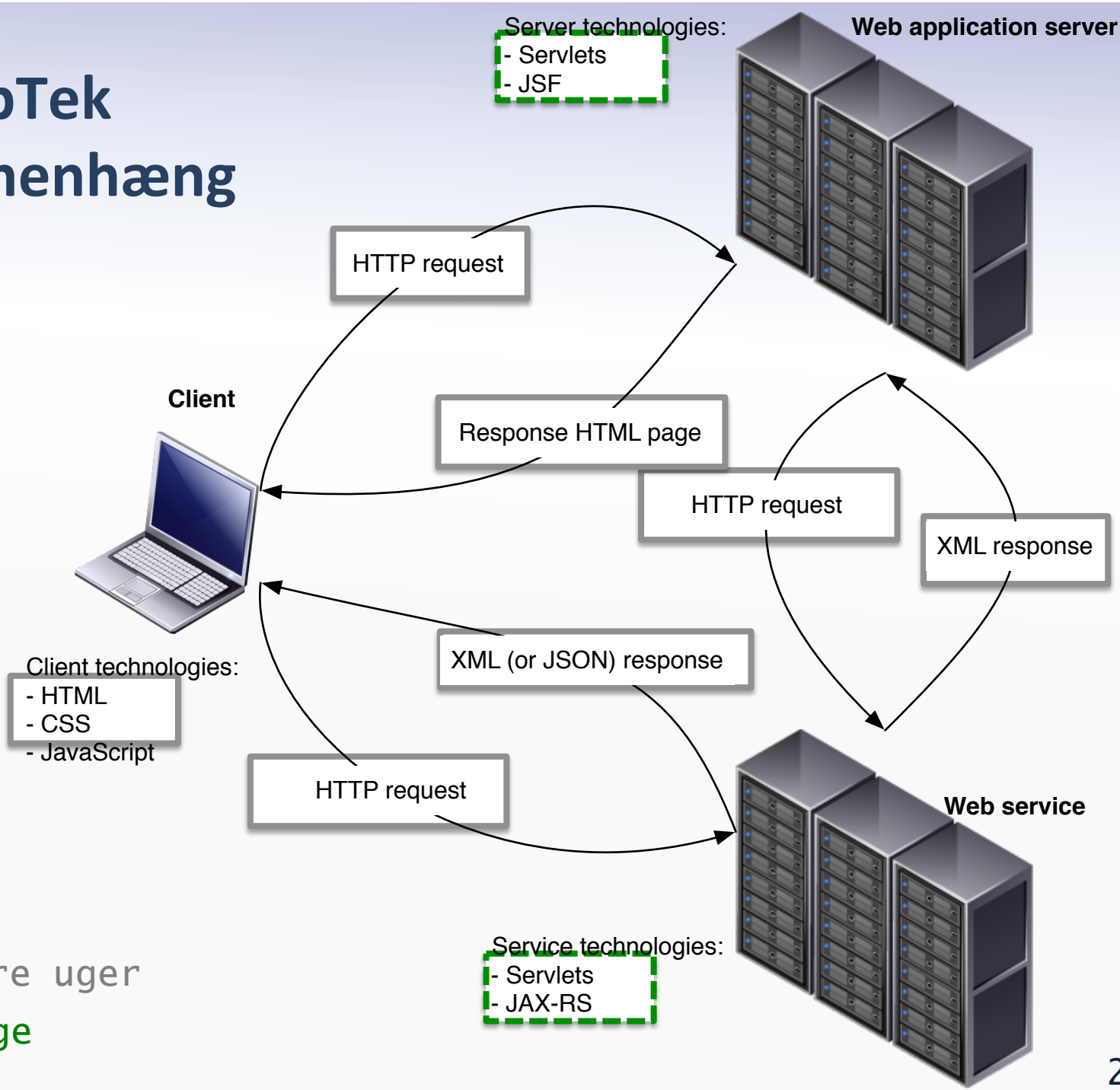
Concepts in web application development

dWebTek sammenhæng



Tidligere uger

Denne uge

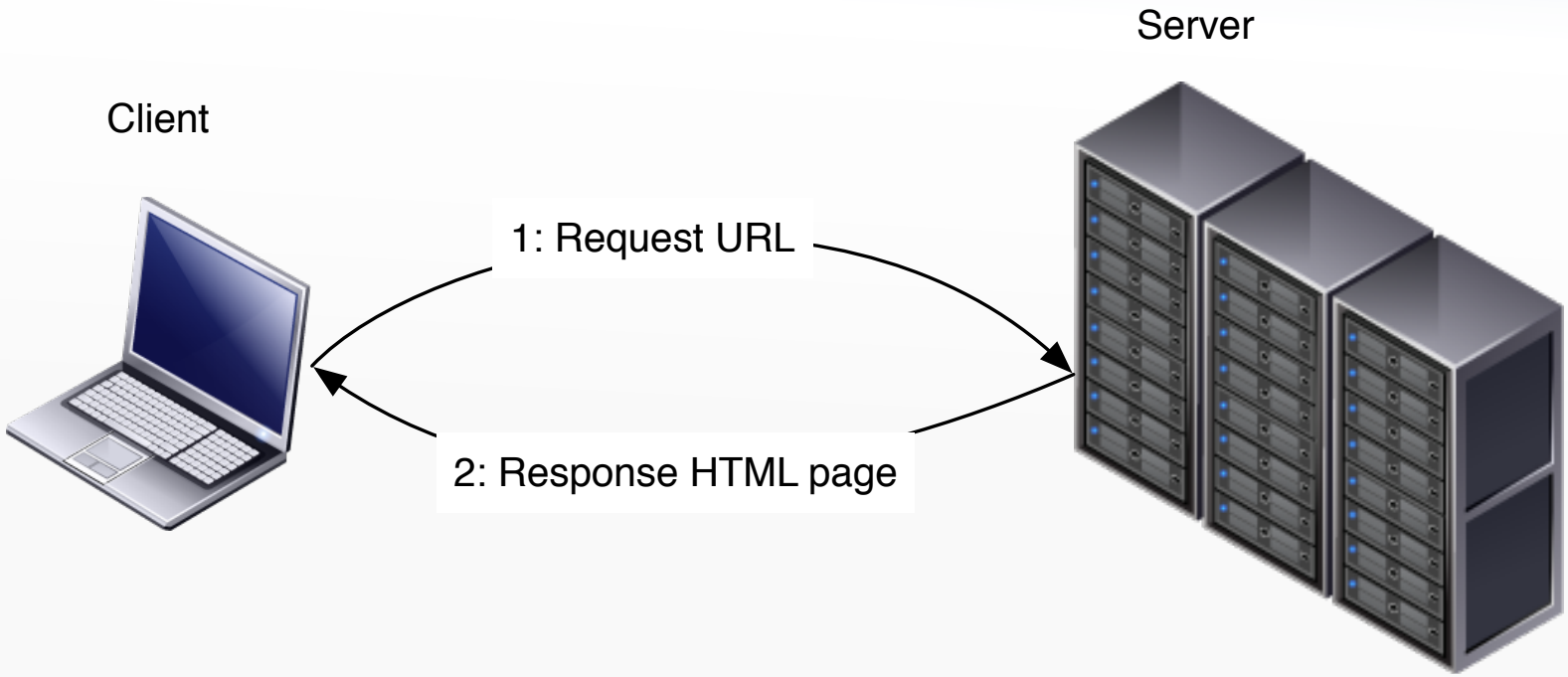
Overview

- Basic definitions
- Requests and responses
- State management
- Application structure
- Summary and goal

What is a web application?

- Any software that is used through a web browser
  - UI constructed using HTML and CSS
  - Communication over HTTP
- Contrast to Web services (later in this course):
  - Services expose XML data instead of UI
- Popular because:
  - No installation on user machines
  - Many potential users

Web application basics



Many ways of representing this in a programming language!

Client vs. server based

- Server-based web applications:
  - Generate HTML on the server
  - Run computations on the server
  - Input handled through web form submission and links
- Client-based web applications:
  - Generate HTML on the client side (typically using DOM)
  - Run computations in the browser
  - Communicate with server (XML, JSON) to store, read data while the user interacts with the page
- Hybrid approaches exist

Overview

- Basic definitions
- Requests and responses
- State management
- Application structure
- Summary and goal

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 (today) is an example of this
  - Values could be ***injected***:
    - Request parameter **foo** becomes value **foo**
      - Field, method parameter, etc..
    - JSF (next week) and JAX RS (week 5) do this
  - ...

Concept: Response handling

- HTTP response representation in the host language
- Design space:
  - Output could be written directly to a stream,
    - Servlets (today)
  - Be generated through a template system, or
    - JSF (next week)
  - Be represented as response objects
    - JAX RS (week 5)

Concept: Reponse templates

- Template example (from JSF):
  - A representation of the response document:
    - Values are inserted from (Java) objects
    - XML! ☺

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html">
  <head><title>Response</title></head>
  <body>
    <h:form>
      <h2>Hello, #{helloBean.name}</h2>
      <h:commandButton value="Back" action="welcome" />
    </h:form>
  </body>
</html>
```

Overview

- Basic definitions
- Requests and responses
- State management
- Application structure
- Summary and goal

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
- Not all web application frameworks expose all scopes:
  - PHP needs a database for application state
- Design space examples:
  - Untyped Java map (Servlets)
  - Injected Java beans (JSF)

More about session state

- Session track data related to a single client
  - Save data while handling one request, and
  - Read it while handling another request from **the same client**
  - Examples: Login information, shopping baskets...
- The web framework:
  - Track clients using cookies, URL rewriting, etc. transparently
  - Makes the data available to the programmer

- Design space examples:
  - Session object that holds an untyped map (Servlets)
  - Injected Java beans (JSF)

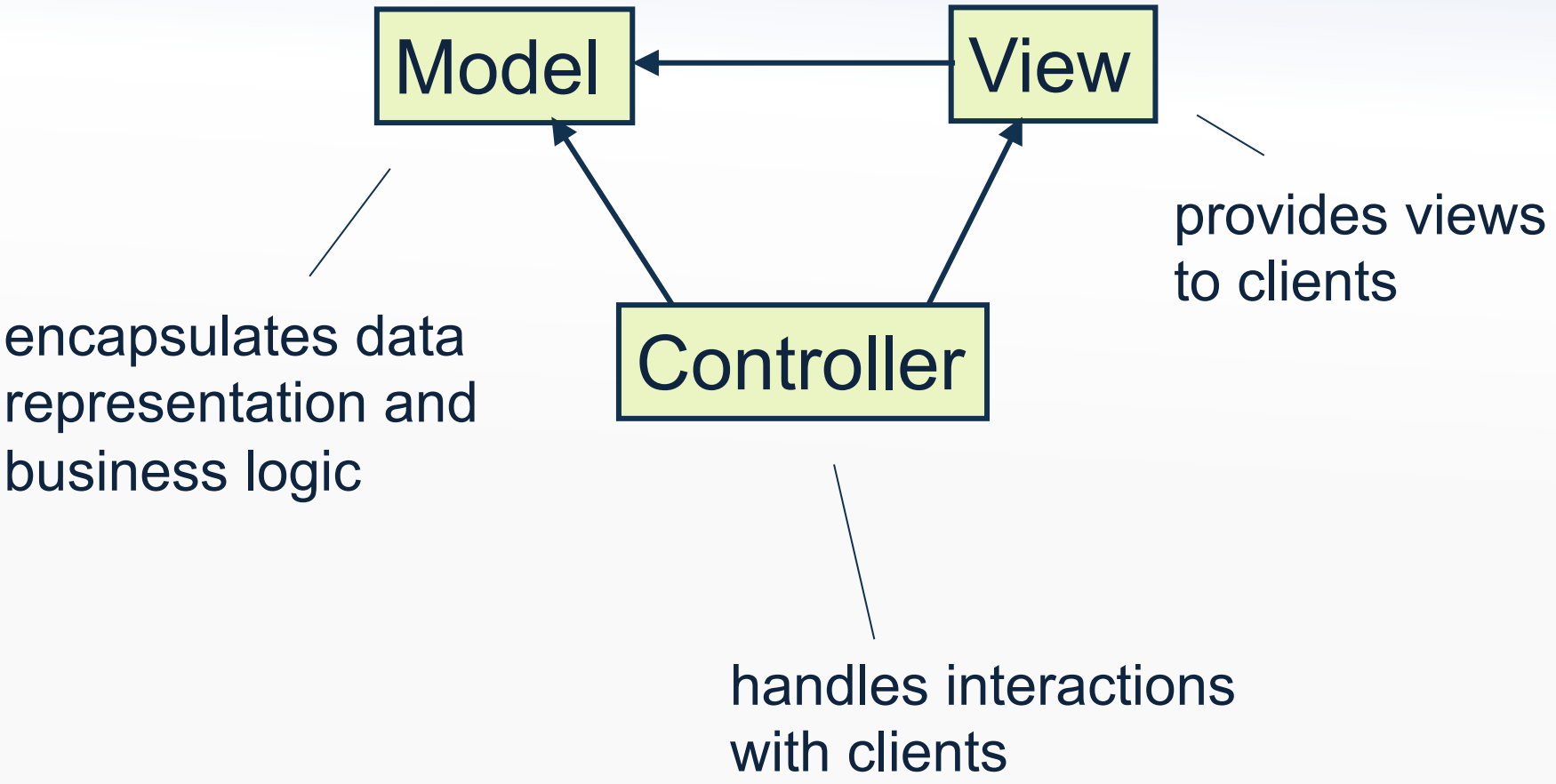
Overview

- Basic definitions
- Requests and responses
- State management
- Application structure
- Summary and goal

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:
  - 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-Ciew-Controller

- MVC dominates the web framework landscape:
  - ASP.NET MCV (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...

- To learn how a new MVC framework works:
  - 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

- Goal of the coming weeks:
  - Cover enough concepts and design possibilities to make you able to understand a new framework when you see it!