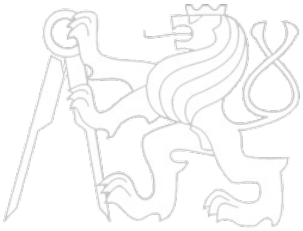
Middleware Architectures 1

Motivation and Course Overview

doc. Ing. Tomáš Vitvar, Ph.D.
tomas@vitvar.com • @TomasVitvar • https://vitvar.com



Czech Technical University in Prague
Faculty of Information Technologies • Software and Web Engineering • https://vitvar.com/lectures

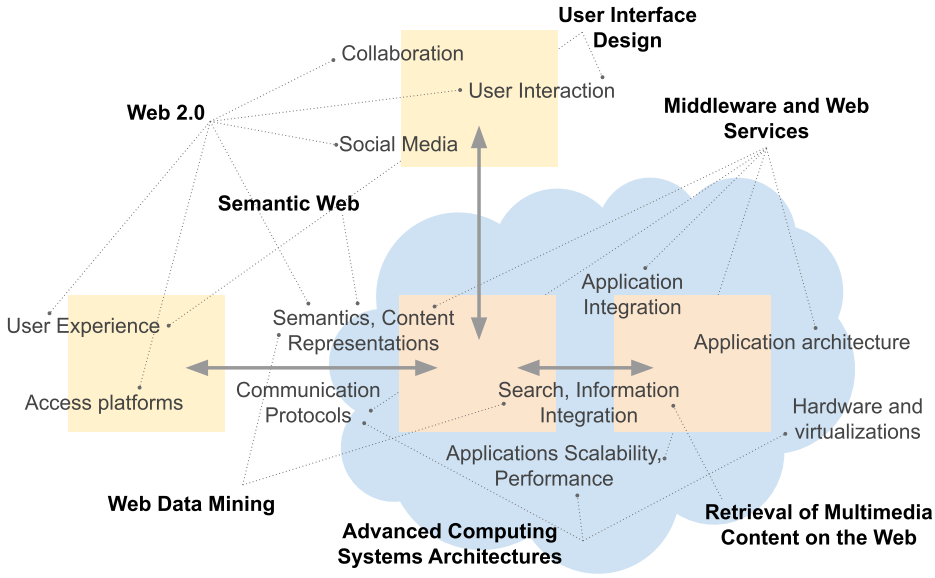
Modified: Sun Nov 19 2023, 19:53:57
Humla v1.0

Hellos

- Tomáš Vitvar – lectures (Czech)
  - *Web engineering program chair at CTU FIT*
  - *Technology Architect Director at Oracle*
  - *Research*
    - → *Web Services, Web Intelligence, Semantic Web*
  - *Industry*
    - → *Large scale integration architecture projects (Vodafone UK, IKEA IT Sweden, Turkish Telecom)*
    - → *Design, governance, troubleshooting, performance tuning*
- Jaroslav Kuchař – labs (Czech)
  - *Research assistant at CTU FIT, Seznam.cz*
  - *Web usage mining, big data analytics, Web services*
- Milan Dojčinovski – lectures and labs (English)
  - *research assistant at CTU FIT, Ph.D. candidate*
  - *Semantic Web, Linked Data, NLP, Web services*

Web Engineering Curricula – bird's view

- Web engineering is...
  – *far beyond PHP and HTML!*



Overview

- Course at a Glance
  – *Motivation and Scope*
  – *Requirements and Organization*
- Assessment
- Communication and Resources

What is Middleware?

- Say anything you think has something to do with middleware
  - *Architecture*
  - *Tehnology*
  - *Tools*
  - *Concept or style*
  - *Methodology*
  - *...*
  - *No worries, there is no bad answer!*

Motivation in Brief

- Systems rely on complex infrastructures
  - *A lot of data and many processes, internal and external*
  - *As people communicate, underlying systems must too*
  - *But:*
    - *→ variety of data formants, technologies, protocols*
    - *→ variety of architectures, client-server, peer-to-peer, ...*
- Rapid changes in applications' functionalities
  - *modular development*
  - *reuse of application logic*
  - *low costs – do it now and quickly!*
  - *Good performance*
    - *→ frequent changes in applications' loads, peek hours*
    - *→ scalability – effective load balancing*
    - *→ low costs – cheaper to outsource?*

Scope

- Architectural and conceptual basis
  - *What is architecture – enterprise, processes, data, software*
  - *Service concepts, integration patterns, middleware*
- Backend technologies
  - *Communication protocols – HTTP, TLS, HTTP/2*
  - *Application backend and requests handling*
  - *Advances of REST, MOM*
- Performance and Scalability
  - *Performance tuning*
  - *Load balancers*

Overview

- Course at a Glance
  - *Motivation and Scope*
  - *Requirements and Organization*
- Assessment
- Communication and Resources

Prerequisites

- Object-oriented programming
  - *Principles*
    - → *class, object, inheritance, encapsulation, ...*
      - → *basis for service concepts*
- JavaScript
  - *All code examples will be in JavaScript*
  - *All lab work will be in JavaScript*
- Web Architecture
  - *Basics of REST, HTTP, URI/URL*
- Other
  - *Networking*
  - *Basics of OS*

Organization of Lectures

- 12 Lectures
  - *Czech: Mon 9:15-10:45, T9:107*
  - *English: TBA*
- Plan
  1. *22.09.2024 – Motivation and Course Overview*
  2. *30.09.2024 – Information System Architectures*
  3. *09.10.2024 – Service Architecture 1*
  4. *07.10.2024 – Service Architecture 2*
  5. *14.10.2024 – Communication protocols, HTTP, TLS*
  6. *21.10.2024 – HTTP/2 and HTTP/3*
  7. *28.10.2024 – No lecture*
  8. *11.11.2024 – HATEOAS, Caching, Concurrency Control*
  9. *04.11.2024 – Monolithic vs Microservices Architectures*
  10. *18.11.2024 – Messaging middleware*
  11. *25.11.2024 – High Availability and Performance 1*
  12. *02.12.2024 – High Availability and Performance 2*
  13. *09.12.2024 – Rserve*
  14. *16.12.2024 – Rserve*

Organization of Labs

- Individual work (no teams!)
- Labs every second week
- Number of labs: 6
  1. *Introduction - FIT Workspaces, setup*
  2. *REST – Basics, development of a REST service*
  3. *Security, TLS*
  4. *Messaging systems*
  5. *Load balancers, nginx*

Methodology for Lab Work

- No app development, not directly related assignments
  – *assignment every second week*
  – *be prepared for the lab!*
  – *work alone, ask others for advices*
  – ***Results:***
    → *5 completed tasks*
    → *you will submit results to gitlbab@FIT*

Overview

- Course at a Glance
- Assessment
- Communication and Resources

Assessment

- Labs
  - *Every task gives you the maximum of 6 points = 30 points in total*
  - *Activity in labs gives you the maximum of 10 points*
  - *Total maximum points = 40, **to pass**: 20 points minimum*
- Final exam
  - *Written exam: 3 exercises, 1 hour*
    - *→ each gives you a max. of 20 points, the total is 60 points*
    - *→ To pass, you need to have at least 50% from each exercise!*
  - *Final score:*
    - *→ 100 points maximum*

Final Marks

Mark	Points	In words
A	100–90	výborně
B	89–80	velmi dobře
C	79–70	dobře
D	69–60	uspokojivě
E	59–50	dostatečně
F	49–0	nedostatečně

Source: http://www.cvut.cz/pracoviste/pravni-odbor/dokumenty/studijni-predpisy/studijnirad.pdf

Overview

- Course at a Glance
- Assessment
- Communication and Resources

Communication

- Language
  - *Text: English (slides, tweets, posts, instructions, etc.)*
  - *Voice: Czech and English (English version of the course)*
- Direct
  - *you can always contact me directly at tomas@vitvar.com or @TomasVitvar*

Overview of Resources

- Overview of resources

Item	URL
Course slides	http://mdw.vitvar.com
Courses@FIT	https://courses.fit.cvut.cz/NI-AM1/
Lab projects	https://gitlab.fit.cvut.cz/
Assessment	https://grades.fit.cvut.cz/courses/NI-AM1/

- Books
  - *L. Richardson, S. Ruby: RESTful Web Services: Web services for the real world, O'Reilly Media, May 2007, ISBN 9780596529260.*
  - *I. Grigorik: High Performance Browser Networking, O'Reilly Media, May 2013, ISBN 978-1-449-34476-4.*
  - *Thomas Erl: Service-Oriented Architecture: Concepts, Technology, and Design. Prentice Hall, Aug 2, 2005.*

About Slides

- Humla – Open Source HTML5 Presentation System
  - *every slide has a unique URL*
  - *all figures linked with Google drawings*
  - *auto-generated PDFs (1 and 2 slides per page) using travis-ci*
  - *running local (with local nodejs-based http server), and in github pages*
  - *Suggest edits or correct errors by pull requests at mdw github repo*
- Keys
  - 1 *default browsing mode*
  - 2 *slideshow mode (automatically scales to fullscreen)*
  - 3 *grid (overview) mode*
  - 4 *print mode, 2 slides per page*
  - ← *slide left*
  - → *slide right*