Java-Success.com

Industrial strength Java/JEE Career Companion to open more doors

Home › Interview › Java Architecture Interview Q&A › 03: ♦ What should be a typical Java EE architecture?

03: ♦ What should be a typical Java EE architecture?

Posted on May 3, 2015 by Arulkumaran Kumaraswamipillai

This question is a very popular white board session question for both Java architects and experienced JEE Developers. You need to draw on your experience to tackle this question as there are no right or wrong answers. These high level diagrams and summary will help you refresh your memory.

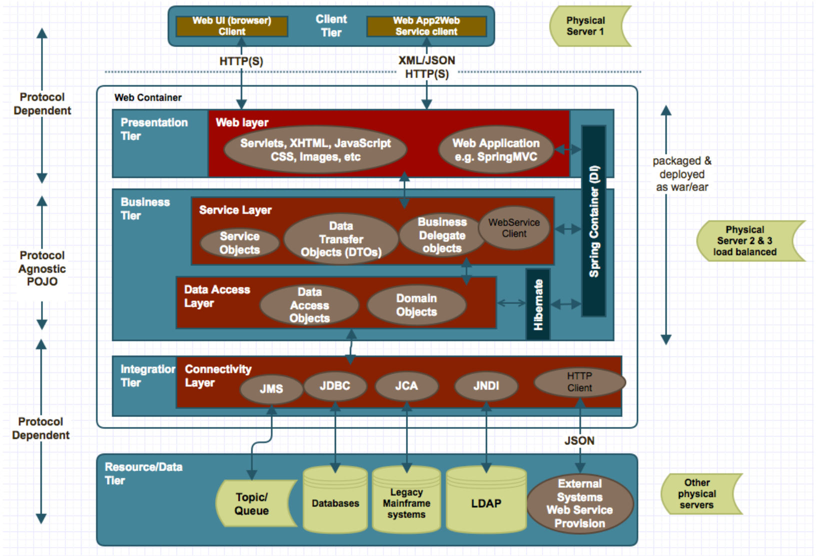
Q. What should be a typical Java EE architecture for let's say, a medium-size web-based application?
A. Before start drawing the components on the whit board, you need to show that you gather requirements.

#1. Ask some questions like How many transactions per minute or hour should the system handle? How many concurrent users should it handle?, etc.

600+ Full Stack Java/JEE Interview Q&As ♥Free ♦FAQs

open all | close all
- Ice Breaker Interview
- Core Java Interview (
- JEE Interview Q&A (3
- Pressed for time? Jav
- SQL, XML, UML, JSC
- Hadoop & BigData In
- Java Architecture Inte
  - ♥♦ 01: 30+ Writing
  - 001A: ♦ 7+ Java in
  - 001B: ♦ Java archi
  - 01: ♥♦ 40+ Java W
  - 02: ♥♦ 13 Tips to w
  - 03: ♦ What should
  - 04: ♦ How to go ab
  - 05: ETL architectur
  - 1. Asynchronous p
  - 2. Asynchronous p
- Scala Interview Q&As
- Spring, Hibernate, & I
- Testing & Profiling/Sa
- Other Interview Q&A

#2. Draw a big picture diagram. Mark the tiers, layers, key components, frameworks, etc. This is a 100 feet bird's eye view of the components.



JEE High Level Architecture

Be prepared for drill down questions like

1) Client pull Vs Server push. **Client pull** requires a special directive either in the HTML document header. This directive instructs the client to retrieve a specified document after a certain amount of time. In other words, the client opens a new connection to the server. **Server push** involves sending packets of data to the client periodically. The HTTP connection between the client and the server is kept open indefinitely. For example, you can use an asynchronous servlet.

2) When to use queue vs topic. **Queue** is for single receiver and a **topic** is for multiple subscribers.

3) When to use LDAP vs Database. LDAP stores data hierarchically and more suited for read intensive operations like looking up users, roles, etc for authorization, and database is more suited for CRUD operations.

⊞ ▶Free Java Interviev

16 Technical Key Areas

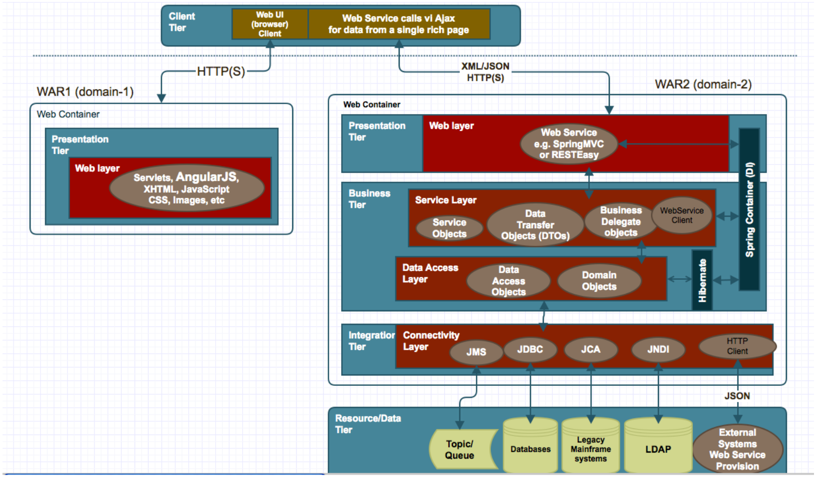
open all | close all
⊞ Best Practice (6)
⊞ Coding (26)
⊞ Concurrency (6)
⊞ Design Concepts (7)
⊞ Design Patterns (11)
⊞ Exception Handling (3
⊞ Java Debugging (21)
⊞ Judging Experience I
⊞ Low Latency (7)
⊞ Memory Managemen
⊞ Performance (13)
⊞ QoS (8)
⊞ Scalability (4)
⊞ SDLC (6)
⊞ Security (13)
⊞ Transaction Managen

80+ step by step Java Tutorials

open all | close all
⊞ Setting up Tutorial (6)
⊞ Tutorial - Diagnosis (2
⊞ Akka Tutorial (9)
⊞ Core Java Tutorials (2
⊞ Hadoop & Spark Tuto
⊞ JEE Tutorials (19)
⊞ Scala Tutorials (1)
⊞ Spring & HIbernate Tı
⊞ Tools Tutorials (19)
⊞ Other Tutorials (45)

4) What is Rich Internet Application (RIA)? Modern web applications are rich single page applications making use of JavaScript code and frameworks like angularjs. A single rich page makes ajax calls to back end systems to populate a particular section or tab of a rich page. This architecture is summarized below with an architecture diagram.

The modern web applications using AngulaJS or any other JavaScript based MVC frameworks may have **two war files**. One for the GUI and the second one for exposing the data via web services in JSON format.



Separate war files for UI and data

When using multiple domains as shown above, you need to use **CORS (Cross Origin Resource Sharing)** to overcome the JavaScript restriction on cross domains. CORS allows you to share GET, POST, PUT, and DELETE requests and CORS is supported by the modern browsers. The CORS make use of 2 requests.

Request 1: "**OPTIONS**" request as part of the handshake to determine if cross domain is allowed by the server.

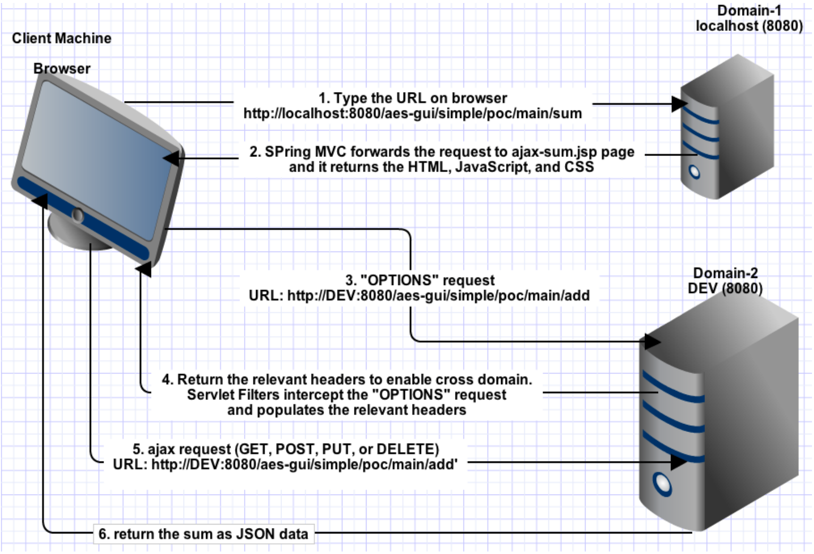
Request 2: GET, POST, PUT, or DELETE request that performs the actual operation on the server.

100+ Java pre-interview coding tests

open all | close all
⊞ Can you write code? (
⊞ ♦ Complete the given
⊞ Converting from A to I
⊞ Designing your classe
⊞ Java Data Structures
⊞ Passing the unit tests
⊞ What is wrong with th
⊞ Writing Code Home A
⊞ Written Test Core Jav
⊞ Written Test JEE (1)

How good are your .....?

open all | close all
⊞ Career Making Know-
⊞ Job Hunting & Resum


Client Machine
Browser
Domain-1
localhost (8080)
1. Type the URL on browser
http://localhost:8080/aes-gui/simple/poc/main/sum
2. SPring MVC forwards the request to ajax-sum.jsp page
and it returns the HTML, JavaScript, and CSS
Domain-2
DEV (8080)
3. "OPTIONS" request
URL: http://DEV:8080/aes-gui/simple/poc/main/add
4. Return the relevant headers to enable cross domain.
Servlet Filters intercept the "OPTIONS" request
and populates the relevant headers
5. ajax request (GET, POST, PUT, or DELETE)
URL: http://DEV:8080/aes-gui/simple/poc/main/add'
6. return the sum as JSON data

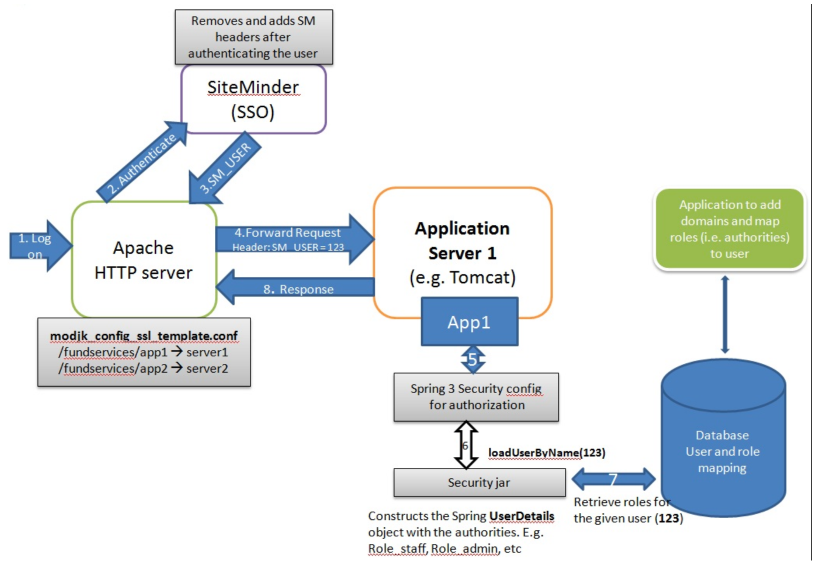
CORS

#3. Security considerations

Enterprise applications make use of SSO (Single-Sign-On) with enterprise level products like SiteMinder, Tivoli Access Manager, etc.

For example, SSO application like SiteMinder is configured to intercept the calls to authenticate the user. Once the user is authenticated, a HTTP header “SM_USER” is added with the authenticated user name. For example “123”. The user header is passed to Spring 3 security. The “Security.jar” is a custom component that knows how to retrieve user roles for a given user like 123 from a database or LDAP server. This custom component is responsible for creating a UserDetails Spring object that contains the roles as authorities. Once you have the authorities or roles for a given user, you can restrict your application URLs and functions to provide proper access control.

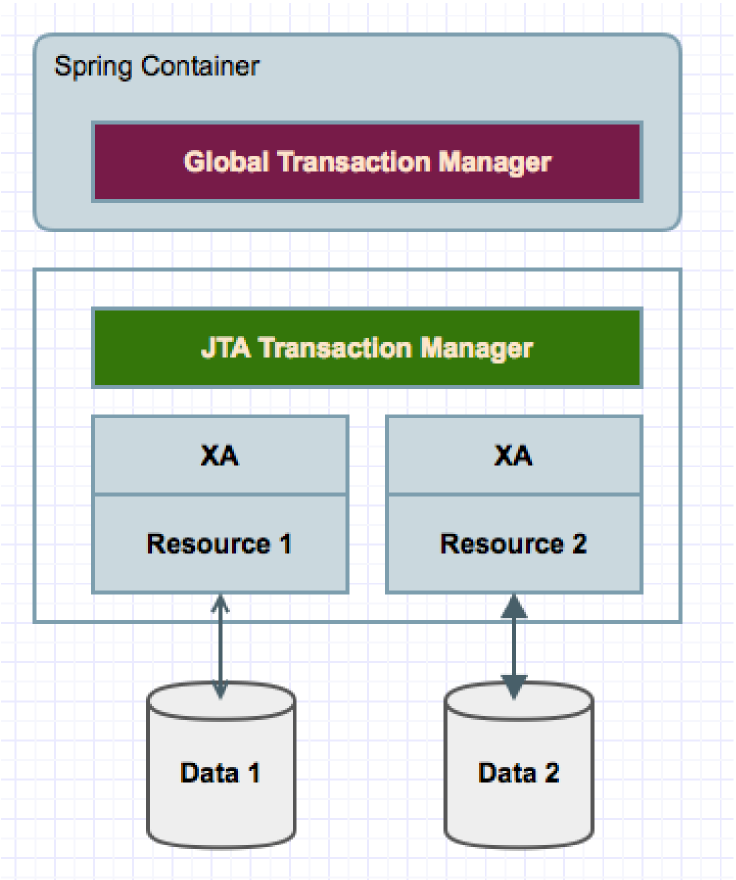
Learn more about SSO


Removes and adds SM headers after authenticating the user
SiteMinder (SSO)
2. Authenticate
3.SM_USER
1. Log on
Apache HTTP server
4.Forward Request
Header: SM_USER = 123
8. Response
Application Server 1 (e.g. Tomcat)
App1
Application to add domains and map roles (i.e. authorities) to user
modjk_config_ssl_template.conf
/fundservices/app1 → server1
/fundservices/app2 → server2
5
Spring 3 Security config for authorization
6
loadUserByName(123)
Security jar
7
Database User and role mapping
Retrieve roles for the given user (123)
Constructs the Spring UserDetails object with the authorities. E.g. Role_staff, Role_admin, etc

SSO with SiteMinder

#4. Transaction Management

Transaction management takes place at the service layer. You can use a Spring transaction manager with annotations such as **@Transactional** to demarcate transactional boundaries. The service class can call multiple data access object (DAOs) within a transactional context.



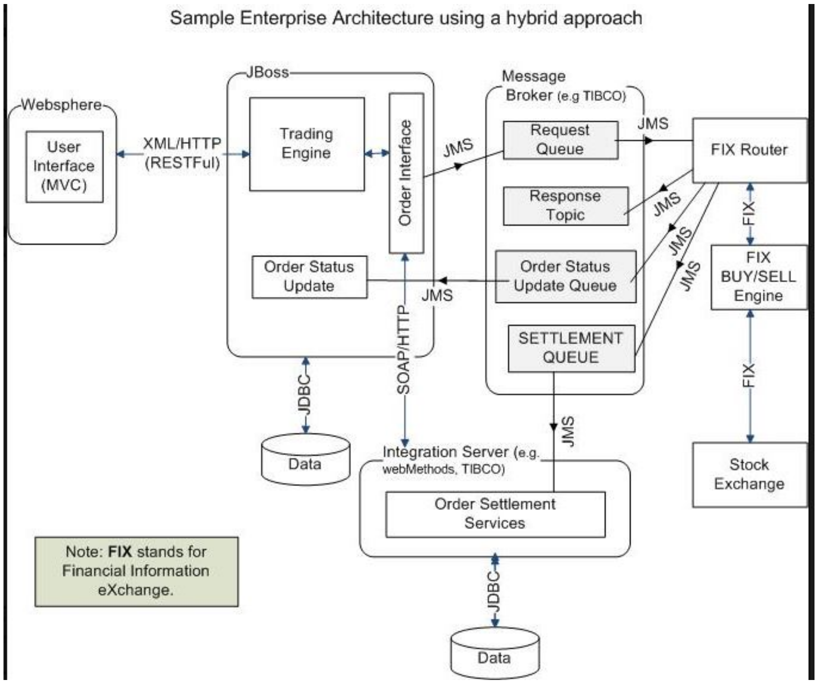
Transaction Manager

#5. Quality of Service (QoS) considerations

Quality of service (QoS) requirements are technical specifications that specify the system quality of features such as availability, scalability, serviceability, etc. This is covered in detail under QoS interview questions and answers.

Finally, the modern applications are highly distributed making use of various architecture and integration styles described in:

Java integration styles and Java architectures. A typical enterprise application will make use of a combination of these integration styles and architecture. For example, here is a very simplified trading application making use of "**synchronous**" and "**asynchronous**" calls. This allow traders to place buy/sell trades online.



FIX to send trades to stock exchange

as you can see, it makes use of

1) Message Oriented Middle wares (MOM) like WebMethods or Tibco to publish or subscribe messages (i.e. trades).
2) Web services to perform CRUD (Create Read Update and Delete) operations.
3) MVC architecture to display data on the GUI to provide a user interface.
4) JDBC to persist/read user interactions via CRUD operations.
5) FIX protocol to communicate to the stock exchange. It is a standard to exchange financial information.

Popular Member Posts

♦ 11 Spring boot interview questions & answers

850 views

♦ Q11-Q23: Top 50+ Core on Java OOP Interview Questions & Answers

768 views

001A: ♦ 7+ Java integration styles & patterns interview questions & answers

398 views

18 Java scenarios based interview Questions and Answers

387 views

♦ 7 Java debugging interview questions & answers

307 views

01b: ♦ 13 Spring basics Q8 – Q13 interview questions & answers

305 views

01: ♦ 15 Ice breaker questions asked 90% of the time in Java job interviews with hints

297 views

♦ 10 ERD (Entity-Relationship Diagrams) Interview Questions and Answers

293 views

♦ Q24-Q36: Top 50+ Core on Java classes, interfaces and generics interview questions & answers

246 views

001B: ♦ Java architecture & design concepts interview questions & answers

204 views

Bio | **Latest Posts**

Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via Amazon.com in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site.**945+** paid members. join my LinkedIn Group. **Reviews**

About Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job

interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via Amazon.com in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site.**945+** paid members. join my LinkedIn Group. **Reviews**

‹ Graph from scratch Java example adjacent matrix approach

Two Pointers: Partitioning around a pivotal number ›

Posted in Java Architecture Interview Q&A, member-paid

Empowers you to open more doors, and fast-track

Technical Know Hows

☀ Java generics in no time ☀ Top 6 tips to transforming your thinking from OOP to FP ☀ How does a HashMap internally work? What is a hashing function?
☀ 10+ Java String class interview Q&As ☀ Java auto un/boxing benefits & caveats ☀ Top 11 slacknesses that can come back and bite you as an experienced Java developer or architect

Non-Technical Know Hows

☀ 6 Aspects that can motivate you to fast-track your career & go places ☀ Are you reinventing yourself as a Java developer? ☀ 8 tips to safeguard your Java career against offshoring ☀ My top 5 career mistakes

Prepare to succeed

☀ Turn readers of your Java CV go from “Blah blah” to “Wow”? ☀ How to prepare for Java job interviews? ☀ 16 Technical Key Areas ☀ How to choose from multiple Java job offers?

Select Category ▼

© Disclaimer

These are general advice only, and one needs to take his/her own circumstances into consideration. The author will not be held liable for any damages caused or alleged to be caused either directly or indirectly by these materials and resources. Any trademarked names or labels used in this blog remain the property of their respective trademark owners. No guarantees are made regarding the accuracy or usefulness of content, though I do make an effort to be accurate. Links to external sites do not imply endorsement of the linked-to sites.

↑