Question 9 of 9

5 Points

[10 minutes]

Describe how we can relate a **stream based architecture** to one or more principles of SCI. Your answer should be about 2 to 3 paragraphs. The number of points you get for this ques
the relationship between a **stream based architecture** and one or more principles of SCI.

Maximum number of characters (including HTML tags added by text editor): 32,000

Show Rich-Text Editor (and character count)

A streaming data architecture is a framework of software components built to ingest and process large volumes of streaming data from multiple sources and this is the Unified Field is the abstract field that unites all diversity in creation as we have the stream it self Unified all services to send and receive from it Pure Consciousness one gets access to all intelligence of creation As same as in stream base we have one stream and all services Producer–consumer use it

Question 8 of 9

10 Points

[15 minutes]

a. Explain clearly why in a microservice architecture typically does not make use of an ESB.

b. Explain clearly why every microservice uses its own database instance? Why don't they share the same database?

Maximum number of characters (including HTML tags added by text editor): 32,000

Show Rich-Text Editor (and character count)

A-

Microservices does not use ESB becouse ESB connect services using orchestration "central brain " but Microservices use choreography "No central brain" becouse it have small services

B-

Microservices advocate design constraints where each service is developed, deployed and scaled independently. This philosophy is only possible if you have database per service

Question 7 of 9

10 Points

[10 minutes]

In this course we learned that the registry uses a heartbeat to check if a service is still up and running. Suppose serviceA instance1 is down and service ServiceB will call the registry, and gets the address of serviceA instance2 because the registry knows that serviceA instance1 is down.

If we also add load balancing to these calls between serviceA and serviceB we basically have failover. If a service fails then the registry knows about registry and load balancing give us already failover, why do we need hystrix? In other words, what does hystrix provide that we don't get already w

Maximum number of characters (including HTML tags added by text editor): 32,000

Show Rich-Text Editor (and character count)

Hystrix (circuit breaker) will use the Resilience patterns to introduce fault tolerance and latency tolerance by isolating failure and by preventing them from cascading into the other part of the system Circuit breaker will periodically check if Service A instance 1 is back online and restore it automatically once the service came back

Load balancing lets you evenly distribute network traffic to prevent failure caused by overloading a particular resource.
If one instance stopped, automatically the other instance will be used, this strategy improves the performance and availability of applications

Contents

of 1 .

Question 5 of 9

5 minutes]

15 Points

A -Option 1 : Advantage: Full control of the synchronous flow, Easy to monitor the process Disadvantages: Coupling, Orchestrator is single point of failure , No parallel processing
Option 2 :Advantage: Less Coupling , Easy to add/remove services without impact on other services , Fast: parallel processing , No single point of failure Disadvantages: Harder to monitor the process
B-I'd say we need to focus on business logic. Where a single point of logic does the job and the app does not need to be that scalable, while I am looking for simplicity, we do option 1, Where a problem cannot be covered by a centralized logic, or we're looking for a scalable application that will serve a huge amount of requests we are forced to go with option 2.

pose you have an existing application that contains about 10 components. You need to implement the logic of one business process that executes logic in 6 different components.
example business process would be that a customer purchases a flight at an airline. In the customer component we have to add the customer. In the flight component we have to reserve the se
e meals component we have to add the meal for this customer. In the rewards component we have to add the rewards points for this flight.

ave 2 options for implementing the logic of this business process:

n 1: You create a separate component that contains the overall business process itself, and this separate component will call the other 6 components to perform the necessary logic.

2: You divide the overall business process in smaller parts, and implement these smaller parts in the existing 6 different components.

are the advantages and disadvantages of option 1 and option 2?

oss asks you for your advice. Explain clearly when you would choose option 1 and when you would choose option 2.

Question 6 of 9

10 Points

[10 minutes]

Suppose we have a system that contains many microservices. Now we need to write a client web application in Angular that shows data that com
Give **3 valid reasons** why it is not a good idea to have this client web application talk directly to the different microservices.
(Be careful, only give valid reasons. If you give one or more reasons that are not valid you will lose some points)

Maximum number of characters (including HTML tags added by text editor): 32,000

Show Rich-Text Editor (and character count)

Valid reason why it is not good idea to have web client talk directly to microservices

1- Not all services support HTTP. Maybe they use only messaging, and the client does not support messaging
2- Tight coupling: Client needs to know all details of how to call a service
3- Difficult to implement crosscutting concerns: security, logging, transformation
4- Performance issue when you have chatty communication

Question 3 of 9

10 Points

[15 minutes]

Suppose you are responsible for designing and building Microservice A and Microservice B.
In System A you have to call system B in an **asynchronous** way.

a. We learned **2 completely different techniques/protocols** to implement an **asynchronous** call between Microservice A and Microservice B. Give the **name** of both **techniques/protocols**.

b. Clearly explain the architectural relevant **advantages and disadvantages** of both techniques/protocols given in part a. Explain clearly when you would use which techniques/protocols.

aximum number of characters (including HTML tags added by text editor): 32,000

Show Rich-Text Editor (and character count)

1- Reactive System
Advantage:
performance:no need to wait results to be available
scaling: less thread needing

Disadvantage:
The whole the whole calling stack need to be reactive, Hard to debug

2- event driven architecture (messageing system)
Advantage:
Fast, Flexible, Less dependencies, Loosely coupled Producers and Consumers , Using buffer if the service slow or down

Question 4 of 9 — 20 Points

[20 minutes]

We have an existing hotel booking system that offers generic hotel booking functionality like :

- Search hotels on keyword(s) like city, price, etc.
- View hotel details
- Book a hotel room
- Manage hotels (add new hotel, update hotel details, remove hotel, etc.)
- Etc.

a. You are the responsible architect for this system, and one of your colleagues advices to modify the existing system and apply the **CQRS** pattern for this system. Give **2 valid r**
CQRS pattern for this system. (**Be careful, only give valid reasons. If you give one or more reasons that are not valid you will lose some points**)

b. In the existing hotel booking system we have many different domain classes.
If we apply the CQRS pattern for this system, explain clearly what would change in the existing domain classes. In other words, how will the new domain classes differ from

Maximum number of characters (including HTML tags added by text editor): 32,000

Show Rich-Text Editor (and character count)

Read model or DTO need not have all the fields as a command model, and a read model can have required fields by the client view which can save the capacity of the read store

A - Separation of commands and queries help us to make simpler domain models by Separating the querying from command through providing two models instead of one.
One model is built to handle and process commands
One model is built for queries needs
In the above system there is 2 valid reason why we can apply CQRS

Question 2 of 9

5 minutes]

10 Points

icroservice architecture has many advantages, but also disadvantages. Suppose you are the architect of a large application build with many microservices

ne problem of a microservice architecture is to make the whole system secure.
t technique(s) would you use to solve this problem?

e problem of a microservice architecture is to make the whole system resilient to failure.
echnique(s) would you use to solve this problem?

problem of a microservice architecture is that it is difficult to get a good overview of the status of the business processes that are executed in the different microservices.
chnique(s) would you use to solve this problem?

roblem of a microservice architecture is that it is difficult to make a collection of actions that execute in different microservices in a transactional way. So I want to update data in different mi
on.
nique(s) would you use to solve this problem?

umber of characters (including HTML tags added by text editor): 32,000

Show Rich-Text Editor (and character count)

a- security use OAuth2, JWT
b- Resilience use Hystrix
c- Monitoring use zipkin, sleuth ,ELK
d-Transaction use SAGA pattern, Compensating transaction, Eventual consistency

Question 1 of 9

10 Points

[10 minutes]

Circle all statements that are true

- [ ] A. Suppose we need to apply load balancing for the API gateway. Client side load balancing is then the best option.
- [ ] B. Suppose we need to apply load balancing for the API gateway. Server side load balancing is then the best option.
- [ ] C. Suppose we have a microservice architecture and we need to apply load balancing for one of the microserices. Client side load balancing is then the best option.
- [ ] D. Suppose we have a microservice architecture and we need to apply load balancing for one of the microserices. Server side load balancing is then the best option.
- [ ] E. One reason why kafka uses event sourcing is that it allows producers to load balance the messages send to kafka
- [ ] F. A microservice architecture is always the best architectural style to use
- [ ] G. With JWT tokes, the user role is included in the token
- [ ] H. In a stream based architecture you always should use CQRS.
- [ ] I. We can implement the blackboard architectural style using kafka
- [ ] J. With the saga pattern we can implement strict consistency between microservices

B
C
G
I
J

Save