

Criteria Catalog

- Based on requirements for the prototypical implementation -

The aim of this criteria catalog is to verify a successful prototypical implementation of a monitoring solution.

During a workshop with the business side and the operations team, the following criteria regarding the prototypes and the monitoring solution were defined, and the system environment was specified.

No.	Description	Criteria	Fulfilled?
1.	Prototypes		
1.1.	Close-to-production implementation with Spring Webflux	The Spring Webflux Framework is used as the basis for the prototypes.	
1.2.	Close-to-production implementation with the modules developed internally	The following internally developed modules are used: - Security Module - Mongo Module - Objectmapper Module - Validation Module	
1.3.	Trace simulation across multiple microservices	Two microservices were created that can simulate a trace across multiple microservices.	
1.4.	Database connection	- Service uses the MongoDB Docker container locally and can access it via a repository.	
		- Service uses the existing MongoDB in the cloud and can access it via a repository.	
1.5.	Kibana (APM) connection	- Service runs locally with Docker container for Kibana APM	
		- Service runs in the cloud with existing Kibana.	
		- Data is sent correctly through microservices to Kibana.	
2.	System environment		
2.1.	Local development	The prototypes can be run in the local system environment for development purposes.	
2.2.	Delivery to the cloud	The existing CI/CD solution is used to build, to test and to deploy the prototypes.	
2.3.	Runs in the cloud	The prototypes can be run in the cloud system environment.	

3.	Monitoring values		
3.1.	Runtime request	The duration of the request to the partner system is measured.	
3.2.	Response duration	The duration to process the response is measured.	
3.3.	Request/response successful	The successful requests/responses are measured.	
3.4.	Request/response with error	The unsuccessful requests/responses are measured.	
3.5.	Number of occurring errors	The number of occurring errors is measured.	
3.6.	Number of occurring warnings	The number of occurring warnings is measured.	
3.7.	Microservice availability	The availability of the system can be measured.	
3.8.	GUI (Graphical User Interface) availability	The availability of the GUI can be measured. <i>-Only applicable to microservices with GUI-</i>	
3.9.	CPU usage in %	The CPU usage is measured in %.	
3.10.	Memory usage in %	The memory usage is measured in %.	

