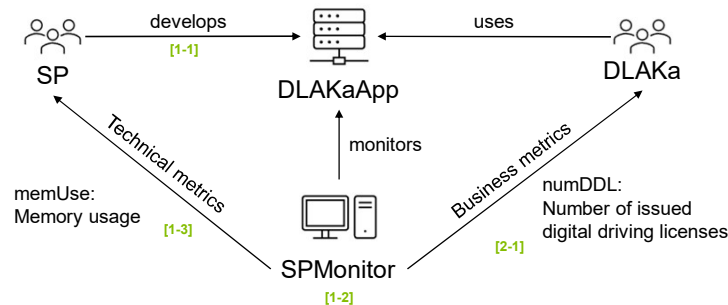


## Monitoring BestRentalPoC

- (1) Operator ServiceProvider (SP)
  - (1) Develops and operates DLAKaApp
  - (2) Creates SPMonitor to collect metrics from DLAKaApp
  - (3) Interested in technical metrics
- (2) Monitoring target DLAKaApp
  - (1) DLAKa is interested in business metric



1

13.08.2023

WASA2

Cooperation & Management (C&M, Prof. Abeck)  
KIT Faculty of Informatics

[1] ServiceProvider (SP) is a fictitious software development company.

[1.1] SP is hired by DLAKa to develop and operate the DLAKaApp.

[1.2] To improve the operation of DLAKaApp, SP creates a monitoring tool called SPMonitor which monitors DLAKaApp by collecting metrics from it.

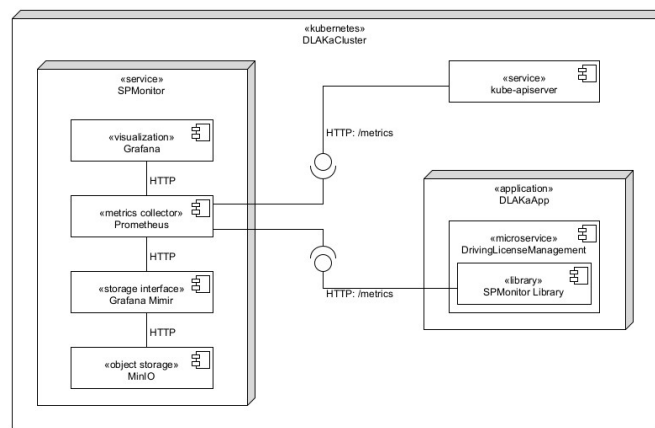
[1.3] SP is interested in technical metrics from DLAKaApp like the metric memUse (memory Usage). Monitoring technical metrics allows SP to gather performance data which can be used to improve the system. These technical metrics are provided to SP by SPMonitor.

[2] DLAKa uses the DLAKaApp in its business processes to issues digital driving licenses.

[2.1] DLAKa is interested in business metrics from DLAKaApp like the metric numDDL (number of Digital Driving Licenses). This metric allows DLAKa to understand how often the system is being used. These business metrics are provided to DLAKa by SPMonitor.

## (1) Integration with BestRentalPoC

- (1) Kubernetes provides required metrics endpoint
- (2) AM-DrivingLicenseManagement needs to be extended



[2]

2

13.08.2023

WASA2

Cooperation & Management (C&M, Prof. Abeck)  
KIT Faculty of Informatics

[1] DLAkaApp requires some changes to be made to it, so that it can be integrated with SPMonitor.

[1.1] Kubernetes already provides all required metrics through an HTTP endpoint.

[1.2] AM-DrivingLicenseManagement needs to integrate the SPMonitor client library which will be responsible for exposing the services metrics through an HTTP endpoint.

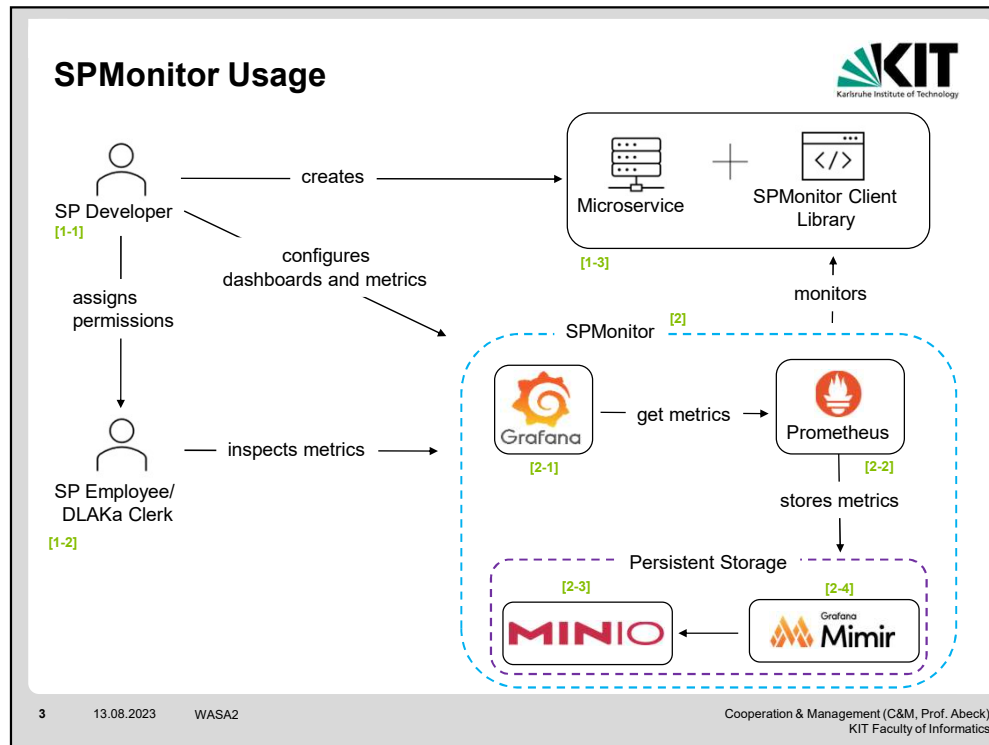
[2] SystemPlusSoftware Architecture from the Bachelor Thesis Engbrocks.

(«visualization» Grafana) Grafana is a tool that provides dashboards to visualize metrics. It acts as the main interaction point for users with SPMonitor. Dashboards consist of different types of graph which can display time-series data like metrics. Grafana also offers pre-built dashboards for example for monitoring a Kubernetes cluster. Grafana retrieves the metrics from Prometheus.

(«metrics collector» Prometheus) Prometheus is responsible for collecting metrics from DLAkaApp and its runtime environment, the Kubernetes cluster DLAkaCluster. Prometheus scrapes metrics through an HTTP endpoint. The AM-DrivingLicenseManagement provides its metrics via the SPMonitor client library which exposes an HTTP endpoint at /metrics. Kubernetes also provides metrics on the same endpoint.

(«storage interface» Grafana Mimir) Mimir acts as a storage interface for Prometheus to use persistent storage which it does not normally have. Prometheus periodically syncs it's data to persistent storage from where it can be used to access old data and be used to restore Prometheus in the case of an outage.

(«object storage» MinIO) MinIO is the persistent storage that is accessed by Prometheus through Mimir. MinIO is a object storage system that mimics Amazon Web Service's S3 but is open-source and free to use. Data is stored as objects in buckets. Object storage systems are ideal for data which has to be read more than it has to be written and which has an arbitrary format. In the case of SPMonitor, data in the form of metrics will only be written once to storage.



[1.1] SP Developer: An employee from SP working in the position of a developer. SP Developer is responsible for developing and operating DLAKaApp and SPMonitor.

[1.2] SP Employee/DLAKa Clerk: A generic SP employee or clerk from DLAKa. They want to inspect their respective metrics. SP inspects technical metrics like memUse and DLAKa inspects business metrics like numDDL.

[1.3] A generic microservice developed by SP Developer for the DLAKaApp. This could for example be AM-DrivingLicenseManagement. The microservice uses the SPMonitor Client Library to expose its metrics.

[2] SPMonitor is a collection of services and not one specific microservice.

[2.1] Grafana presents dashboards to users through which the collected metrics can be inspected.

[2.2] Prometheus collects the configured metrics.

[2.3] Grafana Mimir is a storage interface for Prometheus to access persistent storage.

[2.4] MinIO is an object storage system built to mirror Amazon Web Service's S3. It is the persistent storage for the collected metrics.