This WASA course unit [CM-W-INT] describes the content and the organization of the lecture and practical course "Web Applications and Service-oriented Architecture" (WASA) provided by the research group Cooperation & Management (C&M, Prof. Abeck). Current concepts of software development and architectures (including Behavior-Driven Development, Domain-Driven Design, Microservices, RESTful Webservices, 12 Factor App, CI/CD Build Pipelines, DevOps, Container-virtualized Infrastructures) as well as related standards and technologies (including HTTP, Java, JavaScript/TypeScript, Angular, Spring, GitLab-CI, Docker, Kubernetes, Prometheus) are introduced which are needed to develop advanced (i.e. microservice-based, IoT aware, cloud-native, mobile) web applications. The web applications stem from the connected car domain which includes concepts from the domain of Internet of Things. Since the concepts presented in the lecture must be practically applied to really understand them the WASA lecture is only offered in a combination with the WASA practical and/or seminar course.

IMPORTANT: In the winter semester 2021/22, the lecture and practical/seminar course are organized as an online event. This format is called WASAOnline.

The WASAOnline kickoff lecture will take place on Wednesday, 20th October 2021 at 10:00 am.

Each student who wants to take part in WASAOnline and in the kickoff lecture should send an email to

cm.research@lists.kit.edu

to receive the web link to the web session which is made available for his/her personal use. Please use your depseudonymized KIT student email address (see https://my.scc.kit.edu/shib/pseudonymisierung.php for further information). Thank you!

The lecture material is made available in English. During the online lecture, the content is presented and discussed in German. The oral examination is conducted exclusively in German. Each student writes his/her practical/seminar thesis in English. Thesis templates are made available in LaTeX.

C&M Cooperation & Management
KIT Karlsruhe Institute of Technology
WASA Web Applications and Service-oriented Architectures

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

(Connected Car, Healthcare, Environmental) In these business domains application based on the concept of domain modeling and microservice architectures are developed. Relevant concepts applied in the development process include Behavior-Driven Development (BDD) and Domain-Driven Design (DDD). All microservice-based applications developed by C&M include aspects of Internet of Things (IoT). An important standard on which our implementations are based on is the SensorThings API from the Open Geospatial Consortium.

(DevOps, Identity and Access Management) DevOps concerns the continuous integration (CI) and continuous deployment (CD) of microservice-based software systems into a container-virtualized (Docker/Kubernetes-based) cloud infrastructure. Identity and Access Management (IAM) is a highly relevant crosscutting concern appearing in every web application. One of the projects is concerned with the provisioning of IAM-as-a-service (IAMaaS) provided in a cloud infrastructure.

BDD       Behavior-Driven Development
CI/CD     Continuous Integration / Continuous Deployment
DDD       Domain-Driven Design
IAM       Identity and Access Management
IAMaaS    IAM-as-a-Service
IOSB      Fraunhofer Institute of Optronics, System Technologies, and Image Exploitation (Optronik, Systemtechnik und Bildauswertung)
IoT       Internet of Things
DevOps    Development and Operations
The software development process applied by C&M combines the concepts of Behavior-Driven Development (BDD) and Domain-Driven Design (DDD). Both concepts provide complementary contributions to the layered microservice architecture as the figure illustrates.

(1) An implemented feature can be seen as a deployable increment of the software system. (Feature 1, Feature 2, ...) The ordering of the features in the figure implies that the first feature should cover the core functionality of the software system.

(2) The domain model makes sure that the static and dynamic domain knowledge is consistently used by each feature. This ensures that the features build a consistent whole although each feature is developed and deployed independently from other features.

BDD Behavior-Driven Development
DDD Domain-Driven Design
(PredictiveCarMaintenance, ClinicsAssetManagement, EnvironmentalMeasurementDashboard) These are the microservice-based applications which are developed by C&M in cooperation with his partner institutes.

(IAM Service) The Gherkin features include Identity and Access Management (IAM) requirements, such as authentication of the user of the PCM application. The IAM service Auth0 is used to implement these requirements.

(PCM Microservice, The Twelve-Factor App) The application itself is based on the microservice architecture. The development of the PCM microservice should follow the Twelve-Factor App requirements resulting in a cloud-native application.

(Message Broker, RabbitMQ) The communication between the Backend-For-Frontend (BFF) and microservices is based on the exchange of events. The event bus is provided by a message broker tool named RabbitMQ.

(IoT, IoT Service, SensorThings, Web of Things) IoT standards, such as SensorThings API from OGC and Web of Things from W3C, are taken into account in the development of the IoT-based application.

(API Security) The APIs must be protected against unauthorized access. The IAM service provides the authentication and authorization information needed to take the access decision at the API.

(DevOps, Build Pipelines, Kubernetes Cluster, Prometheus Monitoring) The goal is to develop a build pipeline based on GitLab-CI by which the PCM application is continuously integrated (CI) and deployed (CD) on a Kubernetes cluster. The microservice-based applications provide metrics which are monitored tools such as Prometheus.
The table describes the planned sequence of course units presented during the lecture.

(Content) All titles printed in upper case are course units that are the basis for the oral examination carried out at the end of the semester. After the course unit was held a range of page numbers are added. This indicates which pages must be prepared by the student for the oral examination.

COURSE UNIT (P. X – P. Y) All course units written in upper case letter are primary examination material. If page ranges are indicated only these pages are treated in the oral exam. No page range means that questions of the complete course unit can be asked in the oral exam.
The acronym WASA stands for "Web Applications and Service-oriented Architectures". Four different types of WASA courses are offered: (i) lecture courses WASA1 and WASA2 (ii) practical courses WASA1 and WASA2 associated to the lecture courses (iii) proseminar course associated to WASA1 lecture course and seminar associated to WASA2 lecture course (iv) key qualification course (germ. Schlüsselqualifikation SQ).

(1) The lecture courses WASA1 and WASA2 each comprise 2 semester hours. A student who attends one of the lectures acquires 4 credit points (germ. Leistungspunkt).

(2) The practical course runs in parallel with the lecture course and counts 5 credit points meaning a workload of 150 hours. The capacity of students C&M can offer the practical course depends on the current projects carried out in the research group.

Hint: In the Wirtschaftsinformatik study programme the name of the module is "Microservice-basierte Web-Anwendungen".

(3) The examiners are Prof. Abeck and one of the C&M's PhD Researchers. Since the examination is in 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)
On this page the specifics of the WASA practical/seminar courses which are offered in parallel to the WASA lecture are described.

(1) A project consists of about 4 to 6 practical/seminar students.
   (1.1) A SeniorStudent is a student who is writing his/her bachelor thesis or master thesis at C&M.
   (1.2) The meetings take place at a defined time which is fixed at the beginning of the semester.

(2) The topics dynamically evolve from the work done by the SeniorStudent in their bachelor/master thesis.
   (2.1) The JuniorStudent should actively participate in the discussion and make own proposals how the topic should be treated.
   (2.2) Reviews are an integral part of the work in the project team.
   (2.3) This means that the practical/seminar work has a dynamic characteristic.
   (2.4) There is a high flexibility and liberty concerning the focal points of the practical/seminar work.
The project team agenda is a markdown document by which the work of all project team members is coordinated.

(1) The project team leader takes care that all team members contribute to the agenda of each project team meeting.
(1.1) For each project team a subgroup exists in the C&M GitLab. In the README file of the repository "Projektteamtreffen" the agendas of the project team meetings are collaboratively worked out.
(1.2) The project team meets for one hour every week. The day and time are defined by the project team members.
(1.3) That is why the agenda is worked out in German.

(2) There are templates available for the different variants in which a JuniorStudent can write his/her practical/seminar thesis.

(3) C&M-TEAMARBEIT is a document [CM-CMT] which describes how the members of the research team C&M efficiently work together.

(Screen dump on the right hand side) This is an excerpt of the GitLab document "Projektteamtreffen" which provides an agenda draft of the first two project team meetings.

Each project team makes a specific contribution to the overall microservice-based system environment.

(PT Schneider) PredictiveCarMaintenance, ClinicsAssetManagement, Microservice Engineering, IoT, SensorThingsAPI, Data Protection

(PT Sidler) EnvironmentalMeasurementDashboard, Domain/Application Microservices

(PT Sänger) MicroserviceDeveloperPortal, API Management, IAM

(PT Throner) DevOps, DevOps Templates, Pipelines, Kubernetes Cluster
(1) The goal of the subteam Augustin of PT Schneider is to provide the current position of a medical device to the ClinicsAssetManagement (CAM) application. For this purpose, the position of the medical device must be known. If a device is moved to another location, the current position must be adjusted. Sensors of a medical device provide the geolocation. The geolocation format needs to be mapped to the data format of CAM. As an example, new sensor information need a timestamp which is newer than the last recorded value. After the bounded contexts have been established, they need to be implemented as microservices. As a result, parts of the IoT domain microservices IoTDataStorage and DeviceManagement should be implemented and tested.

(2) The subteam Brosch is concerned with developing the capability "Monitoring of Vehicles". One task contains the development of the capability, meaning the microservice engineering process is followed. For implementation, the implementation and test concept is applied. Problems and insights should be stated and suggestions for improvements are part of this task. Since sensor data is required in a similar way as the for the localization of medical devices, the connection to the provided services of the IoT domain need to be established. Furthermore, the IoT services need to be implemented and tested by following the implementation and test concept.

(3) The part of Lubitz concerns data protection and privacy which is important since implemented applications need to follow the General Data Protection Regulation (GDPR). For example, a person has the right to know which data is processed and stored. Therefore, privacy by design principles are followed, and the analysis artifacts are enhanced with those principles. A further concern is the architecture, which may need to be adjusted to fulfill the General Data Protection Regulation (GDPR). This topic is part of the project teams research.
The first task is the development of the application microservices. They are based on the frontends of the LUBW environment applications "air", "flowing water" and "radioactivity". This includes the following steps:

1.1 An analysis of the corresponding frontends is required. In this analysis, all the potential functionality for the microservices has to be identified.

1.2 This step deals with the implementation of the mentioned functionality from the first task. For this step, it will be necessary to generate test data as well which deliver input for the application microservices. Later, the application microservices will be connected to the improved domain microservices. During this step, experimental frontends can be implemented, too.

1.3 In the aftermath of the previous implementation, an analysis of the created implementations is done. Similarities between the microservices (and consequently between the different frontends) have to be found as well as metrics which help to measure the similarity. For this step, additional support is given by the project team leader.

The second task deals with the development of the domain microservices and the according artifacts. This includes:

2.1 The analysis artifacts have to be adapted to meet the changed requirements for the EMD application, which means the additional LUBW applications which have to be considered ("flowing water", "radioactivity"). Especially, the capabilities have to be adapted and other artifacts like UI flows and features have to be adjusted accordingly.

2.2 The API and the documentation have to be improved and updated. Based on the adapted artifacts, changes have to be implemented.

2.3 This task is referred to the current status of the implementation. It has to be extended by all the features which were specified in the previous tasks.

**EMD**: Environmental Measurement Dashboard

**MS**: Microservice

**LUBW**: Landesanstalt für Umwelt Baden-Württemberg
The MicroserviceDeveloperPortal (MDP) was developed in the project team SängerHanselmann during the winter semester 2020/2021. The MDP provides a GitLab pipeline extension which registers microservices with the deployment information and API information. Furthermore, the MDP provides a basic health monitoring. The further developments to the MDP are made following the C&M microservice engineering process.

1. The subteam Wank establishes the connection between the Microservice Engineering Process and the MDP. The goal is to ensure that the DevOps concepts are applied to the different C&M applications such as ClinicsAssetManagement (CAM) and PredictiveCarMaintenance (PCM) by supporting the respective teams.

2. The subteam Bozhilov is concerned with adding API management to the MDP.
   2.1 The current version of the MDP only links to the API information in the GitLab repository. If no API is provided in the repository, the MDP will not provide any API information.
   2.2 The goal of the subteam Bozhilov is to provide advanced API management capabilities. The API should be collected, stored, and displayed within the MDP. Further aspects such as Identity Access Management (IAM) should be added to manage API access.

3. The subteam Sänger is concerned with advancing the monitoring capabilities of the MDP.
   3.1 The current health monitoring is done through a Kubernetes operator which only provides limited information (i.e., UP or DOWN).
   3.2 The goal of the subteam Sänger is to extend the monitoring aspects to provide more fine-grained information such as workload. Ideally, the monitoring information are displayed withing the dashboard in (close to) real-time.
   3.3 The monitoring aspect of the MDP should be enriched by logging information which can be provided to the developer through the dashboard.

API Application Programming Interface
CAM ClinicsAssetManagement
IAM Identity Access Management
MDP MicroserviceDeveloperPortal
PCM PredictiveCarMaintenance
(1) In the first subteam of PT Throner, the goal is to develop a pipeline that takes the artifacts from the engineering process and integrates them into the architecture and design, making them accessible to all developers. Regular Continuous Integration / Continuous Delivery (CI/CD) concepts use specific artifacts that are not directly related to the development process. Likewise, there are artifacts that are only used for the design of the software. The connection of these artifacts in the different disciplines has to be examined and the interaction of these is to be pointed out. By the development of content, the current pipeline structure at C&M should be demonstrated and optimized.

(1.1) The existing pipeline concept at C&M as well as the ClinicsAssetManagement (CAM) artifacts have to be examined in order to elaborate the interrelationships of the pipeline with the engineering artifacts.

(1.2) To illustrate the relationships between the pipeline and the artifacts, a generic model for a single step of the pipeline should be developed and a UML stereotype should be created. Based on this model, the complete pipeline of the CAM will be modeled.

(1.3) The pipeline of the CAM is supposed to be optimized on the basis of the created model. The developed correlations and the connected artifacts of the pipeline should be highlighted and made accessible to all developers during the execution of the pipeline.

(2) The goal of the second subteam of PT Throner is the creation of a declarative description for the infrastructure requirements for cloud native application. The delivery of software usually requires an existing infrastructure, which is provisioned via a separate process according to the current DevOps practices. The project will investigate whether and how the separation of infrastructure and software delivery can be combined into a single process. To test the provisioning of infrastructure the education program of the cloud provider AWS will be used.

(2.1) The operational requirements should be extracted from the analysis and design artifacts of the CAM application. Missing requirements should be added accordingly.

(2.2) Basic operational requirements should be summarized and integrated in the MicroserviceDevelopmentPortal (MDP). The need and a unit of measurement for the requirements should be introduced. As an example, the CAM application should be used.

(2.3) A suitable format should be introduced for the provisioning of the operational aspects, so that it can be integrated into existing technologies such as Terraform, Pulumi or Helm.
Requirements for Participating in the Practical / Seminar Course

1. Project team meeting each week
   - The dates are individually fixed by the project team leaders and the Junior Students

2. Students who want to pass both a practical course and a seminar course should have completed their practical course (= 150 working hours) at least 5 weeks before the end of the lecture period

3. Presentation of the results in the WASA lecture
   - See dates for each project team on the page ++Lecture Plan++

4. Delivery of the final practical /seminar thesis on the last day of the lecture period

This page summarizes all relevant dates that are relevant for each participant of the practical /seminar course offered in combination with the WASA lecture.

(1) It is absolutely necessary that a student has a free slot in his/her personal time table in order to be able to take part in these weekly meetings. The meetings start in the next week (i.e. the second week of the lecture period).

(2) The precise date is communicated in the project teams. The date is also fixed in the team calendar of the C&M Teamserver.

(3) The content produced for the presentation should conform to the WASA course material.

(4) This day is the Friday of the last lecture week. The whole Friday (i.e. 12 pm) is available to finish the documentation.
Next Steps

(1) Personal decision if you want to participate in the WASA lecture and an accompanying practical/seminar course

(2) If YES

(1) Latest until Thursday, 21.10.2021, 10 am: Send an email with your depseudonymized KIT mail address to cm.research@lists.kit.edu with the following information:

   (1) Prenname, surname, matriculation number, interest in practical course (preferred) or (pro)seminar course or both

   (2) Personal motivation and experiences in this area (e.g. IT project experiences)

   (3) 2 to 4 project teams of interest (highest priority first)

(2) The answer to your email will contain all relevant information (esp. access to team server, date of kickoff meeting, first work packages) to start the lecture and the practical/seminar course

(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).

(2.1) Check on the page of the KIT Steinbuch Computing Centre if your email is already depseudonymized.

(2.3) The motivation and the experiences should be summarized in at least one or two paragraphs.

(2.4) This means that the first project team in the list is your favorite team.

(3) This email will be sent by the co-supervisor of the members of the project team which usually is a SeniorStudent.

The current lecture material is stored on the C&M Teamserver in the following folder: https://team.kit.edu/sites/cm-tm/Mitglieder/2-0.Aktuelles_Semester
C&M Teamserver

(1) Accessible via the web link
https://team.kit.edu/sites/cm-tm/

(1) Access is given with the admission to the practical/seminar course

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

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

(1) Document C&M-TEAMARBEIT

(2) WASA course units presented in the current semester

(3) Personal working folder to store your own documents related to the practical/seminar course are subfolders of the co-supervising SeniorStudent/PhDResearcher

The C&M Teamserver is described in detail in the document C&M-TEAMARBEIT (in German) which is available
(i) on the C&M web site: https://cm.tm.kit.edu/
(ii) on the C&M Teamserver: https://team.kit.edu/sites/cm-tm/Mitglieder/1-1.Teamarbeit
Activation of the Name-Related E-Mail Address (Depseudonymization)

(1) 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-Address <prename><surname>@student.kit.edu" exists additionally to the "uxxx@student.kit.edu" email address.