

# 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: Mon Sep 22 2025, 10:19:00  
Humla v1.0

## 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, peak hours*
  - *scalability – effective load balancing*
  - *low costs – cheaper to outsource?*

## Scope

- Architectural and conceptual basis
  - *What is an architecture – methodology, global, software*
  - *Service concepts, integration patterns, microservices, middleware*
- Backend technologies
  - *Communication protocols – HTTP, TLS, HTTP/2*
  - *Application backend and requests handling*
  - *Advances of REST, gRPC*
  - *Kubernetes*
- Performance and Scalability
  - *Performance tuning*
  - *Load balancers*

## Overview

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

## Prerequisites

- Operating systems
  - *Basics of Linux*
- Computer Networks
  - *Basics of networking concepts*
- JavaScript
  - *Code examples will be in JavaScript*
  - *Lab work will be in JavaScript*
- Web Architecture
  - *Basics of REST, HTTP, URI/URL*

## Organization of Lectures

- 12 Lectures
  - *Czech: Mon 9:15-10:45, JP:B-571*
  - *English: TBA*
- Plan
  1. 22.09.2025 – *Motivation and Course Overview*
  2. 29.09.2025 – *Information System Architectures*
  3. 06.10.2025 – *Introduction to Service Architecture*
  4. 13.10.2025 – *Microservices and Cloud-Native Architectures*
  5. 20.10.2025 – *Communication protocols, HTTP, TLS*
  6. 27.10.2025 – *HTTP/2 and HTTP/3*
  7. 03.11.2025 – *HATEOAS, Caching, Concurrency Control*
  8. 10.11.2025 – *gRPC*
  9. 17.11.2025 – *No Lecture*
  10. 24.11.2025 – *Synchronous and Asynchronous I/O*
  11. 01.12.2025 – *High Availability and Performance*
  12. 08.12.2025 – *Performance Tuning*
  13. 15.12.2025 – *Reserve*

## 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 **FIT GitLab***

## FIT Workspaces

- All work you will do in **FIT Workspaces**
  - *Cloud-native Development Environment running in Kubernetes*
  - *Each student has an access to their own workspace*
  - *Workspace is ready for the **lab** work*
- Access
  - *You use your FIT username and a key (stored in GitLab)*
    - `ssh vitvatom~am1@fit-workspaces.ksi.fit.cvut.cz`
  - *You can use SSH CLI, VSCode or IntelliJ with Remote SSH plugin*
  - *You have sudo access if you want to add packages*
  - *For more Information see [FIT Workspaces Wiki](#)*

## 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](mailto:tomas@vitvar.com) or [@TomasVitvar](https://twitter.com/TomasVitvar)*



# Overview of Resources

- Overview of resources

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

- 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