WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Modern Workspace Based Policy Management with AI Based Records Management

Deepthi N H[1], Gayathri R E[2], Akshaya Lakshmi T M[3], R . Dhanalakshmi[4]

[1-3]Student, IV Year, Department of Computer Science and Engineering at R.M.K. Engineering College, Kavaraipettai-601206, India.
[4]Associate Professor, Department of Computer Science and Engineering at R.M.K. Engineering College, Kavaraipetta-601206, India.

A B S T R A C T

To create an entitlement-based platform where users can create policies, have them approved through a flow based on metadata information within SharePoint and then view them and edit for future rectification.The purpose of the project is to create a modern workspace-based policy management with both publishing and consumption views with multiple level authoring and approval. Companies have their policies created and maintained locally within the legal department and/or by individual policy owners. These policies reside in the locally within the computers of the policy owners and sometimes in multiple locations, without appropriate controls. In order to rectify this business problem, we are creating thisentitlement-based platform. The business criterions will be taken care by the AI based records management function, a feature of this software. This will be achieved using existing public cloud based cognitive services and leverage Microsoft Office 365 Graph API to provide records management.

Keywords: Policy, Azure Cognitive Services, Power Automate, Cloud Services, SharePoint, Office 365.

1. Introduction

An ad hoc approach to policy management exposes the organization to significant liability. This liability is intensified by the fact that today's compliance programs affect every person involved in supporting the business, including internal employees and third parties. To defend itself, the organization must be able to show a detailed history of what policy was in effect, how it was communicated, who read it, who was trained on it, who attested to it, what exceptions were granted, and how policy violation and resolution was monitored and managed. In order to overcome this business problem, we are going to create an consolidated platform using SharePoint where employees can create their own policies and have them approved through a workflow based on metadata information within SharePoint and can view them and edit for future rectifications. This application enables the automated verification of properties with respect to both the authorizations enforced by single policies and the relationships between multiple policies. Power automate is used to help in multiple level authoring and approving workflow. Azure protection service helps in security of documents from unauthorized access. And azure AI is used for PDF conversion of the policy documents which all the employees of the organization will be able to access them through a common Microsoft office 365 platform is used for security.In this application all the policies are approved through a hierarchy of approvals starting from policy owners and then being approved by different members of the organization.

* *Corresponding author.* Tel.: +918667659981.
E-mail address:gaya17126.cs@rmkec.ac.in

2. Architecture

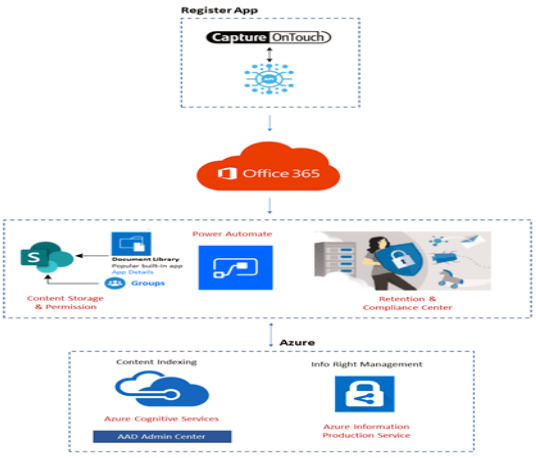

Fig 1 Modern workspace-based policy management

This system is to support provable policy decision making and verifiable attribute tokens among multiple authorities.

1. SPFx - This is used for developing custom form and grid web parts
2. SharePoint Online - Repository for storing the Policy documents.
3. Power Automata - Used for Approval and Email notifications workflow.

Once the Form is uploaded, the power automata workflow will start and the Email generation process takes place in that.

3. Related Search

3.1 A markov-based update policy for constantly changing database systems

Author: Wei zong, Freng Wu, ZhengruiJiang

Despite the importance of data timeliness, updating a database as soon as new data arrives is typically not optimal because of high update cost. Therefore, a critical problem in this context is to determine the optimal update policy for database systems. In this study, we develop a Markov decision process model, solved via dynamic programming, to derive the optimal update policy that minimizes the sum of data staleness cost and update cost. Based on real-world enterprise data, we conduct experiments to evaluate the performance of the proposed update policy in relation to benchmark policies analysed in the prior literature. The experimental results show that the proposed update policy outperforms fixed interval update policies and can lead to significant cost savings. Keeping data up to date in a database system is critical in order to maximize the value of an organization's data assets. However, constantly changing data sources often make it a challenging task to assure data timeliness in enterprise database systems. Because enterprise resource planning (ERP) systems are one of the most widely adopted types of software systems in today's organizations. We use ERP systems to illustrate the challenges, the solutions, and the benefits of achieving data timeliness in a modern database system

3.2 A bductive analysis of administrative policies in rule-based access control

Author: Puneet Gupta, Scott D. Stoller, ZhongyuanXu.

Fully understanding the consequences of an administrative policy in an enterprise system can be difficult, because of the scale and complexity of the access control policy and the administrative policy, and because sequences of changes by different users may interact in unexpected ways. Administrative policy analysis helps by answering questions such as user-permission reachability, which asks whether specified users can together change the policy in a way that achieves a specified goal, namely, granting a specified permission to a specified user. This paper presents a rule-based access control policy language, a rule-based administrative policy model that controls addition and removal of facts and rules, and an abdicative analysis algorithm for user-permission reachability. Abdicative analysis means that the algorithm can analyse policy rules even if the facts initially in the policy (e.g., information about users) are unavailable. The algorithm does this by computing minimal sets of facts that, if present in the initial policy, imply reachability of the goal.

3.3 A rigorous framework for specification, analysis and enforcement of access control policies

Author: Andrea Margheri, Massimiliano Masi, Rosario Pugliese and Francesco Tiezzi

In this paper, we introduce a formally defined, fully implemented framework for specification, analysis and enforcement of attribute-based access control policies. The framework rests on FACPL, a language with a compact, yet expressive, syntax for specification of real-world access control policies and with a rigorously defined de-notational semantics. The framework enables the automated verification of properties regarding both the authorisations enforced by single policies and the relationships among multiple policies. Effectiveness and performance of the analysis rely on a semantic preserving representation of FACPL policies in terms of SMT formulae and on the use of efficient SMT solvers. Our analysis approach explicitly addresses some crucial aspects of policy evaluation, such as missing attributes, erroneous values and obligations, which are instead overlooked in other proposals. The framework is supported by Java-based tools, among which an Eclipse-based IDE offering a tailored development and analysis environment for FACPL policies and a Java library for policy enforcement. We illustrate the framework and its formal ingredients by means of an e-Health case study, while its effectiveness is assessed by means of performance stress tests and experiments on a well-established benchmark.

4. Proposed System

SharePoint is used as a primary document repository in this proposed system. There is a built-in entity in the SharePoint which is called as Document Library. This document library is used as repository for storing the documents. These documents can be viewed by the creator/user later. The document library has metadata fields. These metadata fields are used for storing the properties of the document and all other related data. A form will begiven to the users, which they will be using for keying the details of the policy and will upload the policy. Once the user uploads the policy, the uploaded policy will be stored in the document library by using Microsoft Graph REST API call. Once the policy gets uploaded by the user, Power Automate workflow gets triggered and the approval process gets started after this. Here the workflow is responsible for sending email notification and updating the status of the policy based on approver input. After this, the policy will be moved to the appropriate approvers login who decides whether to approve or reject the policy created. They also have the rights to make the policy to be approved by some other authorized approver. Only the approver or the author can view, update and delete the policy before it is fully approved and published. If the policy is getting rejected, then it is the responsibility of the approver who rejects the policy to upload the comments stating the reason for rejecting the policy. Users other than the approver/author do not have access to view the policy which is not yet approved. After approving all the policy by all the approvers, the workflow will call the Azure Cognitive Services and pass the policy document as input to the service. This service will extract the keywords based on the defined text extraction algorithm and populate the metadata for the uploaded policy. The approved policy will be updated in the form of PDF in a common access in order to avoid unnecessary modification done in the approved policy by unauthorized users. At any point of time or in any case the policy author/approver can update or delete the policy which will also be updated in common access listing of all the policies created which can be viewed by other users through web application. The updated policy which are in common access were sorted in the last come first serve order. So, the latest policy will be shown first to the users.

5. Conclusion

Policies in an organization are created and stored at multiple places and all of them are integrated. Thus, with the help of this application an employee can search and have access to the right policy whenever they need it. A rule has been implemented to update the policies after a specific period everything is a audit. When an organization is found with an out-of-date policy, it will be fined, and this application enables automatic policy renewal process. Outdated policies can be retrieved within a specified period. Currently updated policies can be accessed efficiently. Multiple level authored are approved by multiple people before publishing of the policies. The project can be added with automatic renewal based on time validity of policies as a future enhancement. Policies can be moved to a master cloud as certain policies are similar between multiple organisations. In follow up, the policies can be moved from SharePoint Online to Office 365 as it allows more coordination between its Office Apps. It can also leverage cloud services, thus reducing the burden of on-premises server management.

Acknowledgements

We earnestly portray our sincere gratitude and regards to our Chairman Shri. R. S. Munirathinam, our Vice Chairman, Mr. R. M. Kishore, our Director, Shri. R. Jyothi Naidu Principal, Dr. K. A. Mohamed Junaid, Dr. T. Sethukarasi,Assistant Professor, Project Guide Dr.R. Dhanalakshmi, Associate Professor forvaluable suggestions and spontaneous guidance to complete our project.

REFERENCES

Wei zong, Freng Wu, ZhengruiJiang,"A Markov-based update policy for Constantly changing database systems", IEEE Transactions on Engineering Management Volume: 64 , Issue: 3 , Aug. 2018 .

Puneet Gupta, Scott D.Stoller, ZhongyuanXu,"Abductive analysis of administrative policies in rule-based access control", IEEE Transactions on Dependable and Secure Computing Volume: 11 , Issue: 5 , Sept.-Oct. 2020 .

Andrea Margheri , Massimiliano Masi , Rosario Pugliese and Francesco Tiezzi,"A Rigorous Framework for Specification, Analysis And Enforcement of Access Control Policies",IEEE Transactions on Software Engineering,Volume: 45, Issue: 1 ,24 October 2019.

Masoud Narouei, Hassan Takabi and Rodney Nielsen,"Automatic Extraction of Access Control Policies from Natural Language Documents",IEEE Transactions on Dependable and Secure Computing,23 March 2018 .

Eiji Adachi Barbosa ; Alessandro Garcia ; Martin P. Robillard ; Benjamin Jakobus,"Enforcing Exception Handling Policies with a Domain-Specific Language",IEEE Transactions on Software Engineering,Volume: 42, Issue: 6,01 JUNE 2018.