WASA INTRODUCTION

Lecture "Web Applications and Service-oriented Architectures" WASA

Practical Course "Microservice2Go" M2Go
(Pro-)Seminar Topics

WASA1 - Winter Semester - Bachelor
WASA2 - Summer Semester - Master

COOPERATION & MANAGEMENT (C&M, PROF. ABECK), KIT FACULTY OF INFORMATICS

Niklas Sänger

Stefan Throner

Sebastian Abeck

Anh Kha Nguyen

Sergiu Panainte

Michael Schneider

Fabian Lübbe

KIT – The Research University in the Helmholtz Association

www.kit.edu

This course unit [CM-W-WAS] describes the content and the organization of the lecture "Web Applications and Service-oriented Architectures" (WASA) and the practical course Microservice2Go (M2Go) provided by the research group Cooperation & Management (C&M, Prof. Abeck). The M2Go practical course is closely linked with the WASA lecture. Although only the WASA lecture can be taken without passing the M2Go practical course, it is recommended to participate both in WASA and M2Go in parallel. Since the number of WASA lecture places and M2Go practical course places is limited, interested students must apply for a place.

WASA1 (Bachelor): Current concepts of software development and architectures (including Microservices, REST, gRPC, Domain-Driven Design, DevOps, CI/CD, Build Pipelines, Container-virtualized Infrastructures) as well as related technologies and tools (including HTTP, Go, Swagger, Postman, JavaScript/TypeScript, Angular, GitLab-CI, Docker, Kubernetes, Prometheus) are introduced. These concepts and technologies are applied in a systematic engineering approach, called Unified Microservice Engineering (UME), to develop and deploy microservice-based web applications.
The practical course Microservice2Go1 (M2Go1) can optionally be taken in parallel to the WASA1 lecture. In M2Go1, the UME approach is practically applied with the example of a microservice-based car rental application. After a compact introduction to the programming language Golang, the M2Go1 participants learn the systematic engineering of a domain microservice and an application microservice which are both implemented in Golang.

WASA2 (Master): A concise summary of the concepts covered by WASA1 is provided. In WASA2, Identity and Access Management (IAM) and Decentralized Identities (DI) as an advanced IAM topic are presented. In its core, IAM is responsible for the authentication and authorization of users and services in a software application. In the lecture, leading IAM concepts and solutions (e.g., Keycloak, Open Policy Agent, Microsoft Entra Verified ID) are introduced to illustrate how the IAM challenges are solved in IT practice. A concrete microservice-based application dealing with the rental of cars is extended by authentication, authorization, and DI functionality based on the current Internet standards OpenID Connect, OAuth2, and DI-related adaptions of these standards.
The practical course Microservice2Go2 (M2Go2) can optionally be taken in parallel to the WASA2 lecture. In M2Go2, the IAM concepts presented in the lecture are practically applied on the existing analysis, design, and implementation artifacts of the microservices which are written in Golang.

The WASA lectures will be offered as a hybrid (i.e., mixture of presence and online) event. The WASA kick-off lecture will take place **online**
=== **on Wednesday, 23rd October 2024 at 9:45 am**

Each student who wants to take part in the online WASA kickoff lecture must
=== **send an email (in German) to cm.research@lists.kit.edu**
to apply for one of the restricted WASA lecture (and practical) places. Please do **only** use your **depseudonymized** KIT student email address (see https://my.scc.kit.edu/shib/pseudonymisierung.php for further information).

The lecture material is made available in **English**. During the lecture, the content is presented and discussed in **German**. The **oral examination** is conducted exclusively in **German**. All students write their **practical/seminar thesis** in **English**. Thesis templates are made available in **LaTeX**. Overleaf is used for the writing of the practical and seminar thesis.

C&M	Cooperation & Management
IAM	Identity and Access Management
KIT	Karlsruhe Institute of Technology
M2Go	Microservice2Go
UME	Unified Microservice Engineering
WASA	Web Applications and Service-oriented Architectures

[CM-W-WAS] Cooperation & Management: WASA Kick-off. \\sccfs.scc.kit.edu\OE\TM\VR\Mitglieder\2-1.WASA_M2Go_Aktuell\1.Lecture_Material

C&M's Research Areas and Cooperation Partners

Microservice Engineering

Microservice Architecture

Domain-Driven Design

Microservice APIs

DevOps

Identity and Access Management

Advanced Authorization

Decentralized Identities
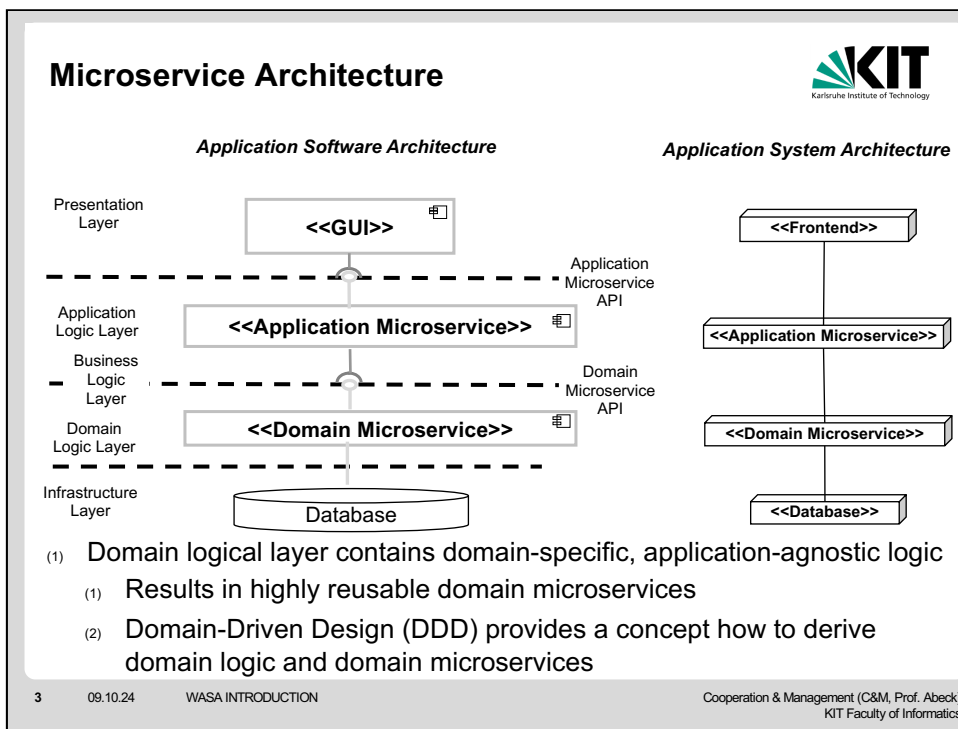
Data-Driven Decision Making

The research work carried out by C&M can be divided into two main areas:

(Microservice Engineering) For the business domain Connected Car, applications based on the concept of domain modeling and microservice architectures are developed. Relevant concepts applied in the microservice engineering approach include Domain-Driven Design (DDD), microservice API design and the implementation of the microservices and their APIs based on a microservice architecture. In addition to development (Dev), the operational aspects (Ops) are intensively taken into account.

(Identity and Access Management) Identity and Access Management (IAM) is a highly relevant crosscutting concern appearing in every web application. Basic IAM concepts include the authentication and authorization of human and technical users of a microservice-based application. Advanced IAM topics include

(i) Advanced authorization, which takes access decisions outside the application (external authorization) based on a broad spectrum of different attributes (fine-grained authorization).

(ii) Decentralized identities, which change the provision of identity in a way that users are owners of and have control over their identity data.

(iii) Data-Driven Decision (D3) making for an IAM company to support the answering of strategic questions.

DDD	Domain-Driven Design
D3	Data-Driven Decision
DevOps	Development and Operations
IAM	Identity and Access Management

2

Microservice Architecture

Application Software Architecture   *Application System Architecture*

(1) Domain logical layer contains domain-specific, application-agnostic logic
    (1) Results in highly reusable domain microservices
    (2) Domain-Driven Design (DDD) provides a concept how to derive domain logic and domain microservices

A microservice architecture is located on the application plane as it is introduced in ++Network, System, and Application Plane++. While the application software architecture is described by the logical layers specified by the DDD pattern LAYERED ARCHITECTURE, the system architecture introduces several subsystems. Two types of application programming interfaces (domain microservice API, application microservice API) are separating the logical layers on the application software architecture side and the microservices on the application system architecture side.

(Presentation Layer) This layer renders the UI elements in the browser. Technologies that support the implementation are Angular and Bootstrap. The presentation includes a logic which controls the interaction with the application microservice API. An optional API gateway is often used to provide cross-cutting concerns, such as load balancing or security aspects.

(Application Logic Layer, Application Microservice) This layer realizes the orchestration of domain microservices in order to provide the application logic to fulfill the requirements made to the application. A technology that supports the implementation of this functionality is Spring.

(Domain Logic Layer, Domain Microservice) This layer implement the domain microservices which mainly are Create, Read, Update, Delete (CRUD) operations on the domain objects.

(1) In contrast to a traditional three-layer application architecture, the business logic layer in a microservice architecture is split into two layers, the domain logic layer and the application logic layer. The reason for that is to promote the reuse of business logic functionality by distinguishing between application-agnostic (= domain logic) and application-specific (= application logic) functionality.

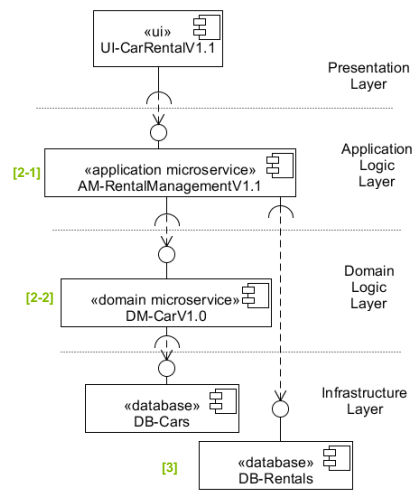|	
API	Application Programming Interface
CRUD	Create, Read, Update, Delete

Component Diagram CarRentalAppV1.1

(1) The application software architecture is based on the Domain-Driven Design (DDD) pattern LAYERED ARCHITECTURE

(2) Core functionality of CarRentalApp is the rental of cars

    (1) Application logic is provided by the application microservice AM-RentalManagement

    (2) Domain microservice DM-Car concerns the application-agnostic-functionality related to a car

(3) Information on cars and rentals is persisted in different databases

The CarRentalApp is a microservice-based application which serves as the example in WASA and M2Go to demonstrate the concepts and technologies covered in the lecture and the practical exercises. In the WASA lecture, Version 1.1 of CarRentalApp (i.e., CarRentalAppV1.1 [CM-G-CRA-V1]) is used as the continuous example.

(1) Domain-Driven Design (DDD) is an approach to develop software systems based on a domain model. An important DDD pattern is LAYERED ARCHITECTURE which introduces a domain logic layer into the application software architecture.

(2) Customers of a rental company (which in this specific case has the name BestRental) can rent cars for a defined period of time.

(2.1) An example of a concrete application logic in the CarRentalApp is keeping an overview of the rentals and making sure that a car is not rented twice in a certain time period.

(2.2) So far, only static information of a car (e.g., Vehicle Identification Number (VIN), model, brand) is handled by the domain microservice DM-Car.

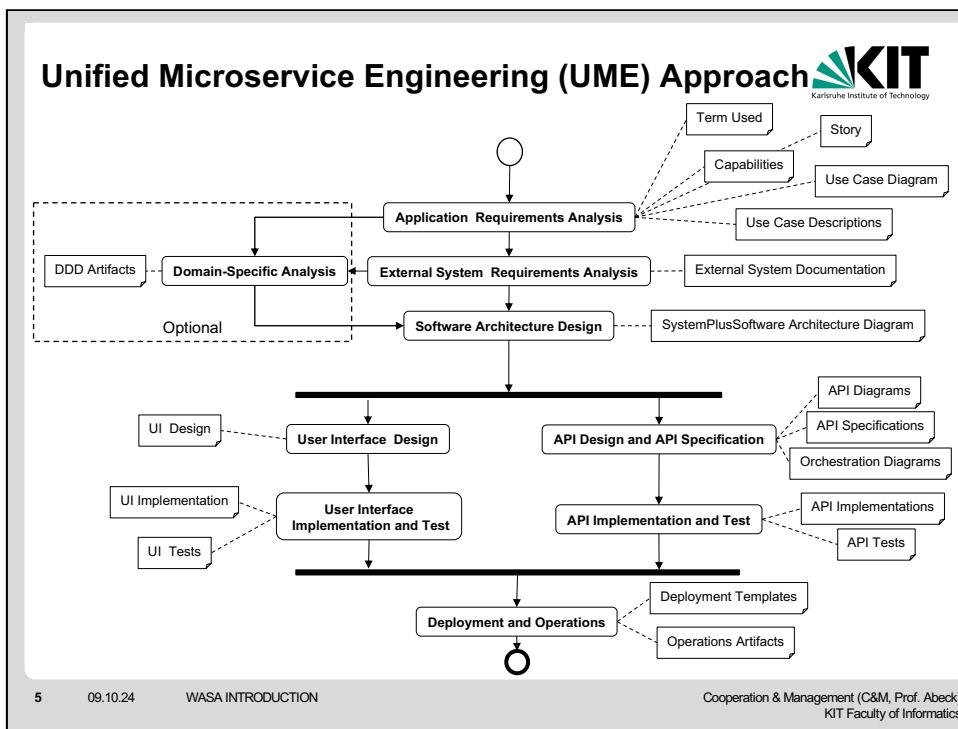(3) The information is stored in two separate SQL databases.

[CM-G-CRA-V1]   Cooperation   &   Management: CarRentalAppV1.
https://gitlab.kit.edu/kit/cm/teaching/carrentalapp/carrentalappv1

The Unified Microservice Engineering (UME) approach unifies two former approaches developed by C&M. UME consists of the well-known phases analysis, design, implementation and test, and deployment and operations. In the UME approach, the domain-driven aspects are an optional part of the engineering process.

(Application Requirements Analysis) The requirements are expressed by use cases with a specific structure. Optional analysis artifacts are vision and goals or the application sketch.
(Terms Used) Analysis artifact which defines all relevant for the understanding of the application.

(External System Requirements Analysis) The external systems (esp. enterprise applications, business services, or databases) into the microservice application are considered.

(Domain-Specific Analysis) In UME, the consideration of domain-specific aspects based on the concept of Domain-Driven Design (DDD) is intentionally kept optional. This analysis leads to Domain APIs which become part of the software architecture.
(DDD Artifacts) These include the ubiquitous language(s) and the domain model(s) of the domains relevant for the application to be developed.

(Software Architecture Design) The software architecture consists of (i) application microservices which are derived from the use cases, (ii) system microservices which integrate the external systems, (iii) optional domain microservices which provide the domain-specific logic, and (iv) Experience APIs which support the requirements of the different user interfaces.

(User Interface Design) (User Interface Implementation and Test) The User Interface (UI) can be designed and implemented and tested in parallel to the design and implementation of the different API types.

(API Design and API Specification) The API is systematically specified based on an API diagram. The API diagram of a Process API is derived from the use case descriptions and the software architecture diagram.

(API Implementation and Test) In UME, the microservices are coded and tested by using a Go-based framework.

(Deployment and Operations) A template-based deployment approach is used in the UME approach.

UI   User Interface
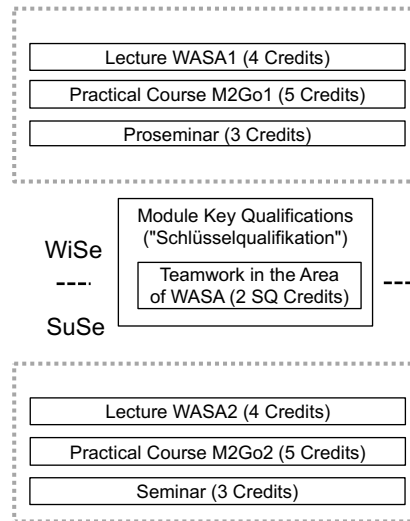UME   Unified Microservice Engineering

Lecture Plan

Datum	WASA	Praktikum
24-10-23	WASA INTRODUCTION	
24-10-30	Veranstaltungsorganisation, WASA INTRODUCTION: ONBOARDING, TOOLS AND PROGRAMMING	1_ONBOARDING 2_GOLANG
24-11-06	TOOLS AND PROGRAMMING, FOUNDATIONS	2_GOLANG
24-11-13	*24-11-12 (bis 12 Uhr): Anmeldung zur mündlichen Prüfung der WASA1-Vorlesung* ANALYSIS, DESIGN	*24-11-12 (bis 12 Uhr): Initiale Statusmitteilung -> INITIAL SUBMISSION* 2_GOLANG, 3_1_ME_Introduction
24-11-20	DESIGN	2_GOLANG, 3_2_ME_DM-Car, 3_3_ME_CarRentalApp
24-11-27	DESIGN	2_GOLANG, 3_2_ME_DM_Car, 3_3_ME_CarRentalApp
24-12-04	IMPLEMENTATION, DEPLOYMENT	3_2_ME_DM-Car, 3_3_ME_CarRentalApp
24-12-11	DEPLOYMENT	3_2_ME_DM-Car, 3_3_ME_CarRentalApp
24-12-18	DATA-DRIVEN DECISION: INTRODUCTION	*24-12-17 (bis 12 Uhr): Zweite Statusmitteilung* 3_2_ME_DM-Car, 3_3_ME_CarRentalApp
25-01-08	DATA-DRIVEN DECISION: ENGINEERING OF D3MAKERV1.0	3_2_ME_DM-Car, 3_3_ME_CarRentalApp
25-01-15	IDENTITY AND ACCESS MANAGEMENT *Lehrevaluation*	4_1_IAM_Fundamentals
25-01-22	AUTHORIZATION (Sänger) *Seminarvorträge*	
25-01-29	DECENTRALIZED IDENTITIES (Schneider) *Seminarvorträge*	
25-02-05	DEVOPS (Throner) *iCC-Gastbeitrag*	
25-02-11/12	*Mündliche Prüfungen*	*25-02-14 (bis 24 Uhr): Finale Statusmitteilung -> Abgabe der finalen Ausarbeitung*

This table is stored in C&M's document folder Mitglieder > 2-1.WASA_M2Go_Aktuell in the German-language PDF document Veranstaltungsorganisation. The folder can only be accessed by students who take part in the WASA lecture.

WASA Modules and Courses

(1) WASA Lecture
  (1) WASA1 for "Informatik / Wirtschaftsinformatik Bachelor" students each winter semester
  (2) WASA2 for "Informatik / Wirtschaftsinformatik / Informationswirtschaft Master" students each summer semester
(2) In parallel to the WASA lecture the practical course M2Go or a proseminar/seminar can optionally be attended
(3) Oral examination of the WASA lecture
  (1) 20 minutes in German
  (2) Last week of the lecture term

Lecture WASA1 (4 Credits)
Practical Course M2Go1 (5 Credits)
Proseminar (3 Credits)

Module Key Qualifications ("Schlüsselqualifikation")
Teamwork in the Area of WASA (2 SQ Credits)

WiSe

SuSe

Lecture WASA2 (4 Credits)
Practical Course M2Go2 (5 Credits)
Seminar (3 Credits)

The acronym WASA stands for "Web Applications and Service-oriented Architectures". The following courses are offered: (i) lecture courses WASA1 an WASA2 (ii) practical courses M2Go1 and M2Go2 associated to the lecture courses (iii) proseminar course and seminar (iv) key qualification course (germ. Schlüsselqualifikation SQ).

Remarks: In the Wirtschaftsinformatik study programme the name of the module is "Microservice-basierte Web-Anwendungen".
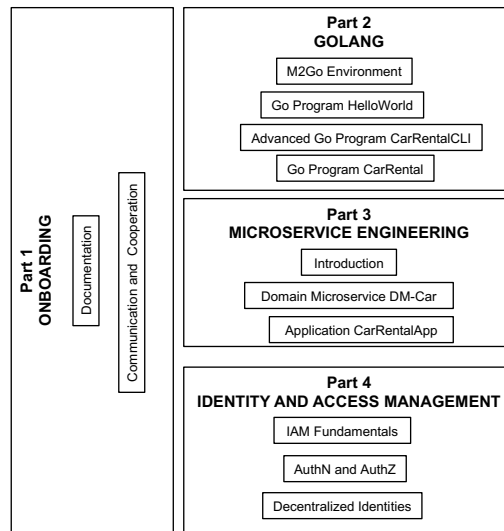
(1) The lecture courses WASA1 and WASA2 each comprise 2 semester hours. A student who attends one of the lectures acquires 4 credit points (German: Leistungspunkt).

(2) The practical course M2Go is closely linked with the lecture course. The practical course counts 5 credit points meaning a workload of 150 hours. A proseminar and seminar count 3 credit points meaning a workload of 90 hours. A student who wants to carry out a (pro)seminar in parallel to WASA_M2Go will have a workload of 360 hours (4+5+3 = 12 credits) in the semester.

(3) The examiners are Prof. Abeck and one of the C&M's PhDResearchers. Since the examination is in the last week of the lecture term, the students should have a good personal resource management in order to have enough time for the preparation of the examination.

SQ   Schlüsselqualifikation (Key Qualification)

M2Go Content and Practical Course

(1) The practical course can optionally be attended in parallel to the WASA lecture

(2) The exercises and challenges are to be worked out by each M2GoParticipant who documents the results in an individual English-language thesis document

(3) A participation in the practical course requires 150 h / 15 weeks = **10 working hours per week**

Part 1 ONBOARDING — Documentation — Communication and Cooperation

Part 2 GOLANG
- M2Go Environment
- Go Program HelloWorld
- Advanced Go Program CarRentalCLI
- Go Program CarRental

Part 3 MICROSERVICE ENGINEERING
- Introduction
- Domain Microservice DM-Car
- Application CarRentalApp

Part 4 IDENTITY AND ACCESS MANAGEMENT
- IAM Fundamentals
- AuthN and AuthZ
- Decentralized Identities

(1) The WASA lecture content and the M2Go practical course content are closely coupled.

(2) For each M2Go part, exercise documents are made available. M2GoParticipants create their own practical thesis document in which they document the solutions of the exercises. The practical thesis document should make clear that an M2GoParticipant has carried out each part of an exercise. Therefore, the solutions should be long enough, but not longer. The text should be expressed in the own words of the M2GoParticipant. If text is copied, the source must be referenced. M2GoParticipants can make contributions to the presentation of the M2Go content in the WASA lecture. An English-language LaTeX document is made available in Overleaf.

(3) The work starts immediately and the workload constantly arises every week. Therefore, students interested in participating in M2Go must make sure that they have enough time resources before they decide to do the M2Go practical course.

The practical course consists of the following four parts:

(ONBOARDING) All participants of the M2Go practical course must observe a few rules which exist at C&M to ease the cooperation between its members.

(GOLANG) This part provides a concise and practical introduction to the Go programming language. This includes the installation of the needed environment to code and run Go programs. Besides the well-known program "Hello World" some of the central Go language elements are investigated with the example of a more complex Go program CarRental and an advanced Go program CarRentalCLI.

(MICROSERVICE ENGINEERING) In the following part, Go is used as the programming language to implement microservices of the application CarRentalApp. C&M's Unified Microservice Engineering (UME) approach introduces specific analysis and design artifacts which build the basis for the Go-based microservice implementation.

(IDENTITY AND ACCESS MANAGEMENT) The last part is concerned with the relevant cross-cutting concern of Identity and Access Management (IAM). The access management includes the authentication and authorization of human and technical subjects who need access to the application BestRentalApp and their functions and data. A specific topic investigated in this part are IAM chatbots.
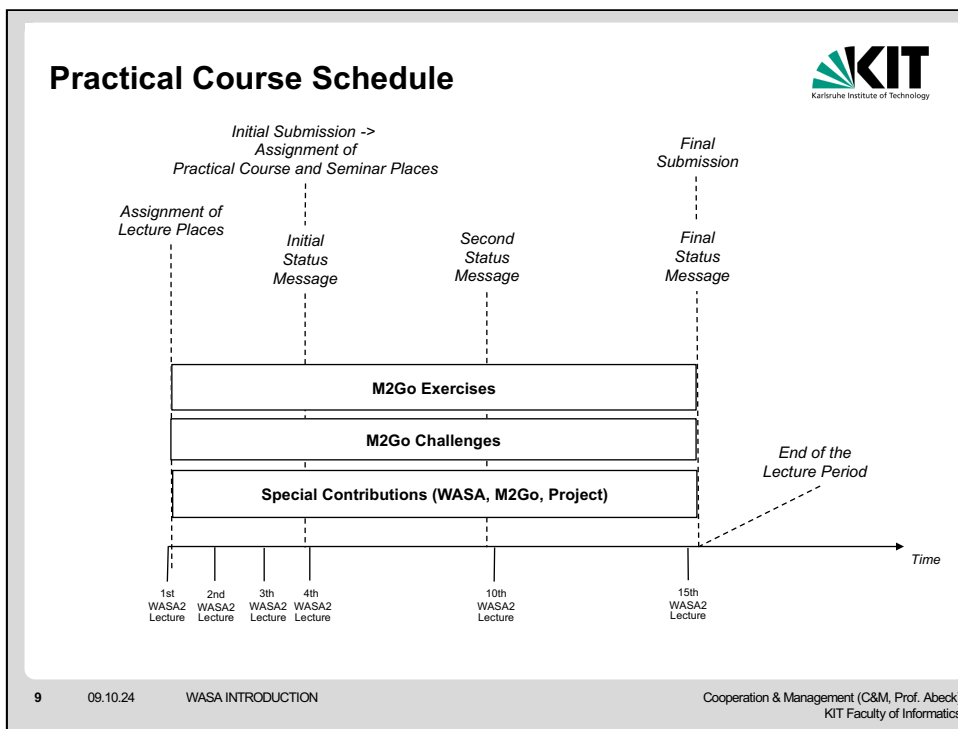
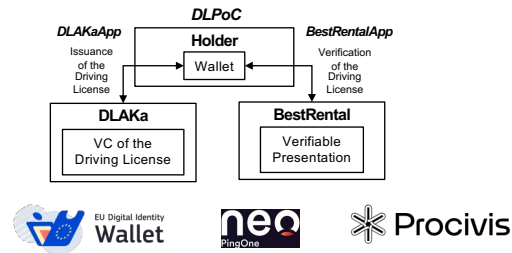The figure illustrates the schedule of the M2Go practical course.

(Initial Submission) The practical course starts with exercises of M2Go Part 1 ONBOARDING which makes sure that the M2Go candidate has successfully carried out the onboarding to the C&M environment and spends enough time for working out solutions of M2Go exercises.

(M2Go Challenges) The M2Go challenges are extensions of the M2Go exercises for which the M2GoParticipant works out individual and more complex solutions.
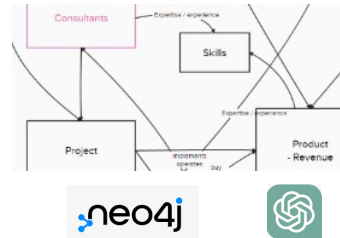
(Special Contributions) They go beyond the pure solutions of the M2Go exercises and challenges. Valuable special contributions are a prerequisite for a very good grade.

(1) Interoperability between the different DI systems and wallets is currently not given. The goal of the seminar is to investigate EUDI Wallet (EUDIW) from Digital Wallet Consortium (EWC) and to elaborate what needs to be done to achieve interoperability with an existing DI system. This is researched using a concrete DI system (such as Procivis One) and the existing DrivingLicensePoC.

(2) C&M develops a microservice-based application D3Maker in which several concepts and technologies from the area of Generative AI and graph databases are applied. In the seminar, a literature analysis of the chosen and further concepts and technologies for data-driven decision making should be carried out.

Requirements to be Fulfilled to Receive a Place in the Practical Course and/or Seminar

(1) Working hours to be carried out and documented in the time sheet when the initial status message on **12th November 2024** (about **19 days**) is sent
  (1) Practical course: at least **30 hours**
  (2) Seminar: at least **20 hours**
  (3) Practical course and seminar: at least **50 hours**

(2) Content of the initial submission
  (1) 1_ONBOARDING successfully completed
  (2) In addition
    (1) Practical course: 2_GOLANG partially carried out
    (2) Seminar: Introduction to the seminar topic
    (3) Practical course and seminar: 2_GOLANG including challenges (nearly) finished

(3) Further requirements to be fulfilled
  (1) M2GoParticipants make sure that their time sheets are in the green / violet area each time a status message is sent

On this page the requirements are summarized which must be fulfilled to successfully participate in the practical course and/or the seminar. A participation in the practical course and/or the seminar is only possible in combination with the WASA lecture. The WASA lecture can be attended without participating in the practical course or seminar.

(1) (2) Only when the initial submission is completely provided the student is assigned to the practical course and/or the seminar.

(2.1) ALL students who want to take part either in the practical course or the seminar or both should work through the exercises of 1_ONBOARDING. The M2Go document 1_ONBOARDING is made available
(i) on the C&M website: https://cm.tm.kit.edu/download/ONBOARDING.pdf
(ii) in the document storage: Mitglieder > 2-1.WASA_M2Go_Aktuell > 2.M2Go > 1_ONBOARDING.pdf
(2.2) A student who is interested in a seminar (and not the practical course) should NOT work on 2_GOLANG. The introduction into the seminar topic should take into account the literature cited in the seminar topic presentations (see previous pages).
(2.3) This means that a student who wants to carry out the seminar in parallel to the practical course, will concentrate in the initial phase on the practical course.

WASA-Bewerbungs-Mail

(1) **Latest until Thursday, 24.10.2024,10 am**:
Send a mail **(in German)** using your depseudonymized KIT mail address to **cm.research@lists.kit.edu** which contains the following information:

   (1) Matriculation number, KIT account (uxxxx)

   (2) Personal motivation

      (1) Text from the "Interessensbekundungs-Mail" can be reused

   (3) Interest in

      (1) Lecture only

      (2) Practical course

      (3) Practical course and seminar } **Favorite combinations**

      (4) Seminar only: which seminar topic

A student who participates in the WASA lecture and optionally in the practical/seminar course must have the necessary time resources to cope with the workload (lecture: 120 hours, practical course: 150 hours, seminar course: 90 hours).

(1.1) The uxxxx student name is needed for the invitation of the participants to the C&M GitLab. An invitation requires that the participant has once logged in the GitLab.

(1.2) The motivation and the experiences should be summarized in one or two paragraphs (about 3 to 6 sentences).

(1.3.1) It is possible only to apply for the WASA lecture.

(1.3.2) (1.3.3) In the case of the need to select participants, students who are interested in the practical course are preferred.

(1.3.3) Students who take the seminar in parallel to the practical course will be assigned the seminar topic after the INITIAL SUBMISSION.

(1.3.4) A detailed description of the seminar topic is made available to the selected seminar participants.

Next ToDo and Next WASA Lecture

(1) What to do next
- (1) Decide if you want to take part in the lecture and practical course and/or seminar
  - (1) If YES: see slide "WASA-Bewerbungs-Mail"
(2) Next WASA lecture
- (1) When: 30th October 2024, 9:45 am
- (2) Where: Building 50.34 (Informatikgebäude am Fasanengarten), Room SR 301
- (3) Next week no online transmission in parallel to the physical event is offered

This page summarizes what a student interested in WASA should do next.

(1.1.1) Please make sure to use your depseudonymized KIT mail address.

(2.3) This means that all students who received a place in the WASA lecture come to the room (2.2) at the given time (2.1).

13

=== ONBOARDING

M2Go ONBOARDING Exercises: DocumentStorage, ThesisDocument

Exercise DocumentStorage

1. **Connect to the Document Storage**

   Connect to the documentation storage to your machine.

2. **Adjust the File Names**

   Locate your own directory and replace the substring in the names of the PDF file and the PowerPoint figures file by your surname (written in lower letters).

Exercise ThesisDocument

1. **Import Template to Overleaf**

   Open Overleaf and log in with your KIT account. Upload the LaTeX template by using the provided ZIP file. Afterwards, ensure that the main document is set to the correct file and compile the document.

2. **Rename Overleaf Project**

   Rename the project name by replacing the "n" behind m2go with 1 or 2 (depends on M2Go1 or M2Go2) and the string "name" with your surname. As a result, the document can be downloaded according to the naming specifications.

3. **Adjust Title Page**

   Open the file named 0-1_titlepage.tex and fill out the form. Finally, upload the resulting PDF document to your working folder.
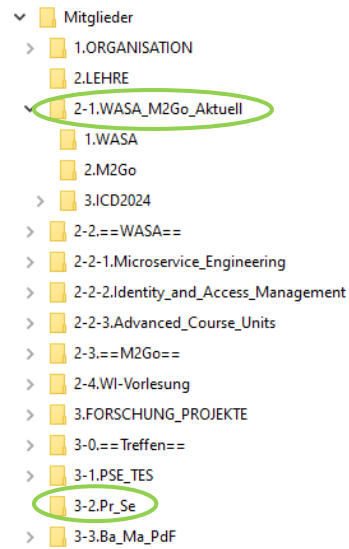
(1) Web link for Windows users: \\sccfs.scc.kit.edu\OE\TM\VR\Mitglieder

Web link for MAC/Linux users: smb://sccfs.scc.kit.edu/OE/TM/VR/Mitglieder

M2Go ONBOARDING Exercises: TimeSheet, CMGitLabSetup

Exercise TimeSheet

1. **Create a New Work Element**
   Open the page for recording the work on the SharePoint and add a new element by clicking on "Neues Element". Fill out the form as follows: Art: Praktikumsarbeit, Anfangsdatum: see Organisationsdokument Enddatum: see Organisationsdokument Stunden: 150.

2. **Add First Hours**
   Add a new time element which covers the organizational work and fulfilled exercises. Choose the work type "Produktive Arbeit" and add the invested hours. The work type "Besprechungen" should be chosen in the case of meetings. No further work type (e.g., C&M-bezogene Arbeit) is relevant for M2Go. Only capture the working hours related to the M2Go practical course and not those related to the WASA lecture.

Exercise CMGitLabSetup

1. **Create GitLab Account**
   Login to KIT GitLab and register your account.

2. **Add a Profile Picture (Optional but Recommended)**
   Edit your GitLab profile and add your profile picture as avatar.

3. **Create SSH Key Pair**
   Generate an SSH key pair with your KIT mail address.

4. **Add SSH Public Key to GitLab Account**
   Follow the GitLab guideline to add the SSH public key to your GitLab account.

M2Go ONBOARDING Exercises:
OnboardingCompletion, Status Messages

Exercise OnboardingCompletion

1. **Describe the M2Go Coordination Concept Applied at C&M**

   Which are the roles (e.g., M2GoCandidate) and the systems (e.g., GitLab) and artifacts (e.g., time sheet) used in M2Go? Include a PowerPoint figure by which you illustrate the M2Go coordination concept. Make sure that the figure is part of the file pr_<your_surname>_fig.pptx (practical course) or se_<your_surname>_fig.pptx (seminar or proseminar) which is stored in your working folder.

2. **Investigate Current Research Projects Carried Out at C&M**

   In the rear part of the document Veranstaltungsorganisation (stored in Mitglieder > 2-1.WASA_M2Go_Aktuell) the research projects carried out by the C&M SeniorStudents who currently write their Bachelor/Master Thesis (stored in subfolders of Mitglieder > 3-3.Ba_Ma_PdF) are presented. Choose two research projects in which you would like to work on. Describe the core of the project and outline competences and contributions you could bring to the project.

Exercise StatusMessages

1. **Send Status Message**

   Add your thesis document to your working folder. Ensure, that the naming convention is met and update the time sheet. Send the status message to cm.research@lists.kit.edu on the corresponding dates no later than 12 am.

2. **Provide Initial Submission**

   The initial submission is part of the first status message. Copy the PDF file and the PowerPoint figure file stored in your working folder into the subfolder "1.Initial_Submission" and add the suffix "_initial" to both files. Note that a place to the practical course or the seminar is only assigned to students whose initial submissions fulfill the defined requirements.

Michael Schneider, Sergiu Panainte, Niklas Weber (?):
Decentralized Identities (DI)

(1) Main topic

(2) Preliminary work

(3) Main challenges

19

Anh Kha Nguyen, Fabian Lübbe:
Data-Driven Decision (D3) Making

(1) Support of an IT security company in answering strategic questions
  - (1) Examples: Cross-selling potentials, future partnerships, training of consultants
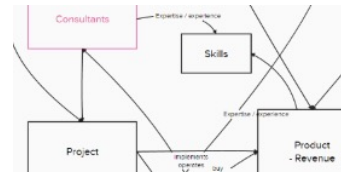
(2) Solution: DataDrivenDecisionMaker (D3Maker)
  - (1) gRPC-based microservice architecture
  - (2) GenAI technologies and chatbots
  - (3) Graph-database Neo4j to store the data

(3) In this project, D3Maker should be extended to support further strategic questions

Further sources related to Data-Driven Decision Making:

[CM-W-DAT] Cooperation & Management: DATA-DRIVEN DECISION. WASA Course Un
\\sccfs.scc.kit.edu\OE\TM\VR\Mitglieder\2-2-1.Microservice_Engineering

[Ko24] Simon Korte: Engineering and Strategic Integration of a Chatbot Platform, Masterarbe
\\sccfs.scc.kit.edu\OE\TM\VR\Mitglieder\3-3.Ba_Ma_PdF\Ma_Korte

[Ng25] Anh Kha Nguyen: Support of Strategic Decision Making in the Field of IT Security, Masterarbe
\\sccfs.scc.kit.edu\OE\TM\VR\Mitglieder\3-3.Ba_Ma_PdF\Ma_Nguyen