Today

Distributed Objects (Lesson 6)

- ✓ Spring Kernel
- ✓ Java RMI
- ..... EJB

Distributed Subsystems (Lesson 7)

- * Global memory system

Friday

Yousef Khalidi Guest Lecture

Today

Distributed Objects (Lesson 6)

✓ Spring Kernel

✓ Java RMI

..... EJB

Distributed Subsystems (Lesson 7)

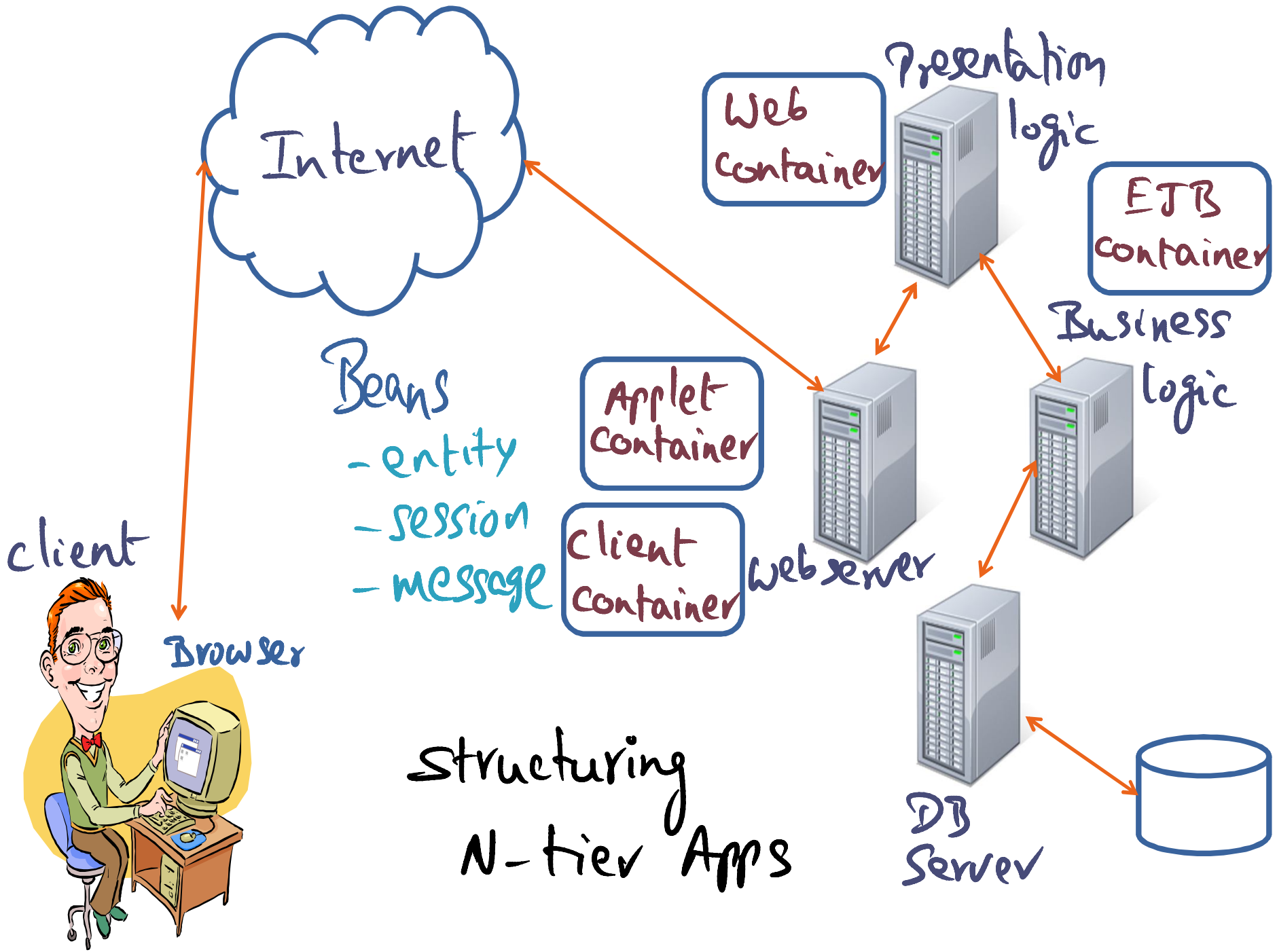
* Global memory system

Friday

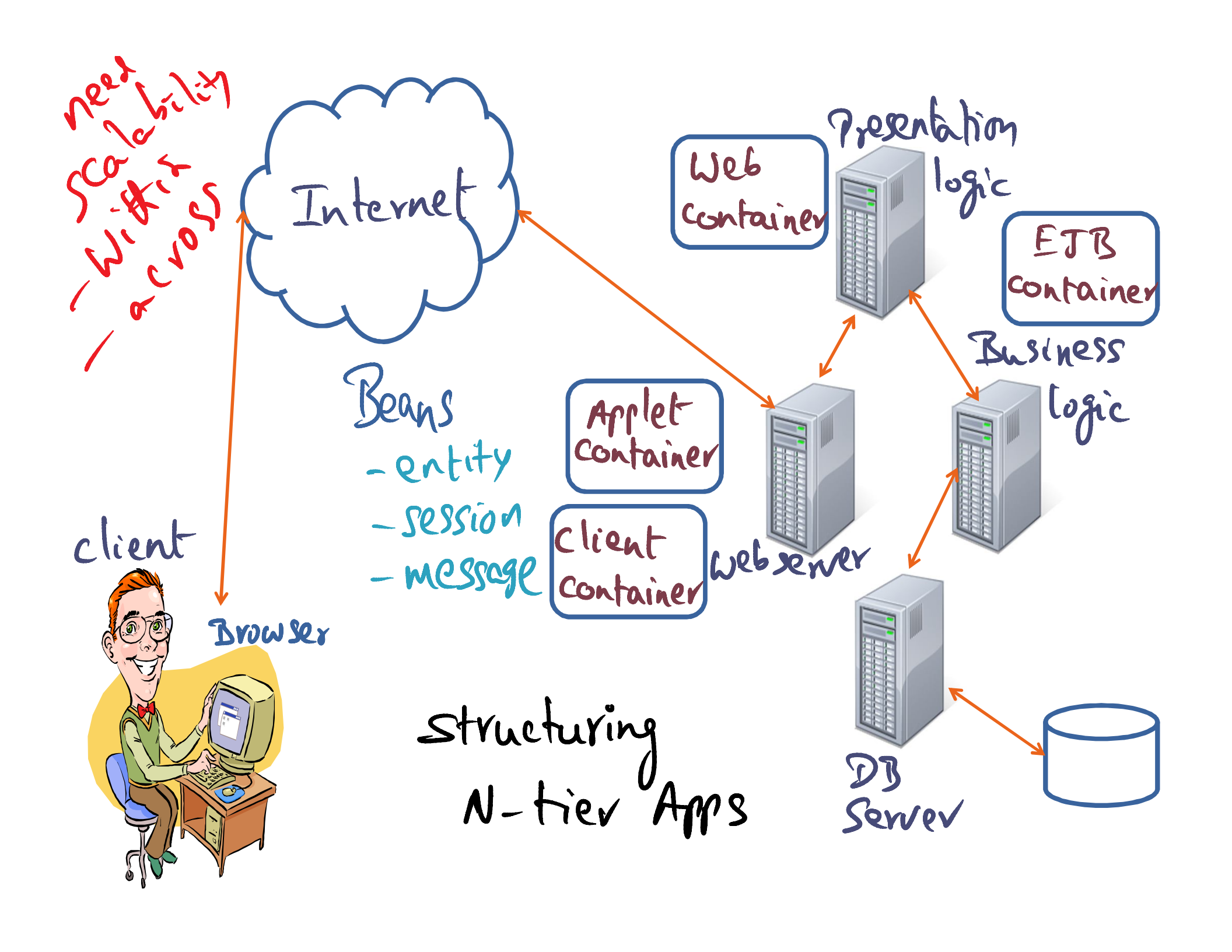
Yousef Khalidi Guest Lecture

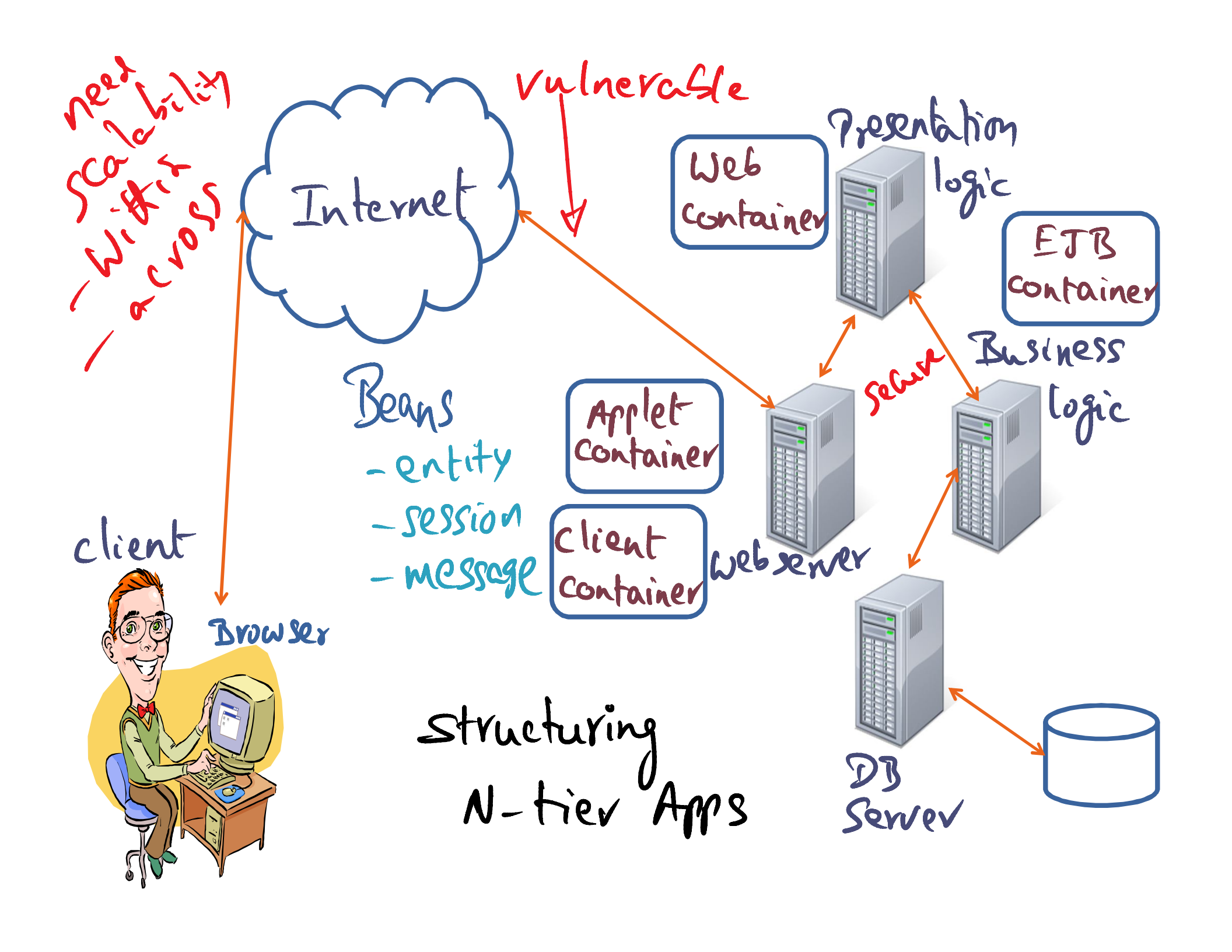
+

Pizza !!

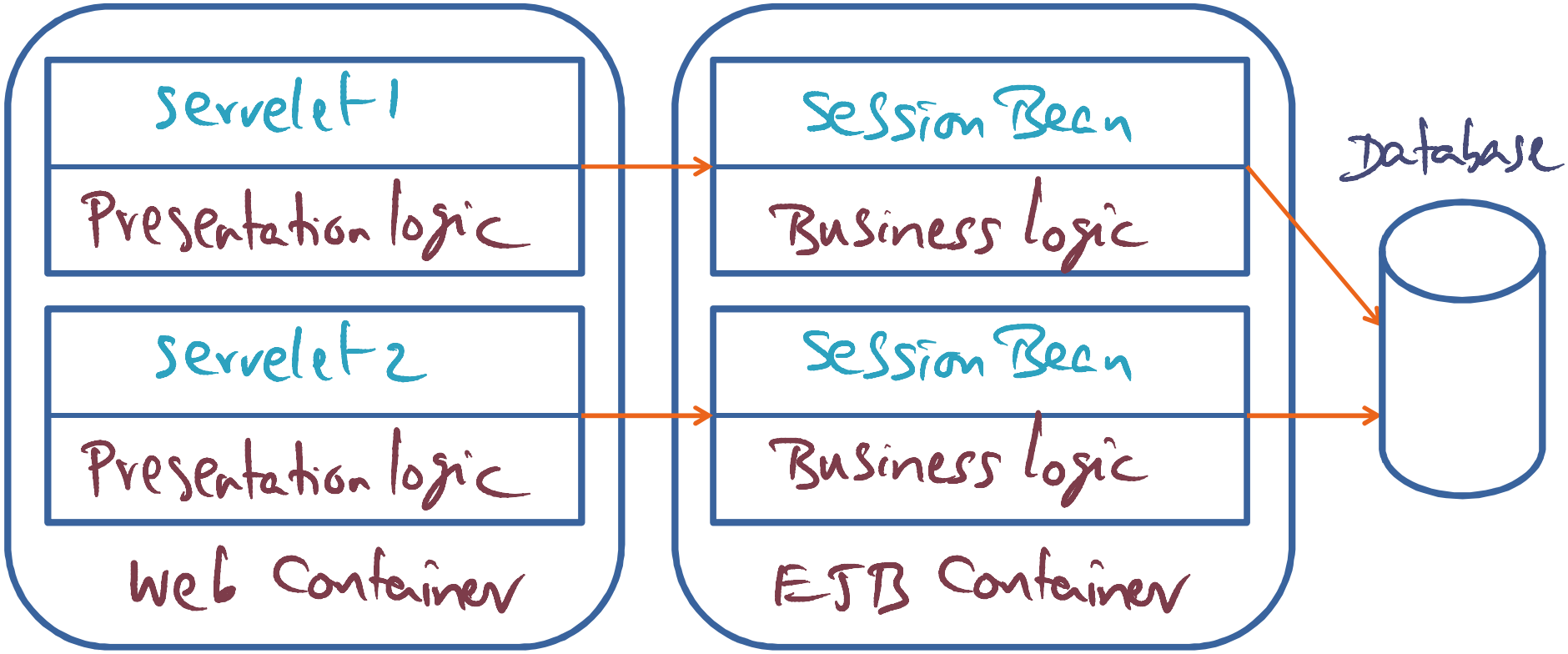


Structuring N-tier Apps


need
scalability
- within
across
Internet
Web container
Presentation logic
EJB container
Business logic
Beans
- entity
- session
- message
Applet container
Client container
Web server
client
Browser
DB Server
Structuring
N-tier Apps


need scalability
- within
- across
Internet
vulnerable
Web container
Presentation logic
EJB container
Business logic
Beans
- entity
- session
- message
Applet container
Client container
secure
Web server
client
Browser
Structuring N-tier Apps
DB Server

Design Alternative 1 : coarse grain session Beans



Pros

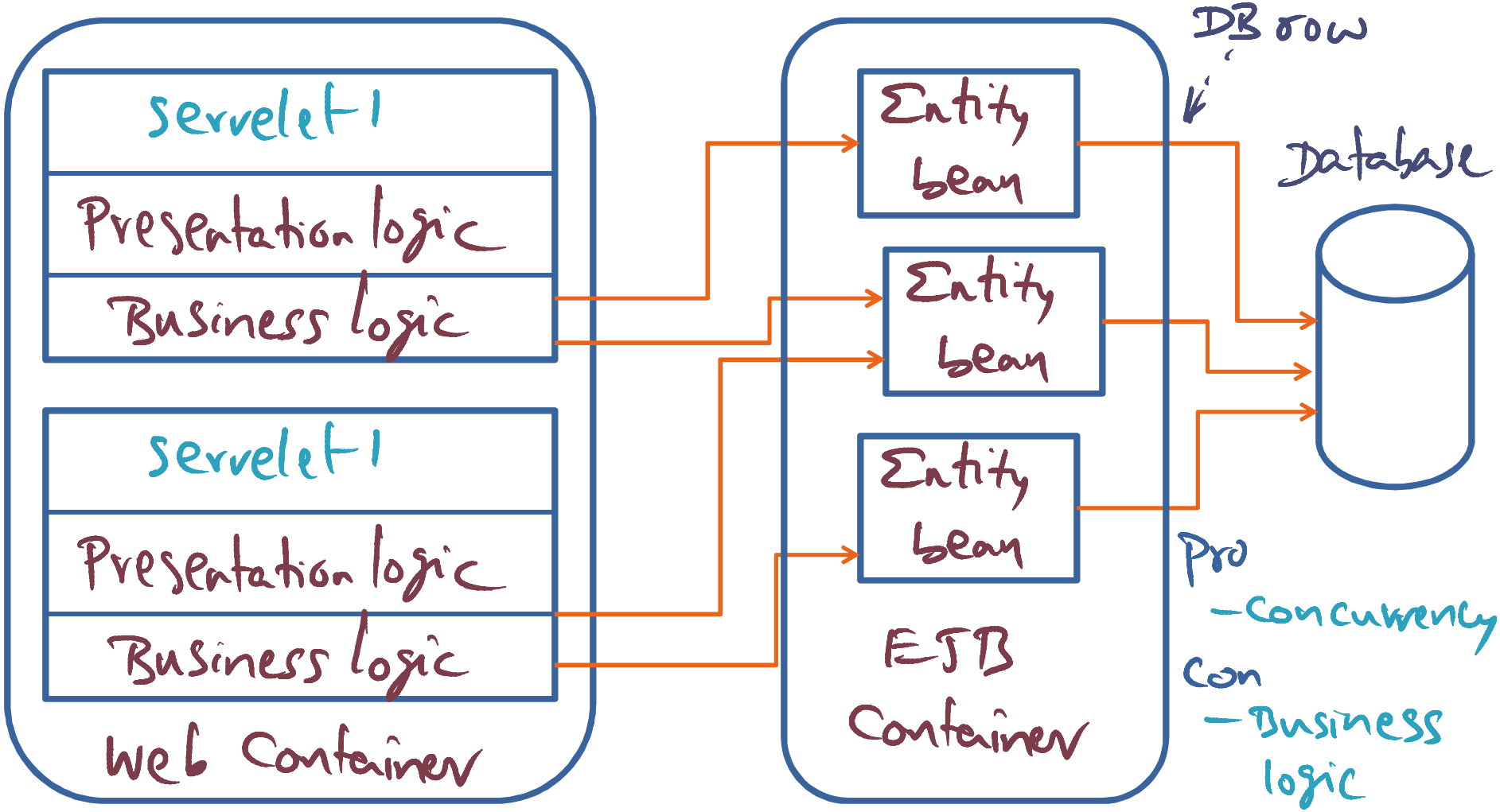
- minimal container services
- Business logic not exposed

Cons

- app structure akin to "monolithic kernel"

scalability "across" not within

Design Alternative 2: Data Access Object



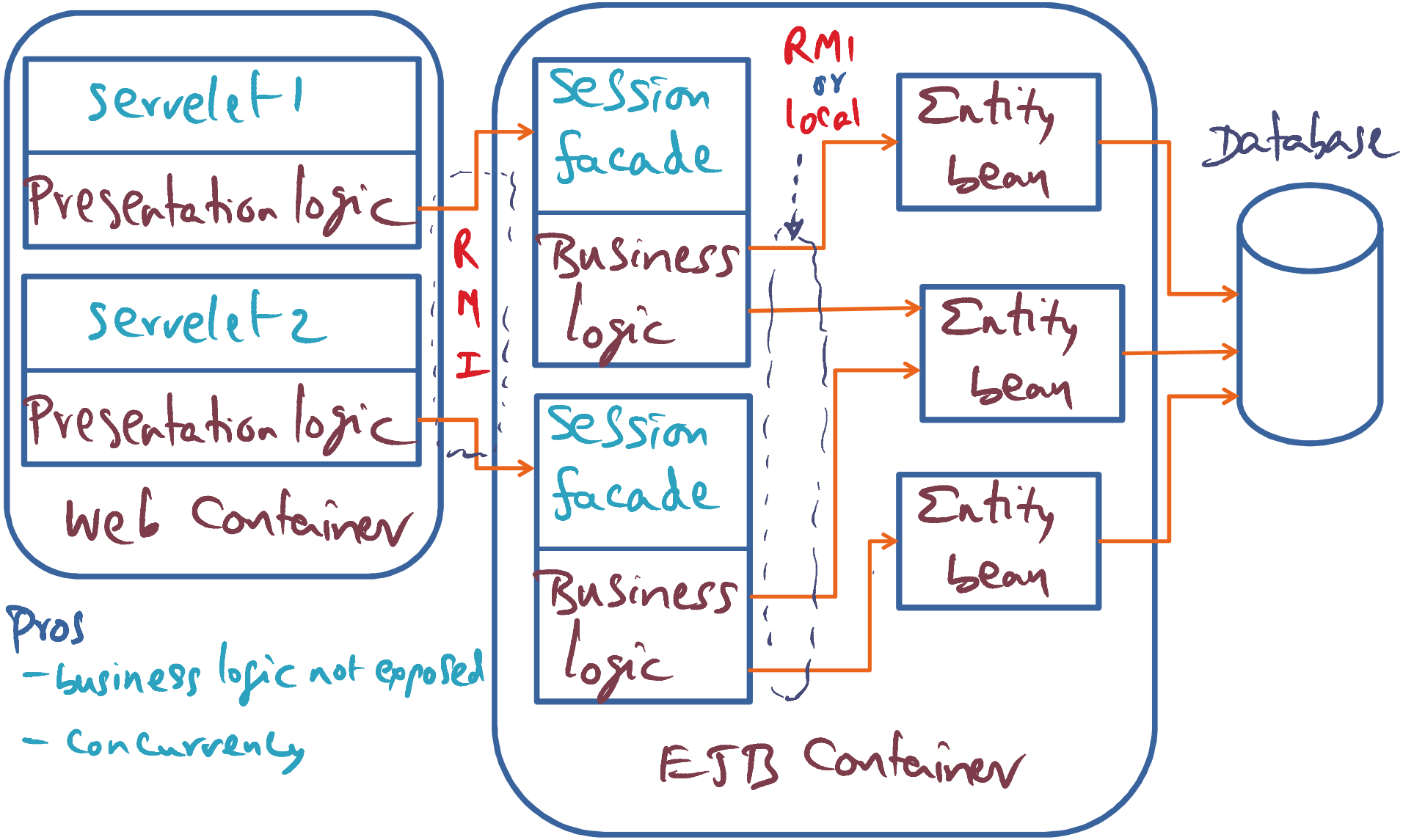
Pro
- Concurrency

Con
- Business logic exposed

DAO using Entity bean → CMP
DAO using Entity bean → BMP

Scalable for both within + across

Design Alternative 3 : Session bean with entity bean



Pros

- Business logic not exposed
- Concurrency

Key takeaways

- Power of the object technology for structuring complex application servers
- EJB allowed developers to write business logic without worrying about cross-cutting concerns like security, logging, persistence, etc.

Expectations:

- Understand the design choices presented in this lesson and analyze their performance implications qualitatively
- Read the paper and relate them to the design choices and your own qualitative analysis

Caveat: EJB has evolved considerably from the time of this paper but still the principles discussed in this paper apply to the way complex N-tier application servers are built to this day.