

Fog-to-Cloud Computing for Animal Farming: Towards Low-Cost Technologies, Data Exchange and Animal Welfare

Admela Jukan*, Francisco Carpio*, Xavi Masip**, Ana Juan Ferrer⁺, Nicole Kemper⁺⁺ and Birgit U. Stetina^x
Technische Universität Braunschweig, Germany*
Universitat Politècnica de Catalunya (UPC), Barcelona, Spain**
ATOS Research and Innovation, Barcelona, Spain⁺
University of Veterinary Medicine Hannover, Foundation, Germany⁺⁺
Sigmund Freud University, Vienna, Austria^x

Abstract

Coordinated fog-to-cloud computing systems are expected to expedite the evolution of cloud-based smart farming, towards openness and data sharing, while making farming economically sustainable for smaller farms. We demonstrate a fog-to-cloud enabled animal farming system that not only deploys low-cost technology for data collection and sharing, but also uniquely considers animal welfare and quality-of-life. On an practical example of a multi-master replication of SQL Server databases between Microsoft Azure Cloud and multiple fog nodes on the farm, we show that fog-based systems can help improve performance, scale to higher amounts of database entries, pre-process data with aid of machine learning, while providing reliability of the system in an easy-to-use fashion. Considerations of animal welfare, cloud-based data analysis, fog-enabled farming along with mobile applications are envisioned as common place for future farmers and consumers alike, and are in line with a recent trends connecting animal welfare, human health and the environment.

I. INTRODUCTION

Smart animal farming is an emerging high tech area focused on automating farm animal production, and thus reducing cost of human (manual) effort involved in daily tasks, such as feeding, milking or cleaning [1]. In line with the agriculture imperative of reducing costs of production, modern farming systems have recently began to offering cloud-based solutions as part of their big data features that can enable visualizing and analyzing data. The cloudification of this sector is rather significant, since the amount of data collected and analyzed exceed today’s abilities of various stakeholders to assess the quality of production; whether it is the quality of animal food products, or risk assessment by public health officials, analysis of data from livestock farming is becoming the norm. Since most of the smart farming systems and applications have primarily been developed in a proprietary fashion, focused on serving large corporations and not smaller farms, - fog (edge) computing concepts, another rapidly emerging computing technology paradigm, is evolving as a promising architectural solution to fill in the gap. The deployment of computing and storage infrastructures at the edge of the network, where data is generated, while its seamless interoperability with the cloud, is the fog’s main promise to improving the performance of the overall system [2].

There are a few new key areas of interest where fog computing can contribute to animal farming like no other solutions to date. First, low-cost and programmable systems, like those based on Raspberry-Pis (RPI), are the prime candidates for implementations in smaller farms, making the capabilities of monitoring, data collection and exchange transferable to smaller farms who cannot afford expensive proprietary systems. Second, fog computing can effectively leverage the capabilities of cloud computing systems, along with a customized and inexpensive implementation of mobile applications for farmers. At the same time, farmers can deploy fog computing alone, without reliance on the cloud. Third, data exchange and openness can be enabled as need based, and to various stakeholders, such as to veterinarians, health organizations, policy makers and consumers of animal products, which current system do not offer. Finally, important and currently missing features of animal welfare and quality of life can be integrated as farmers can afford them, and consumers request it. For instance, smart livestock farming can integrate animal welfare indicators along with production indicators, indicating that animals are healthy, free of pain, and positively stimulated in their environment, both physically and cognitively. This is in line with is a recent concept of “OneWelfare” that recognizes the direct and indirect links of animal welfare to human welfare (and environmentally friendly animal production systems) and the inter-connectedness of all three [3].

In this paper, we present and experimentally demonstrate a coordinated fog-to-cloud computing smart livestock farming system that can enable a transformative transition towards low-cost, open and inter-operable technologies, enabling data exchange and sharing, while contributing to an improved animal welfare and quality of life. We develop and integrate a mobile application for an open and accessible use of fog and cloud computing as its key system solution, while storing and visualizing data in a user-friendly fashion. The system enables farmers to document not only farm management related factors, which are important to economic sustainability, but also animal welfare of various species, including cows, pigs and chicken. We analyze data measured in a fog computing testbed, and the related system performance. On an example of a multi-master

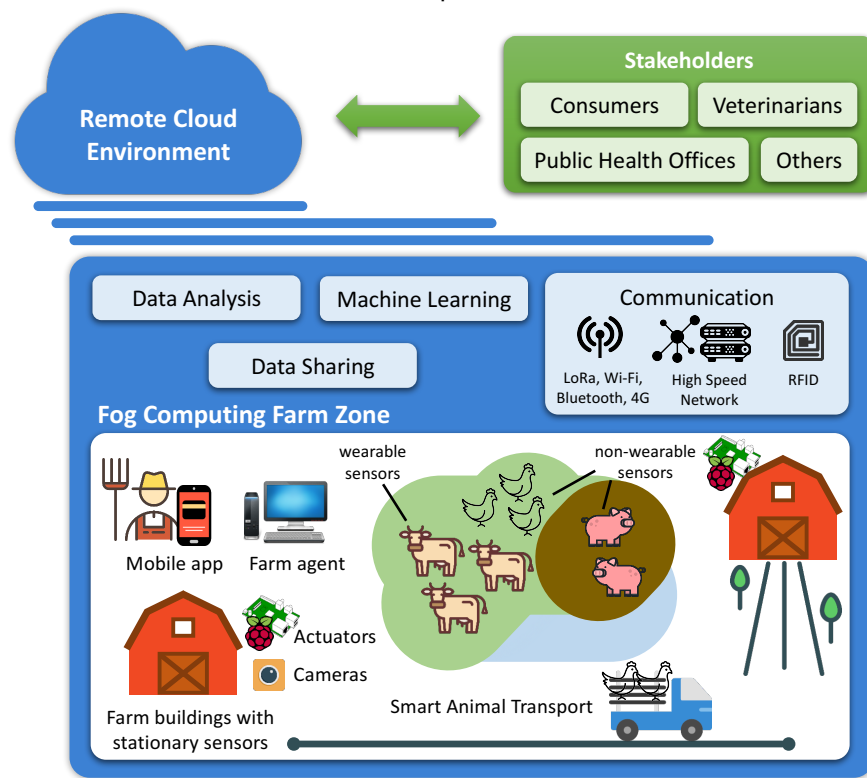


Fig. 1: Fog-to-cloud computing, data processing, and communication ecosystem for animal farming.

replication of SQL Server databases between Microsoft Azure Cloud and multiple fog nodes running on the farm, we show the fog-based systems also here help decrease the communication latency for data update, scale to higher amounts of database entries and number of animals, while providing modularity and reliability of the system and data in a seamless and easy-to-use fashion. We also analyze the effectiveness of machine learning techniques to assess and analyze indicators on animals and in the animal? environment and thus enable the farmer to document the quality of the animal?s life and its performance and act pro-actively in case certain thresholds are exceeded.

The rest of the paper is organized as follows. Section II gives briefly the background and motivation. Section III presents the system architecture. Section IV shows the real world experiments and analyzes the results. Section V concludes the paper.

II. BACKGROUND AND MOTIVATION

Low-cost, programmable and open systems have only recently been recognized as important candidate technologies for animal health monitoring [4]. Our earlier review on smart technologies [5] for animal welfare revealed the maturity and existence of various other non-commercial and programmable predecessor technologies, both in hardware and software, used to monitor physical health of farm animals, for all animal species. A vibrant commercial area of Precision Livestock Farming (PLF), on the other hand, has enriched the agriculture business sector with a myriad of specialized and proprietary end-devices and systems, that can enable agriculture to scale their production and reduce cost [1], [6]. As such, current systems are typically not open for experimentation, do not interoperate, and being in commercial domain remain difficult to analyze on performance, both from the system?s perspective, as well as from the perspective of animal-based indicators. This is where fog and cloud computing can jointly bring value, and lead to a new wave of high-tech innovations in this sector. There is a pressing need to make these technologies widely accessible, as the current policy makers urge all the stakeholders to "strengthen the links between animal welfare and the agricultural policy," according to the Special report No 31/2018, *Animal welfare in the EU: closing the gap between ambitious goals and practical implementation*, issued in by European Court of Editors, Guardians of EU Finances, in Nov 2018.

Fig. 1 shows our envisioned smart farming ecosystem with fog and cloud computing at its heart, and a variety of other architectural components enabling combined sensing, fog, cloud and networking technologies, as well as tools for data processing and analysis. Inside the fog computing zone, a variety of sensors and actuators is deployed both as wearables and non-wearables for animals and humans, or used in farm buildings (sheds), both stationary and during animal transport. The end-devices not only collect valuable data for analysis and processing, but can also serve as a distributed backup platform to the farm server, due to the, albeit limited, processing and storage capabilities of the boards like Raspberry-Pis, just like a centralized farm server would. The farmer can use a mobile application on a tablet or smartphone to manage the farm, and

1 retrieve relevant data both in real-time, as well as as data analyzed over various periods of time. All data collected can be
2 analyzed with data analysis tools, either in the fog zone, or in the cloud, or both [2]. For instance, cloud can be used to
3 periodically analyze any shared farm data, which in turn can be are pre-processed and analyzed locally on the farm. For some
4 data processing, fog computing can even provide the full cloud-like functionality, without reliance on the cloud. Farmers can
5 furthermore use either mobile apps or the farm servers to configure and monitor parameters of interest. For instance, various
6 stakeholders can be configured for data sharing over the cloud, such as veterinarians, and even consumers.

7 In terms of animal species, we postulate that fog-to-cloud computing is able to incorporate and correlate various indicators,
8 and be extensible like no other system. For instance, machine learning has been shown very effective in recognizing health-
9 or behavioral issues in pigs [7]–[9], or for using pig tracking as indicators of the spread of infectious diseases [10]. In a fog
10 system, in fact, this would require just a few low cost cameras, and machine learning can be used locally on the farm to
11 monitor health of individual pigs or the antibiotics intake. Such system can be useful to veterinarians who normally manually
12 analyze lengthy videos on animal behavior, or manually exchange data with farmers. As it is well-established that antimicrobial
13 resistance caused through usage of antibiotics in pigs poses a risk to human health, as concluded in Review on Antimicrobial
14 Resistance, report on "Tackling drug-resistant infections globally", dated 2016, as well as in an open-ed article in The New
15 York Times, "Health Leaders Must Focus on the Threats From Factory Farms" dated 2017, the effective data is critical. For
16 cattle, it is commonly accepted in farming that health can be monitored on wearable sensors on collars, lower limbs, or based on
17 RFID eartags, which can be further be augmented with an accelerometer. We believe ambitiously that coordinated fog-to-cloud
18 computing based systems can reduce or even eliminate the need for wearables devices on animals, and thus increase their
19 comfort, - due to the ability to process and analyze data locally. Based on the continuous monitoring on the farm, data collected
20 and information shared as well as human interactions with animals, and among animals themselves, the supporting systems
21 are expected to narrowing the knowledge gaps by bringing society closer to farm animals via direct information. Initially the
22 idea of connecting animal and human health was promoted in relation to companion animals but studies on livestock farming
23 get closer to the field of study of human-animal interactions and relationships. To give an example, animal experts have started
24 exploring farmers' attitudes regarding animal welfare and the concurrent outcomes. Specific psychological instruments and
25 trainings have been developed for farmers [11] to improve their attitudes on animal welfare. Interdisciplinary teams like ours
26 are working today worldwide on study designs to get deeper insight into connection between animal and human health.

27
28

III. SYSTEM ARCHITECTURE AND IMPLEMENTATION

29

A. System Architecture

30 The system architecture shown in Fig.2 incorporates two main abstraction layers: cloud computing and fog computing. In
31 addition, the architecture integrates a mobile application, which resides in both layers. The entire system includes five types
32 of agents: cloud-, farm-, shed-, animal- and administrative agent, the software modules designed for specific tasks on various
33 devices. For reliability purposes, all devices from the fog computing zone can interface with the cloud agent, however, by
34 default, only the farm agent does so. In the fog computing abstraction layer, the farm agent runs a service and a database to
35 manage the farm as a whole. The principle of reliability and modularity in the architecture applies. In other words, the shed
36 agents, located in every shed (farm buildings), are also able to manage the farm in case the farm agent fails. In fact, shed
37 agent and farm agent differ only in their ability to run the database, while having in common the implementation of the server
38 (here: RESTful Web Server, RWS). The animal agents run on wearable or non-wearable devices dedicated to monitoring of
39 individual animals. It should be noted, for illustration, that while the RPi that hosts an animal agent is exactly the same kind of
40 RPi hosting the shed agent, the former has battery limitations while the latter does not, which indicates that the animal agent
41 is optimized for low power consumption. Finally, we also implement an Android app which runs an administrative agent on a
42 smartphone (or tablet) and is used as a farmer-friendly interface to the farm management system. The admin agent can connect
43 to the farm both locally, though the fog computing system, or remotely through the cloud. For instance, a veterinarian visiting
44 the farm can connect to the system locally, based on proximity and share data locally with the farmer. It should also be noted
45 that the system architecture shown in Fig.2, only shows one agent per type, but the number of agents can be as many are the
46 number of sheds and animals in the farm to be managed. In addition, smaller farmers can create fog-based cooperatives, where
47 one farm is producing milk and the other cheese, creating a fog-based federation through proximity. Each farm is expected to
48 control its own set of devices and processing capabilities as well as provide direct connectivity to data sources, such as sensors
49 and actuators. Fig. 2 also shows the current implementation with the software and hardware technologies used in each agent
50 implementation, as we describe next.

51
52

B. Cloud Storage

53 In our cloud implementation, we demonstrate storage capabilities and complementary usage of the cloud and fog to provide
54 high reliability of data collected. To this end, the cloud agent includes a Web app and a SQL database, both running on
55 Microsoft Azure. To perform the common CRUD operations on the database, the web app allocates the PHP files that are
56 public and accessible to the fog computing smart farming zone. In the fog layer, the farm agent is the agent by default that has
57
58
59
60

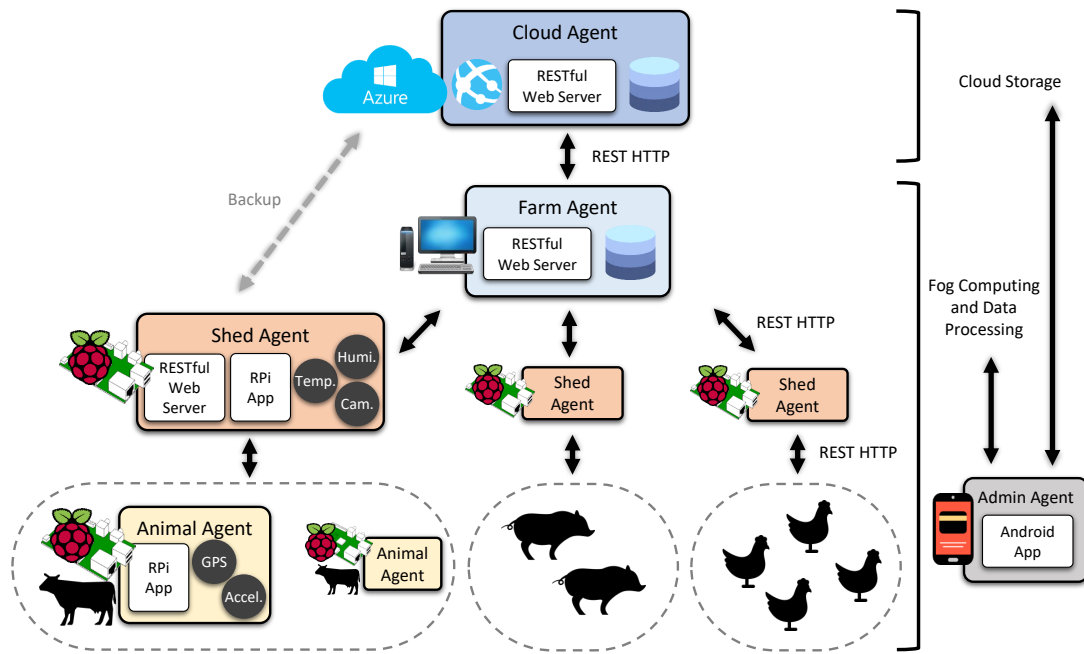


Fig. 2: System architecture and implementation.

full access to and the control of this database in the cloud. To access the data by using the PHP files, a REST API is designed on both sides, i.e., in the farm agent and in the cloud agent, which also uses REST HTTP protocol for the communication. To provide for data reliability, the architecture replicates the cloud database into the farm agent and the cloud following a master-master approach, being both able to read and write data simultaneously and synchronize the data periodically every 5 minutes. We also allow reading rights to the administrative agent, useful in a situation where the farmer is not on the farm premises, and needs remote access to the farm. For instance, administrative agent is not able to register or un-register cows when not on the farm premises, but is able to monitor everything else on the farm.

C. Fog Computing Agents

The farm agent runs a RESTful Web Server (RWS), which we also refer to as farm server, which is microservice built on SpringBoot in an off-the-shelf computer server located in the farm. This agent is not designed for any sensing or actuating functions, and its main role is to manage and coordinate the physical devices in the fog computing farm zone and use the local database as stand-alone storage system. In this way, the smart farming system can operate as an independent entity from the cloud. The RWS includes all the functions related with the administration of other agents (shed, animal) which can be performed through the admin agent on a smartphone or tablet.

The shed agent is implemented in every farm building on a RPi with an attached Sense HAT module that includes an integrated circuit-based sensors. As such, it can be used for many different types of applications. In our implementation, it measures the temperature and the humidity, it provides a LED matrix display to signal alarms, as well as a camera module for video streaming capabilities. Since the RPis used to monitor shed data are fixed (and not wearable), there are no major limitations on power consumption. It should be noted that this agent also has a RWS instance running in background to process the data when the main farm agent fails, or is not reachable (denoted as dotted backup interface in in Fig.2).

The animal agent, which in our implementation also runs on a RPi, integrates an accelerometer, for the detection of animal movement, and a GPS module, to determine the animal location outdoors. To reduce the communication overhead and extend battery life, only when the movement and location status change with respect to a pre-defined threshold, this information is sent to the server. Since this RPi is wearable, we concentrate our efforts to reducing the power consumption by reducing the communication with the server, which is critical and processing the data as locally as possible. That is why we often call this entity "Fog Processing," meaning that the raw data that is locally processed whenever possible, i.e., close to the measured entity, while at the same considering practical hardware constraints of the low-cost devices.

D. Communication

All software clients (agents) use REST HTTP for communication with each other, whereby the platform controller is running a RESTful Web Server. REST HTTP is a stateless communication approach that enables flexibility by using Universal Resource Identifiers (URIs). REST generates more overhead compared to other protocols specially designed to run in constrained devices,

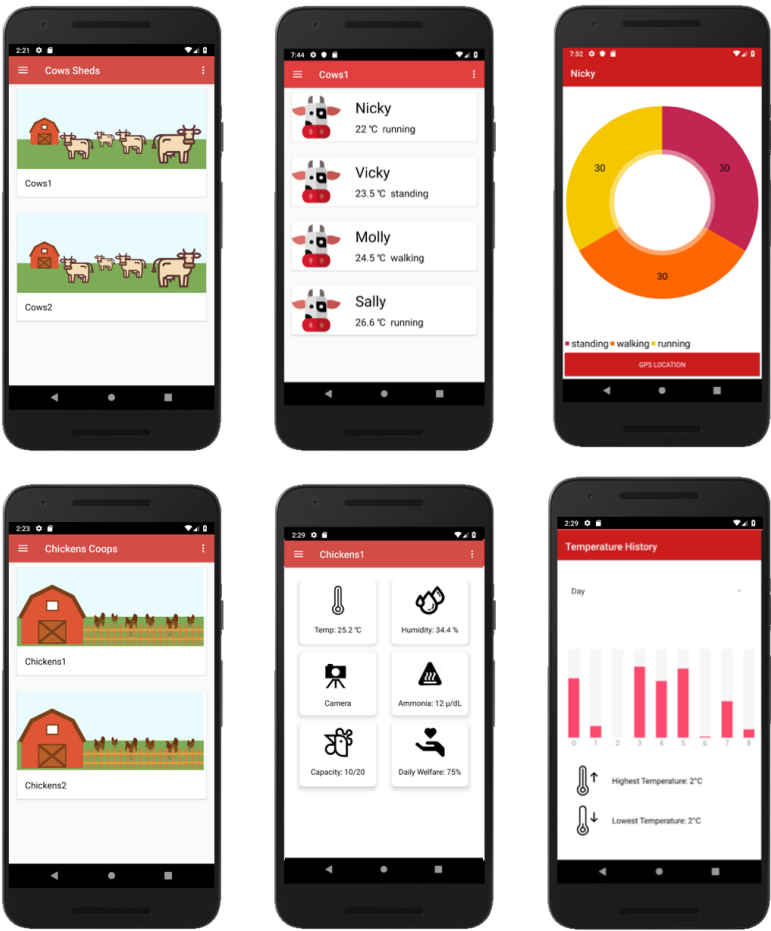
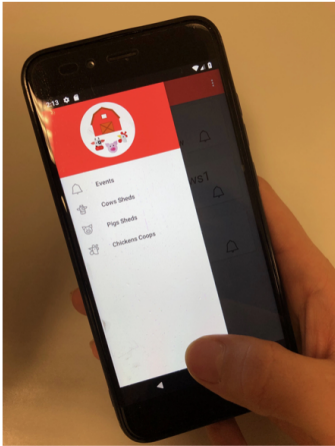


Fig. 3: Screen shots of the Android application menu.

but increases flexibility since is able to work in scenarios where different hardware and operating systems coexist, reducing the programming efforts significantly. Moreover, the stateless property helps to decrease the power consumption since the client remains completely disconnected from the server when there is no communication. In our system, the objective is to minimize the number of times that a animal agent has to open a new connection in order to reduce power consumption. Finally, it should be noted that for the video streaming between the shed agent and the smartphone, Real Time Streaming Protocol (RTSP) protocol is used. This choice of communication protocol is reasonable where there are no limitations on power consumption.

E. Mobile App

The mobile app has been developed for Android devices using the Android SDK platform 25. As auxiliary libraries, MPAndroidChart is used for the representation of data through dynamic charts and Srping for Android for the communication with other agents. The app is used as an interface to manage all the configuration parameters in the farm while also enabling data sharing and data storage in the cloud, for a more global data analysis. The main goal of the app is to enable farmers to visualize and interact with the farm in real-time, as well as to consider the historical data as important indicators. The mobile app is designed to work both locally, on the farm and without the cloud, and remotely, i.e., in the cloud.

Various menus in the mobile app are illustrated in Fig. 3. As it can be seen, different sheds registered in the app are classified by animal species. If we selected one of the sheds, we would open the shed activity, for instance, where the farmer has access to a live video stream to monitor the specific shed on temperature and humidity. In the app, all animals per shed can be monitored either individually or as group. Here, the critical information is updated in real-time and individual details of each animal can be visualized by selecting the animal, for instance, by name. We also illustrate the animal welfare parameters, such as daily physical activities for cows, including the animal location (which can be close to feeding, milking station or brushing "wellness" station). Similarly, a menu exists also for chicken and pigs, where the data collected is different, and can, among others related to social playing, for instance, for young pigs, or space and light consideration for chicken.

TABLE I: Time Average per Operation in Milliseconds

	Cloud_DB	Farm_DB	Farm_RWS	Shed_RWS
Create	535.356 \pm 16.9	23.259 \pm 12.8	13.454 \pm 16.8	25.003 \pm 18.7
Read	262.625 \pm 2.4	3.456 \pm 0.02	1.586 \pm 0.02	4.794 \pm 0.08
Update	558.730 \pm 42.0	36.057 \pm 12.2	40.216 \pm 0.11	227.952 \pm 63.70
Delete	553.880 \pm 79.0	4.378 \pm 0.06	1.453 \pm 0.1	7.179 \pm 0.5

F. Adding Tools and Methods

While cows can be identified individually with aid of wearables, pigs and hens for instance, can benefit from non-wearable sensing for monitoring of their behavior and health. We have implemented and experimented with applications of ML for pig and chicken recognition and identification. In one of our scenarios, we use TensorFlow Object Detection API and a Convolutional Neural Network combined with the Single Shot Detection technique for the measurement and analysis of the movement of individual hens. Our solution makes use of an already pre-trained model used for detection of movement objects, but modified in order to detect chicken. This method, known as Transfer Learning, allows to retrain models that were used for another objectives but with equivalent functionality. We tuned this model to work with labeled data for chickens and we trained it using the farm agent, where the computing capabilities are not limited. Once the model is able to recognize chickens from a video in the Farm agent, we transferred the model to the RPi, where the input source is not a video, but a live feed captured in real-time from the attached camera module. Then, the RPi is only used for evaluation, which consists on determining how many chickens are on captured frames from a live feed. To achieve that, the model will try to find patterns in a frame that matches with previous learned patterns during the training process. The number of analyzed frames, frame resolution, brightness, etc, will determine the effectiveness of the algorithm but also will have implications on the computing time. The results from the algorithm determines the number of chickens and their movements, sending this processed data to the farm agent, avoiding the transmission of the live feed to the farm or the cloud. This has some benefits, not only because of the reduced amount of data to transfer, but because of the reduced latency perceived by the users located on the farm.

IV. SYSTEM PERFORMANCE

In this section, we illustrate a few performance metrics. First, we show fog-to-cloud coordinated data processing, where fog and cloud computing are used in a highly complementary fashion. Second, we show how the system can scale in terms of number of cows, and data managed, by analyzing the system throughput in number of data base operations per second. Finally, we experiment with the effectiveness of machine learning for detection of chicken. We measure the performance in a real world laboratory setting, which has been also tested on its applicability in an experimental farm Ruthe, see [12]. In this case study, we consider a typical tasks in a diary farming. The system setup is shown in Fig. 4. As it can be seen, the data collected are related to posture (animal health), location (animal activities) and actuation (open a gate). The posture of the animal is estimated based on accelerometer values, while the position is recorded based GPS. Other sensors have also been tested, including temperature, humidity and activity (feeding, brushing, milking).

Once data is collected from the sensors, the RPi running an animal agent sends the data, over a wifi connection established by a Linksys WRT54GL wifi router, to the RESTful Web Server (Farm_RWS) running on a desktop computer with an i5-6500 CPU and 16GB of ram, running a farm agent. The farm agent is responsible for storing the data into the local SQL database (Farm_DB) which is synchronized using a master-master configuration with another SQL database in the cloud (Cloud_DB). In case of failure of farm agent, the animal agent sends the data to the shed agent (running on a RPi located in the shed) which also runs a RESTful Web Server instance (Shed_RWS). In case that only the local database fails, either the farm or the shed RWS instances are able to connect to the cloud database trough the Web App hosted in Microsoft Azure (<http://smarthof.azurewebsites.net/>). Because the system is developed modularly, there is no a unique sequence of these operations, but the system adapts depending on the states, which is its salient feature. All the evaluations have been performed by using Java Microbenchmark Harness (JMH), scheduled to become part of the upcoming Java 9 release, over 1000 iterations.

The Table I shows the time average to perform a single HTTP request of type: POST, GET, PUT and DELETE. For the farm database, the worst case of communication delay is the update (in real-time) of an entry. In case the CRUD operations are performed directly to farm server, the results are better as compared to the case when they are performed into the database. This behavior is as expected because the server works with volatile memory, while the database works in cold storage mode. In case the farm agent fails and the shed server becomes active, serving as temporary farm agent for continuous fog service, the delay deteriorates, specially in the case when the RPi has to update information.

Fig. 5a illustrates the number of update (real-time) requests that can be successfully performed per second when there are 10000 (large number) registered cows in the farm. These results show the performance of the farm depending where the information is being updated. For instance, if the system is connected to the cloud only, it is able to update one single parameter (per second) of ≈ 2 animals. When the system is working normally, the database of the farm agent is chosen as default to

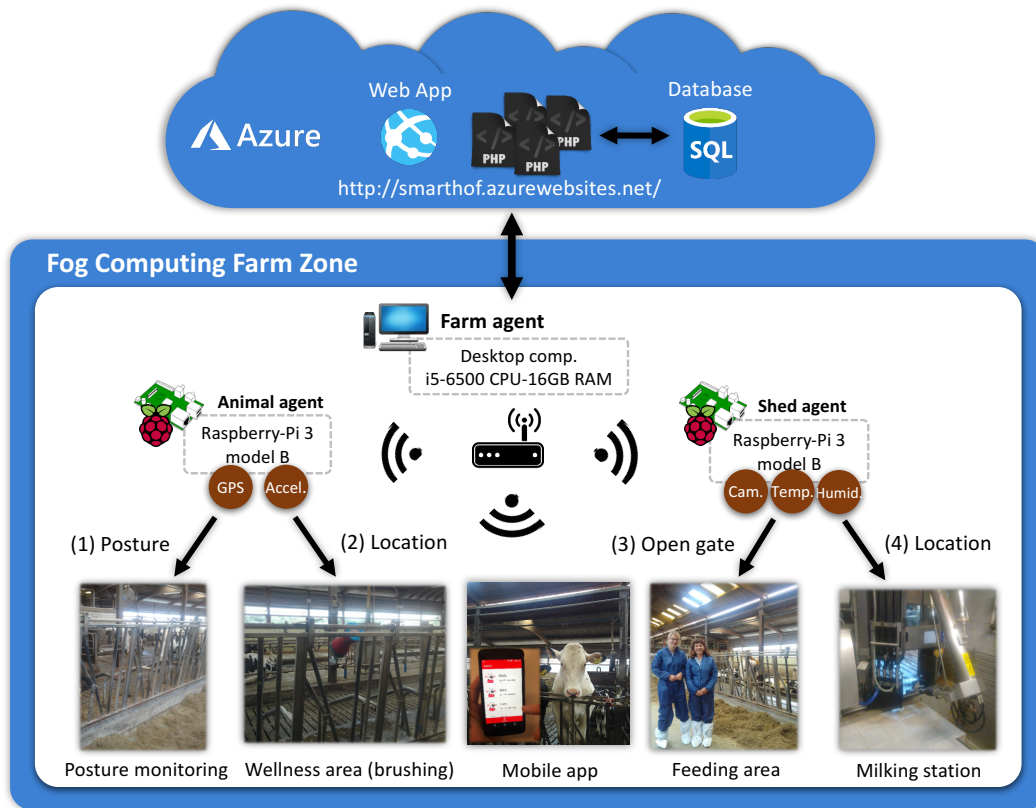


Fig. 4: Experimental setup of the system.

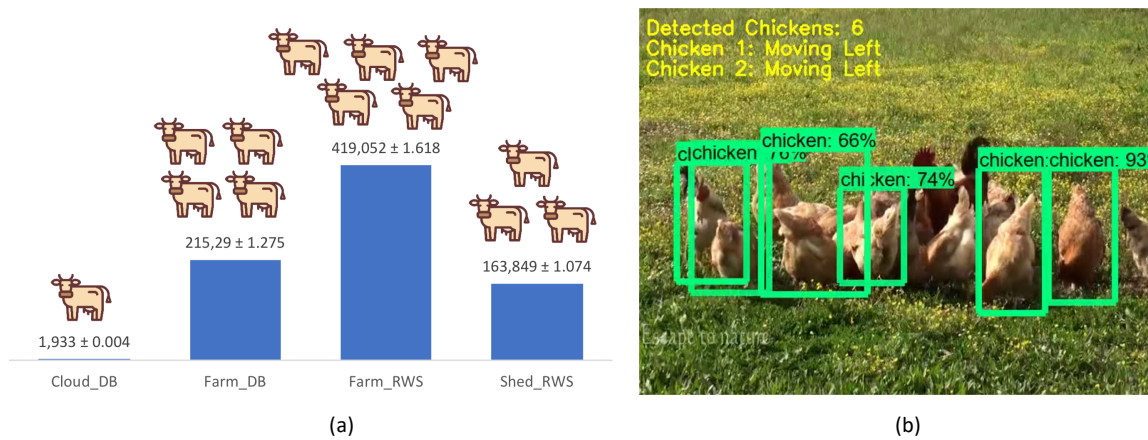


Fig. 5: (a) Data base update requests scalability per animal, and (b) effectiveness of ML model output for chicken.

store the information. In this case, the system is able to update the information of ≈ 215 animals per second. This result shows that, even with limited fog resources, when the database is on the farm premises, the system improves the throughput achieved in comparison with the cloud-based operation. In case the update requests are performed by the farm agent, the number of managed requests can be doubled (≈ 420) as compared to the previous case. It should be noted that the RWS manages the database in the background, therefore if the server breaks the updated information will remain in cold storage. In this case, the server can be seen as a queue of requests able to response faster than the database. It is therefore that RWS shows the throughput and scalability results superior to the DB solutions, albeit databases are deemed essential architecture components in both cloud and fog. Finally, in case the farm agent fails and, therefore, the shed agents have to be used as farm service (RWS), a shed is able to support ≈ 160 animal updates per second. This latter result is interesting, because a few RPis can deliver the performance of a standard desktop computer, which can save costs, since the farmer does not need to buy expensive servers running only as backups.

In terms of ML applications for non-wearable sensing and to find optimum functional results, significant implementation effort on the farm is needed to make reliable conclusions. We experimented with our ML detection model for chickens with different values of the network's hyperparameters. Our optimal version was trained with 300 images in 200k steps with a learning rate of 0.001, batch size 8 and allowing dropout. We tried with higher learning rate, which reduces the training time, but the model failed to converge. We evaluated the accuracy of the model by calculating the ratio between the detected chickens and the total number of present chickens at all times. As input for the model we analyzed a video of 60 seconds, taking 10 frames per seconds which results in 600 frames analyzed in total, with 640x480 pixel resolution per frame. Four different videos are evaluated with 1, 2, 4 and 8 chicken (images), achieving an accuracy of 92, 74, 59 and 32%, respectively. An example of the output from model is shown in Fig.5b, where the model is able to detect 6 chickens among many others. Here, we can see how the accuracy of individual recognition decreases when increasing the number of chickens, which is due to the maximum resolution that a RPi can support, the overlapping between the chickens, illumination conditions, etc. Similar directions for further improvements are in applications for pig farming.

V. SUMMARY AND OUTLOOK

The first gap that we considered important to bridge was our collective inability to experiment with today's farming systems at scale, and analyze and share relevant data on animal welfare. To this end, we designed, implemented and demonstrated a coordinated fog-to-cloud computing system with low-cost technology for data collection and data sharing along with an Android-based mobile application for farmers with applications of animal welfare. There are with no doubt a myriad of further technological and engineering challenges, in addition to social and economic ones, that we need to address and overcome in the future. They include the standardization of communication interfaces, robustness of low-cost systems, interoperability of proprietary and open systems, consideration of human-in-the-loop factors in combination with artificial intelligence methods, and adoptability by farmers and animals. There are many more opportunities though. With standardized interfaces, such as RESTful, fog computing can integrate not only the cloud computing but also proprietary systems, fostering interoperability. Fog-based systems with their connection to cloud-based data analysis are easy and inexpensive to implement, and carry potential for creating a new innovation business sector. We also see a significant potential of fog-based processing and computing to reducing the need for wearable devices on animals, as fog devices can be collectively used for non-intrusive sensing with aid of machine learning, which not only reduces cost, but also increases the level of comfort and safety for animals, freeing them of wearable and potentially hazardous communication hardware. Finally, the complex relationship between human- and animal welfare can be easily and jointly captured, resulting in healthier animals, healthier consumers and healthier environment.

VI. ACKNOWLEDGEMENTS

We thank the experimental animal farm "Lehr- und Forschungsgut Ruthe," in Hannover, Germany. We thank our students Marta Delgado Lopez and Zied Naceur for programming and testing efforts. This work has been partially supported by European Union's H2020 Research and Innovation Programme, under grant agreement 730929, "Towards an Open, Secure, Decentralized and Coordinated Fog-to-Cloud Management Ecosystem (mF2C)."

REFERENCES

- [1] T. Banhazi, H. Lehr, J. Black, H. Crabtree, C. Schofield, M. Tschärke, and D. Berckmans, "Precision livestock farming: An international review of scientific and commercial aspects," vol. 5, pp. 1–9, 01 2012.
- [2] X. Masip-bruin, E. Marin-Tordera, G. Tashakor, A. Jukan, and G. Jie Ren, "Foggy Clouds and Cloudy Fogs: A Real Need For Coordinated Management of Fog-to-Cloud Computing Systems," *IEEE Wireless Communications*, no. October, pp. 120–128, 2016.
- [3] R. G. Pinillos, M. C. Appleby, X. Manteca, F. Scott-Park, C. Smith, and A. Velarde, "One welfare – a platform for improving human and animal welfare," *Veterinary Record*, vol. 179, no. 16, pp. 412–413, 2016. [Online]. Available: <http://veterinaryrecord.bmj.com/content/179/16/412>
- [4] D. T. M. L. Narayan and D. A. Nagappan, "Animal health monitoring system using raspberry pi and wireless sensor," in *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, ser. International Journal of Scientific Research and Education (IJSRE), March 2015, pp. 1000–1010.
- [5] A. Jukan, X. Masip-Bruin, and N. Amla, "Smart computing and sensing technologies for animal welfare: A systematic review," *ACM Comput. Surv.*, vol. 50, no. 1, pp. 10:1–10:27, Apr. 2017. [Online]. Available: <http://doi.acm.org/10.1145/3041960>

1 [6] J. Gray, T. M. Banhazi, and A. A. Kist, “Wireless data management system for environmental monitoring in livestock buildings,” *Information*
2 *Processing in Agriculture*, vol. 4, no. 1, pp. 1 – 17, 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S2214317316300348>
3 [7] J. Lee, L. Jin, D. Park, and Y. Chung, “Automatic recognition of aggressive behavior in pigs using a kinect depth sensor,” *Sensors*, vol. 16, no. 5, 2016.
4 [8] M. Oczak, G. Ismayilova, A. Costa, S. Viazzi, L. T. Sonoda, M. Fels, C. Bahr, J. Hartung, M. Guarino, D. Berckmans, and E. Vranken, “Analysis
5 of aggressive behaviours of pigs by automatic video recordings,” *Computers and Electronics in Agriculture*, vol. 99, pp. 209 – 217, 2013. [Online].
6 Available: <http://www.sciencedirect.com/science/article/pii/S0168169913002433>
7 [9] S. G. Matthews, A. L. Miller, J. Clapp, T. Pltz, and I. Kyriazakis, “Early detection of health and welfare compromises through
8 automated detection of behavioural changes in pigs,” *The Veterinary Journal*, vol. 217, pp. 43 – 51, 2016. [Online]. Available:
9 <http://www.sciencedirect.com/science/article/pii/S1090023316301538>
10 [10] J. Haladjian, A. Ermis, Z. Hodaie, and B. Brügge, “ipig: Towards tracking the behavior of free-roaming pigs,” in *Proceedings of the Fourth*
11 *International Conference on Animal-Computer Interaction*, ser. ACI2017. New York, NY, USA: ACM, 2017, pp. 10:1–10:5. [Online]. Available:
12 <http://doi.acm.org/10.1145/3152130.3152145>
13 [11] H. Hansson and C. J. Lagerkvist, “Measuring farmers’ attitudes to animal welfare and health,” *British Food Journal*, vol. 114, no. 6, pp. 840–852,
14 2012. [Online]. Available: <https://doi.org/10.1108/00070701211234363>
15 [12] F. Carpio, A. Jukan, A. I. M. Sanchez, N. Amla, and N. Kemper, “Beyond production indicators: A novel smart farming application and system for
16 animal welfare,” in *Proceedings of the Fourth International Conference on Animal-Computer Interaction*, ser. ACI2017. New York, NY, USA: ACM,
17 2017, pp. 7:1–7:11. [Online]. Available: <http://doi.acm.org/10.1145/3152130.3152140>
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60