Repetition Questions (dt. Wiederholungsfragen) Lesson 12

Topics and Concepts (Link to Lecture)

The lesson of the lecture today covered the following concepts:

1. State(lessness) and state management patterns
2. Event sourcing and CQRS
3. Coupling criteria and service granularity
4. Architectural decisions

In the corresponding exercise[1], we applied these concepts in a sample scenario.

Questions

Topic/Concept: State(lessness)

1. What is the difference between session and resource state (both also called application state by different authors)?
2. Give one example for each type of state (session, resource) in an online shopping scenario.
3. List at least one advantage and one disadvantage per state management pattern (i.e., Client Session State, Server Session State, Database Session State).
4. What is the relationship between state management and event sourcing?

Topic/Concept: Event Sourcing and CQRS

1. Why and when could the CQRS pattern be applied? When should it be avoided?
2. Decide whether the following names make good event names: `MoneyTransfer`, `MoneyTransferCompleted`, `TransferMoney`, `MoneyTransferringInProcess`
3. What are the usage scenarios and benefits of event sourcing?
4. What is the relation between event sourcing, CQRS, and DDD?

Topic/Concept: Coupling Criteria and Service Granularity

1. Is loose coupling a property of a single service or a relation between two services calling each other? How about granularity?
2. List at least three coupling criteria supported by Service Cutter method and tool.
3. Name the two types of granularity and list the four interface representation (granularity) patterns introduced in the lecture.
4. Which general quality attribute ("ilities") drive the service cutting and granularity pattern selection decisons?

[1] ../3-exercises-solutions/ZIO-AppArch-ExerciseWeek12.pdf

Topic/Concept: Architectural Decisions (ADs)

1. Why is it important to capture ADs?
2. Is there a standard that governs architectural descriptions? If yes, which one?
3. Name at least two AD capturing templates from industry that were intrduced in lecture and exercise.
4. How/where do ADs work relate to quality attributes? In which part of the AD capturing templates can/should QAs be referenced?

Answers

Topic/Concept: State(lessness)

1. *Session state holds temporary data required to control page/window flow (presentation layer responsibility); resource state is permanent and owned and managed by the Business Logic Layer (BLL). Once a use case or user story completes (or a sequence of related ones), session state is handed over from presentation layer to BLL and becomes part of resource state.*
2. *The shopping basket is a canonical example for session state (unless its content is kept permanently on server side); orders, customer profiles and product catalog are all part of resource state.*
3. *Client Session State (CSS) scales well (no work for server) but has some development effort as well as performance and security risks attached to it. Server Session State (SSS) is supported out of the box in many application servers and other Web frameworks, but does not scale well (if sessions have to be made sticky/persistent because server is clustered). Database Session State (DSS) scales well and is as secure as SSS, but requires DevOps work for server team and also has a performance penalty (extra database traffic).*
4. *Event sourcing removes the need for (resource) state management as current state is recalculated from event stream/log.*

Topic/Concept: CQRS and Event Sourcing

1. *Many reads, few writes; high performance applications; occasional connectivity; should be avoided if strict consistency is needed and many business rules need to be validated (in domain model)*
2. `MoneyTransferCompleted` *is a good name (indicates that event has occurred in the past);* `MoneyTransfer`, `TransferMoney`, `MoneyTransferringInProcess` *are not suited because they do not unveil the event nature.*
3. *Bounded Context and Aggregate-to-Aggregate integration in DDD; full audit log, temporal queries, rebuild/replay (as in version control systems); Event-Driven Architectures (EDAs)*
4. *See previous answer, lecture slide 11 and this article: https://www.infoq.com/articles/microservices-aggregates-events-cqrs-part-1-richardson.*

Topic/Concept: Coupling Criteria and Service Granularity

1. *Coupling occurs between service consumers and providers; granularity is a property of an individual service.*
2. *Semantic Proximity, Identity and Lifecycle Commonality, Consistency Criticality*
3. *Business Granularity, Technical Granularity; Atomic Parameter, Atomic Parameter List, Parameter Tree, Parameter Forrest*
4. *e.g. "RPS" subset of FURPS (the F is functional and the U category/NFR targets end user interface design for the most part, and application-to-application connectivity/integration to a lesser extent). API calls should not crash and return correct results, they should be responsive; APIs need to evolve and therefore must be supportable (other answers possible)*

Topic/Concept: Architectural Decisions

1. *See lecture slide 24: answer "why" questions, have a single point of reference, etc.*
2. *Yes, IEC/IEEE/ISO 42010:2011*
3. *IBM UMF, Y statements; M. Nygard's ADRs and many more, e.g. Tyree/Akerman (based on IBM table format too, published here[2])*
4. *QAs can justify ADs (see appendix to lecture slides Version 1.0.1 for examples of good and bad decision justifications)*

[2] http://ieeexplore.ieee.org/document/1407822