

Web application development

Software engineering perspective

Kari Systä
28.4.2014

Week	Lecture	Exercise
10.3	Quality in general; Quality management systems	Patterns
17.3	Dependable and safety-critical systems	ISO9001
24.3	Work planning; effort estimation	Code inspections
31.3	Version and configuration management	<i>Effort estimation</i>
7.4	Role of software architecture software evolution	
14.4	Software business, software start-ups, IPR	Break
21.4	Easter	Break
28.4	Web application development; SW engineering viewpoint	
5.5	Last lecture; summary; recap for exam	
15.5	EXAM 9-12	

End seminar

- (2nd) traditional end seminar 16.5. at 1300 in SE203 (initial info)
- Participants collect one "point"
- Best project will win a prize!
- Each presentation
 - 2.5 minutes demo of the game (or video)
 - 2.5 minutes presentation "lessons learned from Software Engineering and process point of view"
- If you suspect that your computer does not cooperate with video projector, bring a memory stick

Jobs

- Our department looks for thesis workers who
 - Are good programmers
 - Can act as teachers (fluent English and/or Finnish)
 - True ambition to post grad studies is a plus
 - Currently seeking: Web application programming; cloud systems; Continuous Integration; GUI toolkits
- In case you are interested, drop me an email
- Companies are constantly seeking. See posters on our wing.
- And next slide

www.swd.fi

Odotamme

- Ohjelmointitaitoa
- Valmiutta toimia asiakaskontakteissa
- Kykyä tiimityöskentelyyn
- Hyvää englannin ja suomen kielen taitoa
- Matkustusvalmiutta
- Kiinnostusta uusien taitojen oppimiseen sekä oman osaamisen jatkuvaan kehittämiseen
- Soveltuvia opintoja

Eduksi luemme

- Tuotantotalouden ja/tai tuotantotekniikan ymmärrystä
- Simulointi- ja optimointiosaamista
- Logistiikkaprosessien hallintaa
- Tietokanta- ja SQL-taitoja
- Myyntihenkisyyttä
- Ruotsin kielen taitoa
- Kokemusta ERP- ja tuotannonsuunnittelujärjestelmistä
- Scrum-menetelmien tuntemusta

Hae tehtävää lähettämällä hakemus ja ansioluettelo osoitteeseen: eeva.koskinen@swd.fi. Työt alkavat sopimuksen mukaan.

Lisätietoja: Eeva Koskinen puh. 050 406 9894

Translation

(because this is educational)

We require

- Programming skills
- Ability to work in customer contacts
- Team-work skills
- Good English and Finnish language skills
- Ability to travel
- Interest to continuous learning
- Suitable studies

These are also counted

- Understanding of production technology and business
- Simulation and optimization skills
- Management of logistics processes
- Database and SQL skills
- Sales attitude
- Swedish skills
- Experience with ERP systems
- Scrum knowledge

Feedback requested

- As normally, you should give feedback through the "Kaiku" system
- Since this is a renewed course we are especially interested in feedback. This feedback will have influence to next implementation. We have a few specific questions:
 - Language. This year we only lectured in English. If were sources next spring should it be used for separate implementation.
 - Overlapping. We have intentional and unintentional overlapping with some other courses. Do we have too much or too little overlapping?
 - Content. Which topics should be reduced; what topics should be added?
 - More hands-on work vs. more "theoretical" concepts
 -
- You can use Kaiku to give this, or you can send by email. (but preferably in writing so that it is not forgotten)
- The course staff will have a retrospective session in June. Interested students may receive an invitation...

TIE-23500 Web-ohjelmointi, kevät 2014

- This course is about practical programming
- And only in Finnish
- Today we will talk about Web applications
 - From software engineering point of view
 - In English

Content

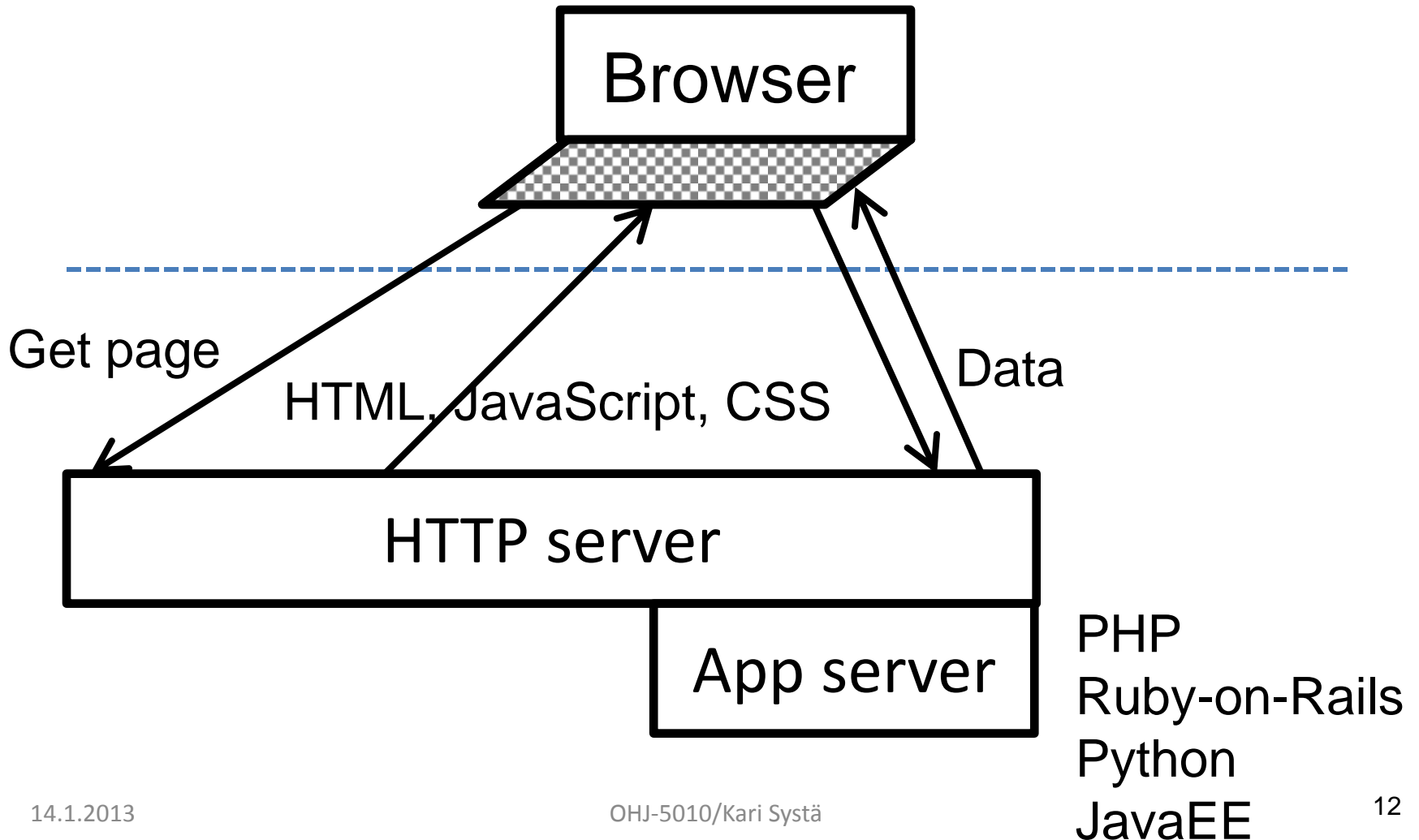
- Basic framework of all Web applications
- Short history of Web Apps
 - Static web, animations, Ajax, RIA, HTML5, one-page apps, asm.js
- Major architecture choices
- SW engineering challenges
- JavaScript as a programming language
- Service-oriented systems and mash-ups
 - SOAP, REST
 - ATOM, RSS, ...
- TUT Research

Basic framework of all Web applications

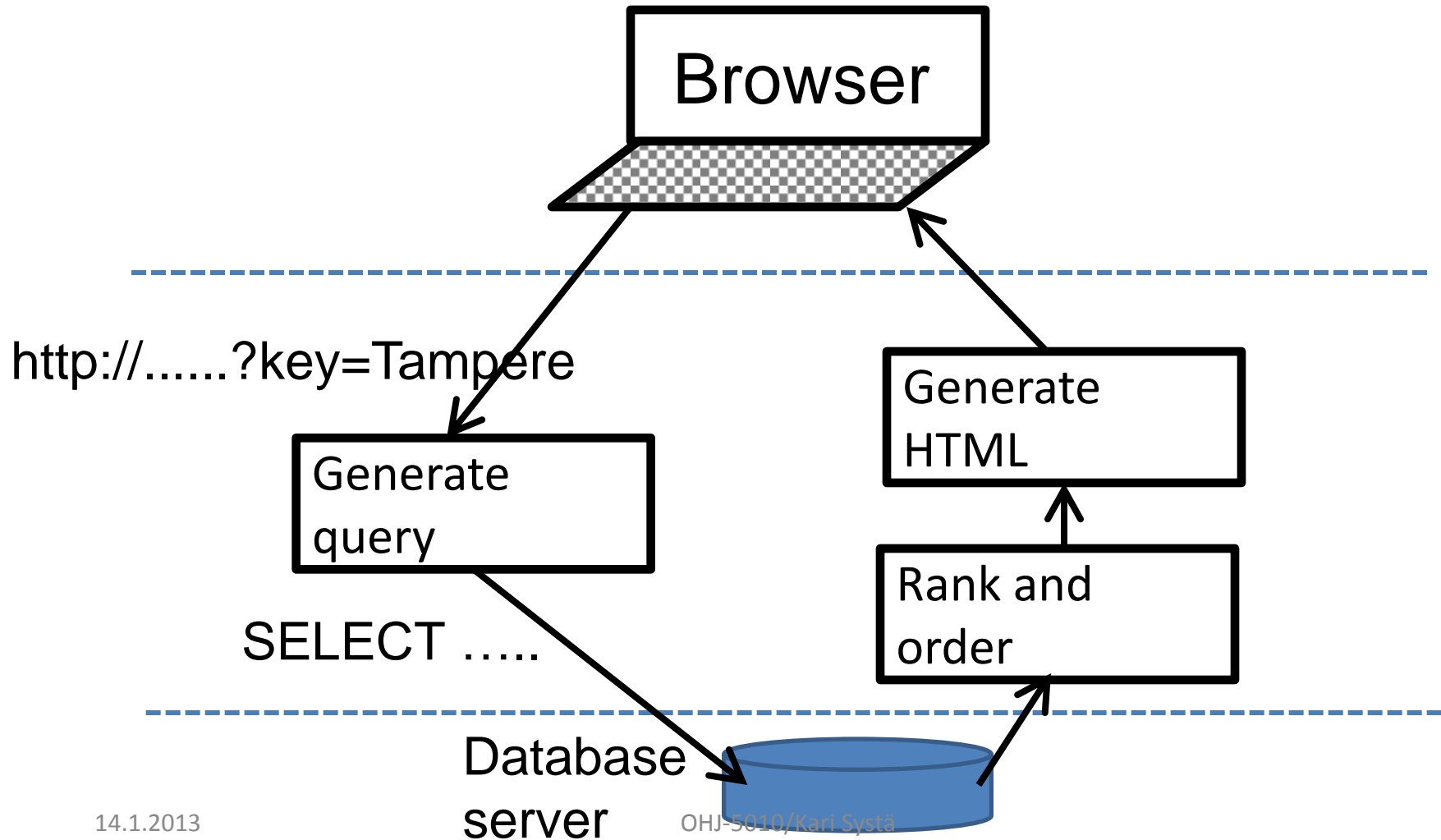
The concept of "computer" has changed!



Simple architecture



Typical 3-tier architecture



Cookie

(source wikipedia)

- A **cookie**, is a small piece of data sent from a website and stored in a user's [web browser](#) . Every time the user loads the website, the browser sends the cookie back to the server to notify the website of the user's previous activity.¹
- Cookies were designed to be a reliable mechanism for websites to remember [stateful](#) information (such as items in a shopping cart) or to record the user's browsing activity (including clicking particular buttons, [logging in](#), or recording which pages were visited by the user as far back as months or years ago).

Three components

Content

```
<html>
<script src="example.js">
</script>
<link href="example.css" type="text/css" rel="stylesheet">
<body onload="showDate()">
<P>
Today is: <span class="dateString" id="today">...</span>
</body>
</html>
```

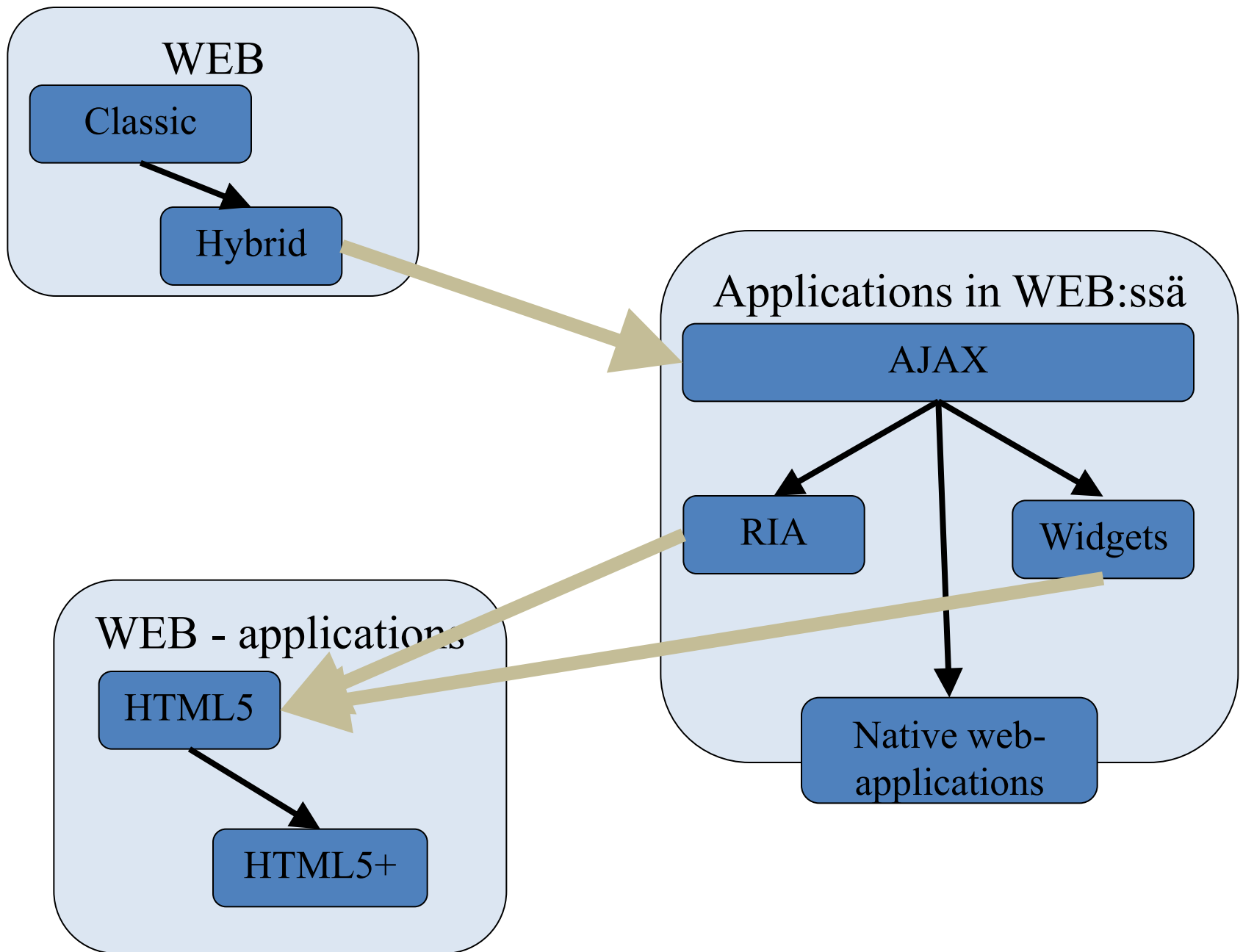
Style

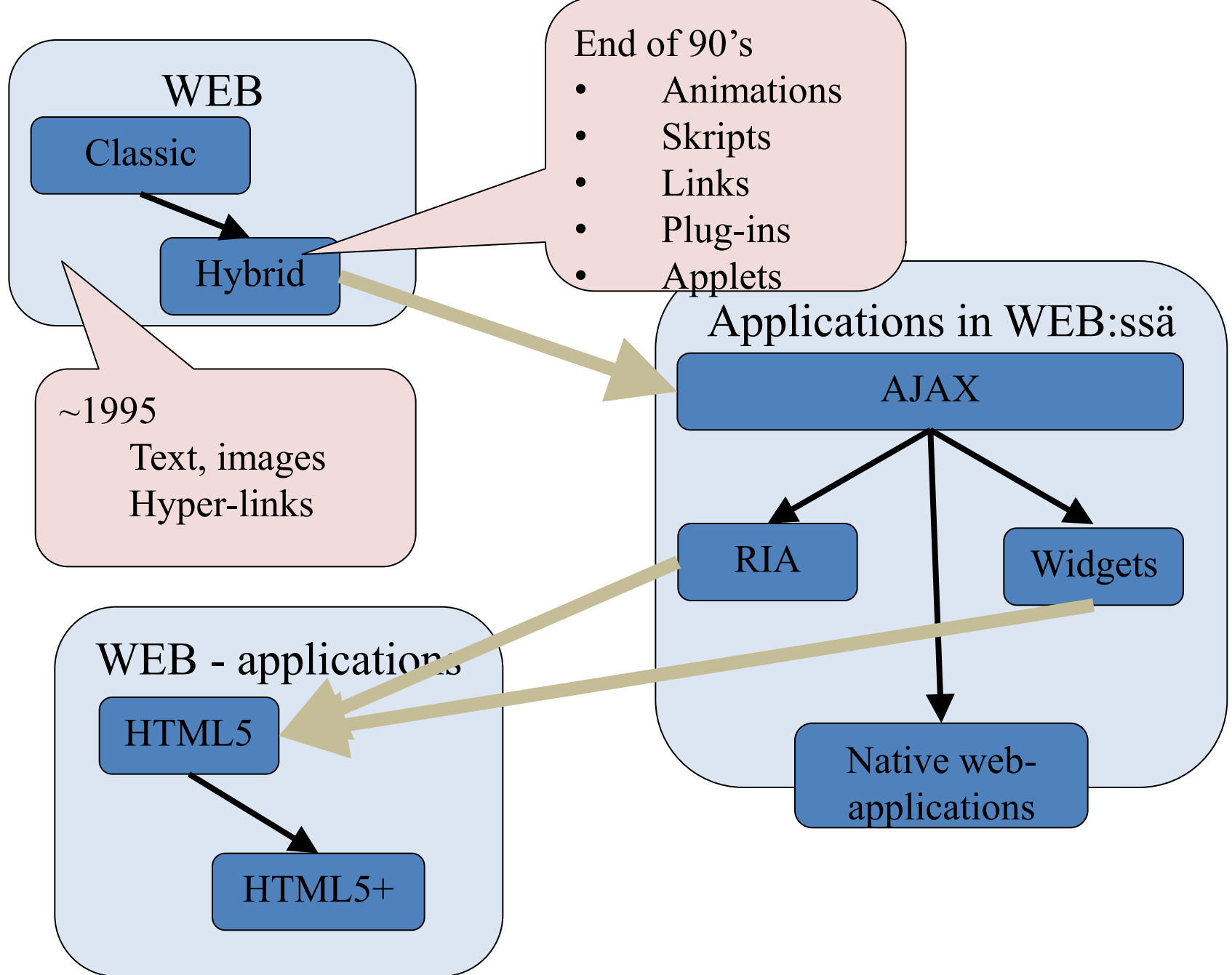
```
.dateString {
  font-family:"Times New Roman";
  font-size: 20px;
}
```

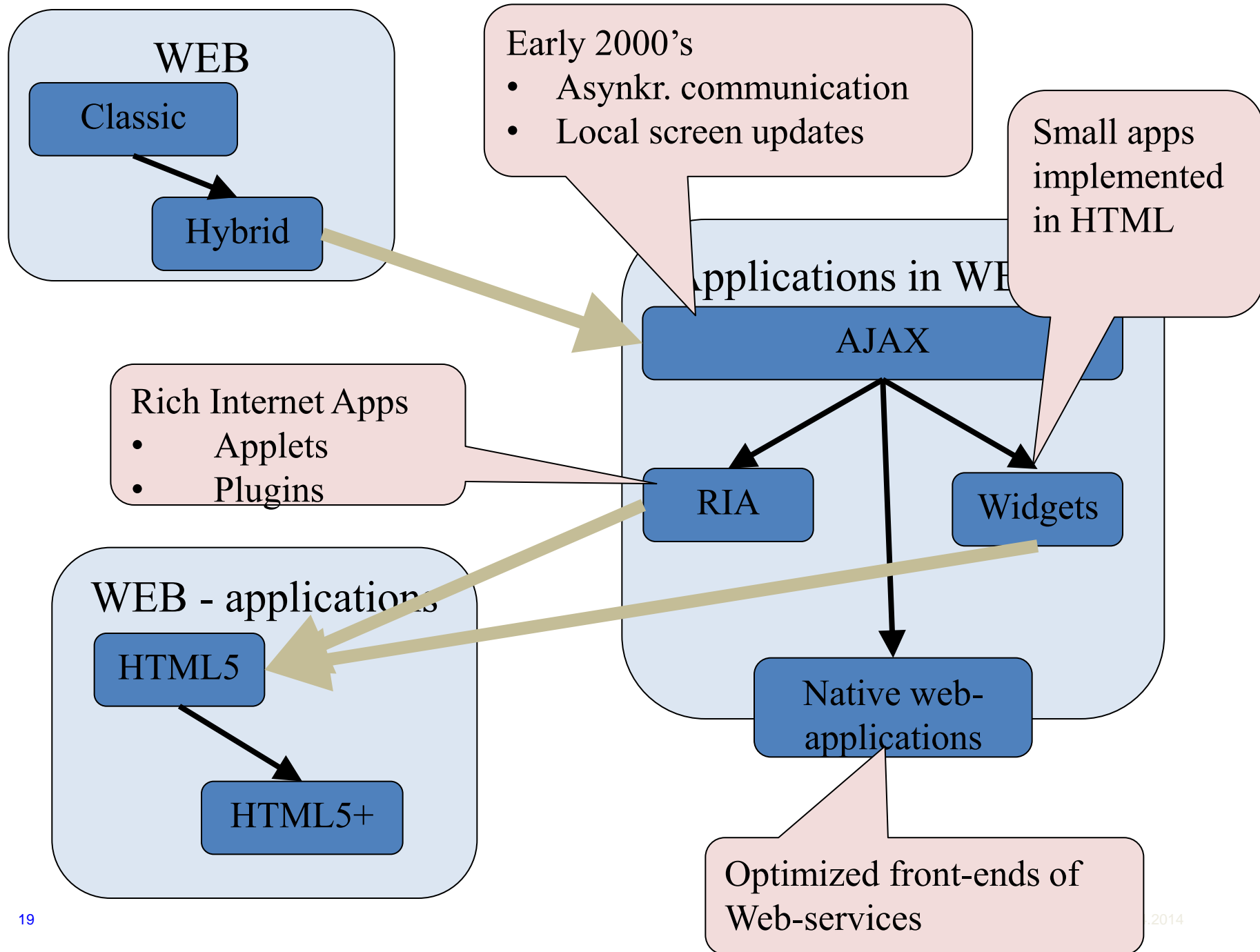
Functionality

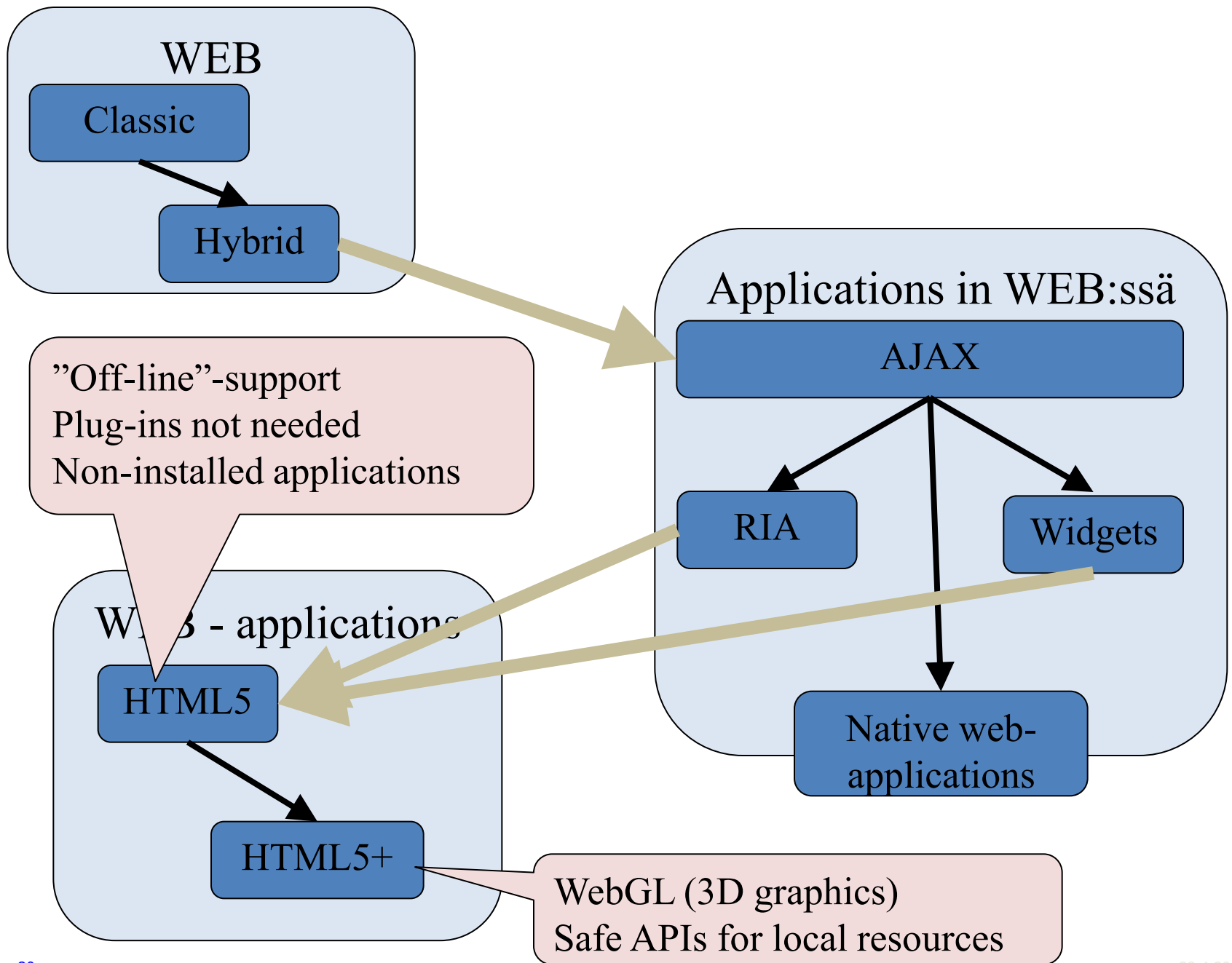
```
function showDate() {
  var x = document.getElementById("today");
  if (x !== null) {
    x.innerHTML= "" + new Date();
  }
}
```

Short history of Web Apps



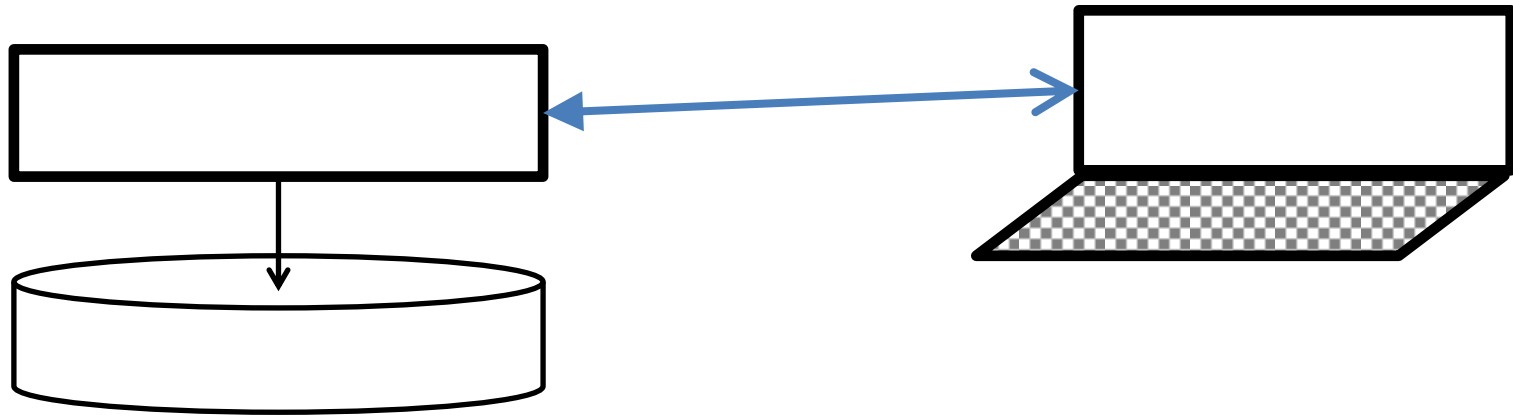






Major architecture choices

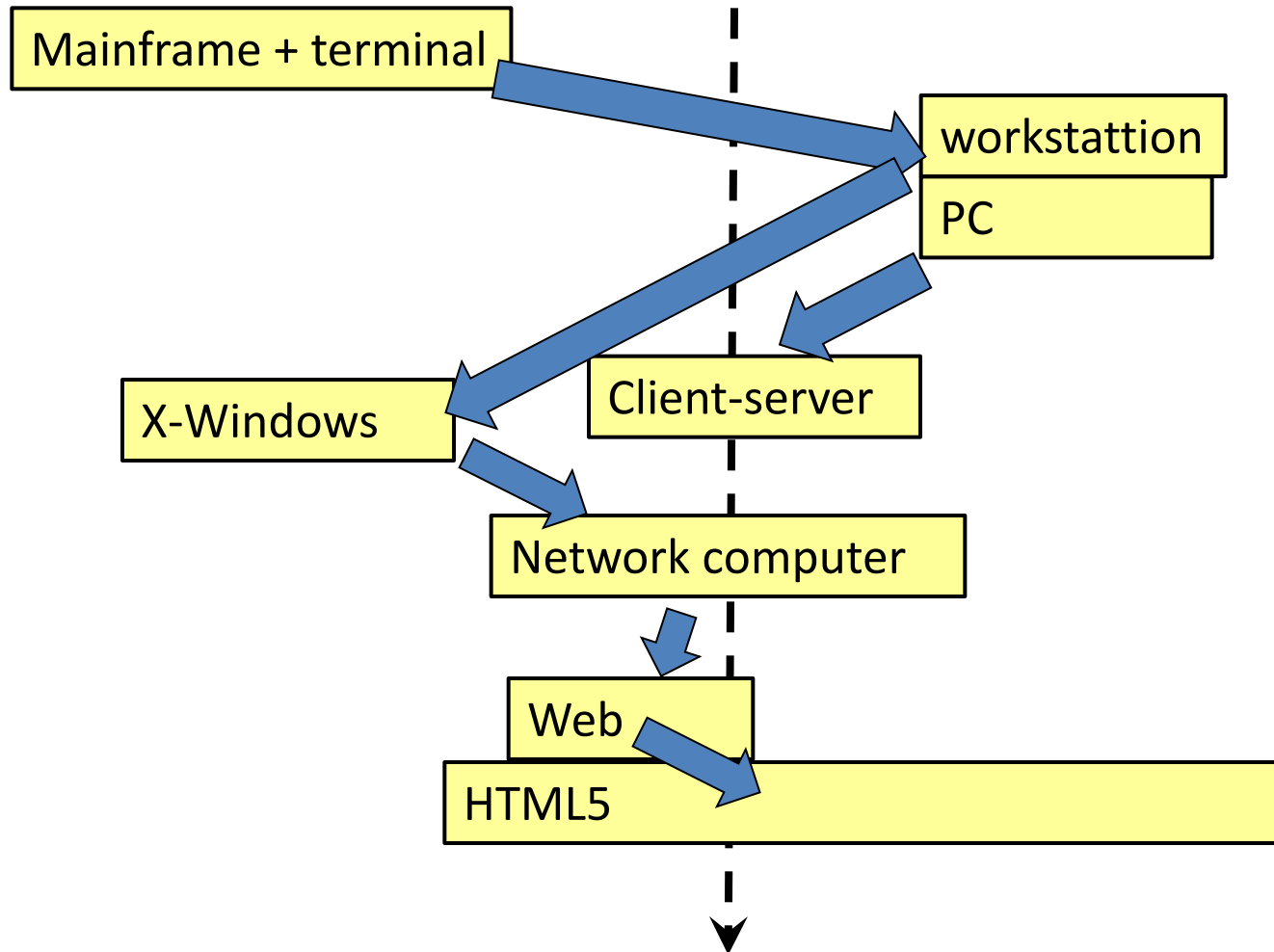
Thin vs thick (fat) client



Simple maintenance
Cheap client hardware
Independence of client device
BYOD

Lower server requirements/
better performance
Working offline
Better multimedia perform.
More flexibility
Using existing infrastructure

Ohut vai paksu asiakas (rankasti yksinkertaistaen)

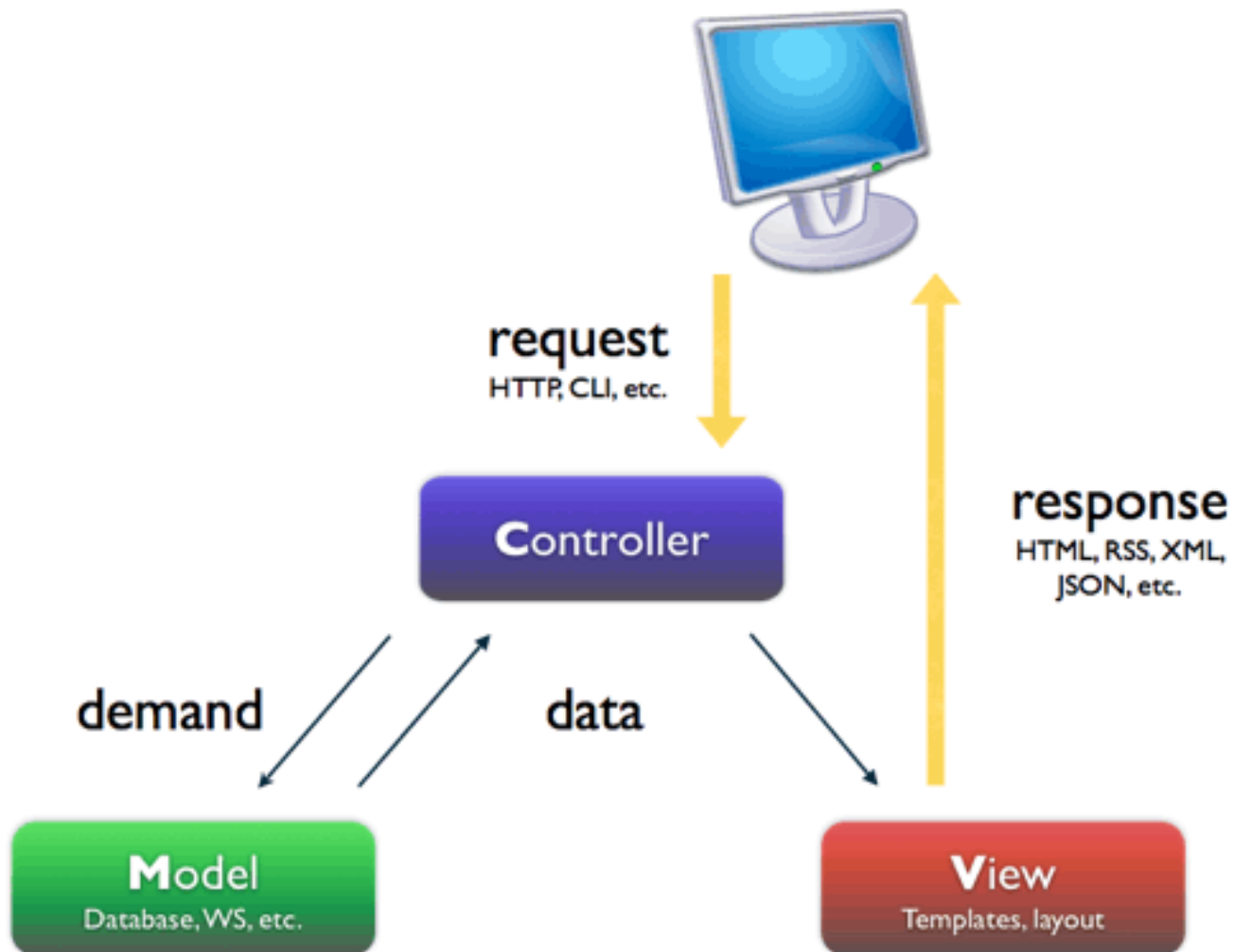


Browser vs. Server programming

- Web system have code in both server and client
- Programming for client and server could be separate – or programming could be either server or client centric
- Examples server centric
 - Ruby-on-Rails
 - Google Web toolkit & Vaadin
- Example of client centric
 - BaaS (back-end as a service)

Ruby-on-Rails

(<http://bpesquet.developpez.com/tutoriels/php/evoluer-architecture-mvc/>)



Google Web Toolkit & Vaadin

- The idea: the UI is written as server-side Java code and HTML+CSS+JavaScript is autogenerated

```

package com.vaadin.demo;

import com.vaadin.ui.Label;
import com.vaadin.ui.Window;

@SuppressWarnings("serial")
public class HelloWorld extends com.vaadin.Application {

    /**
     * when a user accesses the application for the first time).
     */
    @Override
    public void init() {

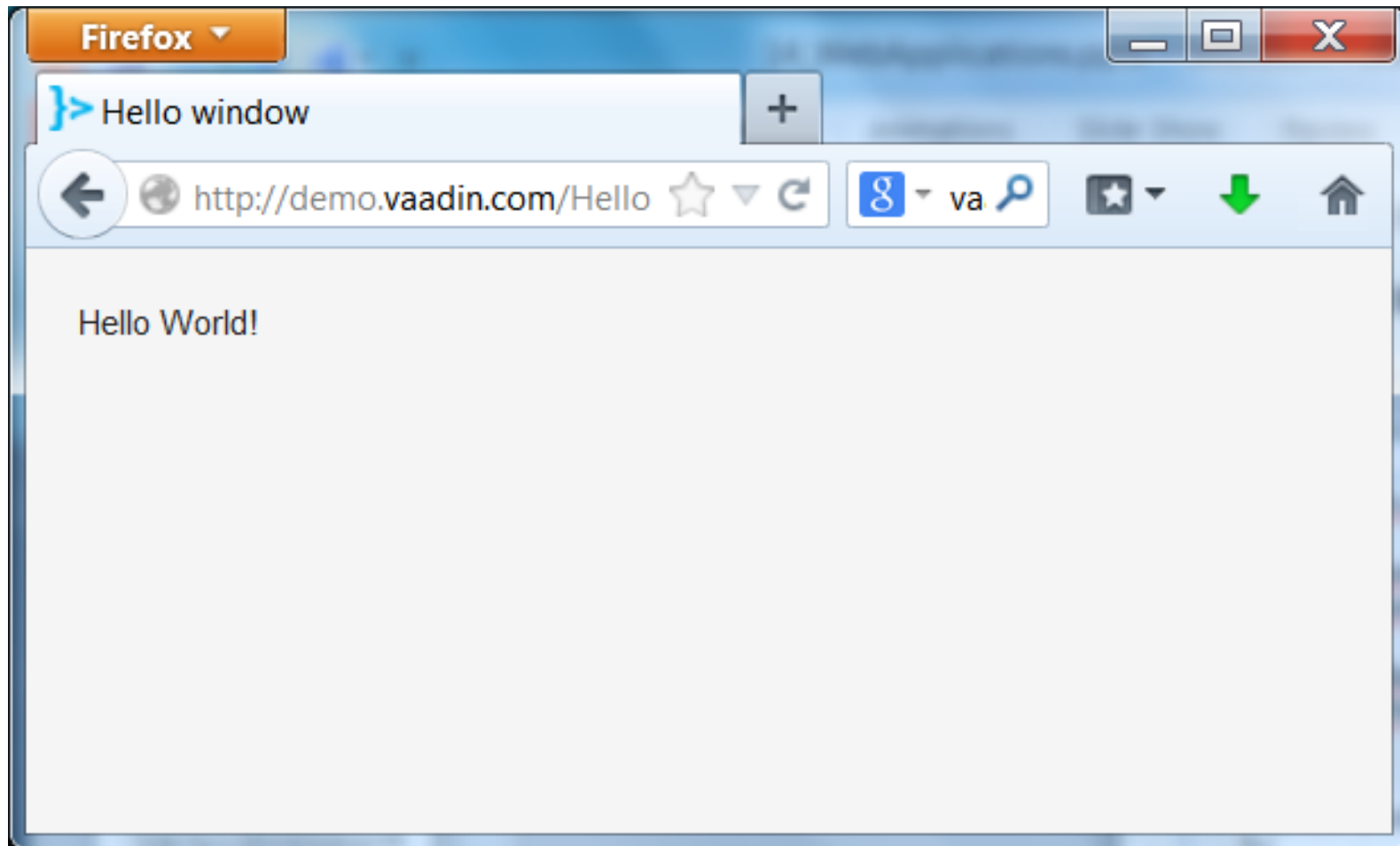
        // Main window is the primary browser window
        final Window main = new Window("Hello window");
        setMainWindow(main);

        // "Hello world" text is added to window as a Label component
        main.addComponent(new Label("Hello World!"));
    }

}

```

Result



File Edit View Help

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/1
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
5 <style type="text/css">html, body {height:100%;margin:0;}</style><link rel="shortcu
6 </head>
7 <body scroll="auto" class="v-generated-body">
8 <script type="text/javascript">
9 //<![CDATA[
10 if(!vaadin || !vaadin.vaadinConfigurations) {
11   if(!vaadin) { var vaadin = {} }
12   vaadin.vaadinConfigurations = {};
13   if (!vaadin.themesLoaded) { vaadin.themesLoaded = {}; }
14   vaadin.debug = true;
15 }
16 vaadin.vaadinConfigurations["HelloWorld-439329280"] = {appUri:'/HelloWorld', standa
17 //]]>
18 </script>
19 <iframe tabIndex='-1' id='__gwt_historyFrame' style='position:absolute;width:0;heig
20 <script language='javascript' src='/VAADIN/widgetsets/com.vaadin.terminal.gwt.Default
21 <script type="text/javascript">
22 //<![CDATA[
23 if(!vaadin.themesLoaded['reindeer']) {
24   var stylesheet = document.createElement('link');
25   stylesheet.setAttribute('rel', 'stylesheet');
26   stylesheet.setAttribute('type', 'text/css');
27   stylesheet.setAttribute('href', '/VAADIN/themes/reindeer/styles.css');
28   document.getElementsByTagName('head')[0].appendChild(stylesheet);
29   vaadin.themesLoaded['reindeer'] = true;
30 }
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>...</head>
  <body scroll="auto" class="v-generated-body v-sa v-ch v-webkit v-win">
    <script type="text/javascript">...</script>
    <iframe tabindex="-1" id="__gwt_historyFrame" style="position:absolute;
width:0;height:0;border:0;overflow:hidden;" src="javascript:false">
...</iframe>
    <script language="javascript" src="/VAADIN/widgetsets/
com.vaadin.terminal.gwt.DefaultWidgetSet/
com.vaadin.terminal.gwt.DefaultWidgetSet.nocache.js?
1398587429061"></script>
    <script defer="defer">...</script>
    <script type="text/javascript">...</script>
    <script type="text/javascript">...</script>
    <div id="HelloWorld-439329280" class="v-app v-theme-reindeer v-app-
HelloWorld">
      <div class="v-view v-scrollable" tabindex="1">
        <div class="v-loading-indicator" style="position: absolute; display:
none;"></div>
        <div class="v-verticallayout" style="overflow: hidden; width: 806px;
height: 54px;">
          <div style="overflow: hidden; margin: 18px; width: 770px; height:
18px;">
            <div style="height: 18px; width: 770px; overflow: hidden; padding-
left: 0px; padding-top: 0px;">
              <div style="float: left; margin-left: 0px;">
                <div class="v-label" style="width: 770px;">Hello World!</div>
              </div>
            </div>
          <div style="width: 0px; height: 0px; clear: both; overflow:

```

Styles	Computed
element.style {	
.v-scrollable {	overflow: auto
.v-view {	height: 100%; width: 100%; outline: none; margin-top: -1px; border-top: 1px solid transparent; position: relative
div {	user agent; display: block
Inherited from div#H	
.v-app, .v-window, .v-popup, .v-tooltip, .v-app select, .v-app textarea, input, .v-window window button, .v-textarea, .v-popup input, .v-popup select, .v-popup button, .v-popup textarea, .v-file	

Idea of BaaS

- **Backend as a Service** *a model for providing web and mobile app developers with a way to link their applications to hosted backend cloud storage while also providing features such as user management, push notifications, and integration with social networking services, all as an integrated offering.*
- Key goal is to **abstract away all the complexity related to cloud and cloud management** and provide simple APIs that can be used across all popular mobile platforms.
 - “Don’t worry about the server side, we’ll take care of it for you.”

Usage Example – Parse Data API

https://parse.com/docs/js_guide

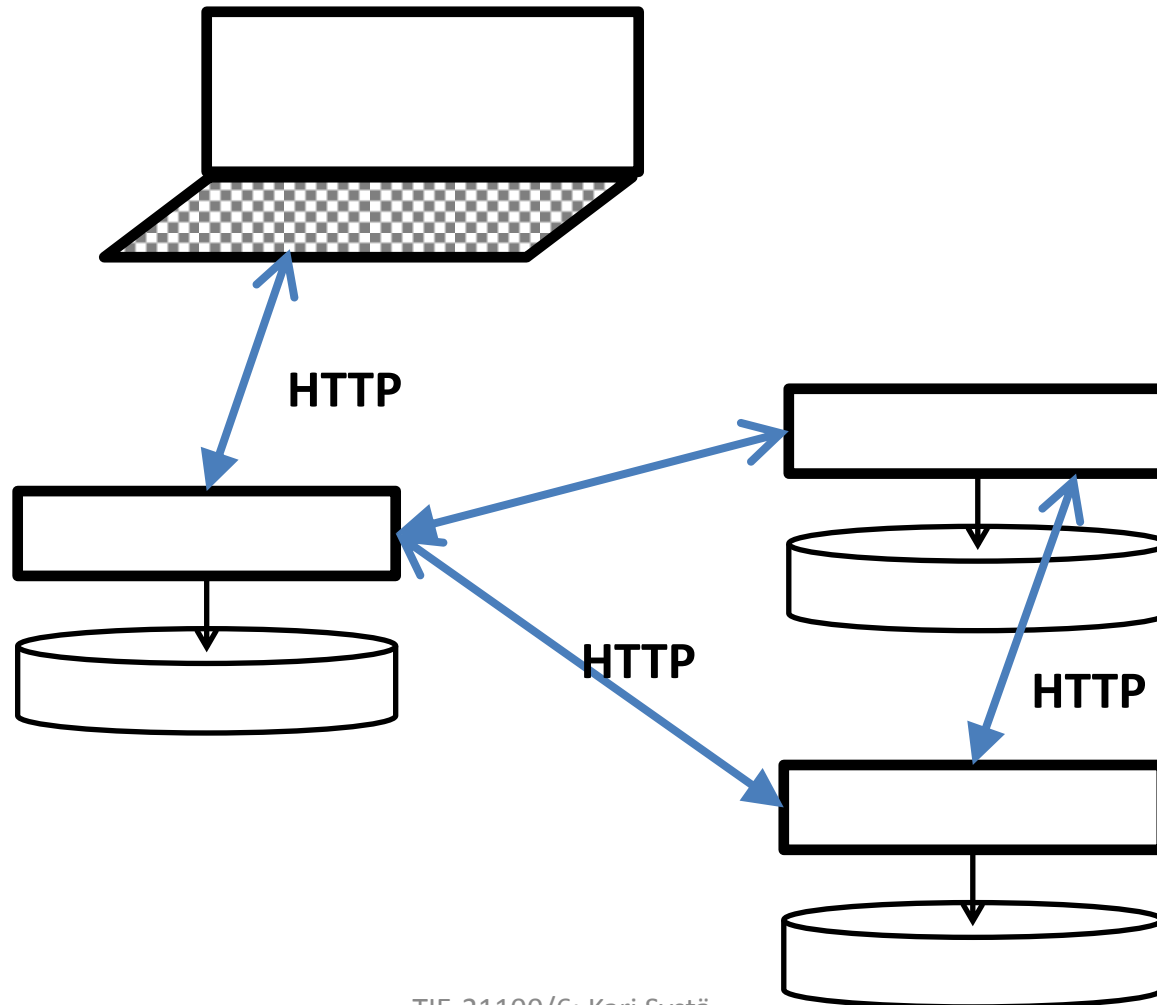
```
var GameScore = Parse.Object.extend("GameScore");
var gameScore = new GameScore();

gameScore.save({
  playerName: 'New Player',
  score: 1000, cheatMode: false
}, {
  success: function(gameScore) {
    // Execute any logic that should take place after the object is saved.
  },
  error: function(gameScore, error) {
    // Execute any logic that should take place if the save fails.
    alert('Failed to create new object, error code: ' + error.description);
  }
});
```


Single page vs. chain of pages

- Traditionally the UI logic has been based on moving from page to another.
 - “Back” of browser interferes with the app
- Single page app has only one “page load”
- Additional resources may be loaded later
- These apps look like any app

REST & SOAP



SOAP & REST

- SOAP
 - Remote procedure call over HTTP
- REST
 - **Client-Server** The primary principle behind client-server pattern is 'separation of concerns', i.e. client and server are separated.
 - **Stateless** communication must be stateless in nature
 - **Cacheable** to improve the network efficiency
 - **Uniform Interface**
 - URL
 - CRUD (Create, read, update and delete)

Service-oriented systems and mash-ups

- Google Mashup
(<http://code.google.com/gme>)
- IBM Mashup Center
(<http://www.ibm.com/software/info/mashup-center/>)
- Intel Mash Maker
(<http://mashmaker.intel.com/>)
- Microsoft Popfly (<http://popfly.com/>)
- Yahoo! Pipes (<http://pipes.yahoo.com/>)

JavaScript as a programming language

JavaScript

- Developed by Brendan Eich at Netscape ~1995
 - Fast design for simple purpose to add dynamic behavior to web pages
- Nice language for simple scripting
- Can be used for almost everything
- But is not optimum for large scale SW development
- Very popular nowadays
 - For example see <http://langpop.com/>

Unearthing the excellence in JavaScript



JavaScript: The Good Parts

O'REILLY®

YAHOO! PRESS

Douglas Crockford

SW Engineering and Web development

Current state

Web	Conventional
Documents	Code
Pages / forms	Direct manipulation
Managed graphics, static layout	Directly drawn, dynamic
Instant world-wide deployment	Conventional deployment
Source code and text	Binaries
Conventions and folklore	Software engineering principles
Target env not designed for apps	Target env intended for apps

Challenges of Web SW Engineering

- software engineering principle violations,
- usability and user interaction issues,
- networking and security issues,
- browser inoperability and incompatibility issues,
- development style and testing issues,
- deployment model changes, and
- performance issues.

software engineering principle violations

1) Modularity and related principles

- Declarative and procedural development style are mixed up
- User interface component placement, user interface style elements, event declarations and application logic are mixed up (latest toolkits fix a bit)
- Well-defined (manifest) interfaces mainly missing (except DOM)
- Information hiding problems, like DOM tree is exposed and manipulated through side effects
- No privacy mechanisms available in JavaScript

2) Consistency, simplicity and elegance

- There are several ways to perform the same function
- Things should happen explicitly rather than through side effects
- Current web applications are unstructured and hard to read

3) Reusability and portability

- Elements of reuse are scattered and mixed with the rest of the application
- Hard-coded references and other implementation details are exposed

Three components

Content

```
<html>
<script src="example.js">
</script>
<link href="example.css" type="text/css" rel="stylesheet">
<body onload="showDate()">
<P>
Today is: <span class="dateString" id="today">...</span>
</body>
</html>
```

Style

```
.dateString {
  font-family:"Times New Roman";
  font-size: 20px;
}
```

Functionality

```
function showDate() {
  var x = document.getElementById("today");
  if (x !== null) {
    x.innerHTML= "" + new Date();
  }
}
```

Usability and user interaction issues

- The browser I/O model is poorly suited to desktop-style applications
 - XMLHttpRequest and WebSockets improve this
- The semantics of many browser features are unsuitable for applications
 - Single Page Apps

Networking and security issues,

- In Web content is assumed to come anywhere
- The JavaScript run-time has no security mechanisms
- Cross-site scripting is forbidden by default

browser inoperability and incompatibility issues

- All components
 - HTML rendering
 - JavaScript engine
 - CSS styling

have some differences

- One reason to use toolkits like JQuery or Prototype.js is to overcome these differences

Development style and testing issues

- Pretty much "try-and-error" and "follow the cow-path"
- Even cargo-cult programming
- Several sources of error
- Debuggers usually not available

Sources

- <http://lively.cs.tut.fi/publications.html>

Nice videos (semi serious)

<https://www.destroyallsoftware.com/talks/the-birth-and-death-of-javascript>

Brendan Eich's Fluent 2014 talk on JavaScript:
http://www.youtube.com/watch?v=aZqhRICne_M

Scott Hanselman's Fluent 2014 talk on "Virtual Machines, Javascript and Assembler":
<https://www.youtube.com/watch?v=UzyoT4DziQ4>

Week	Lecture	Exercise
10.3	Quality in general; Quality management systems	Patterns
17.3	Dependable and safety-critical systems	ISO9001
24.3	Work planning; effort estimation	Code inspections
31.3	Version and configuration management	<i>Effort estimation</i>
7.4	Role of software architecture software evolution	
14.4	Software business, software start-ups, IPR	Break
21.4	Easter	Break
28.4	Web application development; SW engineering viewpoint	
5.5	Last lecture; summary; recap for exam	
15.5	EXAM 9-12	