Reflection Report Template

Stefan-Daniel Horvath*, Michael H Pedersen*, Jacob B Florczak*, Ignatios Mantanis*,
University of Southern Denmark, SDU Software Engineering, Odense, Denmark
Email: * {sthor23,micpe18,jaflo18,igman23}@student.sdu.dk

I. Contribution

This section will describe each group members contribution and what has been done together.

Section	Jacob	Michael	Ignatios	Stefan
Introduction and motivation	X			
Problem, research questions and approach		X		
Literature review	X	X	X	X
Use cases and quality attribute scenarios	X			
The solution	X		X	X
Evaluation				X
Future work	X			
Conclusion	X	X	X	X

TABLE I: Contribution table

Table I describes the work distribution of the group. Some parts have been worked on by multiple people, and some have mainly been worked on by one person. The table shows the main contributor(s) to each section. The Report has been read and edited by all group members meaning that only one person has been writing the section but all group members have been involved in the process of making sure that whats written is correct and understandable. Thus the distribution of work is not 100% accurate but it gives a good overview of who has been the main contributor to each section.

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, which describes the ability of two or more systems to exchange and use information [?]. 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 ongoing 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. The benefits of these systems ensure the requirements specified for the architecture, this is done by the technologies achieving the quality attributes by the use of different patterns and tactics.

Interoperability and Coordination Issues: As highlighted by the related work of the paper, 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. The group has been able to utilize Kafka for having an intermediate between the different systems.

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 were only written down as a short description. This sometimes lead to some miscommunication between group members, which could have been avoided by a more detailed description.

This point also leads into the fact that the systems generated for the architecture were generated by the group with some inspiration from related work. To fully grasp a system for smart livestock farming a project should consult true experts on the topic. In the related work section papers with machine learning was excluded, but the use cases of machine learning is as wide as ever and has a huge influence on current fields in technology [?]. It was out of scope for this paper, but should be considered in the future.

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

This short reflection report has highlighted the work contribution from each group member. It has also highlighted to what extend the design goals was achieved. The design goals was achieved by utilizing technologies that by the use of tactics and patterns were able to support the specified requirements. At last a reflection on the process of fulfilling the project, including what wasn't solved and some issues faced. This included the architecture not being fully implemented, reflecting on the possible use of views for a better understanding of the specific systems within the architecture. At last the need for expertise when deciding on the contents of the system should be considered for similar projects, and to finish off the section some of the techincal difficulties of the project is highlighted which was already known from the choosing of the technologies.