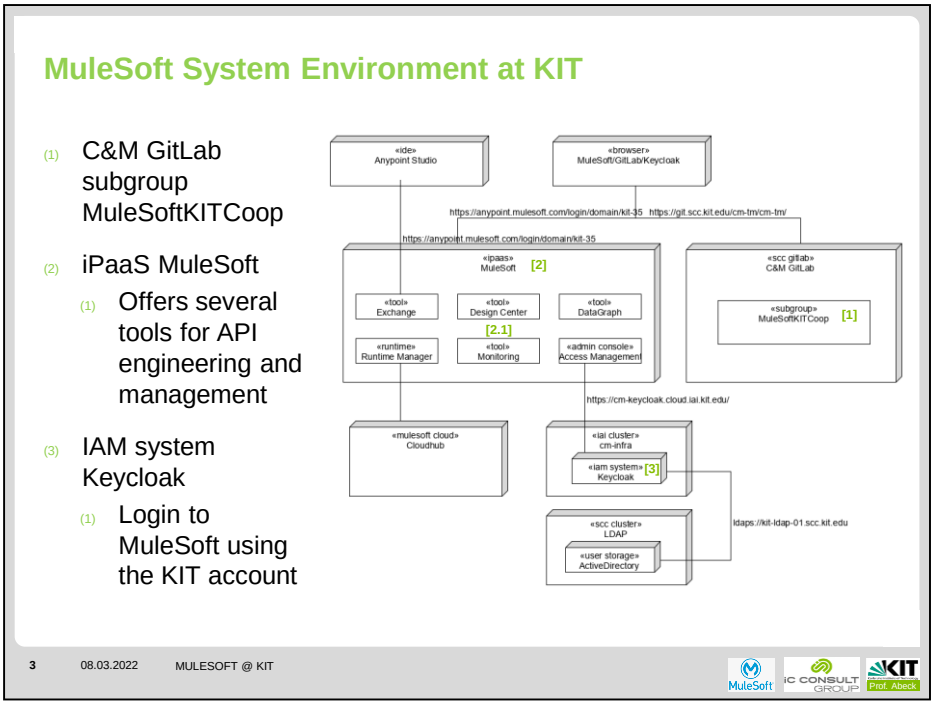


The system architecture describes the MuleSoft system environment at KIT as a UML deployment diagram.

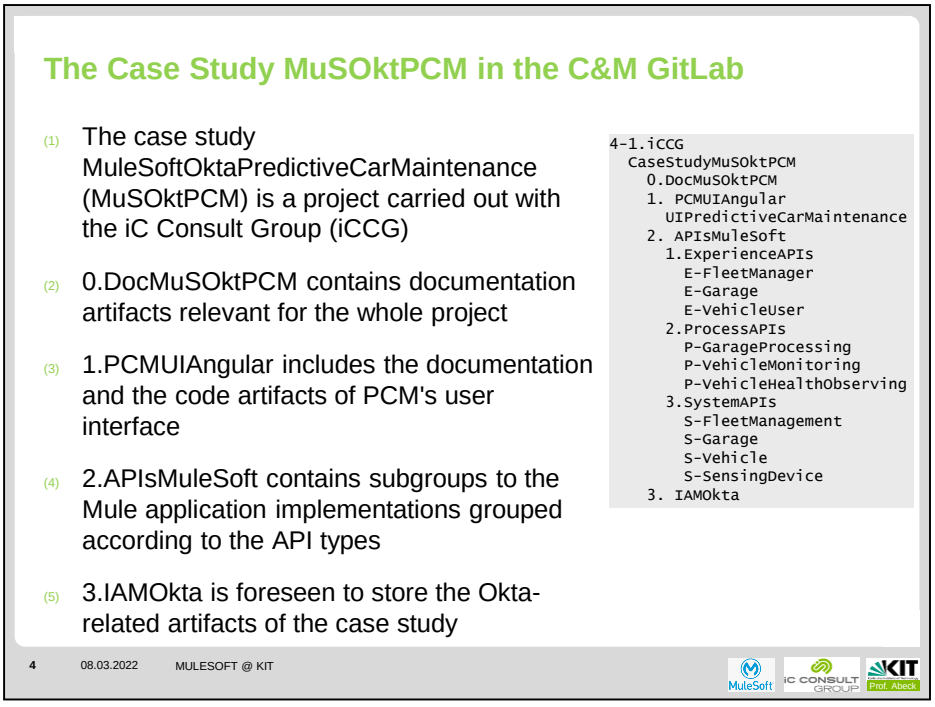
(1) MuleSoftKITCoop is the name of a subgroup of the C&M GitLab which holds relevant documentation of the MuleSoft environment used by C&M in the cooperation with the iC Consult Group [CM-G-Mul]. The description of the environment is part of "1.Organization" in which the access of C&M members to the environment and the administration of the environment are documented.

(2) MuleSoft is provided as a Software as a Service (SaaS) solution which can be accessed via a standard web browser.
(2.1) The tools were introduced in the previous chapters.
(AnypointStudio, Exchange) The «tool» Exchange represents the Anypoint Exchange which can be accessed by the «ide» Anypoint Studio.

(3) In the environment MuleSoftKITCoop, the «iam system» Keycloak running in the «iai cluster» cm-infra allows users the login with their KIT account credentials (i.e., ab1234 or uxxxx).
(3.1) As a pre-condition, the KIT user needs to be added to the «directory» ActiveDirectory group "TM-CM-MuleSoft" which is provided by the «scc cluster» LDAP and used to manage the access for C&M members.
(Access Management, Keycloak) Within the «admin console» Access Management, identity providers, such as Keycloak, can be added to the «ipaas» MuleSoft.
(Browser, Keycloak) Keycloak can be accessed via the browser using the following link by using the KIT account: https://cm-keycloak.cloud.iai.kit.edu/auth/realms/CM/account/#/applications. The Keycloak instance only provides the functionality to revoke the access grant to MuleSoft.

SaaS	Software as a Service

[CM-G-Mul] Cooperation & Management: MuleSoftKITCoop, C&M GitLab repository. https://git.scc.kit.edu/cm-tm/cm-team/4-1.iccg/environments/mulesoftkitcoop



On the following pages, a case study named MuleSoftOktaPredictiveCarMaintenance (MuSOktPCM) will be introduced and designed, implemented and tested based on different technologies, especially MuleSoft and Okta. For a systematic development of the software components that are needed in the case study, a well conceived structure is introduced on this page.

(1) Therefore, a subgroup CaseStudyMuSOktPCM exists as part of 4-1.iCCG. This subgroup contains several subgroups which structure the content of the case study.

(2) The documentation repository is located directly beneath the subgroup CaseStudyMuSOktPCM. The documentation concerns the analysis (to be provided) and the design of the case study solution which includes on overview of all layers of the software architecture (software architecture and system architecture). Furthermore, the documentation links to the specified API documentations (MuleSoft and/or GitLab). This especially depends on the way how the APIs will be documented inside MuleSoft.

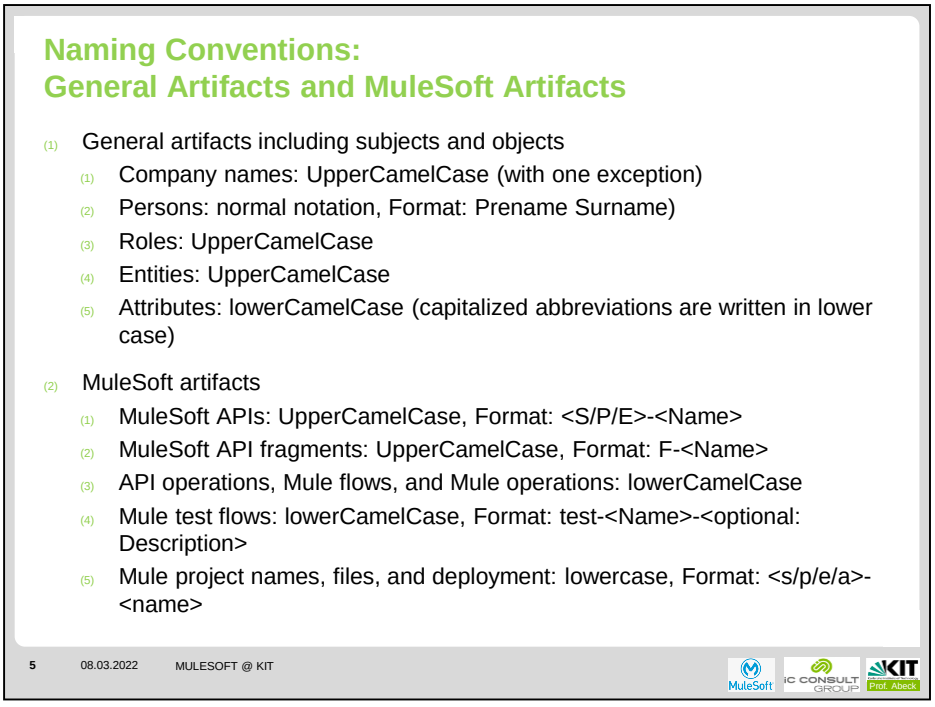
(3) The PredictiveCarMaintenanceUserInterface (PCMUI) is developed with the Angular framework. Authentication and authorization aspects are implemented by using Okta. The documentation contains an overview of the implementation, and the deployment.

(4) For each API type a further subgroup exists. In the corresponding subgroup, the implementation of a Mule App is stored in a repository. This repository is using the following schema: <E/P/S>-< Name>. The corresponding API specification (RAML file) is not stored in GitLab since it is managed by MuleSoft. However, the documentation to the API specification is located in GitLab in the Readme.md of the Mule App to which it belongs. Furthermore, the implemented API specification to MuleSoft is linked in the documentation.

(5) This subgroup is further described in [CM-W-OKT].

E	Experience API
MuSOktPCM	MuleSoftOktaPredictiveCarMaintenance
P	Process API
PCMUI	PredictiveCarMaintenanceUserInterface
S	System API

[CM-W-OKT] Cooperation & Management: OKTA @ KIT. WASA course unit. https://team.kit.edu/sites/cm-tm/Mitglieder/2-2.WASA_Lecture



On this page, conventions for the naming of the subjects and as well as the artifacts are introduced.

(1) (2) For each of the subjects and objects, examples are set.
(1.1) CaseStudyCompany, CaseStudyGarage; Exception: KarlsruheinspiredConsult
(1.2) Alex Twin, Dev Ops
(1.3) Vehicle User, Garage
(1.4) Vehicle, Sensor (all entities used so fa in the case study consist of only one word)
(1.5) dateOfBirth, usedVehicles, vin, (not: vIN), engineerId (not: engineerID)

(2.1) S-Vehicle, P-VehicleMonitoring, E-VehicleUser
(2.2) F-CaseStudyEntities
(2.3) getVehicleByEmailAndVin, getVehicles
(2.4) test-getVehicleByEmailAndVin, test-getVehicles-validVins
(2.5) s-vehicle, p-vehiclemonitoring, e-vehicleuser, p-vehiclemonitoring-test-suite.xml