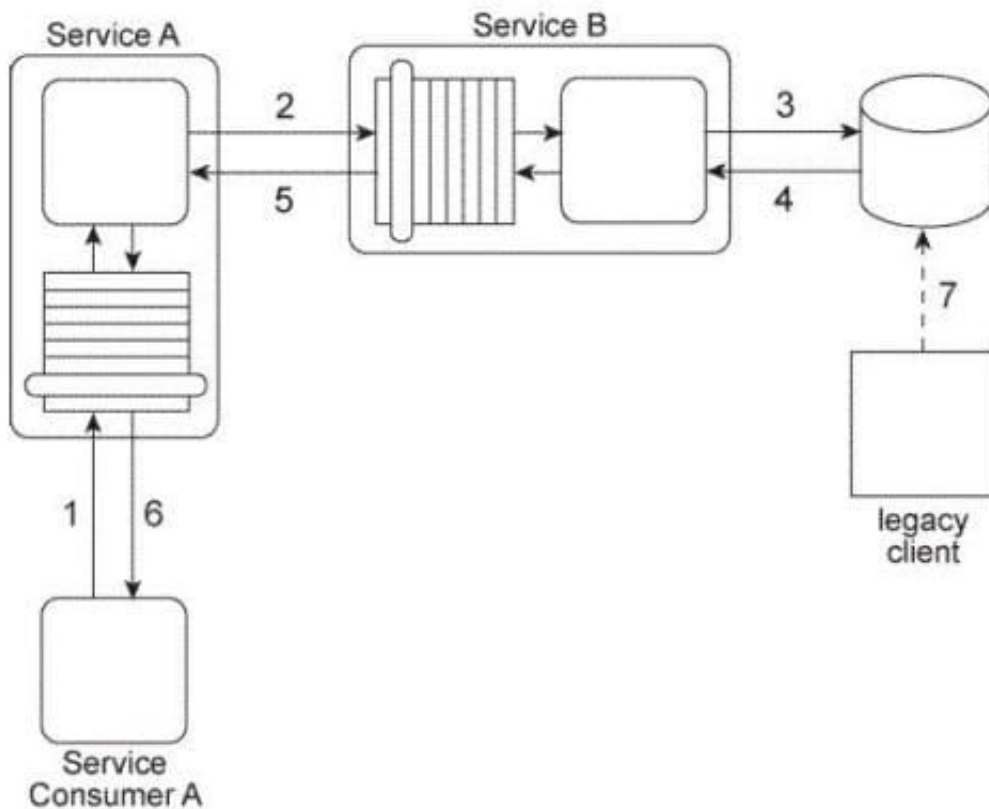
QUESTION 4

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.

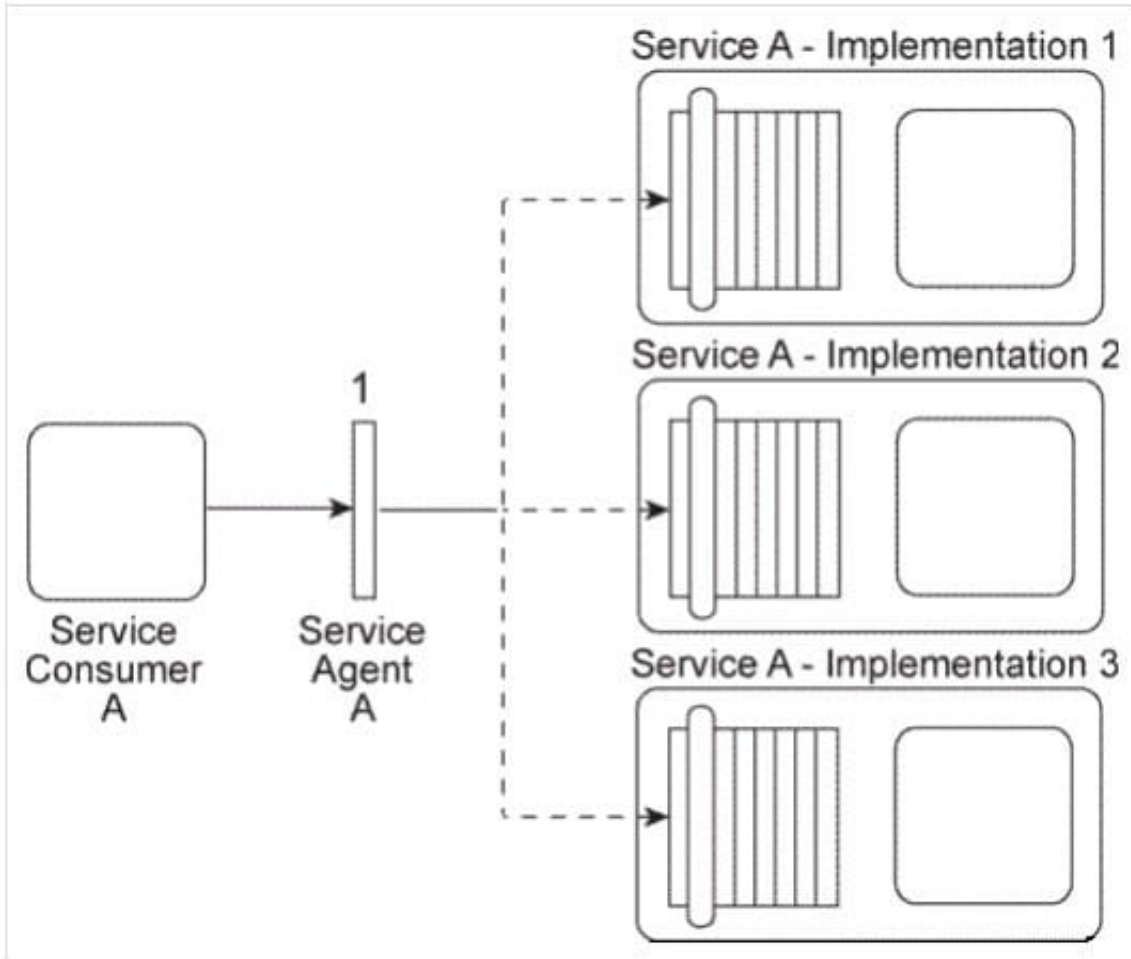
D. None of the above.

Correct Answer: D

QUESTION 5

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 .



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

[S90.09 PDF Dumps](#)

[S90.09 VCE Dumps](#)

[S90.09 Braindumps](#)