A Microservice-based Condition Monitoring System for the Beer Brewing Process

Faruk Pasic[1] and Matthias Becker[1]

Fraunhofer IEM, Paderborn, Germany
faruk.pasic@iem.fraunhofer.de, matthias.becker@iem.fraunhofer.de

Abstract

In this work, we present our condition monitoring system for the beer brewing process that is based on the microservice architecture. The system follows the condition monitoring software specification defined by the ISO 13374 standard, where each data processing functional block, e.g., prognostics assessment, is implemented as an independent software module. Microservices showed to be a good choice for condition monitoring systems as these systems often demand new analytic services and scalable software architecture. We outline our condition monitoring microservice architecture and report on lessons learned.

1 Introduction

The process of beer brewing is very complex as it has to fulfill strict demands on the product quality as well as on the availability and the performance of the plant. Even a small disruption in the machine operation can significantly influence the overall beer quality and, consequently, lead to economic losses for the beer brewing companies. In this context, a condition monitoring of the beer brewing process plays an important role such that all relevant deviations are detected as early as possible by the production manager. However, the development of a condition monitoring system for the beer brewing process requires a software architecture that allows frequent and easy integration of new analytic services and, hereby, fosters their scalability. Furthermore, these condition monitoring services have to utilize powerful analytic libraries written in different languages, e.g., Python and R, in order to aggregate numerous sensor values and provide a prognostic and/or diagnostic assessment.

In this report, we present our microservice architecture for the condition monitoring of the beer brewing process, which was implemented stepwise within our agile development approach.

2 Condition Monitoring Microservice Architecture

Our condition monitoring system (CMS) for the beer brewing process is based on a microservice architecture [2] as depicted in Figure 1. Using this approach, the CMS is highly scalable and can run outside the production site, e.g., in a cloud. Still, on the customer site, the beer production system includes a wide set of machines, e.g., mill and mash tun, that are equipped with numerous sensors and which pass the measured values to different PLCs (Step 1). An industrial gateway collects and structures all values passed by the PLCs and forwards them to the cloud (Steps 2-3).

The CMS follows the software specifications defined in ISO 13374 Standard for Condition Monitoring and Diagnostics [1]. This standard defines six functional steps for condition monitoring where each has to be implemented as a software module. The six functional steps are: data acquisition, data manipulation, state detection, health assessment, prognostics, and advisory generation. Our condition monitoring software currently implements the first three

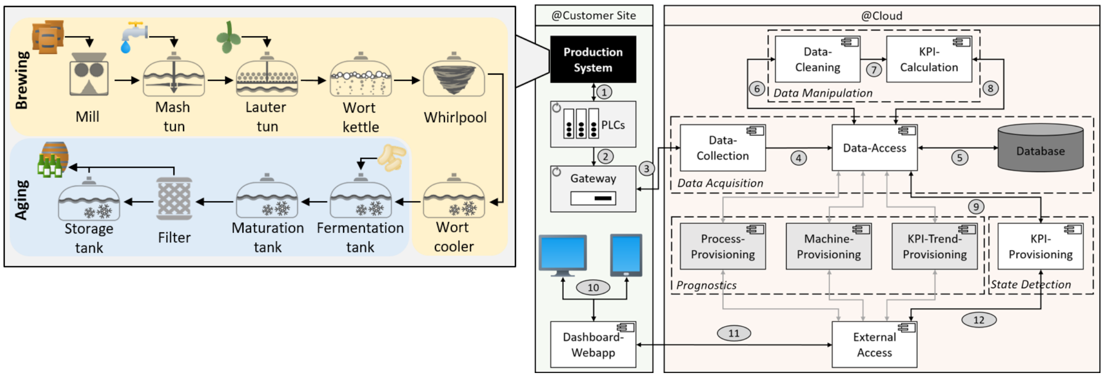

Figure 1: Microservice Architecture of the Condition Monitoring System

modules, whereas the prognostics module is under development. Each software module contains one or more software components implemented as microservices. For example, Data Acquisition module contains Data-Collection microservice that gathers sets of sensor data from the gateway (Step 3) and forwards them to the Database via the Data-Access microservice (Steps 4-5). The collected data is to this point unprocessed and needs to be refined within the Data Manipulation module in order to be understood by the brewery personnel. Data-Cleaning microservice removes incomplete or redundant unprocessed sensor data and prepares it for the calculation of Key Performance Indicators (KPIs) [3]. This action is done in the KPI-Calculation microservice. The calculated KPIs, e.g., Total Occupation Time and Overall Equipment Effectiveness, are stored in the database for further use as well as for traceability purposes (Steps 6-8).

State Detection software module provides the current status of the brewing production system via KPI-Provisioning microservice. This microservice queries the calculated KPIs and prepares profound reports for the brewery personnel (Step 9). For example, the production manager uses the Dashboard-Web-App microservice to get these reports (Step 10-12). The Dashboard-Web-App is designed to present the reports on different devices and platforms. However, the External-Access microservice filters the reports before forwarding it to the Dashboard-Web-App based on the user-role provided. In this way, user-specific views are generated to allow optimal and necessary insights for specific users-roles as well as to enforce data privacy.

Prognostics software module contains three microservices: Process-Provisioning, Machine-Provisioning, and KPI-Trend-Provisioning. Each microservice uses the Data Access module to query the data (unprocessed data and calculated KPIs) with the goal to project future brewing quality. For example, KPI-Trend-Provisioning uses machine learning techniques to estimate the Overall Mashing Time based on current data to prevent unintended behavior.

3 Conclusion

This modular, plug-and-play, approach of our microservice architecture allows easy integration of additional software modules, e.g., health assessment without the need of restructuring the software. Additionally, the approach allows scaling different microservices in order to provide better performance. As intended, our solution supports the use of multiple technologies for software realization. For example, to achieve optimal performance regarding the KPI-Calculation we use analytic libraries of the programming language R. Other microservices are realized using other programming languages.

References

[1] Dieter Hansen and Alf H Olsson. Iso standard 13373-1: 2002condition monitoring and diagnostics of machinesvibration condition monitoringpart 1: General procedures. *International Standards Organization*, 2002.

[2] Johannes Thönes. Microservices. *IEEE software*, 32(1):116–116, 2015.

[3] Benedict Wohlers, Stefan Dziwok, Faruk Pasic, Andre Lipsmeier, and Matthias Becker. Monitoring and control of production processes based on key performance indicators for mechatronic systems. *International Journal of Production Economics*, 220:107452, 2020.