IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

A dynamic access control model using authorising workflow and task-role based access control

Mumina Uddin[1], Shareeful Islam[2]

[1]Information Risk Manager - PWC and PhD student, University of East London, U1146764@uel.ac.uk
[2]School of Architecture Computing and Engineering (ACE), University of East London, London, E16 2RD,shareeful,@uel.ac.uk

Corresponding author: e-mail: U1146764@uel.ac.uk, shareeful@uel.ac.uk

ABSTRACT Access control is fundamental and prerequisite to govern and safeguard information assets within an organisation. Organisations generally use web enabled remote access coupled with applications access distributed on the various networks facing various challenges including increase operation burden, monitoring issues due to the dynamic and complex nature of security policies for access control. The increasingly dynamic nature of collaborations means that in one context a user should have access to sensitive information and not applicable for another context. The current access control models are static and lack of Dynamic Segregation of Duties (SoD), Task instance level of Segregation and decision making in real time. This paper addresses the limitations and supports access management in borderless network environment with dynamic SoD capability at real time access control decision making and policy enforcement. This research makes three contributions: i) Defining an Authorising Workflow Task Role Based Access Control using the existing task and workflow concepts. It integrates the dynamic SoD considering the task instance restriction to ensure overall access governance and accountability. It enhances the existing access control models such as RBAC by dynamically granting users access right and providing Access governance. ii) Extended the OASIS standard of XACML policy language to support the dynamic access control requirements and enforce the access control rules for real time decision making to mitigate risk relating to access control such as escalation of privilege in broken access control and insufficient logging and monitoring iii) The model is implemented using open source Balana policy engine to demonstrate its applicability to a real industrial use case from a financial institution. The results show that, AW-TRBAC is scalable consuming relatively large number of complex request and able to meet the requirements of dynamic access control characteristics.

INDEX TERMS: Identity and Access Management, Role Based Access Control, eXtensible Access Control Markup Language, Attribute Access Control, Dynamic segregation of duties

I. INTRODUCTION

Identity and access management (IAM) is a framework for business processes that facilitates the management of legitimate user identity and access control of business sensitive assets. The term access control refers to an organisation's policy for authorising process for access, the mechanisms provides and enforces the policy and the model which on the policy and process is based on. There are two fundamental types of access control; Discretionary Access Control (DAC) and Mandatory Access Control (MAC). While initial research and applications addressed preventing the unauthorized access to the classified information, recent applications have applied these policies to commercial environment (O'Connor et.al, 2010). Other research considered the approaches on the decentralised granular level of entitlements such as; Role Based Access Control (RBAC) (Rajpoot. et. al, 2015), Attribute Based Access Control (ABAC) (Biswas and Sandhu, 2016), XACML (Oasis,2010), and Risk Adaptive Access Control (RAdAC) (Farroha and Farroha, 2012). Organisations are now dynamically changing roles to the users or revoke any existing role due to the various business demands. There are many applications that are running from an outsourced environment such as cloud or from supply chain partners, which needs to deal access control dynamically comparable to the traditional in-house application. Despite of the significant development, there is a lack of focus of the existing access control models for granting access, enforcing dynamic SoD (Segregation of Duties) (Ma, et.al, 2011), BoD (Binding of Duties) and access governance through workflow management (Crampton, 2004).

The novel contributions of this research are: Firstly, a dynamic access control model, i.e., Authorising Workflow Task Role-Based Access Control (AW-TRBAC) , is proposed that uses dynamic segregation of duties (SoD) and process workflow and considers the task instance restrictions for the roles restriction, access governance and logs (Compliance). The approach uses the existing task and workflow concepts to build identity and access management solutions. Therefore, this research enhances the existing access control models such as RBAC and ABAC by dynamically granting users access right to promote access governance and risk mitigation. Secondly, this work extends the OASIS standard of XACML for developing a dynamic access control policy language so that it can enforce the access control rules and additional functionalities to enforce SoD at the task instance,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2947377, IEEE Access

remediating broken access control risk (OWASP,2019). Through the logging of instance task events, it enhances access governance to provide visibility of unmanaged data. Finally, the model is implemented using the open-source Balana policy engine (Chen and Gasparini,2013) to demonstrate its applicability using an industrial use case of a financial institution to test the performance and alleviate the risk of escalation of privilege and data disclosure. The test results show that, AW-TRBAC does not impact on overall system performance despite of changing user access request dynamically and mitigate the risk of escalation of privilege to prevent data disclosure.

II.RELATED WORK

There exists several widely used access control models, however, wide adoption of these models remains a challenge. This section presents the existing access control models and works that focused on the identity and access control management.

A.ACCESS CONTROL MODEL

RBAC is a framework using roles to control access to resources, permissions are grouped into a role, a role will have several members and will have set of defined granular level of credentials (Fenstein and Youman,1996). The RBAC model achieves the two principles of Security systems "Least Privilege "and "Segregation of Duties". In recent years, a considerable amount of work has been done on the use of RBAC to support access control in workflow systems (Wainer and Barthelmess, 2003). However, role-based access control has its own set of limitations such as role explosion and role-permission explosion (Rajpoot. et. al, 2015). Termination of the role which has not been defined in the NIST standard for RBAC, this feature has not been defined in any authorisation model, if a user terminated (revoked) what happens to their session, which has been activated through role, should the role be terminated instantly or retained for a period before terminating it (deleting it) (Thomas and Sandhu, 1997). The TBAC is lacking from rules for revoking of user sessions immediately and retaining the session active for a period while disabling the account for audit purposes when requested .

Attribute Based Access Control (ABAC) have no consensus model up to date (O'Connor et.al, 2010), the concept is that users and resources have attributes known about them, either through situational data, such time of the day, person logged on to the network, or the user data such as title or location. ABAC is more flexible than RBAC because it does not require separate roles for relevant sets of subject attributes, and rules can be implemented quickly to accommodate changing needs. Disadvantages are, on the other hand, that access control policy might become more dynamic than preferable for audit and attestation, it requires many rules which makes the analysis difficult (Wang, et.al,2004). The user entitlement access report is difficult to retrieve as the access is based on attributes rather than entitlements. ABAC (Hu and Kuhn, 2015) has been around for over two decades and numerous models (Biswas and Sandhu,2016) have been proposed. . Despite the existence of these different ABAC models, there is no consensus on a specific standard ABAC model (O'Connor et.al, 2010) Recently, multiple dynamic attributes such as application usage and unlock failure is considered for ensuring access control and data confidentiality of mobile cloud environment (Neah and Shashikala,2019). The model needs preinstallation in the hand set to capture the attributes and addresses mobile authentication adversary. The results show an efficient uninterrupted communication between the users and the cloud storage server. Three factors authentication scheme is proposed by (Jolfaei et al ,2019) for wireless sensor networks. Formal and informal security analysis is done of proposed protocol using known and unknown attacks such as stolen smart card attack and privileged insider attack. The result shows that the approach is more secure and efficient than the existing schemes.

XACML is based upon XML and was developed to specify access control policies in a machine-readable format (Oasis, 2010). Policy creation can be complicated and the use of XACML does not necessarily make the task of creating, specifying, and enforcing good access control policy any less difficult. There is also a need to ensure that the entire enterprise uses the same attributes for access, and that all the attributes are from an authoritative source. In simple terms, an Authoritative Attribute Source (AAS), policies should be able to specify which sources of attributes are authoritative for the policy, and there should be mechanisms to verify that the attributes provided by a requester come from the AAS. According to (Ma, et.al, 2011) policy-based workflow management (PBWF), which entails policies based on the business processes, including access control, authorisation and authentication. Authors have used the notation of TBAC and RBAC to depict the flow of information and shown both dynamic and static access control needed in a workflow. However, there are various authorisation policies within the organisation which has not been studied. Furthermore, there is also a lack of focus on resolving conflicts among the different authorizing policies within the access control models.

B. ACCESS CONTROL ADVERSARIES

There are works that focus on the vulnerabilities and risks of existing access control model. A survey results from (OWASP, 2017) mentioned two critical access control risks. i.e., Insufficient Logging and monitoring and broken access control. The design of the adversary is based on the established STRIDE (Spoofing, Tampering, Repudiation, Information disclosure, Denial of Service, Elevation of privilege) and DREAD(Damage, Reproducibility, Exploitability, Affected users, Discoverability) exploitability model to determine the likelihood of adversary(Do et.al, 2019). Typically, the goal of the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2947377, IEEE Access

adversary is to disrupt or prevent proper operation of a secure system. Exploitation of insufficient logging and monitoring is the bedrock of nearly every major incident. Improper or absence of proper logging of events, failed logins transactional logs allows attackers to further attack systems, maintain persistence, pivot to more systems, and tamper, extract, or destroy data. Broken access control confers to restrictions on what authenticated users are allowed to do which are often not properly enforced, risk associated with this is attackers can exploit these flaws to access unauthorized functionality and/or data, such as access other users' accounts, view sensitive files, modify other users' data, change access rights, etc. Exploitability occurs when the attacker changes the parameter value, which directly refers to a system object for which he is unauthorized through applications and APIs where all the user request privileges are not verified resulting in Privilege escalation. Another research on the risk Adaptive access control (RAdAC) uses the information from the environmental condition and risk level, it combines information about a subject machine, corporate IT infrastructure, environmental risk factors for the decision making process (Farroha and Farroha, 2012). An advantage of this approach is if the policy allows then it can override the decision in a situation where necessary, for example, in a high security it will enforce dual authentication and in a relaxed secure it will make a decision based on the digital policy. However, we have also observed limitation of this work , specifically similar to PBAC, it relies on digital policies. If the policy is ambiguous, then it could result towards possible security breach.

All these works above are important and contributed towards the improvement of identify and access management. However, it has been observed several limitations within the Dynamic SoD, BoD and access governance and risk mitigation. This research contributes to bring all silo components into a whole novel access control to fill the gaps of Dynamic Role Change, SoD, BoD, governance and policy compliance to improve risk posture

III.AN INDUSTRIAL USE CASE AND LIMITATIONS

As mentioned before this research is based on an industrial use case which is presented briefly by this section. The use case is about an investment bank located in London. Due to the confidential reason, identity of this institution will not be disclosed and only use the non-confidential information about the business as running example to demonstrate our approach. The organisation has asset over trillion dollars and located in 17 locations worldwide. It is a privately held financial institution and has been a leader and a solution provider for over 200 years. The company is expert in Corporate Banking, Merger & Acquisition advisory, Investment Management, and wealth management and investor services.

Access Provisioning process triggered by the security coordinator it is then "Security Administrators" who close the Access request after access has been provisioned this enables segregation of duties (SoD). There is additional level of SoD between the roles "Security Coordinator" and "Security Request Approver" security Coordinator cannot be the "approver". However, there are certain cases where the approver of one workflow is the submitter of another and it is not possible to enforce this restriction based on a request.

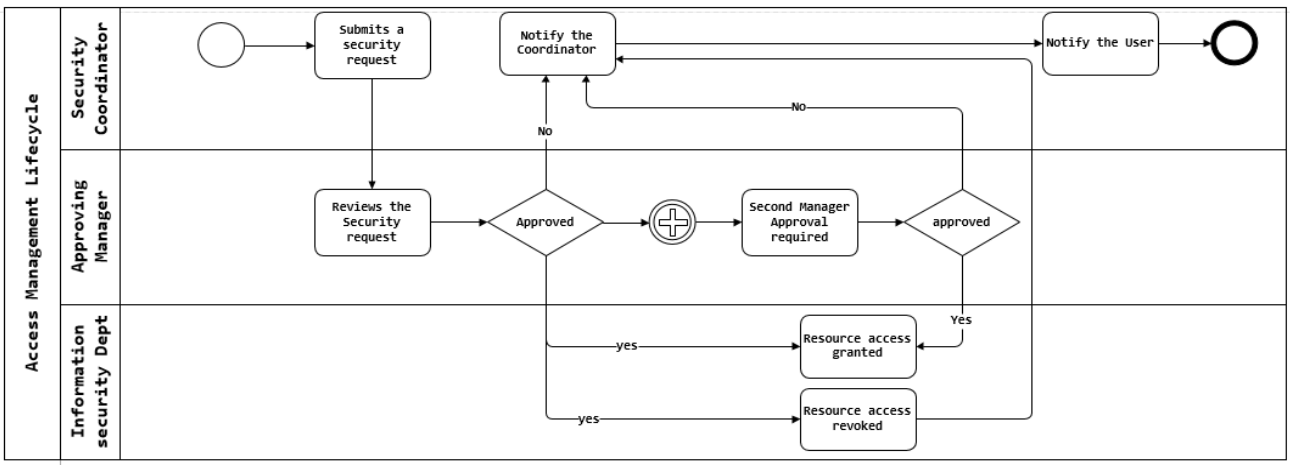

Figure 1: **Current Access Management Lifecycle changes made to the diagram**

The access control workflow process for high security environment of this Investment Bank would be submitted via Coordinator into a security request database, which will then send notification to the approving manager for approval. The request will reach the Information Security Department to be actioned (Fig1). Challenges are number of different silos authorisation systems operating independently, lacking in governance and user access reconciliation. The organisational security policies and rules of "least privilege" and "separation of Duties (SOD)"

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2947377, IEEE Access

to minimise fraud and error becomes onerous. role level restriction, it requires restriction within task level which is referred to as "instance level restriction".
In a more dynamic process where role transfer or departmental transfer, two levels of approval required, approval from manager of the transferring department as well as the approval from the manager the user is transferring to. It is essential to enforce that the 1st approver approves the requests before the 2nd approver; sequence of tasks needs to be maintained. The old credential needs to be revoked before granting new credentials. Hence, the organisation is facing various challenges for the overall access control management.

- There is a lack of dynamic access control to accommodate the diverse hosting of information. This could also impose the escalation of privilege risk
- By passing vetting process, no visibility of data access due to inadequate business process workflow.
- Lack of visibility and access control governance due to inadequate access control policy verification and limited support for centralized identity repository
- Processes are manual, cumbersome and inconsistent between business units due to missing streamlined access management process across business. This makes the governance of Access Management becomes cumbersome as multiple silos systems is resort to for validations.

IV.THE PROPOSED AW-TRBAC

The proposed Authorising Workflow Task Role Based Access Control (AW-TRBAC) model considers granting users access rights through role change in a dynamic context. The proposed model consists of several components such as requirements, conceptual view, policy language, and Proof of Concepts (PoC). These components support to present the AW-TRBAC in detailed and presented in this section.

A.REQUIREMENTS

From the analysis of the Use Case as stated before and the review of the existing access control models: RBAC (Fenstein and Youman,1996) and ABAC (Biswas and Sandhu, 2016), a set of characteristics and specification derived for the functional requirements of the Dynamic Access Control Model. We have considered a list of requirements which are necessary to support the characteristics of emphasis dynamic segregation of duties and access governance.

- Requirement 1: Access request shall only be submitted by the role "Security Coordinator".
- Requirement 2: Authorised Business Process Owner should approve the Security request
- Requirement 3: Only Authorised users shall be permitted access to resources.
- Requirement 4: User Access to be revoked after termination of service and service change.
- Requirement 5: Service transfer through role change requires two level of Process Owner approvals (departing Service and Onboarding) and revocation of existing and provisioning on new credentials.
- Requirement 6: Sufficient logging of events to be retained and monitored.

Requirements 1,2,3 have been addressed in various access control models independently, (Ferraiolo and Sandhu, 2001), which is very relevant, well-known and most used in the security area. There are many variations of constraints of SoD, despite various research, it imposes challenge in dynamic environment. The focus in this paper is Requirements 4,5, and 6, which are unique functional requirements for the Dynamic access control model to meet dynamic borderless network environment. Requirement 5 which is role change, it has tasked constraints of SoD, BoD, revocation of access, this is an integral requirement which has dependencies on the functionality with other requirements. This paper focuses on the Role Change process and associated dependencies with the other processes.

B.CONCEPTUAL VIEWS

The second component of the model is the concepts necessary to define the AW-TRBAC. It is based on the existing identity and access control concepts such as user, Role, Permission and considers new concepts such as Task, IT Workflow.

User: Users are the subjects of an access control, they execute their job function to achieve the company's goal. They produce business information and this information is stored for future business activities. They may use information resources that were created by other employees.
Task: The concept, task is a fundamental unit of business work or business activity. 'Job function' is another expression of task. Tasks are assigned to users by their job positions or business roles. At the access control's point of view, users read or write information objects when executing their tasks. Access rights are required only for executing the assigned tasks. For Example, 'Material resource planning', 'check issuing', 'purchase approval', and 'sales decision', are examples of tasks. Tasks are assigned to users by their job positions or business roles
Workflow: is an IT term of business process. In general, it means a product or method for supporting business process in the enterprise environment. The task 'approve customer orders' belongs to receiving customer order process. Executing tasks in the business process should submit to a defined process order and available time. Although task 'approve customer orders' is assigned to the user, can

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2947377, IEEE Access

activate their access rights when the prior tasks 'check customer credit' and 'check product stock' are completed. In this case, authorization (access right assignment) is separated from activation of access rights. This case of access control is called active access control

Resources: Information resources are the objects of access control, such as files, tables in a database, executable programs, etc. Information resources contain business information and support to execute the task within workflow resource that can be tangible and intangible.

Business Process: Is a collection of linked tasks which find their end in the delivery of a service or product to a client. A business process has also been defined as a set of activities and tasks that, once completed, will accomplish an organizational goal. A business process is an access control management in Information security function.

Execution List: An execution list is for all users who performed certain task instance, this will contain name, role and task that has been performed by a user. It lists transaction logs of an event that has been auctioned by certain user, which is used in incidence response root cause analysis and compliance. This is a critical control within information security for data analytics as well.

Figure 2 shows class model of AW-TRBAC, it shows the class user, which has a direct association link to Business Process class, as a user belongs to a business process. Role class has a composition relationship with the Task class, a role may have many tasks associated with it. Workflow is another class that has three generalised (association) classes Termination, Role Change and Emergency Password (privileged account). Workflow can have termination request, role change request and Emergency Password request, each of the request has a task. An activation class has inheritance association with Task class and association link with Task instance class, the task is only activated if the condition is met with the task. Execution list class has inheritance relationship with the Task Instance ID class by obtaining the list of executioners from the execution task class for historical information

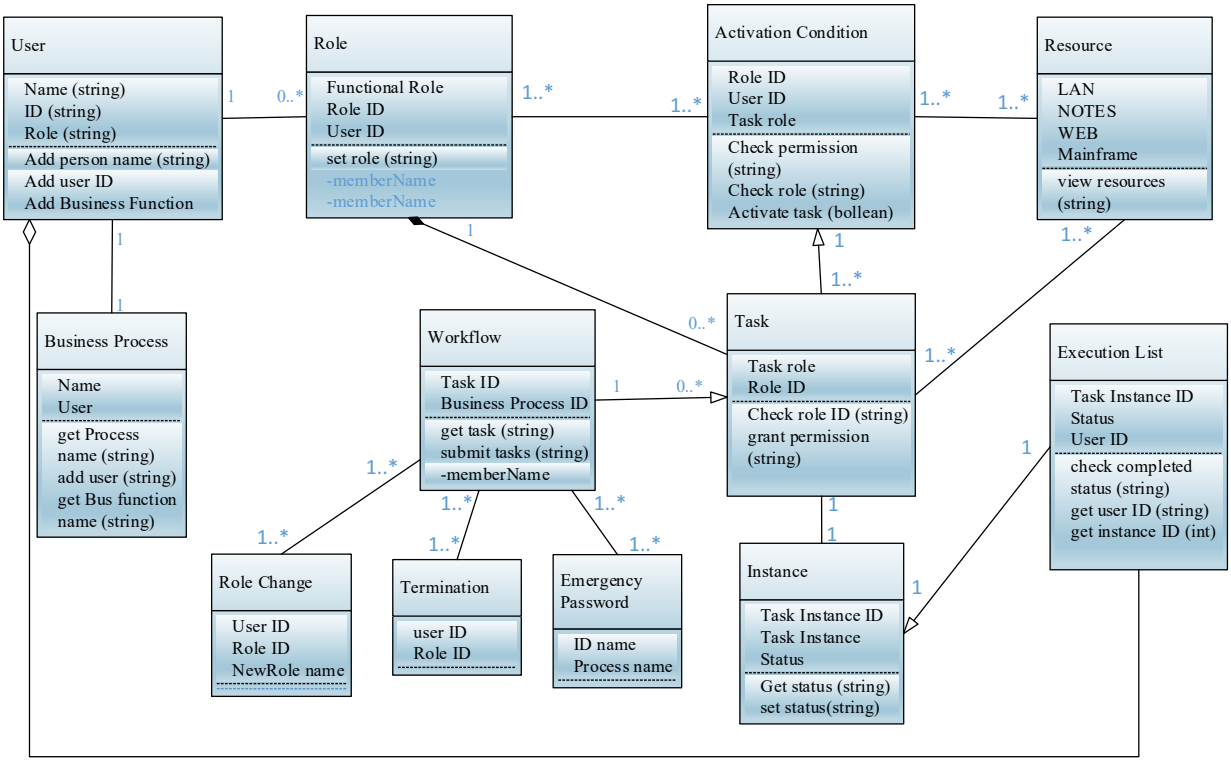

Figure 2: **Conceptual view of the AW-TRBAC access control Model**

C.EXTENSION OF XACML STANDARD

As stated previously, this research extends the XACML standard to support the implementation of the dynamic access control, to meet the Use Case requirements and enforce the rules of the dynamic access control model through policy language. XACML support the notation of the proposed dynamic access control model, such as Role, Task, Operation, so that it can act as policy enforcement, which interact with the access control model to make decisions. The focal point of this research is on dynamic Role Change, SoD, BoD functional and security requirements to enhance the risk posture and visibility. To satisfy the use case requirements, five new functions and two new data stores has been introduced, which are utilised by the XACML policy engine in the decision making. These extended functions enable dynamic access control model to provide real time history-based instance-level segregation to mitigate the risks of broken access control

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2947377, IEEE Access

and insufficient logging events. XACML is an OASIS standard that defines a general purpose access control and authorisation system (Oasis,2010). It consists of a policy language based on XML and a processing system that knows how to interpret the policy with respect to the relevant application. The policy language is used to create policies where each policy enlists the requirements to access a resource in a protected environment.

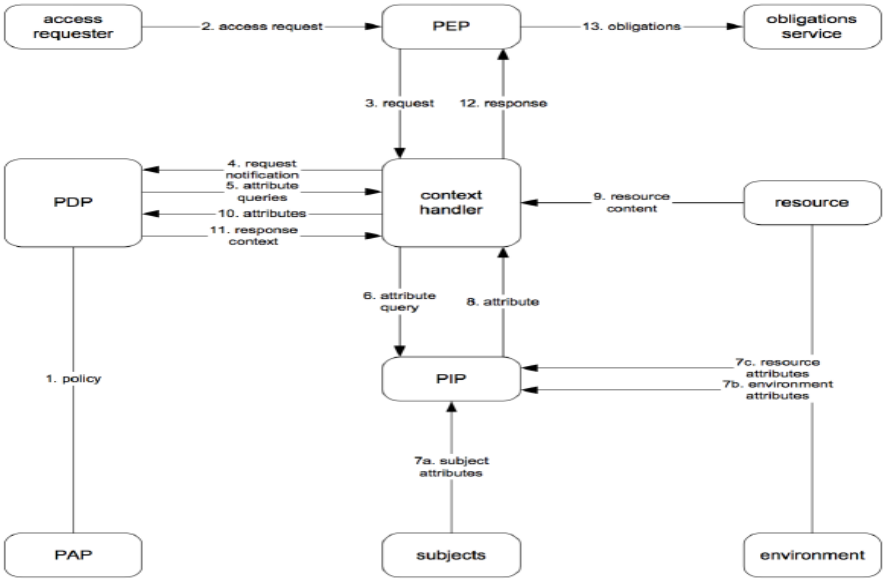

Figure 3: **XACML Standard (Oasis, 2010)**

As shown in the Fig 3 above, the major components of XACML Standard are;

Policy Administration Point (PAP), which handles creating and managing all policies. Policy Enforcement Point (PEP), which handles intercepting users' requests and enforcing XACML decisions received from the Policy Decision Point (PDP). Policy Decision Point (PDP) handles evaluating users' requests based on the existing policies and return XACML decisions to the PEP. Finally, Policy Information Point (PIP) facilitates gathering additional attributes of a user.

AW-TRBAC model extends the existing RBAC-XACML oasis standard by introducing two new Repository called Role Change store and Role Assigned Task store and five new functions: SoD check, BoD check, Role Check, Role Change check and Role change approve checks (**coloured in blue** Fig 4 below). Each function is utilised for the different security request, for example SoD Check function will be utilised for requests that require segregation of duty constraints on submitter and approver role, it contain conditional obligations to enforce policy rules.

As shown in Fig 4 below, Context handler is responsible for translating the received requests into the XACML context and the reverting results back to the native language of the system and communicating between the other components. In XACML the PDP (Policy Decision Point) handles decisions making of the authorisation requests based on the policy sets. With RBAC-XACML standard (Oasis,2010) there has been a new type of request that deals with role activation, it is decided that role activation should be out of the scope of PDP. For this reason, the Role Enablement Authority (REA) is introduced as a specialised repository containing a policy store in the decision making for role activation. The purpose of REA is to show existing XACML-RBAC oasis standard architecture. AW-TRBAC has introduced a new type of request, to perform a workflow task. To deal with such request, this research extended the XACML OASIS standard with five new functions: SoD check, BoD check, Role Check, Role Change check and Role change approve checks, Lastly, the PDP functionalities are extended by using Context Handler to query the additional functions and data store; Role Change store and Role Assigned Task store, to forward the request to the PDP for decision making.

D.XACML POLICY REQUESTS

This section describes attributes in an XACML policy request, XACML policy has been defined to meet the use case policy requirements as part of the XACML extension, as shown below in Figure 5. Access to resource request sent to the policy engine to be authorized by one or more policies, such requests need to be composed in a structured way that can be utilised by the policy execution engine. A policy request is divided into three parts: subject, resource and action. A subject is defined as the user (request originated from), is implemented using XACML as User. Objects are expressed using XACML Resources such as files, web services. Operations are expressed using XACML as Actions. Permission is the ability or right to perform some action on some resource, possibly only under certain specified conditions. The term "attribute" refers to an XACML <attributes>, is an element in an XACML request having among its components an attribute name, identifier, a data type identifier, and an attribute value. Each is associated either with one of the subjects (Subject

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2947377, IEEE Access

Attribute), the protected resource (Resource Attribute), the action to be taken to the resource (Action Attribute), or the environment of the Request (Environmental Attribute). Illustrated below is an xacml policy to enforce SoD in AW-TRBAC.

```
<Request
xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
Combined Decision="false" ReturnPolicyIdList="false">
 <!-- Task -->
 <Attributes Category="urn:uel:ac:uk:xacml:3.0:task-
category:access-task">
  <Attribute
AttributeId="urn:oasis:names:tc:xacml:1.0:task:task-id"
IncludeInResult="false">
   <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">sec
urity-request-approve</AttributeValue>
   </Attribute>   </Attributes>
 <!-- User -->
 <Attributes Category="urn:oasis:names:tc:xacml:1.0:subject-
category:access-subject">
  <Attribute
AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
IncludeInResult="false">
   <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">ma
t</AttributeValue>
   </Attribute>
  </Attribute <!-- Policy to match -->
  <Attributes
Category="urn:oasis:names:tc:xacml:3.0:attribute-
category:environment">
   <Attribute
AttributeId="urn:oasis:names:tc:xacml:1.0:environment:envir
onment-id"IncludeInResult="false">
    <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">SE
G002</AttributeValue>
    </Attribute>
   </Attributes>
 <!-- Task instance reference -->
 <Attributes Category="urn:oasis:names:tc:xacml:1.0:subject-
category:access-resource">
  <Attribute
AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-
id" IncludeInResult="false">
   <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">tif
917803b</AttributeValue>
   </Attribute>
  </Attributes>
</Request>
```

Figure 5: **AW-TRBAC SoD Xacml Request for SoD**

V. DESIGN & ARCHITECTURE OF AW-TRBAC

This section presents the architecture and design of the Proof of Concept (POC). It also provides a high-level overview of the implementation of the solution, and its integration with the existing access management policies within the use case. XACML-based policy language has been considered to support requirements as policies to validate on IT workflow task and APIs to integrate the AW-TRBAC model with WSO2 product (balana Open Source) service, to facilitate the additional functionality capabilities, dynamic SoD and IR (instance level Restriction), which currently not supported by Balana Engine.

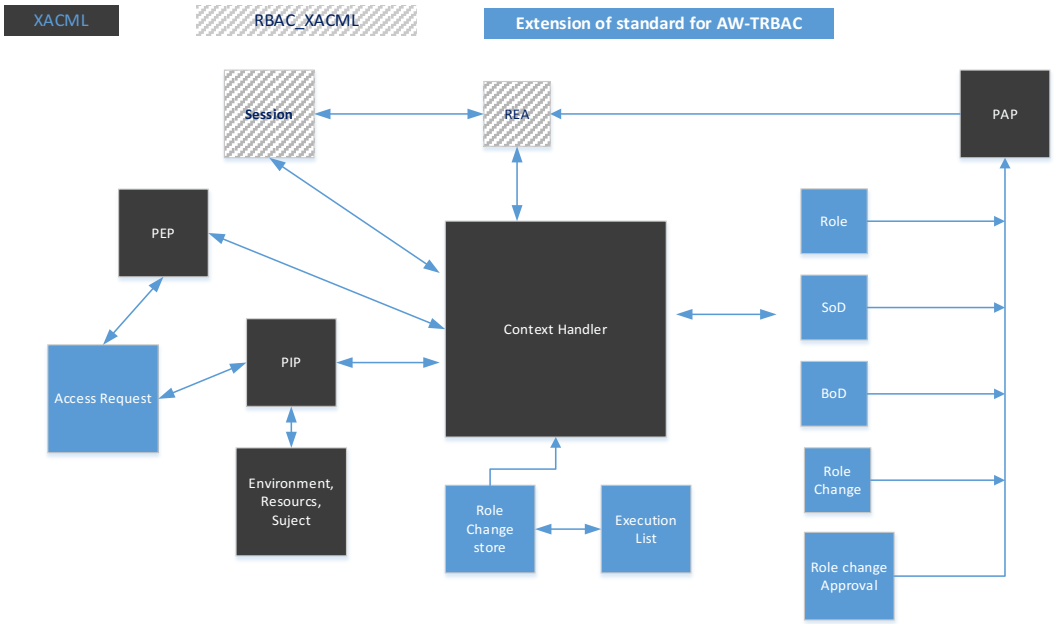

Figure 4: **Extended XACML Standard**

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2947377, IEEE Access

In this research approach, the web portal is the main gateway to the AW-TRBAC, (as shown in Figure 6 high level architecture of the AW-TRBAC model), use REST API request, which is then processed by extracting various information from the security request into an XACML equivalent request. The next component is the Authorisation to access secured resource/service. User identity will be verified against the XACML policies and executed in the AW-TRBAC engines (extended XACML engine) leveraging data store, task services and policy stores to provide correct permission required for the roles, decision is then passed on to the PEP module to direct to the user. This allows the authorised user to access to resources with appropriate permissions.

The web portal is the initial point a service, in invoking various authorization requests. The invocation is done through a series of REST calls. There is a total of three REST URI's implemented for the portal: Role_Change, SOD and BOD. For the illustration purpose in this research, the first operations of Role Change will be explained "initial-request", "first_manager_approval" operation and the Xacml request/response associated with the Operations.

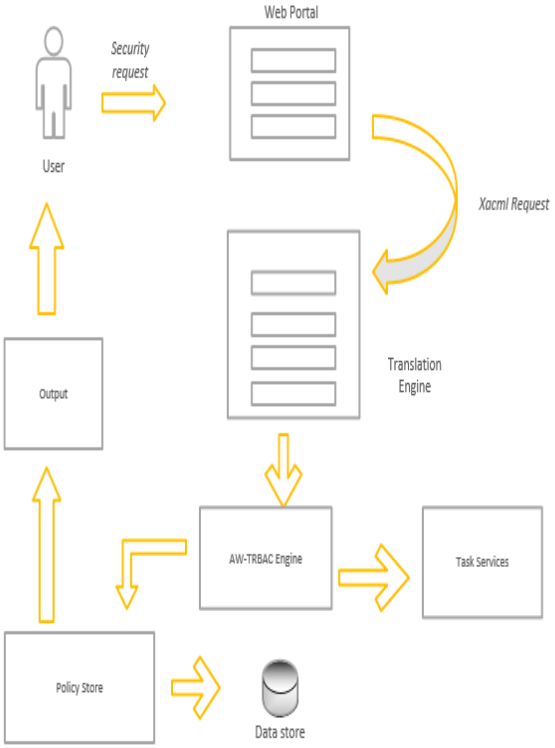

Figure 6: **High Level Architecture of AW-TRBAC**

A. ROLE CHANGE URI

The URI for the role change is as follows for the illustrations purpose.

http://<profdoc.uel.ac.uk>/awtrbac/portal/auth/task/rolechange/{operation}

The following operations are supported within role change request: initial-request, first-manager-approval, second-manager-approval and close-request. The initial Role Change request operation for role change is as follows:

http://<profdoc.uel.ac.uk>/awtrbac/portal/auth/task/rolechange/initial-request

Request Parameters: Table 1 presents the initial role change request parameters using JSON structure including UserId, RoleId, TaskId and resource. The request is activated when the user initial a role change, These parameters are used by the portal to construct XACML request using POST method.

TABLE I
REQUEST OPERATION "INITIAL ROLE CHANGE REQUEST"

Resource Information	Description
Operation	Initial-request
Request Body	UserId - Unique identifier of a subject (user) RoleId- Unique identifier of a role Taskid- Unique identified of an IT workflow ResourceId - Unique service/object id
Request format	JSON
Action	POST / awtrbac/portal/auth/task/rolechange/initial-request

*Response :*The table 2 above shows the response structure of a role change request using JSON format. A successful response will return a code of 201, otherwise a 400 for a bad request. The necessary parameters are taskinstanceid and status of the request.

TABLE 2
RESPONSE PARAMETERS INITIAL_REQUEST

Resource Information	Description
Response Code	HTTP/1.1 201 created
Response error	HTTP/1.1 400 bad request
Response Body	TaskInstanceid, status
Response format	JSON

As shown below in Figure 7, an example of XML request & response in Role Change request. It contains the URI, operation, body of the request, action and response format.

```
Request: POST /awtrbac/portal/auth/task/rolechange/initial-request http/1.1
Host:profdoc.uel.ac.uk
Content-type: application/json
Content length: nnn
```

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2947377, IEEE Access

Response: HTTP/1.1 210 Created {"response":{"TaskInstanceid": "t001", "Status":"successfully created role change request" } }

Figure 7: **XML Request Change**

Operation: ***MANAGER APPROVAL URI FOR ROLE CHANGE***

As shown below in Table 3, the First Manager Approval request. The table 3 shows the first approval request structure. The request body is also JSON document which contains the UserId, TaskId and TaskInstanceId. The following URI shows how the first manager approval process, which takes in various parameters as a JSON entries:

First Manager Approval REST URI Operation:
http://<profdoc.uel.ac.uk>/awtrbac/portal/auth/task/rolechange/first-approval

TABLE 3
REQUEST OPERATION PARAMETERS FOR THE "FIRST MANAGER APPROVAL"

Resource information	Description
Operation	First_approval
Request Body	UserId - Unique identifier of a subject (Manager) Taskid- Unique identified of an IT workflow TaskInstanceid – Unique identifier of the task instance
Request format	JSON

VI.IMPLEMENTATION OF AW-TRBAC POC (PROOF OF CONCEPT)

The PoC implementation language of choice is Java (Gosling, 2000) for the AW-TRBAC. It is a well-established, mature and relatively secure language extensively used in the industry. It has a large range of open-sourced projects, technologies and frameworks available with large community support. Java related technologies such as JAXB (Fialli and Vajjhala,2003), JAXP (Sun Microsystem), JAX-RS (Li,2011) are used to develop the backbone of the framework, that includes processing and handling of XML (Bray and Paoli,1998), REST (Pautasso,2009) based APIs and interactions amongst the framework components. The back-end server is based on the largely adopted Tomcat Server.

The core part of the system is the AW-TRBAC policy language to leverage on the latest industry standard XACML 3.0 (Rissanen,2010). The open source implementation of this standard is the Balana (Chen and Gasparini, 2013) by WS02. The policies within the XACML reliant on the requests, it contains conditional statement and target, which are derived from the requests to allow or deny resource access. IT workflow task on the other hand is not solely dependent on the request values. This requires XACML XPath functions to operate on the data store. However, they restricted to content' XML from the request. While it may take some values from the request, policies are primarily focused on the data from the Data stores for its assertions. To overcome this issue, this research introduced new function of the target to meet the additional requirement to provide the dynamic SoD instance level restriction. Role Check Function has been considered for PoC demonstration as it a fundamental part of the AW-TRBAC model.

A. ROLE CHANGE FUNCTION

Role Check function validates Coordinator role, ensuring request is authorized. The target statements for this function (defined as an ID) are handled by this function, it matches against the Role store (see 1a, 1b and 1c on Fig 7 below). When request is received the function checks the user ID against the role within the user role store, if the user role match is "true" then it updates the Role Assigned Data store with the entry and response back with decision true or false.

urn:uel:ac:uk:xacml:3.0:function:role-check

.

B. STATIC & DYNAMIC SEGREGATION OF DUTIES CHECK (SOD)

SoD checks function validation and enforce segregation between the Coordinator role and Manager role at the task instance level, to ensure authorised users are allowed to perform action upon sequence of tasks. IT workflow task require static and dynamic SoD for its statements. One statement may generate reference IDs stored in a variable, which is later required/used by another statement, such a concept is not present in XACML. To address this issue, this research introduces another function:

urn:uel:ac:uk:xacml:3.0:function:sod-check

The `instanceid' and `new' variables are declared in the target section of the policy. The instanceid value is extracted from the input request type (e.g. Subject) compared against the subject ID in the role assigned task store, to check that the submitter is not an approver and a new status of the task instance is stored in the role assigned task store (see 2a-2d in Fig 7). For a 'new' variable it creates an entry in the store for the statement, assertion (see 3a-3b). The content is of a new variable populated and used by the conditional statements.

C.STATIC & DYNAMIC BINDING OF DUTIES (BOD)

BoD check function will enforce restriction on Coordinator role through validation of role check and task instance ID. IT workflow task requires Binding on Duties (BoD), which entails match ID against the instance ID in the target

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2947377, IEEE Access

section of policy. To address this functionality, we have introduced another function:

urn:uel:ac:uk:xacml:3.0:function:bod-check

The `instanceid' and `new' variables are declared in the target section of the policy. The instanceid values are extracted from the input request type (e.g. subject) compared against the subject ID in the role assigned task store for match and a new status of the task instance is stored in the role assigned task store (see 3a-3c).

D.SINGLE TO MULTIPLE MAPPING (Role Change)

XACML conditional statements are single value entry attributes, whereas IT workflow task statements are multi-valued parameters. To map single-to-multi-values, fourth function is considered:

urn:uel:ac:uk:xacml:3.0:function:role-change-check

This function first obtains the attribute value of an XACML policy conditional statement (this value needs to be a unique ID). It uses this as an xPath reference to a role Policy. If a match is found, the role and its properties are matched against the Role Change store. If all is successful, it will return true, otherwise false (see 4a-4d)

E.SINGLE TO MULTIPLE (Role Change approver Check)

For the static & a dynamic change of role it requires single to multiple valued parameter with multiple conditional statement and policy enforcement to generate an outcome result to grant/deny. This function is the most complex function as it carries out two levels of approver check; one for existing managers in the existing department to approve the task role change, then the onboarding manager approval for the new role change. To carry out task in sequence and carry out SoD check, three different policies are incorporated in a conditional statement with variable parameters. To solve this issue a fifth function has been created.

urn:uel:ac:uk:xacml:3.0:function:role-change-approve-check

This function first obtains the attribute value of an XACML policy conditional statement (this value needs to be a unique ID). It uses this as an xPath reference to a role Policy. If a match is found, the role and its properties are matched against existing department business process store Role Change store. It then requests approver check in the role assigned task store and approve the request. If all is successful, it will return true, otherwise false (see 5a-5g Fig 8(b)). For a `new' variable it creates an entry in the role change store and the Role assigned task store for the statement assertion. The content is of a new variable populated and used by the conditional statements.

urn:uel:ac:uk:xacml:3.0:function:role-change-approve-check

The second condition of this function is to carry out 1st approver checks before 2nd approval. It obtains the attribute value of an XACML policy conditional statement (this value needs to be a unique ID). It uses this as an xPath reference to a role Policy. If a match is found, the role and its properties are matched against the Role Change Store and Business process. It then checks 1st authorisers instance in Role assigned Task Store, it approves the request. if all successful it will return true, otherwise false (see 6a-6i Fig 8(b)). For a 'new' variable it creates/check for an entry and in the role change store and the Role assigned task store for the statement assertion. The content is of a new variable populated and used by the conditional statements.

urn: uel:ac:uk:xacml:3.0:function:role-change-approve-check

This condition of this function is to carry out the BoD duties check, it obtains the attribute value of an XACML policy conditional statement (this value needs to be a unique ID). It uses this as an xPath reference to a role Policy. If a match is found, the role and its properties are matched against the subject ID in Role Assigned Task store, if all successful, it will return true, otherwise false (see 7a-7d Fig 8(b)). For a 'new' variable it creates/check for an entry in the Role assigned task store for the statement assertion. The content is of a new variable populated and used by the conditional statements.

F.CUSTOMISED FUNCTIONS

This section depicts Custom Functions developed for AW-TRBAC to enable Role Change capabilities within the AW-TRBAC model. There are two specific functions developed for service change (role change) requirement, which would require to adapt to dynamic environment based on the rules and policy enforcement, to mitigate escalation of privilege and information disclosure risk. To remain consistent with the story in the paper, Role change functions and policy will be described below.

FunctionId="urn:uel:ac:uk:xacml:3.0:function:role-change-check"

FunctionId="urn:uel:ac:uk:xacml:3.0:function:role-change-approve-check"

XACML Policies: This section describes the policies that were executed for the evaluation of the Service Change (role change) request and each policy has been designed to evaluate unique value within the XACML security request.

xacml-change-role-policy - This policy executed to ensure authorised user submitted the request and the user instance exits.

xacml-change-role-current-approve-policy – This policy provides appropriate governance within the current department, it verifies that the authoriser is within the current department and there is no conflict of interest by performing SoD check.xacml-change-role-new-approve-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2947377, IEEE Access

policy – This policy executed to provide authorisation for the provision of the new role

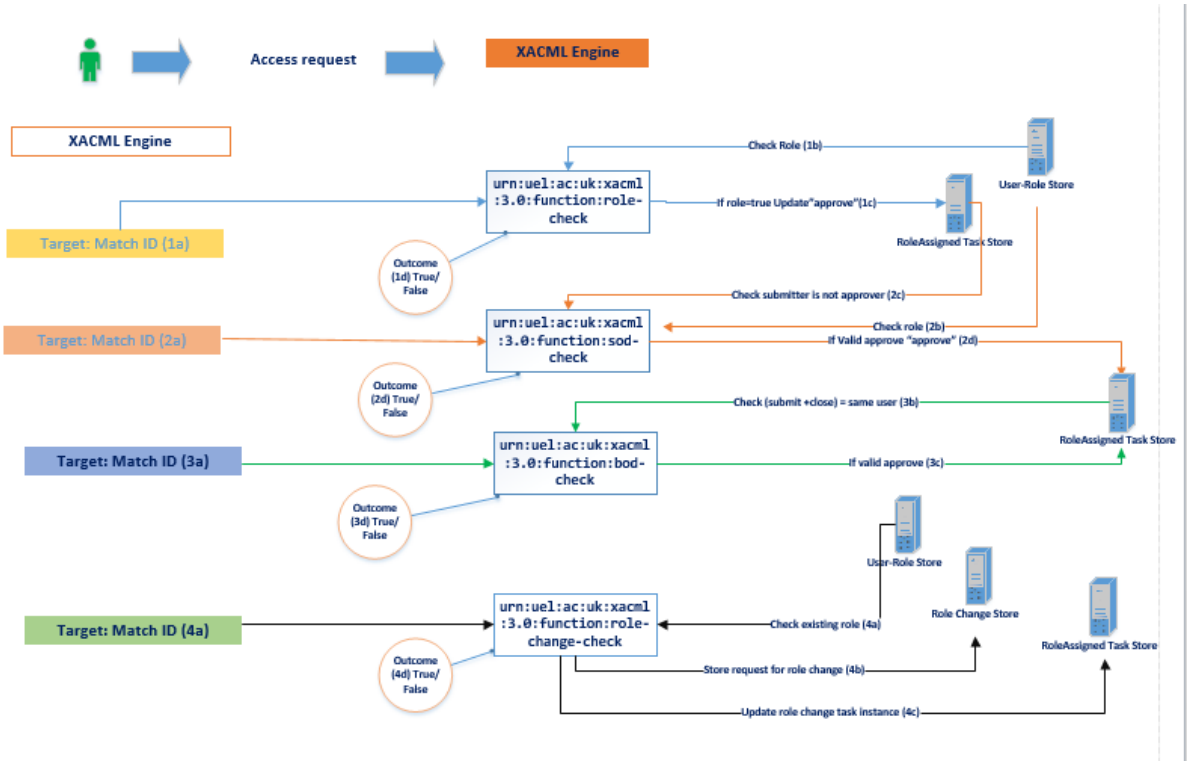

Figure: 8 (a) - **Extended XACML capability for SoD/BoD function of AW-TRBA**

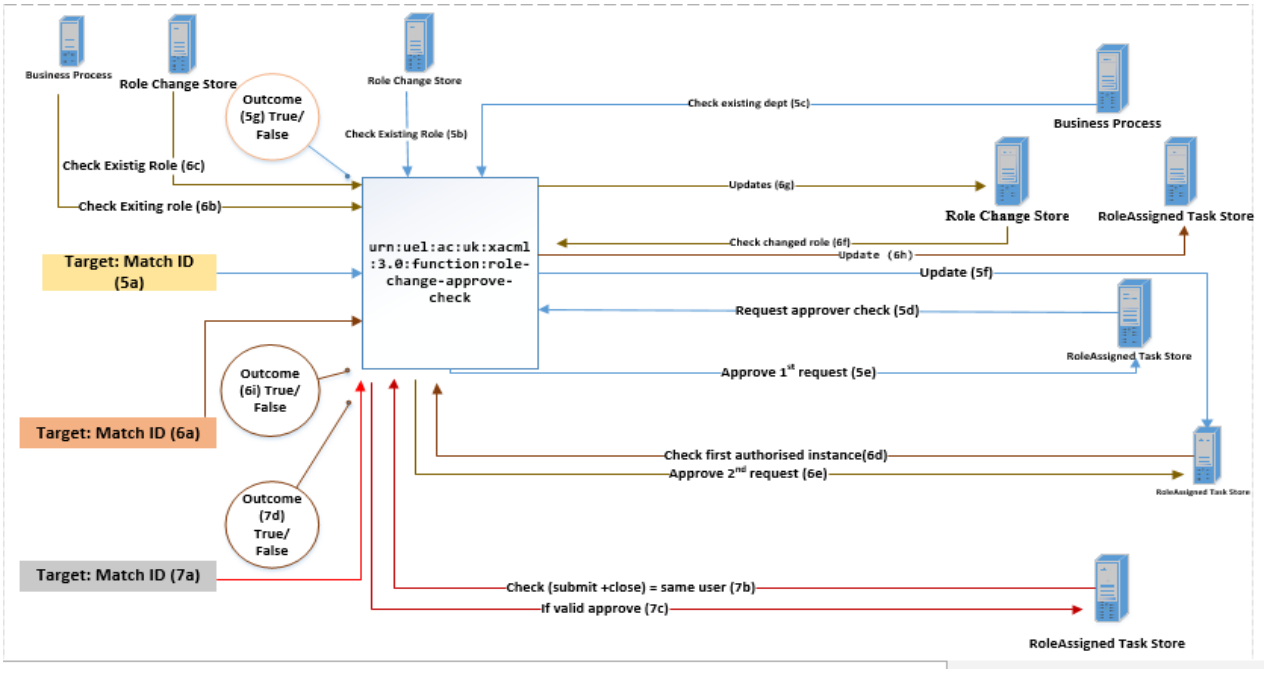

Figure: 8(b): **Extended XACML Policy Engine for AWTRBAC Model (role change)**

VI. EXPERIMENTAL SETUP

Proof of Concept (POC) has been experimented against the requirements satisfaction and applicability solution against the use case. Requirements were tested by simulating the six test cases to meet constraints and characteristics defined

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2947377, IEEE Access

in the Dynamic Access Control Model. Script simulates a user making a security request using REST URI which invokes the TR-BAC system implemented as microservice architecture, that performs the dynamic access control. Access request are converted to XACML equivalent which is then validated through XACML policy implemented using the open-source Balana engine. The engine in turn makes various data assertion on data store before allowing access to the resource (access Request system). There was a total of six scripts ran to test the requirements identified in section 4.1 and the execution of the constraints were recorded in a backend SQL database table

A. REQUIREMENTS SATISFACTION

To validate against the implemented solution has been conducted and associated outputs were recorded in Table 4. For Example UserId "Mat" has role permission "manager" which allows him to approve the security request submitted by UserID "Bob" who is the role "coordinator" , as shown in row one of the Table 4. This satisfies the requirement of the "Only authorised users access the resources" and "second row showing SoD through role and task instance" through restriction on task instance ID "Tif917803b" row one of the table, on which the Manager Role acted upon in row one, this is to ensure segregation of duties are performed at task instance level as well as role level, which satisfies the requirement of "Dynamic segregation of duties at the Instance level".

The action column in row two status changed to "approved", it records the role that performed the action on a task (security request) and resources (PC) that has been authorised by the role manager (Mat) on a task instance (Tif917803b) at a time event. This satisfies the requirement of "adequate event logging and access availability in real time ensuring governance", of access management process. Dynamic segregation of duties at instance level is also shown in row four, "tf317701a" an instanceId for a role change request, submitted by role "Coordinator", which shown in task column as "Role Change", that is approved by the role "manger" and new task instance recorded "change-rolecurrent-approve" and action status set to "approve, this proves that SoD enforced at task instance Id, Subject and role. Also shown in row six subjects "Duncan" who is the second level of approver for onboarding service manager approves the same task instance Id. "tf317701a", ensuring task contingency and sequence flow of tasks approval maintained and role change data store will be updated and existing role will no longer be active for the subject. This rule of enforcement in the policy allows revocation of existing entitlements and provision of new credentials.

Results in Table 4 depicts that Policy engine successfully, enforced the task constraints for SoD, BOD and Role Change, Instance level restriction, event logging, ensuring governance and mitigating broken access control risk through remediation of escalation of privilege vulnerability though instance level restriction and validation through the function in policy engine, meeting dynamic access control requirements.

TABLE 4: REQUIREMENT SATISFACTION EVENT LOGS.

taskInstanceRef	userId	roleId	taskId	resourceId	action	dateCreated
tif917803b	bob	coordinator	security-request	PC	approve	03/03/2018 22:11
tif917803b	mat	Manager	security-request-approve	PC	approved	03/03/2018 22:15
tif917803b	bob	coordinator	security-request-approve-close	PC	closed	11/03/2018 20:10
tif317701a	bob	coordinator	change-role		open	11/03/2018 20:12
tif317701a	mat	manager	change-role-current-approve		approve	11/03/2018 20:19
tif317701a	duncan	manager	change-role-new-approve		approve	11/03/2018 20:21
tif317701a	bob	coordinator	change-role-close		closed	11/03/2018 20:23

B.APPLICABILITY IN REAL-LIFE SOLUTION

To measure the system performance against the use case and a sustainable real life solution, this research experiment benchmarked against similar work carried out by (Ali & Moreau, 2013) whereby the author extended Balana engine to translate the provenance-based policy language into XACML request. To authors knowledge there are no existing experiments carried out and attempted to extend the Oasis standard for dynamic access control requirements.

The system was setup to measure performance of the policy enforcement by recording the cumulative time for end to end execution of a policy, this includes policy request, translation and execution in a policy engine. A total of one million execution recorded. Request was executed in a sequence ten thousand batch and each result (contain mean value with error bar at 95% confidence level) were recorded against the two hypotheses:

- System performance will not degrade with the dynamic access control request.
- Increased conditional statement with a role change will affect the processing time.

Benchmark Environment: The experiments used to evaluate the performance of the framework is based on Intel (R) Core (TM) i7-2820QM CPU @2.30 GHZ, with 6Gb of RAM and 600Gb of disk space.

Methods: The requirements from 4.1 of the use cases were tested for policy evaluation generated using REST API. Rest client would make a query to the server which executes the XACML policy and responses back to rest client. Experimental setup running 1m end-to-end runs in a batch of 10000 of a security request execution, then calculating the variance for simple security request against the complex role change request.

C *RESULTS*

As expected, results were consistent, it has taken an average of 0.12 (S) seconds for simple end to end security request to complete, with standard error of 0.024 (S) and confidence interval set at 95%. This indicates despites addition of additional overhead on system performance did not degrade. Whereas Role Change request mean value is 0.26 (S). Time taken to process the request with complex conditions is almost double. However, performance remains consistent during the stress test with standard deviation 0.039 (S) from mean remain constant with 0.039 (S) margin error and Confidence Interval within 95% is 0.076 (S), this indicates that if the system performance retested again at 95% confidence interval it will have 0.95 probability of containing the mean 0.26 (S) and 95% of the access request distribution is contained in the confidence interval.

VII. CONCLUSION

The research focused primarily to resolve the real-world problem, however the challenge was to produce a piece of research which would be sustainable in academia, resolve the industry problems and viable within the borderless security in the new technology era. This research was carried out to resolve industrial problem and provide sustainability for the dynamic emerging technology in a borderless environment through development/adoption of a dynamic access control model leveraging on XACML policy enforcement to improve overall risk posture of the firm. This research developed a dynamic access control model leveraging on Existing RBAC and ABAC access control model to provide the capability of task instance segregation coupled with role level of segregation. Instance level restrictions were imposed upon tasks that are permissioned through role enablement for a user. Other aspect which this research focuses on is the IT workflow ensuring the audit trail of process owner approvals though sequence of task being followed and enforced through role, task, process and task instance. The research extended the OASIS standard, introducing five new functions and two repositories to enhance the functionality through further development of open source Balana Engine. Extension were enhancement/gaps within current Access control model to enable real time decision making in a dynamic borderless environment such as adoption of cloud and AI Machine Learning Technology.

The research also focused on mitigating the critical web application risk highlighted by OWASP standard, enhancement to the broken access control through policy/rule enforcement and dynamic access control model, incorporating Dynamic SoD and governance. Research also mitigates the insufficient logging and monitoring risk which through policy enforcement on data store through creation of task instance level with events and actions. This will be an enabler for cutting edge IT deployment through enhancement of risk posture. AW-TRBAC model framework was able to meet the requirements for borderless network perimeter access control that require dynamic and real time decision making for resources to authorised users. It was noted that simple security request has taken 0.12(s) to process, while the complex request such as change in service role with additional conditional statements and targets doubled in time 0.26(s), this in comparison to the benchmark experiment by (Ali & Moreau,2013) is commercially viable.

Limitations within the research to further develop the PoC into industrial scale solution to understand the complexity within the policy rules and performance, AW-TRBAC could be extended to support as a broker CASB (Cloud Access Security Broker) Acts as a gatekeeper, allowing the organization to extend the reach of their security policies beyond their own infrastructure. AW-TRBAC is effectively could be integrated with organisation Identity management solution to provide holistic view, visibility and governance.

REFERENCES:

1. Mufajjul Ali and Luc Moreau. A provenance-aware policy language (cprovl) and a data traceability model (cprov) for the cloud. In 2013 International Conference on Cloud and Green Computing, pages 479_486. IEEE, 2013.
2. Quang Do, Ben Martini1, Kim-Kwang Raymond Choo. The Role of the Adversary Model in Applied Security Research. Computers & Security, Vol 81, Pages 156-181, 2019
3. Prosunjit Biswas, Ravi Sandhu, and Ram Krishnan. Label-based access control: Anabac model with enumerated authorization policy. In Proceedings of the 2016 ACM International Workshop on Attribute Based Access Control, pages 1_12. 2016.
4. Tim Bray, Dave Hollander, and Andrew Layman. Namespaces in xml. WWW Consortium REC- xml-names-19990114. http://www. w3.org/TR/1999/REC-xml-names-19990114, 1999.
5. Liang Chen, Luca Gasparini, and Timothy J Norman. Xacml and risk-aware access control. Resource, 2(10):3_5, 2013.
6. Jason Crampton. On the satisfiability of constraints in workflow systems, 2004.

7. Bassam Farroha and Deborah Farroha. Challenges of _operationalizing_ dynamic system access control: Transitioning from abac to radac. In 2012 IEEE International Systems Conference SysCon 2012, pages 1_7. IEEE, 2012.
8. David F Ferraiolo, Serban I Gavrila, and Wayne Jansen. Policy machine: features, architecture, and specification. Technical report, 2015.
9. Joseph Fialli and Sekhar Vajjhala. The java architecture for xml binding (jaxb). JSR Specification, Jan, 2003.
10. M Gallaher, Alan O'Connor, Brian Kropp, and G Tassey. The economic impact of role- based access control. Technical report,(NIST), 2002.
11. James Gosling, Bill Joy, Guy Steele, and Gilad Bracha. The Java language specification. Addison-Wesley Professional, 2000.
12. Garet M Cogdell, Adam Schnitzer, Kenneth Sandlin, Robert Miller, Karen Scarfone, et al. Guide to attribute based access control (abac) definition and considerations (draft). NIST 800(162), 2013.
13. Maria Leitner, Stefanie Rinderle-Ma, and Jurgen Mangler. Aw-rbac: access control in adaptive workflow systems. In 2011 Sixth International Conference on Availability, Reliability and Security, pages 27_34. IEEE, 2011.
14. Hongjun Li. Restful web service frameworks in java. In 2011 IEEE International Con-ference on Signal Processing, Communications and Computing (ICSPCC), pages 1_4.IEEE, 2011.
15. Gang Ma, Kehe Wu, Tong Zhang, and Wei Li. A exible policy-based access control model for work_ow management systems. In 2011 IEEE International Conference onComputer Science and Automation Engineering, volume 2, pages 533_537. IEEE, 2011.
16. O'Connor and R. Loomis. Economic analysis of role-based access control. TR, RTI International, 2010.
17. Top OWASP. Top 10-2017 the ten most critical web application security risks.
18. Cesare Pautasso. Restful web service composition with bpel for rest. Data & Knowledge Engineering, 68(9):851_866, 2009.
19. HF Ravi Sandhu, E Coyne, and Charles Youman. Role-based access control models. IEEE Comput, 29(2):38_47, 1996.
20. Erik Rissanen. Oasis extensible access control markup language (xacml) version 3.0. OASIS committee specification, 1.
21. Edelberto Franco Silva, Natalia Fernandes Castro, and Debora C Muchaluat Saade. Across-_: Attribute-based access control with distributed policies for future internet testbeds. ICN 2015, page 210, 2015.
22. Mark Strembeck and Jan Mendling. Modeling process-related rbac models with extended uml activity models. Information and Software Technology, 53(5):456_483, 2011.
23. Jacques Wainer, Paulo Barthelmess, and Akhil Kumar. W-rbac a workflow security model incorporating controlled overriding of constraints. International Journal of Co- operative Information Systems, 12(04):455_485, 2003.
24. Peng Wang and Lingyun Jiang. Task-role-based access control model in smart health- Care system. In MATEC Web of Conferences, volume 22, page 01011. EDP Sciences,2015.
25. Neha Agrawal, and Shashikala Tapaswi. A trustworthy agent-based encrypted access control method for mobile cloud computing environment. Pervasive and Mobile Computing, 52, 13-28,2019
26. AmirHosein Adavoudi-Jolfaei, Maede Ashouri-Talouki, Seyed Farhad Aghili, Lightweight and anonymous three-factor authentication and access control scheme for real-time applications in wireless sensor networks, Peer-to-Peer Networking and Applications , Vol 12(1),2019, Springer

Mumina Uddin is currently working at the PWC UK (PricewaterhouseCoopers) as an Information Risk Manager. She has been awarded M.Sc. in Information Systems from Brunel University and B.Sc. in Biomedical Sciences from Kingston University. Mumina is certified Auditor and awarded with CISA certification. She has extensive knowledge in Information security, Risk Management, AI, threat management, Robotic Process automation (RPA) and Identity Access Management, she has over 10 years' experience working in Financial Institutions and big 4 professional services in London. Ms Uddin interest in risk management in borderless network, serverless deployment and Blockchain technology.

Shareeful Islam Dr. Shareeful Islam is currently working at the School of Architecture, Computing and Engineering (ACE), University of East London, UK. He was awarded his PhD from Technische Universität München, Germany and M.Sc. in Information Communication System Security from the KTH, Sweden and M.Sc. in CS and B.Sc. (Hons) in APE from the University of Dhaka, Bangladesh. He is a Fellow of the British Higher Education Academy (HEA) and has published more than 60 referred papers in high-quality journals and international conferences. He participated in EU, industry, KTP projects. His research interests and fields of expertise are cyber security, risk management, requirements engineering, security, privacy, and cloud computing.