

Web Applications: Overview and Architecture

Computer Science and Engineering ■ College of Engineering ■ The Ohio State University

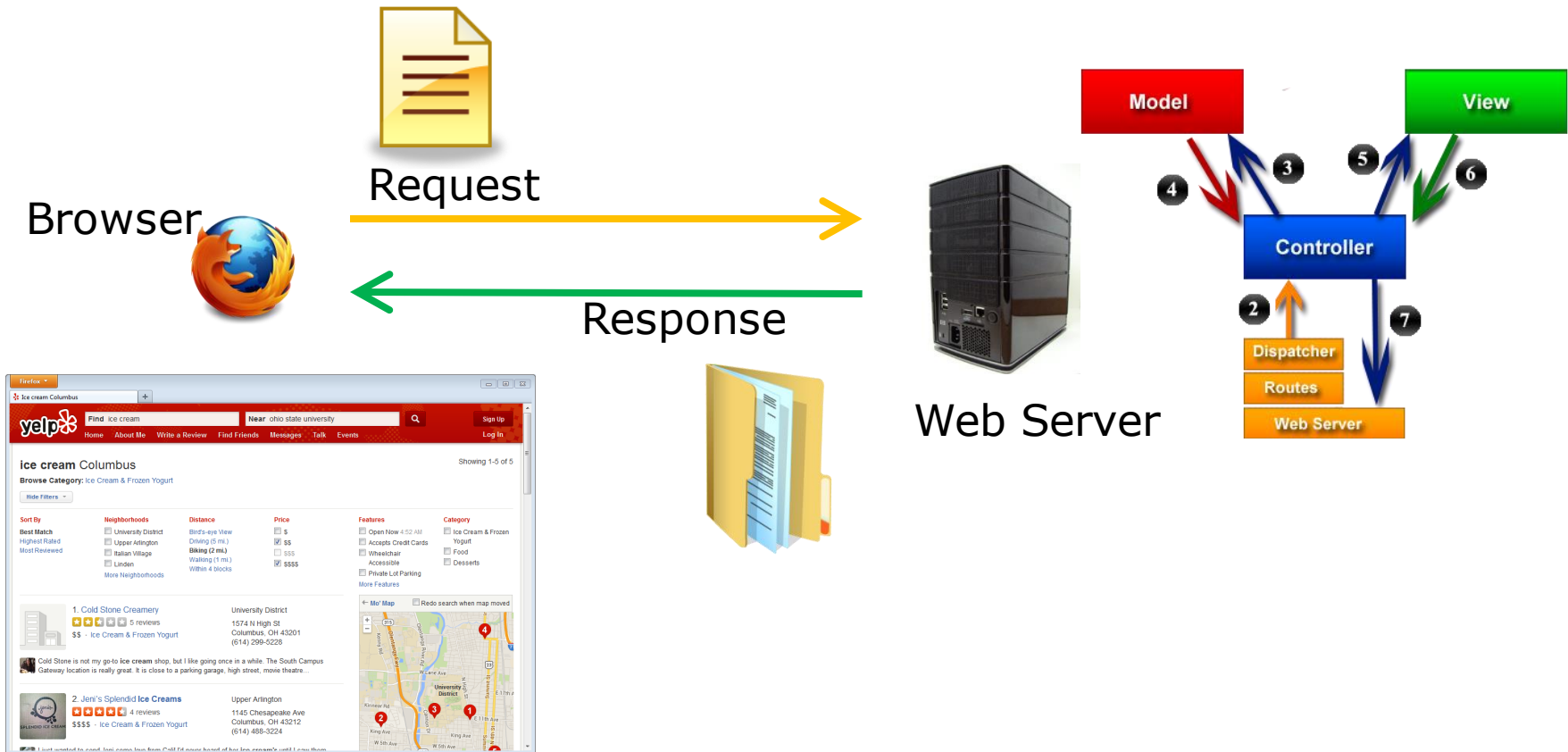
Lecture 1

Road Map in Pictures: Web App

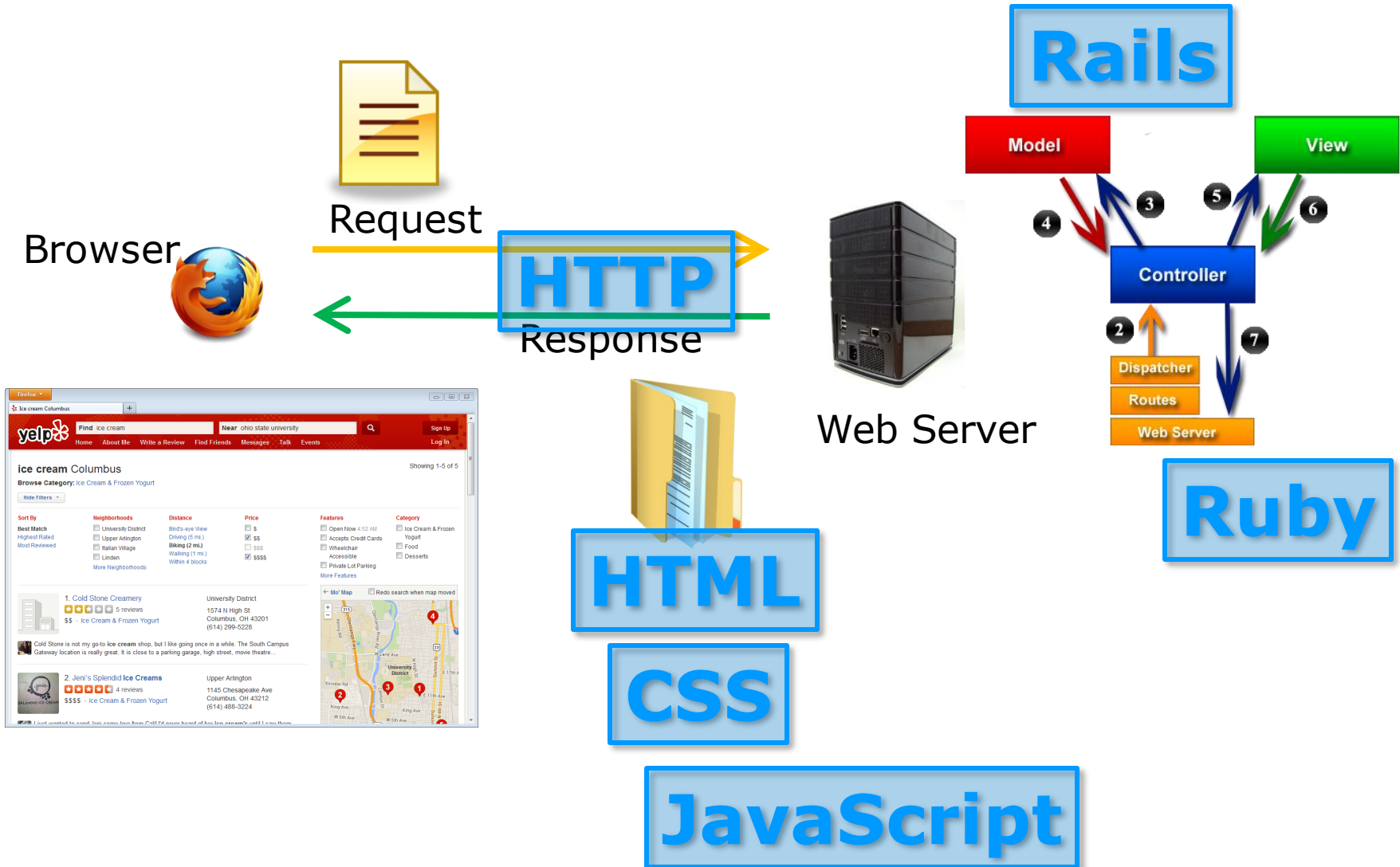
The screenshot shows a web browser window with the following elements:

- Browser:** Firefox, address bar shows "Ice cream Columbus".
- Yelp Header:** Search for "ice cream" near "ohio state university". Navigation links: Home, About Me, Write a Review, Find Friends, Messages, Talk, Events. Sign Up, Log In.
- Search Results:**
 - ice cream Columbus** (Showing 1-5 of 5)
 - Browse Category:** Ice Cream & Frozen Yogurt
 - Hide Filters** button
 - Sort By:** Best Match, Highest Rated, Most Reviewed
 - Neighborhoods:** University District, Upper Arlington, Italian Village, Linden, More Neighborhoods
 - Distance:** Bird's-eye View, Driving (5 mi.), Biking (2 mi.), Walking (1 mi.), Within 4 blocks
 - Price:** \$, \$\$, \$\$\$, \$\$\$\$
 - Features:** Open Now 4:52 AM, Accepts Credit Cards, Wheelchair Accessible, Private Lot Parking, More Features
 - Category:** Ice Cream & Frozen Yogurt, Food, Desserts
- Results:**
 - 1. Cold Stone Creamery**
 - University District
 - 1574 N High St, Columbus, OH 43201 (614) 299-5228
 - 5 reviews, \$\$\$ - Ice Cream & Frozen Yogurt
 - Review snippet: "Cold Stone is not my go-to ice cream shop, but I like going once in a while. The South Campus Gateway location is really great. It is close to a parking garage, high street, movie theatre..."
 - 2. Jeni's Splendid Ice Creams**
 - Upper Arlington
 - 1145 Chesapeake Ave, Columbus, OH 43212 (614) 488-3224
 - 4 reviews, \$\$\$\$ - Ice Cream & Frozen Yogurt
 - Review snippet: "I just wanted to send Jeni some love from Calif. I'd never heard of her ice cream's until I saw them..."
- Map:** Mo' Map showing the locations of the two shops (marked 1 and 2) in the University District area of Columbus, OH. Other streets shown include W Lane Ave, N High St, Summit St, E 11th Ave, E 17th Ave, King Ave, W 5th Ave, and Olentangy Fwy.

Road Map in Pictures



Road Map in Pictures



Road Map: Schedule of Topics

- A Language
 - Ruby
- Foundations
 - Version Control, Networking, Regular Expressions
- Static web pages
 - HTML & CSS
- Dynamic web pages
 - JavaScript
- Framework for web applications
 - Rails
- Applied Topics
 - Security, Encodings

Resources

- Class website
 - Syllabus (note exam requirement)
 - Handouts, lecture notes, lab assignments
 - Pointers to more resources
- Piazza
 - Discussion forum, news, announcements
- Slack
 - Group collaboration, messaging, chat
- Carmen
 - Grades
- *Face time* (not FaceTime™)
 - Instructor, TA
 - Each other

Technical Content

- Languages and Technologies
 - HTTP
 - XML, HTML, CSS, JavaScript
 - Ruby, Ruby on Rails
- Tools and techniques
 - Design patterns (MVC)
 - git, linux
 - Regular expressions, unicode, system time
- Advanced topics
 - Programming languages, networking, cryptography, databases, operating systems

Stability of Content: Concepts

- Conceptual underpinnings will be relevant forever
- In this course:
 - Single-point of control over change
 - Abstraction (vs realization)
 - Design patterns
 - Regular Expressions (the math part)
 - Cryptography (the math part)
 - Motivation for version control
 - Time-space performance trade-offs

Stability of Content: Technology

- Some technologies have been around a long time, and will likely be relevant for many more years
- Examples in this course:
 - Linux
 - SQL
 - HTTP
 - HTML
 - CSS
 - JavaScript

Stability of Content: Tools

- Some tools come and go
- They are useful for getting things done now, but may not be as relevant or fashionable in 10 years
- Examples in this course
 - Ruby
 - JQuery
 - git

Stability of Content: Framework

- There are many frameworks and libraries for web development
- They come and go so quickly, there is always something new
- Examples:
 - Web frameworks like Rails, Express.js...
 - Ruby gems like Middleman, Nokogiri, Cucumber...
 - JavaScript libraries like Angular, React
 - HTML/CSS libraries like Bootstrap, Baseline, Foundation...

Meta Content: Software Eng.

- Lasting relevance
- Project development in the "real world"
 1. Vague open-ended requirements
 2. Large, complex problems
 3. Teams

Topic 1: Vague Requirements

- Two aspects to engineering:
 - Satisfying the constraints (solving the problem)
 - Optimizing the solution (better, faster, cheaper)
- Must first identify and understand the problem
 - Requirements elicitation
- Recognize tradeoffs
 - Improvement in one aspect at the expense of another

Topic 2: Size and Complexity

- “Programming in the large”
 - Does not all fit in one person’s head or schedule
 - Interfaces, modules, components, classes
- Design
 - Measure twice, cut once
- Process
 - Agile, waterfall, TDD,...
- Documentation
- Testing

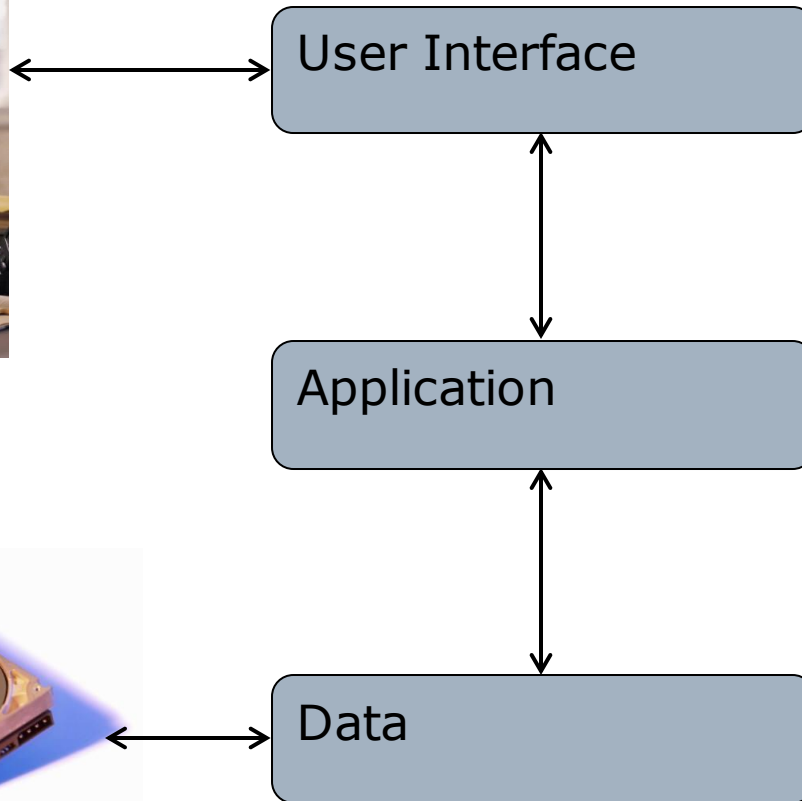
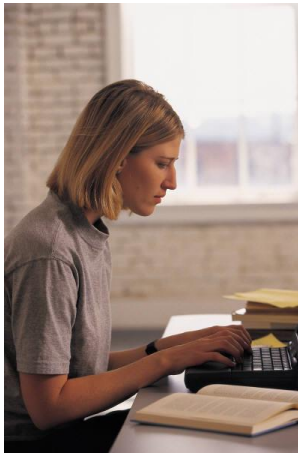
Topic 3: Group Work

- Naïve view of CS: Lone wolf hacker
- Reality: large multidisciplinary teams
 - Developers, testers, marketing, HR, management, clients
 - Communication skills are critical
- Many challenges
 - Rely on others
 - Compromises become necessary
 - Personalities
- Many rewards
 - Accomplish more
 - Learn more

In This Course...

- Group work: 4-5 people / group
 - Your “home group” for project
- Multidisciplinary teams
 - Cross-cutting technical areas
- Open-ended projects
- Communication skills
 - Presentations to class

Architecture: Desktop App



Graphical events
(mouse moves,
button pushed)

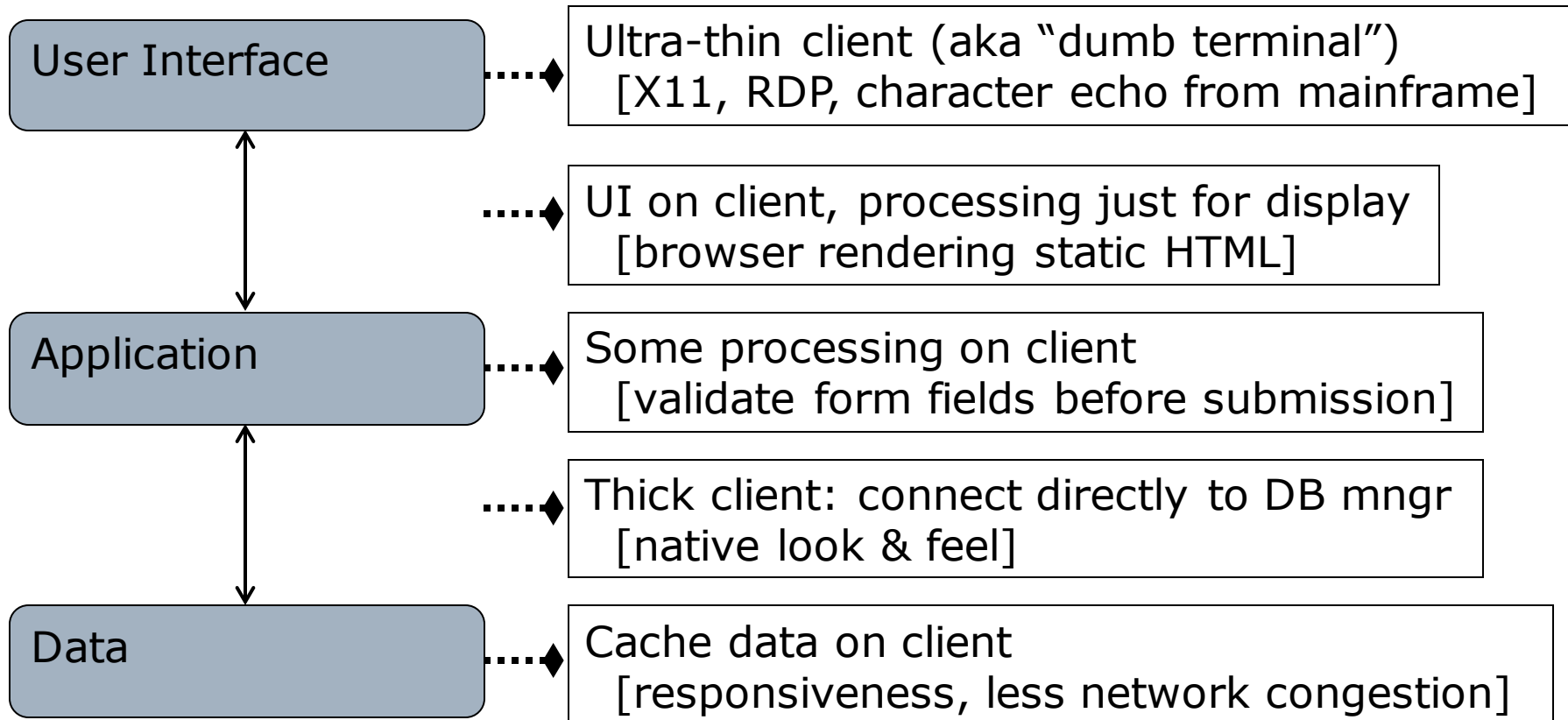
Processing,
Calculating

Persistence,
Transactions,
Triggers



Client-Server App: 2-Tier

Where should we cut?



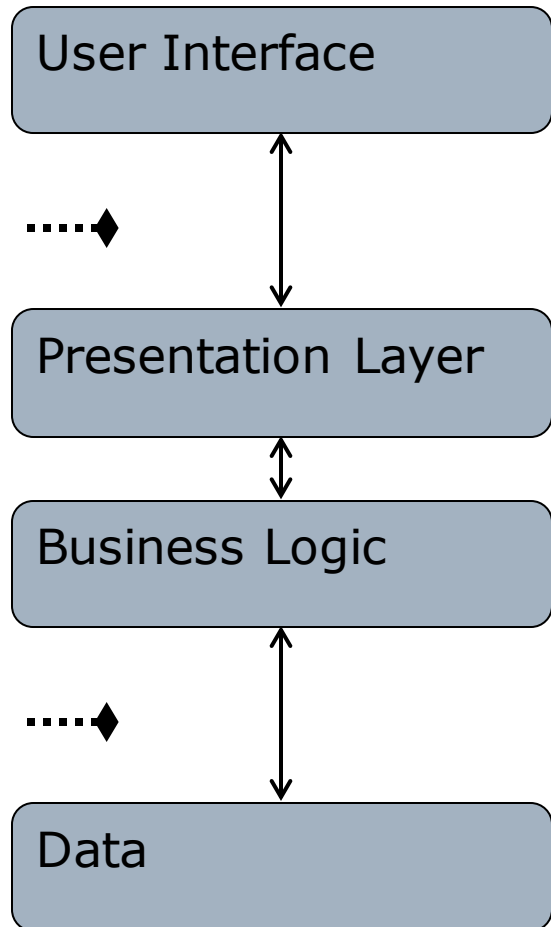
Basic Web App Skeleton: 3-Tier



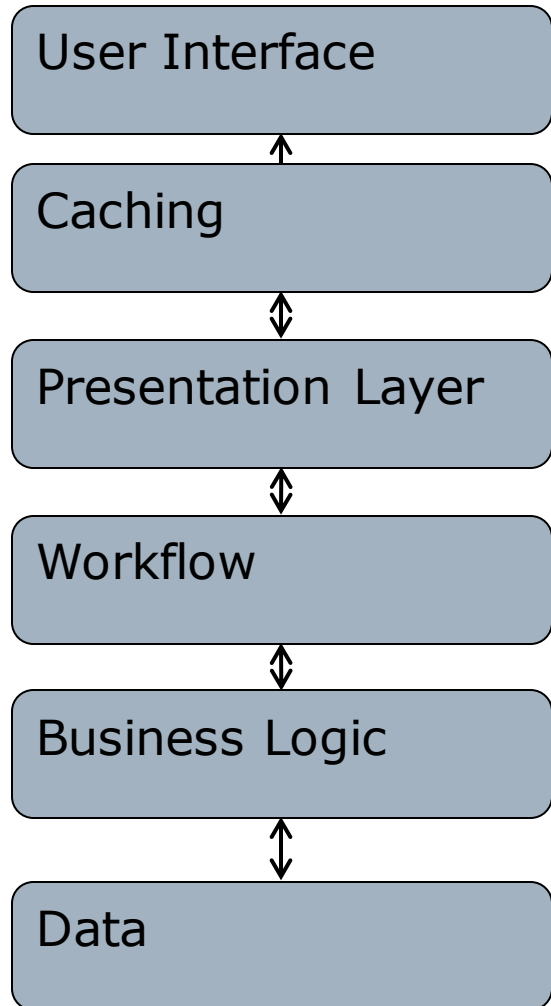
Advantages over Thick Clients

- Performance
 - 1 (expensive) network call to app layer results in many calls to data layer
 - Compute-intensive part on faster machine
- Flexibility
 - Update app logic without changing client
- Robustness
 - Transactions, logging at app level
- Security
 - Login, authentication, encryption all better at app level than data level

Web App Skeleton: 4-Tier



Web App Skeleton: n-Tier...

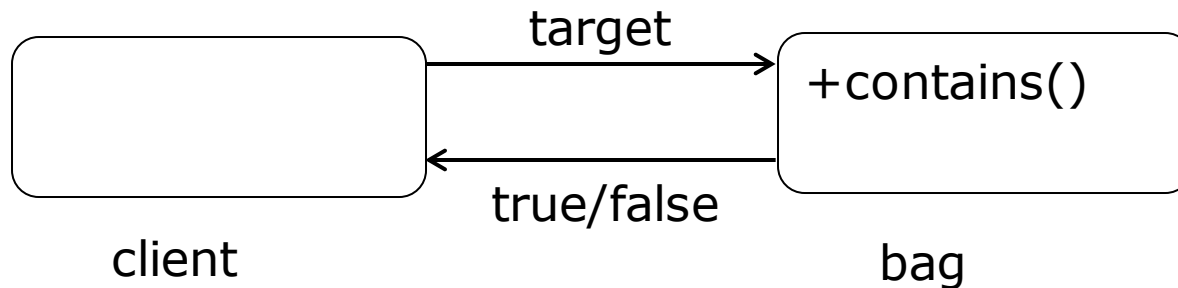


Summary

- Technical aspects of course content
 - Many different web technologies
 - Rapidly evolving landscape
- Meta content: Software engineering
 - Vague requirements
 - Large systems
 - Teams
- 2-, 3-, 4-, n-Tier Architectures

Familiar Example

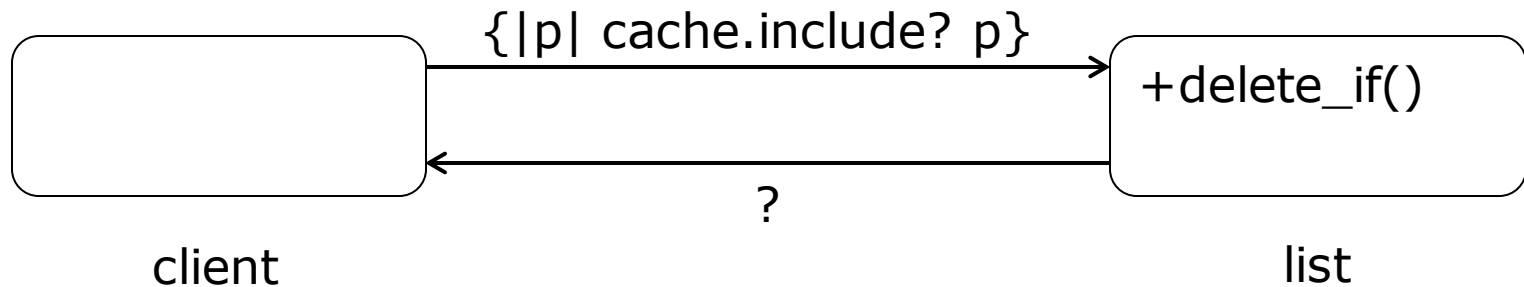
- Calling a method on an object
`result = bag.contains(target)`



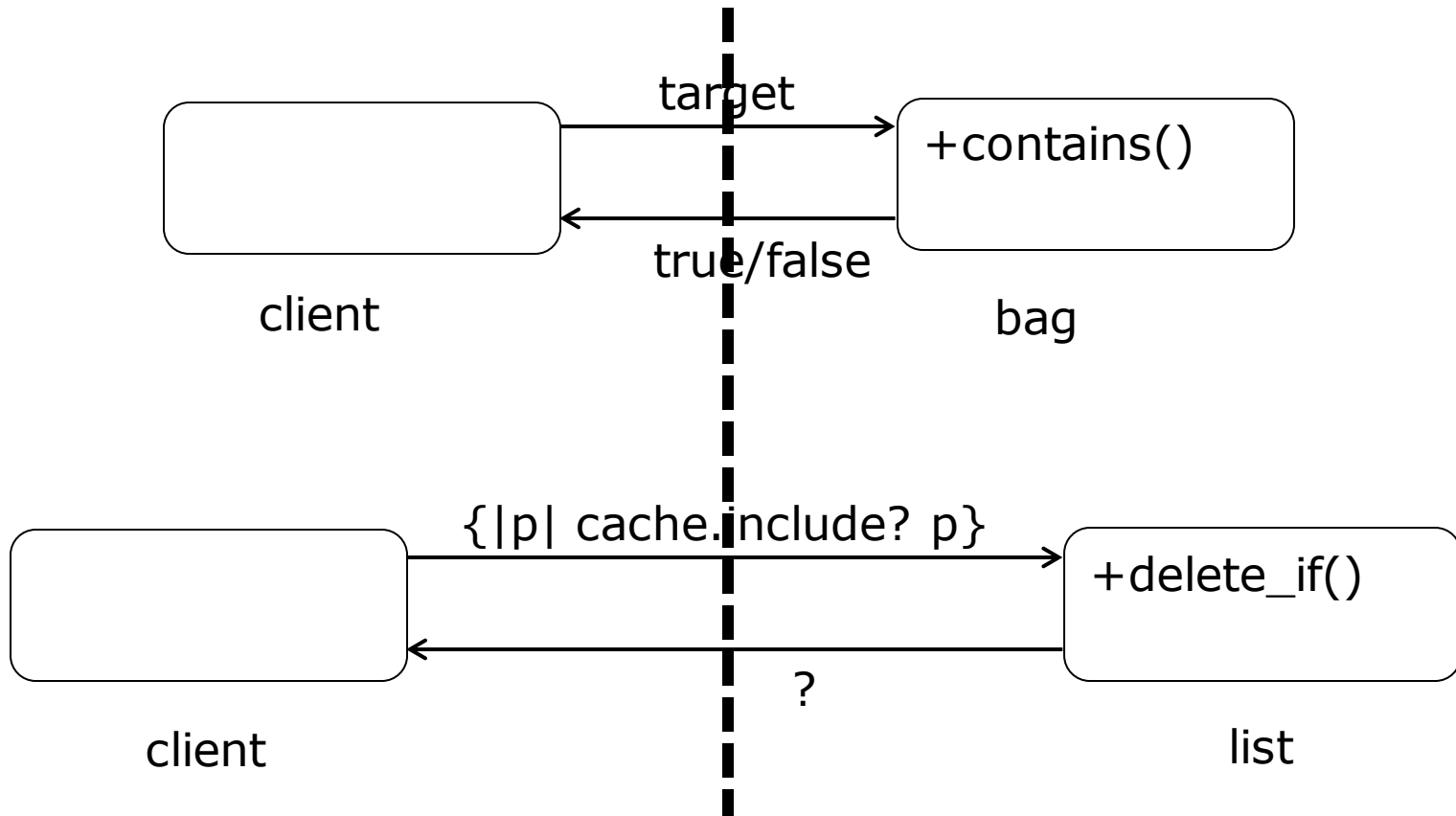
Familiar Example

- Calling a method on an object

```
list.delete_if {|p| cache.include? p}
```



Small Change



Distributed Systems

- Common object-oriented metaphor:
Send message to receiver object
- Natural mapping to distributed system
 - Send message across network to server
 - Response (eventually) comes back
- But this small step changes everything
- Problems that were trivial before,
become hard or even impossible!

Two Generals Problem

