Server side web application roundup

Overview

- Web application concepts
- Web application structure
- Specifications vs. implementation
- Case study

Concepts: XML, HTML, and CSS

- Omnipresent:
  - All web frameworks ultimately generate HTML and CSS
  - XML is used everywhere (maybe in too many places)

- We have seen: Static HTML pages, HTML with JSF
- We will see: Dynamic HTML pages with JavaScript

- This knowledge is highly reusable:
  - No matter what framework, no matter what server language

Concept: Request handling

- HTTP request representation in the host language

- Design space (different ways of representing):
  - A ***request*** object could represent all data of a HTTP request
    - Such a request object could have a ***getParameter*** method, etc...
    - Servlet framework is an example of this
  - Values could be ***injected***:
    - Request parameter **foo** becomes value **foo**
      - Field, method parameter, etc..
    - JSF and JAX RS do this
  - ...

Concept: Response handling

- HTTP response representation in the host language
- Design space:
  - Output could be written directly to a stream,
    - Servlets
  - Be generated through a template system, or
    - JSF
  - Be represented as response objects
    - JAX RS

Concept: Server state scopes

- Server state:
  - Shared state: Shared for all requests for all clients
  - Session state: Shared for all requests for a single client
  - Transient state: Shared only during a single request

- We have seen: Beans in JSF, maps in Servlets
- Our knowledge is reusable:
  - Other frameworks and other languages have similar scopes
    - Some may not have all: PHP needs a database for application state
- When learning new frameworks: Find these scopes!

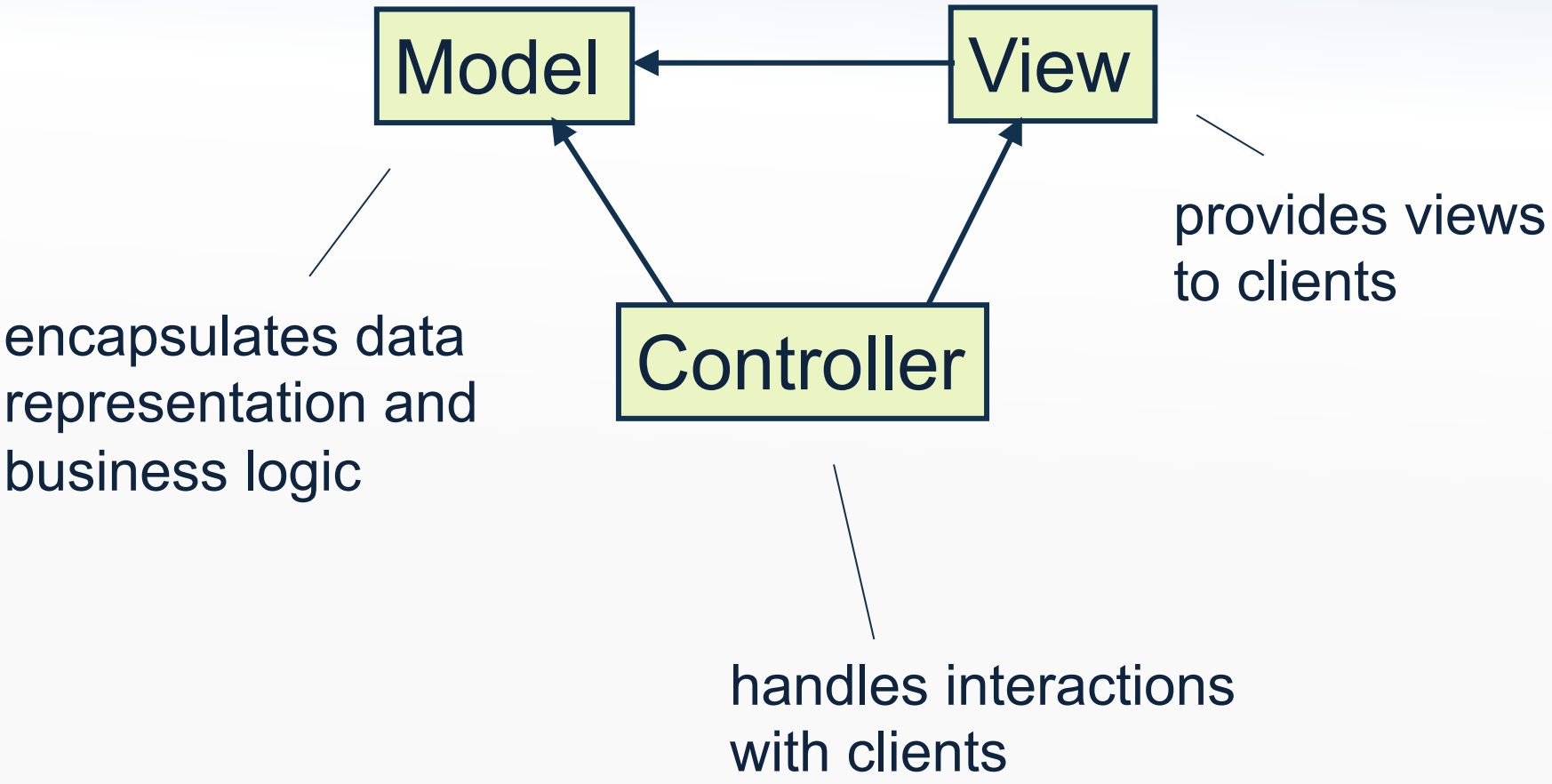
Overview

- Web application concepts
- Web application structure
- Specifications vs. implementation
- Case study

Application structure

- Web applications may be structured as pages:
  - One page to browse items in a shop,
  - One page to show payment handling,
  - ...
- Frameworks provide ways to go from page to page:
  - HTTP Parameters
    - Security?
- Design space examples:
  - Hard-coded URLs and manual parameter passing (Servlets)
  - State machine with value beans (JSF)

Concept: The Model-View-Controller Pattern



More on Model-View-Controller

- MVC dominates the web framework landscape:
  - ASP.NET MVC (by Microsoft)
  - MonoRail for ASP.NET (by Castle Project)
  - **JSF for Java** (by Oracle)
  - Struts for Java (by Apache)
  - Joomla for PHP (by The Joomla Project Team)
  - Yii for PHP
  - And many, many more...

- We can re-use our knowledge:
  - Find out how the model, the view, and the controller work

Many web application frameworks!

- There are dozens (if not hundreds of frameworks)
  - Wikipedia lists:
    - 37 Java frameworks,
    - 28 PHP frameworks,
    - 17 Python frameworks,...
    - (all of them claim to be superior in some way)
  - We cannot cover them all

- Most of these combine things that you have seen now
  - When learning a new one: relate to what you know

Overview

- Web application concepts
- Web application structure
- Specifications vs. implementation
- Case study

Concept: Web services

- Web services: platform-independent communication
  - Expose data via XML over HTTP
  - Very widely used on all platforms
- We have seen: The cloud server, JAX RS

- Our knowledge is reusable:
  - Many web services out there
    - Google maps API, GeoIP, Many, many others...
- Even if you never write another web application after dWebTek, you ***will*** integrate with web services again!

Specification vs. implementation

- We have seen three frameworks:
  - All of them are presented as specifications!
- All of them have multiple implementations:
  - Servlets have *Tomcat*, Glassfish, Jetty, JBoss...
    - Wikipedia lists 24 implementations of the Servlet framework(!)
  - JSF has Mojarra, *MyFaces* and several closed sourced ones
  - JAX-RS has Apache CXF, *Jersey*, RESTEasy, Restlet
- All implementations are different from each other

Specification vs. implementation

- Examples of implementation-specific details:
  - How does the generated HTML look in JSF?
  - Can Servlet methods be invoked concurrently (from different threads at the same time)?
  - Which Servlet is dispatched if two Servlets match a URL?
  - ...
- Not as many differences as with HTML in browsers
- Code according to the specification

Overview

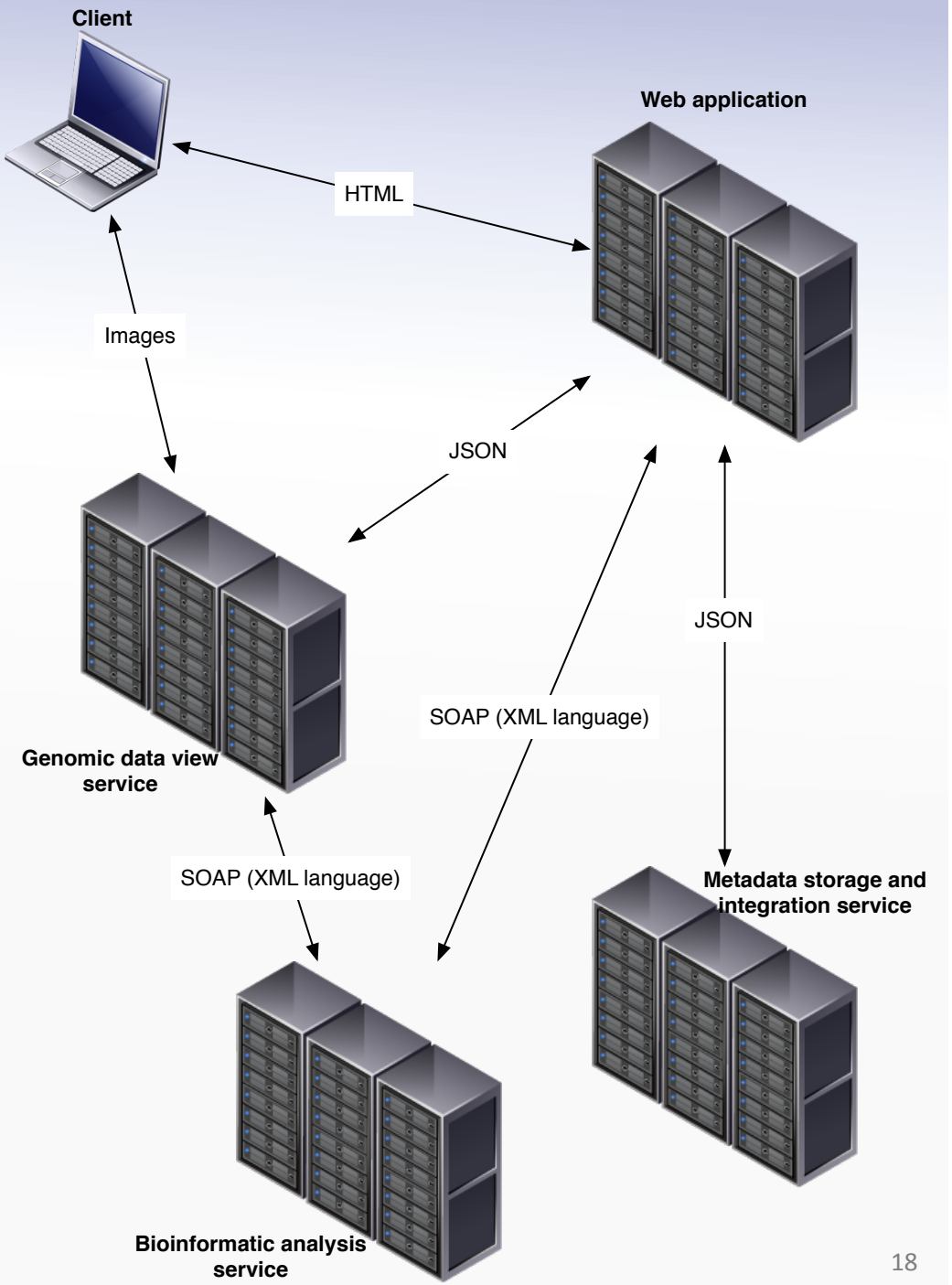
- Web application concepts
- Web application structure
- Specifications vs. implementation
- Case study

Case study

- DNA:
  - Our genes are encoded as DNA,
    - not an XML format ;-)
  - DNA is read in very small chunks by a *sequencing machine*
- Bioinformatics:
  - Analysis of very large DNA datasets (think 100GB or more)
    - Assemble small chunks to a full DNA string
    - Find variants (genetic differences in people, healthy/tumor samples)
    - …
- Genomics:
  - The study of DNA, genes, their functions, and relationship

Case study overview

- Web application:
  - MVC-like structure
  - JavaScript
  - Compiled Java
- Service orientation:
  - Separation
  - Simplification
- Replication:
  - Data viewing is heavy
    -> spin up another server



I think the biggest innovations of the 21st century will be at the intersection of biology and technology.
A new era is beginning.

Steve Jobs 2011

Changing perspective a bit now

- In the remainder of this course, we will:
  - Re-visit our concepts
    - DOM in JavaScript
    - Write our own web service
    - Structure our web service data
    - Structure our own application code
  - Discover the similarity to what we already know

- Re-use your knowledge when programming JavaScript!