This 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 compact summary of the concepts covered by WASA1 is provided. In WASA2, Identity and Access Management (IAM) as an advanced topic is 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 and authorization functionality based on the current Internet standards OpenID Connect and OAuth2. 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
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.

CI/CD  Continuous Integration / Continuous Deployment
DDD  Domain-Driven Design
D3  Data-Driven Decision
DevOps  Development and Operations
IAM  Identity and Access Management
LLM  Large Language Model
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 implements the domain microservices which mainly are 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
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.

https://gitlab.kit.edu/kit/cm/teaching/carrentalapp/carrentalappv1
The Unified Microservice Engineering (UME) approach unifies two former approaches developed by C&M, called CEng and MuleEng. UME consists of one process, the structured development process consisting of the well-known phases analysis, design, implementation and test, and deployment and operations. In the UME approach, the domain-driven aspects are not necessarily part of the engineering process (i.e., they are optional).

(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.

(Application Term List) Application-specific terms are not introduced as a ubiquitous language. Ubiquitous languages are only provided by domains.

(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

<table>
<thead>
<tr>
<th>Datum</th>
<th>WASA KICK-OFF</th>
<th>Praktikum</th>
</tr>
</thead>
<tbody>
<tr>
<td>24-10-23</td>
<td>WASA KICK-OFF</td>
<td>1. ONBOARDING</td>
</tr>
<tr>
<td>24-10-30</td>
<td>Veranstaltungsorganisation Verantwortung der Projektthemen WIRKLEISTUNGSDIENST ENGINEERING (WE) OVERVIEW</td>
<td></td>
</tr>
<tr>
<td>24-11-05</td>
<td>Anmeldung zur mündlichen Prüfung der WASA-Mitglieder</td>
<td>24-11-09 (bis 12 Uhr): Staatsermittlung I = INITIAL SUBMISSION</td>
</tr>
<tr>
<td>24-11-08</td>
<td>TOOL ENVIRONMENT</td>
<td>2. GOLANG</td>
</tr>
<tr>
<td>24-11-18</td>
<td>ANALYSIS DESIGN</td>
<td>2. GOLANG</td>
</tr>
<tr>
<td>24-11-22</td>
<td>DESIGN</td>
<td>2. GOLANG, 3.2_MG_Di-Car</td>
</tr>
<tr>
<td>24-11-28</td>
<td>DESIGN</td>
<td>2. GOLANG, 3.2_MG_Di-Car</td>
</tr>
<tr>
<td>24-12-04</td>
<td>IMPLEMENTATION DEPLOYMENT</td>
<td>3.2_MG_Di-Car</td>
</tr>
<tr>
<td>24-12-11</td>
<td>DEPLOYMENT</td>
<td>3.3_MG_CoRienhapp</td>
</tr>
<tr>
<td>24-12-18</td>
<td>DATA-DRIVEN-DECISION INTRODUCTION</td>
<td>24-12-17 (bis 12 Uhr): Staatsermittlung II = Staatsermittlung II</td>
</tr>
<tr>
<td>24-01-08</td>
<td>DATA-DRIVEN-DECISION ENGINEERING OF軟件 / HARDWARE</td>
<td>3.3_MG_CoRienhapp</td>
</tr>
<tr>
<td>25-01-15</td>
<td>ENTITY AND ACCESS MANAGEMENT</td>
<td>4.1_MG_Basiskiekreta</td>
</tr>
<tr>
<td>25-01-22</td>
<td>AUTHENTICATION (Bäreg)</td>
<td></td>
</tr>
<tr>
<td>25-02-28</td>
<td>DECENTRALIZED IDENTITIES (Schnieder)</td>
<td></td>
</tr>
<tr>
<td>25-02-05</td>
<td>D2C/PS (Theorie)</td>
<td></td>
</tr>
<tr>
<td>25-02-11/12</td>
<td>Vorlesungs-Rücksichten</td>
<td>20-02-14 (bis 12 Uhr): Staatsermittlung III = Regeln der Praktikum Ausarbeitung</td>
</tr>
</tbody>
</table>

Store in C&M's document folder
Mitglieder > 2-1.WASA_M2Go_Aktuell
in the German-language PDF document
veranstaltungsorganisation_sose24

Remark: The folder can only be accessed by students who take part in the WASA lecture.
The acronym WASA stands for "Web Applications and Service-oriented Architectures". The following courses are offered: (i) lecture courses WASA1 and WASA2 (ii) practical courses M2Go1 and M2Go2 associated to the lecture courses (iii) proseminal 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 proseminal 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.

(4) 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.
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 the M2Go practical course:
   YES or NO

(4) Interest in one or more proseminar topics
   (1) If YES: which topic(s)

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) (1.4) It is possible to apply only for the WASA lecture.
If at least one of answer is YES, a personal folder in the C&M document storage and a personal repository in the C&M GitLab will be made available.
The practical course can optionally be attended in parallel to the WASA lecture.

The exercises and challenges are to be worked out by each M2Go Participant who documents the results in an individual English-language thesis document.

A participation in the practical course requires 150 h / 15 weeks = 10 working hours per week.

---

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

2. For each M2Go part, exercise documents are made available. M2Go Participants create their own practical thesis document in which they document the solutions of the exercises. The practical thesis document should make clear that an M2Go Participant 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 M2Go Participant. If text is copied, the source must be referenced. M2Go Participants 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 M2Go exercises of Part 1 which makes sure that the WASA2Participant 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 (i.e., 1,0 or 1,3).
(1) C&M currently 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.

(2) In addition to authorizing requests from human users, requests between microservices must also be authorized. This is especially necessary when aiming for a zero-trust architecture. In this seminar, the existing literature on service-to-service authorization will be reviewed and the concepts applied to the CarRentalApp.

(3) Currently, 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.
Necessary Actions to Take Part in M2Go

(1) Student interested in the M2Go practical course work in the next 20 days on an initial submission
   (1) At least 20 hours are documented on the time sheet when the students deliver their initial submission with their first status mail on 5th November 2024
   (2) M2Go Part 1 ONBOARDING is successfully completed
   (2) Exercises and challenge of the M2Go Part 2 GOLANG are partially carried out

(2) Further requirements to be fulfilled
   (1) M2GoParticipants makes sure that their time sheets are in the green / violet area when the status message is sent

On this page the requirements are summarized which must be fulfilled to successfully participate in the M2Go practical course. A participation in the practical course M2Go2 is only possible in combination with the WASA2 lecture. The WASA lecture can be attended without participating in the M2Go practical course.

(2.1.1) The M2Go document 1_ONBOARDING is made available at
   (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
Next ToDo and Next WASA Lecture

(1) What to do next
   (1) Decide if you want to take part in the lecture and/or practical course or seminar
       (1) If YES: see slide "WASA-Bewerbungs-Mail"

(2) Next WASA lecture
   (1) When: 30th October 2024
   (2) Where: Building 50.34 (Informatikgebäude am Fasanengarten), Room SR 301
   (3) No online transmission in parallel to the physical event is offered