



Aalto University
School of Electrical
Engineering

Microservice architectures and serverless computing

Santeri Paavolainen

10.1.2019

Introduction

1. **Basic course information**
2. **Course structure and schedule**
3. **Evaluation criteria**
4. **Group and personal course work**

A?

Aalto University
School of Electrical
Engineering

Course practicalities

Overall

- **Masters-level course**
- **Periods III and IV / 2019**
- **One weekly lecture** (more on schedule later)
- **All participants are a member in a group**
- **One weekly Q&A session (per group)**
- **Exam and personal coursework DL in April**
- **Teacher:**
Santeri Paavolainen
santeri.paavolainen@aalto.fi
- **Formal announcements and submissions via MyCourses**
- **Informal discussion on a Slack channel**
 - aalto-services-2019.slack.com (see mycourses for invitation link)

Groups

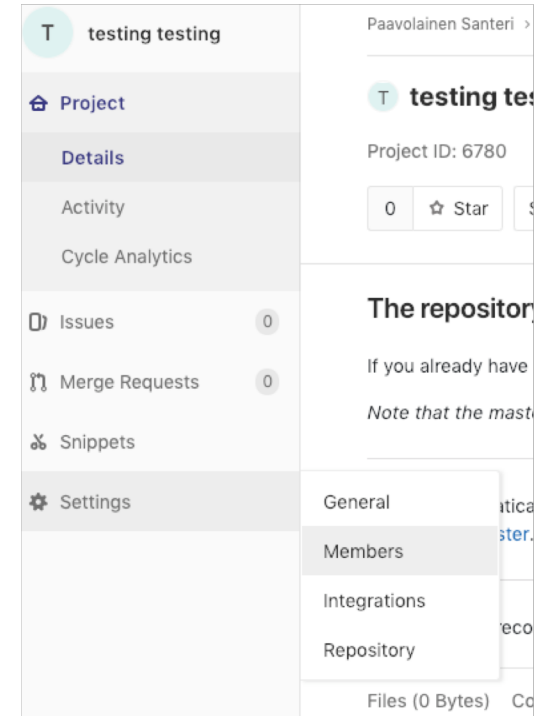
- **2-5 students per group**
- **Either**
 - Form a group yourself, send member list to me by 10.1. 19:00
 - (you can give the group a name, if you want to)
- **or**
 - You will be assigned to a random group after 10.1. 19:00
 - (group name at my own digression and whim)
- **If the group has >2 members and there's a slacker, please contact me**
 - We'll have a discussion first
 - If there's a clear case, slackers will form a group of their own
- **Each group is responsible for their own activities**
- **Group assignment grades are for the whole group**

Exercise Q&A session scheduling

- **Assigned on a per group basis (not individually)**
- **Available slots:**
 - Mondays: 16-17, 17-18
 - Tuesdays: 9-10, 10-11, 11-12
 - Fridays until 15.2.: 14-15, 15-16 (except on 25.1.),
 - Fridays from 1.3.: 12-13, 13-14
- **Maximum on 7 people per slot, mind group sizes**
- **Once groups are finalized**
 - Will post a link to Google Sheets with group names and slots
 - Fill in preferences for at least three per group from 1 to 3 (most to least preferred) by 15 o'clock 11.1.
 - Groups will be assigned on 11.1. (because first sessions already on 14.1.)

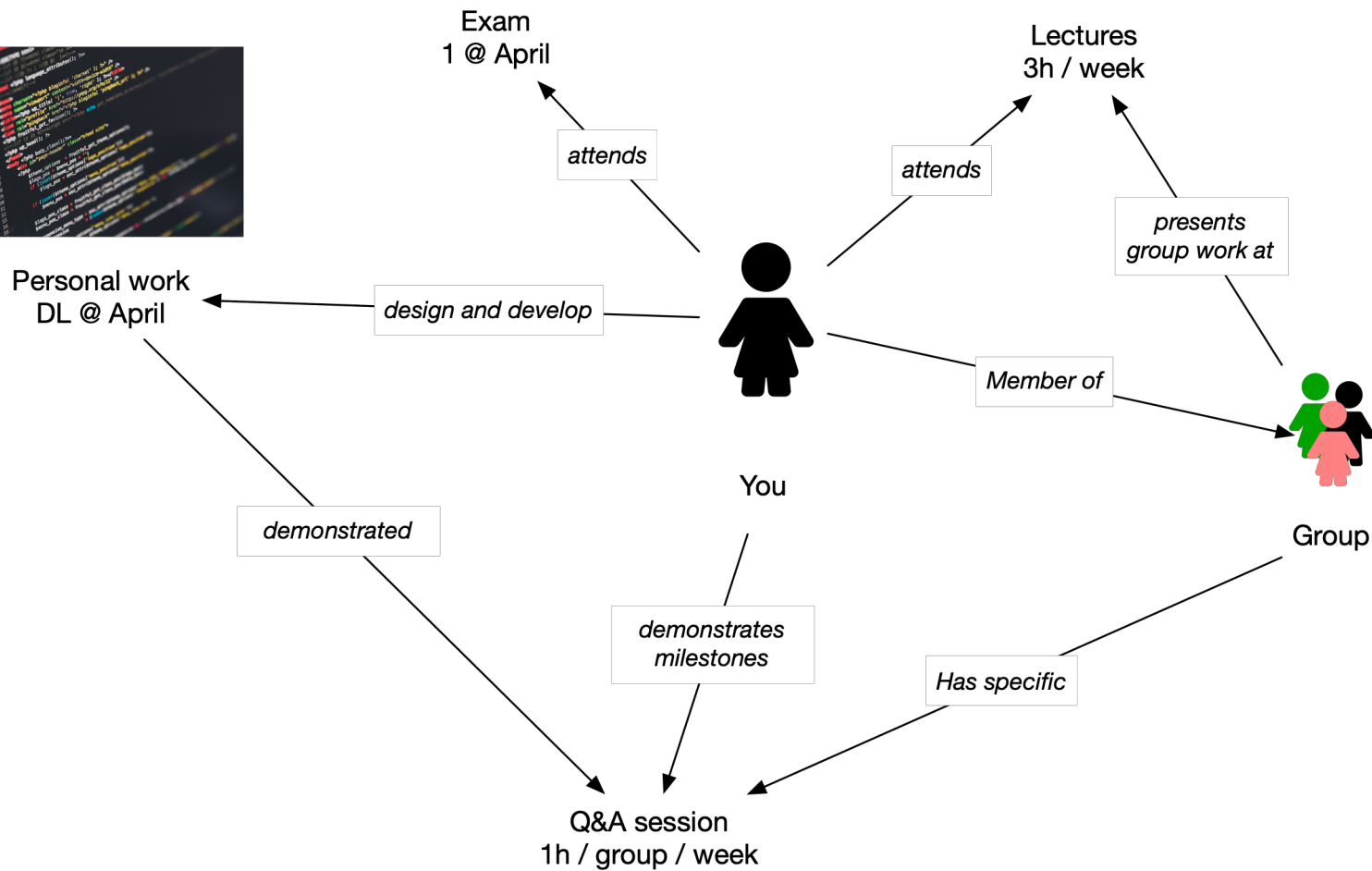
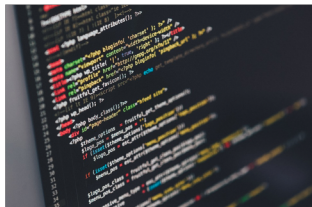
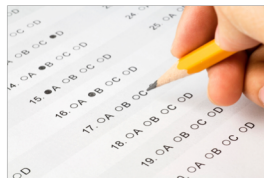
Submitting personal coursework

- **Has to be accessible to staff via version.aalto.fi**
- Aalto university students use HAKA login
- **How to:**
 - Create a personal repository
 - Share it to group **microservices-serverless-course** with **guest** privileges
- **If you need help, ask on Slack, or at your group's session**
- **Final submission will be done by providing a commit hash on the repository**



Computing environments

- **You should be able to do everything on a laptop**
 - Docker and Kubernetes (we'll have a tutorial later)
 - If you don't have one or cannot run them, contact staff
- **If you want to, you can use Google Cloud or Amazon Web Services**
 - Vouchers available for GCP that are worth some \$\$\$ of usage
 - See mycourses for details and instructions
 - In practice, this would be running your containers on GKE/ECS/EKS, but you could also leverage other infrastructure services (databases, tracing, etc.)
 - Just be careful, if you do not have experience in AWS/GCP previously, using them would add a learning load (without extra credits)





Aalto University
School of Electrical
Engineering

Course structure & schedule

Overall course schedule

- **Lectures**
 - Thursdays 14-17 in SimLab (OIH building)
 - No lecture on 24.1. → moved to Friday 25.1. 12-14
 - No lecture on 21.2. (exam week)
 - Last lecture on Thursday 4.4.
- **Group work and presentations**
 - 1st presentations: Friday 25.1.
 - 2nd presentations: Thursday 28.2.
 - 3rd presentations: Thursday 28.3.
- **Exercise sessions**
 - Primarily a Q&A session for groups, 1-3 groups per session
 - Participation not mandatory, **but highly recommended**
- **Personal coursework**
 - Final submission DL: Sunday 14.4. 23:59
 - Milestone demos in exercise sessions: weeks 5-6, 7-9 and 11-12
- **Exam: 11.4. 16:30-19:30 (TU7, TUAS)**

Lectures (rough schedule, subject to change)

- **Lecture 1**
 - Practicalities, microservice and serverless basics,
- **Lecture 2**
 - Containers, Docker and Kubernetes
- **Lectures 3-6**
 - Loosely coupled architectures, architectural patterns, labor division, tools, patterns, development, ...
- Lecture 3, 7 and 11: Group presentations (about 1h total)
- **Lectures 7-9**
 - HA, failure handling, performance issues, ...
- **Lectures 10-11**
 - Serverless computing
- **Lecture 12: Course recap**

Q&A sessions

- **1-2 groups per slot, once a week, 1 hour per slot**
 - Except holidays, exam week and some other exceptions etc.
 - Start from 14.1. onwards (week 2), **no sessions on week 3 (21.-25.1.)**
- **Primarily meant as Q&A sessions for**
 - Discussions with teacher about assignments (personal and group)
 - Open questions on course topics
 - Informal course work progress demonstrations
- **Milestone demonstrations**
 - These affect grading
 - Short demonstrations of progress in personal work — to everybody present
 - Some leeway in schedule for these (2 weeks)
- **Participation not mandatory**

What this course is and is not

- **Microservice architectures and serverless computing**
 - What, why and how
- **These are coincidental and are referred, but they are not within the scope of the course**
 - DevOps, Development of Web Services, or Operations (but yes to *architecting for operations*)
 - Continuous Integration and Continuous Delivery
 - REST API design (see OpenAPI and Zalando guidelines)
 - AWS, GCP and Azure
 - Security and secure programming (but I recommend CS-E4330)



Aalto University
School of Electrical
Engineering

Course evaluation criteria

Santeri Paavolainen

Overall grading

20%	Group assignments
30%	Exam
50%	Personal coursework (exercise)
5%	Completing course feedback

- 50% of all points required for grade 1

Group assignment grading

- **Weight on final grade: 20%**
- **Three group assignments, each graded from 0 to 3**
 - 0 = no-show at presentation session
 - 1 = presented, but ...
- **Group work is evaluated based on short 5-10 minute presentation from the group on the given topic**
- **Total: 9 pts (x 20%)**

Exam grading

- **Weight on final grade: 30%**
- **Pretty standard exam, potentially something like**
 - Explaining terminology
 - Designing or describing an architecture
 - Etc.
- **3-5 questions, 4-6 points per question**

Personal coursework

- **Weight on final grade: 50%**
- **0-3 pts for general evaluation of the architecture**
 - Separable? Logically consistent? Division of responsibility?
- **0-3 pts for maintainability**
 - Can you give the code to someone else? Would they understand how it works and be able to work on it?
- **3 topic areas x 0-3 points each**
 - Implementation, demonstratability, coverage in implementation
- **0-3 pts for milestone demonstrations**
 - 1 point for demonstrating progress in milestone exercise sessions
- **Rejected on unattributed copying, deductions on excessive code re-use**
- **Total: 18 pts (x 50%)**

Completing course feedback

- **Weight on final grade: 5%**
 - Yes, weights add up to 105%
- **This is a “beta” version of the course, so feedback and comments are important!**

A?

Aalto University
School of Electrical
Engineering

Group and personal course work

Santeri Paavolainen

Group work (1)

- **Three group assignments over the course**
 - Topics will be given on 17.1., 14.2. and 21.3.
- **5-10 minute presentation per group per assignment**
 - During beginning of lectures on 25.1., 28.2. and 28.3.
 - Equals to intro slide + 1-3 content slides
 - A few minutes of Q&A from audience
 - Slides distributed to all course participants (mycourses)

Group work (2)

- **Qualitative in nature**
 - Take a point of view / make a claim — and provide rationale / defend your PoV
- **Not expected to take significant amount of time to complete**

Personal course work

- **This is the largest single work item on the course**
 - Based on ECTS: 135h of work for the whole course, removing lectures, group assignments and exercise sessions leaves ~ 80h → 6h / week
- **You are expected to design and develop a**
 - Microservice architecture with >2 distinct services
 - At least one serverless or stateless component
 - Integrates three aspects of microservice architectural patterns (see later)

Personal course work

- **It must work at the level of a “demonstrator” piece**
 - Actual functionality can be trivial or mocked
 - However, the three microservice aspects cannot be mocked, and the overall inter-service operations must be functional
- **Focus is on how the service is structured and operated**
 - The “business functionality” is relevant only as much as it is needed for testing, demonstrations etc.
 - Of course, working on a project is probably more meaningful if the functionality makes sense to you

Practical example

- **“To do” management service (see todomvc.com for inspiration) with separate components:**
 1. Reverse proxy for directing requests to different back-end services
 2. Authentication service (using OAuth2)
 3. User interface endpoint (SPA or HTML renders)
 4. Model storage service (abstracts storage)
 5. SQL database (for model persistence)
- **Functionality very minimal, does not have to be pretty**

“Aspects”?

- **Course covers a wide variety of design and operational aspects of microservices**
 - HA, failovers, tracing, logging, service authentication, discovery, ...
- **Infeasible to implement all of these in the course work**
- **Each student must pick three separate aspects and implement these within the scope of their work**
 - These must be functionally “complete” (no mock-ups)
 - These must be demonstratable, e.g. can be shown to work and cover the problem they are meant to solve

Aspects

Logging	Service AAA	Role-based user AAA	Discovery
Service degradation	Monitoring	Tracing	Continuous deployment
Chaos engineering	Backups and disaster recovery	Caching	Secret management
Dynamic configuration	Geographical distribution	Automated scaling	Versioning
High availability	A/B testing	...	(suggest)

Selecting aspects

- **There is no firm deadline on selecting**
 - Apart from the actual course work deadline
- **You can pick one now, all three, and change your mind later**
- **... or just choose them after something like a month or so (you've still got about 2 months until course work DL)**
 - In the meantime, you can work on the basic microservice structure and functionality first
 - Which you have to define yourself, too
- **Feel free to suggest similar aspects too**

- You may do more than 3 aspects in your project — grading is based on the best 3

Final personal work submission

- **Done by adding the repository URL and a commit hash to mycourses course work submission (text file, e.g. do not submit ZIP files)**
 - Remember to keep the repository accessible until grading is finished
- **The repository MUST include**
 - README or other documentation that
 - *Describes setup steps necessary to get the environment running*
 - *Describes the overall system functionality and its architecture*
 - *Lists the three aspects that have been implemented, how they have been implemented, and how they can be demonstrated to work*
 - Actual code of the things mentioned above

Plagiarism and IPR

- **Use of project generator stubs is okay**
 - And similar scaffolding and samples from documentation
- **Since the functionality of the deployed service is not at core here, a well-documented use of existing examples is acceptable as long as it follows the license of the code correctly**
 - No explicit license = NO RE-USE PERMITTED
- **Remember: Non-attributed copying of code is plagiarism**
- **Also: Submitting a course work that is mostly copy-pasted, even when properly attributed, is not capable of demonstrating your grasp of the course matter**
 - Make your own conclusions ...



Aalto University
School of Electrical
Engineering

Reminder of action points

Groups

- Send list of members via email by 19:00 **today**
- ... or be randomly assigned

Slack

- See mycourses for invitation link

Repository

- Create for your personal work and share to staff

- Exercise slots

- Once groups assigned, fill per-group preferences to Google Sheets
- Deadline 11.1. at 15 o'clock
- **AWS/GCP vouchers**
- See mycourses for information