iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** VI **Month of publication:** June 2021

DOI: https://doi.org/10.22214/ijraset.2021.35367

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VI Jun 2021- Available at www.ijraset.com

Artificial Intelligence based BOT to Manage and Administer Cloud Services

P. Z Muzzamil[1], N. Ayyanathan[2]
[1]*Master of Computer Applications,* [2]*Associate Professor, Department of Computer Applications, B.S Abdur Rahman Crescent Institute of Science & Technology Vandalur, Chennai*

Abstract: ***In the era of cloud computing, every company uses cloud technology for its applications and other infrastructure to provide a highly available and easily accessible user experience. While monitoring and managing these assets becomes a hectic work for the IT admins. On which the Level of Effort (LOE) of the resource allocated will be high and the resource must reach different console for different information. Introducing an AI-powered bot which can monitor and manage the cloud assets will reduce the manpower drastically. Most enterprises currently have very rudimentary systems of resource management where someone in the role of an Azure or resource administrator log on to the Admin Portal of their resources and have to apply filters and search through multiple screens to find even the most basic information regarding utilization and cost.***

This leads to inefficient management of resources and almost leads to overspending in resources that are being underutilized.

The implementation of the project will involve creating a cloud services management bot that can be integrated with an enterprise's collaboration suite as a way to enhance the enterprise's modern workspace. The bot is to be trained on a set of query data as part of the artificial intelligence process using the natural language processing packages that are included in the Azure Cognitive Services suite. Once queries are processed, the system will connect with the respective endpoints of the Azure Resource Management REST APIs to retrieve relevant resource utilization information and show that to the end-user.

Keywords: Cloud Computing, Azure, REST APIs, Azure Cognitive Services, Artificial Intelligence, Bot, NLP.

I. INTRODUCTION

A chatbot is an artificial intelligence software that can recreate a conversation with a client in Natural Language through informing applications, sites, portable applications or the phone. A chat bot is frequently depicted as quite possibly the most progressive and promising articulations of association among people and machines. A chatbot is a artificial intelligence software that can recreate a conversation (or a visit) with a client in Natural language through informing applications, sites, portable applications, or the phone.

A chatbot is frequently depicted as quite possibly the most progressive and promising articulations of association among people and machines.

The bot is to be trained on a set of query data as part of the artificial intelligence process using the natural language processing packages that are included in the Azure Cognitive Services suite. Once queries are processed, the system will connect with the respective endpoints of the Azure Resource Management REST APIs to retrieve relevant resource utilization information and show that to the end user.

Modern workspace-based cloud services management bot that provides responses to user queries regarding resource usage across a company's cloud and SaaS resources. It uses AI based natural language parsing of user queries to respond to users with the latest and most relevant information regarding Azure resource usages. It helps the IT admins by reducing the time consumed while switching between multiple windows.

II. METHODOLOGY

The methodology followed in this project is Top – down approach. Top down approach emphasizes planning and complete understanding of the system. This project separated into three modules. Each module is processed that generate result from the given data. The Bot is created using MVC model, Model is the data, which is used, View is the user interface which is the channel in which the bot is deployed, Controller is which handles the requests.

Figure 1 Shows the hierarchical diagram of the system. It mentions all the components that are being used in the AI bot. All the components are categorized into three modules.

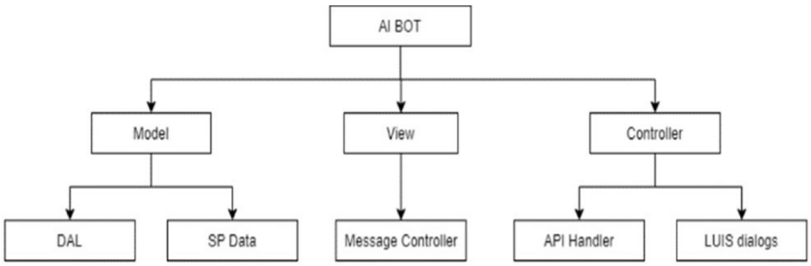


Figure 1 Hierarchical diagram of the AI BOT

III. DEVELOPMENT PROCESS

Development process and documentation are one of the activities in the software development life cycle. Development processes include requirement analysis, technologies, used to implement the project, architectural design, design and implementation.

A. Requirements Analysis

Requirement analysis determines the requirement of the system. This project analyses on product and resources requirement, which is required for this successful system. A complete understanding of the software requirement is essential for the success of the software development effort. No matter how well designed or coded, a poorly analyzed program will disappoint the user. The software scope initially established by the system engineer and redefined during software project planning, is refined in detailed. Alternative solutions are analyzed for various software elements.

Requirement analysis is classified into:

1) *Input Requirements:* The Input requirement of the BOT is the admin should have a Microsoft 365 account so that they could have teams' channel, in which the bot is incorporated. And the admin can maintain and monitor all the cloud resources associated with it.
2) *Output Requirements:* The output of the project is it is very easy for the cloud/resource admin to find the details of the cloud resources. The admin can get the details all in a single pane of glass.
3) *Resource Requirements:* Software Requirements

a) Programming Language: C#
b) Software Required: Office 365, Visual Studio.
c) API Required: ARM REST API access.
d) Subscriptions Required: Azure subscription, Azure Bot Framework subscription, Azure Cognitive Services subscription.

IV. ARCHITECTURAL DESIGN

Figure 4.1 depicts the system architecture of the entire project. It defines the structure of the developed system comprising different or modules, the externally visible properties and the relationships shows the overall architectural design of the system.

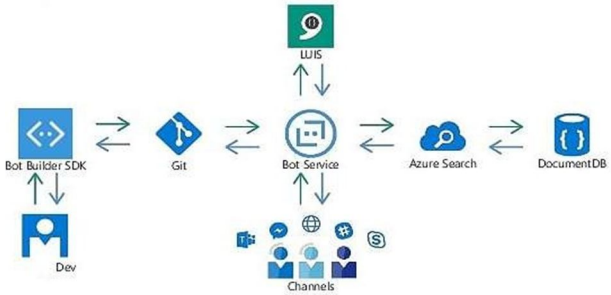


Figure 4.1 – Architecture of Cloud Management Bot

A. *Bot Developer*

Bot Developer is the one who develops the bot and provides updates and new features to the bot based on the admin's request.

B. *Bot Builder SDK*

Bot builder SDK is the development toolkit which is used to develop bot based on the requirement in the required programming language. C# in our case

C. *Version Control*

Version control is used to commit and deploy the code to the platform in azure. This is optional and can be used with any version control system like Git, Azure DevOps, Jira etc.

D. *Bot Service Framework*

Bot service framework is the platform on which our bot is deployed and made available to the production. It provides lots of flexibility in deploying and management of the bot itself.

E. *LUIS*

LUIS is the brain and the intelligent part of the bot which translates the user queries into the machine understandable scripts, commands, and API calls.

F. *Channels*

Channels are enterprise collaboration suite on which the bot is going to be used. Usually it is used to collaborate within the organization. The channel in which our bot is going to be used is Microsoft 365's teams.

V. DETAILED DESIGN

Detailed design is like architectural design but with components explained to the granularity level. Figure 5.1 explains the overall structure and components of the Bot.

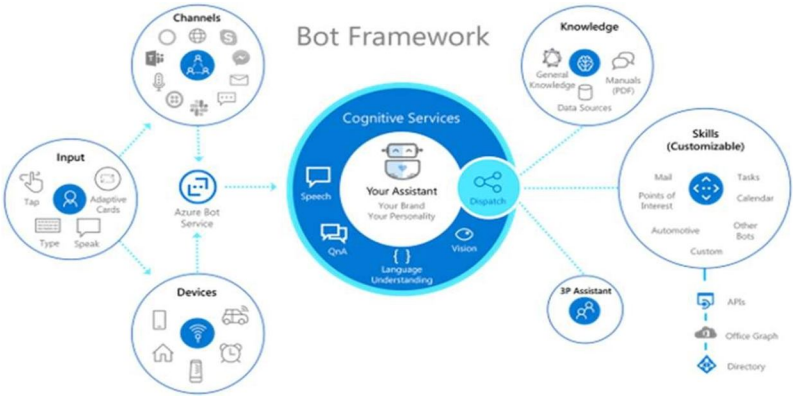


Figure 5.1 - Detailed Architectural diagram

Data flow diagram

Figure 5.2 describes the data flow of the Bot Application. Which clearly describes the how the conversations between the bot happens.

A. *User Initiation*

When the user opens the Teams channel and enters the bot's chat. Conversation request will be initiated to the bot and updates the status user joins.

B. *Bot Initiation*

After the user joins, the bot will also join the chat and ready to respond to queries.

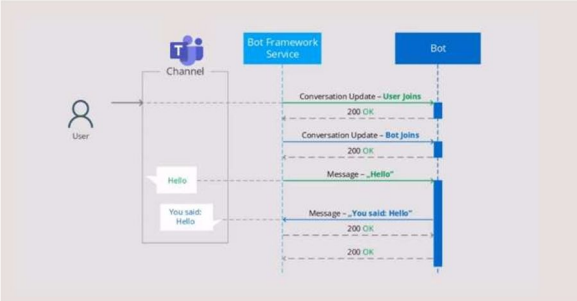


Figure 5.2 - Dataflow diagram

VI. IMPLEMENTATION

A. *Language Specification*

The programming language used in the creation of the AI based bot to manage and administer cloud services is C#.

B. *C# Programming Language*

C# is a general-purpose, modern, and object-oriented programming language pronounced as "C sharp". It had been developed by Microsoft led by Anders Hejlsberg and his team within the .Net initiative and was approved by the ecu Computer Manufacturers Association (ECMA) and International Standards Organization (ISO). C# is among the languages for Common Language Infrastructure and therefore the current version of C# is version 7.2. C# may be a lot almost like Java syntactically and is straightforward for the users who have knowledge of C, C++ or Java.

A bit about .Net Framework .Net applications are multi-platform applications and framework are often used from languages like C++, C#, Visual Basic, COBOL etc. it's designed during a manner in order that other languages can use it.

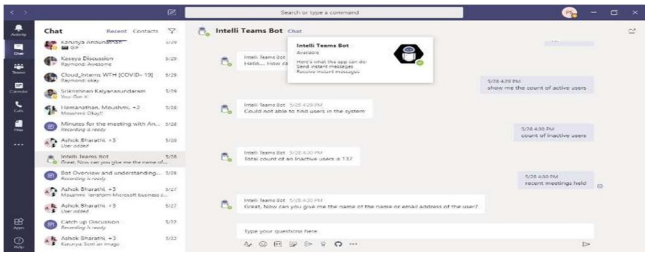
C. *Azure Bot Service*

Bot service from azure is used to Develop intelligent, enterprise-grade bots that help you enrich the customer experience while maintaining control of your data. We can build any type of bot from a simple QnA bot to our own branded virtual assistant to quickly connect your users to the answers they need. Through which the admins can easily see all the information that they need in single chat.

D. *Why use Azure Bot Service framework?*

1) *Add Intelligence at Ease:* Integrate powerful AI capabilities with Azure Cognitive Services to keep pace with technological advances without machine learning experience. So that the bot can do the hard part of translating user language to a machine understandable commands or API calls.
2) *Easily Scalable:* Since we are leveraging azure's compute resources, it is easy to scale up when needed and scale down when it isn't. So that our bot could handle a large volume of requests at once.
3) *Wide Integrations:* Integrate Bot Service across multiple communication channels to reach more customers, more often. Apply bots to channels like your website or app, Microsoft Teams, Skype, Slack, Cortana, and Facebook Messenger

VII. RESULTS AND CONCLUSION



The Figure 7.1 shows that the bot is added successfully to the chat and conversation is initiated.

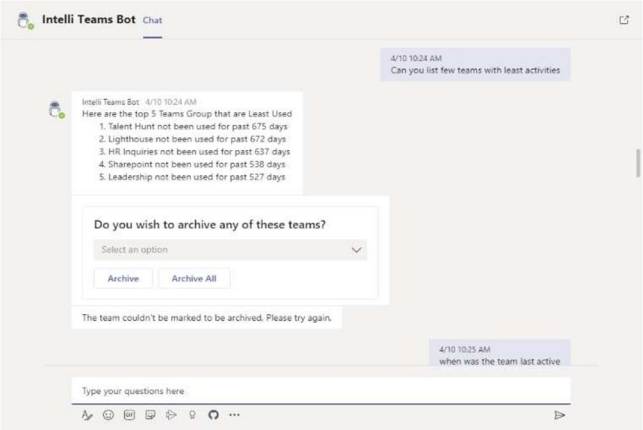


Figure 7.2 – Basic conversation with bot

The Figure 4.3 displays the adaptive cards from the bot for a specific type of queries. So, the admin/user could directly archive teams or resources.

Figure 7.3 – Adaptive cards for easy interaction

The figure 4.4 displays the command for getting insights about a team or user. The bot also gives a link in the adaptive card.

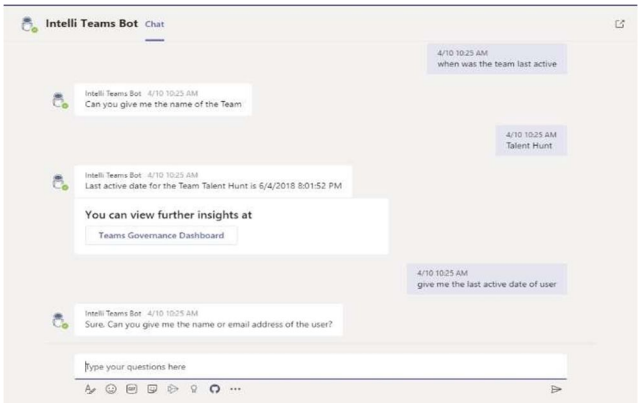


Figure 7.4 – Adaptive Card with links

The figure 7.5 displays the generated report on the user with email.

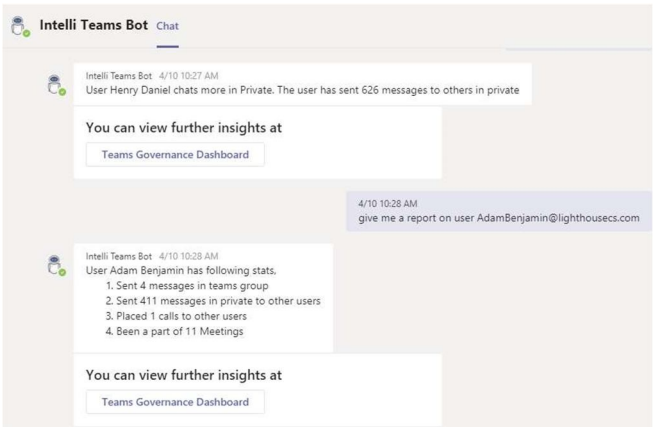


Figure 7.5 – Report generation

The figure 7.6 displays the commands used for generating quick report about the mentioned user with a link attached with an adaptive card.

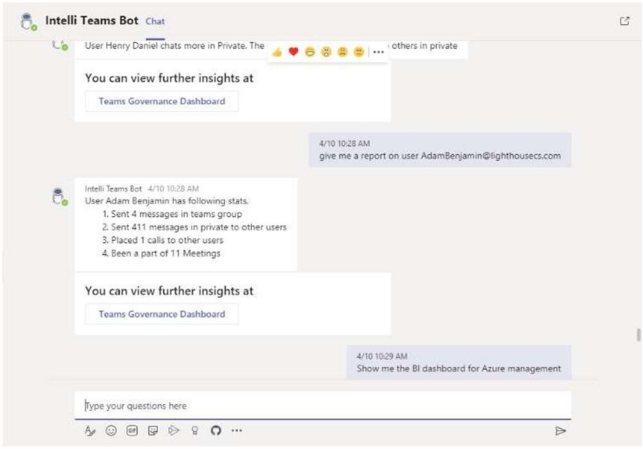


Figure 7.6 – Displays reports about user

Figure 7.7 – Sample Power BI dashboard.

VIII. RESULT

The figure 7.1 shows the successful completion of conversation of bot. In the previous chapter the requirement analysis, overall design of project implementation and testing are discussed. The chapter deals with the result analysis, conclusion, and future enhancement. This project result is shown in figure 7.9

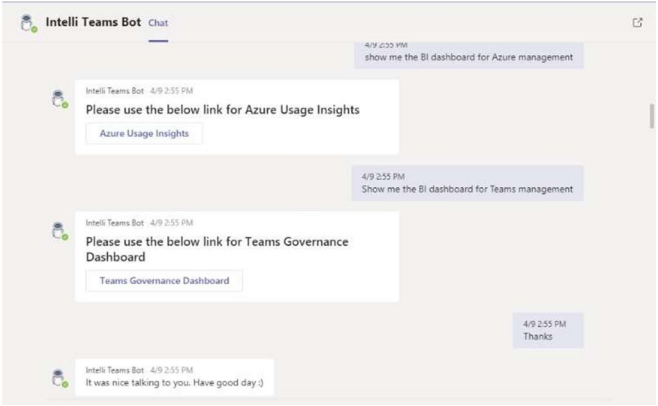


Figure 7.9 – Successful completion of Bot conversation.

IX. CONCLUSION

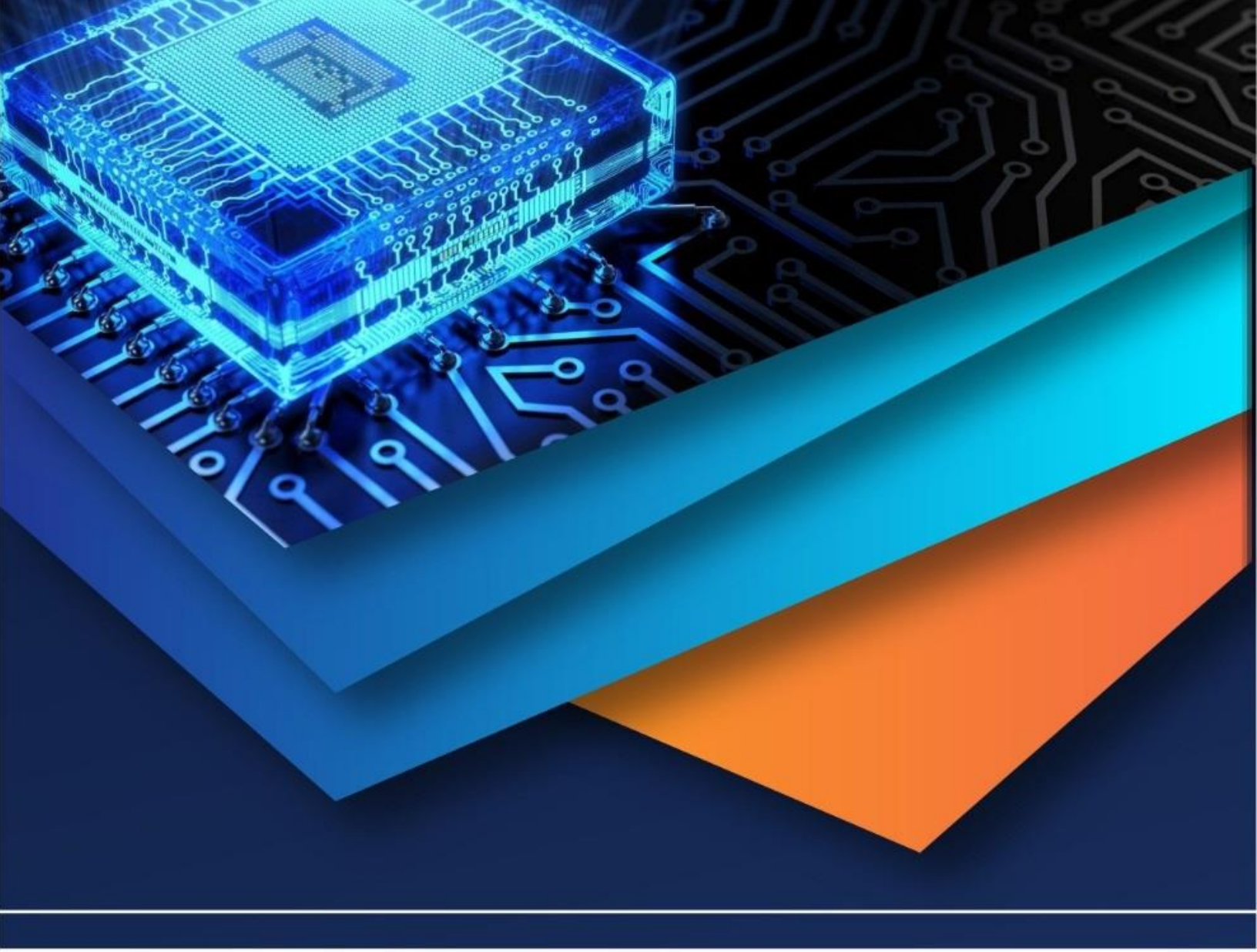
The result shows that the bot is added successfully to the Teams channel and to the chat of the user. And the user can manage, monitor, and administer the cloud services using the single bot interface.

A. Future Enhancements

There can be a lot of enhancements that can be made to this cloud bot. The bot can be made to communicate with Azure's Graph API using which we can monitor and deploy the resources in the Azure. We can also trigger alerts using the resource utilization of the cloud assets and send those notifications to the cloud admin. Since Microsoft's bot framework service is utilized, our bot can be added to the smart assistants like Google assistants or Alexa and get inputs like audio.

REFERENCES

[1] P Tadejko et al. "Cloud Cognitive Services Based on Machine Learning Methods in Architecture of Modern Knowledge" - Data-Centric Business and Applications pp 169-190, 2020 – Springer
[2] K Tajane, S Dave, P Jahagirdar "AI Based Chat-Bot Using Azure Cognitive Services" Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), 2018 - ieeexplore.ieee.org
[3] M Śmiałek, K Rybiński, R Roszczyk, K Marek "Towards a Unified Requirements Model for Distributed High- Performance Computing" - Data-Centric Business and Applications pp 1-20, 2020 – Springer
[4] R Lennartsson, J Edqvist "Chat Bots & Voice Control: Applications and limitations of combining Microsoft's Azure Bot Service and Cognitive Services' Speech API" 2019 - diva-portal.org
[5] A Sheth, HY Yip, A Iyengar "Cognitive services and intelligent chatbots: current perspectives and special issue introduction" - IEEE Internet Computing, 2019 - ieeexplore.ieee.org
[6] Jianbing Ni ; Kuan Zhang ; Yong Yu ; Xiaodong Lin ; Xuemin Sherman Shen "Providing Task Allocation and Secure Deduplication for Mobile Crowdsensing via Fog Computing" - IEEE Transactions on Dependable and Secure Computing (Volume: 17 , Issue: 3 , May-June 1 2020)
[7] Kan Yang ; Xiaohua Jia "An Efficient and Secure Dynamic Auditing Protocol for Data Storage in Cloud Computing" - IEEE Transactions on Parallel and Distributed Systems (Volume: 24 , Issue: 9 , Sept. 2013)
[8] Srikanth Machiraju ; Ritesh Modi "Azure Cognitive Services" Developing Bots with Microsoft Bots Framework pp 233-260 - 2018 – Springer
[9] Pathak N. (2017) Using Natural Language Understanding. In: Artificial Intelligence for .NET: Speech, Language, and Search. 2017 – Springer



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY