

Software architecture topics for the midterm

Lesson 1

Software qualities

Architecture (design) principles

Communicating architecture

- Context diagram
- Container diagram
- Component diagram , sequence diagram of components
- Class diagram , sequence diagram of classes

Identify risk: risk storming

Clustering

Failover

Lesson 2

Architecture styles

- Layering
- Client-server
- Pipe and filter
- Master-slave
- Microkernel

Service class

DOA/repository class

Proxy/Gateway class

Relational database versus NOSQL database

Scaling databases

Brewers cap theorem

Strict consistency – eventual consistency

Lesson 3

Dependency injection

REST

Mongo

You do not need to write Spring dependency injection, REST or Mongo data access code.

You need to understand the concepts.

Lesson 4

Entity

Value object

Domain service

Events

Aggregate root

You should be able to make designs with these concepts

Lesson 5

Ubiquitous language

Rich domain model

Bounded context

Components

Interface design

DTO's

Adapter

Spring Events string publisher and subscribe

You need to know how to write code in Spring that uses events.

You should be able to make designs with these concepts

Lesson 6

JMS

Hub and spoke

ESB

Integration patterns what are the patterns and concepts, for example what is router?

You do not need to write JMS code. You need to understand the concepts.

You should understand and be able to apply the different integration patterns we studied

use router, adapter,

Lesson 7

ESB versus Integration Framework

Integration patterns in spring integration

You need to understand the concepts of how spring integration works.

You need to be able to understand the XML configuration for the following patterns

- Service activator
- Gateway
- Channels
- Point-to-point vs. Publish-subscribe
- Synchronous vs asynchronous
- Custom router
- Filter
- Transformer

You do not need to write XML configuration.

You do need to know how to write Java implementation of the following patterns:

- Custom router
- Filter
- Transformer