

Middlesex University Research Repository

An open access repository of
Middlesex University research

<http://eprints.mdx.ac.uk>

Dawarka, Viraj and Bekaroo, Girish ORCID: <https://orcid.org/0000-0003-1753-4300> (2018)
Cloud robotic architectures: directions for future research from a comparative analysis.
Proceedings 2018 International Conference on Intelligent and Innovative Computing
Applications. In: ICONIC: MAURICON 2018 International Conference on Intelligent and
Innovative Computing Applications, 06-07 Dec 2018, Plaine Magnien, Mauritius. e-ISBN
9781538664773, pbk-ISBN 9781538664780. (doi:10.1109/iconic.2018.8601264)

Final accepted version (with author's formatting)

This version is available at: <http://eprints.mdx.ac.uk/29768/>

Copyright:

Middlesex University Research Repository makes the University's research available electronically.

Copyright and moral rights to this work are retained by the author and/or other copyright owners unless otherwise stated. The work is supplied on the understanding that any use for commercial gain is strictly forbidden. A copy may be downloaded for personal, non-commercial, research or study without prior permission and without charge.

Works, including theses and research projects, may not be reproduced in any format or medium, or extensive quotations taken from them, or their content changed in any way, without first obtaining permission in writing from the copyright holder(s). They may not be sold or exploited commercially in any format or medium without the prior written permission of the copyright holder(s).

Full bibliographic details must be given when referring to, or quoting from full items including the author's name, the title of the work, publication details where relevant (place, publisher, date), pagination, and for theses or dissertations the awarding institution, the degree type awarded, and the date of the award.

If you believe that any material held in the repository infringes copyright law, please contact the Repository Team at Middlesex University via the following email address:

eprints@mdx.ac.uk

The item will be removed from the repository while any claim is being investigated.

See also repository copyright: re-use policy: <http://eprints.mdx.ac.uk/policies.html#copy>

Cloud Robotic Architectures: Directions for Future Research from a Comparative Analysis

Viraj Dawarka

Faculty of IT, Design and Communication
Curtin Mauritius, Charles Telfair Campus
Telfair, Moka, Mauritius
vdawarka@curtinmauritius.ac.mu

Girish Bekaroo

School of Science and Technology
Middlesex University Mauritius
Coastal Road, Uniciti, Flic-en-Flac, Mauritius
g.bekaroo@mdx.ac.mu

Abstract— Advances in robotics and cloud computing have led to the emergence of cloud robotics where robots can benefit from remote processing, greater memory and computational power, and massive data storage. The integration of robotics and cloud computing has often been regarded as a complex aspect due to the various components involved in such systems. In order to address this issue, different studies have attempted to create cloud robotic architectures to simplify representation into different blocks or components. However, limited study has been undertaken to critically review and compare these architectures. As such, this paper investigates and performs a comparative analysis of existing cloud robotic architectures in order to identify key limitations and recommend on the future of cloud robotic architectures. As part of this study, 7 such architectures have been reviewed and compared and results showed limited evaluation of existing architectures in favour of security weaknesses.

Keywords— *Cloud-Robotic Architectures; Robotics; Cloud Computing; Comparative Analysis; RaaS;*

I. INTRODUCTION

With the expeditiously increased capacity of artificial intelligence methods, improvements in microprocessors and development in cloud platforms, robot technology have climbed up the ladder in terms of potential and automation. Fundamentally grounded on cloud and robotic technologies, the emergence of cloud robotics enables the fusion between infrastructure cloud empowered by machine-to-cloud (M2C) communications and an ad-hoc cloud formed by machine-to-machine (M2M) communications among cooperative robots [1]. In other words, cloud robotics is the evolution of conventional robotics technology such that when connected to the cloud, facilities such as robust computational power, limitless data storage and communication resources are obtained from the state of the art data center found in the cloud that can process and share data from diverse robots and agents [2]. During recent years, the conceptualization of cloud robotic architectures has provided means for automation in large scale systems. This has been facilitated by automating robots with sensory and actuation capabilities that can capture data publish to the cloud for post processing [2]. The use of appropriate architectures in addition to cloud systems have also shown to provide a myriad of benefits [3]. Firstly, access to remote libraries for robots is made easily available through big data. Also, through collective robot learning, trajectories could be easily shared between robots and cloud computing gives access to on-demand statistical analysis.

Due to these benefits, several cloud robotic architectures have been developed during the past decade. The creation of such architectures also provide a simplistic representation of the complex underlying structures of cloud robotics. Some architectures are made up of components such as Service-Oriented Architecture (SOA) which uses web services to communicate between the cloud and the robots whereas some use client and cloud side techniques and even the incorporation of Internet of things in the architecture [4]. In terms of research related to cloud robotic architectures, few published literature is available that establishes the comparison between a ranges of implemented architectures. As related work, a previous study was followed on the cloud robotics architectures, its associated challenges and applications [1] but provided limited comparative analysis between existing architectures. Another study provided survey of related work in the field of cloud robotics and automation [4] with over 75 references in the area. Nevertheless, the study reports five ways how cloud robotics and automation can improve performance. Within this study, improvement of performance was addressed in the following areas: Big data, Cloud computing, collective robot learning, open source and open access, and crowdsourcing and call centers but the major scope was not on the comparative analysis of existing cloud robotic architectures. To address this gap, this research paper investigates and performs a comparative analysis of existing cloud robotic architectures in order to identify key limitations and recommends on the future of cloud robotic architectures.

This paper is structured in the following consecutive way: The first section gives an overview and introduction of the topic, then the second section describes the methodology used to achieve the purpose of this paper, followed by a review of current cloud robotic architectures given in the third section. In the fourth section, a comparative analysis of the existing cloud robotic architectures is provided and in fifth section provides the recommendations on future cloud robotic architectures, before concluding the paper in the sixth section.

II. METHODOLOGY

The purpose of this research paper was accomplished by adapting methodologies used in previous studies related to comparative analysis of architectures [1, 2]. The process started by a comprehensive paper search involving different online databases including Google, Google Scholar and online research

databases (IEEE, ACM, Springer and Elsevier). These platforms were utilized because of their popularity and the plethora of papers published. Ultimately, Google and Google Scholar were used to target any other papers not published within the mentioned research databases. The search process involved different keywords including mainly “cloud robotics” and “cloud robotics architecture”. From an initial selection of 68 conference and journal articles to be reviewed thoroughly, 7 cloud robotics architectures were identified and critically analyzed. Once the architectures were selected, literature search for each particular architecture was thoroughly conducted through further investigation using relevant articles and related key websites. The information obtained was then analyzed thoroughly to write-up this paper and are presented in the next sections. This methodology was utilized in different studies conducting comparative analysis [1, 2].

III. REVIEW OF CLOUD ROBOTIC ARCHITECTURES

Based on the methodology defined in the previous section, different cloud-robotic architectures were identified and are further discussed as follows:

A. Cloud Robotics Architecture by Terrissa and Ayad

A previous study [5] proposed a new cloud robotics architecture where robots can be provided as a service easily, efficiently and cheaply. The purpose of this architecture is to provide a way for users of robots (e.g. household, military robots, etc.) to utilize on-demand cloud platforms as a runtime environment of their operating systems, while also permitting to customize tasks of robots without interaction with the provider. The architecture is composed of two parts, namely, the client side and the cloud side. The client side consists of modules that occur at the client including a client administrator interface and communication module that links the mentioned interface and robots. The cloud side has modules involved in the Cloud robotics provider platform including a cloud robotics administrator interface, Virtualization Layer, Virtual Robot Systems, among others. Compared to classic cloud computing architectures, this proposed architecture has an integrated Virtual Robot Layer which consists of two components, namely a Robot Management System and a Virtual Robot System. The former is designed to manage and control robots from web whilst the latter is the operating system of the robot that is executed virtually onto the cloud environment. Within the architecture, three types of actors are present, namely the Client Administrator, Cloud Robotics Administrator and Cloud Administrator. The client Administrator takes the responsibility of the management and configuration of the local robots via a web-based interface. On the other hand, the Cloud Robotics Administrator operates mainly on the Virtual Robotics layer and finally, the cloud administrator manages the cloud infrastructure.

B. Cloud Robotics by Wan et al

In the study by Wan et al [3], cloud robotics architecture is viewed as an arrangement of two parts, namely the cloud infrastructure and the bottom facility. Whilst the cloud platform consists of key equipment including servers and database as shown in Fig. 1, the bottom facility includes robotic equipment in the form of mobile robots and unmanned aerial vehicles, among others. In terms of key features, the cloud infrastructure provides

various benefits such as dynamic computing tasks and elasticity of resources. Furthermore, most of the processing is conducted in the cloud, whilst being facilitated by networking devices. Moreover, computational load could be shifted to the cloud for processing thus resulting into smaller robot loads which also benefit from longer battery life.

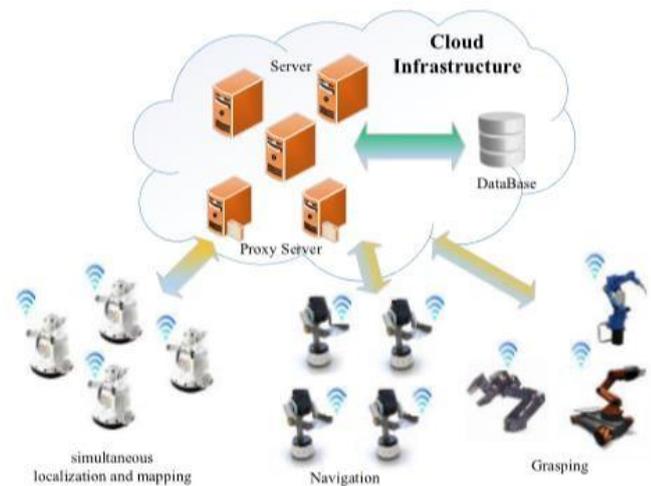


Fig. 1 - Cloud Robotic Architecture by Wan et al [11]

C. Internet of Things (IoT) Infrastructure-as-a Service (IaaS) Architecture by Mouradian et al

In order for robots to virtualize robots and to allow them to upstream applications as a service, Mouradian et al proposed an Internet of Things (IoT) Infrastructure-as-a Service (IaaS) [4]. This architecture was conceptualized so as to extend another existing architecture [5] while providing benefits such as flexibility, elasticity and cost efficiency. The architecture consists of four layers as shown in Fig. 2, namely, Network-Level Virtualization Layer, Node-level Virtualization Layer, Physical Robots Layer and Gateway Layer. The top-most layer, namely the Network-Level Virtualization Layer interfaces with the IaaS platform and consists of different modules such as a Publication Engine and Robot Monitor, among others. The Node-level Virtualization Layer is made up of different virtualized robots and the Physical Resources Layer consists of the supported robots. Finally, the Gateway Layer aims to conceal the heterogeneity and specificities of the robots such as involved communication protocols, APIs, among others. The proposed architecture was also implemented in the earthquake search and rescue domain to implement a fire suppression feature by robots. For evaluation, the prototype was compared with a peer-to-peer overlay network and results showed that the proposed architecture meets a set of key requirements.

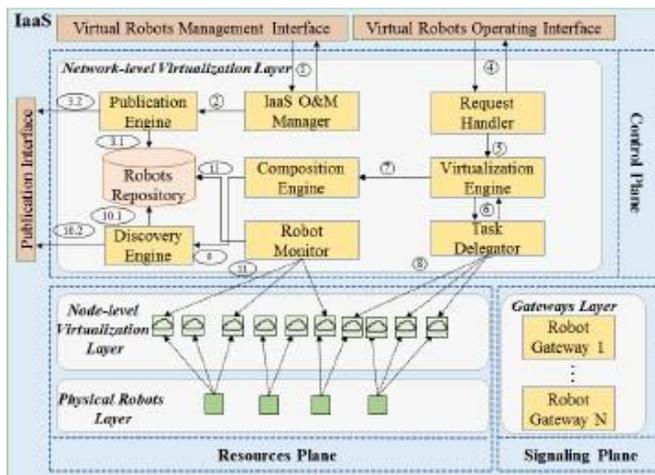


Fig. 2 - IoT IaaS Architecture by Mouradian et al [4]

D. The RoboEarth Systems Architecture

Announced in 2009, the RoboEarth project envisioned “a World Wide Web for robots: a giant network and database repository where robots can share information and learn from each other about their behavior and environment” [3]. The architecture consists of three components namely the Server, Generic Components and Robotic Specific. The Server deals with database-related objects including images, the environment (maps and actions) and also web services for reasoning. The Generic components is built-up of four parts. Firstly, action and situation recognition and labelling that facilitate the generation of action recipes. Secondly, Action Execution ensures that action is executed on the robots through proper coordination of the RoboEarth database. Thirdly, Environment Modelling combines existing information from RoboEarth database and robots sensor and fourthly, the Semantic Mapping which uses Simultaneous localization and mapping (SLAM) to mix observations of environment with identified objects from the database. Finally, a learning module is present to allow knowledge to be obtained in the form of feedback based on the performance of robot’s work. The third component, Robot Specific, consists of hardware abstraction layer which allows interaction between the computer and the robot through means of drivers and motion primitives.

E. Cloud-Based Robot Grasping System Architecture by Bekris et al.

Bekris et al. [4] presented a system architecture for Cloud-Based object recognition and grasping, consisting of two phases, namely, the offline and the online. The offline phase consists of three sections, specifically Cloud, Humans and Robots/Humans. In the cloud component, there are the Google Object Recognition Engine, the Google Cloud Storage which sends Computer Aided Design (CAD) models for analysis to the Grasp Analysis section which in term sends the Candidate Grasps to the Google Cloud Storage. The Human component has two sections, that is, Label and Domain Knowledge. The Label interacts with the Google Object Recognition Engine to train the images with the object labels whereas the Domain Knowledge sends all the semantic data such as CAD Model, Center of Mass, weight, texture and material to the Google Cloud Storage. The

last components, Robots/Humans deal with the camera where image training occurs with the Label. In the offline system, the images of each object are stored so as it can be trained with the object recognition server. The object is then utilized for the creation of a grasp which is further analyzed to know the robustness to spatial uncertainty. The second phase is the online phase which consists of two components namely, the Cloud and Robots. The cloud section contains the Google Object Recognition Engine connected to the Google Cloud Storage through the Object Label. On other hand, the Robot component has the camera which sends images to the Google Object Recognition Engine, in addition to the 3D Sensor which points cloud to the Pose Estimation. The final module is the Select Feasible Grasp with Highest Success Probability which sends the grasp execution results to the cloud storage and in terms received the candidate grasps. The execution of the online phase allows the robots to analyse the results and store them onto the cloud server for further references.

F. Integrated Service-Oriented Architecture with Robot as a Service by Chen et al

In their architecture, Chen et al [10] defined the concept of Robot as a Service (RaaS), grounded on the Service Oriented Architecture (SOA). The architecture consists of three major blocks, particularly the RaaS unit, RaaS in the cloud environment and the interfacing devices to SOA. The key blocks of the architecture include the RaaS unit and the RaaS cloud. The RaaS unit is a service broker, where clients can search for services and applications accessible within the directory of the unit. The RaaS cloud consists of different applications deployed to the units. To implement the concept, a prototype was implemented, which functions using the basis that each unit or robot hosts a repository of preloaded services. Units can benefit from the RaaS cloud which contains applications deployed by developers or clients. To enable communication between the RaaS units and services in the cloud, different interfaces were implemented. Experiments were conducted to evaluate the prototype and architecture and results highlighted effective software and hardware system to support the complex underlying infrastructure.

G. IoT-based Cloud Robotic Architecture by Karnouskos et al

In a recent work [6], Karnouskos et al demonstrated integrated an Internet of Things (IoT) layer within a proposed cloud robotic architecture. The architecture was motivating such that IoT devices could be endowed with open source software in order to perform automated tasks by robots. The architecture consists of three major components as shown in Fig. 3, namely, the Cloud layer, Robot Operating System (ROS) nodes, and the Things layer. The cloud layer contains key modules including Data Lake, IoT service and Web Dashboard through which an end-user can interact. ROS nodes contains a series of independent processes namely Workflow Engine, ROScore, Thing Integrator and Thing Controller. The final layer relates to the “things” of IoT and could be different Internet-enabled objects. In order to implement the architecture, a prototype was also created so as to show autonomous robot interactions and behavior while also enabling enterprise integration.

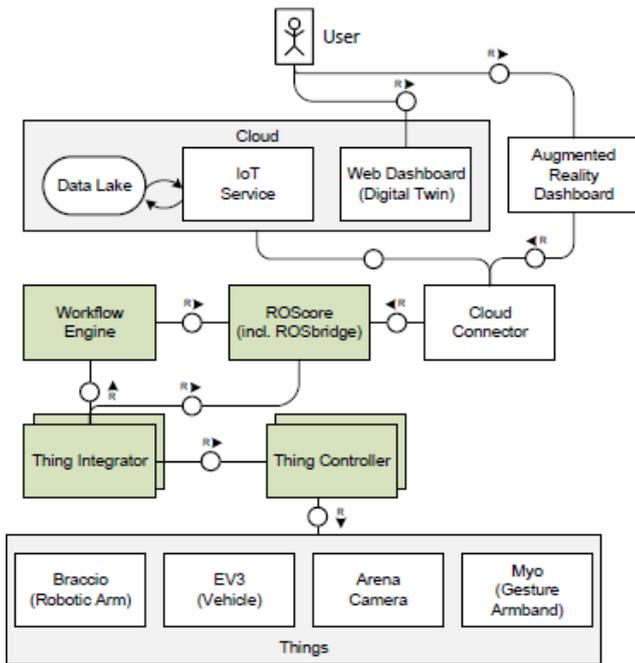


Fig. 3 - IoT-based Cloud Robotic Architecture [6]

IV. COMPARATIVE ANALYSIS

The review of cloud-robotic architectures shows that most of them consist of four components, namely, a cloud administrator, a client, cloud and a network. The key feature among the architectures studied relate to the use of cloud platform to store data and makes access of information between the robots easier, which aligns with the basis of cloud robotics. All the architectures were designed with a different purpose to address some focused problem as listed in **Error! Reference source not found.** For instance the architecture by Mouradian et al focused on IoT and IaaS, whilst the one proposed by Karnouskos et al focused on the integration of OSS technologies with IoT within the architecture. The IoT-based Cloud Robotic Architecture by Karnouskos et al was also found to be significantly different as compared to others as it provides virtualization of robots which

other architectures do not use as a service. When the same architecture is compared against the Cloud-Based Robot Grasping System Architecture by Bekris et al, the latter uses real robots for obtaining data such as images for grasp analysis. Having the new trends of facilities incorporated within the architecture leads to a major disadvantage which can discourage people of using as there is dependence between nodes where if one node fails, the other process cannot run smoothly.

The Cloud-Based Robot Grasping System Architecture by Bekris et al makes a difference when compared against existing architectures through the use of grasping which others do not use. Despite having common attributes such as Cloud for storage purposes and the use of robots to gather data, this architecture enables object recognition through the use of Google Object Recognition Engine. The grasp analysis which is performed by this architecture allows to determine the robustness to spatial uncertainty and pose estimation to select a grasp for reference. On the other hand, the architecture by Terrissa and Ayad and the one by Chen et al have the Service-Oriented Architecture as basis of the proposed architecture. As claimed in these studies, the use of such underlying architecture improves communication with the cloud through the use of web services. The only difference between the two architectures is that the one by Terrissa and Ayad has not been tested onto a real cloud infrastructure. On the other hand, the architecture proposed by Chen et al has an increase overhead due to service interaction with another service as a SOAP and REST protocol is being used. It has also been observed that SOA is not desirable when developing real-time services as asynchronous communication occurs between the services. Contrarily, the RoboEarth Systems Architecture was found to have a massive advantage compared to the other architectures as it is equipped with an immense database where large amount of data are stored and robots can retrieve data for analysis. Paradoxically, having this main advantage, it lacks privacy and security concerns when it concerns cloud connectivity. Thus, it can discourage this architecture to be used if the database is handling confidential data. A comparative summary of the architectures analysed are given in TABLE I.

TABLE I. COMPARATIVE SUMMARY

| Cloud Robotic Architecture | Purpose | Type of Architecture | Key Components | Key Features | Key Limitations |
|---|---|-------------------------------|--|---|---|
| A. Cloud Robotics Architecture by Terrissa and Ayad | <ul style="list-style-type: none"> Allows robots to conduct several work through connection of clouds and not relying on the features or hardware of robots. | Service-Oriented Architecture | <ul style="list-style-type: none"> SOA(Service Oriented Architecture) Paradigm ROS (Robot Operating system) Cloud Administrator Client Cloud Network | <ul style="list-style-type: none"> The SOA paradigm enables the use of web services to communicate with the cloud. ROS is the operating system of the robots Cloud Administrator deploys new services on the cloud to a client. Client can be either the administrator who configures the robots or the robots who which benefits the services. Cloud stores all the packages and acts as the infrastructure Network allows the communication between the cloud and the client. | <ul style="list-style-type: none"> Poor Network connection can lead to delay in communication Based on SOA paradigm and ROS as a robotic middleware thus not tested onto real cloud infrastructure. |

| | | | | | |
|--|---|--|--|---|---|
| B. Cloud Robotics by Wan et al | <ul style="list-style-type: none"> Allows multi-robot cooperative works using SLAM and navigation through the use of cloud infrastructure | Network architecture | <ul style="list-style-type: none"> Cloud Infrastructure Bottom facility | <ul style="list-style-type: none"> Cloud Infrastructure: Composed of high performance of servers, large databases, proxy servers and other components. Bottom facility: Consists of mobile robots, machinery, unmanned aerial vehicles and others. | <ul style="list-style-type: none"> Large delay when using network robotics system |
| C. Internet of Things (IoT) Infrastructure-as-a Service (IaaS) Architecture by Mouradian et al | <ul style="list-style-type: none"> It proposes an IoT IaaS architecture which enables virtualization of robots and provides them as-a-service for applications It enables the use of robots for flexibility, elasticity and cost-efficiency making into consideration the advantages of cloud such as scalability and virtualization. | IaaS (Infrastructure as a Service) and Virtualization architecture | <ul style="list-style-type: none"> PaaS & SaaS Domain Robots Service Marketplace IaaS Domain Gateway Domain Physical Robots Domain | <ul style="list-style-type: none"> In the PaaS & SaaS Domain, a Google app engine is used to host and execute the application. Robots Service Marketplace domain provides the presence server. IaaS Domain provides four RESTful web services using Java Restlet framework for two robot services. In the Gateway Domain, a robot gateway is present to map the IaaS HTTP java REST with the robots API. Physical Robots Domain consists of the robots namely the LEGO Mindstorms NXT which are being used in architecture. | <ul style="list-style-type: none"> Difficulty in homogenizing the same node-level virtualization procedures for the entire IoT resource. Existing SLA (Service Level Agreement) and QoS management procedures in the IaaS not able to handle robots services due to mobility. |
| D. The RoboEarth Systems Architecture | <ul style="list-style-type: none"> It comprises of a massive database and network where the robots can share details and can learn from one another regarding their environment and behavior. | Robotic and Embedded system | <ul style="list-style-type: none"> Server Generic Components Robot Specific | <ul style="list-style-type: none"> The server consists of several types of databases namely the Object database, Environment database, Web services and Action Databases. Generic Components have the object recognition, action and situation recognition and labelling, Action execution, master control, Data encoding, environment modelling and Learning. Robot Specific consists of the hardware abstraction layer. | <ul style="list-style-type: none"> Privacy and security concerns regarding the cloud connectivity. Potential of robots being attacked remotely. |
| E. Cloud-Based Robot Grasping System Architecture by Bekris et al. | <ul style="list-style-type: none"> It enables the estimation of robustness to spacial uncertainty through use of robots and cloud using grasping. | Online and Offline system of Cloud & Robot grasping architecture | <ul style="list-style-type: none"> Cloud (Google Object Recognition Engine, Google Cloud Storage, Grasp Analysis) Robots (Camera, Label, Domain Knowledge, 3D Sensor, Pose Estimation) | <ul style="list-style-type: none"> The system architecture consists of Offline phase and online phase. The offline phase functions with the recording of digital photos onto an object recognition server. A 3D CAD model of every object is issued and a candidate grasp is generated where each grasp is analysed to estimate the robustness to special uncertainty. The online phase works as when a photo is taken from the robot and sent to the object recognition server through network. The server then issue an object for the stored data. A 3D point set is sued to measure the robot for pose estimation and a grasp is selected from a pool of grasps available. After analysis, the robot stores the result onto cloud for reference. | <ul style="list-style-type: none"> 3D Point sets when doing pose estimation. A better analysis of image recognition is needed as it is difficult when dealing with false positive and false negative. Low confidence values with images. Grasp analysis issues when dealing CAD Models. |
| F. Integrated Service-Oriented Architecture with Robot as a Service by Chen et al | <ul style="list-style-type: none"> It proposes a robot to be an all-in-one SOA unit and can communicate with the cloud. | Service-Oriented Architecture | <ul style="list-style-type: none"> A service Provider, A Service Broker, A Service Client Generic hardware based Intel architecture, USB and Serial Port | <ul style="list-style-type: none"> The RaaS unit consists of services and applications Directory. The software in the RaaS communicate with drivers, hardware and Operating system. | <ul style="list-style-type: none"> Increase Overhead due to service interaction with another service. SOA is not desirable for Real-time as services communicate asynchronously. |

| | | | | | |
|---|--|----------------------------|---|--|--|
| | | | <ul style="list-style-type: none"> • Graphic composition based on Robotics Developer Studio and VPL Language | <ul style="list-style-type: none"> • The RaaS units can communicate with each other through Wi-Fi or Bluetooth. • The RaaS unit communicates with other services in the cloud through standard service interface WSDL. • Communication protocol used is SOAP or REST Protocol. | |
| G. IoT-based Cloud Robotic Architecture by Karnouskos et al | <ul style="list-style-type: none"> • It brings forward the aspects of autonomous behavior, decentralization, object detection, decision-making, track & trace integration with enterprise system. • It allows the developers familiarize with new OSS technologies pertaining IoT integration. | Cloud and IoT Architecture | <ul style="list-style-type: none"> • Cloud • Augmented Reality Dashboard • ROSnodes • Cloud connector • Things | <ul style="list-style-type: none"> • Cloud consists of a data lake linked with the IoT service and a Web Dashboard. • The Augmented Reality Dashboard visualizes data via an iPad pointed to devices or designated locations. • ROSnodes consists of a collection of independent process such as Workflow Engine, ROScore, Thing Integrator and Thing Controller. • Cloud connector use the ROSbridge to get access to data available via ROScore. • Things consist of a Braccio (Robotic Arm), EV3(Vehicle), Arena camera and Myo(Gesture Armband) | <ul style="list-style-type: none"> • The architecture relies on each components to function, if one node fails, the flow will not work. |

V. THE FUTURE OF CLOUD ROBOTIC ARCHITECTURES

Through the research conducted with multiple types of cloud robotic architectures implemented so far, it was observed that more work needs to be done to critically evaluate each architecture, while also comparing and benchmarking between them. The different papers reviewed in this study revealed that most architectures have not been experimentally validated against many factors that affect the real world. This could be due to limited availability of evaluation frameworks meant for cloud robotic architectures. Also, it was found that architectures based on SOA paradigm and ROS as a robotic middleware have not been tested onto real cloud infrastructure thus not making the architecture viable and efficient. A new architecture also needs to be implemented where each component does not need to rely on each other making the architecture independent to function and having a good flow with the communication from the robots to cloud.

Some studies also reported a large delay when using network robotic system and this was found to slow-down communication between the cloud and robots. This problem could be addressed by reducing the length of messages interchanged between the robots and the cloud. Additionally, limited architectures have integrated IoT components and more focus could be put in the IoT IaaS architecture in regards to mobility. This is because using this architecture causes existing Service Level Agreement difficult to handle thus resulting to loss within the system using it.

In this era where security and privacy about data is regarded as a key aspect, the architecture such as the RoboEarth needs more security concerning cloud connectivity so as the data being communicated is secure and not leaked. Furthermore, the operation over the Internet and networks also imply that such systems inherit the vulnerabilities of operating in these

environments. As such, more security testing of these architectures is needed. In addition, the computer security research community need to propose cloud robotic security architectures and standards to improve security of robots operating in these modes against different types of attacks.

VI. CONCLUSIONS

This paper analyzed and compared 7 cloud robotic architectures proposed by researchers and roboticists. From the review, it was found that the Service Oriented Architecture has been commonly used as basis of some cloud robotic architectures due to improved communication as well as the ease of integration with web services. Moreover, recent architecture has also been designed to accommodate the Internet of Thing aspect where the communication flows from cloud to robots through use of nodes in between. However, different gaps were identified during the review where the major one was that limited evaluation were conducted for these architectures, while also testing large scale applications. In addition, more work is needed to ensure low delay network traffic communication from cloud to robot and even an efficient real-time processing architecture. Overall, although some cloud robotics architectures have been proposed, various avenues for future work remain available to address the identified and future insights given in this paper. As future work, implementation of the architectures could be considered for practical analysis, comparison and benchmarking.

REFERENCES

- [1] G. Hu, W. Tay and Y. Wen, "Cloud robotics: architecture, challenges and applications," IEEE network, vol. 26, no. 3, pp. 21-28, 2012.
- [2] B. Krishna, J. Oviya, S. Gowri and M. Varshini, "Cloud robotics in industry using Raspberry Pi," in IEEE Second International Conference on Science Technology Engineering and Management (ICONSTEM), Chennai, India, March 2016, pp. 543-547

- [3] S. Miratabzadeh, N. Gallardo, N. Gamez, K. Haradi, A. Puthussery, P. Rad and M. Jamshidi, "Cloud robotics: A software architecture: For heterogeneous large-scale autonomous robots," in World Automation Congress (WAC), Rio Grande, Puerto Rico, 4 October 2016.
- [4] B. Kehoe, "Cloud-based methods and architectures for robot grasping," UC Berkeley, Berkeley, California, 2014, Available at: ProQuest ID: Kehoe_berkeley_0028E_14814. Merritt ID: ark: /13030/m56d8xmn. Retrieved from <https://escholarship.org/uc/item/898085r2>
- [5] C. Mouradian, S. Yangui and R. Glitho, "Robots as-a-service in cloud computing: search and rescue in large-scale disasters case study," 15th IEEE Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, Nevada, USA, 12-15 January 2018.
- [6] A. Tang, J. Han and P. Chen, "A comparative analysis of architecture frameworks," in 11th Asia-Pacific Software Engineering Conference, Busan, Korea , 03 December 2004, pp. 640-647
- [7] T. Magoulas, A. Hadzic, T. Saarikko and K. Pessi, "Alignment in enterprise architecture: A comparative analysis of four architectural approaches," *Electronic Journal of Information Systems Evaluation*, vol. 15, no. 1, pp. 88, 2012.
- [8] B. Kim and R. Neff, "Measurement and communication of greenhouse gas emissions from US food consumption via carbon calculators," *Ecological Economics*, vol. 69, no. 1, pp. 186-196, 2009.
- [9] A. Zakari, A. Lawan and G. Bekaroo, "Towards Improving the Security of Low-Interaction Honeypots: Insights from a Comparative Analysis," in International Conference on Emerging Trends in Electrical, Electronic and Communications Engineering, Mauritius, 25-27 November 2016, pp. 314-321
- [10] L. Terrissa and S. Ayad, "Towards a new cloud robotics approach," in 10th IEEE International Symposium on Mechatronics and its Applications (ISMA), Sharjah, United Arab Emirates, 8-10 December 2015, pp. 1-5
- [11] J. Wan, S. Tang, H. Yan, D. Li, S. Wang and A. Vasilakos, "Cloud robotics: Current status and open issues," *IEEE Access*, vol. 4, pp. 2797-2807, 2016.
- [12] B. Kehoe, S. Patil, P. Abbeel and K. Goldberg, "A survey of research on cloud robotics and automation," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 2, pp. 398-409, 2015.
- [13] K. Bekris, R. Shome, A. Kroutiris and A. Dobson, "Cloud automation: Precomputing roadmaps for flexible manipulation," *IEEE Robotics & Automation Magazine*, vol. 22, no. 2, pp. 41-50, 2015.
- [14] Y. Chen, Z. Du and M. García-Acosta, "Robot as a service in cloud computing," in Fifth IEEE International Symposium on Service Oriented System Engineering (SOSE), Jiangsu, China, 04-05 June 2010, pp. 151-158
- [15] S. Kamouskos et al., "Experiences in integrating Internet of Things and cloud services with the robot operating system," 2017 IEEE 15th International Conference on Industrial Informatics (INDIN), Emden, 2017, pp. 1084-1089.