OPEN-SOURCE ROBOTIC PROCESS AUTOMATION – A CASE STUDY WITH ROBOCORP

Ramkumar R
Principal Architect
Blueconch Technologies
Ramkumar.iyer@gmail.com

Satsish R
Senior Software Engineer
Blueconch Technologies
satish5684rajput@gmail.com

Aman
Student, Software Engineer
Blueconch Technologies
Amanshri88@gmail.com

Chetan S
Software Engineer
Blueconch Technologies
Chetan12official@gmail.com

Vijaya Kumara Patnana
Software Engineer
Blueconch Technologies
vijay860@gmail.com

Rohit Gulve
Software Engineer
Blueconch Technologies
rohitgulve0101@gmail.com

ABSTRACT:

Robotic Process Automation (RPA) refers to the conversion of a manual activity to an automatic software driven process. Monotonous, Repetitive, Structured and Mundane tasks are often ripe for automation. Open-source RPA refers to RPA solutions whose code is open. This allows other products, to integrate with the open-source RPA, and extend their functionality, instead of partnering with larger closed source players. In this paper we describe the evaluation, selection, study and use of an open-source RPA framework for a major software products firm. We also discuss issues with respect to adoption and usage of open-source RPA. We also discuss the presentation made to the products firm as well as the leadership of Robocorp. As an important concern with enterprise solutions, we detail the scheduling and security mechanisms that were used to protect the Robots. Also, importantly, we highlight the creation and use of a computer vision module in the Robots which will be of interest to AI/ML scientists and users. The applications that were the target of the RPA are JIRA, Microsoft Outlook, and Microsoft Excel. Other technologies used include Windows Credential Manager, Windows Scheduler, Selenium and Python. The theoretical background for RPA was provided by the courses at Pluralsight.

Keywords:

RPA, Open Source, Robocorp, Computer Vision, Robot, JIRA, Microsoft Outlook, Microsoft Excel, Windows Credential Manager, Windows Scheduler, Selenium, Python

1. INTRODUCTION:

Robotic process automation (RPA) is a field of enterprise where software robots execute processes that were once manual, in an efficient and maintainable manner [1]. Initially, RPA was applied to legacy processes in domains like banking, finance, government and human resources and healthcare. Most RPA processes are built using closed source systems like UIPath [2], Blue Prism [3], Automation Anywhere [4] and Power Automate [5]. These include features like record and replay as well as big library of functions for common tasks like searching or opening a browser or editing a file. The big library of functions is put together in the form of an RPA robot. These robots are then started, stopped, and maintained by an orchestrator which is centrally located. The orchestrator also provides logging as well as security features. RPA also allows the possibility of combining the functionality of Legacy applications to create new functionality, instead of commissioning a new application. Modern RPA works with newer systems and has cognitive features. These cognitive features can be image recognition or machine learning of optimal paths. Such RPA is called Cognitive RPA [6] and refers to the most advanced form of RPA. The actions of the robots built using Cognitive RPA cannot be predicted in advance. Typically, RPA can run in attended and unattended mode. In attended mode, a human oversees and provides input to the RPA whenever required. In unattended RPA, there is no human. Most RPA development is preceded by process mining. As opposed to data mining which deals with finding patterns in data, process mining involves finding the process that underlies a series of interactions which contain both an identifier and a time stamp. Such mined processes are the analysed using Record and Playback to identify bottlenecks or circularity of large inventory pileup and then optimized. The optimized process is then converted into a robot. KNIME [22] and Celonis [23] are two process mining tools.

Open-source RPA, is a category of RPA in which the development of the RPA is completely open and the entire source code is available under a free license ranging from MIT [24] to GPLv3 [25]. Being open source, such products can often be integrated into existing platforms without cost. However, a thorough understanding of the dynamics of open source including

the degree of community support and the road map of the organization is needed before adoption. It is also advantageous to have a professional firm offer commercial support for adoption and usage when the product is used for mission critical uses.

2. LITERATURE REVIEW:

Since, RPA is often proprietary in nature, extensive use was made of PluralSight which is a learning provider [7][8][9][10][11]. The learning provider has courses that covered the big picture of RPA, building robots for business, building robots for a use case, introduction, and in-depth coverage of BluePrism and PowerAutomate. Further, the websites of the various Open-Source RPA providers were accessed and the level of documentation was studied. The links to all the above courses is provided in the Reference section of the document. Most RPA vendors have both RPA and domain specific courses. The rationale for using RPA versus humans along with the various aspects of RPA implementation is discussed in [38]. The difference between RPA and BPM is discussed in [39] thus providing a differentiation. An application for RPA for auditing is discussed in [40] which bolsters the ability to apply it to more use cases. The intersection of Cognitive Conversation Commerce in chatbots [42][43] also provides the impetus of Cognitive Automation to RPA is discussed in [41].

3. RESEARCH METHODOLOGY:

The research methodology had the objective of selecting and using an Open-Source RPA tool that would form the basis of product integration of the products firm. The search engine Duck Duck Go [12], along with the browser Brave [13] was used to search for "Open-Source RPA". All the options were listed. Independently a questionnaire with around 100 items was drafted. If the Open-source RPA product satisfied the question, it was given 3 points. If the Open-source RPA product could be enhanced then we gave it 1 point. If the Open Source could not be viably enhanced, it was given 0 point. The following is the ranking of the top Open-Source RPA in score decreasing order (Item 1 being Highest Scored option). This ranking is subjective to the requirements of the product firm

1. Robocorp [16]

2. Open RPA [17]

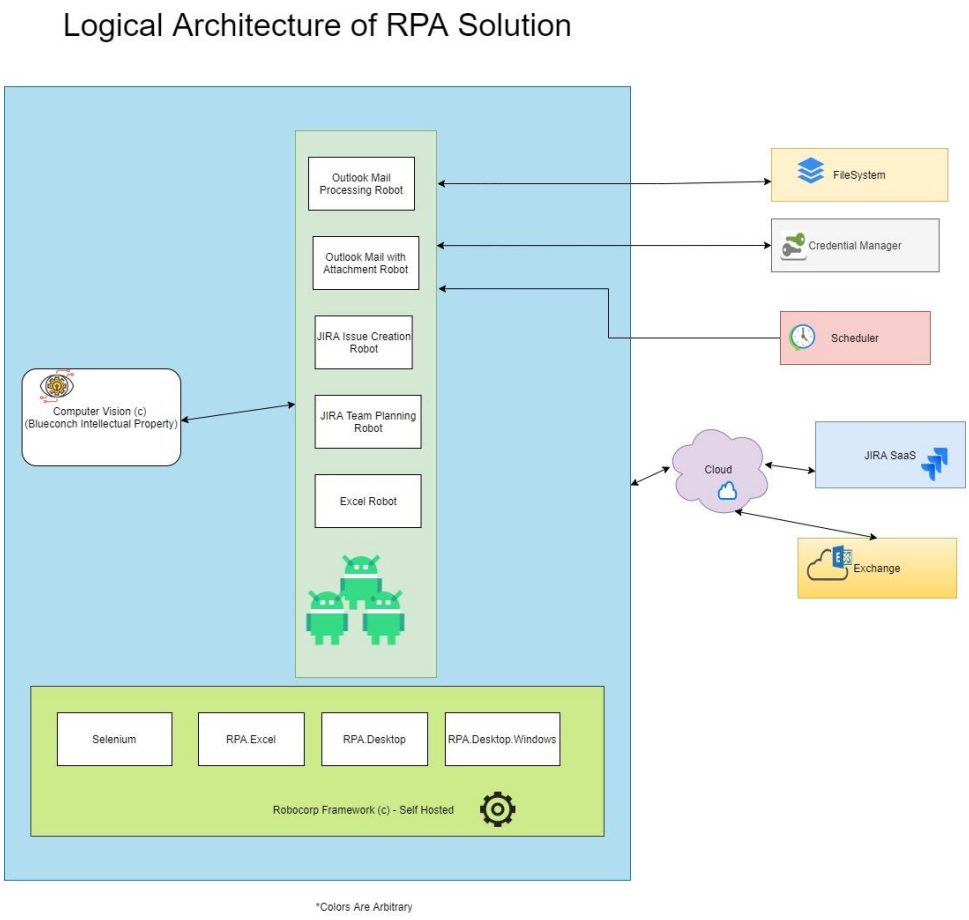
3. TagUI [18] and taskt [19]

4. RobotGo [20]

4. USE CASE AND ANALYSIS:

The use case for the selected Open-Source RPA, Robocorp (Apache License [26]) was to convert an email sent by a customer in Microsoft Outlook [15] to a customer ticket in JIRA [14] associated to a delivery team and download the Excel spreadsheet [15] and send it back to the customer. If the delivery team did not exist, then it had to be created by the customer and a developer had to be assigned to the delivery team by an invite. The RPA also must automatically login as the developer and accept the invite. A process definition document and Team Foundation Server tickets [27] were raised for the requirements of the use case. An architecture diagram for the use case is given in Figure 1.

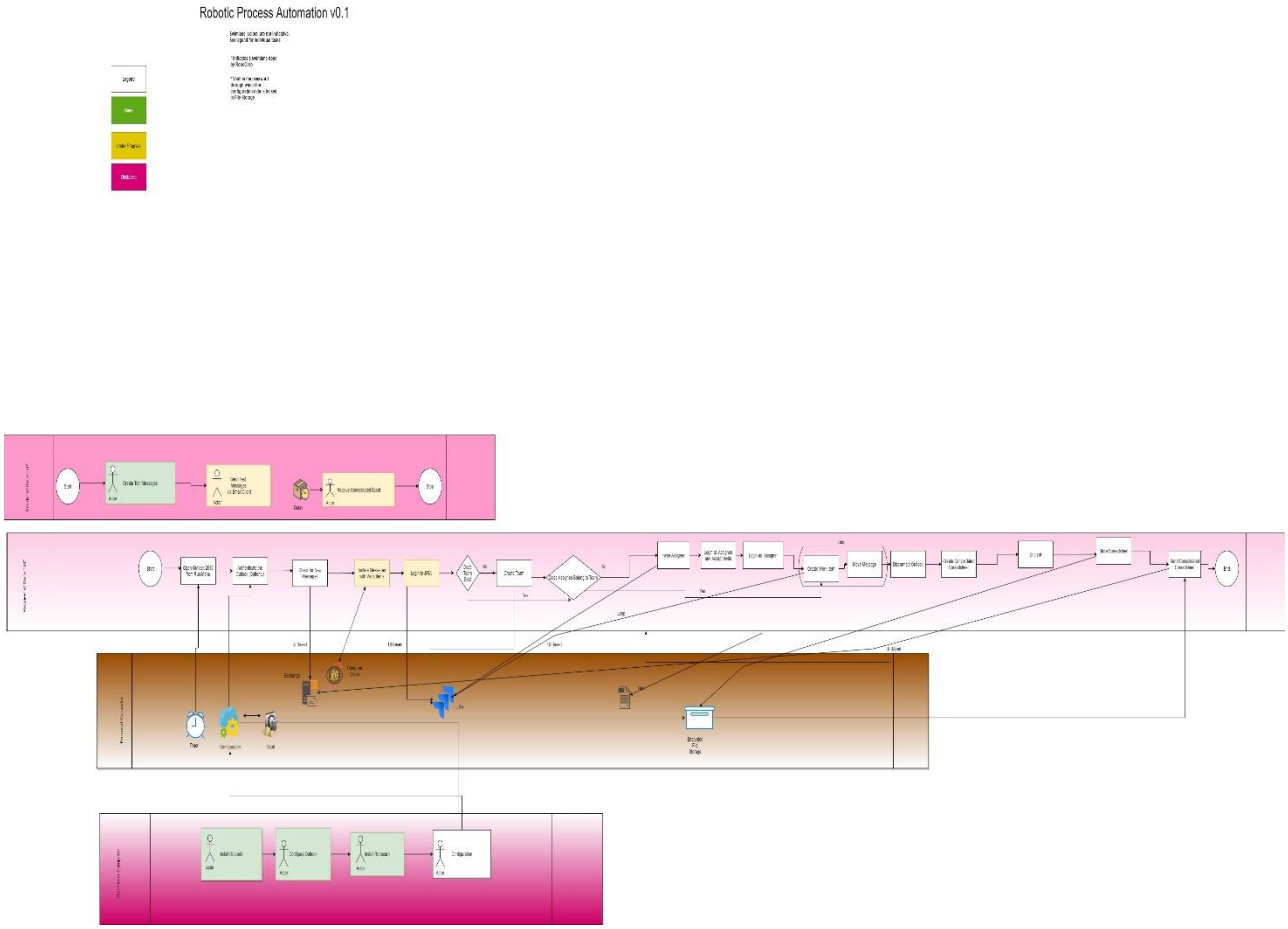
Fig 1: Use Case Architecture



The implementation of the entire RPA solution was split among four teams as per Figure 2.

1. Team A: Focus on building bots
2. Team B: Focus on Computer Vision [37]
3. Team C: Focus on Integration
4. Team D: Focus on Scheduling and Security

Fig 2: Use Case Flow



The Outlook bots were written using Robocorp DSL (Domain Specific Language) [28]. The time taken for the first bot was 24 hours. The time taken for the second bot of similar complexity was only 3 hours. Hence training took around 21 hours. The JIRA bot was written completely in Selenium [21] by a senior developer since Robocorp actively encourages the same [29] and was done by a single person. The Excel bot did not require Excel to be

installed and used an inbuilt library in the RPA framework [30]. Since the solution was to be self-hosted, python key-ring [31] along with Windows Credential Manager (WCM [32]) was used to store sensitive data like JIRA passwords. The scheduling happened through a manual run using Windows Scheduler [33] which called a Python API (Application Programming Interface) [34]. The computer vision team provided the Outlook image analysis and segmentation code that was used to extract the email and its contents to call the JIRA bot. The entire operation took around 300 hours (40 worker days with each day being 8 hours. 1 hour is reserved for meetings in 8 hours) and the time split was below

A. Outlook message receipt bot: 24 Hours
B. Computer vision: 100 Hours
C. Integrating Computer vision with Outlook: 20 hours
D. JIRA Create Issue bot: 21 hours
E. JIRA Create Delivery Team bot: 16 hours
F. JIRA Create Invite bot: 16 hours
G. JIRA Accept Invite bot: 16 hours
H. JIRA Download Spreadsheet bot: 7 hours
I. JIRA convert to Excel bot: 3 hours
J. Outlook send Excel sheet bot: 3 hours
K. Integration: 40 hours
L. Scheduling and Encryption: 16 hours
M. Demo Readiness and Demo: 4 hours

The entire solution was demonstrated to the top management of the products firm using screen-sharing. An artificial delay was introduced within the robots so that they can be perceived by the people. People were happy with the performance of Robocorp and remarked that the use case was much more complicated and powerful than they thought.

The demo was also presented to the top management of Robocorp and they were happy at the usage and provided feedback that we could have used their custom Outlook module.

5. RESULTS:

RPA is a buzzword in the industry today. Open-source RPA tool Robocorp allows RPA to be integrated at a source code level with other IT products. We learnt that mature Open-Source RPA tools exist in the market that can be used to build complex robots that are of real use to reduce manual effort. We learnt that state of the art computer vision technologies can be used in RPA. We also understood that RPA has a learning curve. We also learnt about the ideal way to do any RPA based project that involves building multiple Robots.

6. CONCLUSIONS:

After a survey of open-source tools with a scaled questionnaire, Robocorp was identified as a suitable framework for RPA. Robocorp is an excellent Open-Source RPA tool that can be used to build multiple Robots. We could also integrate with AI/ML like Computer Vision to satisfy product firms that look for cognitive processes. Even though there is a learning curve, the creation of Robots gets progressively simpler and complex Robots can be built. These include most of the important software used by Customer like JIRA, Microsoft Excel, and Microsoft Outlook. The integration of the entire robots for a use case takes more time when multiple parts are involved. For a self-hosted case, we can wrap the Operating Systems functionality like scheduling and credentials, but in future either a private or public control room functionality of Robocorp must be utilized. We were also able to contact the apex decision makers in Robocorp which further indicates the market maturity of open-source RPA like Robocorp. Robocorp has further created a dedicated Slack channel for us [44]. Further we found that Selenium can be integrated into the Robocorp robots for Web Automation whereas for desktop elements we used Robocorp wrapped tools like RPA.Excel. We were not able to use RPA.Outlook as it did not offer the functionality to move messages. We also are getting more requirements from ETL Solutions [45] for Excel Automation.

7. SCOPE OF FUTURE RESEARCH:

Robocorp offers a public cloud hosted model where the orchestrator and security components are cloud based and can manage an on-premise workforce of robots. This will be especially useful for robots that span multiple organizations or premises. Further, a discussion with the Robocorp CEO revealed that the most important question for RPA is the business outcome. He further encouraged researchers to find fitness functions [35] for RPA as well as discuss if RPA robots can collectively exhibit emergent behaviour [36] if they are independent.

8. REFERENCES:

1. https://searchcio.techtarget.com/definition/RPA
2. https://www.uipath.com/
3. https://www.blueprism.com
4. https://www.automationanywhere.com/
5. https://flow.microsoft.com/en-us/
6. https://research.aimultiple.com/cognitive-automation/
7. https://app.pluralsight.com/library/courses/getting-started-robotic-process-automation/table-of-contents
8. https://app.pluralsight.com/library/courses/rpa-workflow-automation-executive-briefing/table-of-contents
9. https://app.pluralsight.com/library/courses/getting-started-blue-prism-rpa/table-of-contents
10. https://app.pluralsight.com/library/courses/creating-basic-robots-rpa/table-of-contents
11. https://app.pluralsight.com/library/courses/power-automate-rpa-beyond/table-of-contents
12. https://duckduckgo.com/
13. https://brave.com
14. https://www.atlassian.com/software/jira
15. https://www.microsoft.com/en-in/download/office.aspx

16. https://robocorp.com/

17. https://www.openrpa.dk/

18. https://github.com/kelaberetiv/TagUI

19. http://www.taskt.net/

20. https://github.com/go-vgo/robotgo

21. https://www.selenium.dev/

22. https://www.knime.com/

23. https://www.celonis.com/

24. https://opensource.org/licenses/MIT

25. https://www.gnu.org/licenses/gpl-3.0.en.html

26. https://www.apache.org/licenses/LICENSE-2.0

27. https://azure.microsoft.com/en-in/services/devops/server/

28. https://robocorp.com/docs/courses/beginners-course/your-first-robot

29. https://robocorp.com/docs/libraries/rpa-framework/rpa-browser-selenium

30. https://robocorp.com/docs/libraries/rpa-framework/rpa-excel-files

31. https://pypi.org/project/keyring/

32. https://support.microsoft.com/en-us/windows/accessing-credential-manager-1b5c916a-6a16-889f-8581-fc16e8165ac0

33. https://docs.microsoft.com/en-us/windows/win32/taskschd/task-scheduler-start-page

34. https://www.python.org/

35. https://www.thoughtworks.com/radar/techniques/architectural-fitness-function

36. https://www.britannica.com/science/emergent-property

37. https://towardsdatascience.com/everything-you-everwanted-to-know-about-computer-vision-heres-a-look-why-it-s-so-awesome-e8a58dfb641e

38. Van der Aalst, W. M., Bichler, M., & Heinzl, A. (2018). Robotic process automation.

39. Willcocks, L. P., Lacity, M., & Craig, A. (2015). The IT function and robotic process automation.

40. Huang, F., & Vasarhelyi, M. A. (2019). Applying robotic process automation (RPA) in auditing: A framework. International Journal of Accounting Information Systems, 35, 100433..

41. Wróblewska, A., Stanisławek, T., Prus-Zajączkowski, B., & Garncarek, Ł. (2018). Robotic process automation of unstructured data with machine learning. *Annals of Computer Science and Information Systems*, *16*.

42. Ramkumar, R. VISUALIZATION, CONFIGURATION AND AUTOMATED TESTING OF A NATURAL LANGUAGE PROCESSING APPLICATION FOR CONVERSATIONAL COMMERCE–A CASE STUDY.

43. https://servisbot.com/the-intersection-of-smart-bots-and-robotic-process-automation-rpa/

44. https://slack.com/intl/en-in/

45. https://www.guru99.com/etl-extract-load-process.htm