

# Reflection Report Template

Student 1

University of Southern Denmark, SDU Software Engineering, Odense, Denmark

Email: student1@mmmi.sdu.dk

## I. Contribution

## II. Discussion

### A. Achieved Design Goals

In the proposed Industry 4.0 livestock farming system, the main design goal as stated in the paper is to achieve sufficient interoperability, availability, and deployability of the production software. After thorough review and discussion we concluded that the solution appears to be successful in complying to our quality attributes, supported by the architectural design choices and the use of technologies like Apache Kafka, Java-based subsystems, Docker and Hadoop Distributed File System.

1) Interoperability: The use of Apache Kafka and Java-based subsystems support well the interoperability requirement. The easy implementation in Java-based subsystems, the Java's ability to avoid dependence of existing operating system or hardware and the ability to connect and communicate with a wide variety of systems makes Java-based subsystems ideal interoperability choice. Moreover, Kafka's high compatibility with various systems and programming languages plays a crucial role in ensuring strong and on-going communication between subsystems.

2) Availability: The system's inclusion of Kafka's fault tolerance through data replication and Java's flexible memory management capabilities supports a high sufficiency of availability. This is critical for both sensor data traffic and storage, in which system reliability is crucial.

3) Deployability: Lack of modification need in Java-based systems and the use of Docker for containerization supports the continuous deployability of the system. Our design succeeds in supporting the requirement of the system to be adaptable to future changes such as increased livestock data without any significant downtime. Furthermore, Docker's ability to provide a consistent environment across different platforms is a vital element in achieving deployability.

### B. Challenges

While the solution achieves its primary goals, there are some challenges emerging:

Complexity and Resource Issues: The technological complexity of Kafka and the Hadoop ecosystem needs specialized knowledge and demanding system management.

Interoperability and Coordination Issues: As highlighted in multiple papers (1, 2, 3, 4, 5, 6, 9), a serious challenge in implementing smart farming technologies is to achieve

efficient interoperability. Coordination problems often occur due to the lack of common communication protocols or orchestration middleware, preventing the efficient use of shared data.

Bandwidth Limitations: Another challenge (1, 5) is the bandwidth limitation in rural areas. This highlights the functionality of large, interconnected systems essential for the modern farming.

## III. Reflection

This section will describe aspects of the project which hasn't been solved, and also our reflections on the process of the project work.

The full architecture of the project was not implemented, which unfortunately leaves out some of the functionality which was intended for the full system. Implementation of the full system would have allowed for a more complete experience of the architecture, but due to time constraints the focus was being able to evaluate on the required quality attributes. During the initial designing of the system, an overall architecture was created with the different systems and the message bus in the middle. It could have been a good idea to flesh out the subsystems further by including the use of views. Views have several forms, as described by Clements et al the documentation of our architecture must serve as a blueprint for construction [?]. The view presented in the project showed the overall architecture and how the systems communicated, but now exactly how each system would work essentially, this was only written down as a short description. This sometimes led to some miscommunication between group members, which could have been avoided by a more detailed description.

The project wasn't without its technical challenges, these challenges were also discussed in the main project as a tradeoff for some of the technologies used. The systems using Apache Kafka, Hadoop distributed file system (HDFS) and Hive took a significant amount of time to get working in a state that was acceptable. Also creating the connection between some of the systems, ended up consuming more time than what initially was planned. In the end it all resulted in the full architecture not being implemented.

## IV. Conclusion