Introduction

GÖTEBORGS UNIVERSITET

DAT076/DIT126 Web Applications

Robin Adams

Who we are

Course responsible and Examiner

Robin Adams <robinad@chalmers.se>

Teaching assistants

Patrick Franz <gusfranpa@student.gu.se>

Matthías Páll Gissurarson <pallm@chalmers.se>

Guest lecturer

Benjamin Eriksson <beneri@chalmers.se>

How to teach Web Applications?

- Lots of frameworks out there
  - J2EE, NodeJS, Ruby on Rails, ASP.NET, …
- Ways to structure a course like this:
  - Brief introduction to some (all?) of them
  - Choose one and teach its model and best practices
- This course concentrates on Jakarta EE (a.k.a. Java EE, J2EE, Java Enterprise)
  - Widely used, especially in enterprise apps (e.g. PayPal)
  - No need to learn a new language
  - Robust, well-designed, encourages good software design practices
  - You are learning **how to learn a framework**

Goal of the course

To learn and understand the basics of scalable enterprise-grade web application development

- Focus on the JavaEE / JakartaEE technology stack on the backend
- Focus on Java Server Faces for the frontend with some additional coverage of React. Although Java Server Faces is preferred, you have a free choice of frontend technology in the project.
- Learn deployment, building and debugging on an enterprise-grade application server with a modern development environment

Course philosophy

You will learn a large number of technologies

- HTML and CSS
- Java Server Faces
- Dependency Injection
- Web services
- JavaScript and AJAX
- Persistence
- Java Server Faces and React
- Enterprise Java Beans

- The lectures and labs introduce you to the basics of many technologies
- You choose which ones you want to focus on in your project
- We encourage you to read and experiment

Changes from last year

- Course is entirely distance learning
  - If guidelines change:
    - We may move to a hybrid model
    - It will be possible to follow the course 100% distance learning
  - Supervision and group project meetings given by Zoom
  - Slack channel for course announcements, Q&A, waiting list
  - Lectures will be recorded or pre-recorded

Working at a Distance

- Slack workshop for group communication (see Canvas for link)
  - Use #general for general Q&A
  - Each lab assignment has a channel for questions about that lab
  - If you PM me a question, I will share it (anonymously) on #general unless you ask me not to.
  - #random for off-topic chat (keep it clean!)
- During lab sessions:
  - When you want to speak to a supervisor, start a Zoom meeting and post an invitation to #supervision-help-requests
  - We will answer the requests in order
  - Please do not use this channel for anything else

Working at a Distance

- During Zoom meetings
  - Using camera and mic is optional
  - If you send me a question by private message, I will share the question and answer (without your name) with everyone unless you ask me not to.
- Pair programming:
  - You can work simultaneously on the same file using Zoom's "Remote control" feature
  - Or use driver-navigator style
  - Or work on separate parts of the code
  - Use good git discipline!
- When you have lots of Zoom meetings: take regular breaks
- Go outside, get exercise

Course website

https://chalmers.instructure.com/courses/12196

Structure of the course

- Group project to be completed in groups of 4-5 by end of the course
- Lectures uploaded to Canvas plus assigned readings over the course
- 4 mandatory laboratory assignments, starting from the first week of the course. Assignments are not graded, but must be completed and passed
- You need to have formed project groups by Wednesday on the second week of the course
- Weekly supervised group meetings with your supervisor starting from week 2
- Presentation during exam week to demonstrate your application to us and the class

Group project

- Please start forming groups today - work in groups of 4-5 people (ask me about exceptions)
- Use channel #seeking-project-partners
- During the second week of the course (W5) you will meet your supervisor for the first time. Discuss your project ideas with your supervisor and make sure it’s a good fit for the course - anything that has a lot of user interaction and requires persistent storage is generally a perfect fit

Group project grading criteria

When we grade the project we look at a number of things

- The technical level and functionality of the project
- The general design of the code. Is the code well-structured and layered? Is there a coherent code style throughout the code base?
- Is the project structure clean and void of external dependencies?
- Is there proper unit testing of the model?
- Does the code follow good practices and use JavaEE functionality properly?

Individual grading

In addition to the grading of the project we also grade the individual contribution to the project from each student

- Make sure you are active in the project from the beginning and continuously part of the programming and design decisions. Not being so will adversely affect your final grade
- We will use github.com/ejwa/gitinspector to gauge your contribution to the project, but only as a rough guide
- We will also ask you to complete an anonymous self-evaluation

Group project documentation

When handing in the project you should provide some documentation in PDF format containing the following

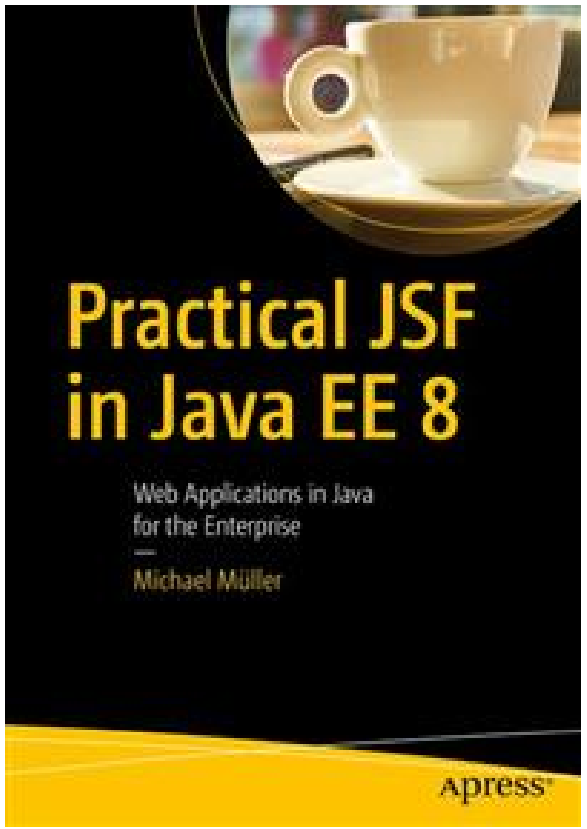
- Description of the project, including the technologies used
- A list of use cases
- A top level UML diagram describing the design and layers of the project
- An UML diagram of the model
- Anything else you think might help us to better grade your project

When you write your code

Please follow these guidelines when writing code

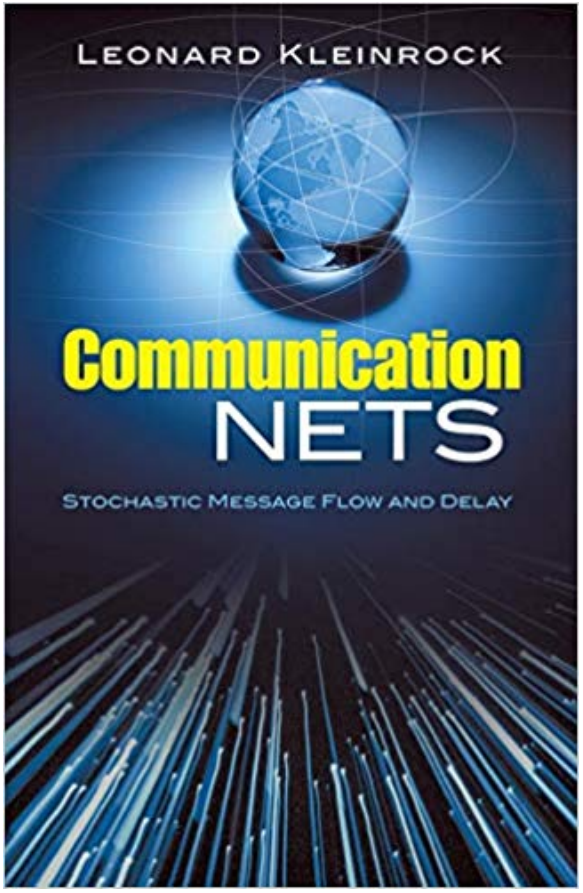
- Less is more. Try to keep your methods short and your indentation level to a minimum
- Don’t use unnecessary code comments – write self-documenting code with good class names, method names and variable names.
- Follow the same code conventions throughout the code base. Tools such as **CheckStyle** (checkstyle.org) and **PMD** (pmd.github.io/) can help you to achieve this

Resources



- The course book is *"Practical JSF in Java EE 8"* by Michael Müller (ISBN: 9781484230299)
- Use the web and Google for anything not covered by the book
- As the course progresses I will put additional links to articles, tutorials and documentation
- *"The Definitive Guide to JSF in Java EE 8"* by Bauke Schultz and Arjan Tijms (ISBN: 9781484233863) is recommended as additional reading material.

History of the Internet



1964

- Leonard Kleinrock of MIT publishes *"Communication Nets: Stochastic Message Flow and Delay".* Available in a 2007 edition (ISBN: 9780486458809)
- Paul Baran applies Kleinrock's theory for secure voice communication in the military
- The purpose? To defend U.S computers from Soviet nuclear attack, creating a network design with built-in redundancy

History of the Internet

Paul Barans implementation is later used in ARPANET

- *Advanced Research Projects Agency (ARPA)*
- Originally created in February 1958 by Eisenhower in response to the Soviet Union launching *Sputnik 1* in 1957
- Late 1960s: ARPANET *(Advanced Research Projects Agency Network) conceived*
- Packets sent across wires with data shared by multiple simultaneous sessions. Increases network efficiency and robustness

What was the ARPANET?

For each of these three terminals, I had three different sets of user commands. So if I was talking online with someone at S.D.C. and I wanted to talk to someone I knew at Berkeley or M.I.T. about this, I had to get up from the S.D.C. terminal, go over and log into the other terminal and get in touch with them.…

I said, oh man, it's obvious what to do: If you have these three terminals, there ought to be **one terminal that goes anywhere you want to go** where you have interactive computing. That idea is the ARPAnet.

– J. C. R. Licklider (1969)

What was the ARPANET?

We set up a telephone connection between us and the guys at SRI (Stanford Research Institute)

We typed the L and we asked on the phone, *"Do you see the L?"*

"Yes, we see the L," came the response.

We typed the O, and we asked, *"Do you see the O."*

"Yes, we see the O."

Then we typed the G, and the system crashed…

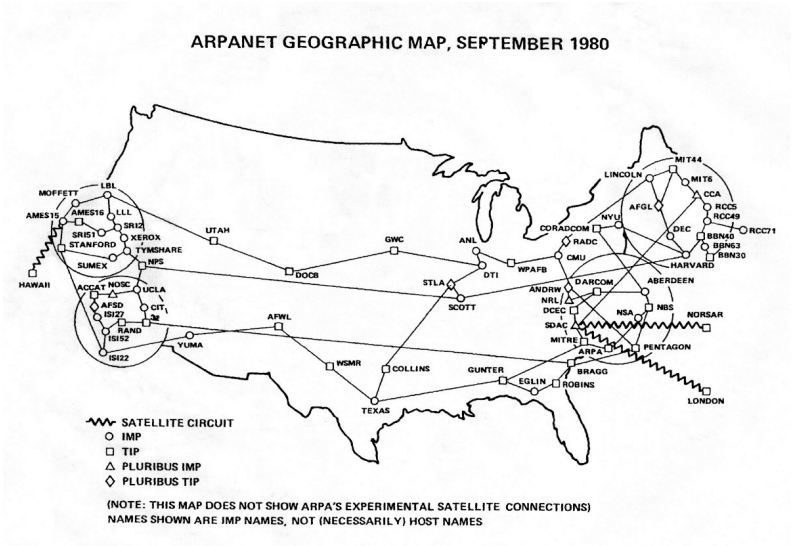
Yet a revolution had begun

What was the ARPANET?

The ARPANET in December 1969

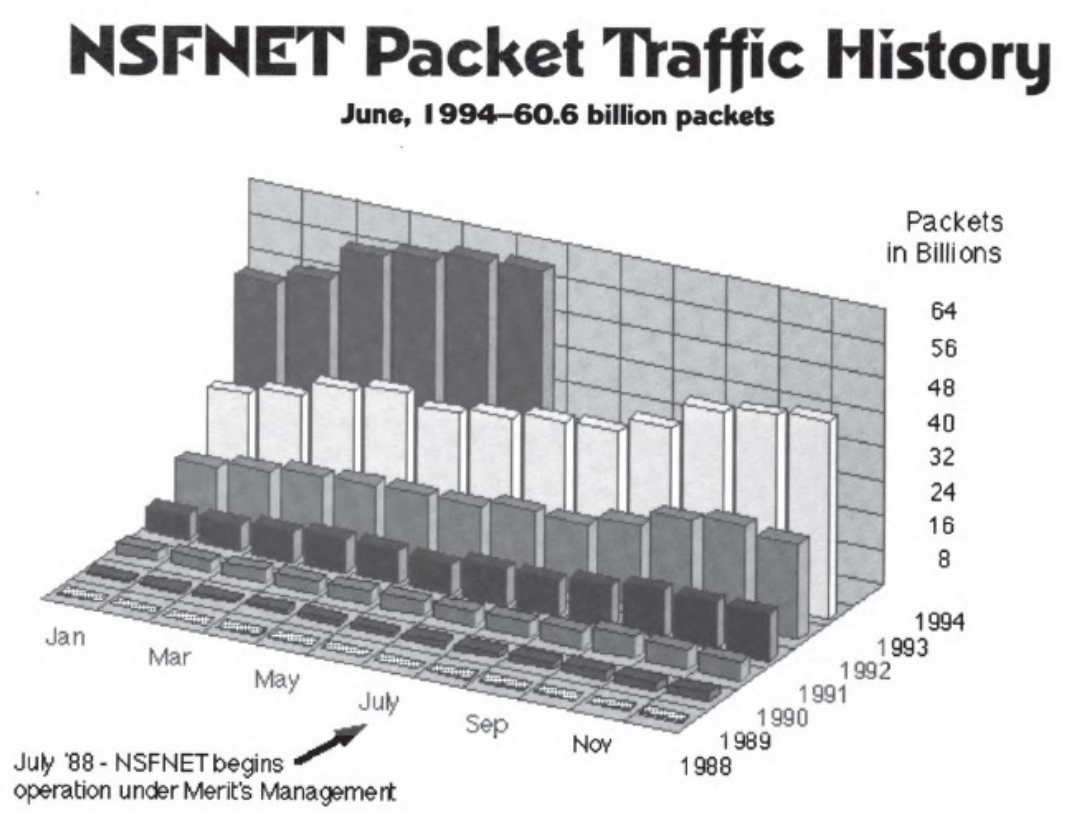
- 1969: Network consists of 4 computers
- 1974: First implementation of TCP/IP and packet switching to allow multiple computers to communicate on a single network. First use of *Internet* to describe the internetworked system
- Data consists of a header, used by networking hardware to direct the packet to its destination, and a payload, which is extracted and used at the destination

What was the ARPANET?



- Grew very quickly during the 1970's with many institutions joining. The network gained a direct link to London via a satellite Link
- 1985: NSFNET project initiated - sponsored by the *National Science Foundation (NSF)* from 1985 to 1995
- ARPANET decommissioned in 1990 in favor of the NSFNET (becoming the Internet)

NSFNET starts growing



- July 1988: NSFNET is officially launched
- 1991: NSF removes access restrictions, resulting in rapid growth of the commercial ISP business

The World-Wide-Web

- 1990: First running web server developed by Tim Berners-Lee
- A NeXT machine at CERN running the NeXTSTEP operating system
- **CERN httpd** was released to the public in June 1991
- Still available here, www.w3.org/Daemon

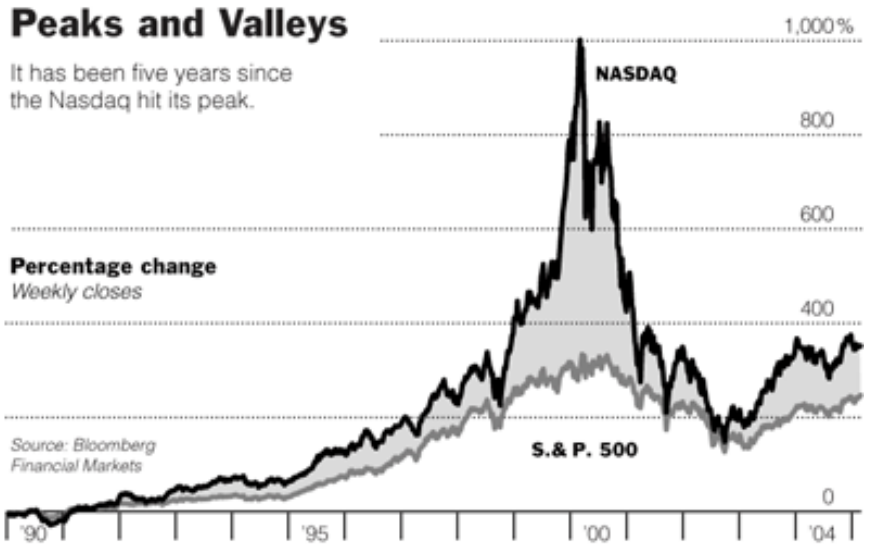
www.w3.org/People/Berners-Lee/1991/08/art-6484.txt

Public adoption of the Internet

- 1993: AOL spreads commercial Internet access to millions of people (dubbed the eternal september)

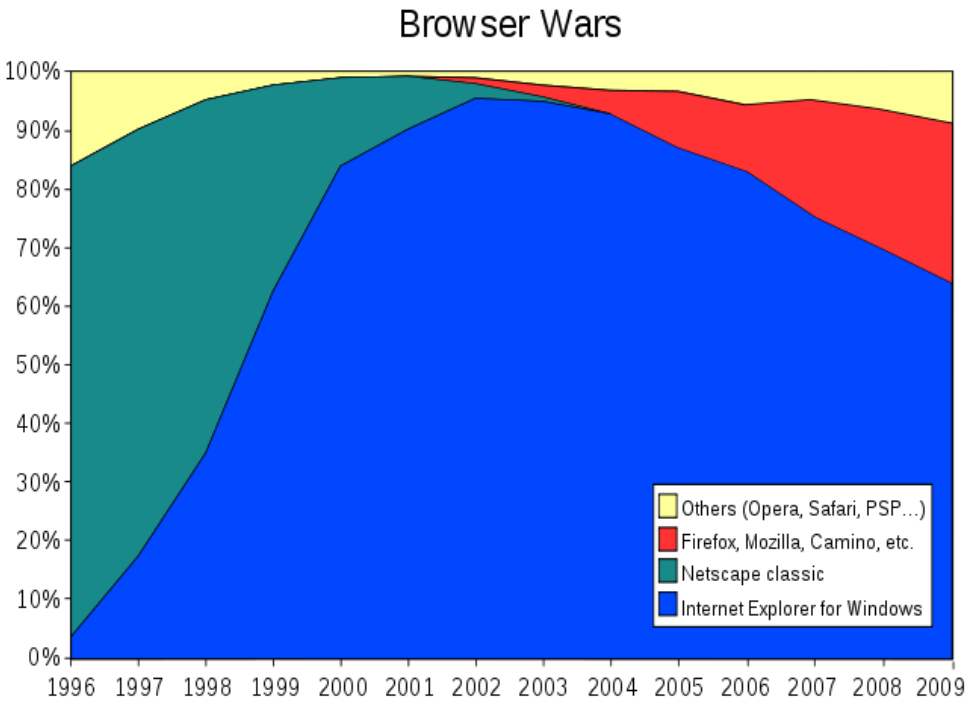
- Usenet was previously mostly restricted to universities and research institutions. Every September, many incoming students would acquire access to Usenet for the first time, taking time to become accustomed to standards of conduct and "netiquette". After a month or so, these new users would either learn to comply with the social norms or tire of using the service

The .com boom



- 1995-2000: Investors throw money at anything related to the Internet. Some shares can rise 2000% in one year
- 2000-2003: NASDAQ market crashes Loses $5 trillion (roughly 78% of its value)

The Browser Wars



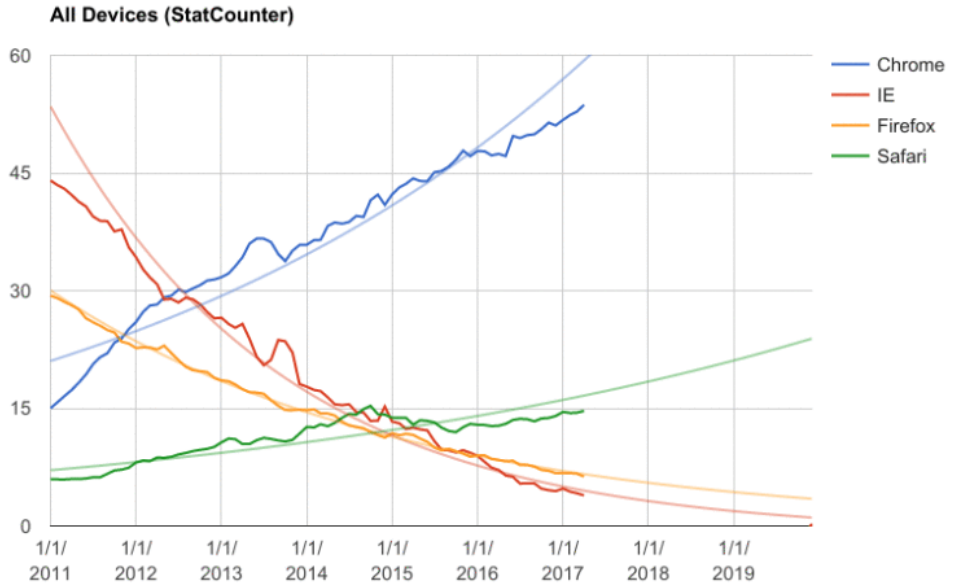
- The 1990s: different browsers had different features
- Web sites would work on some browser but not on others
- In the very early beginning, Mosaic was popular. Eventually Netscape Navigator and Internet Explorer became the most popular alternatives

The Browser Wars get Ugly

- Some people claim that Microsoft intentionally sabotage competing browsers by relying and promoting features specific to Internet Explorer

- People get really passionate, leading to *“browser activists”* doing campaigns to ridicule the competitor.

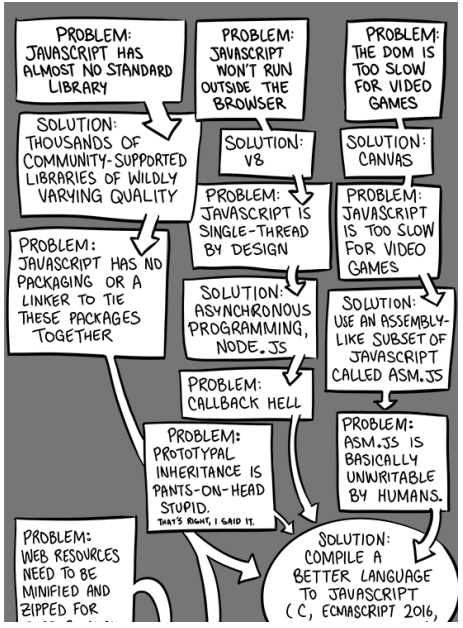
Browser wars continue



- Apple and Firefox fight back
- 2008: Google Chrome is released. Causing mass-adoption to that browser
- With more and more interactivity being introduced, the web rapidly transitions from a repository of documents to a repository of applications
- 2020: Google Chrome has about 70% market share

Web development is messy

The Web is now about apps. The problem is that none of the core technologies were designed with apps in mind



- NPM and Yarn
- Sass
- The TypeScript language and transpiler

Web standards

- W3C – World Wide Web Consortium
- WHATWG - Web Hypertext Application Technology Working Group. Founded in 2004 by individuals from Apple, Mozilla and Opera Software to force W3C to push the envelope and develop HTML
- HTML5 and CSS3 were designed with better document structuring and interactivity in mind
- The process will most likely always be a work in progress. Different browsers support different features and proposals of the standard. A good resource to use when checking if you should use a feature is caniuse.com

What to expect during the course

- Don’t expect mathematically precise definitions
- Expect to do a lot of debugging
- Expect a lot of frustration
- Expect us (the lab supervisors) to sometimes pull our hair out in frustration when helping you
- But also: expect a lot of fun and opportunity for experimentation