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, 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 summer semester 2020, the lecture and practical/seminar course are organized as an online event. This new format is called WASAOnline.

The WASAOnline kickoff lecture will take place on Wednesday, April 22nd 2020 on 9:45 am.

All students who want to take part in WASAOnline and in the kickoff lecture should send an email to Sebastian Abeck (sebastian.abeck@kit.edu) to ask for the web link to the web session.

The lecture material is made available in English. In the online lecture, the content is presented and discussed in German. The oral examination is conducted exclusively in German. Each student 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
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, Internet of Things) In this area connected car application and IoT 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).

(Identity and Access Management, SecDevOps) SecDevOps concerns the continuous integration (CI) and continuous deployment (CD) of (hopefully not monolithic, but microservice-based) software systems deployment into a container-virtualized (Docker/Kubernetes-based) cloud infrastructure. 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.

BDD        Behavior-Driven Development
CI/CD      Continuous Integration / Continuous Deployment
DDD        Domain-Driven Design
IAM        Identity and Access Management
IOSB       Fraunhofer Institute of Optronics, System Technologies, and Image Exploitation
           (Optronik, Systemtechnik und Bildauswertung)
IoT        Internet of Things
SecDevOps  Security Development 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
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 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
Der IAM Coding Days (& Night) liefert den Studierenden die Möglichkeit, anhand spannender Projekte aus dem Bereich von IAM (Identity and Access Management) technische Kompetenzen zu entwickeln und Praxiserfahrungen zu sammeln.
(IAM & Connected Car) This is the main topic of the IAM Coding Day 2020.

(Predictive Car Maintenance, Connected Car API, OpenAPI) The connected car application to be developed implements the predictive maintenance of a car. The information that is needed for predictive maintenance is provided by a connected car's API specified in the OpenAPI format.

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

(Users, BDD, Cucumber) The functionality the predictive maintenance application should provide to the user is specified by Gherkin features. Such a feature is an analysis artifact introduced by the concept of Behavior-Driven Development (BDD). The tool Cucumber is used to carry out the end-to-end tests going through each Gherkin feature.

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

(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.

(API Security, Connected Car API) The API offered by the connected car must be protected against unauthorized access. The IAM service provides the authentication and authorization information needed to take the access decision at the API.

(IoT, IoT Service, SensorThings, Web of Things) Internet of Things (IoT) is an additional topic covered by ICD2020. A connected car today is one of the most relevant "Things of the Internet". Therefore, IoT standards, such as SensorThings API from OGC and Web of Things from W3C, should be taken into account in the development of the predictive maintenance application.

(DevOps, Build Pipelines, Kubernetes Cluster, Prometheus Monitoring) DevOps is the second additional topic for which IC2020 will make relevant contributions. The goal is to develop a build pipeline based on GitLab-CD by which the predictive maintenance application is continuously integrated (CI) and deployed (CD) on a Kubernetes cluster. The predictive maintenance application provides metrics which are monitored by 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-orientierte 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 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.
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 which is available for 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 ICD2020 show case.

**ToDo: A short description of the contribution is to be provided by each project team leader**

(PT Piatkowski)

(PT Sidler)

(PT Hippchen)

(PT IAM_NN)

(PT QattanGiner)

(PT SchneiderErol) The goal within this project team and ICD2020 is to provide a streaming-oriented enhancement for the given car API.

(PT ThronerGogel)
(1) The goal within this project team and ICD2020 is to provide a streaming-oriented enhancement for the given car API [IC19 Car]. A connected car could be seen as a thing which is connected to the internet and provides a huge amount of sensor information. A first approach to handle streamed data within a car is provided by Sensor Car Data Manager (SCDM) [Ho20]. However, SCDM does not use a uniform data format.

(1.1) It is beneficial to use a standardized format, because otherwise, different data formats will be used (e.g. by car manufacturers). This can lead to the typical interoperability problem within IoT.

(1.2) Since SCDM does not provide or utilize a uniform data format (in contrary to SensorThings API), the services need to be adjusted to match the SensorThings API data format.

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

(2.1) Especially for the ICD2020, there should be a clarification between the IoT-based streaming API and the resource-oriented car API. At the current standing, the resource-oriented should be extended by the project team by implementing a streaming-based addition. During this work package, it must be clarified which functionality is offered by which API part.

(2.2) The adjustments should be utilized by following the SensorThings API standard.

(2.3) The SensorThings API standard leaves room for unprecise implementation. Therefore, there should be created guidelines for implementing the SensorThings API in order to assist developers.

(2.4) Not every subject is allowed to see the provided data. Therefore, there is a need to protect the streaming and resource-oriented data. The considered solution presumes that policies are defined in the vehicle, which determine which data streams are available at the streaming API.

(2.5) Fraunhofer's FROST server could be used to implement the SensorThings API for the connected car.

OGC       Open Geospatial Consortium
SCDM      Sensor Car Data Manager
FROST     Fraunhofer Open Source SensorThings API Server
PCM       Predictive Car Maintenance

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 content produced for the presentation should conform to the WASA course material.

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