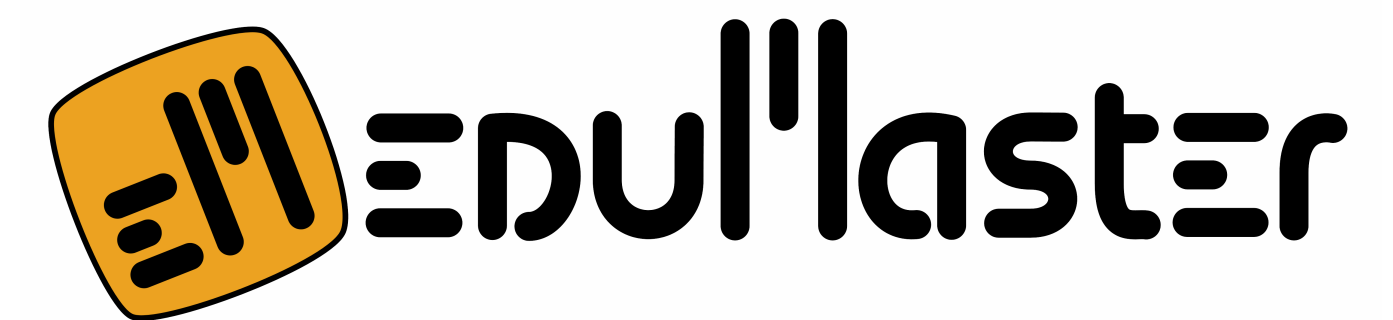


**Java Developers &
Solaris Administrators**

DDAY



ORACLE®



Guide to Architecture of JEE Applications

Zdeněk Troníček
Edumaster s.r.o.

Zdeněk Troníček

- Assistant professor at Faculty of Information Technology, Czech Technical University in Prague
- 10+ years experience with Java trainings
- Certifications:
 - Sun Certified Java Programmer
 - Sun Certified Web Component Developer
 - Sun Certified Business Component Developer
 - Sun Certified Specialist for NetBeans IDE
- Blog
 - <http://www.java.cz/blog/tronicek>
- Project
 - <http://kenai.com/projects/refactoringng>

Agenda

- Introduction
- Architecture decisions
- Application server
- Conclusion



Agenda

- Introduction
- Architecture decisions
- Application server
- Conclusion



Software architecture

IEEE 1471

“The fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution.”

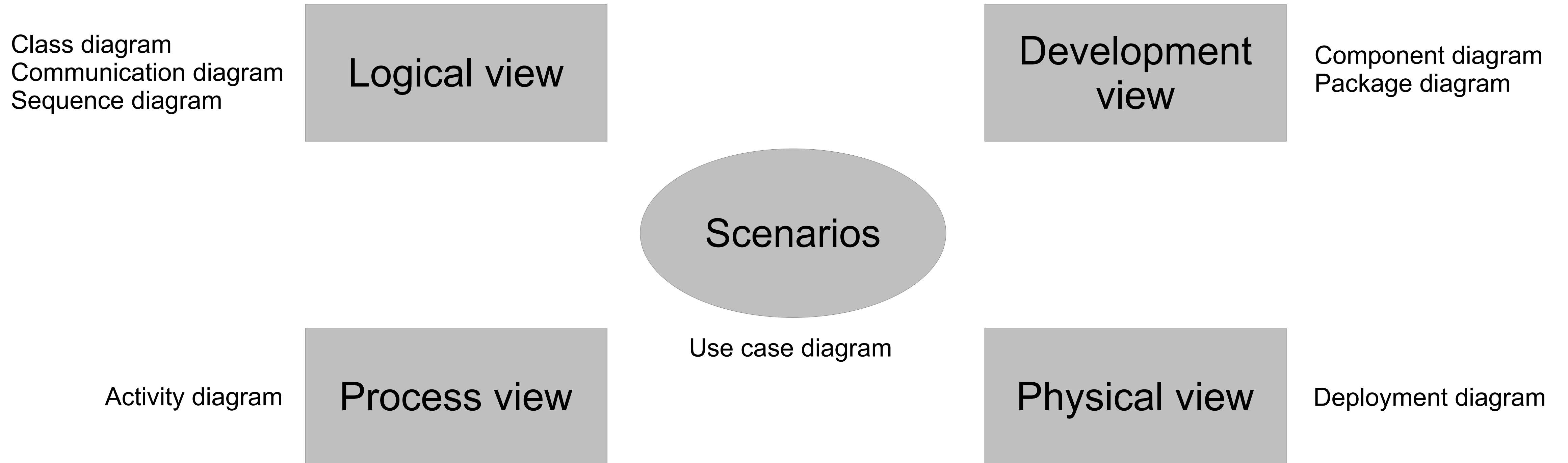
Architecture vs. Design

	Architecture	Design
Abstraction level	High and broad focus on a few details	Low and specific focus on many details
Deliverables	System and subsystem plans, architectural prototype	Component design, code specifications
Area of focus	Non-functional requirements, risk management	Functional requirements

Non-functional requirements

- **Business rules**
e.g. verification against a given database
- **Constraints**
e.g. access through HTTP
- **Systemic qualities**
e.g. performance, availability, usability, scalability, flexibility

4+1 Architectural View Model



Architecture styles

- Service Oriented Architecture, Message bus
- Client / server, N-Tier, 3-Tier
- Component-based, Layered

High Level Concerns

- How will the users be using the application?
- How will the application be deployed into production?
- What are the systemic quality requirements?
- How can we design the application to be flexible and maintainable over time?

Key Architecture Principles

- Design to change, not to last
- Identify risks and reduce them
- Use models and visualizations

Key Design Principles

- Separation of concerns
- Single responsibility principle
- Principle of least knowledge
- Don't repeat yourself

Agenda

- Introduction
- **Architecture decisions**
- Application server
- Conclusion



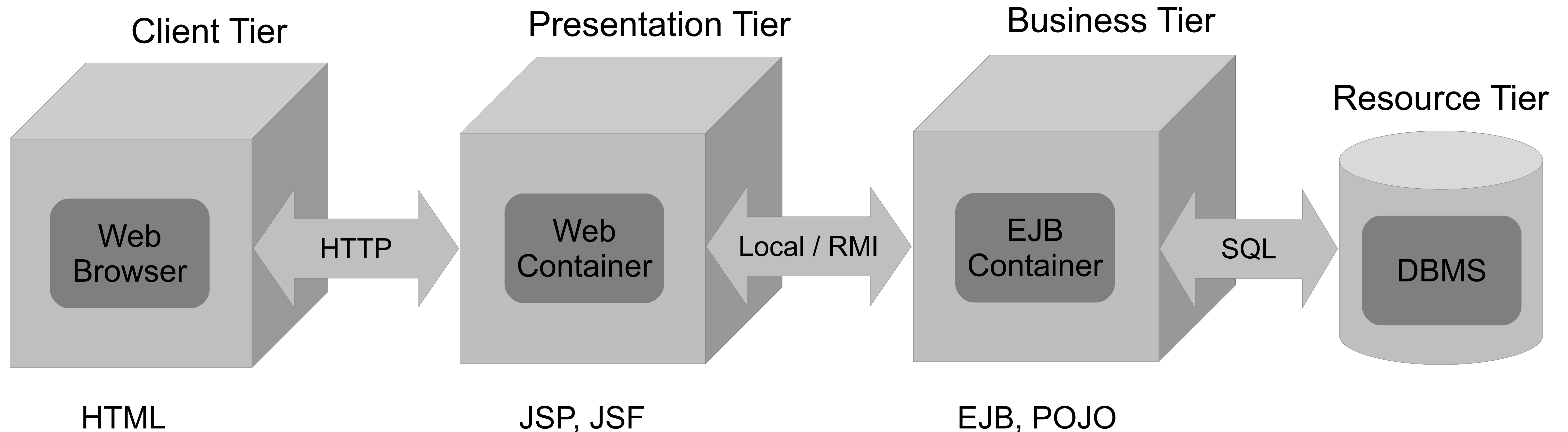
Major Decisions

- Application type
- Deployment strategy
- Technologies
- Systemic qualities
- Crosscutting concerns

Archetypes

- Web application
- Rich Internet application
- Rich client application
- Mobile application
- Service application

Web Application

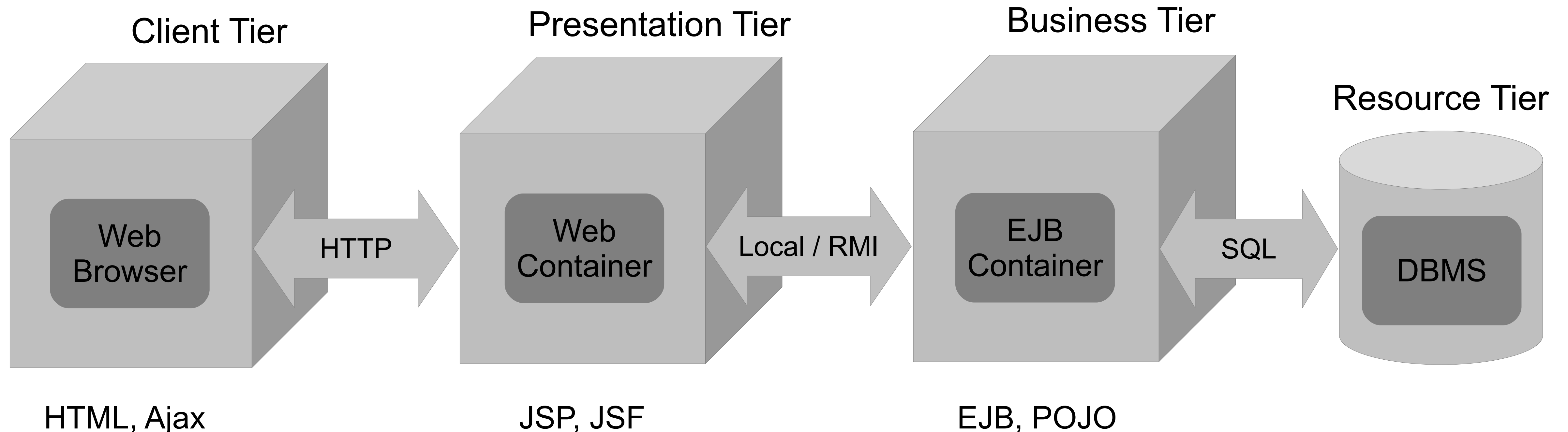


+ modest client hardware
+ portability

+ simple upgrade
+ simple management

- simple UI
- connected scenarios only

Rich Internet Application

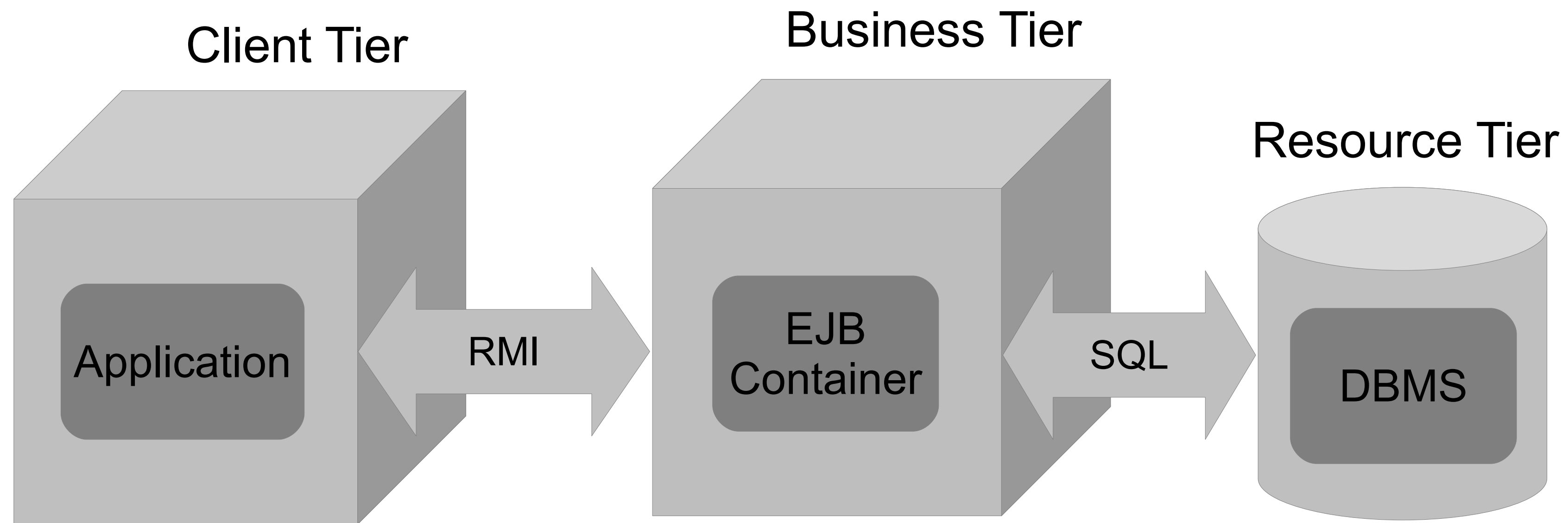


+ rich UI
+ support for streaming media

+ simple upgrade
+ simple management

- requires runtime framework

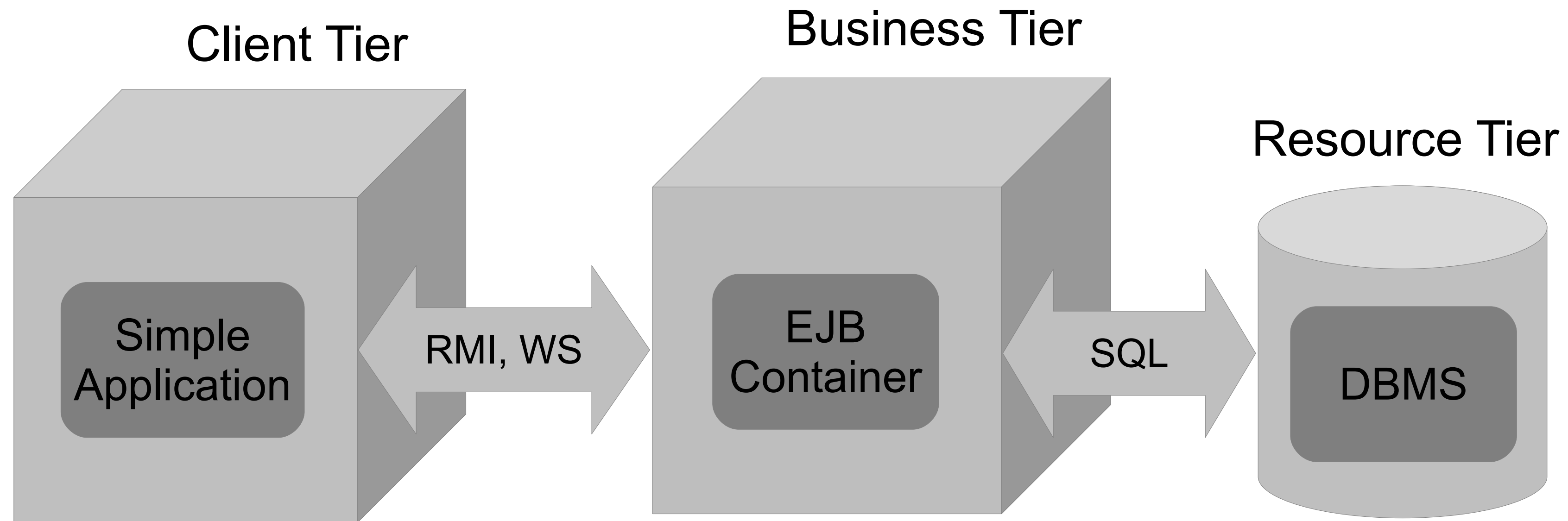
Rich Client Application



- + rich UI
- + interactive and responsive UI
- + offline support

- upgrade
- management

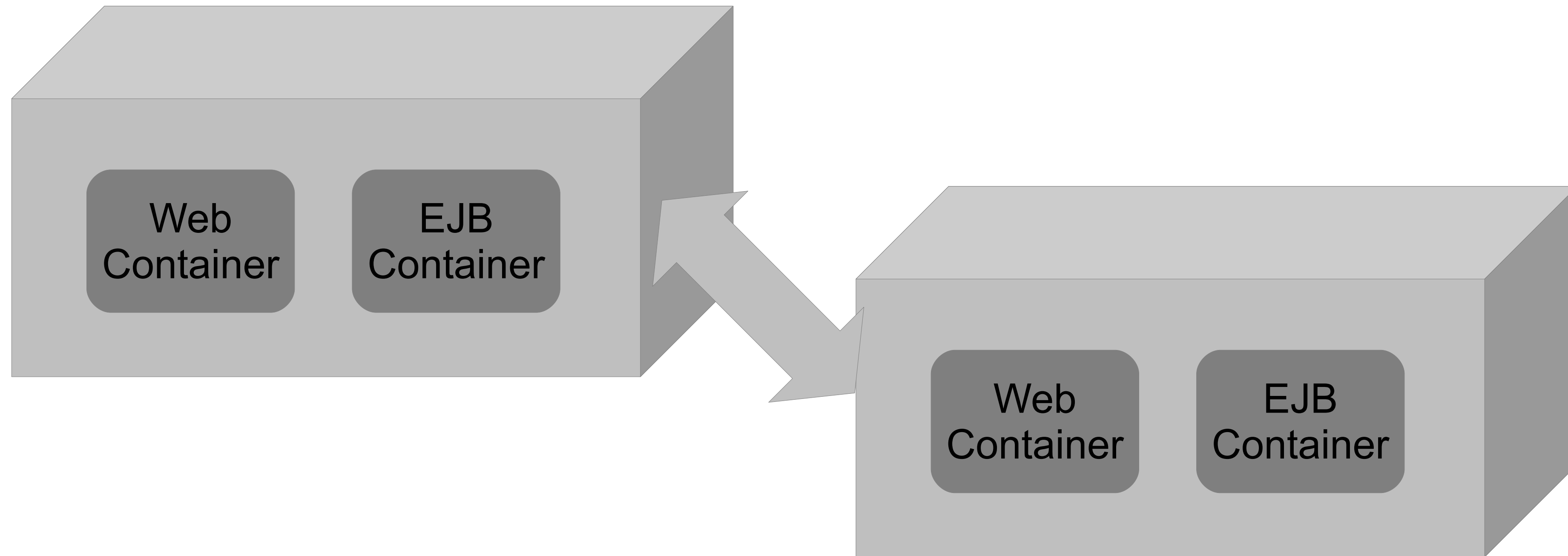
Mobile Application



+ support for handheld devices
+ offline support

- simple UI
- upgrade
- management

Service Application



+ loose coupling
+ no UI

- XML processing

Deployment strategy

Deployment patterns

- Client-server deployment
- 3-tier deployment
- 4-tier deployment
- Load-balanced cluster
- Failover cluster

Technologies

What should you consider?

- Application type
- Deployment topology
- Architecture style
- Organization policies
- Infrastructure limitations
- Team skills

Systemic qualities

- What are the key systemic qualities?
- What are the requirements for these qualities? Are they quantifiable?
- What are the acceptance criteria for these qualities?
- Tradeoffs: e.g. security vs. performance, availability vs. manageability

Crosscutting concerns

Crosscutting concerns are key areas that are not related to a specific layer. For example:

- Logging
- Authentication and authorization
- Exception management
- Communication
- Caching

Agenda

- Introduction
- Architecture decisions
- **Application server**
- Conclusion



JEE profiles

Full profile vs. Web profile:

- EJB Lite (no MDB, no remote beans, no EJB timers,...)
- Java Message Service
- Java Mail
- Connector
- Web Services (JAX-WS, JAX-RS)
- JAXB
- JACC
- ...

Application server selection criteria

- Is the server compliant with JEE 6?
- What are the license costs? Is technical support available?
- What version of JRE is the server running on? Can JRE be upgraded?
- What are the published values for systemic qualities (scalability, availability,...)?
- Can you run multiple instances on a single physical server (virtual servers)?
- Does the server support remote application monitoring?
- How flexible the server is? For example, can you tune the number of threads?
- Does the application server support clustering?

Final question

Are you a JEE architect yet?

Questions & answers

