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). Since the lecture WASA is closely linked with the practical course M2Go, the lecture and the practical course can only be taken in combination (WASA_M2Go). Since the number of WASA_M2Go places is limited, interested students must apply for a place.

WASA1_M2Go1 (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. A practical course, called Microservice2Go (M2Go), is offered in combination with the WASA1 lecture in which the UME approach is practically applied with the example of a car rental application.

WASA2_M2Go2 (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. IAM is a highly relevant part of the digitization strategy of each organization. In the lecture, leading IAM solutions and products are introduced to illustrate how the IAM challenges are solved in IT practice. A practical course, called Microservice2Go (M2Go), is offered in combination with the WASA2 lecture in which an existing car rental application is extended by IAM functionality dealing with different topics, such as OIDC/OAuth-based authentication and authorization, decentralized identities, and authorization policies.

The WASA_M2Go kick-off lecture will take place

==== on Wednesday, 25th October 2023 at 9:45 am
==== in the Building 50.34 (Informatikgebäude am Fasanengarten), Room SR301

Each student who wants to take part in the WASA kickoff lecture must

==== send an email (in German) to cm.research@lists.kit.edu

To apply for one of the restricted WASA_M2Go 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
KIT Karlsruhe Institute of Technology
M2Go Microservice2Go
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 make the user the owner and controller of their identity data.
A microservice architecture is located on the application plane as it is introduced in the course unit MICROSERVICE ENGINEERING. While the software architecture is described by the logical layers specified by the DDD pattern LAYERED ARCHITECTURE, the system architecture introduces several subsystems (domain microservices, application microservices, API gateway). Two types of application programming interfaces (domain microservice API, application microservice API) are separating the logical layers on the software architecture side and the microservices on the 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 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.
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/carrentalapp
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.

<table>
<thead>
<tr>
<th>UI</th>
<th>User Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>UME</td>
<td>Unified Microservice Engineering</td>
</tr>
</tbody>
</table>
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) prosemnar 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.

(3) A prosemnar 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.

SQ Schlüsselqualifikation (Key Qualification)
Remark: The lecture plan can only be accessed by students who will be accepted as WASA_M2Go participants.
(1) Besides the traditional (one-way) lecture part, each lecture event additionally consists of an interactive part which is shaped in a more dynamic way.

(2.1) For each lecture event, the agenda is made available in [CM-G-WAS]. The content of the markdown file is in German since this is the language spoken in the WASA lecture.

https://gitlab.kit.edu/kit/cm/teaching/wasam2goagenda
The practical course M2Go is carried out in parallel to the WASA lecture.

The M2Go content and the included exercises are presented in the WASA lecture.

The exercises are to be worked out by each M2GoParticipant who documents the results in an individual English-language thesis document.

A participation in WASA_M2Go requires about 2 work days per week.

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

(2) M2GoParticipants can make contributions to the presentation of the M2Go content in the WASA lecture.

(2.1) An English-language LaTeX document is made available in Overleaf.

(3) The work starts immediately and the workload is constantly arises every week. Therefore, students interested in participating in M2Go must make sure that they have enough time resources before they decide to apply for a WASA_M2Go place.
(1) One of the goals pursued by chatbot systems is to support the transfer of knowledge. Based on the found literature, scenarios of the usage of such systems in the area of education should be investigated in this seminar thesis. Proposals for the support of C&M's teaching and research activities C&M's should be made.

(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) End-to-end testing ensures that the acceptance criteria of an application are met. Various approaches and tools found in the literature which address the testing of distributed systems should be examined and discussed. The result of the seminar should be a discussion/proposal on how to advance end-to-end testing in C&M’s Unified Microservice Engineering (UME) approach and using the CarRentalApp covered by WASA_M2Go.
Next Steps

(1) Personal decision if you want to participate in WASA_M2Go
(2) If YES
   (1) Latest until Thursday, 26.10.2023,10 am: Send an email (in German) with your depseudonymized KIT mail address to cm.research@lists.kit.edu with the following information:
      (1) KIT account (uxxxx), matriculation number
      (2) Personal motivation and experiences in this area (e.g., IT project experiences)
      (3) Optional: Interest in a (pro)seminar topic in addition to the lecture WASA and the practical course M2Go
(3) The answer to your email will contain all relevant information (esp. access to document storage and C&M GitLab) to start WASA_M2Go

(1) It is absolutely important that a student who participates in the WASA lecture and practical/seminar course has the necessary resources to cope with the workload (lecture: 120 hours + practical course: 150 hours = 270 hours in total).

(2.1) Check on the page of the KIT Steinbuch Computing Centre if your email is already depseudonymized.
(2.1.1) The uxxxx student name is needed for the invitation of the M2Go participants to the C&M GitLab. An invitation requires that the M2Go participant has once logged in the GitLab.
(2.1.2) The motivation and the experiences should be summarized in at least one or two paragraphs.

(3) This email will be sent by the WASA_M2Go supervisors.
The current WASA_M2Go material is stored on the C&M Data Storage in the following folder: \sccfs.scc.kit.edu\OE\TM\VR\Mitglieder\2-1.WASA_M2Go_Aktuell
C&M Document Storage

(1) Accessible via the web link
\sccfs.scc.kit.edu\OE\TM\VR\Mitglieder

(2) External access (i.e., outside the KIT network) via OpenVPN

(2) Important folders and documents stored on the C&M Document Storage

WASA course units and M2Go practical course material in:
WASA_M2Go_Aktuell

Personal working folder to store the M2Go thesis document in:
3-2.Pr_Se

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

(2-1 WASA_M2Go_Aktuell) This is the most important folder which is available via \sccfs.scc.kit.edu\OE\TM\VR\Mitglieder\2-1.WASA_M2Go_Aktuell
Activation of the Name-Related E-Mail Address (Depseudonymization)

Can be carried out via Shibboleth (https://my.scc.kit.edu/shib/pseudonymisierung.php)

This function can be found in the Studierendeportal by clicking on "Meine Benutzerdaten" > "De-/Pseudonymisierung" and accepting "Ich stimme der Sichtbarkeit meiner namensbezogenen Daten zu". The name-related E-Mail-Adresse <prenom><surname>@student.kit.edu" exists additionally to the "uxxx@student.kit.edu" email address.