Number: S90.09
Passing Score: 800
Time Limit: 120 min

S90.09

SOA Design & Architecture Lab

Exam A

QUESTION 1

Currently, due to the increasing amount of concurrent access by service consumers, the runtime performance of both the Client and Vendor services has worsened and has therefore reduced their effectiveness as service composition members. Additionally, a review of the logic of both services has revealed that some of the business rules used by the Client and Vendor services are actually the same. What steps can be taken to improve performance and reduce redundant business rule logic?

A. The Rules Centralization pattern can be applied by extracting the business rule logic from the Client and Vendor services and placing it into a new Rules service, thereby reducing the redundancy of business rules logic. The Redundant Implementation pattern can then be applied to establish a scalable Rules service that is capable of supporting concurrent access from many service consumers.
B. The Redundant Implementation pattern can be applied to the Client and Vendor services, thereby establishing duplicate service implementations that can be accessed when a service reaches its runtime usage threshold. The Intermediate Routing pattern can be further applied to provide load balancing logic that can, at runtime, determine which of the redundant service implementations is the least busy for a given service consumer request.
C. The Rules Centralization pattern can be applied to isolate business rules logic into a central and reusable Rules service. Additionally, the Service Abstraction principle can be applied to hide the implementation details of new the Rules service. D. None of the above.

Correct Answer: A
Section: (none)
**Explanation**

Explanation/Reference:

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?
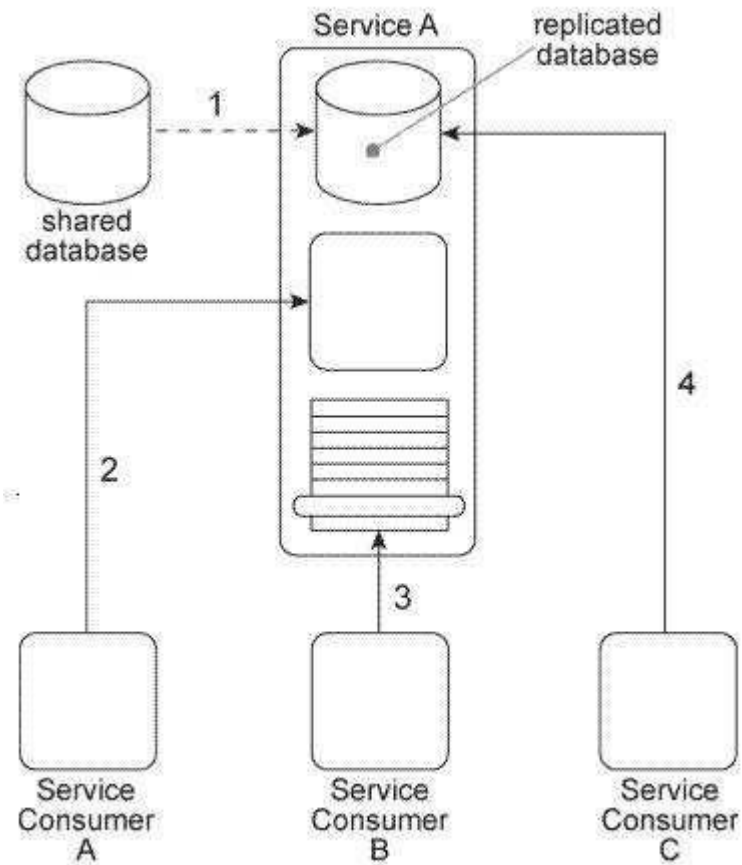
A. The Contract Centralization pattern can be applied to force all service consumers to access the Service A architecture via its published service contract. This will prevent negative forms of coupling that could lead to problems when the database is replaced. The Service Abstraction principle can then 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
Section: (none)
Explanation

Explanation/Reference:

QUESTION 3
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.

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).
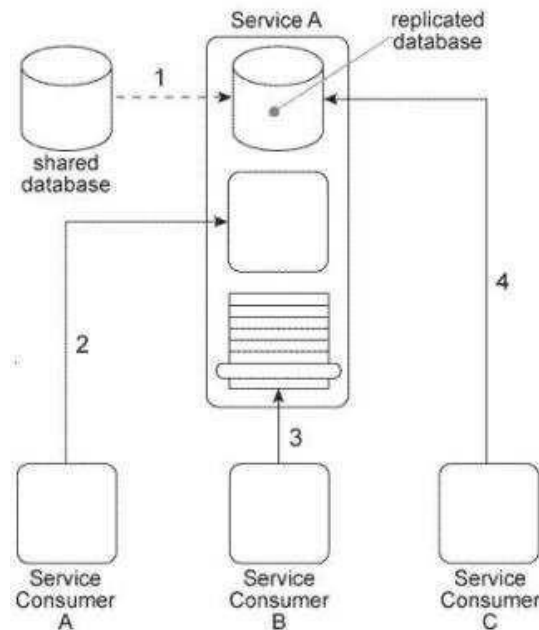
You've been told that the reason Service Consumers A and C bypass the published Service A service contract is because, for security reasons, they are not allowed to access a subset of the operations in the WSDL definition that expresses the service contract. How can the Service A architecture be changed to enforce these security restrictions while avoiding negative forms of coupling?

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 architecture 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 consumers to access the Service A architecture via its published service contract only. The Service Loose Coupling principle can then be applied to ensure that the centralized service contract does not contain any content that is dependent on or derived from the underlying service implementation.
C. The Concurrent Contracts pattern can be applied to Service A in order to establish one or more alternative service contracts. This allows service consumers with different levels of security clearance to continue accessing the service logic via its published service contracts. D. None of the above.
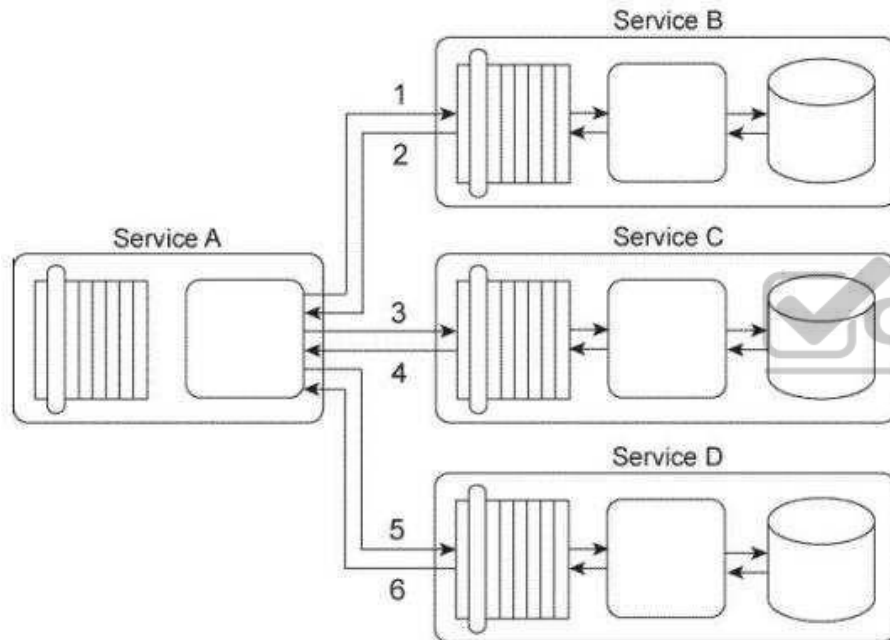
Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:

QUESTION 4
Service A is a task service that is required to carry out a series of updates to a set of databases in order to complete a task. To perform the database updates Service A must interact with three other services, each of which provides standardized data access capabilities.

Service A sends its first update request message to Service B (1), which then responds with a message containing a success or failure code (2). Service A then sends its second update request message to Service C (3), which also responds with a message containing a success or failure code (4). Finally, Service A sends a request message to Service D (5), which responds with its own message containing a success or failure code (6).

You've been given a requirement that all database updates must either be completed successfully or not at all. This means that if any of the three response messages received by Service A contain a failure code, all of the updates carried out until that point must be reversed. Note that if Service A does not receive a response message back from Services B, C, or D, it must assume that a failure has occurred. How can this service composition architecture be changed to fulfill these requirements?

A. The Reliable Messaging pattern can be applied to guarantee the delivery of positive or negative acknowledgements. This way, Service A will always be informed of whether a failure condition has occurred with any of the database updates performed by Services B, C, and D. Furthermore, the Service Loose Coupling

principle can be applied to ensure that the request and response messages exchanged by the services do not contain any implementation details that would indirectly couple Service A to any of the databases.

B. The Atomic Service Transaction pattern can be applied individually to Services B, C, and D so that each of these services performs its own database update within the scope of an atomic transaction. If anyone update fails, that change can be rolled back on that database. Furthermore, the Service Loose Coupling principle can be applied to ensure that Service A is kept out of the scope of the atomic transaction so that it is not negatively coupled to the proprietary database technologies that are required to enable the atomic transaction functionality.

C. The Compensating Service Transaction can be applied to Service A so that when any one response message containing a failure code is received by Service A, it can invoke exception handling logic that will log the failed database updates. The Service Loose Coupling principle can be further applied to ensure that Services B, C, or D are not indirectly coupled to the exception handling logic, especially if Service A requires additional access to Services B, C, or D in order to collect more information for logging purposes. D. None of the above.

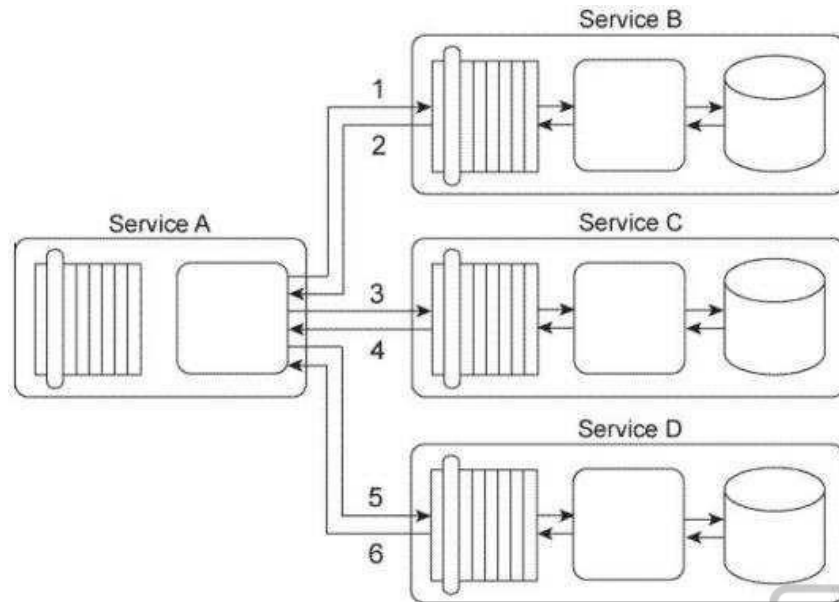**Correct Answer:** D
Section: (none)
Explanation

Explanation/Reference:

QUESTION 5
Service A is a task service that is required to carry out a series of updates to a set of databases in order to complete a task. To perform the database updates Service A must interact with three other services, each of which provides standardized data access capabilities.

Service A sends its first update request message to Service B (1), which then responds with a message containing a success or failure code (2). Service A then sends its second update request message to Service C (3), which also responds with a message containing a success or failure code (4). Finally, Service A sends a request message to Service D (5), which responds with its own message containing a success or failure code (6).

You've been asked to change this service composition architecture in order to fulfill a set of new requirements: First, if the database update performed by Service B fails, then it must be logged by Service A. Secondly, if the database update performed by Service C fails, then a notification e-mail must be sent out to a human administrator. Third, if the database update performed by either Service C or Service D fails, then both of these updates must be reversed so that the respective databases are restored back to their original states. What steps can be taken to fulfill these requirements?

A. Service A is updated to perform a logging routine when Service A receives a response message from Service B containing a failure code. Service A is further updated to send an e-mail notification to a human administrator if Service A receives a response message from Service C containing a failure code. The Atomic Service Transaction pattern is applied so that Services A, C, and D are encompassed in the scope of a transaction that will guarantee that if the database updates performed by either Service C or Service D fails, then both updates will be rolled back.

B. The Compensating Service Transaction pattern is applied to Service B so that it invokes exception handling logic that logs failed database updates before responding with a failure code back to Service A . Similarly, the Compensating Service Transaction pattern is applied to Service C so that it issues an e-mail notification to a human administrator when a database update fails. The Atomic Service Transaction pattern is applied so that Services A, C, and D are encompassed in the scope of a transaction that will guarantee that if the database updates performed by either Service C or Service D fails, then both updates will be rolled back. The Service Autonomy principle is further applied to Service A to ensure that it remains consistently available to carry out this sequence of actions.

C. The Atomic Service Transaction pattern is applied so that Services A, C, and D are encompassed in the scope of a transaction that will guarantee that if the database updates performed by either Service C or Service D fails, then both updates will be rolled back. The Compensating Service Transaction pattern is then applied to all services so that the scope of the compensating transaction includes the scope of the atomic transaction. The compensating exception logic

that is added to Service D automatically invokes Service B to log the failure condition and Service C to issue the e-mail notification to the human administrator. This way, it is guaranteed that the compensating logic is always executed together with the atomic transaction logic.

D. None of the above.

Correct Answer: A
Section: (none)
**Explanation**

Explanation/Reference:

QUESTION 6
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.

Service A

Service B

Service C

2

3

1

4

Service Inventory A

Service Inventory B
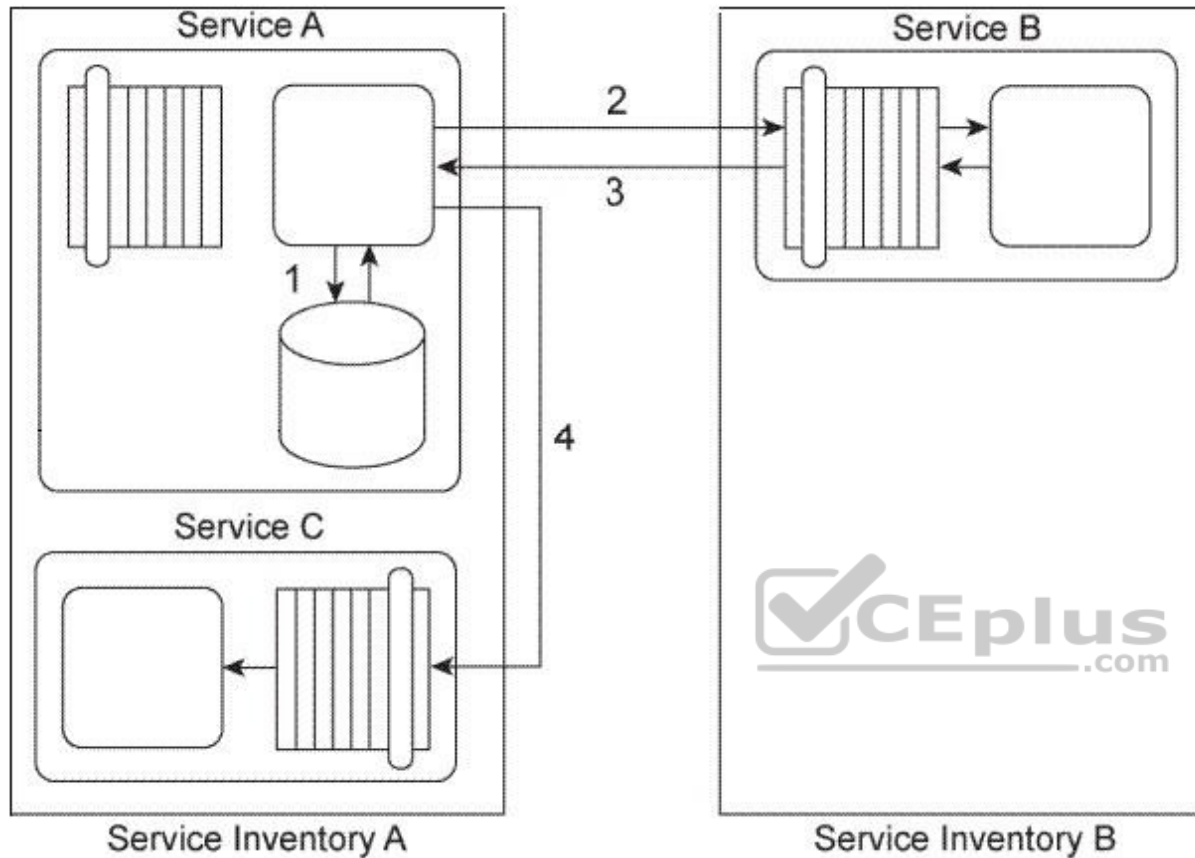
You are told that the services in Service Inventory A were designed with service contracts based on different design standards than the services in Service Inventory B. As a result, Service A and Service B use different data models to represent the data they need to exchange. Therefore, Service A and 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
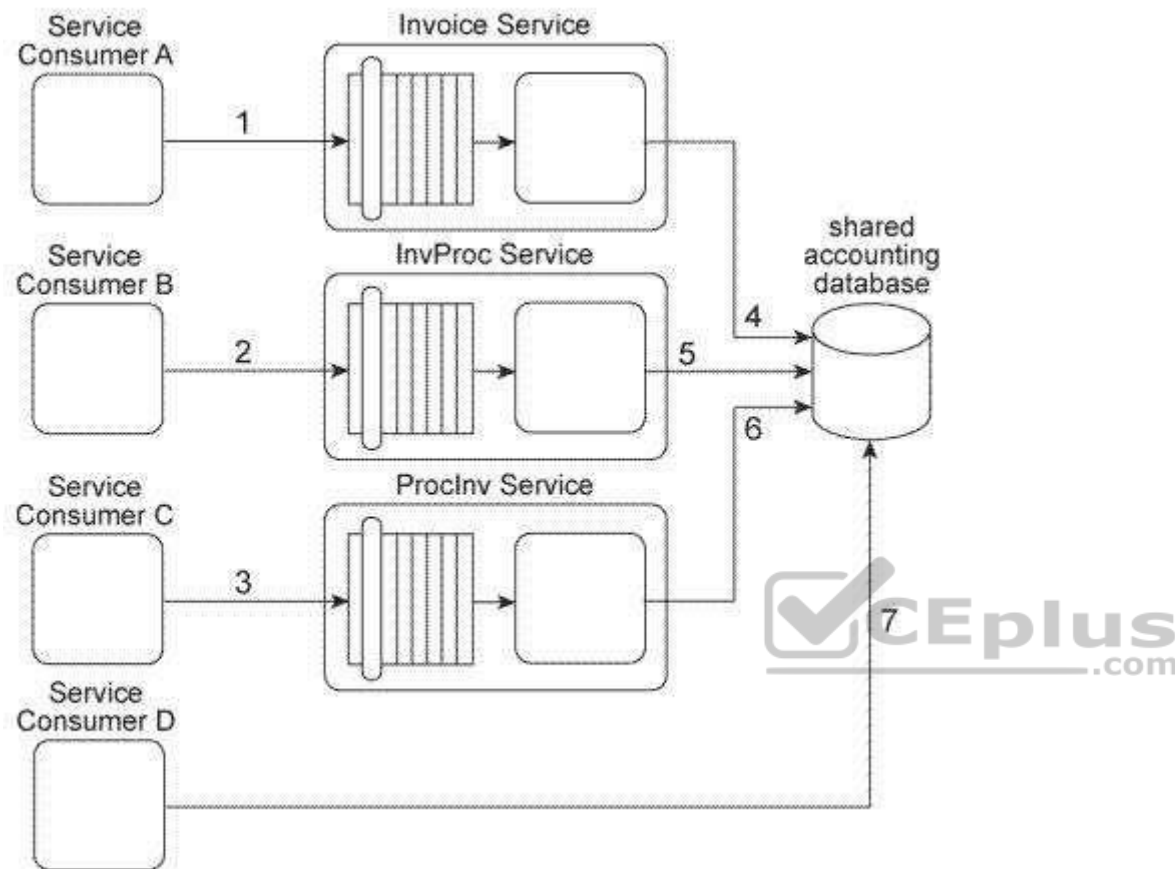**Section: (none)**
Explanation

Explanation/Reference:

QUESTION 7
Our service inventory contains the following three services that provide invoice-related data access capabilities: Invoice, InvProc, and ProcInv. 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 ProcInv 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.)

Assuming that the Invoice service, InvProc service, and ProcInv service are part of the same service inventory, what steps would be required to fully apply the Official Endpoint pattern?

A. One of the invoice-related services needs to be chosen as the official service providing invoice data access capabilities. Service Consumers A, B, and C then need to be redesigned to only access the chosen invoice-related service. Because Service Consumer D does not rely on an invoice-related service, it is not affected by the Official Endpoint pattern and can continue to access the accounting database directly. The Service Abstraction principle can be further applied to hide the existence of the shared accounting database and other implementation details from current and future service consumers.

B. One of the invoice-related services needs to be chosen as the official service providing invoice data access capabilities. Service Consumers A, B, and C then need to be redesigned to only access the chosen invoice-related service. Service Consumer D also needs to be redesigned to not access the shared

accounting database directly, but to also perform its data access by interacting with the official invoice-related service. The Service Abstraction principle can be further

applied to hide the existence of the shared accounting database and other implementation details from current and future service consumers.

C. Because Service Consumers A, B, and C are already carrying out their data access via published contracts, they are not affected by the Official Endpoint pattern. Service Consumer D needs to be redesigned to not access the shared accounting database directly, but to perform its data access by interacting with the official invoice-related service. The Service Abstraction principle can be further applied to hide the existence of the shared accounting database and other implementation details from current and future service consumers. D. None of the above.

Correct Answer: B
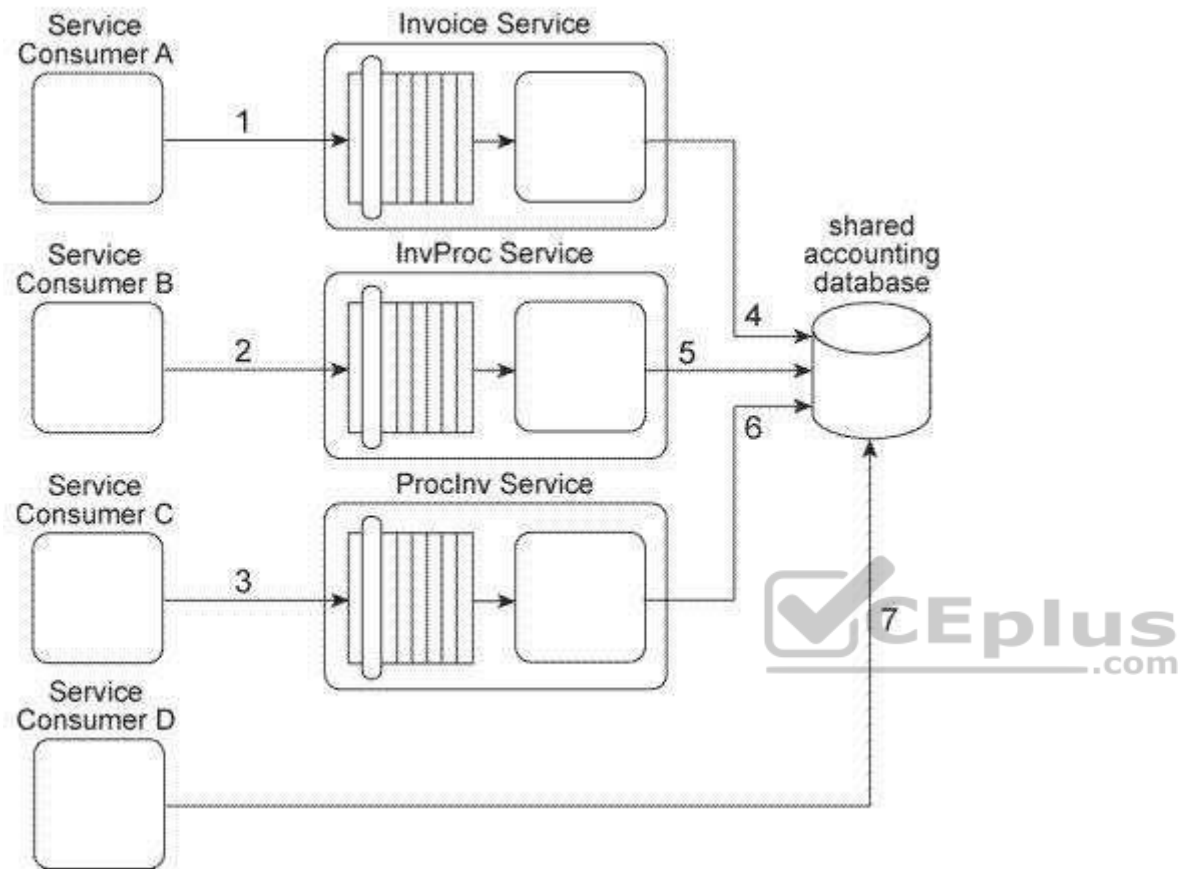**Section: (none)**
Explanation

Explanation/Reference:

QUESTION 8
Our service inventory contains the following three services that provide invoice-related data access capabilities: Invoice, InvProc, and ProcInv. 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 ProcInv 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 ProcInv 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 ProcInv 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
Section: (none)
Explanation
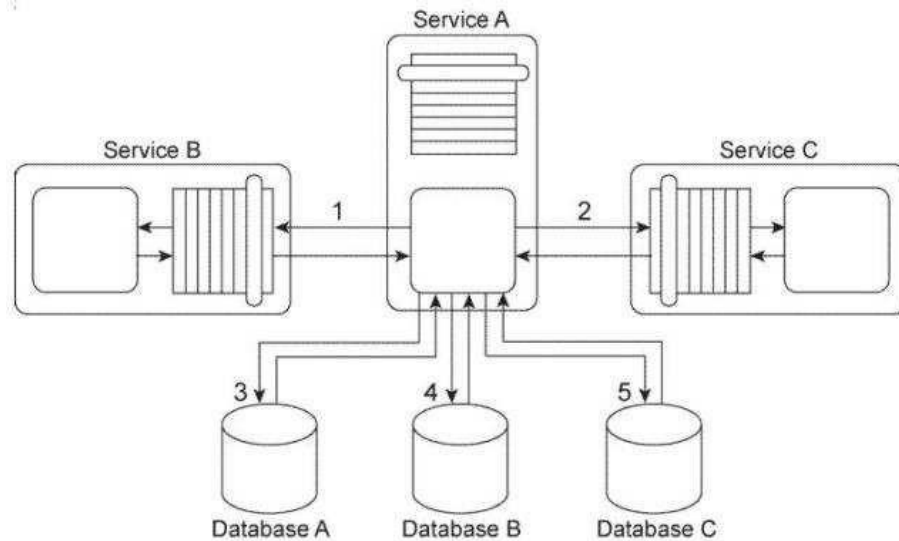
Explanation/Reference:

QUESTION 9
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.

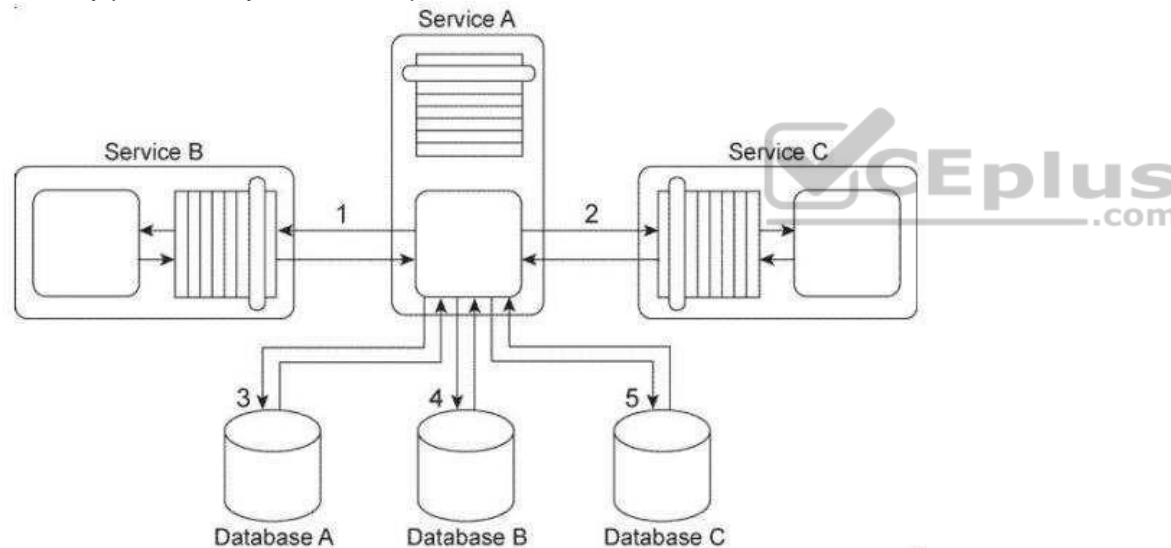**Correct Answer:** B
Section: (none)
Explanation

Explanation/Reference:

QUESTION 10
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
**Section: (none)**
Explanation

Explanation/Reference:

QUESTION 11
Services A, B, and C are non-agnostic task services. Service A and Service B use the same shared state database to defer their state data at runtime.

An assessment of these three services reveals that each contains some agnostic logic, but because it is bundled together with the non-agnostic logic, the agnostic logic cannot be made available for reuse.

The assessment also determines that because Service A and Service B and the shared state database are each located in physically separate environments, the remote communication required for Service A and Service B to interact with the shared state database is causing an unreasonable decrease in runtime performance.
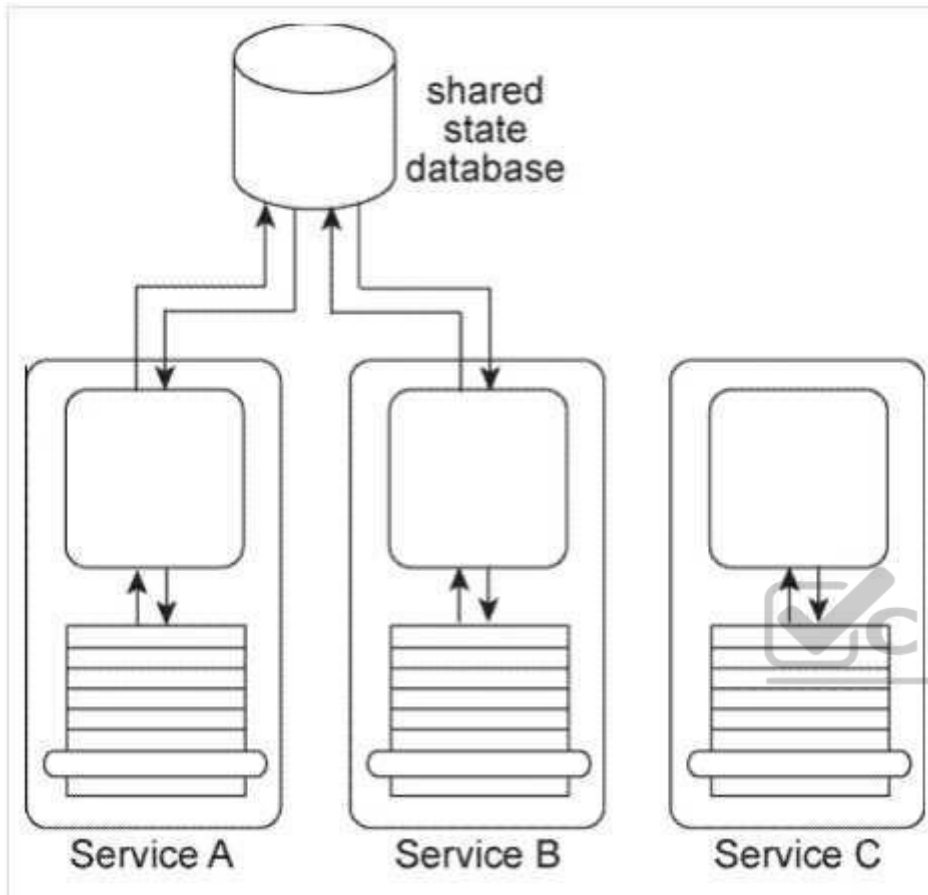
How can the application of the Orchestration pattern improve this architecture?

A. The application of the Orchestration pattern will result in an environment whereby the State Repository and Service Data Replication patterns are naturally applied, allowing the shared state database to be replicated for Services A and B so that each task service can have its own dedicated state database. The Process Centralization pattern can also be applied to Services A and B, so that their logic is physically centralized, turning them into orchestrated task services.

B. The application of the Orchestration pattern will result in an environment whereby the Process Abstraction and Process Centralization patterns are naturally applied to Services A, B, and C, resulting in a clean separation of non-agnostic task services from newly designed agnostic services with reuse potential. Also, the State Repository pattern can be applied by the availability of a central state database that can be shared by Services A and B. This database can be made available as a local part of the environment so that Services A and B can avoid remote communication.

C. The application of the Orchestration pattern will result in an environment whereby the Compensating Service Transaction is naturally applied, resulting in the opportunity to create sophisticated exception logic that can be used to compensate for the performance problems caused by Services A and B having to remotely access the state database. The Process Abstraction and Service Broker patterns are also naturally applied, enabling the separation of non-agnostic logic and agnostic logic while providing common transformation functions required to overcome any disparity in the service contracts that will need to be created for the new agnostic services. D. None of the above.

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:

QUESTION 12
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.
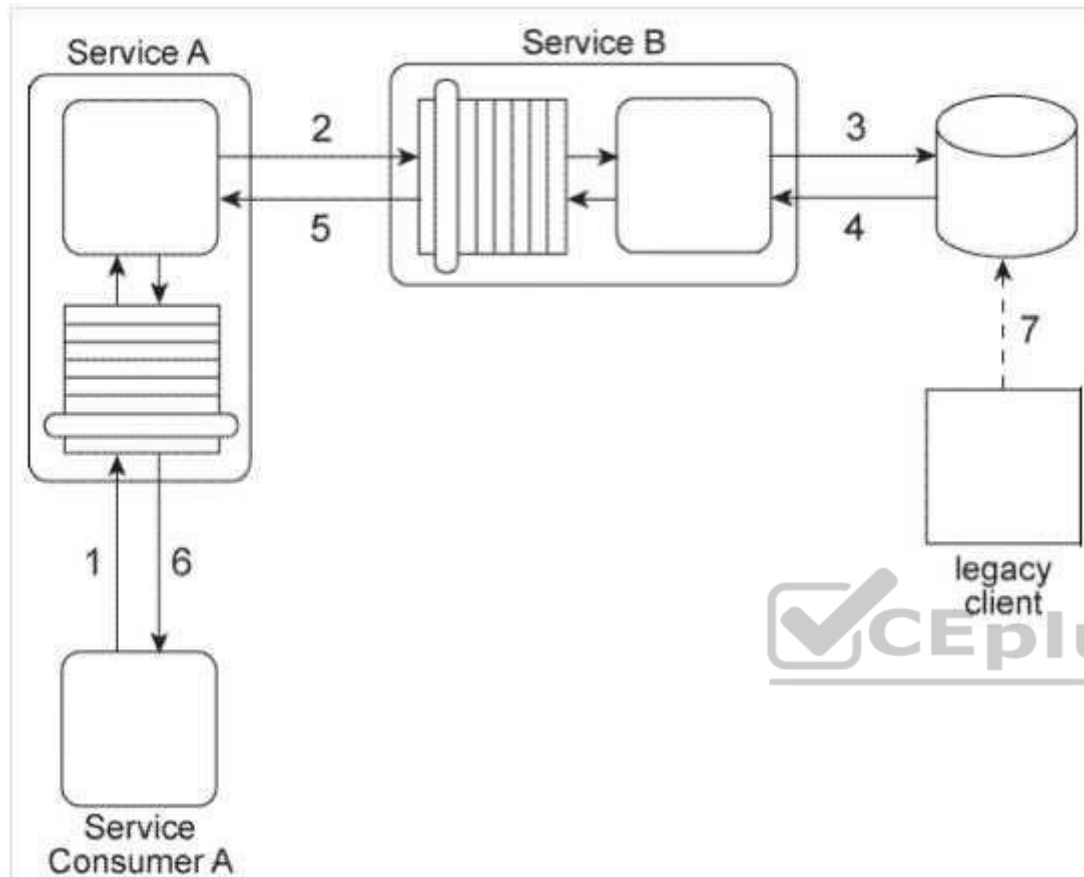
What steps can be taken to solve these problems and to prevent them from happening again in the future?

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 façade 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 Façade 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
**Section: (none)**
Explanation

Explanation/Reference:

QUESTION 13
The architecture for Service A displayed in the Figure shows how the core logic of Service A has expanded over time to connect to a database and a proprietary legacy system (1) and to support two separate service contracts (2) that are accessed by different service consumers.

The service contracts are fully decoupled from the service logic. The service logic is therefore coupled to the service contracts and to the underlying implementation resources (the database and the legacy system).

Service A currently has three service consumers. Service Consumer A and Service Consumer B access Service A's two service contracts (3, 4). Service Consumer C bypasses the service contracts and accesses the service logic directly (5).
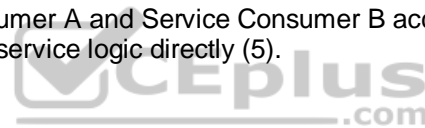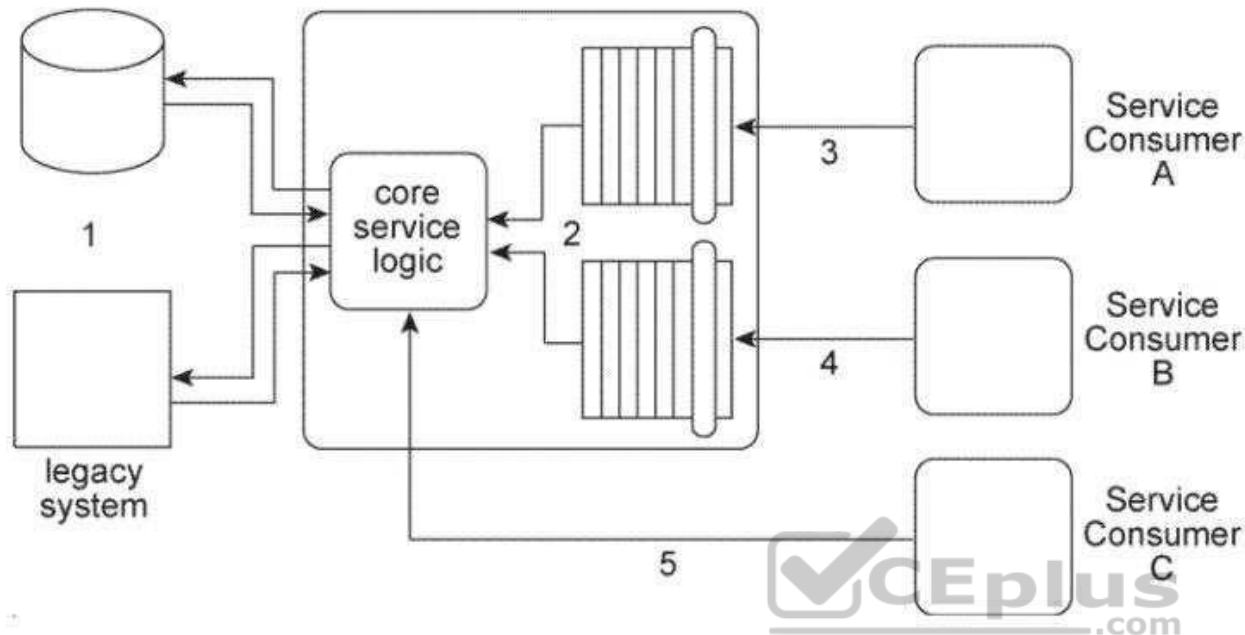
You are told that the database and legacy system that are currently being used by Service A are being replaced with different products. The two service contracts are completely decoupled from the core service logic, but there is still a concern that the introduction of the new products will cause the core service logic to behave differently than before. What steps can be taken to change the Service A architecture in preparation for the introduction of the new products so that the impact on Service Consumers A, B, and C is minimized?

A. The Service Abstraction principle can be applied to hide the implementation details from the core service logic of Service A, thereby shielding this logic from changes to the implementation. In support of this, the Service Facade pattern can be applied to position Facade components between the core service logic and Service Consumers A and B. These Facade components will be designed to regulate the behavior of Service A . The Contract Centralization pattern can be applied to force Service Consumer C to access Service A via one of its existing service contracts.

B. A third service contract can be added together with the application of the Contract Centralization pattern. This will force Service Consumer C to access Service A via the new service contract. The Service Facade pattern can be applied to position a Facade component between the new service contract and Service Consumer C in order to regulate the behavior of Service A . The Service Abstraction principle can be applied to hide the implementation details of Service A so that no future service consumers are designed to access any of Service A's underlying resources directly.

C. The Service Facade pattern can be applied to position Facade components between the core service logic and the two service contracts. These Facade components will be designed to regulate the behavior of Service A . The Contract Centralization pattern can also be applied to force Service Consumer C to access Service A via one of its existing service contracts. D. None of the above.
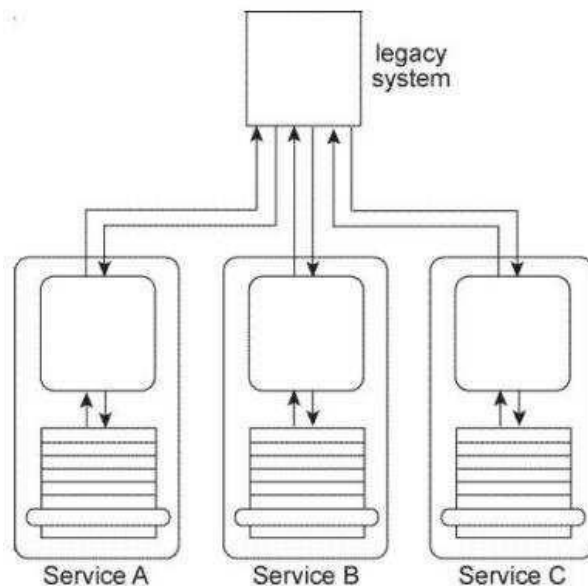
Correct Answer: C

Section: (none)
Explanation

Explanation/Reference:

QUESTION 14
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.

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 A, B, and C and any future services that are designed to rely upon it?

A. The Legacy Wrapper pattern can be applied together with the Standardized Service Contract principle to position a standardized service contract between the legacy system and any services that require access to it. This effectively establishes a new utility service dedicated to the encapsulation of the legacy system.

When the legacy system is replaced, the utility service can keep its standardized service contract. To build the utility service, the Data Format Transformation pattern is applied to convert between the proprietary legacy system file format and the XML format used in the standardized service contract.

B. The Legacy Wrapper pattern can be applied together with the Official Endpoint pattern so that the Service A service contract is positioned as the sole access point for the legacy system. The Data Format Transformation pattern is applied to enable the conversion between the proprietary legacy system file format and the XML format used in the Service A service contract. Finally, the Contract Centralization pattern is applied so that Service A is forced to only access the legacy system via its published standardized service contract.

C. The Legacy Wrapper pattern can be applied together with the Data Format Transformation pattern and the Standardized Service Contract principle in order to establish an intermediate layer of standardized transformation logic that is positioned between the legacy system and Services A, B, and C. This way, if the legacy system is replaced, the services will not be affected because of the abstraction established by the standardized transformation layer. D. None of the above.

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:

QUESTION 15
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.
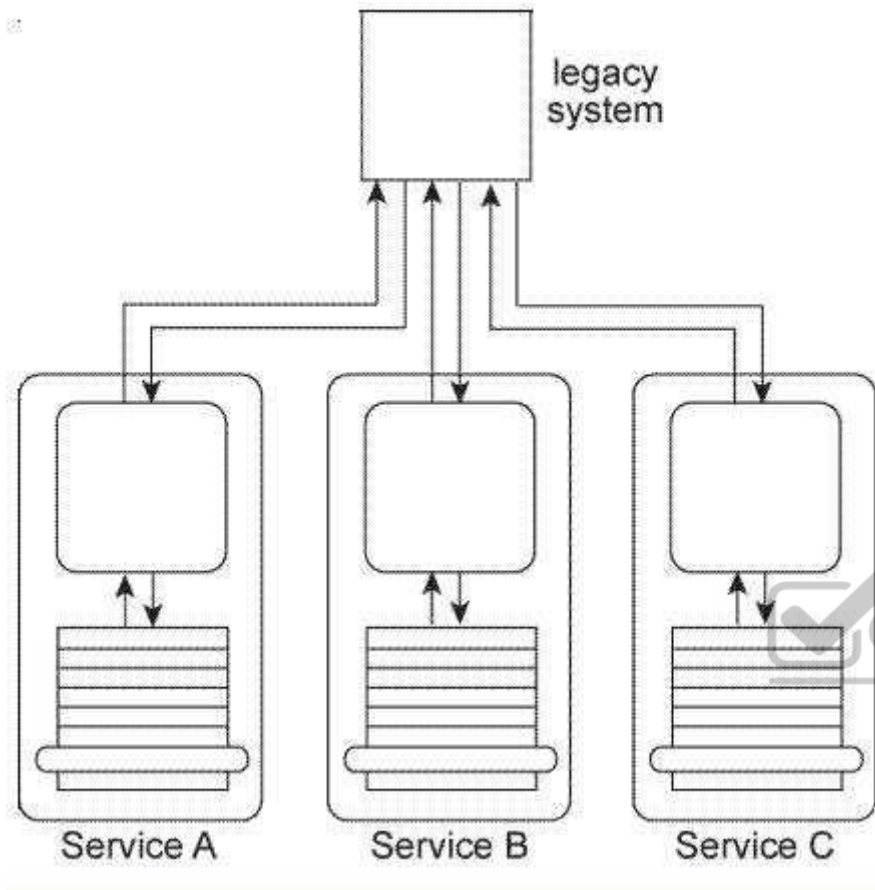
Service A is an agnostic utility service that is used by other services to gain access to the legacy system. Services B and C were not designed to access the legacy system via Service A because the Service A service contract was derived from the legacy system API and is therefore not standardized and 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
Section: (none)
Explanation

Explanation/Reference:

QUESTION 16
Service Consumer A sends a message to Service A. Before the message arrives with Service A, it is intercepted by Service Agent A (1). which checks the message for compliance to Policy A that is required by Service A. If the message fails compliance, Service Agent A will not allow it to proceed and will instead write the message contents to a log. If the message does comply to the policy, it continues to be transmitted toward Service A, but before it arrives it is intercepted by Service Agent B (2), which validates the security credentials in the message header. If the security credential validation fails, the message is rejected and a runtime exception is raised. If the security credentials are validated, the message is sent to Service A.

Upon receiving the message, Service A retrieves a data value from a database and populates the message header with this data value (3) prior to forwarding the message to Service B. Before the message arrives at Service B. it is intercepted by Service Agent C (4) which checks the message for compliance with two policies: Policy B and Policy C. Policy B is identical to Policy A that was checked by Service Agent A. To check for compliance to Policy C. Service Agent C uses the data value added by Service A. If the message complies with both of the policies, it is forwarded to Service B (5), which stores the message contents in its own database.
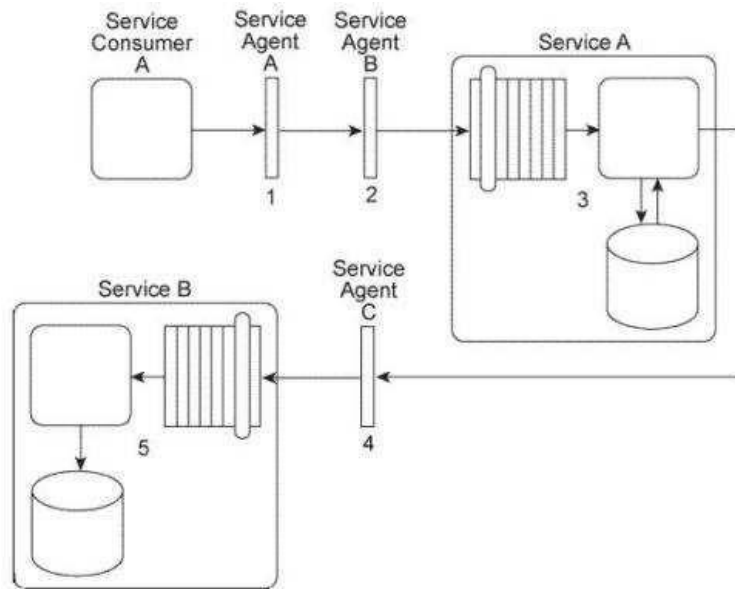
You are told that Policy B and Policy C have changed. Also, in order to carry out the compliance check of Policy C, Service Agent C will now require a new data value from the Service B database. How can this service composition architecture be changed to fulfill these new requirements?

A. The Policy Centralization pattern can be applied so that only one service agent is used to enforce Policy A and Policy B. Service A is redesigned to first query Service B for the value required by Service Agent C to check the compliance of the updated Policy C. If the compliance check is successful, the message is sent to Service B .

B. The Policy Centralization pattern can be applied so that only one service agent is used to enforce Policy A and Policy B. Service Consumer A is redesigned to first query Service B for the value required by Service Agent C. This way, Service Consumer A can include this value in the message header prior to sending the message to Service A .

C. The Policy Centralization pattern can be applied so that only one service agent is used to enforce Policy A and Policy B. The policy enforcement logic for Policy C is removed from Service Agent C and instead embedded within the logic of Service B . This way, Service B can itself retrieve the value required to check compliance with Policy C. If the message received is not in compliance, Service B will reject it. D. None of the above.

Correct Answer: D
**Section: (none)**
Explanation

Explanation/Reference:

QUESTION 17
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.
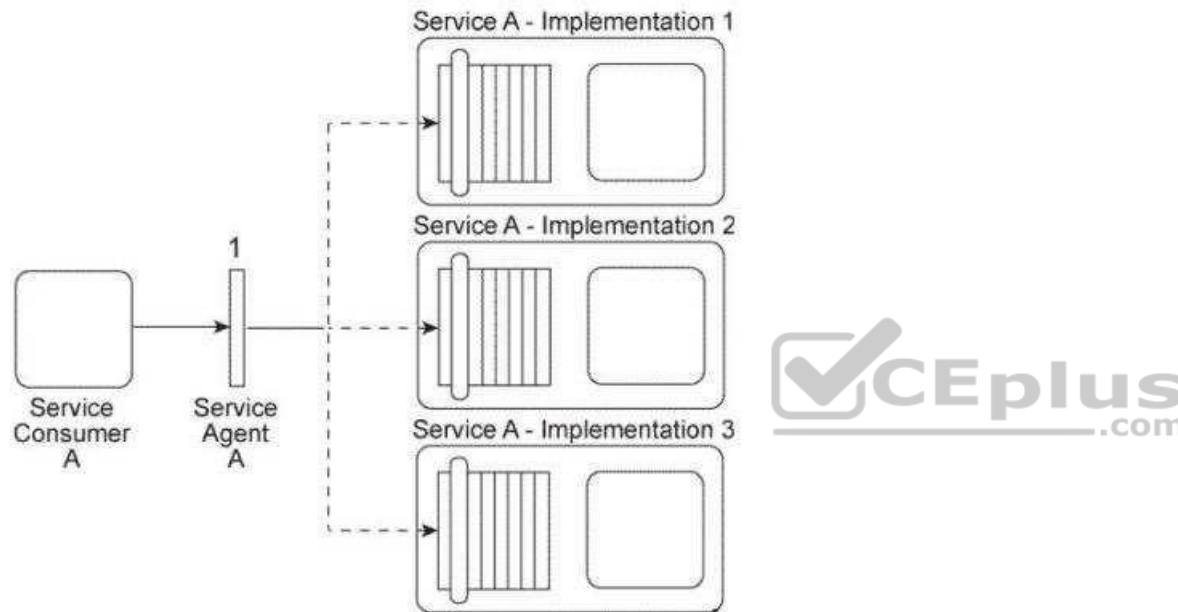
You are told that despite the fact that duplicate implementations of Service A exist, performance is still poor at times. Also, you are informed that a new service capability will soon need to be added to Service A that will introduce functionality that will require access to a shared database that is 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
**Section: (none)**
Explanation

Explanation/Reference:

QUESTION 18
Service A sends a message to Service B (1). After Service B writes the message contents to Database A (2) it issues a response message back to Service A (3). Service A then sends a message to Service C (4). Upon receiving this message, Service C sends a message to Service D (5), which then writes the message contents to Database B (6) and issues a response message back to Service C (7).

Service A and Service D are in Service Inventory A. Service B and Service C are in Service Inventory B.
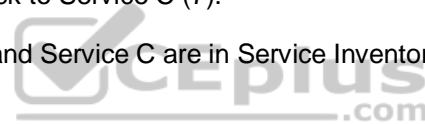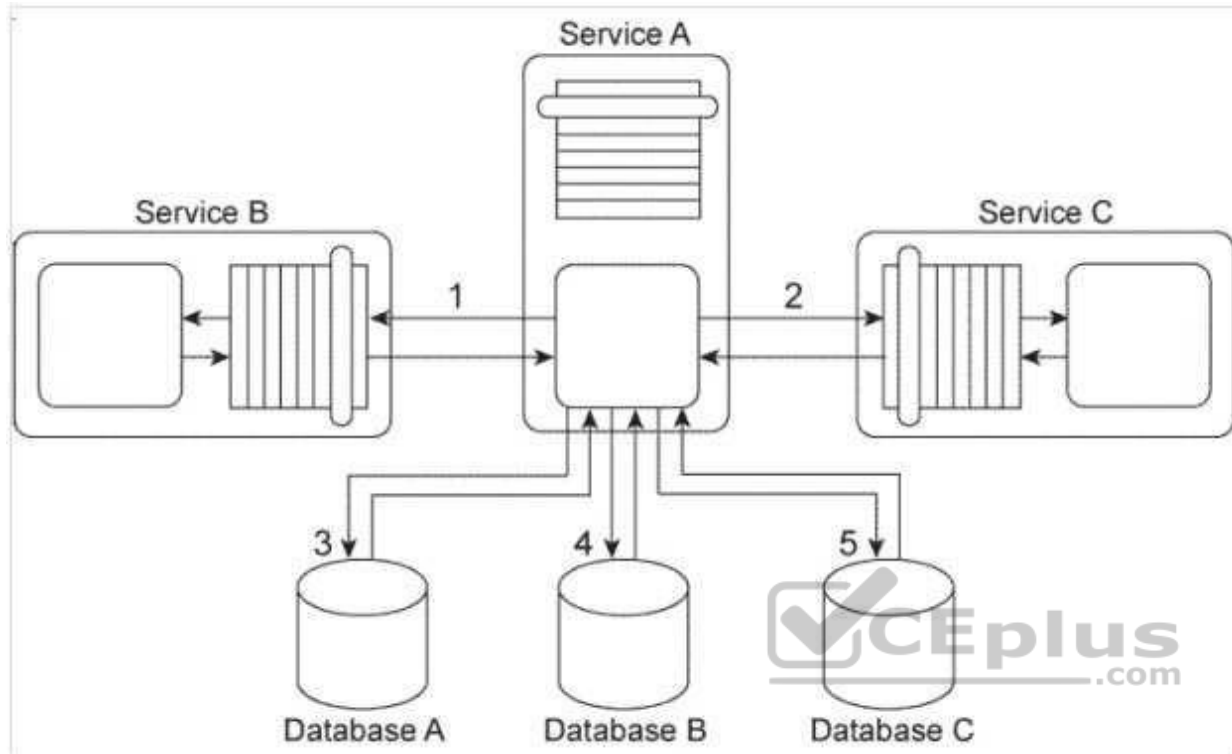
You are told that in this service composition architecture, all four services are exchanging invoice-related data in an XML format. However, the services in Service Inventory A are standardized to use a different XML schema for invoice data than the services in Service Inventory B. Also, Database A can only accept data in the Comma Separated Value (CSV) format and therefore cannot accept XML formatted data. Database B only accepts XML formatted data. However, it is a legacy database that uses a proprietary XML schema to represent invoice data that is different from the XML schema used by services in Service Inventory A or Service Inventory B. 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 C and Service D, and between the Service D logic and Database B. The Data Format Transformation pattern can be applied so that data format transformation logic is positioned between Service A and Service C, and between the Service B logic and Database A.

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

D. None of the above.

Correct Answer: C
Section: (none)
**Explanation**

Explanation/Reference:

QUESTION 19
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?
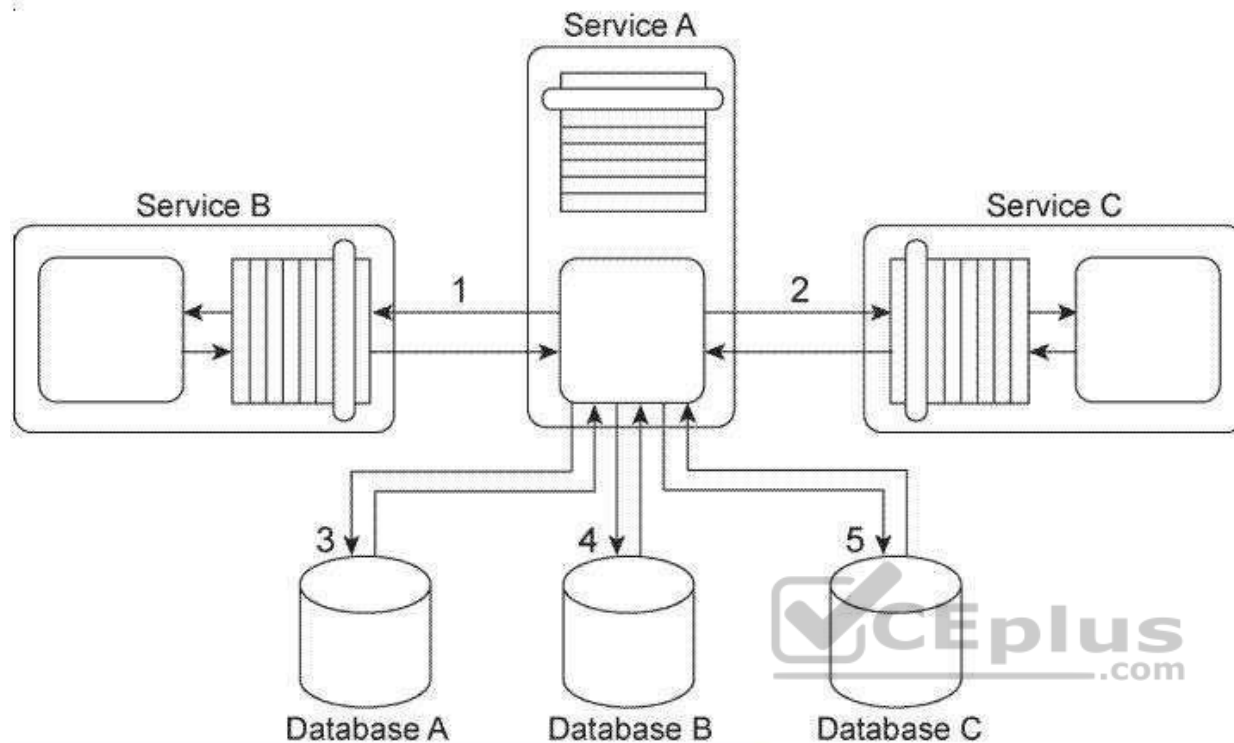
A. The Data Model Transformation pattern can be applied so that data model transformation logic is positioned between Service A and Service B, between Service A and Service C, and between Service C and Service D . The Data Format Transformation pattern can be applied so that data format transformation 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
Section: (none)

Explanation

Explanation/Reference:

QUESTION 20
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 te>X 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.
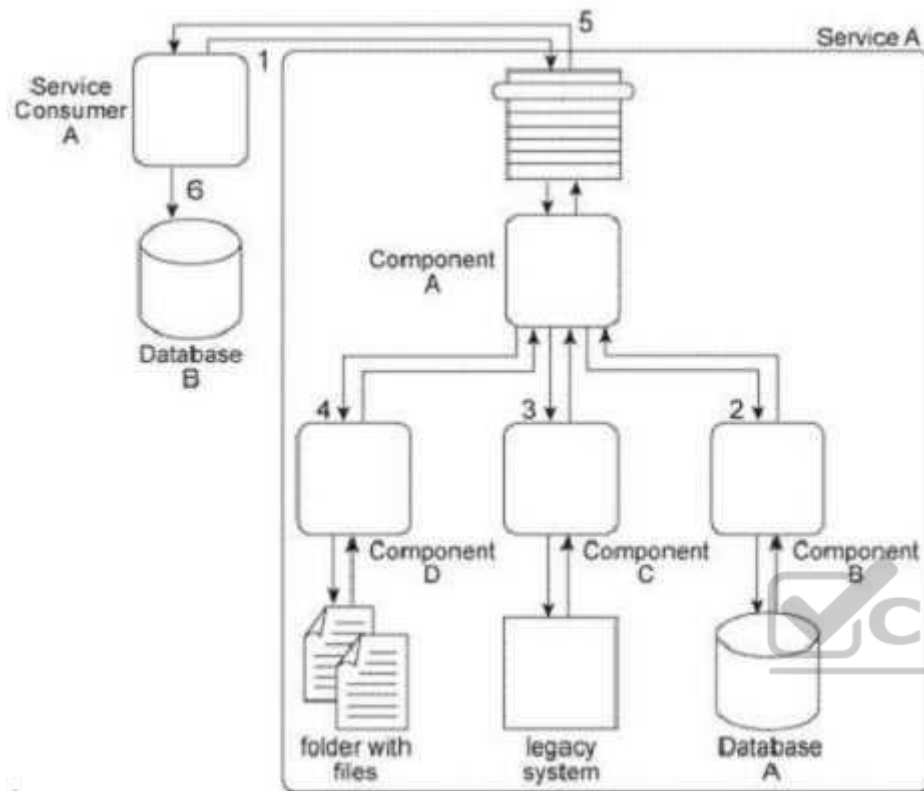
Service A is an entity service with a service architecture that has grown over the past few years. As a result of a service inventory-wide redesign project, you are asked to revisit the Service A service architecture in order to separate the logic provided by Components B, C, and D into three different utility services without disrupting the behavior of Service A as it relates to Service Consumer A . What steps can be taken to fulfill these requirements?

A. The Legacy Wrapper pattern can be applied so that Component B is separated into a separate wrapper utility service that wraps the shared database. The Asynchronous Queuing pattern can be applied so that a messaging queue is positioned between Component A and Component C, thereby enabling communication during times when the legacy system may be unavailable or heavily accessed by other parts of the IT enterprise. The Service Facade pattern can be applied so that a Facade component is added between Component A and Component D so that any change in behavior can be compensated. The Service Autonomy principle can be further applied to Service A to help make up for any performance loss that may result from splitting the component into a separate wrapper utility service.

B. The Legacy Wrapper pattern can be applied so that Component B is separated into a separate utility service that wraps the shared database. The Legacy Wrapper pattern can be applied again so that Component C is separated into a separate utility service that acts as a wrapper for the legacy system API. The Legacy Wrapper pattern can be applied once more to Component D so that it is separated into another utility service that provides standardized access to the file folder. The Service Facade pattern can be applied so that three Facade components are added: one between Component A and each of the new wrapper utility services. This way, the Facade components can compensate for any change in behavior that may occur as a result of the separation. The Service Composability principle can be further applied to Service A and the three new wrapper utility services so that all four services are optimized for participation in the new service composition. This will help make up for any performance loss that may result from splitting the three components into separate services.

C. The Legacy Wrapper pattern can be applied so that Component B is separated into a separate utility service that wraps the shared database. The Legacy Wrapper pattern can be applied again so that Component C is separated into a separate utility service that acts as a wrapper for the legacy system API. 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. The Service Composability principle can be further applied to Service A and the three new wrapper utility services so that all four services are optimized for participation in the new service composition. This will help make up for any performance loss that may result from splitting the three components into separate services. D. None of the above.

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:

QUESTION 21
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.
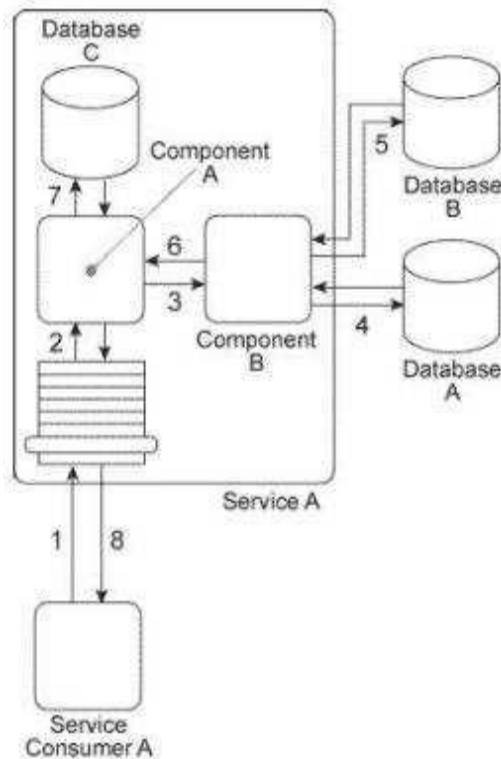
There are several problems with this architecture: First, the response time of Database A is often poor, resulting in Component B taking too much time to provide a response to Component A. This results in Component A consuming too many runtime resources while it holds the business document in memory 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
**Section: (none)**
Explanation

Explanation/Reference:

QUESTION 22
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.
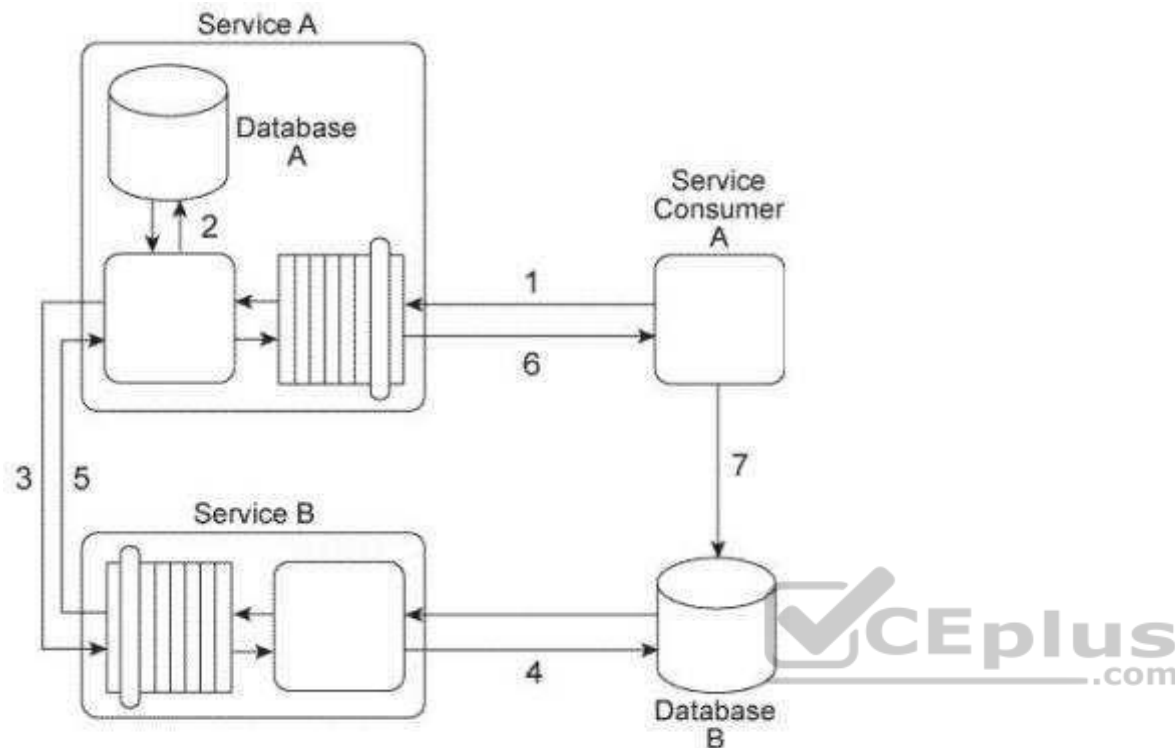
There are two problems with this service composition architecture that you are asked to address: First, both Service Consumer A and Service B need to transform the business document data from an XML format to a proprietary Comma Separated Value (CSV) in order to write the data to Database B. This has led to redundant data format transformation logic that has been difficult to keep in synch when Database B changes. Secondly, Service A is an entity service that is being reused by several other service compositions. It has lately developed reliability problems that have caused the service to become unavailable for extended periods. What steps can be taken to solve these problems?

A. The Legacy Wrapper pattern can be applied so that data access to Database B is separated into a new wrapper utility service. This way, the Data Format Transformation pattern only needs to be applied within the logic of this new service which will expose a standardized contract that both Service Consumer A and
Service B can access. The Asynchronous Queuing pattern can be applied so that messaging queues are established between Service Consumer A and Service A and between Service A and Service B . The Service Autonomy principle can be further applied to Service A in order to establish a more isolated and reliable surrounding infrastructure.

B. The Legacy Wrapper pattern can be applied so that data access to Database B is separated into a new wrapper utility service. This way, the Data Format Transformation pattern only needs to be applied within the logic of this new service which will expose a standardized contract that both Service Consumer A and
   Service B can access. The Reliable Messaging pattern can be applied so that acknowledgements are passed between Service Consumer A and Service A and between Service A and Service B . The Service Composability principle can be further applied to Service A in order to optimize its service architecture for improved participation in multiple service compositions.

C. The service composition can be redesigned with the application of the Contract Centralization pattern so that instead of writing the business document to Database B, Service Consumer A sends the business document to Service B instead. This way, Service B would provide the only location where data format transformation logic for Database B needs to be carried out, which further supports the application of the Service Reusability principle. The Reliable Messaging pattern can be applied so that acknowledgements are passed between Service Consumer A and Service A and between Service A and Service B . The Service Composability principle can be further applied to Service A in order to optimize its service architecture for improved participation in multiple service compositions.

D. None of the above.

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference: