International Journal of Cloud Computing and Services Science (IJ-CLOSER)
Vol.3, No.3, June 2014, pp. 133~143
ISSN: 2089-3337

Exploring Different Architectures to Support Crop Farmers with a Mobile Application on Pesticide Control

Richard K Lomotey, Ralph Deters
Department of Computer Science, University of Saskatchewan, 110 Science Place, Saskatoon SK S7N 5C9, Canada

Article Info

Article history:

Received Apr 15th, 2014
Revised May 25th, 2014
Accepted Jun 10th, 2014

Keyword:

Mobile devices
Middleware
Offloading
Partitioning
Pesticides
Mobile Cloud Computing
Cloud Computing

ABSTRACT

The MobiCrop app, which is a distributed mobile application has been proposed to aid crop farmers with timely decision making on the applicability of pesticides (i.e., which pesticide to apply, when, where, and how to apply them). Due to the vast amount of pesticide and crop data, the application is designed following the three-tier architecture technique which comprises the mobile devices, a cloud-hosted middleware, and cloud-based database. The idea is to enable the mobile device to retrieve the needed pesticide data from the back-end and when necessary, part of the data can be stored on the mobile through caching for offline accessibility. However, constantly updating the mobile cache through data polling is costly for the wireless bandwidth and energy usage on the mobile. Also, it is difficult to update the stale cache data when there is no wireless connectivity. Hence, this work explores three architectural designs of the MobiCrop app which are the: 1) the standalone (network independent), 2) distributed architecture through data offloading, and 3) distributed architecture through data partitioning.

Corresponding Author:

First Author,
Department of Computer Science,
Universitu of Saskatchewan,
Saskatoon, Canada, S7N 5C9.
Email: richard.lomotey@usask.ca

1. INTRODUCTION

The adoption of mobile technology in the agriculture sector is increasing. Mobile technology can aid farmers to access timely information on agricultural products and services, monitoring, mobile agriculture commerce, and so on [2]. In an ongoing work with the College of Agriculture at the University of Saskatchewan, Canada, the MobiCrop [25] application was proposed. The main goal of the project is to support crop farmers to use their mobile devices to access timely information on pesticides. The crop farmers will have to be facilitated to access the pesticide control data on how, when, where, and what chemicals should be applied. The application is a distributed mobile cloud architecture that enables the user (farmer) to access the pesticide information on the mobile from the cloud-hosted back-end. Cloud computing facilitates services outsourcing from third parties (known as cloud providers) over the Internet [1]. The services are categorized into three major groups as:

- *Infrastructure as a Service (IaaS)*—where hardware and networks are offered as virtualized services,
- *Platform as a Service (PaaS)*—where application development is hosted by the provider, and

The initial work is a best paper present at SOCA 2013 and is cited as: Lomotey, R. K., Chai, Y., Jamal, S., and Deters, R. 2013. MobiCrop: Supporting Crop Farmers with a Cloud-Enabled Mobile App. Proc. of the 2013 6th IEEE International Conference on Service Oriented Computing & Applications (SOCA 2013), pp:8 pages, Kauai, Hawaii, USA, December 16-18, 2013.

- *Software as a Service (SaaS)*—where software is made usable to consumers (users) by service providers [3-5].

In the work of Hori et al. [1], the authors show how cloud computing can affect the agricultural sector. Their view is summarized in Figure. 1.

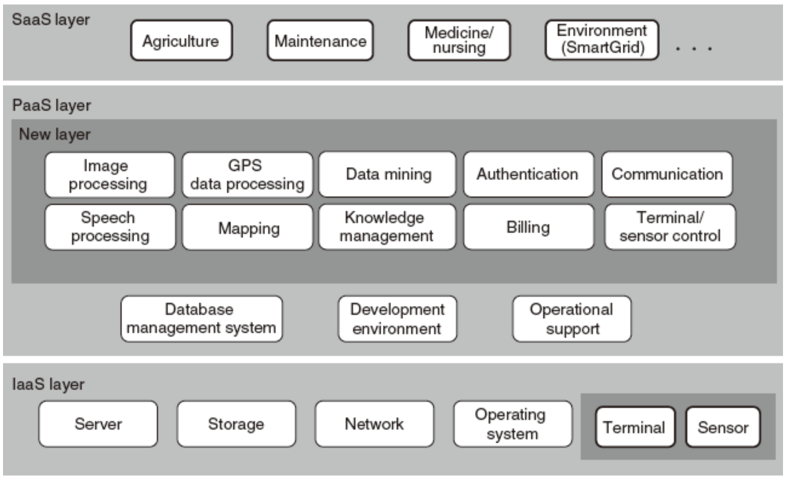

Figure 1. Cloud taxonomy for the agricultural space [1]

However, the fact that the MobiCrop architecture is distributed poses some concerns that require further studies. Mobile devices orthodoxly rely on wireless mediums (e.g., Wi-Fi, Bluetooth, 3.5/4G, etc.) to communicate and these mediums can experience intermittent connection loss as well as bandwidth fluctuations. Also, energy conservation on the mobile is important to support the farmers to work on the field.

Thus, we investigate the design of three different architectures and their ability to support offline usage and battery life management. These architectures are: 1) the standalone (network independent), 2) distributed architecture through data offloading, and 3) distributed architecture through data partitioning. The standalone architecture is where the entire application state is stored on the mobile and therefore requires no network connectivity to function fully. The distributed architecture through data offloading is where we stored the entire data state on the back-end and the mobile device relies on polling and pushing techniques to retrieve the information from the cloud-based back-end. The distributed data partitioning is where we shared the data across the mobile and the back-end nodes. The evaluations show that while the standalone app is better for overcoming network related challenges, its energy consumption is higher. The distributed architectures work better with energy consumption but data offloading has higher latency and consumes high bandwidth in comparison to data partitioning. The details of the results are in Section IV.

The remaining sections of the paper are as follows. The next section explains some mobile cloud computing techniques and Web services. The architectural designed of MobiCrop is discussed in Section 3. Section 4 discusses the implementation and the evaluation of the architecture and the paper concludes in Section 5 with our contribution and future direction.

2. LITERATURE REVIEW

2.1. Mobile Cloud Computing

Cloud computing is the technological model that enables consumers to outsource computing services from third parties over the network. Moreover, smartphones and tablet devices have established themselves as the dominant nodes to access and consume digital assets [6]. The fact that cloud computing is a backend service and offers anytime access aligns very well with the mobile field which complements the cloud by providing anywhere access. However, there are certain challenges that have plagued the sustainable deployment of mobile cloud computing services. These challenges are: unstable wireless networks, limited bandwidth availability, mobile device feature constraints, and very low energy budget that is dictated by battery life. The solutions to some of these challenges have led to wide-array of studies. For brevity, Table I is provided to highlight the research directions in mobile cloud computing on overcoming some of the challenges. For detail review on mobile cloud computing architectures, the reader is referred to [26]. *Computational offloading* is the delegation of computing intensive demands to the cloud in order to free the resources (especially the CPU) on the mobile. *Partitioning or Sharing* is delegating part of the computation or service to the cloud while keeping the remaining on the mobile node.

Table 1. Offloading and Other Techniques in Mobile Cloud Computing

Method	Existing Framework/Approach	Benefits	Authors
computational offloading	Functional replication, code block execution,	Management of mobile resources	Bahl et al. [7]
	ECOS [8]: delegation of resource intensive tasks to the cloud		Gember et al. [8]
	CAM−apps [9]: task distribution in mobile cloud networks	Remote service hosting, services management and allocation	Ferber et al. [9]
	Resources allocation	Mobile battery life extension	Ge et al. [10]
Computation sharing/partitioning and Middleware	***MCM*** [11]: consuming cloud processed services from multiple sources	Achieves high interoperability when integrating multiple non-standardized APIs	Flores [11]
	Figaro [12]: modelling the mobile nodes as agents and using policies to coordinates their activities	Load distribution	Malandrino et al. [12]
	Adopting middleware techniques	Data synchronization	Lv and Zheng [13]
			Yang [14]
	Synchronization Server based on caching	Offline access	Xue [15]

2.2 Mobile Technology in Agriculture

There are several ongoing works on promoting mobile technology usage in the field of agriculture. Tan et al. [16] show that mobile devices can play integral role in labor monitoring. The authors proposed an architecture that comprises of data acquisition layer and processing systems. Overall, the labor workers can be monitored from several crop fields and their wages calculated. The *AgroMobile* architecture proposed by Prasad et al. [17] combine mobile technologies with cloud computing to benefit agriculture. The authors opined that India as a country loses a lot through conventional crop information gathering which are sometimes slow and unreliable. The authors therefore proposed the AgroMobile framework which enables the cloud servers to do image analysis which will be intensive on the mobile. This is a typical offloading approach.

Cho et al. [18] in their work also identify the importance of using mobile cloud computing technologies to disseminate information to the farmers quickly. Cloud computing in their view can promote ubiquitous access to crop and pests' data. Papadavid et al. [19] further show mobile technologies can be employed to deliver real-time information to farmers in Cyprus, where water allocation to crops is an issue due to droughts.

Kim et al. [20] also propose the self-growing agricultural knowledge cloud that offers the farmer to make smarter decisions. This agriculture cloud consists of federated data sources from farmer groups, developer groups, researcher groups, and companies.

Other works on the employment of mobile devices to access agriculture related information in an attempt to promote crop yield can be seen from the works of Steinberger et al. [21], Athanasios et al. [22], Wang et al. [23], and Sugahara [24].

2.3. Research Goal

There are two main issues that we seek to investigate in this work. These are:

i. Energy management in the mobile network, and
ii. Supporting offline accessibility of the pesticide data

These two issues are important for the crop farmers who spend time in network volatile zones as well as not having regular access to electricity. To address these issues, we shall discuss three separate architectures; the standalone architecture, the distributed data offloading architecture, and the distributed partitioning architecture.

3. THE PROPOSED ARCHITECTURAL DESIGNS OF MOBICROP

The actual data that is presented to the farmers is from the Saskatchewan Ministry of Agriculture's "2014 Guide to Crop Protection[1]" on weeds, plant diseases, and insects. There are three major classifications in the guide that we seek to convert into a mobile application for the farmers. This is shown in Table II.

Table 2. Main Data Points for Consideration

Weed Control	Plant Disease Control	Insect Control
Integrated Weed Management	Integrated Plant Disease Management	Insect Control Decisions
Making Spray Decisions	Effects of Weather	Field Scouting
Weed Resistance to Herbicides	Resistance Management	Insecticides and Bees
Adjuvants	Fungicide Mode of Action	Insecticide Poisoning in Humans
Crop and Herbicide Recommendation Tables	Foliar Fungicide Tables	Resistance of Insects to Insecticides
Special Weed Problems	Foliar Fungicide Product Pages	

The coverage of the entire data in Table II occupies a book of over 500 pages with charts, images, comparative tables, and text. Based on our goal, we present three architectural designs that can aid us to deliver the data to the crop farmers even in offline mode and also manage the energy consumption efficiently.

3.1 The Standalone Architecture

The proposed standalone architecture is aimed at enabling the hosting and deployment of the MobiCrop App on the mobile as a self-contain application. This means, the entire business logic, storage, and presentation of the App reside on the mobile. The architectural design is shown in Fig. 2. The entire architecture comprises three major components which are the: User Interaction Layer, Business Logic, and Storage. The components run on top of the mobile OS layer.

User Interaction Layer: This is the user interface view that enables the user to interact with the App. The subcomponents of this layer are discussed as follows.

Application User Interface: This is the actual view that presents the user with UI elements. The buttons, images, query results, etc. can be viewed here. The layer is designed based on some supporting technologies such as HTML5, JQuerymobile[2], CSS, and JavaScript. We relied on these technologies because the architectural deployment follows the Web standard.

Business Logic: The main process flow of the App is coordinated by the business logic layer. Some of the components work with the user interface layer and others with the storage layer. The subcomponents of the business logic layer are:

- *Query*: This is where we write the SQLite queries that determine what pesticide data to retrieve.
- *Image*: Component dedicated to processing image files. This includes resizing and storage of the images.
- *Table Formulation*: The data that requires to be presented in tabular formats are processed here. In the cases where the tables are big, only specific cells and/or rows are processed to be shown.
- *Modules*: These are application specific codes that work with the native features of the mobile phone. Examples are camera, local data storage, file storage, etc.
- *Storage Operations*: This is where the initial actions that require data storage are carried out.
- *Data Fetching*: Reading pesticide data from storage and presenting it to the user
- *Data Creation*: Storing new information into the storage

[1] http://www.agriculture.gov.sk.ca/Default.aspx?DN=5be29ef9-e80c-4ebd-b41d-d8e508b5aaba
[2] http://jquerymobile.com/

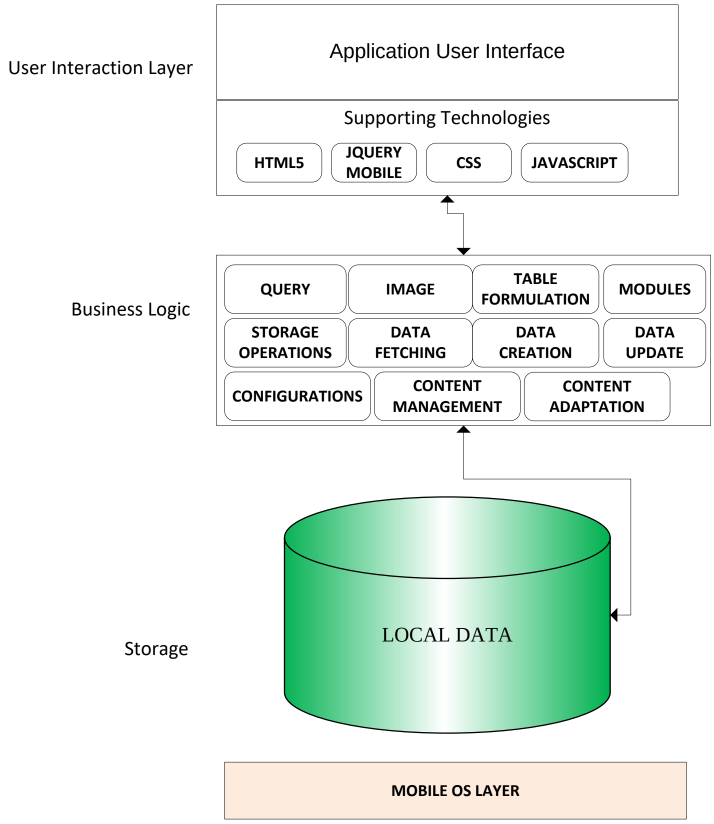

Figure 2. The Standalone Architectural Design

- *Data Update*: Updating pesticide information in the storage based on a new evidence/observation
- *Configurations*: These are the steps that include initially creating a local database (storage area), linking the database to the UI elements, closing the database connection and opening the database connection.
- *Content Management*: This is where the data is organized as text, table, and image.
- *Content Adaptation*: This is where the data is rendered based on the screen of the user and the presentation captures the farmer's needs.

3.2 The Distributed Offloading Architecture

The standalone architecture handles every activity of the App on the pesticide data management on the mobile. However, the size of the data and the associated several queries can become burdensome for the mobile CPU. Since the previous studies by Gember et al. [8] suggests that offloading can be employed to free the device's resources, we adapted the technique.

The entire business logic on the mobile is moved to a middleware-layer that is hosted in a cloud computing environment. The mobile stack is re-designed to contain a network component that sends and receive data from the middleware.

The mobile can connect to the middleware via Wi-Fi or 3.5/4G. We maintained local data storage (cache) that stores the amount of data that can be stored in the cache. The mobile only writes the incoming data into the cache and retrieves the data for display in the event of connectivity loss.

The middleware, which is a Software-as-a-Service (SaaS), has two network interfaces. One end connects to the mobile and the other to the cloud storage server. The middleware processes the requests/data and sends it to the mobile

The cloud storage keeps all the information required in the database. We stored all the tables, images, and text in this database.

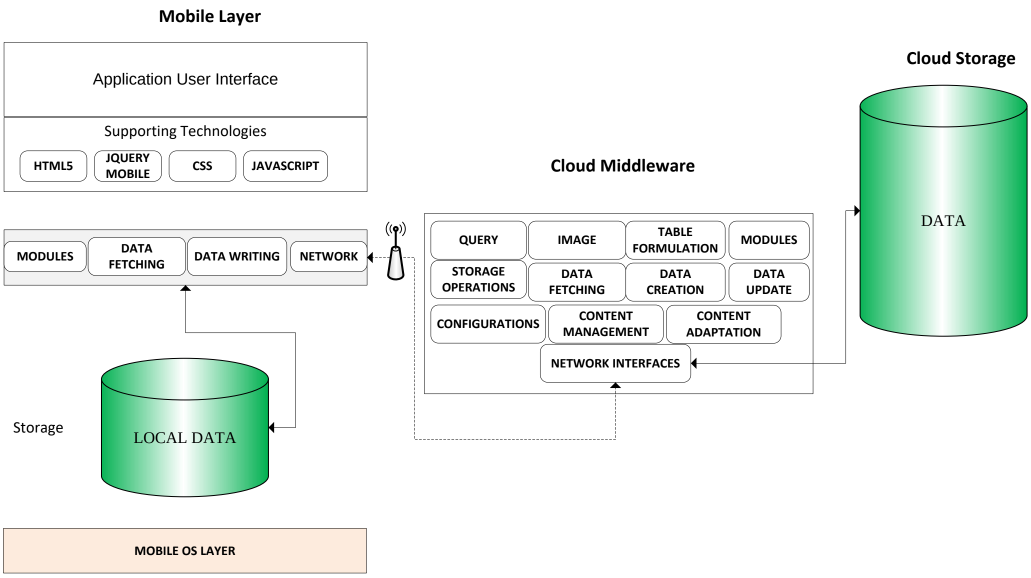

Figure 3. The Distributed Offloading Architecture

3.3 The Distributed Partitioning Architecture

The distributed partitioning architecture extends on the offloading architecture. In the former, all the processing and data management is done on the backend and the processed data is sent to the mobile. The same processed data is stored in the cache to be used later by the farmer when there is no connectivity. The problem however is that, the farmer can only update the cache with new data when there is connectivity. Moreover, since the farmer may not know which information is crucial at what time, there is no guarantee that the initial requests to the back-end can cover all the farmer's interest. This will mean several request has to be issued to the back-end which can lead to high energy consumption or request denied due to wireless connection availability.

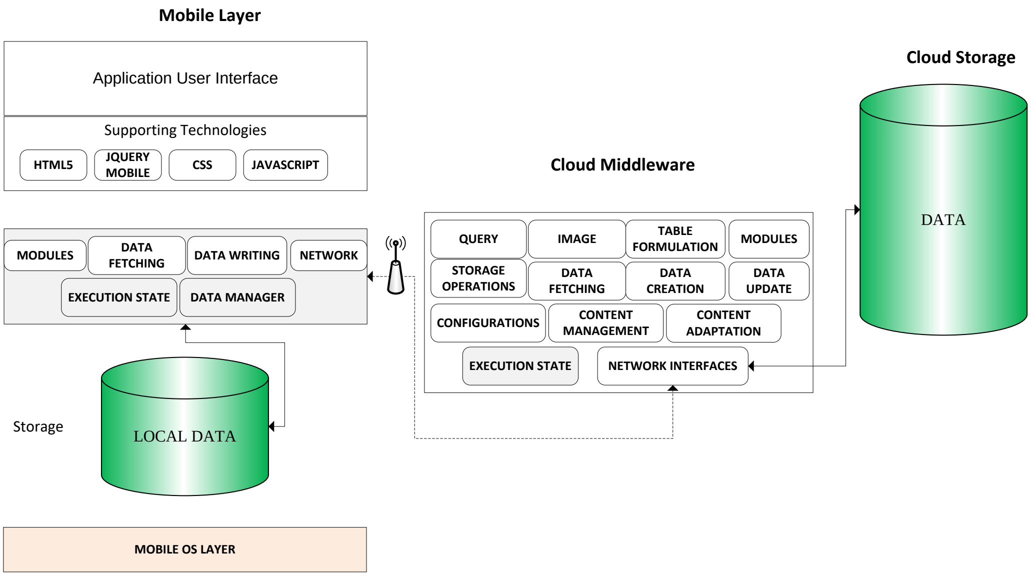

Figure 4. The Distributed Partitioning Architecture

To address the problems in the offloading scenario, we proposed the distributed partition architecture. This is to share the pesticide information between the mobile and the back-end instead of just delegating everything to the cloud-backend and calling for it later. To achieve this, the mobile node is designed to have an execution state engine, a component identical to what Ferber et al. [9] proposed. The execution state is also designed on the cloud-hosted middleware. This is to enable the middleware to determine the information that it is sharing with the mobile and the vice versa. The data manager component on the mobile also ensures the update of the local data.

In the partitioning architecture, the farmer can choose to store any of the following set of information from Table II: *Weed Control*, *Plant Disease Control*, and *Insect Control*. This gives the farmer to concentrate on what to do at a time without worrying over connectivity. In most cases, farmers will focus on any of the three issues which they can store on the mobile. This has the tendency to reduce the need for frequent request making over the network. The local copy of the pesticide information on the mobile can also be replaced by any of the other data groups.

4. EVALUATIONS

Sample user interfaces of the MobiCrop App are shown in Fig. 5. For the purpose of testing, the iOS version of the deployed application is used. We evaluate the proposed system with devices with the following specifications: *iPad 3* — OS: Apple iOS 5.1.0, Resolution: 2048x1536, Processor: A5X (dual-core, w/ quad-core graphics), Storage: 16GB, RAM: 1GB. The middleware is deployed on a privately owned cloud with the following specifications: *Processor: Intel Core i-5, CPU 2400@ 3.10 GHz 3.10 GHz, RAM: 16 GB, System 64-bit operating system.* The mobile devices connect to the middleware through 802.11g Wi-Fi 54Mbps connection. All the devices have approx. 7% of utilized resources since they have pre-installed factory applications.

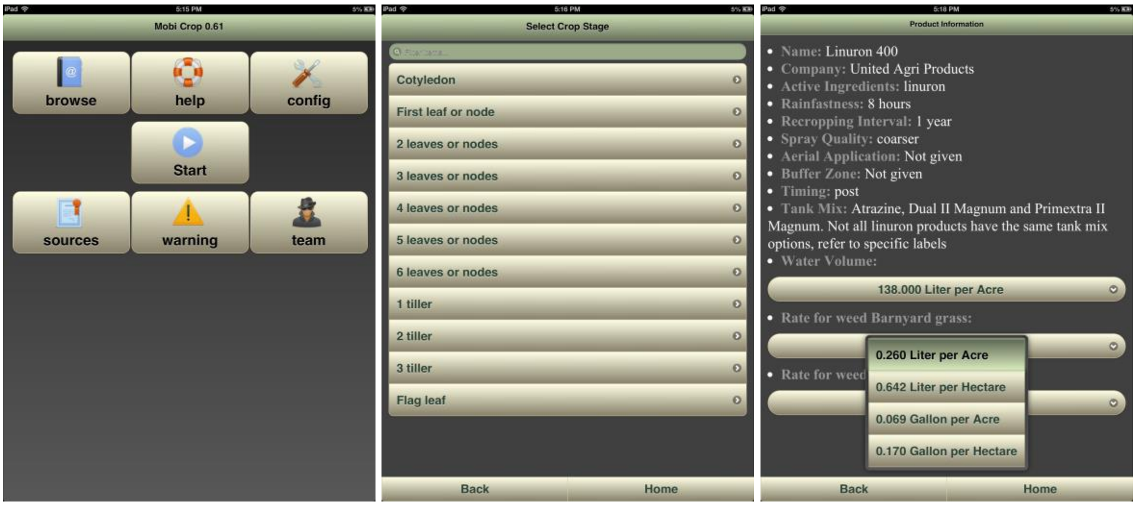

Figure 5. Some screenshots of MobiCrop

4.1. Energy Consumption

The initial experimental setup investigates the energy consumption of each of the proposed architectures. This is to enable us determine which of the approaches is beast to support the farmers in terms of battery life sustainability. We considered data sizes from 50 MB to 1000MB. The result is shown in Fig. 6.

In the case of the standalone architecture, the entire data set and the application is hosted on the mobile. We observed that the energy consumption rate is higher in this architecture because the entire application logic and data is stored on the mobile. This increases the battery consumption on the mobile.

The distributed offloading architecture consumes lower energy in comparison to the standalone architecture but higher than the partitioning approach. The delegation of the intensive business logic to the cloud-hosted middleware reduces the battery drainage. However, there is still significant energy consumption because of the continuous polling of data from the backend sources.

The partitioning architecture consumes lower energy because apart from the delegation of the business logic to the middleware, part of the pesticide data is hosted on the mobile to reduce the request over the network. As observed, as the data size and processing demand increase, the energy consumption level of the partitioning architecture is stable in comparison to the other two architectures.

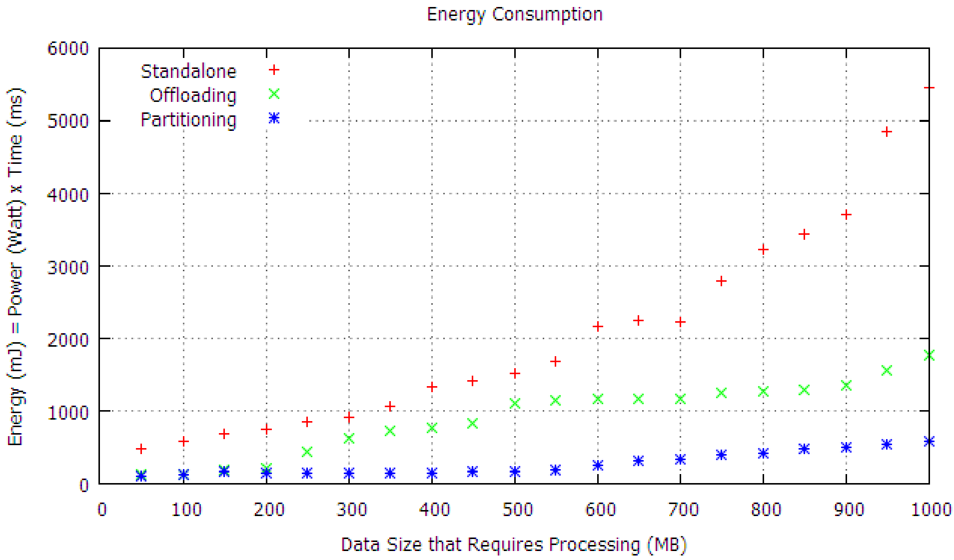

Figure 6. Energy Consumption Analysis

4.2. Bandwidth Usage

The second experiment is to evaluate the bandwidth consumption of the various architectures. The standalone architecture does not consume bandwidth since it is network independent. Thus, the evaluation focuses on the other two architectures. The result is reported in Table III. In the analysis, we considered several updates to the data from 50 to 1000. The updates represent the processing that is done on the cloud-backend and transferred to the mobile.

The offloading technique consumes more bandwidth from the observation due to the fact that the mobile has to frequently poll data from the middleware. When making the requests over HTTP, we measured the size of the message body and the header information. In both architectures, the size of the header information is approximately the same. However, the request body size differs and the number of requests required to transfer the updates also differs. For example, when there are about 200 updates (which can include changes to text, charts, etc.) and these changes occur in the same pesticide group set (e.g., weed control), it will take only one HTTP request to transmit the update to the mobile in the case of the partitioned architecture. However, the offloading architecture requires several requests to synchronize all 200 changes.

5. CONCLUSIONS AND FURTHER OUTLOOK

Mobile applications on smartphones and tablets are being used by farmers nowadays to boost productivity and market expansion. In most cases, the mobile device has to be enabled to access huge datasets and high processing of the agriculture-related information. To achieve the smooth accessibility of the data, the application has to be designed to be energy efficient and lightweight for bandwidth management. Also, to make room for eventualities such as network loss, offline accessibility provisions need to be made.

This paper discusses the MobiCrop application from the architectural design perspectives. We proposed the standalone architecture, the distributed offloading architecture – where the data and the business logic is hosted on the cloud middleware, and distributed partitioning architecture – where the data and the application logic is shared between the backend and the mobile.

Our pilot evaluations show that the distributed partitioning architecture consumes the least amount of energy and also bandwidth. The standalone architecture consumes most energy on the mobile. The work can however be extended to further improve the current state.

In summary, we found out the following:

- While computational offloading has featured prominently in most research results as an efficient methodology to conserve energy on the mobile, we realized that computational partitionining is a better approach if processing is required on both the mobile and the cloud back-end.

- Application partitioning is a better methodology for the management of the mobile wireless network since it is only higher demanding processes and tasks are executed on the cloud, and the lightweight tasks are kept on the mobile.

There can be improvement on supporting N-Screens from the farmers. This means the farmers should be able to switch between devices. Also, the future work will detail the options on fault-tolerance and error recovery of the middleware system.

Table 3. Bandwidth Usage

Number of Updates	Bandwidth Usage (kBytes/second)		
	Standalone	Offloading	Partitioning
50	0.00	5.78	4.01
100	0.00	8.71	7.50
150	0.00	16.27	15.07
200	0.00	30.57	28.94
250	0.00	43.50	42.02
300	0.00	73.98	72.06
350	0.00	132.10	82.06
400	0.00	275.05	89.06
450	0.00	448.39	94.06
500	0.00	600.65	102.06
550	0.00	612.00	102.06
600	0.00	674.00	113.06
650	0.00	702.00	142.06
700	0.00	734.00	152.06
750	0.00	802.00	161.06
800	0.00	850.00	169.00
850	0.00	890.00	174.00
900	0.00	913.00	178.00
950	0.00	944.00	181.00
1000	0.00	989.00	191.00

ACKNOWLEDGEMENTS

We are grateful to our agriculture research partners at the College of Agriculture at the University of Saskatchewan, Canada for their commitment to the success of this project and funding.

Further thanks to the entire Editorial Team of the IJ-CLOSER for their invaluable critique.

REFERENCES

[1] M. Hori, E. Kawashima, and T. Yamazaki, "Application of Cloud Computing to Agriculture and Propects in Other Fields," FUJITSU Sci. Tech. J., Vol. 46, No. 4, pp. 446-454 (October 2010)

[2] Canvas Application Store, http://www.gocanvas.com/mobile-forms-apps?tag=Pesticide

[3] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," Future Generation Computer Systems, Volume 25, Issue 6, June 2009, Pages 599-616.

[4] H. Li, J. Sedayao, J. Hahn-Steichen, E. Jimison, C. Spence, and S. Chahal, "Developing an Enterprise Cloud Computing Strategy," Intel White Paper, January 2009.

[5] H. E. Schaffer, "X as a Service, Cloud Computing, and the Need for Good Judgment," IT Professional , vol.11, no.5, pp.4-5, Sept.-Oct. 2009, doi: 10.1109/MITP.2009.112.

[6] X. Zhang, A. Kunjithapatham, S. Jeong, and S. Gibbs, "Towards an Elastic Application Model for Augmenting the Computing Capabilities of Mobile Devices with Cloud Computing," Mob. Netw. Appl. 16, 3 (June 2011), 270-284. DOI=10.1007/s11036-011-0305-7.

[7] P. Bahl, R. Y. Han, L. E. Li, and M. Satyanarayanan, “Advancing the state of mobile cloud computing,” In Proceedings of the third ACM workshop on Mobile cloud computing and services (MCS '12). ACM, New York, NY, USA, 21-28. DOI=10.1145/2307849.2307856 http://doi.acm.org/10.1145/2307849.2307856

[8] A. Gember, C. Dragga, and A. Akella, “ECOS: leveraging software-defined networks to support mobile application offloading,” In Proceedings of the eighth ACM/IEEE symposium on Architectures for networking and communications systems (ANCS '12). ACM, New York, NY, USA, 199-210. DOI=10.1145/2396556.2396598 http://doi.acm.org/10.1145/2396556.2396598

[9] M. Ferber, T. Rauber, M. H. C. Torres, and T. Holvoet, "Resource Allocation for Cloud-Assisted Mobile Applications," Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on , vol., no., pp.400-407, 24-29 June 2012, doi: 10.1109/CLOUD.2012.75

[10] Y. Ge, Y. Zhang, Q. Qiu, and Y. H. Lu, “A game theoretic resource allocation for overall energy minimization in mobile cloud computing system,” In Proceedings of the 2012 ACM/IEEE international symposium on Low power electronics and design (ISLPED '12). ACM, New York, NY, USA, 279-284. DOI=10.1145/2333660.2333724 http://doi.acm.org/10.1145/2333660.2333724

[11] H. Flores, S. N. Srirama, and C. Paniagua, “A generic middleware framework for handling process intensive hybrid cloud services from mobiles,” In Proceedings of the 9th International Conference on Advances in Mobile Computing and Multimedia (MoMM '11). ACM, New York, NY, USA, 87-94. DOI=10.1145/2095697.2095715 http://doi.acm.org/10.1145/2095697.2095715

[12] F. Malandrino, C. Casetti, and C. Chiasserini, “Pub/sub content sharing for mobile networks format,” In Proceedings of the tenth ACM international symposium on Mobile ad hoc networking and computing (MobiHoc '09). ACM, New York, NY, USA, 351-352. DOI=10.1145/1530748.1530809 http://doi.acm.org/10.1145/1530748.1530809

[13] J. Lv, and X. Y. Zheng, “Research for a Data Synchronization Model Based on Middleware and Rule Base,” Information Science and Engineering (ICISE), 2009 1st International Conference on , vol., no., pp.2998-3001, 26-28 Dec. 2009, doi: 10.1109/ICISE.2009.904

[14] G. Yang, “Data synchronization for integration systems based on trigger,” Signal Processing Systems (ICSPS), 2010 2nd International Conference on , vol.3, no., pp.V3-310-V3-312, 5-7 July 2010, doi: 10.1109/ICSPS.2010.5555804

[15] Y. Xue, “The Research on Data Synchronization of Distributed Real-Time Mobile Network,” Computer Science and Software Engineering, 2008 International Conference on , vol.3, no., pp.1104-1107, 12-14 Dec. 2008.

[16] Li Tan; Haley, R.; Wortman, R.; Ampatzidis, Y.; Whiting, M., "An integrated cloud-based platform for labor monitoring and data analysis in precision agriculture, 2013 IEEE 14th International Conference on "Information Reuse and Integration (IRI), vol., no., pp.349-356, 14-16 Aug. 2013

[17] Prasad S., Peddoju, S. K., and Ghosh D. 2013. AgroMobile: A Cloud-Based Framework for Agriculturists on Mobile Platform. Web Engineering. International Journal of Advanced Science and Technology, Vol.59, (2013), pp.41-52, http://dx.doi.org/10.14257/ijast.2013.59.04

[18] Yongyun Cho, Kyoungryong Cho, Changsun Shin, Jangwoo Park, Eun-Ser Lee. 2012. An Agricultural Expert Cloud for a Smart Farm. In Future Information Technology, Application, and Service Lecture Notes in Electrical Engineering, Volume 164, 2012, pp 657-662

[19] G. Papadavid ; M. Hadjimitsis ; S. Perdikou ; D. Hadjimitsis ; C. Papadavid ; N. Neophtytou ; G. Kountios ; A. Michaelides. 2013. Application of SEBAL methodology for estimating and disseminating through third generation mobile phones crop water requirements in Cyprus. Proc. SPIE 8795, First International Conference on Remote Sensing and Geoinformation of the Environment (RSCy2013), 87951Q (August 5, 2013)

[20] TaeHyung Kim, Nam-Jin Bae, Chang-Sun Shin, Jang Woo Park, DongGook Park, Yong-Yun Cho. (2013). An Approach for a Self-Growing Agricultural Knowledge Cloud in Smart Agriculture. In Multimedia and Ubiquitous Engineering Lecture Notes in Electrical Engineering Volume 240, 2013, pp 699-706.

[21] Georg Steinberger, Matthias Rothmund, Hermann Auernhammer. 2009. Mobile farm equipment as a data source in an agricultural service architecture, Computers and Electronics in Agriculture, Volume 65, Issue 2, March 2009, Pages 238-246.

[22] Athanasios S. Voulodimos, Charalampos Z. Patrikakis, Alexander B. Sideridis, Vasileios A. Ntafis, Eftychia M. Xylouri. 2010. A complete farm management system based on animal identification using RFID technology, Computers and Electronics in Agriculture, Volume 70, Issue 2, March 2010, Pages 380-388, ISSN 0168-1699.

[23] Ning Wang, Naiqian Zhang, Maohua Wang. 2006. Wireless sensors in agriculture and food industry—Recent development and future perspective, Computers and Electronics in Agriculture, Volume 50, Issue 1, January 2006, Pages 1-14.

[24] Koji Sugahara. 2009. Traceability System For Agricultural Productsbased on Rfid and Mobile Technology. Computer and Computing Technologies in Agriculture II, Volume 3 IFIP Advances in Information and Communication Technology Volume 295, 2009, pp 2293-2301

[25] Lomotey, R. K., Chai, Y., Jamal, S., and Deters, R. 2013. MobiCrop: Supporting Crop Farmers with a Cloud-Enabled Mobile App. Proc. of the 2013 6th IEEE International Conference on Service Oriented Computing & Applications (SOCA 2013), pp:8 pages, Kauai, Hawaii, USA, December 16-18, 2013.

[26] Lomotey, R. K., and Deters, R. 2014. Architectural Designs from Mobile Cloud Computing to Ubiquitous Cloud Computing. Proc. of the IEEE 10th 2014 World Congress on Services (SERVICES 2014), pp:8 pages, Ankorage, Alaska, USA, June 2014.

BIOGRAPHY OF AUTHORS

Richard K. Lomotey is currently pursuing his Ph.D. in Computer Science at the University of Saskatchewan, Canada, under the supervision of Prof. Ralph Deters. He has been actively researching topics relating to mobile cloud computing and steps towards faster mobile query and search in today's vast data economy (big data).

Prof. Ralph Deters obtained his Ph.D. in Computer Science (1998) from the Federal Armed Forces University (Munich). He is currently a professor in the department of Computer Science at the University of Saskatchewan (Canada). His research focusses on mobile cloud computing and data management in services eco-systems.