

CIBot: Integrating and Automating Researcher Support in Cyberinfrastructure Helpdesk Operations

IMAN ATAEE LANGROUDY

Department of Physics and Astronomy, Texas A&M University, College Station, TX, iman.ataee@tamu.edu

MARINUS PENNINGS

High-Performance Research Computing, Texas A&M University, College Station, TX, pennings@tamu.edu

DHRUVA K. CHAKRAVORTY

High-Performance Research Computing, Texas A&M University, College Station, TX, chakravorty@tamu.edu

HONGGAO LIU

High-Performance Research Computing, Texas A&M University, College Station, TX, honggao@tamu.edu

This student-led paper introduces a multi-purpose user support tool built to enhance and automate the workflow and the bookkeeping of a ticketing request system. This tool can create custom requests, define custom workflows, and define permission hierarchies. Requests can be customized to have any number of fields and can inherit a portion of their fields from another request. Actions are recorded in a document-based database. This tool's flexibility allows it to accept input from any outside sources with proper integration. To improve its security, for its main functionalities, it can either use a websocket that requires no inbound IP whitelisting, or it can only accept connections with a proper RSA SHA256 signature. This tool is made with the least privilege and controls inbound connections. Its outbound connections are only to Slack and its database, which can be localized. A pilot implementation at Texas A&M University that was integrated with the Slack workspace tool, reduced the time to complete researchers' requests, and improved bookkeeping processes.

CCS CONCEPTS • High-Performance Computing • Software Infrastructure • Design

Additional Keywords and Phrases: workflow management, workflow automation, slack, helpdesk functions

ACM Reference Format:

Iman Ataee Langroudy, Marinus Pennings, Dhruva K. Chakravorty, Honggao Liu 2023. The Title of the Paper: Integrating and Automating CI User Support Functions: This is the subtitle of the paper, this document both explains and embodies the submission format for authors using Word. In Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY. ACM, New York, NY, USA, 10 pages. NOTE: This block will be automatically generated when manuscripts are processed after acceptance.

1 INTRODUCTION

A significant portion of a CI Helpdesk's responsibility is handling incoming requests for file quotas, group directories, licensed software installations, accounts, storage additions, and wall-time increases. These requests often have approval and tracking steps. For example, for a storage addition request, a request must be first approved, then the storage increase should be done by a system admin, and details of the progress should be documented, tracked, and reported to the requestor. Often a Helpdesk technician (a staff or student technician) manually moves and tracks a ticket's progress. This includes communicating with the responsible staff to ensure the request is being processed and with the requestor, e.g., when more information is needed

to process the request or update the requestor on the status. This process typically involves using a ticketing system (like RequestTracker [2]) where the user can submit the request and a designated Slack workspace for internal communications within the center's personnel. Here not only do Helpdesk members need to know what information is relevant for the request, but there are also several opportunities ripe for miscommunication between the various involved groups. Ensuring standardized bookkeeping and accounting practices can be difficult too. This communication chain can be repetitive, circuitous, and largely inefficient for complex requests requiring several layers of approvals.

This tedious process can be automated using commercial products. These, however, can be expensive or require additional commercial plugins. While some commercial software [3][4] offer powerful integrated solutions that can accept messages from various sources, they can be limited in adopting custom workflows, may be coupled to per-person licenses, and can be beyond the means of small centers with limited budgets. Ironically, these centers have few CI people, with hardly any time to spare. There is a dire need for open-source and free software that helps CI professionals automatically route, track, and record the process.

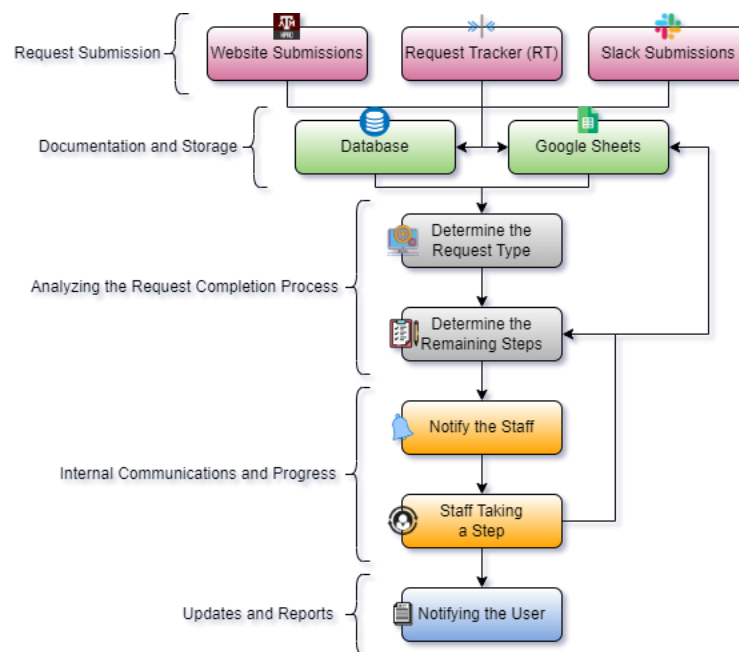


Figure 1: Visual representation of a checklist for monitoring and delegating incoming requests.

2 REQUIREMENTS FOR A WORKFLOW MANAGEMENT TOOL

We need a workflow management tool to maintain and improve the ability to assist its users. A tool that can provide workflows for various types of requests, a tool that is easy to use and get familiar with. A workflow management tool's most important factor is the ability to create and manage workflows. The tool should be able to generate workflows for all current requests while being flexible enough to adapt to future workflows. Another important requirement is automating as many steps in the workflow as possible. Another requirement would be the ability to tag/notify the correct responsible staff member automatically. The workflow tool should

provide a straightforward interface that accepts requests from any front end. In addition, the workflow tool should include a default channel where helpdesk members can submit requests manually and guide a helpdesk member when entering the relevant information.

3 INTRODUCING CIBOT

The CIBot is a student developed software that assists with CI helpdesk operations. CIBot supports researcher-submitted and help desk-generated tickets. It has a built-in feature to guide users as they generate a new request. CIBot narrows down the request type, based on the query, and generates a final request type that can be integrated into Slack, Jira or any other web-app. Communications are automated by predefining the responsible individuals (staff members) for each type of request. It is equipped with an automated reminder system, eliminating the chance of a missed request between communications. “Tagging” features allow requesters to create meta-data to make data accessible and reusable. Requests and approval chains are uploaded into a secure database. Additionally, each request also gets uploaded into a customizable cloud-based spreadsheet that tracks every modification. These sheets are not only used for bookkeeping purposes but are also used by the responsible staff members working on the request. Combined with a good bookkeeping system, the CIBot is capable of performing custom searches and generating statistics as well. Fields on requests can be tagged as “searchable”, ensuring that the status of a request and all associated actions are searchable.

To test and facilitate the CIBot’s adoption, the National Science Foundation-funded SWEETER Cyberteam [5] incorporated it at Texas A&M High-Performance Research Computing (HPRC). This pilot was affectionately named HPRCBot. This is part of the larger software development ecosystem [6][7][8] led by staff and students at HPRC. For our automation, we aimed to achieve good categorization, communication automation between staff and users, automated bookkeeping, which can take a record of each request in both databases and spreadsheets, search tools, and tools to modify existing requests.

Currently, all the connections in our automation are either with Slack, Google, or MongoDB. For Slack, we made a custom asynchronous library for Slack around python-slack-sdk [9]. Our automation connects to Slack using a web socket which requires no inbound IP whitelisting or URL exposure to listen to new events. Our automation is fully asynchronous, which means it can handle another request while facing a blocking task like an API response. For connections with Google, we used the aiogoogle [10] python library, an asynchronous wrapper around google API. Lastly, for MongoDB, we used motor [11], a coroutine-based API for non-blocking access to MongoDB. Unless a user support system wants to accept HTTP requests for any part of the bot mentioned in 3.1, the automation requires no inbound IP whitelisting or URL exposure. It would only need outbound connections to Google, MongoDB, and Slack IPs.

A detailed description of the implementation along with technical documentation has been provided elsewhere. [12]

3.1 REQUEST SUBMISSION AND CATEGORIZATION AUTOMATION

At HPRC, we have developed workflows to manage ticketing requests. These have been integrated into standard operating procedures. These workflows have been integrated into the CIBot pilot as workflow objects. For example, Figure 2 shows a graphical representation of the workflow for a “quota request” for a

researcher who has requested additional storage space on a HPRC cluster. They can either purchase storage (buy-in option) or request the disk space for free for active research purposes (regular request).

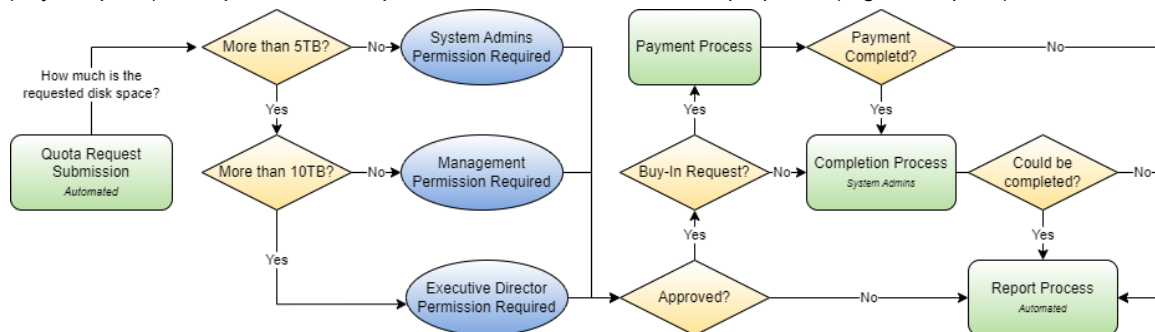


Figure 2: Simplified visualization of a workflow/ approval trail for a quota request.

Here, CIBot simplifies and automates this process. Figure 2 shows a screenshot of a CIBot generated approval chain on a slack workspace. Figure 3 (a) shows the options for a user to get storage. Depending on the selected category (buy-in or regular categories), CIBot offers follow-up questions as shown in Figure 3 (b). CIBot then provides options to search for any request in any state for any responsible staff member to let staff members and helpdesk members to quickly check for any workflow step they are responsible for.

Figure 3: Screenshot of a quota request form, Figure on the right shows the first question, and the Figure on the left shows the follow-up questions.

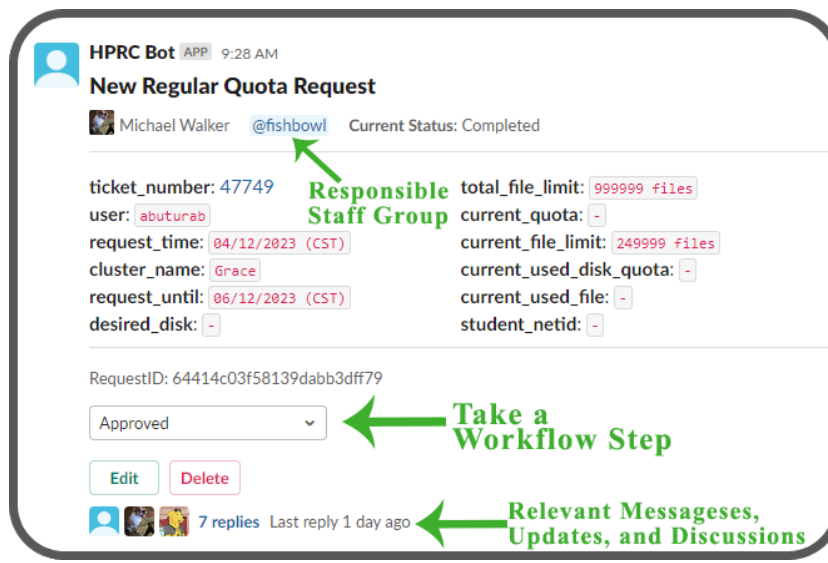


Figure 4: Screenshot of a workflow view for a quota request

CIBot shows the workflow by posting the necessary request information into a dedicated Slack channel on HPRC's slack workspace. Each request has a modifiable slack channel attribute to customize the channel where the request should be posted. Figure 4 shows a screenshot of a workflow for a quota request at Texas A&M HPRC that is aligned to specific policies. It shows the requestor, the current status of the request, information about the ticket, the current and requested quota, the responsible staff members who will be automatically tagged in every step of the workflow, and an action menu. When you click on the "Pick an option" dropdown, it shows all the applicable options for this request. Figure 5 shows all the options on the left side of the figure. In this example, the options are "Completed, Approved, Denied, and Reported". To restrict privilege, only responsible staff can perform certain actions. For example, for a quota request between 5TB and 10TB, only management can approve it. If someone else tries to approve, it will show a private message, this action is not allowed. The screenshot on the right in Figure 5 shows the message when an action is performed. Aside from showing the new status, the post also shows who performed the action and the time stamp the action was performed. Once a request is completed, the HPRCbot sends an email to the researcher, through RT with the relevant information to update the user with the latest status of the request. The staff member updating the request can add additional comments as part of the response.

3.2 EDITING/DELETING EXISTING REQUESTS

In case a requestor makes mistakes or changes have to be made, the outdated request should not be left alone in the records. Therefore, instead of posting a new request, one should be able to edit or delete an existing one. Upon receiving a deletion request, the target request will be deleted from the database, the relevant spreadsheet, and all other platforms where the target request has been posted using automation. Likewise, upon an edit request, all records of the target request will be updated.

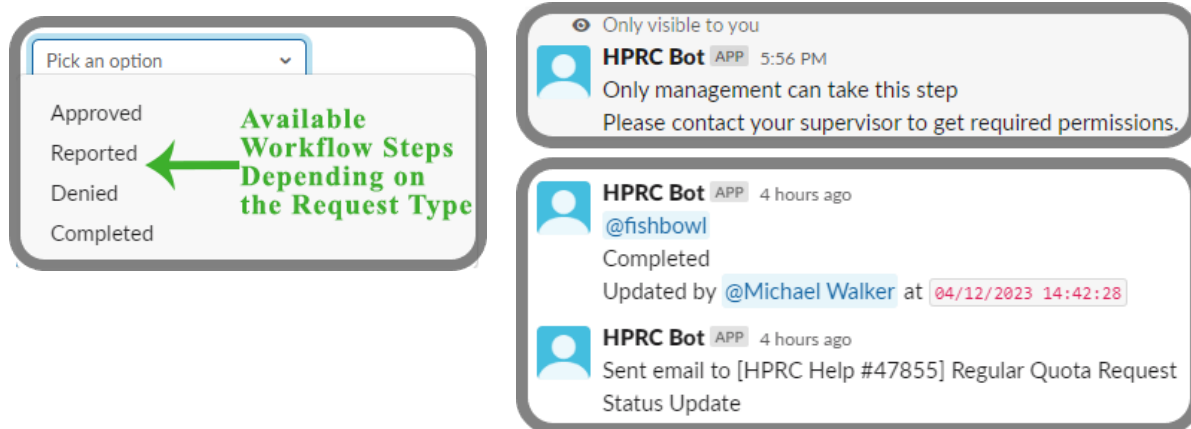


Figure 5: Screenshot of a workflow action chain.

4 IMPACT

Over a period of 30 days in Spring 2023, the HPRCBot pilot was used to facilitate 38 requests for additional storage and increasing the number of files allocated. 15 additional requests for software installation were also processed using the HPRCBot. HPRCBot has pre-assigned fields for different kinds of user-requests, making it easier for Help Desk technicians to generate requests without having to learn all the standard operating procedures that describe how to report a request to relevant staff, who to report to, and how to communicate with the requestor. For storage and file quota requests, we noticed that on receiving a ticket from a user, a Helpdesk technician could generate the approval request with ease. Based on the selected entries, HPRCBot would automatically tag the appropriate people, and continue to propagate the request to various team members till its completion. By having information assigned to specific fields, requests were easier to review. Moreover, since requests were automatically routed to the next person, we estimate that the workflow reduced the time for completion of requests by up to 20%. Upon completion of the request, by say a system administrator, HPRCBot would automatically notify the researcher who submitted the ticket. Depending on the complexity and maturity of the software application, it can take minutes or weeks to install it. For requests to install software applications, we note that the HPRCBots primary function is to keep team members abreast about the status of the process, and where it lies in the workflow. This allows the Helpdesk technician to better communicate with the researcher, and not have to request manual updates from the person installing the software application. Statistics for processing these requests were generated by querying the database that collects generation and completion times, and ensuring that all requests and approvals are stored in a standardized format. Generating completion statistics for these 53 requests was facilitated using defined functions that queried HPRCBots database. While this operation took a matter of seconds, a similar query search in our ticketing systems, slack workspace and email to generate completion time statistics would take well over 30 minutes.

5. FUTURE PLANS

Additional features that improve CI Workflows are planned for CI Bot. As of right now, the HPRCBot is a working backend with endpoints that can be integrated to any website, and has Slack integration by default. We are working on create a generic dashboard for staff, and administrators, in which staff can see their assigned requests and take actions, and administrators can customize and tune the bot behavior and workflows, search through requests, and see various statistics. We are also integrating the bot with common ticketing and workflow systems like RT and JIRA Service Desk, allowing the bot to be used better on HPC sites with an already existing ticketing or workflow systems. Additionally, there are a number of other features and improvements we are planning to implement; including a straightforward way to create/manage automated email templates, a webhook to allow better integration with dashboards or ticketing systems, and automations on common HPC sites requests like wall-time extensions. Our goal is to ultimately provide a free-to-use and flexible open-source tool for HPC sites to automate repetitive tasks with better communication and documentation.

6 CONCLUSIONS

This paper introduced CIBot, a workflow management tool. We showed common issues and bottlenecks in processing requests. We believe these bottlenecks are typical for other HPRC helpdesks as well. We proposed a systematic approach to optimize the workflow for any type of request which will improve the user experience. It also eliminates manual and repetitive tasks helpdesk members have to perform when processing requests. Aside from significantly reducing the chance of mistakes and miscommunications, it also frees up time for both helpdesk members and staff that can be spent on more useful tasks. CIBot is currently in production at the HPRC Helpdesk. The goal is to make it publicly available.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation (NSF) award number 2112356 ACES - Accelerating Computing for Emerging Sciences, NSF award number 1925764 SWEETER - SouthWest Expertise in Expanding, Training, Education and Research, NSF award number 2019136 BRICCs - Building Innovation at Community Colleges, NSF award number 2019129 FASTER - Fostering Accelerated Scientific Transformations, Education, and Research, staff and students at Texas A&M High Performance Research.

6 References

- [1] High Performance Reserach Computing, Texas A&M University, <https://hprc.tamu.edu>
- [2] RequestTracker: <https://bestpractical.com/request-tracker>
- [3] ZenDesk: <https://www.zendesk.com/>
- [4] Jira Service Management:: <https://www.atlassian.com/software/jira/service-management>
- [5] Edmundo Medina-Gurrola, Dhruva K. Chakravorty, Diana V. Dugas, Tim Cockerill, Lisa M. Perez, Emily Hunt. (2022). Regional Collaborations Supporting Cyberinfrastructure-Enabled Research During a Pandemic: The Structure and Support Plan of the SWEETER CyberTeam. In Practice and Experience in Advanced Research Computing (PEARC '22), 4 pages. <https://doi.org/10.1145/3491418.3535186>
- [6] Duy Pham, Kyle Hsu, Marinus Pennings, Tri Pham, and Phi Au. (2022). Extending Functionalities on a Web-based Portal for Research Computing. In Practice and Experience in Advanced Research Computing (PEARC '22), 6 pages. <https://doi.org/10.1145/3491418.3535182>
- [7] Michael Lau, Stuti Trivedi, Zhenhua He, Tri Pham, Lisa Perez, Dhruva Chakravorty. (2021) Research Cloud Bazaar: A software defined

cloud workflow cost management tool. In Practice and Experience in Advanced Research Computing (PEARC '21), 4 pages. <https://doi.org/10.1145/3437359.3465602>

- [8] Richard Lawrence, Tri M. Pham, Phi T. Au, Xin Yang, Kyle Hsu, Stuti H. Trivedi, Lisa M. Perez, Dhruva K. Chakravorty. (2022). Expanding Interactive Computing to Facilitate Informal Instruction in Research Computing. Volume 13, Issue 1, pp. 50-54. Journal of Computational Science Education. <https://doi.org/10.22369/issn.2153-4136/13/1/9>
- [9] "slackapi/python-slack-sdk: Slack Developer Kit for Python." GitHub, <https://github.com/slackapi/python-slack-sdk>. Accessed 2 March 2023.
- [10] "omarryhan/aiogoogle: Async Google API Client + Async Google Auth." GitHub, <https://github.com/omarryhan/aiogoogle>. Accessed 2 March 2023.
- [11] "mongodb/motor: Motor - the async Python driver for MongoDB and Tornado or asyncio." GitHub, <https://github.com/mongodb/motor>. Accessed 2 March 2023.
- [12] ClBot Technical Details - https://docs.google.com/document/d/1Owy53HpTjAdHbl_Oo9Rv5tBFwJ7N9yeoinmleZmYM0/edit