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 Domain-Driven Design, Behavior-Driven Development, Microservices, RESTful Webservices) as well as related standards and technologies (including HTTP, JavaScript/TypeScript, Angular, Spring) are introduced which are needed to develop advanced 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.

Further remarks:
(1) The lecture takes place each Wednesday from 9:45 am to 11:15 am in the computer science building "Am Fasanengarten" (building number 50.34) in the room SR301.
(2) The written lecture content is made available in English. In the lecture, the content is presented and discussed in German. Each student participating in the WASA lecture can choose if he/she wants to write his/her practical/seminar thesis in German or in English. Thesis templates are made available both in LaTeX and in Word.

C&M Cooperation & Management
HTTP HyperText Transfer Protocol
KIT Karlsruhe Institute of Technology
REST REpresentational State Transfer
WASA Web Applications and Service-oriented Architectures

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

(Connected Car, Internet of Things) The development of connected car application and IoT application based on the concept of domain modeling and microservice architectures.

(Identity and Access Management, SecDevOps) The Identity and Access Management (IAM) is a highly relevant crosscutting concern appearing in almost every web application. One of the projects is concerned with the provisioning of IAM as a flexible cloud service.

SecDevOps  SecurityDevelopment Operations
IAM  Identity and Access Management
IOSB  Fraunhofer Institute of Optronics, System Technologies, and Image Exploitation
      (Optronik, Systemtechnik und Bildauswertung)
IoT  Internet of Things
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
The activity diagram gives an overview of the whole engineering process which in fact is divided into two sub-processes: According to our understanding of DDD, a domain model is the foundation of all software applications belonging to this domain. We regard the domain model as the knowledge of a domain expert he or she makes available to the software developer by the model. This knowledge is the application-agnostic part of the application meaning that each application of the domain implements this knowledge in its domain logic layer. We understand this domain knowledge as the "physics" that each application must adhere to. Evans calls this part of functionality the heart of the software system [Ev03].

(Starting Points) (Strategic Modeling of the Initial Context Map) The starting point of the domain modeling process is set earlier as the starting point of the application development process. The reason is that it makes sense to have an initial context map of the domain before the development of the first application starts. In the following we describe the main characteristics of the application development process.

(Context Map) The main artifact of the domain modeling process is the context map in which all micro-service-based applications belonging to the domain must fit into.

(Strategic Modeling of the Bounded Contexts as Part of the Domain’s Context Map) In the application development process, the integration of the application under development into the context map of the domain is done in the second step after the first step in which the the BDD-based requirements analysis is carried out.

(Tactical Modeling of a Bounded Context) A bounded context is part of a context map. It represents a candidate for a microservice which can be developed by an independent team [Ne15]. DDD considers the modeling of the content (i.e. the functionality) of a bounded context as tactical modeling.

(Relation View) The most relevant artifact of the tactical modeling is the relation view which is built in the third step. By the relation view central DDD concepts (such as entity and value objects) including their relationships and operations are represented.

(Implementation of each Bounded Context as a Microservice Accessible via its Web API) The functionality modeled by the relation view is implemented in the domain logic layer of the microservice architecture. A systematic approach to implement the bounded context as a microservice (fourth step) is a core characteristic of the proposed development process. One of the key demands of DDD is to keep the model and its implementation in sync in order to assure that the model and the implementation is the same.

The two remaining steps concern development activities around the core implementation of the microservice, namely the implementation and test and the deployment of the application.

The main dimension according which the development tools can be ordered are the development phases (from analysis to deployment). The tools supporting the analysis and design and the implementation and testing can be grouped according to the software architecture (which in our case is a microservice architecture).

(Project Management and Version Control) These tools support the overall organization of the software project and the communication between the project members. At C&M two different tools sets are used: the Atlassian toolset extended by Microsoft tools, esp. SharePoint on which the C&M Teamserver is based.

(Analysis and Design) Analysis requirements at C&M is done by taking the approach of behavior-driven development (BDD) based on the tool Cucumber. For the design the two most relevant tools are Enterprise Architect for the domain model and Swagger for the API specification. In addition to these tools the Microsoft Office tools (Word, PowerPoint) and Atlassian Confluence (in the case of iCC/xdi projects) are applied for documentation purposes.

(Implementation and Testing) In this phase the frontend and backend of the web applications are constructed. The integrated development environment (IDE) used for frontend development is JetBrains's WebStorm and frontend frameworks are Angular and Bootstrap. As IDE for backend development Eclipse is used and Apache Spring (esp. Spring Boot for the microservice implementation) is used as backend framework.

(Build and Deployment) The build and deployment of the microservices is carried out via a build pipeline by which the concept of continuous integration and continuous deployment is provided. The result of the build pipeline is a Docker image (= application container image) since Docker is used as the container environment at C&M. Kubernetes is used to manage the Docker containers in order to reach a high scalability and robustness of the service landscape.

BDD Behavior-Driven Design
IDE Integrated Development Environment
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.
The acronym WASA stands for "Web Applications and Service-orientietered Architectures". Four different types of WASA courses are offered: (i) lecture courses WASA1 an WASA2 (ii) practical courses WASA1 and WASA2 associated to the lecture courses (iii) proseminal 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 duration of the oral exam is 20 minutes. The examiners are Prof. Abeck and one of the C&M's PhDResearchers. 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.

(3) During the first phase of the practical/seminar course all JuniorStudents fulfill the same tasks. After the successful completion of this phase each JuniorStudent knows how to use the major tools (esp. GitLab) to build microservice-based applications.
The task document (Aufgabendokument) is a Word document which describes the tasks a JuniorStudent has to fulfill during his/her practical/course. The document is stored in the personal working folder on the C&M Teamserver (further described on a following page <<C&M Teamserver>>).

(1.1) The single tasks to be carried out and the results to be achieved during the initial phase can be found in the task document.

(1.2) These tasks are specified individually for each JuniorStudent by the co-coaching PhDResearcher or SeniorStudent.

(2) The exact delivery date can be found in the task document.
(2.1) The result of the review is available one week after the delivery of the document (stored in the review folder of the JuniorStudent).
The figure illustrates which research areas are covered by the Bachelor/Master/PhD theses currently worked out at C&M. Each student who takes part in the WASA lecture will be assigned as a JuniorStudent to one of these theses and the related SeniorStudent/PhDResearcher. The SeniorStudent/PhDResearcher is the co-supervisor of the practical/seminar work carried out by the JuniorStudent. A SeniorStudent/PhDResearcher and the JuniorStudents assigned to him/her make up a project team (PT). The PT meetings take place in the Team Room 032 (alternatively in the NZ1 Seminar Room in the case that Room 032 is occupied).

In the following, each PT is presented by introducing into the topic and by showing how the JuniorStudents will be integrated into the work of the SeniorStudent/PhDResearcher who is leading the PT (Schneider, Holzmann, Gassauer, Kostka; Hippchen; Sidler; Hoyer; Throner, Braun, Qattan)

PT          Project Team
The goal is to create a clean architecture for IoT. This means, that no business parts are integrated into the IoT domain and vice versa. In order to provide a microservice architecture for the IoT domain, the IoT domain needs to be divided into several functional services. Domain-Driven Design (DDD) is one of the approaches that can be used to determine the domain logic of the services.

(1.1) Different artifacts, such as the context map and an application-driven development process are already established. However, the applicability of the artifacts has to be shown. Therefore, one goal of this project team is to verify if the established artifacts are useful and at which points adjustments (or additional artifacts) are required.

(2) The following points show the work packages of the IoT project team.

(2.1) One of the first goals is to get a brief understanding of the different standards such as OGCs SensorThings API [OGC-Sen] and W3Cs Web of Things [W3C-Web] in order to get a basic understanding of IoT problems and proposed solutions.

(2.2) The development process which is part of the WASA course is extended and function as some sort of filter between the applications business domain and the IoT specific domain. The goal behind this process is to separate the IoT domain and the business domain. This process should be applied to an existing IoT application (like ECC) and as a result, it should reveal the architectural deficiencies of the application.

(2.3) The extended development process is applied to one of the application of the connected car domain. The goal is to separate the previous mentioned IoT and the business specific.

(2.4) This work package is related to the previous and should reveal problems and missing artifacts for the development of IoT applications.

(2.5) By the use of the previous generated artifacts, a prototype of an IoT microservice should be developed.

(2.6) Testing IoT services is one of the work packages. There are tests required for the IoT service itself as well as integration tests and end-to-end tests for testing the whole IoT application. The tests should be carried out within a CI/CD pipeline which needs to be set up and configured.

OGC Open Geospatial Consortium
W3C World Wide Web Consortium

(1) The goal of this project team is to achieve loose coupled communication between microservices. Loose coupling means the independence of two microservices communicating with each other [Ne15]. This gives the ability to change a microservice without affecting the counterpart.

(1.1) Asynchronous communication is based on the use of events [BP19]. Microservices only communicate with each other indirectly via another medium.

(1.2) Such a medium is a message broker. It allows to publish microservices on special events or to subscribe to them. Examples of such a message broker are RabbitMQ [Piv-Rab] and Kafka [Apa-Kaf].

(2.1) To work on the project team topic it is necessary to know the basics of loose coupling, asynchronous communication and message broker. To work on the project team topic it is necessary to know the basics of loose coupling, asynchronous communication and Message Broker. Furthermore the event-based communication has to be considered. Within the events there are different concepts for the realization of communication such as event sourcing [Mic-Eve] and saga pattern [Mic-Sag].

(2.2) The new knowledge about microservices is to be applied as an example to the Electric Car Charger (ECC) application. It is therefore also important to familiarize oneself with the application.

(2.3) The message broker has been chosen on RabbitMQ. It is suitable for an easy entry. An instance of the RabbitMQ must be set up for the project team.

(2.4) In order for the ECC to be able to be sensibly based on the RabbitMQ, the application must first be extended by events. These events have to be determined by the project team.

(2.5) If both RabbitMQ and ECC are ready, both can be connected.

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.

(2) This preliminary version is reviewed by the co-supervising PhDResearcher/SeniorStudent.

(3) The presentation material are in conformity with English slides and additional English text (see the WASA course material as an example).

(4) The delivery date is the last day of the lecture period.
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 depseudonymised.

(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.

(3.1) 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
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". Die name-related E-Mail-Adresse <prename><surname>@student.kit.edu" exists additionally to the "uxxx@student.kit.edu" email address.
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