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 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, 17th April 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
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) Usage of chatbot systems based on Large Language Models (LLM) to organize IAM knowledge.
The Unified Microservice Engineering (UME) approach unifies two former approaches developed by C&M, called CMEng 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
(1) The cars to be rented are organized in fleets. This leads to two different functionality parts to be provided by CarRentalAppV2.0, the management of the fleets and the management of the rentals.

(1.1) Initially only one fleet identified by its location is supported. For each fleet, one fleet manager is responsible. In CarRentalAppV2.0, no functionality to coordinate the fleets (e.g., creation of a new fleet, change of a fleet manager) exists.

(1.2) The functionality of AM-RentalManagement was implemented in CarRentalAppV1.0 (CarRentalAppV1.1 added the functionality of customer registration and deregistration to Version V1.0).

(2) The changes of the fleet especially concern the addition, replacement, and deletion of cars to and from the fleet.

(2.1) This means that AM-RentalManagementV2.0 does not need to make requests to AM-FleetManagementV1.0 to determine all available cars which is necessary to carry out a rental.

(«ui» UI-CarRentalV2.0) The user interacts with the UI-CarRentalAppV2.0 in order to rent cars and perform fleet management.

(«application microservice» AM-RentalManagementV2.0) The application logic provides the application-specific functionality in order to allow customers the rental of cars.

(«application microservice» AM-FleetManagementV1.0) The application logic provides the application-specific functionality in order to allow cars in a location to be organized in a fleet and to allow a fleet manager to manage his fleet.

(«domain microservice» DM-CarV2.0) DM-Car concerns the application agnostic functionality related to a car.

(«external system» ES-ConnectedCars) The external system from which DM-CarV2.0 retrieves its information about cars. It provides basic car information such as brand and model.

(«database» DB-Fleets) The fleet information is persisted in this database.

(«database» DB-Rentals) The rental information is persisted in this database.
(1) For the demonstration of IAM aspects, a concrete organization called BestRental and a microservice-based application, called BestRentalApp, are introduced. Two roles, Fleet Manager and Customer are distinguished and two persons, Fred and Alice, are introduced.

(1.1) CarRentalAppV2.0 [CM-G-CRAV2.0] provides the basic functionality with respect to rental management and fleet management which is needed by BestRental. Therefore, CarRentalAppV2.0 builds the technical basis for the development of BestRentalAppV1.1[CM-G-BRAV1.1].

(2) CarRentalAppV2.0 corresponds to BestRentalAppV1.0 which only virtually exists.

(3.1) CarRentalAppV2.0 will be extended by an AuthN (Authentication) solution which leads to BestRentalAppV1.1 [CM-G-BRAV1.1].

(3.2) BestRentalAppV1.1 will be extended by an Internalized AuthoriZation (I-AuthZ) solution which leads to BestRentalAppV1.2 [CM-G-BRAV1.2].

(3.3) BestRentalAppV1.1 will be extended by an Externalized AuthoriZation (E-AuthZ) solution which leads to BestRentalAppV2.0 [CM-G-BRAV2.0].

---

**AuthN and AuthZ in Microservice-Based Applications**

(1) Car rental company BestRental uses BestRentalApp for its business.

(2) CarRentalAppV2.0 builds the starting point for the development of BestRentalAppV1.1.

(3) IAM extensions of BestRentalApp:
   - (1) AuthN (V1.1)
   - (2) Internalized AuthZ (V1.2)
   - (3) Externalized AuthZ (V2.0)

---

**Cooperation & Management: CarRentalAppV2.0**

https://gitlab.kit.edu/kit/cm/teaching/carrentalapp/carrentalappv2.0

**Cooperation & Management: BestRentalAppV1.1**

https://gitlab.kit.edu/kit/cm/teaching/bestrentalapp/bestrentalappv1.1

**Cooperation & Management: BestRentalAppV1.2**

https://gitlab.kit.edu/kit/cm/teaching/bestrentalapp/bestrentalappv1.2

**Cooperation & Management: BestRentalAppV2.0**

https://gitlab.kit.edu/kit/cm/teaching/bestrentalapp/bestrentalappv2.0
The proof of concept concerns the rental of a car at BestRental by the customer Alice. The approach of decentralized identities is used to proof to BestRental that Alice has a valid driving license. The scenario is adapted from [Mic-Dec] and [iC-LLD].

(1.1) Alice uses her wallet application to carry out the request to DLAKa, the Driving License Authority at Karlsruhe (Ka). She is served by Bob, who works as a DLA clerk at DLAKa. A signed Verifiable Credential (VC) is issued by DLAKa and stored in the digital wallet application, and which attests that Alice owns a valid driving license.

(1.2) Alice presents the VC on the BestRental website. The transaction is logged in Alice's wallet application.

(2) The claims contains attributes which specify (i) the type(s) of mobile vehicle, Alice is allowed to drive, (ii) the year when she passed her driving test.

(3) The VCs issued by DLA are digitally signed with the issuer's private key and the Verified Presentations VP presented by Alice (i.e., the user) to BestRental are digitally signed with Alice's private key. Thus, BestRental needs both the issuer's and Alice's public keys which are made available by a verifiable data registry. The trust system provides the public keys of the involved entities in a decentralized manner.

DLA  Driving License Authority
DLAKa DrivingLicenseAuthorityKarlsruhe
DI  Decentralized Identity
PKI  Public Key Infrastructure
PoC  Proof of Concept
VC  Verifiable Credential
VP  Verifiable Presentation

[iC-LL] iC Consult: Lunch & Learn: Decentralized Identities, restricted access. https://ic-consult.atlassian.net/wiki/spaces/EV/pages/3839033345/Lunch%3BLearn%3DDecentralized%3DIdentities

IAM Coding Day (ICD) 2024

1. ICD is a 24 hour hackaton offered each summer semester by KIT and iC Consult to WASA students.
2. Mandatory part of the M2Go2 practical course.
3. This year's ICD topic: Decentralized Identities (DI).
5. Challenges
   1. Exchange of the DI Product.
   2. Introduction of an Employee Card at BestRental.
   3. Deployment in a Kubernetes cluster.

(1) iC Consult is a leading consulting company in the area of Identity and Access Management (IAM). The IAM Coding Day (ICD) is an event which is an important element in the cooperation with the KIT research group Cooperation & Management (C&M).

(1.1) The knowledge transferred by the WASA lecture and the M2Go practical course is applied by the students to solve the ICD challenges.

(2) The concept of Decentralized Identities (DI) is conceptually and practically introduced in WASA and M2Go.

(3) The proposal is to extend C&M's driving license business case by two or three challenges.

(4) The challenges should cover aspects which are interesting for iCC.

(4.1) Challenges should be prepared in current Ba/Ma theses (esp. Ba Maurer).
### Lecture Plan

<table>
<thead>
<tr>
<th>Date</th>
<th>Title</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>24.04.17</td>
<td>WASA Kick-off</td>
<td></td>
</tr>
<tr>
<td>24.04.24</td>
<td>Microservice Engineering Overview</td>
<td>1.2_5WSK01/KS01</td>
</tr>
<tr>
<td>24.05.24</td>
<td>Freitag</td>
<td>3.1_ME Introduction</td>
</tr>
<tr>
<td></td>
<td>Verantwortung: Prof. Abeck</td>
<td></td>
</tr>
<tr>
<td>24.05.28</td>
<td>W2</td>
<td></td>
</tr>
<tr>
<td>24.06.05</td>
<td>Mitglieder &gt; 2-1. WASA_M2Go_Aktuell</td>
<td></td>
</tr>
<tr>
<td>24.06.06</td>
<td>in the German-language PDF document</td>
<td></td>
</tr>
<tr>
<td>24.06.07</td>
<td>Veranstaltungsorganisation_sose24</td>
<td></td>
</tr>
<tr>
<td>24.06.18</td>
<td>Remark: The lecture plan can only be accessed by students who take part in the WASA lecture.</td>
<td></td>
</tr>
</tbody>
</table>

*Remark: The lecture plan 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) 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 PhD Researchers. 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)
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) It is possible to apply only for the WASA lecture.

If the 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.

(1.3.1) This question can be ignored in the case of the winter semester in which WASA1 is offered.
(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:

(C&M Org) 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 M2Go2 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).
Necessary Actions to Take Part in M2Go

(1) Student interested in the M2Go2 practical course work in the next 20 days on an initial submission

(1) At least 30 hours are documented on the time sheet when the students deliver their initial submission with their first status mail on 7th May 2024

(1) M2Go Part 1 ONBOARDING is successfully completed

(2) Exercises and challenge of the M2Go Part 3.1 "MICROSERVICE ENGINEERING - Introduction" are (at least 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 monthly status message is sent

(2) M2GoParticipants must have time on the days July 5th and 6th to take part in the IAM Coding Day 2024 which is a mandatory part of the M2Go practical course.

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

If the 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.

(1.3.1) This question can be ignored in the case of the winter semester in which WASA1 is offered.
Next WASA Lecture

(1) When: 24th April 2024

(2) Where
   (1) Building 50.34 (Informatikgebäude am Fasanengarten)
   (2) Room SR 301

(3) No online transmission in parallel to the physical event is offered