This WASA course unit [CM-W-WAS] 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).

WASA1 (Bachelor): Current concepts of software development and architectures (including Behavior-Driven Development, Domain-Driven Design, Microservices, RESTful Web services, 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 different domains (Healthcare, ConnectedCar) and includes concepts from the domain of Internet of Things.

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. IAM is highly relevant for the digitization strategy of organizations. In the lecture, leading IAM solutions and products are introduced to illustrate how the IAM challenges are solved in IT practice.

The WASA kickoff lecture will take place
== on Wednesday, 19th April 2023 at 9:45 am
== in the building 50.34 (Informatikgebäude am Fasanengarten), room SR301

The WASA lecture places and WASA practical/seminar places are restricted. Each student who wants to take part in the WASA kickoff lecture should
== send an email to cm.research@lists.kit.edu
to apply for a lecture place and a practical or seminar place. 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.
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 Behavior-Driven Development (BDD) and Domain-Driven Design (DDD). In addition to development (Dev), the operational aspects (Ops) are intensively taken into account. 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) Identity and Access Management (IAM) is a highly relevant crosscutting concern appearing in every web application. Advanced IAM topics are (i) decentralized identities which make the user the owner and controller of their identity data and (ii) fine-grained authorization which takes access decisions based on a broad spectrum of different attributes.

BDD Behavior-Driven Development
CI/CD Continuous Integration / Continuous Deployment
DDD Domain-Driven Design
IAM Identity and Access Management
DevOps Development and Operations
A microservice architecture is located on the application plane as it is introduced in ++Network, System, and Application Plane+++. While the software architecture is described by the logical layers specified by a specific 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.

API	Application Programming Interface
CRUD	Create, Read, Update, Delete
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 text, 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 consideration and integration of the external systems (esp. enterprise applications, business services, or databases) into the microservice application which is to be developed is a main characteristic of MuleEng. This is a highly relevant aspect in practice which was not sufficiently taken into account in CMEng.

(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) Process APIs which are derived from the use cases, (ii) System APIs which integrate the external systems, (iii) optional Domain APIs 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) (Experience API) As soon as the experience API is specified, 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 implementation and test usually follows the CMEng approach in which the microservices are coded by using a Java- or Go-based framework. Alternatively, the MuleEng approach can be applied, in which the implementation is based on so-called Mule flows (low code / no code).

Deployment and Operations) The template-based deployment approach used in both the CMEng and the MuleEng approach is applied in the UME approach.

UI User Interface
UME Unified Microservice Engineering
A Minimum Viable Product (MVP) serves as a first minimal, nevertheless complete application CCSApp [CM-G-CCS] to demonstrate the correct interaction of the involved APIs and their deployment. The MVP of the CCSApp has the version number V1 and is, therefore, it is named CCSAppV1. On the right side of the slide the SystemPlusSoftware (SPS) architecture of CCSAppV1 is illustrated.

(1) The limitations of the MVP concern two aspects, (i) the supported functionality, and (ii) the considered manufacturers.
(1.1) By this restriction, the Process API (and as a consequence, also the Experience API and the UI) focus only on this part of the whole CSSApp functionality.
(1.2) Since S-DaimlerCar is mocked the external system Ext-DaimlerCar is not needed in the MVP (leading to a reduction of complexity).

(2.1) This especially eases the maintenance of the APIs.
(2.2) Public mock APIs are used which allows a local testing without the need to deploy all direct and indirect dependencies of an API.
(2.3) This decision was taken to ease the deployment of the MVP. No external pipeline is needed since the deployment is completely carried out by the MuleSoft tools.

MVP Minimum Viable Product
SPS SystemPlusSoftware

https://git.scc.kit.edu/cm-tm/cm-team/connectedcar/mulesoftarchitecture/connectedcarservicesapplication/docccsapp
The acronym WASA stands for "Web Applications and Service-oriented 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) prosemear course associated to WASA1 lecture course and seminar associated to WASA2 lecture course (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 and the prosemear/seminar run in parallel with the lecture course. The practical course counts 5 credit points meaning a workload of 150 hours and the prosemear and seminar count 3 credit points meaning a workload of 90 hours.

(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)
<table>
<thead>
<tr>
<th>Date</th>
<th>Content</th>
<th>Lecturer(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>23.04.19</td>
<td>WASA INTRODUCTION</td>
<td>Abeck, PhD Researchers, Senior Students</td>
</tr>
<tr>
<td>23.04.26</td>
<td>WEB APPLICATION DEVELOPMENT: OVERVIEW, TOOL ENVIRONMENT</td>
<td>Abeck</td>
</tr>
<tr>
<td>23.05.03</td>
<td>WEB APPLICATION DEVELOPMENT: ANALYSES, DESIGN</td>
<td>Abeck</td>
</tr>
<tr>
<td>23.06.10</td>
<td>WEB APPLICATION DEVELOPMENT: DESIGN</td>
<td>Abeck</td>
</tr>
<tr>
<td>23.06.17</td>
<td>WEB APPLICATION DEVELOPMENT: IMPLEMENTATION AND TEST</td>
<td>Abeck</td>
</tr>
<tr>
<td>23.06.24</td>
<td>TESTING Contributions Project Team Schneider</td>
<td>Schneider, Senior/Junior Students</td>
</tr>
<tr>
<td>22-30.31</td>
<td>Lecture free Week</td>
<td></td>
</tr>
<tr>
<td>23.06.07</td>
<td>IDENTITY AND ACCESS MANAGEMENT</td>
<td>Abeck</td>
</tr>
<tr>
<td>23.06.14</td>
<td>Industry Talk Decentralized Identities</td>
<td>André Priebe, IC-Consult</td>
</tr>
<tr>
<td>23.06.21</td>
<td>DECENTRALIZED IDENTITIES</td>
<td>Abeck</td>
</tr>
<tr>
<td>23.06.28</td>
<td>AUTHORIZATION Contributions Project Team Sänger</td>
<td>Sänger, Senior/Junior Students</td>
</tr>
<tr>
<td>23.07.05</td>
<td>WEB APPLICATION DEVELOPMENT: DEPLOYMENT AND OPERATIONS</td>
<td>Abeck</td>
</tr>
<tr>
<td>23-07-07 – 23-07-08</td>
<td>IAM Coding Day</td>
<td></td>
</tr>
<tr>
<td>23.07.12</td>
<td>DEVOPS Contributions Project Team Throner</td>
<td>Throner, Senior/Junior Students</td>
</tr>
<tr>
<td>23.07.19</td>
<td>Seminar Presentations</td>
<td>Seminar Students</td>
</tr>
<tr>
<td>23.07.20 – 23.07.26</td>
<td>Oral Examination</td>
<td></td>
</tr>
</tbody>
</table>

The table describes the planned sequence of course units presented during the lecture.

(Content) All titles printed in non-italic build 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 to the PDF file made available in the folder "\sccfs.scc.kit.edu\OE\TM\VR\Mitglieder\2-1.WASA_Aktuell". This indicates which pages must be prepared by the student for the oral examination.
Lecture Agenda

(1) Each lecture event usually includes two parts
(1) Traditional lecture part
(2) Interactive part with contributions from students or guests

(2) The interactive part is shaped in a dynamic way
(1) Lecture agenda is made available on the C&M GitLab
(2) Contributions from participants are always welcome

WASA-Vorlesung

10.04.23 (KW 10): WASA2-Auftrittveranstaltung Sommersemester 2023

 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) (2.2) 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.

Working in a Project Team

1. Each project team meeting is prepared by an agenda to which all team members contribute.
   1. Markdown document in GitLab
   2. Weekly project team meetings starting next week
   3. Project team members communicate in German
2. Junior Students write their practical thesis in English and use the LaTeX tool Overleaf to format their text
3. In the first week all project team members become acquainted with the C&M-TEAMARBEIT

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 "Projekttreffen" 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 practical thesis and the 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 a GitLab document which provides an agenda draft of the project meetings.

\sccfs.scc.kit.edu/OE/TMV\Mitglieder1-1.Teamarbeit
Case Study in the Practical Course in SuSe23: BestRentalPoC

Two steps
(1) Issuance: Alice requests a Verifiable Credential (VC) from DrivingLicenseAuthority
(2) Verification: Alice presents the needed part of the VC as Verifiable Presentation (VP) verified by BestRental

The VC containing claims about Alice's driver's license is digitally signed by the issuer

The trust system implements a decentralized Public Key Infrastructure (PKI)

User Alice

1.1 Alice uses her wallet application to carry out the request. DrivingLicenseAuthority issues a signed Verifiable Credential (VC) which is 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 might contain 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 the DrivingLicenseAuthority 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.

PKI Public Key Infrastructure
PoC Proof of Concept
VC Verifiable Credential
VP Verified Presentation


The SystemPlusSoftware (SPS) diagram is a draft worked out in current research work. The case study BestRentalPoC (PoC: Proof of Concept) is in the center of the practical course in SuSe 23 and the IAM Coding Day 23.
The goal of this project team is to implement an application called BestRental. For the development, a microservice engineering approach is followed and artifacts such as use cases, API diagrams, and the API specifications are created.

(1) The development excerpt focuses on the following two subgoals:

1.1) This first part of the application development addresses the research questions concerning decentralized identities since someone who wants to rent a car needs a valid driving license. As a use case, Alice has a wallet which contains a Virtual Credential (VC). This credential is provided to the application BestRental which verifies the VC. The decentralized identities excerpt is developed based on the Platform as a Service (PaaS) Azure.

1.2) The test concept includes unit tests, integration tests, and end-to-end tests. The focus during this work lies on unit tests and end-to-end tests. Unit tests ensure that the small units work as intended. End-to-end tests ensure that the software works as intended from a user's perspective which means that frontend inputs lead to expected outcomes and the system behaves as intended. The end-to-end tests are implemented in a designated test repository. Test cases and the corresponding test data are an essential part for test consideration. During the analysis phase, the test data is specified and used for the test cases.

(2) The project team tackles two seminar topics which are intertwined with software engineering.

2.1) The first topic deals with the testing of APIs during the software development. The suggested testing approach (as stated above) includes several ideas from existing publications. The goal of this seminar is to further investigate publications which tackle API testing. The starting point is a publication which discusses consumer-driven contract tests.

2.2) An important part of the engineering is the systematic transformation of one model into the next. Model-Driven Architecture (MDA) provides a systematic approach for that. The goal of this seminar topic is to investigate how MDA is used for the development of microservice application. The starting point will be a publication which utilizes MDA in a service-oriented way for the development of IoT-based applications.

<table>
<thead>
<tr>
<th>MDA</th>
<th>Model-Driven Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>PaaS</td>
<td>Platform as a Service</td>
</tr>
<tr>
<td>VC</td>
<td>Verifiable Credential</td>
</tr>
</tbody>
</table>
The goal of this project team is to develop an excerpt of the ConnectedCarServicesApplication (CCSApp). The focus will be on building a demonstrator to research topics of fine-grained authorization. The excerpt of the CCSApp will be implemented using Golang.

(1.1) For the excerpt of the CCSApp, the rental of a car is chosen. The authorization of the excerpt will be implemented following the authorization development process [Bo23]. The authorization policies are written using Rego and evaluated by Open Policy Agent (OPA). Envoy proxy is used to enforce authorization decisions.

(1.2) When implementing fine-grained authorization, an important question is to locate data (e.g., attributes) which is necessary to evaluate authorization requests. This also includes the requirement of data topicality. The goal is to investigate different patterns (e.g., pull or push) and their effect on the performance of the microservice-based application. Here, one metric to examine is the latency.

(1.3) The development of the demonstrator will further investigate authorization between microservices. This includes researching how to identify which microservice is responsible for a given date.

(2) Two seminar topics will be offered in the context of this project team:

(2.1) The confused deputy problem originated from the development of operating systems and describes a program which tricks another program to misuse the system [Ha88]. The problem can be applied to microservice-based applications. A compromised microservice might access data or functions it is not authorized to from another microservice. The goal of this seminar is to investigate how to authorize requests between microservices. An important question is to examine the determination of data sovereignty.

(2.2) Zero trust tries to eliminate implicit trust between systems [TU+21]. For example, in microservices, a request between two microservices should not be allowed just because of the physical proximity (e.g., running in the same cluster). The goal of this seminar is to research the hurdles and implications of zero trust and how it can be introduced in a microservice-based application.

CCSApp ConnectedCarServicesApplication
OPA Open Policy Agent

[Ha88] Hardy: The Confused Deputy: (or why capabilities might have been invented), in ACM SIGOPS Operating Systems Review, 1988.
(1) The goal of this project team is to automate the integration of the required cloud computing resources into the ConnectedCarServicesApplication (CCSApp). The focus will be implementing and integrating an automated provision step into the pipeline.

(1.1) For the excerpt of the CCSApp, registration of the UI-CSSAppWeb as a relying party has been chosen. The registration of the relying party should follow the guidance provided by Microsoft [Mic-Ent].

(1.2) Besides the provisioning and linking of the cloud computing resources, a challenging aspect is the injection of permissions and a configuration to access the provisioned resources. One significant aspect should be managing and transferring the credentials into the service environment.

(1.3) It should be noted, that several applications can exist in parallel without affecting each other. For this purpose, it is necessary to structurally separate the provisioned resources from each other and enable a unique classification of the resources.

(2) Two seminar topics will be offered in the context of this project team:

(2.1) As more or more processes become automated, pipelines also have access to critical system components and can significantly impact if unauthorized people gain access to or tamper with the pipeline. Therefore, it is necessary to limit the access and permission of those pipelines and only allow escalation of the default privileges under specific circumstances by injection or approval to do so.

(2.2) The consumer must authenticate itself to the supplier to enable secure communication between services. For this purpose, credentials or API keys are injected into the service. How this security-critical data can be managed and delivered with the application should be investigated. Security and user-friendliness are essential aspects that should be taken into account.

CCSApp ConnectedCarServicesApplication

This page summarizes all relevant dates that are relevant for each participant of the practical 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 content produced for the presentation should conform to the WASA course material.

(3) 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, 21.04.2022, 10 am:** Send an email with your depseudonymized KIT mail address to cm.research@lists.kit.edu with the following information:
      1. Prenam, surname, matriculation number, KIT account (uxxxx), 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. At least two projects of interest (highest priority first)

   3. The answer to your email will contain all relevant information (esp. access to document storage and C&M GitLab, 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.
2.1.2 The motivation and the experiences should be summarized in at least one or two paragraphs.
2.1.3 This means that the first project team in the list is your favorite team.

3. This email will be sent by the project team leads.
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
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 document filing: \sccfs.scc.kit.edu\OE\TM\VR\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". Die name-related E-Mail-Adresse <prename><surname>@student.kit.edu" exists additionally to the "uxxx@student.kit.edu" email address.