UNIVERSITY OF EAST LONDON

A Dynamic Access Control Model Using Authorising Workflow and Task Role-based Access Control

by

Mumina Uddin

A thesis submitted in partial fulfilment for the
Professional Doctorate in Security

Supervisor: Dr Ameer Al-Nemrat, Dr Shareeful Islam

in the
School of Architecture, Computing and Engineering
Department of Computer Science

July 2020

UNIVERSITY OF EAST LONDON

ABSTRACT

School of Architecture, Computing and Engineering
Department of Computer Science

Professional Doctorate in Security

by Mumina Uddin

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 across various networks. These networks face various challenges including increase operational burden and 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, whilst not being allowed access in other contexts. The current access control models are static and lack Dynamic Segregation of Duties (SoD), Task instance level of Segregation, and decision making in real time. This thesis addresses these limitations describes tools to support access management in borderless network environments with dynamic SoD capability and real time access control decision making and policy enforcement. This thesis makes three contributions: i) Defining an Authorising Workflow Task Role Based Access Control (AW-TRBAC) using existing task and workflow concepts. This new workflow integrates dynamic SoD, whilst considering task instance restriction to ensure overall access governance and accountability. It enhances existing access control models such as Role Based Access Control (RBAC) by dynamically granting users access rights and providing access governance. ii) Extension of the OASIS standard of XACML policy language to support dynamic access control requirements and enforce access control rules for real time decision making. This mitigates risks relating to access control, such as escalation of privilege in broken access control, and insufficient logging and monitoring. iii) The AW-TRBAC model is implemented by extending the open source XACML (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, can process relatively large numbers of complex requests, and meets the requirements of real time access control decision making, governance and mitigating broken access control risk.

ACRONYMS

ABAC - Attribute Based Access Control

ADMINISTRATOR - Actions a security request

API - Application Programming Interface

AW-TRBAC - Authorising Workflow Task Role Based Access Control

BoD - Binding of Duties

BUSINESS PROCESS - This 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 organisational goal

BYOD - Bring Your Own Device

CatBAC - Category Based Access Control

CISO - Chief Information Security Officer

CH - Context Handler

COMPLIANCE - Compliance means conforming to a rule, such as a specification, policy, standard or law

COT - circle of trust

CAGR - Compound annual growth rate

CSP - Cloud Service Provider

DAC- Discretionary Access Control

DATA SECURITY - Data security means protecting digital data, such as those in a database, from destructive forces and from the unwanted actions of unauthorized users, such as a cyberattack or a data breach

DoD - Department of Defence

DREAD - Threat model (Damage, Reproducibility, Exploitability, Affected Users, Discoverability)

EXECUTION LIST - Is a record of all users who performed certain tasks, this will contain names, roles and tasks that have been performed by a user

FIM - Federated Identity Management

FPE - Format Preserving Encryption

GDPR - General Data Protection Regulation

HTTP - Hypertext Transfer Protocol is an application protocol

IAM - Identity and Access Management

IDM - Identity Management

IDENTITY OWNER - Individuals, organisations or any other entity whose identity information is to be used for authentication purposes

IDMaaS - Identity Management-as-a-Service

IdP - Identity Providers

IAAS - Infrastructure as a Service

IMS - Identity Management Systems

IR - Instance Level Restriction

JSON - JavaScript Object Notation

JVM - Java Virtual Machine

LDAP - Lightweight Directory Access Protocol

MAC - Mandatory Access Control

NIST - National Institute of Standards and Technology

OASIS - Organization for the Advancement of Structured Information Standards

OAuth - Open Authorisation

OpenID - OpenID is an open standard and decentralized authentication protocol

OUI - Open Un-federated Identity

OWASP - Web Application Security Project

PAP - Policy Administration Point

PAAS - Platform as a Service

PBAC - Policy Based Access Control

PBWF - A workflow is an automation of business processes in whole or part, during which information or task is passed between participating in activities or action

PDP - Policy Decision Point

PEP - Policy Information Point

PIP - Policy Information Point

PoC - Proof of Concept

PRISMA- Preferred Reporting Items for Systematic reviews and Meta-Analysis

PROVISIONING - Refers to granting, managing access to an identity with supporting confidentiality, integrity and availability

RAdAC - Risk Adaptive Access Control

RBAC - Role Based Access Control

RESOURCES - Information resources contain business information and support the execution of tasks within workflow resources

REST API - Representational State Transfer

RP - Relay Party

SABSA - Sherwood Applied Business Security Architecture is a framework and methodology for enterprise security architecture and service management

SAM - Security Account Manager

SAML - Security Assertion Markup Language

SAAS - Software as a Service

SCIM - Simple Cloud Identity Management

SECURITY COORDINATOR - Person who submits the security request

SLA - Service Level Agreement

SOAP - Simple Object Access Protocol

SoD - Segregation of Duties

SP - Service Provider

SPML - Service Provisioning Markup Language

SQL - Structured Query Language

SSO - Single Sign-On

STRIDE - Threat Model (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege)

TA - Task Authority

TAL - Trust of Anchor List

TASK - The concept of a task is a fundamental unit of business work or business activity. ‘Job function’ is another expression of task

TP - Task Permission

URI - Uniform Resource Identifier

VM - Virtual Machine

WORKFLOW - This is an IT term describing a business process. In general, it means a product or method for supporting business processes in the enterprise environment

WAYF - Where are you from

XACML - eXtensible Access Control Markup Language

XML - eXtensible Markup Language

Contents

Acknowledgements xv

1 INTRODUCTION 1
1.1 Identity and Access Management (IAM) 1
1.2 Research Problem 3
1.3 Aim & Objectives 3
1.4 Knowledge Contributions 4
1.5 Presentation Overview 5

2 LITERATURE REVIEW 6
2.1 What is Identity Management? 6
2.2 Types of Identity Management Systems 7
2.2.1 Federated Identity Management 8
2.3 Popular Identity Management Models 10
2.3.1 Security Assertion Markup Language (SAML)-based IMS 10
2.3.2 OAuth 11
2.3.3 OpenID 13
2.4 Cloud Based Identity Management 13
2.4.1 Identity Management-as-a-Service (IDMaaS) 15
2.4.2 Smart Applications on Virtual Infrastructure (SAVI) 17
2.5 Identity Provisioning System 17
2.5.1 Heterogeneous Resources-oriented Unified Identity Provisioning Model (HR-OUIPM) 18
2.5.1.1 Intercloud Architecture Framework (ICAF) 21
2.6 Identity and Access Management (IAM) 22
2.7 IAM Functional Taxonomy 23
2.7.1 Data Security 24
2.7.2 Provisioning 24
2.7.3 Compliance 25
2.7.4 Identity and Access Management Issues 25
2.8 Existing Access Control Model 27
2.8.1 Discretionary Access Control (DAC) 27
2.8.2 Mandatory Access Control (MAC) 28
2.8.3 eXtensible Access Control Markup Language (XACML) 28
2.8.4 Risk Adaptive Access Control (RAdAC) 29
2.8.5 Role Based Access Control (RBAC) 29
2.8.6 Attribute Based Access Control (ABAC) 30

2.8.7 Policy Based Access Control (PBAC) 30
2.8.8 Workflow Access Control Model 31
2.9 Access Control Adversaries 32
2.10 Limitation of Existing Access Control Models 33
2.11 An Industrial Use Case and Limitations 34
2.11.1 An Investment Bank Use Case 35
2.11.2 System Context and Existing Business Process 35
2.11.3 Current Access Control Model and Workflow 35
2.11.4 Current Access Provisioning 36
2.11.5 How are Access Request Handled Currently? 37
2.11.5.1 Current Access Authorisation Workflow 37
2.11.5.2 How is Access Administration Actioned? 38
2.12 Access Control Limitation Within the Industrial Case Study 38
2.13 Summary 40

3 METHODOLOGY 42
3.1 Research Methodology 42
3.2 Research Design 42
3.2.1 Inclusion Criteria 44
3.2.2 Study Selection 45
3.2.3 Data Collection Process 45
3.3 Research Phases 46
3.3.1 Phase 1 Initial Use Case - Identifying the Challenges 46
3.3.2 Phase 2 Review of Literature - Identifying the Research Gaps and Limitations 47
3.3.3 Phase 3 Proposed Solution - The development of the Dynamic Access Control Model 47
3.3.4 Phase 4 Evaluation –Usefulness of the Proposed Model 48
3.3.5 Phase 5 Research Conclusion 48
3.3.6 Limitations 49
3.4 Summary 49

4 AUTHORISING WORKFLOW-TASK ROLE BASED ACCESS CONTROL (AW-TRBAC) 51
4.1 Motivation for Dynamic Access Control Model 51
4.1.1 Requirements for Dynamic Access Control Model 52
4.2 Conceptual View of the Dynamic Access Control Model 53
4.2.1 Conceptual View of the Dynamic Access Control Model 53
4.2.1.1 AW-TRBAC Concepts 53
4.3 AW-TRBAC Conceptual View 54
4.4 Mathematical Model of Task Entailment Constraints 56
4.4.1 Task Constraints 56
4.4.2 Task Entailment Constraints Within the Workflow 59
4.4.3 Analysis of Use Case Requirement Constraints 60
4.5 Policy Specification Requirements 63
4.5.1 Access Control Policy Enforcement 63
4.5.2 Extension of XACML Standard 65

4.5.3 Structure of XACML Policy Request 67
4.6 Comparison with Another Workflow Model 68
4.7 Summary 68

5 DESIGN AND IMPLEMENTATION OF AW-TRBAC MODEL 70
5.1 System Architecture 70
5.2 System Design 72
5.2.1 Access Portal API 72
5.2.1.1 Role Change URI 72
5.2.2 First Manager Approval REST URI 73
5.3 Dynamic Role Change 75
5.3.1 Second Manager Approval REST URI 75
5.3.2 Close Request REST URI 76
5.4 System Implementation 77
5.4.1 Task Workflow Support for XACML 77
5.4.2 Coupling of Policy Assertions 77
5.4.3 Function Role Check 78
5.4.4 Static & Dynamic Segregation of Duties Check (SoD) 78
5.4.5 Static & Dynamic Binding of Duties (BoD) 79
5.4.6 Single to Multiple Mapping (Role Change) 80
5.4.7 Single to Multiple (Role Change Approver Check) 80
5.5 AW-TRBAC Model Integration 81
5.6 summary 82

6 EXPERIMENTAL SETUP AND RESULTS 83
6.1 Requirements Satisfaction 83
6.1.1 Test Case 84
6.1.2 Invoking a SOD Security Request Through REST API 84
6.1.3 Policy for SoD Assertion in AW-TRBAC Engine 85
6.1.4 SoD policy to Match 85
6.1.5 XACML Request Generated for a Request Invoked Through API . 85
6.1.6 Response for the Invoked Request Through REST API 87
6.1.7 Binding of Duties (BoD) 88
6.2 Requirement Validation 88
6.3 Applicability of the AW-TRBAC 89
6.3.1 Hypothesis 90
6.4 Summary 93

7 CONCLUSIONS 94
7.1 Contributions of this Thesis 95
7.1.1 First Contribution 95
7.1.2 Second Contribution 95
7.1.3 Third Contribution 96
7.2 Future Research 97

A Appendix 98
A.1 Case Study 113
A.1.1 Introduction and Justification 113

A.1.2 Aim . . . 113
A.1.3 Methodology . . . 113
A.1.4 Choosing the process . . . 114
A.1.5 Key Stakeholder . . . 114
A.1.6 Workshop . . . 114

Bibliography **117**

List of Figures

2.1 Comparison between network-based and claim-based identity management system architecture, signifies through solid lines (Pfitzmann and Borcea-Pfitzmann (2009)) . . . 7

2.2 Federation through shared identity across domains through circle of trust (COT) (Maler and Reed (2008)) . . . 9

2.3 Open un-federated identity model where service providers are linked and interact through run time identity provider protocols; this allows a user to utilise any supported identity protocol (Alrodhan and Mitchell (2007)) 10

2.4 SAML protocol flow of request and sequence of responses returned by each entities (IdP and SP) (Kankaala et al. (2015)) . . . 11

2.5 OAUTH protocol flow describes the negotiation between the four roles;Client, Resource Owner, Resource Server and Authorisation Server (Hammer-Lahav (2010)) . . . 13

2.6 Identity provider is the cloud service to which user authenticates and service provider consumes the service from the identity provider (Mpofu and Van Staden (2014)) . . . 16

2.7 IDMaaS is a user-centric identity management system to manage user information and operations to CSP's through API provided by CSP (Liu et al. (2015)) . . . 16

2.8 Smart Applications on Virtual Infrastructure(SAVI)architecture with centralised identity manager providing Authentication through middleware on the identity provider and authorisation on the resource side (Faraji et al. (2014)) . . . 17

2.9 HR-OUIPM provisioning model concept of unified identity through mapping resource identities to unified identity information (which contains all attributes of that resource) using three mechanisms: a Request Parser, SPML engine and an Adapter (Liu et al. (2013)) . . . 19

2.10 Internal enterprise framework incorporated business requirements and policy for identity management infrastructure (White et al. (2007)) . . . 21

2.11 IAA Identity access management framework using SABSA methodology to map business processes at business owner level to highlight potential impact on business (Damon and Coetzee (2013)) . . . 22

2.12 Comparison of idm on strategy & vision with respect to vision and depth of solution scored out of 10 scale (Kumar and Rodrigues (2010)) . . . 23

2.13 Comparison on Idm's features and capabilities factors scored out of 10 scale (Kumar and Rodrigues (2010)) . . . 24

2.14 Number of articles in information security domain identified through systematic literature review of identity and access management . . . 26

2.15 Discretionary Access Control Model, based on the discretion of the employee functions (Lu and Jiang (2006)) 27
2.16 Mandatory Access Control Model, based on access to objects on the sensitivity of the resources (Tassey et al. (2002)) 28
2.17 Role based access control (rbac) role architecture, showing roles connected to resources and subject to prevent unauthorised disclosure of resource (Sandhu et al. (2000)) 30
2.18 Policy based workflow management model, based on polices on business processes depicting needs for active access control (Ma et al. (2011)) . . . 32
2.19 Current Identity and Access Management life cycle from the inception of a security request, management approval through to action of the request by security administrator 36
2.20 Existing access control models and it's limitations 40

3.1 Summary of Research Design 43
3.2 Data Analysis approach through systematic review method 44
3.3 Research conducted in six Phases 46

4.1 AW-TRBAC Dynamic Access Control Models shaded in green represent the notion of task to support workflow active access control for dynamic access control behaviour 54
4.2 Static structure diagram which describes the structure of the AW-TRBAC system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects 56
4.3 XACML Architecture and its interactions with various components PEP, PDP, PIP and PAP for a request evaluation OWASP (2017) 64
4.4 Showing Extended XACML Standard, shaded in blue components developed to meet AW-TRBAC dynamic access control behaviours 65
4.5 A sequence of events of task execution and authorisation are required to be executed in an orderly manner and a just in time decision need to be made based on the evaluation of various components of XACML in AW-TRBAC access control 66
4.6 Comparison of AW-TRBAC with other authorisation models based on gaps identified in the development of the access control model 68

5.1 Shows the AW-TRBAC architecture, which extends the five core components of the XACML architecture to support Instance restriction for enforcement of use case logic 71
5.2 A Rest HTTP Request/Response for a role change 74
5.3 Implementation of the function in AW-TRBAC engine as part of the extended XACML Standard to support dynamic access control requirement for SoD and BoD. 78
5.4 Implementation of the function in AW-TRBAC engine as part of the extended XACML Standard to support dynamic access control requirement for role change 79
5.5 AW-TRBAC portal Dynamic Access Control Model Integration, demonstrates the interactions between the system components and the service to allow authorised access to resources 81

6.1 Invoking a SOD Security Request Through REST API 84

6.2 SoD policy request handled by the AW-TRBAC engine 85
6.3 Policy for SoD assertion in AW-TRBAC Engine 86
6.4 XACML Request Generated for a Request Invoked Through API 87
6.5 Response for the Invoked Request Through REST API 88
6.6 Audit Logs of the Tests Based on Requirement Satisfaction Experiment . 89
6.7 Security Request Executed via API Client Call to AW-TRBAC Engine . . 90
6.8 Security Request Result for the API client to AW-TRBAC Engine Performance Test 90
6.9 Comparison of System Performance Against Complex Role Change Request 91
6.10 Summary of Results for SoD and Role Change Request 92

A.1 Sequence diagram showing new user request workflow 116
A.2 Sequence diagram showing role transfer request workflow 116
A.3 Sequence diagram showing role termination request workflow 116

List of Tables

4.1 Set Theory Label 57

5.1 REST - Role Change Initial Request, the URI for the Role Change Request within the AW-TRBAC portal 73

5.2 REST- Role Change Initial Response extracted and used by the AW-TRBAC engine to validate the request from the value of UserId, RoleId, TaskId and ResourceId 73

5.3 REST - Role Change First Approval Request the URI for the Role Change Request within the AW-TRBAC portal 74

5.4 REST - Role Change First Approval Response is executed after successful role change initial request, AW-TRBAC engine validate the request from the value of UserId, RoleId, TaskId and ResourceId 75

5.5 REST - Role Change Second Approval Request, the URI for the Role Change Request within the AW-TRBAC portal 75

5.6 REST - Role Change Second Approval Response is executed after successful role change first approval, AW-TRBAC engine validate the request from the value of UserId, RoleId, TaskId and ResourceId 76

5.7 REST - Role Change Close Request, the URI for the Role Change close out Request within the AW-TRBAC portal 76

5.8 REST - Role Change Close Request after successful validation of second approver update on the system data store and responses with HTTP 200 . 77

Acknowledgements

I would like to thank the Almighty for his blessing bestowed upon me in pursuit of ProfDoc.

I would like to dedicate this ProfDoc to my Children aged 11 and 9 Xavier and Zayaan, they were solicitous to cultivate their mummy's ProfDoc, sacrifice and compromise made to fulfil the ProfDoc goals.

A massive thank you to my Supervisors Dr Ameer Al-Nemrat and Dr Shareeful Islam for their help and guidance throughout the course of ProfDoc, their determination to succeed, patience and perseverance through difficult times were the prominent drivers behind reaching my ProfDoc goals.

Nothing in this world is achieved without support of a good family, I am blessed with a pre-eminent mother, through her encouragement and support it has been possible to fulfil the goals in life.

I would like to thank my husband Arman for his support, inspiration and dedication toward my achievements in ProfDoc.

Lastly, I would like to thank my brother for his continuous support and reassurance to see light at the end of the tunnel. I would also like to thank my sister for her support in high and low times through the journey of the ProfDoc.

PUBLICATIONS

- Uddin, M., Islam, S. and Al-Nemrat, A., 2019. A Dynamic Access Control Model Using Authorising Workflow and Task-Role-Based Access Control. IEEE Access, 7, pp.166676-166689.

- Uddin, M. and Preston, D., 2015. Systematic review of identity access management in information security. Journal of Advances in Computer Networks, 3(2), pp.150-156.

- Uddin, M., Al-Nemrat, A., 2014. Is the Security Bubble within Investment Banking Sector about to Burst, Proceedings of the International Conference on Information security and Cyber forensics, Kuala Terengganu, Malaysia.

Chapter 1

INTRODUCTION

This chapter provides an overview of the problem this thesis will address, details of the use case (Investment Bank), and describes the contribution of this thesis to the state of the art of knowledge.

1.1 Identity and Access Management (IAM)

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 access, the mechanisms and processes by which the policy is enforced, and the model on which the policy and processes are based. Access control technology has evolved from research and development efforts supported by the Department of Defence (DoD) (Tassey et al. (2002)). There are two fundamental types of access control: Discretionary Access Control (DAC) and Mandatory Access Control (MAC). While initial research and applications addressed preventing unauthorised access to classified information, recent applications have applied these access control policies to commercial environments such as Banking, Healthcare and Retail (O'Connor and Loomis (2010)). Other research has considered the approaches to the access control model that are restricted based on the organisation role, that are defined access entitlements at a granular level (decentralised granular level of entitlements) such as Role Based Access Control (RBAC) (Rajpoot et al. (2015)), Attribute Based Access Control (ABAC) (Biswas et al. (2016)), eXtensible Access Control Markup Language (XACML) (Rissanen et al. (2013)), and Risk Adaptive Access Control (RAdAC) (Farroha and Farroha (2012)).

Organisations are now dynamically changing the access privileges of users or revoking existing privileges due to various business demands. There are many applications that are running from outsourced environments such as the cloud or from supply chain partners, which need to deal with access control dynamically compared to the traditional in-house application, as network accessed through various endpoint devices such as mobile devices, iPad, bring your own devices (BYOD) and dispersed geographical locations. According to Daniel Crowley, (Martin (2019)) head of research for IBM's X-Force Red "today, network access must be dynamic and fluid, supporting identity and application-based use cases, a sophisticated access control policy can be adapted dynamically to respond to evolving risk factors". Ted Wagner Chief Information Security Officer (CISO) at SAP national Security Service emphasised "In every data breach, access controls are among the first policies investigated, access controls are a key component. When not properly implemented or maintained, the result can be catastrophic".

Due to the borderless network (the technical architecture that allows organizations to connect anyone, anywhere, anytime, and on any device, securely, reliably, and seamlessly. It is the foundation for the network infrastructure, providing optimization, scale, and security to collaboration and visualisation), there has been a mixed approach to access management across sectors. Web-based remote access, coupled with application access distributed on various networks and hosted on the cloud, means that enterprises are faced with various challenges including administrative issues, data privacy, increased operation burden, monitoring issues, and regulatory compliance. For the organisation to sustain a competitive edge, internal and external users are accessing systems from all over the world and from a variety of devices. This means that the identities of these users and their associated access, rather than the network, are forming the new security boundary around the organization, this change in thinking highlights the importance of getting Identity and Access Management (IAM) right, both to facilitate the business and to stay ahead of audit, compliance and regulatory requirements. According to data breach report 2019 (Rafter (2019)) 3800 publicly disclosed breaches, 4.1 billion number of records exposed, and breaches have increased 54% in comparison to 2018. Amongst the breaches there has been a financial breach of Capital One (Rafter (2019)), which resulted in largest category of information accessed and cost the organisation between \$100-\$150 million, the hacking which was as a result of misconfigured security system access control.

Despite significant developments, existing access control models do not focus on granting access, enforcing dynamic Segregation of Duties (SoD) (Ma et al. (2011)), where a role have conflicts of interest, such as an approver and submitter, in other cases where approver becomes the submitter and through his role as an approver allowing approval of his own request, this require restriction at the task level to enable SoD dynamically and Binding of Duties (BoD), requires similar restriction at the role and task level however restriction is at the role and task being performed by the same user and access governance through workflow management (Crampton (2004)), where event log of sequence of tasks performed ensuring visibility of access to data.

1.2 Research Problem

Overly complex security policies of access control in dynamic environments lead to data breaches. According to (Verizon (2019)) Data Breach Investigations Report, 75% of breaches were tied to credential theft and ineffective identity and access management. Responses to the *Executive Perspectives on Top Risks for 2019* from 825 board members and executives of various industries (including Finance) (Beaumier (2019)), indicate that privacy and identity, and access management, are in the seventh position for top 10 risks in 2019. This is supported by Open Web Application Security Project (OWASP) based on data submission from 40+ firms that are specialised in application security and industry survey that was completed by 500 individuals (OWASP (2017)). This data spans vulnerabilities gathered from hundreds of organizations and over 100,000 real-world applications and APIs. Ten most critical Web application security risks selected and prioritized according to this prevalence data, in combination with consensus estimates of exploitability, detectability, and impact, which indicates that broken access control relating to escalation of privilege is in fifth and insufficient monitoring and logging is in ninth position, of the top 10 application risks.

This thesis focuses on an access control model for a global investment bank based in London (due to privacy, name will not be mentioned). It has assets over 1 trillion dollars and offices in 17 locations worldwide. It is a privately held financial institution and has been a thought leader and a solution provider for over 200 years. Expert in Corporate Banking, Merger & Acquisition advisory, Investment Management, and wealth management and investor services. Among the many challenges are balancing complex and ever-changing regulatory and compliance requirements with efforts to boost effectiveness and reduce costs through digital transformation such as cloud deployment, Robotics, Artificial intelligence and blockchain (Beaumier (2019)).

This thesis will develop a dynamic access control model for the industrial use case, with the capability to support a dynamic environment, whilst mitigating the risks associated with accessibility though a diverse range of devices.

1.3 Aim & Objectives

Aims:

The aim of the research is to develop a dynamic access control model to enable authorised access and prevent unauthorised access to information assets.

Objectives:

1. Analyse the access management model of an investment bank to identify access

control requirements and limitations.

2. Use systematic review methodology to identify gaps in the access control model literature.

3. Construct a conceptual access control model using the use case requirements.

4. Proof of Concept (PoC) implementation of the dynamic access control model by extending the open source XACML (Balana) engine.

5. Validation of the case study requirements against the PoC to test for requirement satisfaction and applicability against the case study

1.4 Knowledge Contributions

The contributions of this thesis are:

First: The proposed Authorising Workflow Task Role-based Access Control (AW-TRBAC) model. This approach builds on existing task and workflow concepts to develop a new identity and access management solution. This work uses dynamic segregation of duties and process workflow and considers the task instance restrictions for the role's resource restriction, access governance and logs (Audit compliance and forensic analytic). Therefore, this research enhances the existing access control models such as RBAC and ABAC by dynamically granting users access rights, to promote access governance and risk mitigation.

Second: 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 adds functionalities to enforce Segregation of Duties (SoD) at the task instance level, mitigating the risk of broken access control. (OWASP (2017)). Through the logging of instance task events, it enhances access governance to provide visibility of unmanaged data , such as data that are sensitive in nature, access granted to need to know basis and in unstructured form (unstructured data is information that either does not have a predefined data model or is not organized in a pre-defined manner: excel, pdf, Google drive).

Third: The AW-TRBAC model is implemented using the open source Balana policy engine (Chen et al. (2013)) to demonstrate its applicability to a use case of a financial institution. The test results show that AW-TRBAC model has minimal impact on overall system performance despite changing user access requests dynamically and mitigating the risk of escalation of privilege to prevent data disclosure.

1.5 Presentation Overview

Chapter One: Is an introduction to the area of research, research problem, aims and objectives and contributions to the knowledge.

Chapter Two: Is a literature review taking a broad view of identity management systems and identity and access management paradigms, which then narrows down to a detailed analysis of access control models and its limitations. Next, access control adversaries (threats) are discussed along with the top 10 risks associated with access control model, its limitations and their consequences. Finally, the industrial use case access control model is investigated for its limitations and gaps in sustainability within the emerging technology such as robots and Artificial Intelligence (AI).

Chapter Three: Describes the methodology used to conduct the research in this thesis, including discussion of design, tools, approaches, limitations and advantages.

Chapter Four: Depicts a proposed Authorising Workflow and Task Role-based Access Control model (AW-TRBAC) that enhances the existing access control model through application of RBAC and ABAC access control model concepts. It is designed to provide and enhance additional features and functionalities for sustainability and resilience in a dynamic environment. It also extends the Oasis standard of XACML to meet the policy enforcement requirements of the case study.

Chapter Five: Details the design and implementation, AW-TRBAC model, system architecture, implementation design and integration, Application Programming Interface (API's) and mapping between technologies to support the integration of AW-TRBAC model, and task based policy enforcement for information security.

Chapter Six: Details the experiments that have been set up to meet the objectives of the research. Results were recorded and discussed in the context of the requirements of, and applicability to, the industrial case study that this research is based on.

Chapter Seven: Concludes the thesis and suggests future work.

Chapter 2

LITERATURE REVIEW

This chapter presents a detailed analysis of existing identity systems and widely adopted identity management systems, identity models, provisioning systems, frameworks to learn what is known within the knowledge. It then narrows down to access control models and works that focused on identity and access control management, to unveil what is unknown. Several industrial experts survey and opinions have been studied along with the security breach reports related to the adversaries within access control. An approach which is common in medical research field, where volume of clinical data is in huge quantity and analysis required to narrow down to specific area. Similar approach which have been used to analyse gaps in the published research in this thesis, through a systematic literature survey of Identity and Access Management (IAM), as identity management is well researched area which has attracted wide range of academia and industrial attention for its importance in information security.

Finally, this chapter analyse the use case of an Investment Bank, its current access control model and limitations to identify the requirements to incorporate and compare against the literature review to identify the gaps in the research.

2.1 What is Identity Management?

Identity management consists of the processes and all underlying technologies for the creation, management, and usage of digital identities. A typical identity management system consists of: users (the end user or an agent acting on the user's behalf), identity providers (responsible for validating a user's authentication credentials and "assert for" the in a single sign-on) and service provider (an entity that is responsible for validating a user's authentication credentials and "assert for" the in a single sign-on). The user requests resource access from the service provider, which relies on the identity provider to authenticate information about the user. These three components alone cannot be held accountable, therefore a legal entity (an organisation or individual person), that is responsible and accountable for the activities performed on a system is required. The identity of this entity is the set of all service providers, characteristics that have been attributed to this entity (Joosten et al. (2008)).

2.2 Types of Identity Management Systems

There are many types of identity management systems (IMS) that exist, see Fig 2.1, broadly divided into network-based identity management and claim-based identity management, differing in architecture, which has an impact on the security, privacy and usability issues associated with them (Pfitzmann and Borcea-Pfitzmann (2009)).

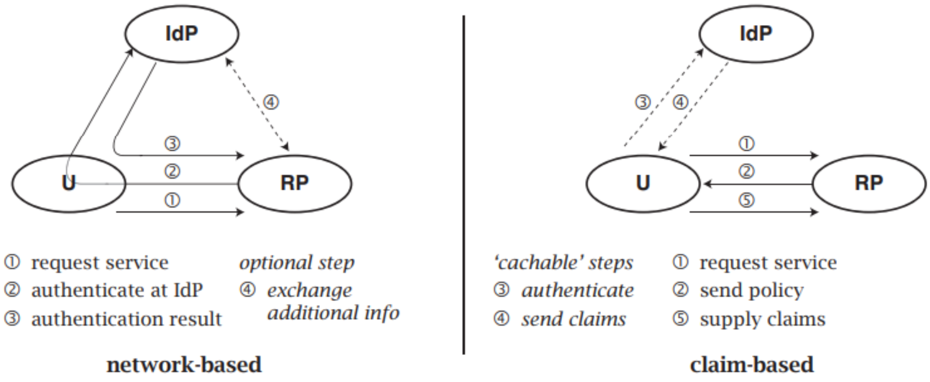


Figure 2.1: Comparison between network-based and claim-based identity management system architecture, signifies through solid lines (Pfitzmann and Borcea-Pfitzmann (2009))

The mechanisms behind Network-based Identity Management System is as follows: when a user requires access to a service, they are authenticated by the identity provider (IdP) and upon successful authentication, the user is given a token, that will then be forwarded to the service provider (SP), as shown in Fig 2.1. The service provider then verifies the token, and if valid, accepts the user as authenticated. To obtain further identifying information about the user, the service provider then contacts the identity provider directly, using the token as a pointer to the user profile stored by the identity provider. In some cases, the user arbitrates this exchange of information between the identity provider and service provider. Examples of network-based identity management systems are OpenID2 (Karim and Adnan (2019)), the Liberty Alliance3 (Fuchs et al. (2011)), and Shibboleth in (Cantor and Scavo (2005)). As there are wealth of identity information stored at the IdP, which makes the IdP a single point of failure and vulnerable to security theft, if a user successful in obtaining credential details to login to IdP.

Claim-based Identity Management System is where the service provider mandates the information it requires to grant access to the resources. The user is required to obtain a claim (attributes about the identity) after authenticating with an identity provider, attributes in this instance such as assertions by Security Assertion Markup Language (SAML 2.0) (Kankaala et al. (2015)), as shown in Fig 2.1, which is then forwarded to the service provider and the user is authorised to access the resources. A SAML assertion is a declaration about a user by an identity provider to a service provider.

The crucial difference between network-based and claim-based identity management systems is that, in the claim-based setup, there is no direct exchange of information between the identity provider and the service provider , giving the user more control over the exchange of authentication information. Even though there are exist policy tools such as uApprove5 (Alpár et al. (2011)) for network based IMS systems that allow a user to deny or give consent to releasing his attributes to the service provider, the actual attribute assertion exchange still takes place with the service provider and identity provider communicating with each other directly as shown by dotted line labelled 4 in the Fig 2.1. Examples of claim-based identity management systems are the Identity Metasystem (Windows CardSpace) (Ahn (2019)), and more privacy friendly concepts like Idemix (Camenisch et al. (2019)) and U-Prove (Paquin (2013)). In the latter two cases, claims are in fact anonymous, and are not transferred to the RP directly. Instead, the statement of the claim is proven to the RP in a zero-knowledge fashion. This ensures the user's privacy, because it retains the user's confidentiality in the two interactions with the service provider, removes the need for application to perform authentication task. The majority of identity management systems are network-based; however, claim-based approaches is a new concept which enhance privacy and have better security and usability as the user is in control.

2.2.1 Federated Identity Management

The concept of federated identity management (FIM) (Maler and Reed (2008)) is sometimes a cause for confusion. At times the term is used to describe the collaboration of several RPs to use a single IdP, all within the same domain. Such a setup is the standard form of identity management, where no real federation takes place. Instead, federated identity management is actually a setup where identity is shared across domains in Fig 2.2. Within such a federation, additional agreements can be made for further optimisation, e.g. to have a centralised authentication authority. The so-called circle of trust (CoT) equals the set of domains that belong to one federation; domains can also belong to several federations and therefore can belong to several circles of trust, as shown in Fig 2.2.

The identity provider issues relevant credentials to the users. To access a resource, the user would be authenticated by the identity provider, which will then redirect the user to the service provider to access the resource. Once a user is authenticated at the federal level, the user can then have authorised access to resources within the federated services; such mechanisms are employed by Single Sign-On (SSO) services. The Google service is

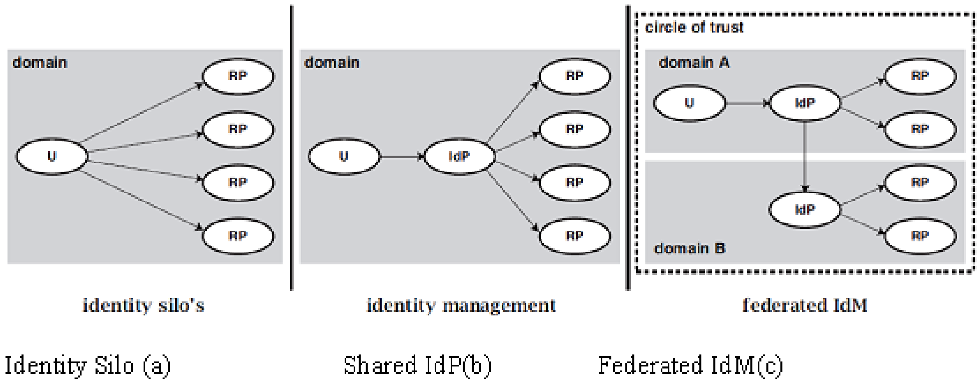
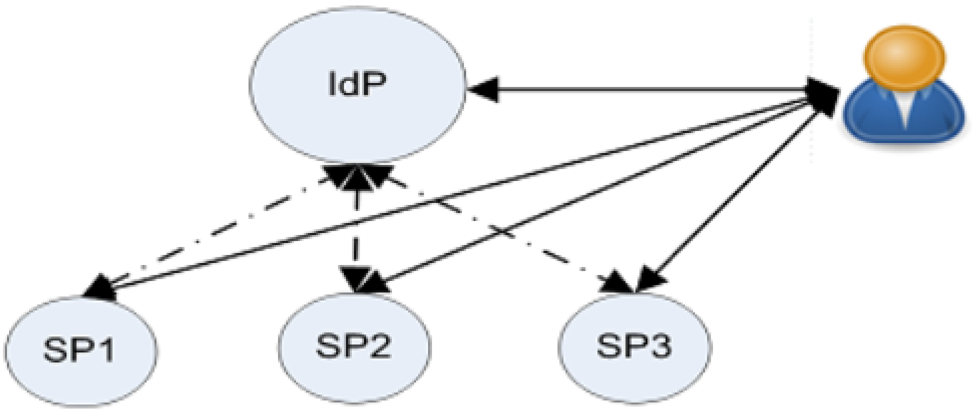


Figure 2.2: Federation through shared identity across domains through circle of trust (COT) (Maler and Reed (2008))

an example of a model consisting of one authenticator and many services such as Gmail, Google Drive, Calendar, Google Scholar, etc. Google acts as a gateway to allows a user to access all resources within Google services through a single sign-on mechanism whereby a user authenticates once, and this is federated through all Google services.

Although federated identity management solutions are widely employed in corporate and academic environments, many problems still arise. These systems can provide convenient user functions (such as single sign-on or automated form-filling), however, the single layer of authentication decreases system security, while it increases the value of user credentials (as it provides access to more resources). As the number of identity providers and service providers increases, FIM systems become difficult to manage and maintain.

Another model which has gained popularity in resolving this issue through employment of social networks is called the Open Un-federated Identity (OUI) model (Alrodhan and Mitchell (2007)). This model differs from federated models as the user is not restricted to a single identity provider, service providers are linked and interact through run time identity provider protocols; this allows a user to utilise any supported identity protocol. This model is used in many popular social networks such as Facebook, Twitter and LinkedIn which store a number of different user attributes and the user can authenticate themselves through their social network profile to access other services online. OpenID (Domenech et al. (2014)) and OAuth (Hardt (2012)) are the two most popular identity protocols for this model. Figure 2.3 illustrates the idea of an entity in the OUI Model. The solid line signifies that this model is not part of a federation thus this model is not reliant on any factor such as being part of CoT, however this model is not suitable where minimum trust is necessary between entities.



,

Figure 2.3: Open un-federated identity model where service providers are linked and interact through run time identity provider protocols; this allows a user to utilise any supported identity protocol (Alrodhan and Mitchell (2007))

2.3 Popular Identity Management Models

Several popular identity management systems have been widely used for identity and access management. In this section, three of the most popular, SAML-based systems (Kankaala et al. (2015)), OpenID (Recordon and Reed (2006)) and OAuth (Hardt (2012)), are discussed.

2.3.1 Security Assertion Markup Language (SAML)-based IMS

SAML identity management system utilises the federated identity model (Jøsang et al. (2005)) and the Security Assertion Markup Language (SAML) (Kankaala et al. (2015)). SAML is an XML (eXtensible Markup Language)-based standard for exchanging authenticated and authorised information between different applications (Kankaala et al. (2015)). It is based on the request/response protocol in which a service provider requests identity information about a user from an identity provider, which responds to the request with appropriate user attributes for authentication as shown in Fig 2.4. SAML consists of four key concepts: assertions, bindings, profiles and protocols. SAML assertions consist of statements: Authentication statements, Attribute statements and Authorisation decision statements. As SAML is widely used for SSO, commonly within web services, this allows ease of use, centralised credential management, and better governance and controls for authorised resources. There are a number of different libraries for building up a SAML-based identity management system such as Shibboleth (Cantor and Scavo (2005)) and SimpleSAMLphp (UNINETT) (Ferdous and Poet (2013)) developed using Java and PHP respectively.

Federation of service providers and identity providers using SAML is achieved using a Trust of Anchor List(TAL). This entails exchanging respective metadata of the identity provider and service provider and storing them appropriately. This ensures mutual trust between the two parties; it provides assurance to the service provider that the identity provider will authenticate the user using reliable security mechanisms and provides attributes to the service provider based on contractual agreement (Jøsang et al. (2005)). On other hand, trust is required from the service provider that confidentiality and integrity of the attributes disclosed will be retained and used for the intended purpose only.

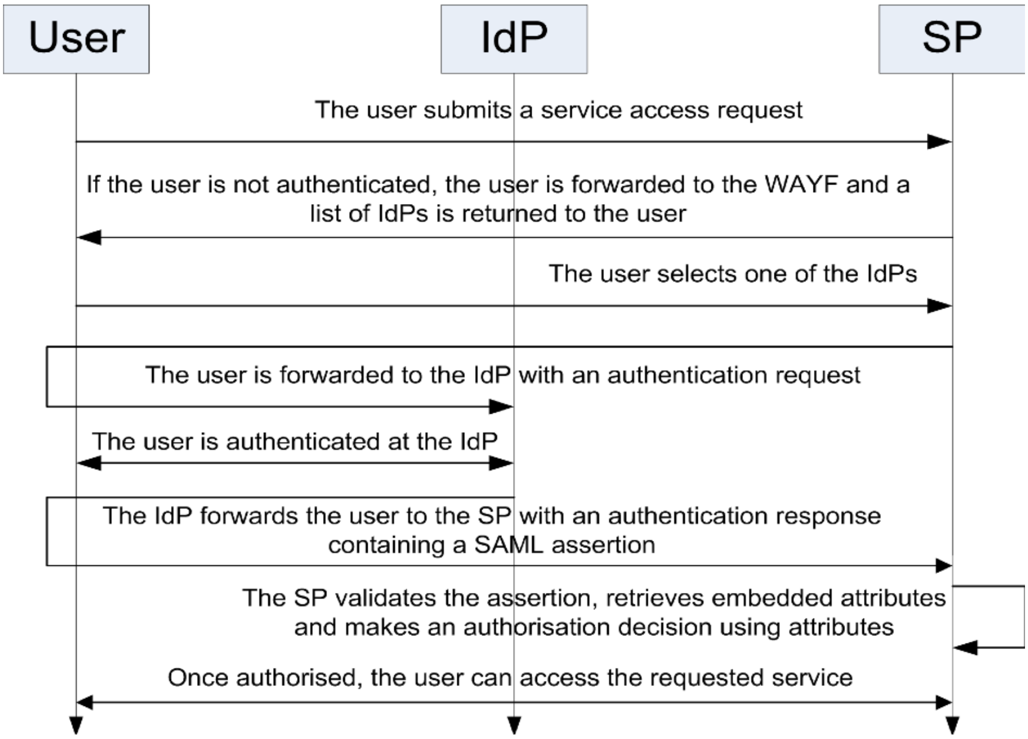

User
IdP
SP
The user submits a service access request
If the user is not authenticated, the user is forwarded to the WAYF and a list of IdPs is returned to the user
The user selects one of the IdPs
The user is forwarded to the IdP with an authentication request
The user is authenticated at the IdP
The IdP forwards the user to the SP with an authentication response containing a SAML assertion
The SP validates the assertion, retrieves embedded attributes and makes an authorisation decision using attributes
Once authorised, the user can access the requested service

Figure 2.4: SAML protocol flow of request and sequence of responses returned by each entities (IdP and SP) (Kankaala et al. (2015))

2.3.2 OAuth

OAuth is an authentication protocol which allows one application to interact with another application on behalf of a user without sharing credentials. This approach circumvents some of the limitations related to access delegation in a traditional method. As an example, let's consider the case of Joe and Mat; Joe would like Mat to post something on his Twitter stream. In the traditional method, Joe would have to share his username and password with Mat as a way of delegating responsibility. This would result in full compromise of the system without any control afforded to Joe, and revocation of such access rights is cumbersome.

OAuth 1.0 was superseded by OAuth 2.0 (Hardt (2012)), which provides a flexible solution to this problem, allowing any user to delegate their access right in a more manageable and secure way. The OAuth protocol comprises of four different classes of entities: User,Consumer,Authorisation Servers and Service Provider.

User. Someone who owns and controls the protected resources and are capable of granting (delegating) limited access rights to (consumer) third parties for accessing protected resources.

Consumer (Client). This is a third-party application that can make requests to access protected resources on behalf of a user. To make such a request, they must receive an authorisation clearance from the user.

Authorisation Servers. Authorisation servers are responsible for granting access tokens to consumers after receiving valid authorisation grants.

Service Provider. Resource servers host protected resources and can accept and respond to requests for access using access tokens. In many cases, resource and authorisation servers may be the same entity as the service provider. A number of "legs" is used to describe the number of entities involved in an OAuth interaction.

As shown below in Fig 2.5 describes the negotiation between the three/four roles and includes the following steps:

1. The Consumer/client requests authorization from the resource owner.
2. The User redirects the request to Service provider (authorisation server)
3. The Consumer requests an authorisation token from the service provider
4. The authorization server validates the Consumer credentials and if valid issues an access token
5. The Consumer requests the protected resource from the resource server and authenticates by presenting the token
6. The resource server validates the access token, and if valid, grant the request

(Noureddine and Bashroush (2011)) Introduce an optimization to OAuth 2.0, where the Authorization Server is provisioned with explicit authorization table to make decision at the Authorization Server prior request reaching the protected resource. This reduces the amount of processing time and alleviates the risk of potential threats such as Denial-of-Service (DoS) attacks and Distributed DoS (DDoS) attacks. Limitation of this model is not suited for multi-tenancy environment which require shared Authorisation Server to secure token for each tenant in a secure way.

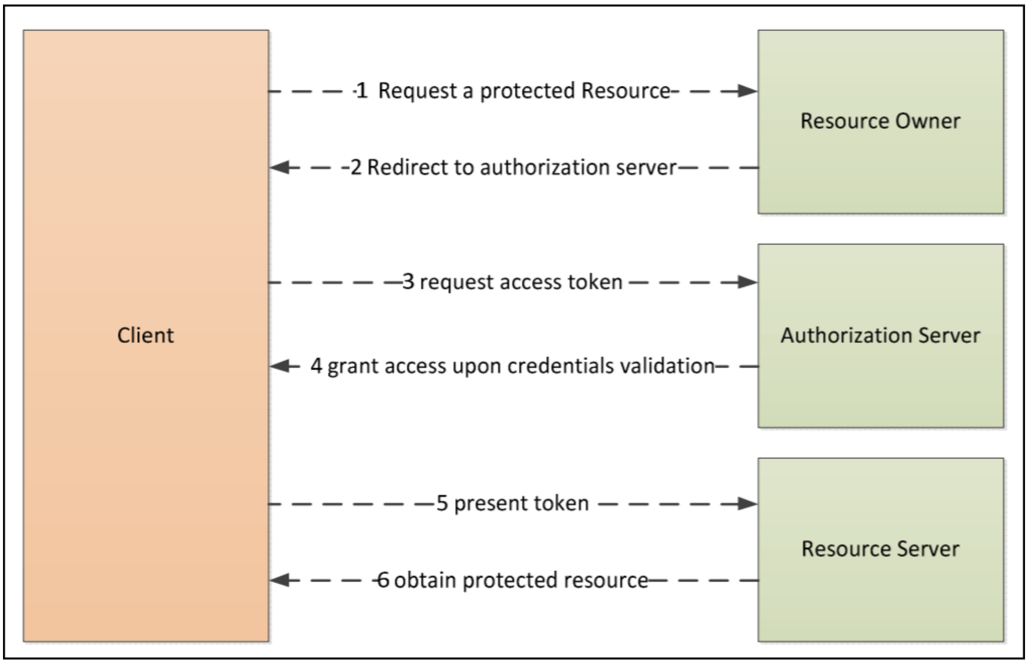

Client
1 Request a protected Resource
2 Redirect to authorization server
3 request access token
4 grant access upon credentials validation
5 present token
6 obtain protected resource
Resource Owner
Authorization Server
Resource Server

Figure 2.5: OAUTH protocol flow describes the negotiation between the four roles;Client, Resource Owner, Resource Server and Authorisation Server (Hammer-Lahav (2010))

2.3.3 OpenID

OpenID is a decentralised identity management system, based on an open unfederated identity model. It is a widely used identity management system, used by web service providers such as the BBC, Google, PayPal, Verisign and Yahoo (Domenech et al. (2014)). OpenID protocols are used as SSO and have three components: user, service provider and OpenID provider. A user would create an account with the OpenID provider, authenticate via the OpenID service provider to receive a token, that would then authorise the user to access resources within the service provider.

2.4 Cloud Based Identity Management

Cloud is built on existing technologies and tools, reducing the cost of service delivery whilst increasing the speed and agility of service deployment (Voas and Zhang (2009), (JoSEP et al. (2010)). The core technology behind cloud computing is virtualisation (Keith and Ole (2006)), (Uhlig et al. (2005)), which empowers the whole cloud computing paradigm. The virtualisation technology allows the separation of physical hardware and the operating system by creating an abstract layer between them. This allows a greater degree of extensibility by enabling sharing of physical resources virtually, by more than one OS. The National Institute of Standards and Technology (NIST) (Rittinghouse and Ransome (2017)) define cloud computing as the composition of five essential characteristics: three service models and four deployment models. There are currently three well defined service models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS).

Infrastructure-as-a-Service (IaaS) provides virtual server instances and storage on demand, service provider is responsible for physical and virtualization and the customer company is responsible for the operating system, application and data. Root accounts are all managed by the provider which creates risk for the customer (Tassey et al. (2002)). In the Platform-as-a-Service (PaaS) setup, software and product development tools are hosted, allowing developers to create an application on the provisioning platform. The provider is responsible for the security of the platform, but securing the application and data is the customer's responsibility. In the Software-as-a-Service (SaaS) setup, customer organisations access data through a portal, and the overall security responsibility falls to the provider organisation. As you move down the stack from SaaS to IaaS overall responsibility falls more on the customer organisation and less on the provider. Although with the SaaS service, overall responsibilities fall to the provider, the total risk is not diminished. This is the reason organisations need to exercise control over privileges either directly or via an enforceable obligation on the part of the provider.

According to the latest update to the International Data Corporation (IDC) (Murray (2019)), Worldwide Semi-annual Public Cloud Services Spending Guide. With a five-year compound annual growth rate (CAGR) of 22.3%, public cloud spending will grow from $229 billion in 2019 to nearly $500 billion in 2023. Cybersecurity Ventures predicts global spending on IAM products and services will exceed $16 billion USD annually by 2022 (Menlo Park (2017)). According to the Gartner Inc IAM spending forecast (Gartner (2019)), world-wide spending on information security products and services reached $114 billion in 2018, an increase of 12.4 percent from previous year and market is forecast to grow 8.7 percent to $124 billion in 2019.

A review of various cloud-based IAM (Identity Access Management) systems using various evaluation criteria concluded that cloud IAM needs improvement in terms of its features and functionalities (Habiba et al. (2013)). Although they address authentication, authorisation and access rights delegation, none of the currently existing cloud IAM systems fulfil the complete requirements auto-provisioning deprovisioning, SSO and entitlement reporting) for identity and access management. According to Kandukuri, five cloud security issues should be addressed in a Service Level Agreement for an organisation (SLA). They are: privileged user access, data location, data segregation, data disposal, and investigation and protective monitoring (Kandukuri et al. (2009)). Privileged user access ensures only authorized users have access to an organisation's data and resources. Therefore, identity and access management are considered security concerns in cloud computing. Various models have been proposed to address identity management in clouds, such as central IAM, trusted third party, federation solutions, etc. Most of the solutions are mainly focused on federation of cloud providers and pay little or no attention to access management.

A list of available technologies and solutions for cloud computing has been compiled (Fuchs et al. (2011)), including Primary and Identity Management for Europe (PRIME), Windows CardSpace, OpenID, Higgins, and Liberty Alliance. Current approaches to cloud IAM concentrate on offering solutions to issues such as federation or finer-grained access control. The lack of a comprehensive analysis, from conception to physical implementation, to incorporate these solutions, has resulted in impractical and fractured solutions. Simple Cloud Identity Management (SCIM) provides a defined standard Application Programming Interface (API) and user schemas which have been adopted by vendors of cloud providers (Lewis (2012)). Unrealistic optimism in information security by the IT managers needs to be resolved (Rhee et al. (2012)).

2.4.1 Identity Management-as-a-Service (IDMaaS)

IdMaaS is a cloud service as shown in Fig 2.6, where a third party assumes the identity management role on behalf of the identity owner (which is an organisation) leaving the organisation to devote almost their entire effort to the core business (Mpofu and Van Staden (2014)). A typical IdMaaS environment at an abstract level consists of the identity provider (also acts as the identity manger in the cloud), identity owner (individuals, organisations or any other entity whose identity information is to be used for authentication purposes) and the relying party (website or online services which consumes identity provider services to obtain security credentials for users). Identity provider is the cloud service to which user authenticates and service provider consumes the service from the identity provider.

1. Identity (ID) owner submits identity attributes for account creation or login details if they are existing users to the ID provider. The ID providers will do the authentication and transmits a package of authentication and authorisation details for the relying party.

2. The relying party will respond directly to the users with the relevant services. Subsequent requests will now be directed to the relying party once a user has been authenticated.

3. In case of account creation, the ID provider will create the account guided by the agreements they entered with the relying party

IDMaaS is a user-centric identity management system as shown in Fig 2.7. It provides a central node to storing users' profiles, files and friend lists (such as Alice and Bob register their details). It is not necessary any more for users to login in different Cloud Service Provider (CSP) to manage data. Users can login to IDMaaS to manage their data and update information or other operations, as shown in Fig 2.7 to CSPs through APIs provided by CSPs. User tokens used to connect to CSPs are stored in IDMaaS (Liu

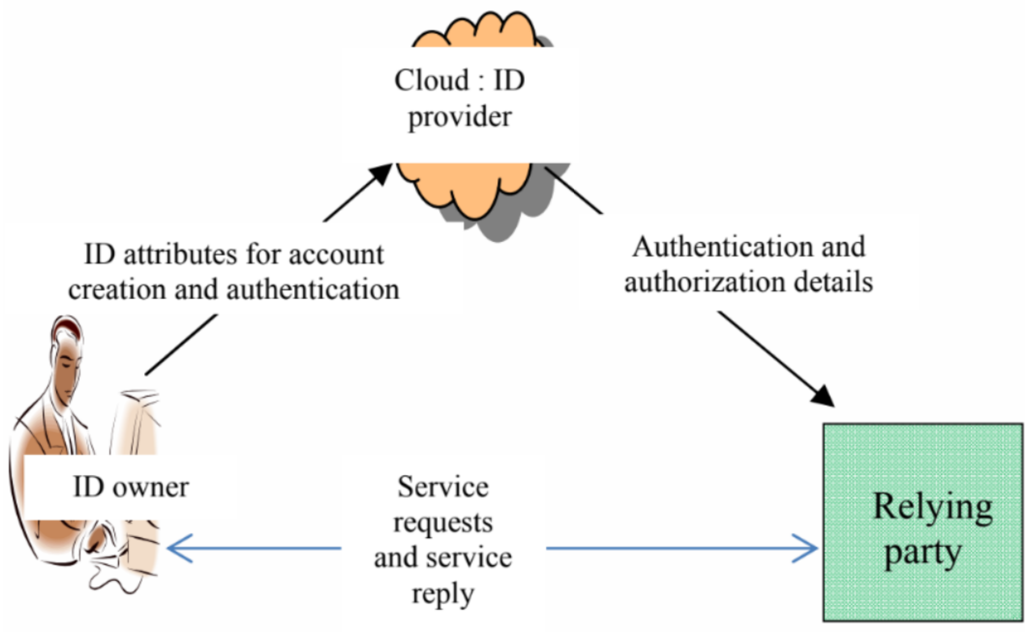

Cloud : ID provider
ID attributes for account creation and authentication
Authentication and authorization details
ID owner
Service requests and service reply
Relying party

Figure 2.6: Identity provider is the cloud service to which user authenticates and service provider consumes the service from the identity provider (Mpofu and Van Staden (2014))

et al. (2015)). IDMaaS contain three roles, User, IDM Service and CSP. Six modules are included in the IDM service. Three of them (myProfile, myFriend, and myFile) are data-related and other three modules (myAuth, myCloud and myAccess) implement identity authentication and access control management. Additionally, unified APIs are provided by the IDM Service for multiple CSPs, aimed at unified identity and data management.

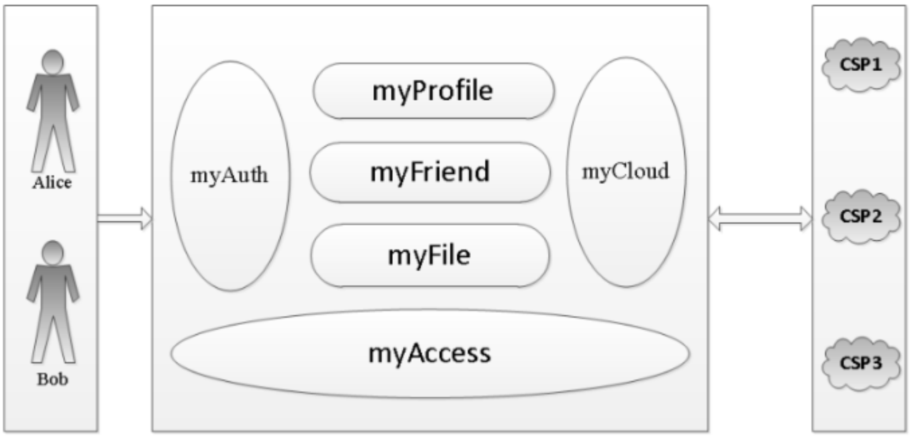


Figure 2.7: IDMaaS is a user-centric identity management system to manage user information and operations to CSP's through API provided by CSP (Liu et al. (2015))

An advantage of this model is a centralised identity provider (IdP), which supports universal communication protocols for multiple service providers' environments. It also contains FPE (format Preserving Encryption) which ensures (FPE) is a useful encryption method which encrypts a plaintext without altering its length or format, and hence the encrypted message can be updated into original data entries of databases or files without replacing the corresponding plaintext. The disadvantage of this model is that it is a new concept and it doesn't detail the mechanism by which the API will connect to the CSP.

2.4.2 Smart Applications on Virtual Infrastructure (SAVI)

Feraji (Faraji et al. (2014)) as shown in Fig 2.8, proposed another centralised IAM model which is based on the Smart Applications on Virtual Infrastructure (SAVI). It consists of a centralised identity provider with decentralised middleware to connect to the resource provider. The SAVI IAM is a central identity manager with distributed middleware based on an IdP/SP model and is comprised of 6 basic components: Manifesting Management, Identity Management, Policy Management, Token Management, Authentication Management, and Middleware. Middleware resides on the identity provider side and provides the authentication, and the middleware resides on the resources sides and provides authorisation.

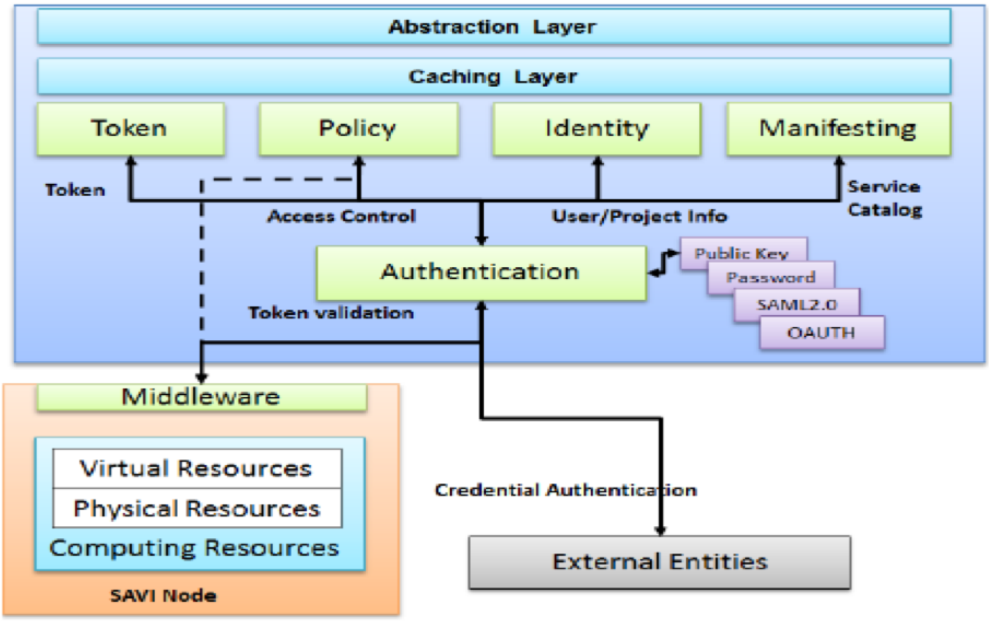


Figure 2.8: Smart Applications on Virtual Infrastructure(SAVI)architecture with centralised identity manager providing Authentication through middleware on the identity provider and authorisation on the resource side (Faraji et al. (2014))

The advantage of this model is, results of the testbed and evaluation indicates that that is it scalable and adaptable. The disadvantage of this model is that it doesn't support multiple service providers and mobile devices.

2.5 Identity Provisioning System

Identity provisioning is a software service used by enterprises to integrate and manage the process of providing users access to enterprise systems and business data. It is interrelated with security services, such as creating a user's account, resetting passwords, and synchronising all certifications among application systems. As a recently emerging technology, identity provisioning simplified the process of software installation and policy configuration. Thereby, identity provisioning has been of widespread concern in the industry (Sakimura et al. (2015)).

2.5.1 Heterogeneous Resources-oriented Unified Identity Provisioning Model (HR-OUIPM)

In order to solve the issue of identity provisioning for heterogeneous resources, identity would need to be assigned to resources directly. Integration of the provisioning process is constrained by differences in identity information formatting between resources. HR-OUIPM (Liu et al. (2013)) put forward the concept of unified identity through mapping resource identities to unified identities, so that users can use a single unified identity to complete provisioning operations. This resolves the issue of heterogeneous identities in the model. The HR-OUIPM model in Fig 2.9 is based on SPML and XML (Liu et al. (2013)). The identity provisioning request contains unified identity information and identity mapping will convert the information from the unified identity to the specific resource identity (which contains all attributes of that resource) using three mechanisms: a Request Parser, SPML engine and an Adapter.

A Request Parser transforms unified identity provisioning to standard XML documents according to request types and resources identities, based on SPML. Standard XML documents are encapsulated by SOAP and send to an SPML parser. The SPML parser tests the legality of the received request according to the XML schema of SPML. If the result is positive, the parser will divide and analyse this request message based on SPML semantics. After parsing, the system will store the information in a persistent directory service. All heterogeneous resources can obtain identity information by connecting to the directory service SPML Engine, the main use of which is to parse SPML requests, transforming them to API invocations. This model uses XML parsing using SAX-based SPML OXMap-ping, as it is memory saving and it also offers random access ability.

The adapter is responsible for integrating these heterogeneous resources. LDAP (Lightweight Directory Access Protocol)is used to store resource identity information so that all kinds of resources could connect to LDAP to obtain identity information. The adapter is implemented based on NSS and PAM. In Windows OS, GinaDLL, developed by Microsoft, is used to replace the default MsGina, so that information in LDAP can be obtained from Windows OS instead of local SAM. The data access interface is published as a web service for applications to obtain identity in-formation. For Apache servers, mod_auth_ldap in NSS is used to implement the adapter.

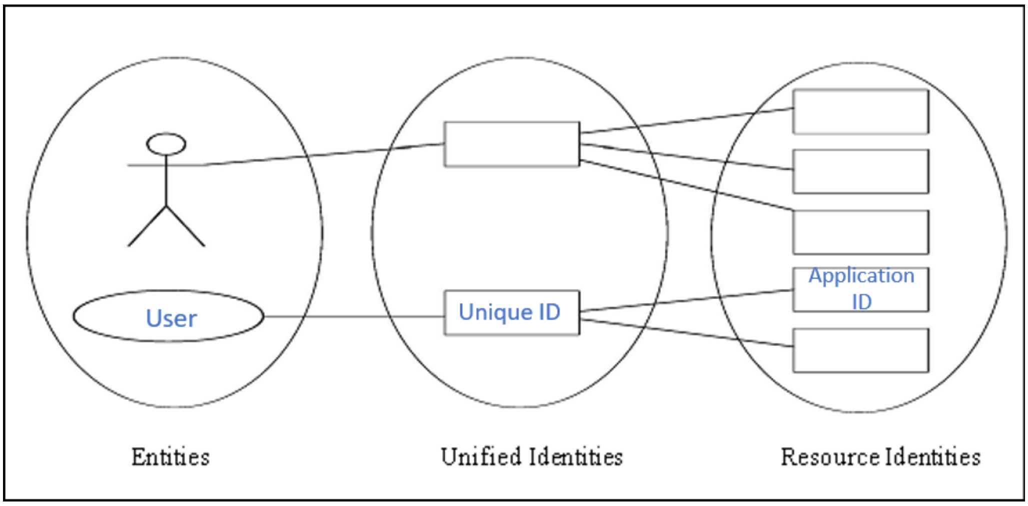


Figure 2.9: HR-OUIPM provisioning model concept of unified identity through mapping resource identities to unified identity information (which contains all attributes of that resource) using three mechanisms: a Request Parser, SPML engine and an Adapter (Liu et al. (2013))

Existing identity provisioning model are mostly user-oriented , service provider-oriented and Network-oriented (Stein et al. (2007)). These models did not consider resources, so they lack the ability to integrate heterogeneous resources efficiently.

In user-oriented identity provisioning models, the user is the centre of the system, such that every kind of information is under the user's control. This can be implemented in several ways, such as SAML (Cantor et al. (2005)), a UAC module in Windows OS, or a SUDO module in Linux OS. User-oriented identity provisioning models can make users obtain and update trust values more efficiently and can protect user's privacy to a certain degree. Shortcomings of these models are that security policy configuration is complicated, and it is difficult to manage identity information.

Service Provider-oriented identity provisioning models mainly concern service providers. These kinds of models maintain mechanisms to select security services dynamically, including authentication, authorisation and access control. Kerberos (Abdul and Wilson (2019)) is an authentication protocol that implements this kind of model and support multi-domain authentication through trust, however scalability can be a problem as the no of domain increase and for hybrid cloud environment where different authentication protocols are used . SP-oriented models are usually inexpensive and easy to deploy. Because the service provider completely controls identity management, identity information is managed safely and efficiently. However, this kind of model is not convenient for users to utilise.

Network-oriented identity provisioning models mainly concern configuration and management of networks, and related security and access control issues. The advantage of these kinds of models is to reduce cost and fully reuse hereditary resources. They also have the ability to control inter-operation between systems and ensures security transformation on the transport layer. However, this kind of model does not take the user experience or the service provider into account. HR-OUIPM implements identity provisioning of heterogeneous resources, offers an access interface for unified identity, which is the basis of identity management and single sign-on.

Khamadja (Khamadja et al. (2013)) proposed another access control as a service model for highly flexible and dynamic environments such as cloud computing, called CatBAC (Category-based Access Control). This is used for building dedicated access control models starting from a generic meta-model of access control, with two stages of refinement. In the first stage, the meta-model is refined into an abstract model according to the high-level policy of the organisation; this stage is completed by the cloud provider. The second stage allows for the generation of several concrete models from the abstract model by network administrators at the various sites of the organisation, respecting the local constraints and specificities of each site. The method illustrated in this paper gives cloud providers an access control, security solution that can be a cloud service for both providers and users. It allows users to define their own low-level policies in such a way that these policies can be refined correctly from the abstract policy defined by their cloud provider.

A review of literature indicates there are no generally agreed frameworks for identity access management within enterprises because cloud IAM (even hybrid IAM) is still a maturing market, there is work that needs to be done for it to continue to gain acceptance by organizations large and small. The first and most important are related to standardization, in process, methods, and tools. This means standards in protocols and authentication methods that need to be supported across IAM providers. Industry standards will emerge, as will IAM frameworks (Waters (2016)). Another identity provisioning model was proposed by Koch (Koch and Wörndl (2001)), which concentrates on storage of user information, privacy and cryptographic means of authentication and concentrates less on access control, provisioning, data management or governance.

According to Windley (Windley (2005)), "a coherent set of standards, policies, certifications and management activities aimed at providing a context for implementing an identity infrastructure that meets current goals and objectives of the business". Later, White (White et al. (2007)) proposed a framework combining identity administration of entities with their identity-based access management, to control access to the resources of an enterprise in Fig 2.10. This framework brings business requirements and policy into a logical structure which then becomes part of the identity management infrastructure, however this is a theoretical model and would require further research to determine that it is suitable for implementation for different enterprises and federation.

Damon (Damon and Coetzee (2013)) proposed Identity and Access Assurance (IAA) model using White's model as seen in Fig 2.11. This framework incorporates nine levels of requirement for IAA using SABSA methodology. This model provides insights into

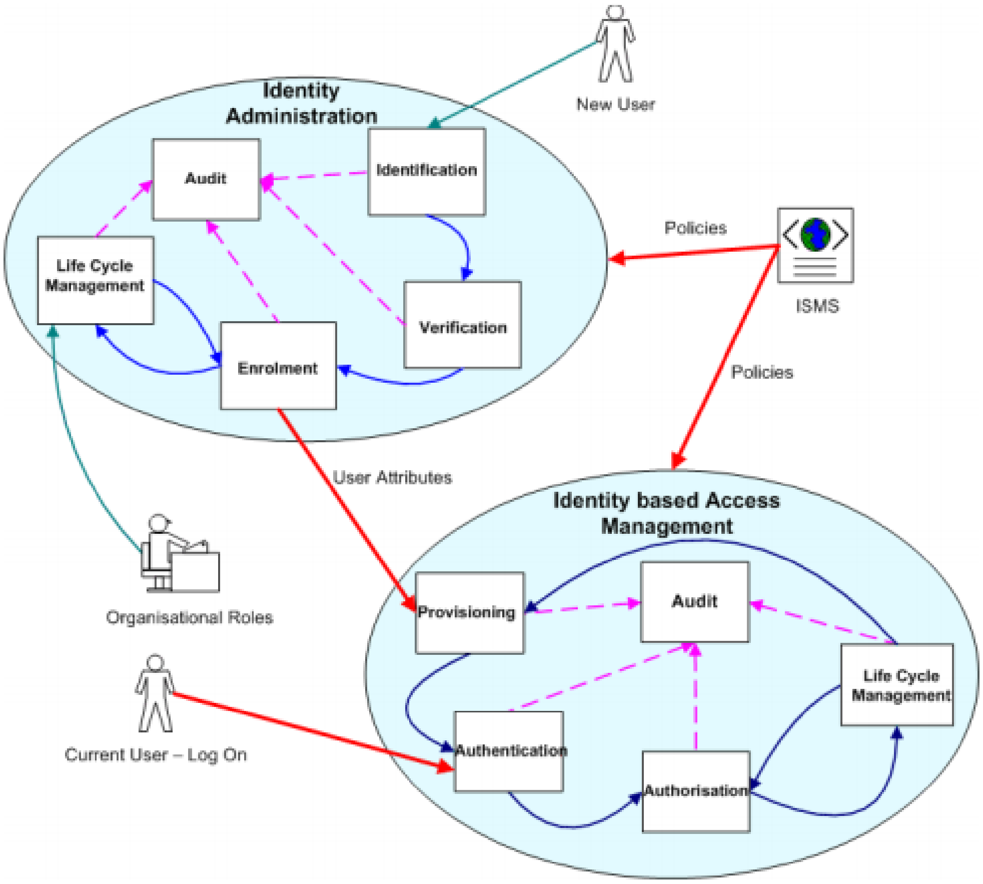


Figure 2.10: Internal enterprise framework incorporated business requirements and policy for identity management infrastructure (White et al. (2007))

the IAA components and business processes at a business owner level, removing technical complexity and providing an explanation of potential impacts on the business. The research references a single framework and architecture and have not mapped or associated their models into the identity and role access management domain.

2.5.1.1 Intercloud Architecture Framework (ICAF)

Demchenko (Demchenko et al. (2014)), has described a research effort at the University of Amsterdam to develop the intercloud Architecture Framework (ICAF), to address the problems of multi-domain heterogeneous cloud-based application integration and inter-provider and inter-platform interoperability. This paper defines the basic scenarios in federated cloud service provisioning and access control that include both a user side federation model and a provider side federation model. The paper defines the main roles and actors in the cloud federations, to address many practical problems in smooth multi-provider service integration and delivery to enterprise or campus users, using (national research and education network) NREN and campus-based identity management services as a trusted third party which is expected to facilitate creation of dynamic federations

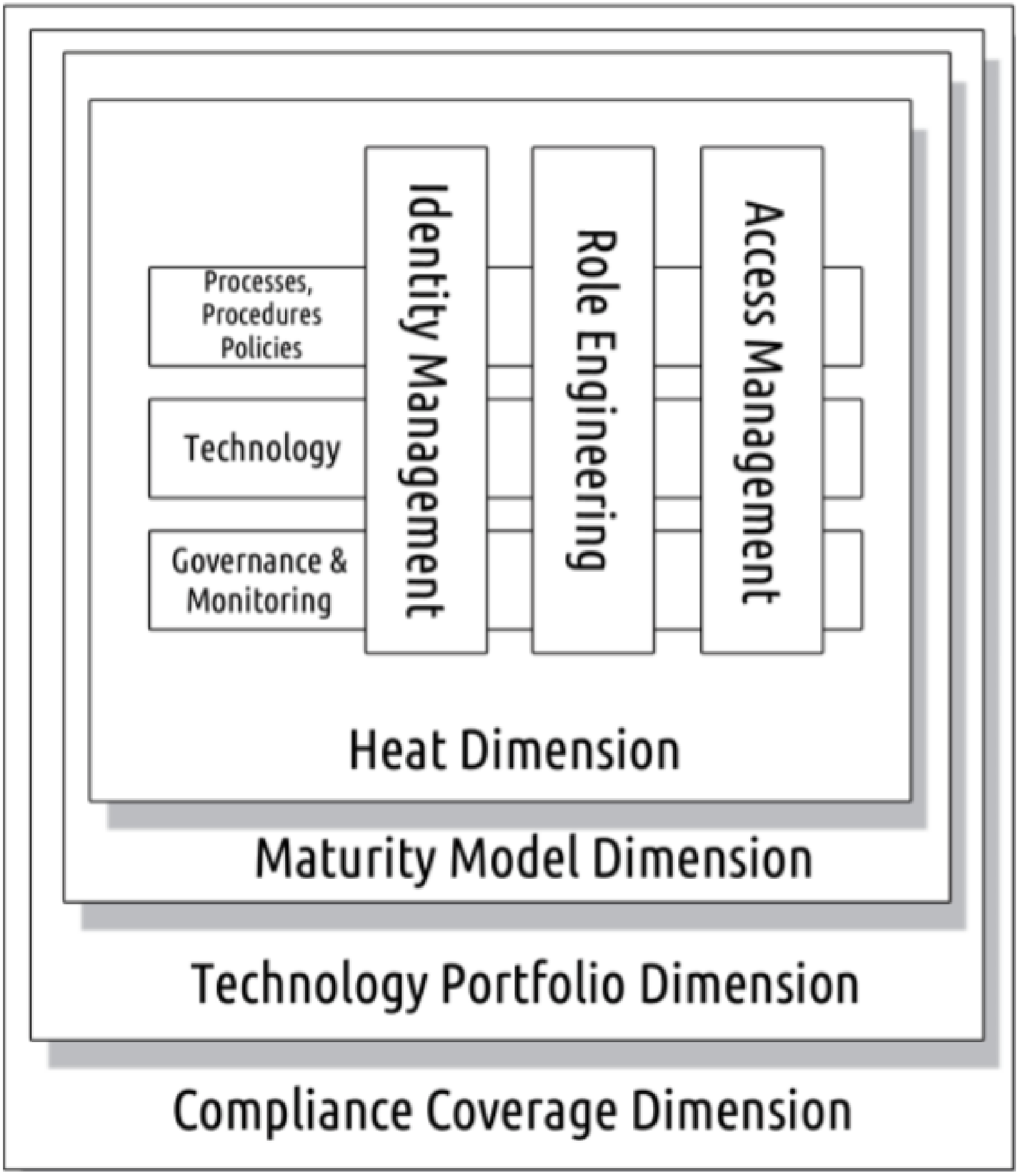


Figure 2.11: IAA Identity access management framework using SABSA methodology to map business processes at business owner level to highlight potential impact on business (Damon and Coetzee (2013))

between multiple cloud service providers and customer organisations. Further research will include modelling of the proposed intercloud federation models to evaluate effective methods for identity provisioning and access control policy evaluation in a heterogeneous intercloud environment.

2.6 Identity and Access Management (IAM)

IAM refers to digital identity in a corporate environment and needs to be treated with high priority. Irrespective of the different applications and platforms used by different organisations, resources need to be managed and allotted to the appropriate identity/user (i.e. provisioning management) with proper access rights (access/policy management). This process is called identity and access management (Kumar and Rodrigues (2010)). Kumar and Rodrigues have used previous survey carried out by the Forrester Research and Burton Group, an independent worldwide technology and market research group to evaluate five top IDM vendors, namely IBM, Novell, SUN, Oracle and CA based on; Strategy and Vision (Identity management vision and breadth of solution) in Fig 2.12, and six features and capabilities of identity management (IDM); policy and role management, data management, access management, setup and integration, administration and self-service and customer reference, in Fig 2.13. The authors concluded that even after years of healthy adoption rates, the IDM market is just beginning its path to broad adoption and deeper penetration. A strong identity and access management (IAM) strategy is an important element of any programme to prepare organisations to comply with new General Data Protection Regulation (GDPR). Recently (Kunz et al. (2014)) carried out another survey on IAM trend predicted by analyst within Capgemini, Ernst & Young, Gartner, Forrester and KuppingerCole (including finance) against published literature, the results suggest that shifts in IAM is towards managing risk and risk reduction and risk is the top strategic priority for industries to prevent security incidents and data breaches.

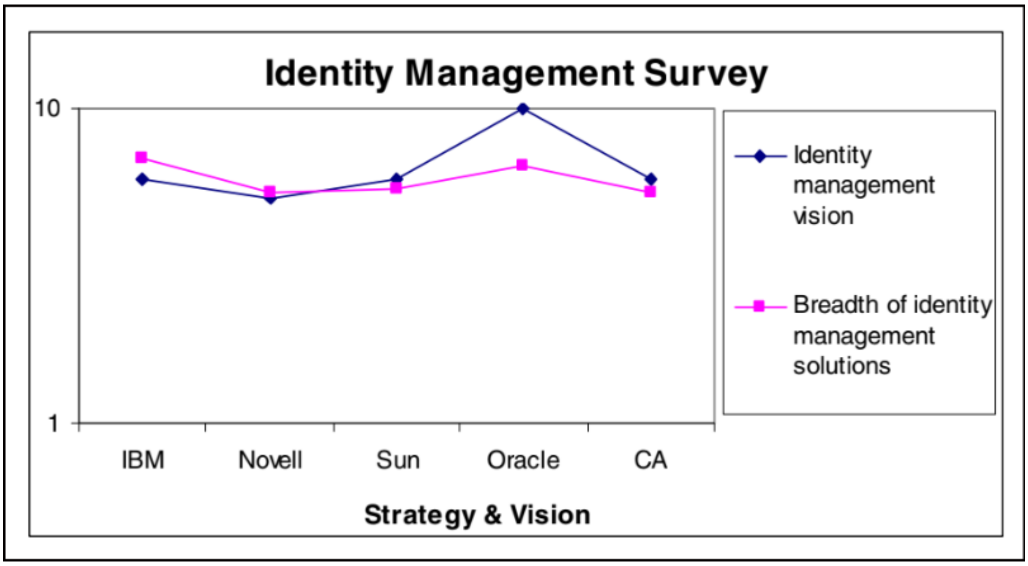


Figure 2.12: Comparison of idm on strategy & vision with respect to vision and depth of solution scored out of 10 scale (Kumar and Rodrigues (2010))

2.7 IAM Functional Taxonomy

Identity and access management encompasses three functional areas: Data Security, Provisioning and Compliance.

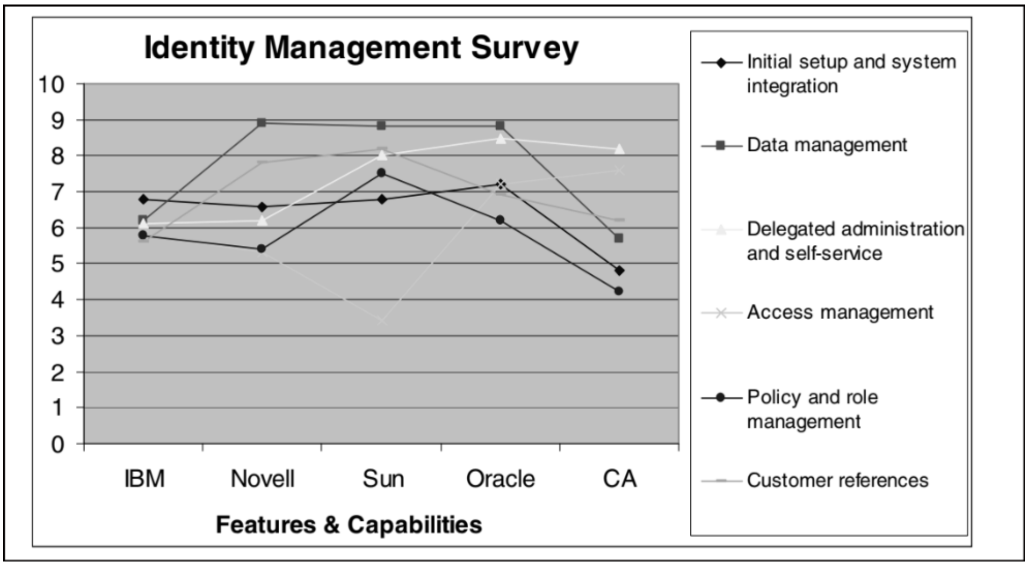


Figure 2.13: Comparison on Idm's features and capabilities factors scored out of 10 scale (Kumar and Rodrigues (2010))

2.7.1 Data Security

A systematic review on identity access management (Uddin and Preston (2015)) revealed results in Fig 2.14, it showed the intensity of research carried out in the different functional taxonomy of identity access management. Between 2010-2013, 33% of all the articles selected contained research on data security such as: Data in transit or still, data storage, data model and privileged accounts access data, According to (Caldwell (2013)) data in transit or still will need to be protected to avoid data breaches. As shown in Fig 2.14, cloud security has been grouped with data security, as this is a form of mobile data storage, and research on this topic comprised about 14% of all articles. There were total of 47% articles containing research on data security. Although there is a high level of research interest in data security, it remains under-researched, as the technology is evolving and new research areas are emerging, such as mobile data, BYOD storage data, cloud data; there is no concrete solution for data security. According to the 2019 data breach report from Verizon Business, 10% breaches were Financial Industry, 38% of breaches were caused by insiders, 15% breaches were misused by authorised users and 71% breaches were financially motivated, and 29% beaches were involved stolen credentials.

2.7.2 Provisioning

Provisioning refers to granting, managing access to an identity with supporting confidentiality, integrity and availability. From the systematic review in Fig 2.14, IAM solutions have been considered as part of provisioning, as the security domains are interrelated and difficult to separate into individual domains. IAM Standard and IAM solutions have been grouped into the provisioning domain, and 6% of all research articles in information security have information related to IAM solutions. The IAM solution provides auto-provisioning tools to reduce operational cost and reduce the risk of security breaches by eliminating redundant accounts and segregation of duties and 6% of the included articles contained research on IAM standards, that's a total of 12% of research articles containing research on Provisioning. According to Brandessence Market Research via COMTEX (MarketWatch (2019)), Identity and access management (IAM) Market based on; Global Size, Trends, Competitive, Historical & Forecast Analysis for 2019-2025, growth in occurrences of cyberattacks has considerably augmented the adoption of consumer IAM solutions among organizations. Global Identity and access management (IAM) Market is valued at USD 10.41 Billion in 2018 and expected to reach USD 24.52 Billion by 2025 with the (Compound Annual growth rate) CAGR of 13.02% over the forecast period.

2.7.3 Compliance

The compliance is the third functional taxonomy of IAM and has been divided into the subgroups of policy, security awareness, and workflow, as shown in Fig 2.14. The articles that have been reviewed contain information on policy (14%), compliance (8%), workflow (6%), and assurance (3%). Combining all four subgroups provides 31% of all research articles. It is also noted that in the year 2013, security awareness was the most heavily researched topic. This could represent the emergence of a new research field in response to ongoing data breaches and loss of data, to understand the underlying causes. In a 2012 survey of 2,000 members of the UK public by Check Point and Yougov, over 50% of office workers said they regularly do not follow security best practice, and 23% weren't aware of what their company's policy stated. Another survey carried out by PWC (PWC (2015)) in 2015 indicates 72% of organisations where policy was poorly understood and staff related breach. According to the 2019 data breach report from Verizon (Verizon (2019)) Business, 10% breaches were Financial Industry, 38% of breaches were caused by insiders. The identity and access management market are primarily driven by the increased demand in security governance, enforcement and concerns due to distributed systems, according to Varonis (Varonis (2019)), three billion Yahoo accounts were hacked in one of the biggest breaches of all time. Whereas in same year Uber reported that hackers stole over 57 million riders and driver's information.

2.7.4 Identity and Access Management Issues

The first phase of IAM which was geared toward on-premise traditional infrastructure within the organisation has been unsuccessful and is still under development and not fully understood (Everett (2011)). Data owner is lacking knowledge to understand every application and its sensitive nature, one of the pitfalls highlighted in Protiviti report

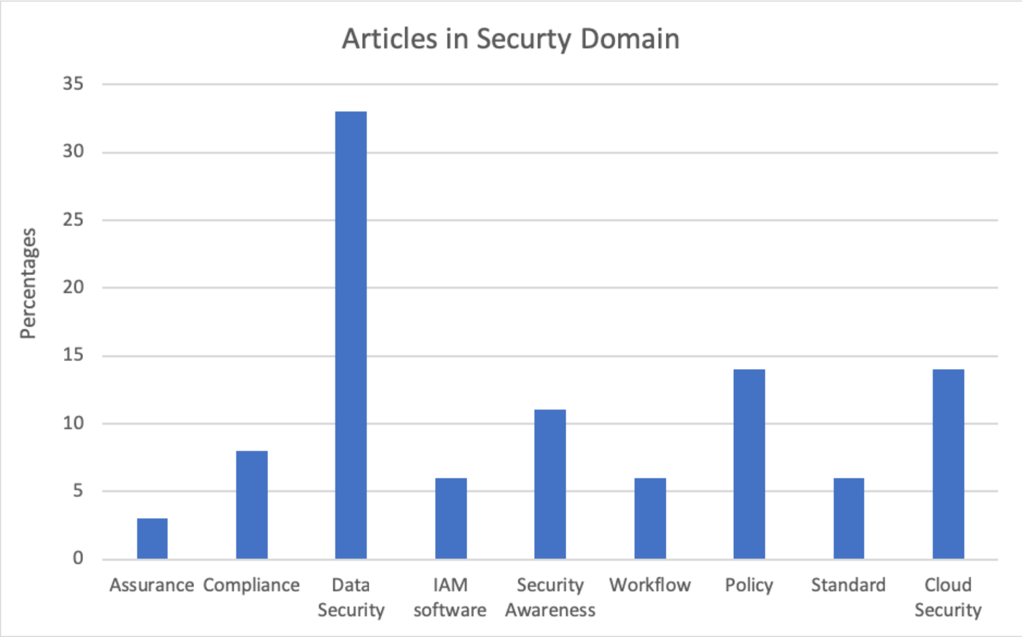


Figure 2.14: Number of articles in information security domain identified through systematic literature review of identity and access management

“Identity and access management in financial service-staying ahead of the curve” (Beaumier (2019)). It is the business that needs to understand the application system and its processes before deploying a tool. It is important not to implement a tool to assist with the workflow process if that process is not working efficiently. Every organisation is implementing controls or patching up broken systems to satisfy compliance legislation (Hart (2013)); there are little, or no involvement of senior stakeholders and the failure of IAM was due to the implementation of an off the shelf IAM system. Change that are been put in place within the organisations are a reactive approach and under the control of higher management.

Due to the de-parameterised nature of the business operation, data are accessed via mobile devices, users are using their own devices at home to connect to sensitive data, and much IT infrastructure is based in SaaS or the cloud. According to (Dinoor (2010)) privilege is not just managing account privilege, it is also the multitude of contexts in which users, accounts, data, applications and processes interoperate. Privilege is generally understood in terms of controlling users who have high levels of authorisation to access, and control over, corporate IT systems, information assets and applications. Poor control over privilege accounts can be risky from a security perspective. Despite overwhelming advantage of Cloud technology Security and privacy issue of user identities is still a concern (Nida et al. (2014)).

Up until now, networking teams have been involved in security system scanning and patching up infrastructure, and information security teams involved in provisioning user entitlements, however, both teams need to work together to find a better security solution for IAM. As a corporate outsource to manage servicing, hosting and cloud providers, increasingly, have direct control over customer organisation data.

2.8 Existing Access Control Model

This section of the thesis analyse the existing access control models within the academic knowlegebase and its limitation, to identify what is known within the literature to identify gaps within the literature.

2.8.1 Discretionary Access Control (DAC)

DAC permits the granting and revoking of access control privileges to be left to the discretion of the individual users, mechanism allows users to grant or revoke access to any of the objects under their control. As such, users are said to be the owners of the objects under their control. However, for many organisations, the end users do not own the information to which they are allowed access. Access priorities are controlled by the organisation and are often based on employee functions rather than data ownership. Disadvantages of the discretionary model are segregation of duties (SoD) violation (due to fail-ing to link entitlements to a user business function (Lu and Jiang (2006)) as shown in Fig 2.15, policy violation with user groups not knowing who a member of the group is, inadvertently creating a breach whereby the object owner granting access using a pre-exiting user group is unaware of how the access has been authenticated. Job rotation leads to orphaned accounts which are inherited from the previous business function.

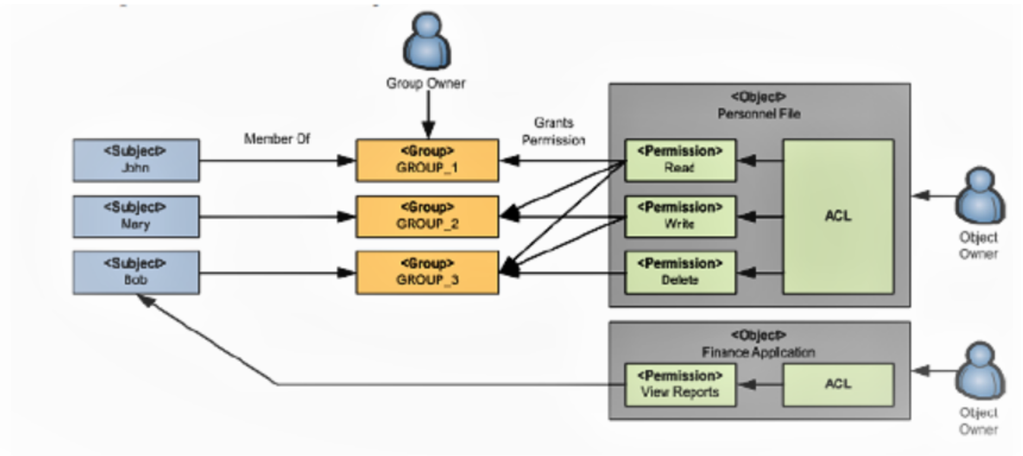


Figure 2.15: Discretionary Access Control Model, based on the discretion of the employee functions (Lu and Jiang (2006))

2.8.2 Mandatory Access Control (MAC)

MAC, as defined in the DoD's Trusted Computer Security Evaluation Criteria (TCSEC), also known as the Orange book, is "A means of restricting access to objects based on the sensitivity (as represented by a label) of the information contained in the objects and the formal authorization (i.e. Clearance) of subjects to access information of such sensitivity" (Tassey et al. (2002)). Although MAC is well-suited for military applications, it is not well-suited to commercial and dependant organisations as shown in Fig 2.16, due to diversified and complex organisation systems require a policy-independent access control model.

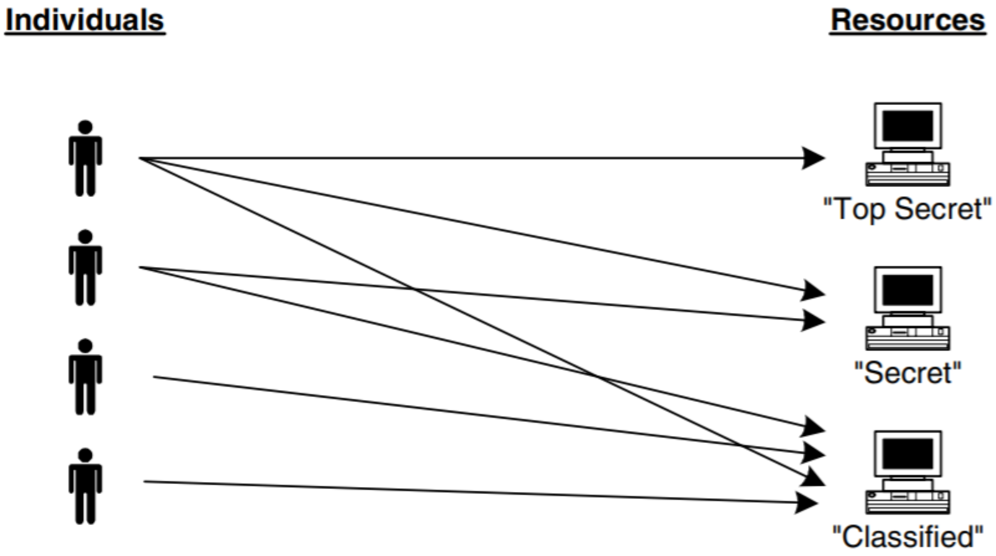


Figure 2.16: Mandatory Access Control Model, based on access to objects on the sensitivity of the resources (Tassey et al. (2002))

2.8.3 eXtensible Access Control Markup Language (XACML)

XACML is based upon XML and was developed to specify access control policies in a machine-readable format (Rissanen et al. (2013)). 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) 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. In practice it can be very difficult to establish one authoritative attribute source. This is especially true in situations in which different enterprises must work together and must implement access control between themselves.

2.8.4 Risk Adaptive Access Control (RAdAC)

Risk adaptive access control uses information from the environmental condition and risk level, combining information about a subject machine, corporate IT infrastructure, and environmental risk factors for the decision-making process (Farroha and Farroha (2012)). An advantage of this approach is that if the policy allows then decisions can be overridden where necessary, for example, in a high risk environment it will enforce dual authentication, and in a low risk environment it will make a decision based on the digital policy. A disadvantage of this approach is that like policy-based access control (PBAC) it relies on digital policies, and if they are ambiguous, then it can result in a security breach.

2.8.5 Role Based Access Control (RBAC)

Role based access control (RBAC) is a framework using roles to control access permissions. Users are grouped into roles, each role may have several members and a set of de-fined granular levels of credentials (Sandhu et al. (1996)). A user will only have access to information which has been allowed according to their role, and this will prevent unauthorised access to information and possible security breaches. The RBAC model achieves the two principles of security systems: “segregation of duties” and “least privilege”. Least privilege is where a user is granted access to perform their day-to-day business function; this prevents intentional or unintentional damage to the system and under-entitlement or over-entitlement or combinations of inherited permission access rights. Segregation of duties is where roles are mutually exclusive, for example a trader cannot both enter and release their own trades. Roles can have separate permissions grouped into a high privilege role.

NIST developed and published a comprehensive RBAC model in 1992, providing the first Role Based Access Control (RBAC) technical specifications and formal description, followed by an expanded model in 1995. NIST, with Ravi Sandhu, at the time with George Mason University, proposed a standard for RBAC in 2000. This proposal was revised in 2001 and NIST drafted the final standard proposal and led the ANSI/INCITS RBAC standardization committee. ANSI/INCITS 359-2004 (Sandhu et al. (2000)) RBAC was adopted in February 2004 as shown in Fig 2.17. The proposals and adopted standard largely eliminated the uncertainty and confusion about RBAC’s utilities and definition; it has served as a foundation for software product development, evaluation, and procurement specifications.

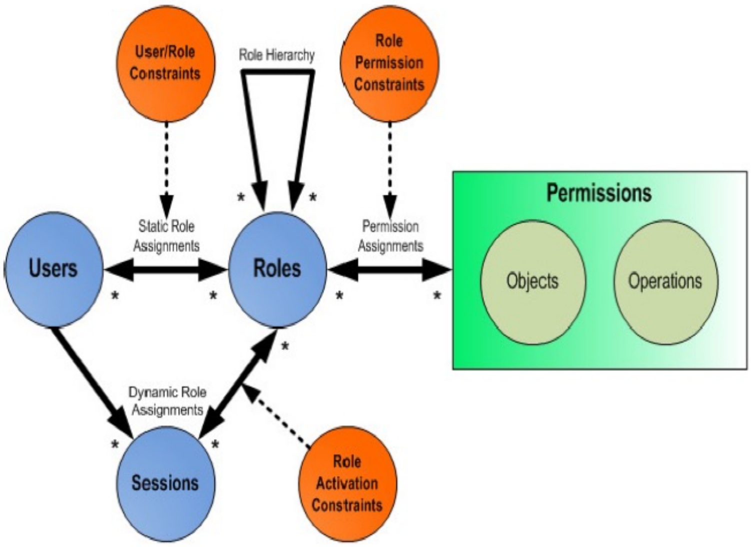


Figure 2.17: Role based access control (rbac) role architecture, showing roles connected to resources and subject to prevent unauthorised disclosure of resource (Sandhu et al. (2000))

2.8.6 Attribute Based Access Control (ABAC)

Attribute based access control (ABAC) has no consensus model to date (Tassey et al. (2002)). The concept is that is user and resources have attributes known about them, either through situational data, such time of the day or which people are logged on to the network, or through user data such as title or location. The system can make instantaneous decisions about whether the user is appropriately authorised to access the object or to perform the function. The data elements analysed are known as attributes. The advantages of this approach are greater 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 that access control policy might become more dynamic than preferable for audit and attestation, as it requires many rules which makes analysis difficult (Wang et al. (2004)). The user entitlement access report is difficult to understand, and the access is based on attributes rather than entitlements (Rajpoot et al. (2015)). 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).

2.8.7 Policy Based Access Control (PBAC)

Policy Based Access Control (PBAC) is an emerging model that seeks to help enterprises address the need to implement concrete access controls based on abstract policy and governance requirements. This approach is an extended approach to ABAC, it supports specific governance objectives, it uses the attributes from resources, the environment and the requestor's information to permit or deny requests for access to resources (Wang et al. (2004)). The disadvantage of this approach is that it requires application-level logic, enter-prise-wide, but also requires a mechanism to have unambiguous policies to prevent authorised access to resources.

2.8.8 Workflow Access Control Model

A workflow is an automation of business processes in whole or part, during which information or task is passed between participating in activities or action (Brambilla et al. (2011)). Ma (Ma et al. (2011)) proposed a policy-based workflow management (PBWF) model, 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 show that both dynamic and static access control is needed in a workflow in Fig 2.18. However, there are various authorisation policies within an organisation which have not been studied much. There are many systems with many different access control models and resolving the conflicts between different authorisation policies to integrate into this is challenging. Bertino (Bertino et al. (1997)) proposed a SoD for workflows, however it focused more on defining roles that are SOD rules and constraints and did not consider task instance constraints. This model focused on SoD within workflows, which requires prior knowledge of the specification and its task, thereby limiting the on-time decision making. A similar model proposed by Chadwick (Chadwick et al. (2007)), Multi-session SoD (MSoD), where the focus is SoD both at the instance level, and permanently, utilises the business context concept, has resolved the instance level restriction, however lacks decision-making intelligence, as it requires predefined business context and identification SoD policies. An active access control mechanism does not provide task-based authorisation. WSession (Botha and Eloff (2001)) proposed a similar model which requires a pre-identification of all conflicting roles, users, tasks, and privileges and lacks support for active access control and task-based authorisation.

Various other workflow systems have been proposed based on the RBAC concept such as Weber's (Weber et al. (2005)), where the focus is primarily on security and flexibility in workflow systems emphasising dynamic aspects of sequence and adaptability at runtime. Authorising Workflow Role based Access Control (AWRBAC) is another workflow model based on extended RBAC and focuses on adaptive workflow systems and lacks instance level restrictions and task level constraints in proceedings (Leitner et al. (2011)). A generic meta-model that can be used to extend workflow languages to support access control requirements in workflow systems has been proposed (Strembeck and Mendling (2011)). This meta-model is designed to be used in conjunction with workflow languages with on-time and run time constraints (including instance level restrictions). This model has been designed to act as enforcement/decision making which would reside on top of an existing model, cannot be used on its own and thereby, another dynamic access control model is required. It also does not have the capability to handle task authorisation requests.

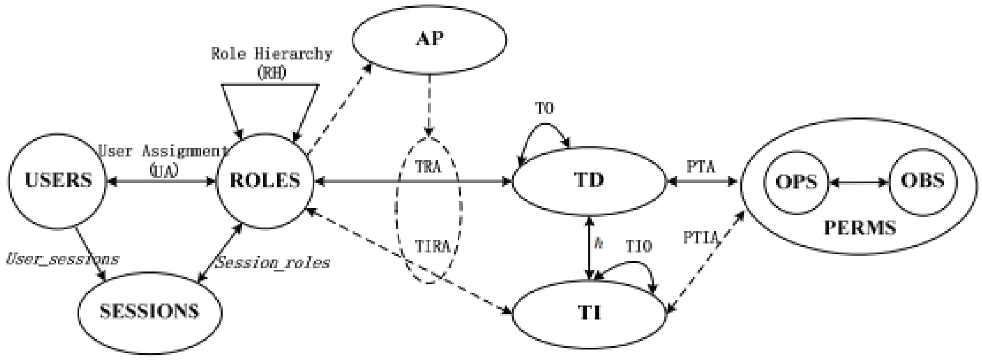


Figure 2.18: Policy based workflow management model, based on polices on business processes depicting needs for active access control (Ma et al. (2011))

2.9 Access Control Adversaries

A recent survey of 40+ security specialist industries by OWASP Inc discovered two critical access control risks, insufficient logging and monitoring, and broken access control in top 10 list of the risks (OWASP (2017)). The threat (adversaries) is designed using the threat model within application security 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 models to determine the likelihood of adversary (threat) (Do et al. (2018)). Typically, the goal of the 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 absent logging of events, such as failed login transactional logs, allow attackers to further attack systems, maintain persistence, pivot to more systems, and tamper with, extract, or destroy data. Broken access control refers to restrictions on what authenticated users are allowed to do, which are often not properly enforced; the 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 a parameter value that directly refers to a system object for which he is unauthorized through applications and APIs, where all the user privilege requests are not verified resulting in improper privilege escalation. Another research project on the risk adaptive access control (RAdAC) has suggested using information from the environmental condition and risk level.

Recently, multiple dynamic attributes such as application usage and unlock failure is considered for ensuring access control and data confidentiality of mobile cloud environment (Agrawal and Tapaswi (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 (Adavoudi-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.

2.10 Limitation of Existing Access Control Models

After careful consideration of the two widely used access control models, they are essential and effective, however there are several limitations of these models in the context of supporting current business processes. These limitations are discussed below.

RBAC supports a limited number of different types of authorisation constraints, which cannot fulfil the requirements that have emerged in modern organisation's business processes (Ferraiolo et al. (2001)). A typical constraint, which is very relevant, well-known and probably the most used in the security area is Segregation of Duty (SoD). Although there are many variations, SoD is fundamentally a requirement that critical operations are divided among two or more people so that no single individual can compromise security.

RBAC is an essential concept in the workflow access authorisation model. However, segregation of duties applies to the role level. In a business process context, segregation of duties can apply on the task instance. This is not supported by RBAC, sometime restriction is on task to be performed by the same person which is referred to as BOD (bind-ing of duties), and the person who issued the task should close the task. This is to prevent fraud, misuse of privilege and error. In workflow authorisation models, SOD and BOD constraints are required for same instance, i.e. one cannot "submit" and "approve" in the same instance, at the same time, the person who "submitted" the request needs to "close it" in the case of emergency password release (security team needs to "issue" the password and "check-in" the password). The current business workflow model does not accommodate instance level restriction (IR), or order of SOD (Knorr and Stormer (2001)). In business processes, multiple role instances allowed for single user, to maintain segregation in this context an "instance level restriction" needs to be enforced. This implies that order-based separation of duty in role-based systems should be used in the context of workflows.

If a user terminated (revoked) what happened to their session, which had been activated through their role, should the role be terminated instantly or retained for a period before terminating it (deleting it). This has not been specified in any authorisation model, nor in the NIST standard (Thomas and Sandhu (1998)). Rules for revoking of user sessions immediately as well as retaining the session active for a period while disabling the account for audit purposes when requested are entirely missing from RBAC.

RBAC is policy neutral and can express DAC and MAC. However, role-based access control has its own set of limitations such as role explosion and role-permission explosion (Rajpoot et al. (2015)). It is also restrictive in nature since the accesses are based only on roles and it is difficult to include other characteristics of users, and contextual or environmental factors (e.g.time, location, etc.) in access control policies. User access entitlement report generation is difficult and authorised access is driven based on attributes (context) of the subject rather than user credentials. Although ABAC research has received significant attention in academia, it is not so common to find implementation of these models in industry. There are a few existing tools such as XAMCL and Policy Machine (Ferraiolo et al. (2001)) that can express different types of Attribute-based Access Control Policies. However, wide adoption of these tools remains a challenge.

Using “tasks” in tasks based access control it easier to deal with tasks rather than permission particularly since a bundle of permissions is often required to perform a unit of work at an application level. Therefore, it is useful to use task-based allocation. Each task has an associated set of related permissions, where each permission is a pair made up of an action and a resource. For example, the task “close role transfer”, includes the set of permissions (access, security Request Database), (read, request), (check, role profile blueprint), and (edit, Access request database). The notion of workflow tasks involved in role transfer, departmental transfers. A request submitted by “Security Coordinator” which needs to be approved by the role “Revocation Manager”, which is then required approval from the on boarding “Department Manager”. It is important that task sequences are followed in order otherwise it will result in audit malfunction and a user having more access than required. This workflow task is not supported by the XACML-RBAC standard as RBAC is a passive policy model; the XACML standard needs to support both task notation and complex role transfer validation through role change policy using an additional function.

2.11 An Industrial Use Case and Limitations

The case scenario developed for this research has been chosen from a real life complex, sensitive environment that uses workflow processes across multiple systems. Processes that run across multiple systems are a common feature of the modern business landscape and they represent a challenge for workflow security and access control.

2.11.1 An Investment Bank Use Case

This research is based on an identity access management for a global investment bank based in London. Due to the data privacy act and the confidential information (GDPR), the investment bank will be referred to only as Investment Bank X. It has assets of over a trillion dollars and 17 locations worldwide. It is a privately held financial institution and has been a leader and a solution provider for over 200 years. Bank X is considered an expert in corporate banking, merger & acquisition advice, investment management, wealth management, and investor services.

2.11.2 System Context and Existing Business Process

In the access control workflow process for the high security environment of an investment bank, requests for access are submitted via a security request database, which will then send a notification to the approving manager and the final approval request will reach the information security department to be actioned in Fig 2.19. The organisational security policies and rules require “least privilege” and “separation of duties (SOD)” to minimise fraud and error.

2.11.3 Current Access Control Model and Workflow

The current access request workflow runs as two separate workflow systems for Internal Client and Privilege Account Management (emergency password). There are three roles involved: “Security Coordinator” (who submits the request), “Manager” (who approves the request), and “Security Administrator” (who grant access to the resource). As can be seen in Fig 2.19 the access request to the banking system should only be submitted by the role “Security Coordinator”. Approval of that request will need to be “approved” by the role “manager”. Access is granted after satisfying the condition that supporting documents with management approval are provided to another role “security administrator”. Once access has been provisioned and the security coordinator has been notified, the role “Security Administrator” can close the access request. To enforce segregation of duties (SoD), a Security Coordinator cannot be an “approver”.

In a more dynamic process where roles or departments are transferred, two levels of approval are required: approval from the manager the user is transferring from, as well as the manager the user is transferring to. It is essential to enforce that the first approver approves the requests before the second approver, as sequence of tasks needs to be maintained. The old credentials need to be revoked before granting new credentials. Maintenance accounts are high privilege accounts which are required for emergency patching and testing. Maintenance accounts are owned by the IT support teams who also can change the password, as part of the binding of duties (BoD). The security team who check out the password should check-in the password.

In the case of role termination, user accounts need to be disabled and the permissions need to be revoked. Accounts are retained for 30 days before deletion.

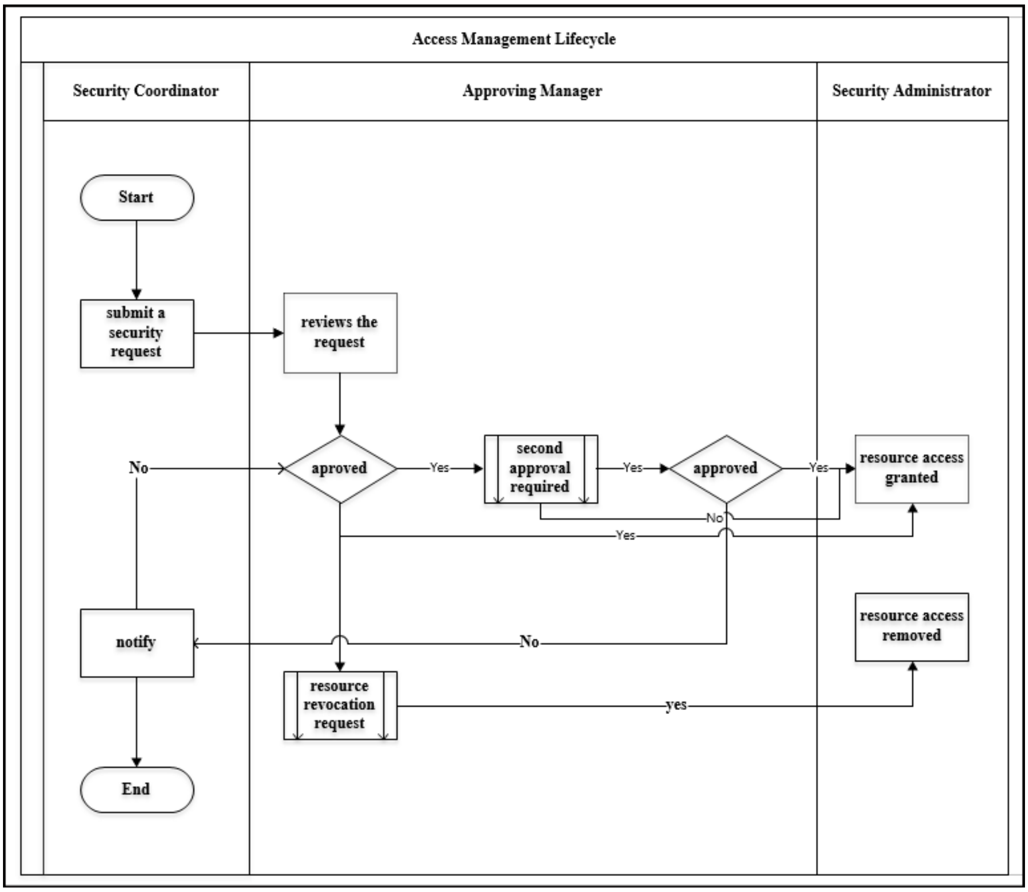


Figure 2.19: Current Identity and Access Management life cycle from the inception of a security request, management approval through to action of the request by security administrator

2.11.4 Current Access Provisioning

Access provisioning is the process of managing user identities into identity stores, initialising their credentials and enabling them to access IT resources. Access control provisioning consists of three processes: Access request, Access Authorisation and Access Administration in Fig 2.19. Access request is submission of a request by a security coordinator for a new joiner, leaver or transfers in the Bank requiring a list of required access based on another person's profile in the department. Access Authoriser is the high-level Manager who authorises the requests for new joiners, transfers and terminations. Access Administrators are responsible for granting access to the WEB, LAN and MAINFRAME applications for the new joiners and transfers, and revoking access from terminated employees.

2.11.5 How are Access Request Handled Currently?

Access request is a high-level policy which entails sequencing of tasks and approval from high level managers. A standardised workflow is very important in protecting information, audit trails and enforcement of policies within organisations. A workflow is a set of tasks to perform business functions, and tasks that are part of workflows require active access control (Jiang and Lu (2006)).

- Workflow: Mixture between role based access control process and traditional as per model ID based access request; this creates confusion between the requester, approver and administrators. Different workflows need to be followed using different systems. This results in ambiguous audit trails and ineffective administration by the system administrators.
- Emergency Password Workflow: During system maintenance and emergency maintenance of infrastructures, high privilege access requests are assigned to the developers. This follows its own workflow and approval; it is then subdivided into emergency access request in absence of authoriser and in absence of authoriser for scheduled maintenance. Audit trails and monitoring becomes an issue, and the Administrators bear the burden of changing the passwords for these accounts.

2.11.5.1 Current Access Authorisation Workflow

Access authorisation is a high-level policy set up by the organisation to enforce policies to reduce the risk of fraud. It is vital that that chain of custody is maintained, and granular level of access understood. Current issues within the authorisation workflow are:

1. High risk applications require additional approval to the main workflow, however, additional approval is primarily maintained via e-mail and not stored centrally for future audit trails.
2. Additional file share approval along with the access request workflow, additional approvals are not followed in the work for access to sensitive information stores (file share).
3. Comprehensive information when approving access requests, granular level details are not available to approvers, and high level managers are unaware if sensitive information will be disclosed to unauthorised parties.

2.11.5.2 How is Access Administration Actioned?

It is said that access administration holds the “key” to the kingdom, as misuse of data could lead to reputation damage, financial loss, fraud and compensation. Effective administration with robust technology coupled with security awareness is the key to secure information. Current issues within organisation administrations are:

1. Automated provisioning - Semi Automated Identity management, access tools, deployment resulted in ambiguous and ineffective administration. API connectivity with legacy systems has proven difficult
2. Role engineering explosion - Role engineering has been difficult and has resulted in a role per user. Role policies defined in the automated tool only function at the business process level as opposed to business function level and the application level. Adequate centralised role storage systems have not been possible to implement.
3. The cumbersome administration, is becoming cumbersome and ineffective, numerous access, workflow to follow and numerous instances of manual administration, which are prone to error.

2.12 Access Control Limitation Within the Industrial Case Study

The access control workflow process for the high security environment of Bank X would be submitted via a coordinator into a security request database, which will then send a notification to the approving manager for approval. The request will reach the Information Security Department to be actioned in Fig 2.19. A challenges is the three different authorisation systems operating independently, lacking in governance and user access reconciliation. The organisational security policies and rules of “least privilege” and “separation of duties” to minimise fraud and error become onerous, and role level restriction requires restriction within the task level, which is referred to as “instance level restriction”. Role change requires management approval in a sequence where manager in the current role is required to approve the change request first then the onboarding role manager is required to approve the request. It is necessary to ensure previous role credentials are revoked prior to granting new credentials. Current challenges and gaps within the identity and access management are as follows;

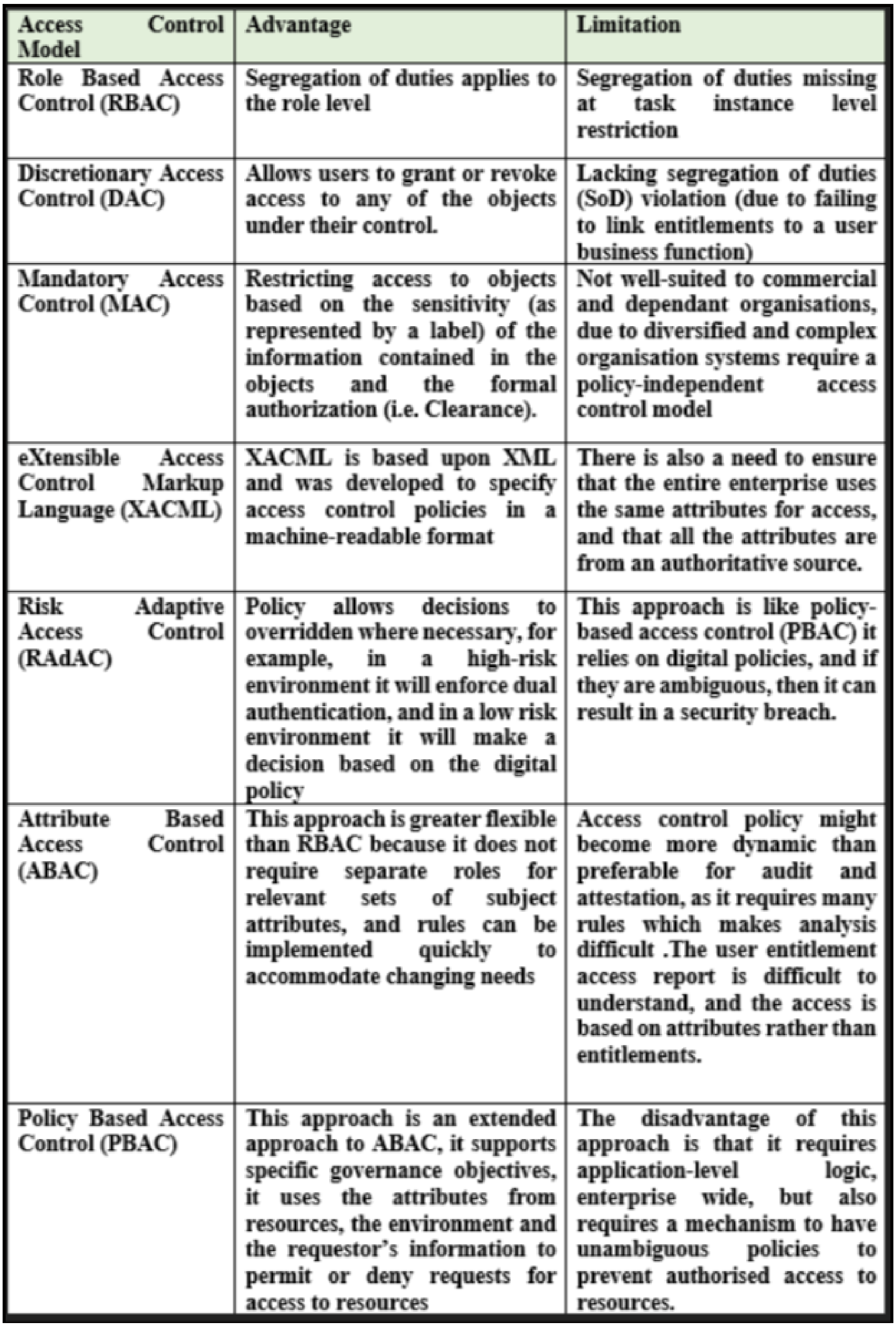
- There is a lack of dynamic access control to accommodate the diverse hosting of information. This could also impose the escalation of privilege risk.
- By passing vetting process, no visibility of data access due to inadequate business process workflow.

- 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.

Combining tasks-based assignment in conjunction with role-based will help make access control more efficient and easier to use. This approach will assist by not adding further to the already large number of proprietary authorisation systems that the organisation must manage. Instead, it is a path to consistent access enforcement based on a single set of organisation-wide security policies. An authorisation model for the enterprise should support both active (task) and passive (role) access control, otherwise the dynamic environment permissions could be switched too early or too late, for example, in the case of an emergency maintenance, passwords need to be assigned after an emergency and revoked within 24 hours (completion). Role-based access control (RBAC) is a natural paradigm to apply to authorisation in workflow systems because of the correspondence between tasks and permissions. A considerable amount of work has been done on the use of RBAC to support access control in workflow systems (Wainer et al. (2003)). However, a role-based model alone is not sufficient to meet all the authorisation requirements of workflow systems such as separation of duty constraints and binding of duty constraints. Separation of duty requirements exist to prevent conflicts of interest and to make fraudulent acts more difficult to commit. A simple example of a separation of duty constraint would be to require two different signatures on a cheque. Binding of duty constraints require that if a certain user executed a task, then that user must also execute a second task in the workflow. Additionally, cardinality constraints are used to specify that a task must be performed a given number of times, optionally by a given number of different users. Role-based access control has its own set of limitations such as role explosion and role-permission explosion (Rajpoot et al. (2015)).

Summary of limitation within existing access control model shown in Fig(2.20

From the literature review, gaps were identified within dynamic segregation of duty constraints, workflow authentication management, binding of duties, dynamic role change mechanisms and enforcement of organisational policies. It is apparent from a gap analysis of Bank X's access control authentication model that this organisation would require a dynamic access control model to accommodate challenges it faces within the new technology era, whilst minimising the data/security breach risk. Unauthorised access could lead to elevation of privilege and it is now more important than ever to comply with relevant privacy legislation.



Access Control Model	Advantage	Limitation
Role Based Access Control (RBAC)	Segregation of duties applies to the role level	Segregation of duties missing at task instance level restriction
Discretionary Access Control (DAC)	Allows users to grant or revoke access to any of the objects under their control.	Lacking segregation of duties (SoD) violation (due to failing to link entitlements to a user business function)
Mandatory Access Control (MAC)	Restricting access to objects based on the sensitivity (as represented by a label) of the information contained in the objects and the formal authorization (i.e. Clearance).	Not well-suited to commercial and dependant organisations, due to diversified and complex organisation systems require a policy-independent access control model
eXtensible Access Control Markup Language (XACML)	XACML is based upon XML and was developed to specify access control policies in a machine-readable format	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.
Risk Adaptive Access Control (RAdAC)	Policy allows decisions to overridden where necessary, for example, in a high-risk environment it will enforce dual authentication, and in a low risk environment it will make a decision based on the digital policy	This approach is like policy-based access control (PBAC) it relies on digital policies, and if they are ambiguous, then it can result in a security breach.
Attribute Based Access Control (ABAC)	This approach is greater 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	Access control policy might become more dynamic than preferable for audit and attestation, as it requires many rules which makes analysis difficult .The user entitlement access report is difficult to understand, and the access is based on attributes rather than entitlements.
Policy Based Access Control (PBAC)	This approach is an extended approach to ABAC, it supports specific governance objectives, it uses the attributes from resources, the environment and the requestor's information to permit or deny requests for access to resources	The disadvantage of this approach is that it requires application-level logic, enterprise wide, but also requires a mechanism to have unambiguous policies to prevent authorised access to resources.

Figure 2.20: Existing access control models and it's limitations

2.13 Summary

This chapter critically analysed the notion of Identity Management, widely adopted identity management models, and several expert's opinion survey. Literature review studies the evolution of identity management, concepts, identity management models, provisioning model,identity and access management framework through to access control model, this has been carried out through systematic literature survey to understand what is known and limitation in the field. Literature review then focuses on the adversaries within the access control to understand the threats vectors and the risk to organisation through breach reports and industrial experts (board of directors, security specialist) opinions. Current identity and access management model have been studied of the case study (investment bank) to derive requirements and current limitations and challenges faced through emerging technology to protect assets from various adversaries and risks.

Various research has been carried out on identity and access management models over the past decades and newer flexible and hybrid models are currently under research for the dynamic cloud environment and protocols; however, none has met the functional requirements of the dynamic access control model providing access governance to ensure access to resources authorised appropriately, logging capabilities, dynamic SoD at the task instance level to ensure access is granted in real time and mitigation of broken access control risk of privilege escalation.

The following chapter proposes methods that have been applied in this thesis to bridge the gaps in the research within the dynamic access control to ensure requirements are incorporated into the development of the AW-TRBAC model in chapter Four.

Chapter 3

METHODOLOGY

This chapter outlines the research methods that were followed in the study. It provides information on the participants, the criteria for inclusion in the study, the research design that was chosen for this study and the reasons for this choice. The instrument that was used for data collection is also described and the procedures that were followed to carry out this study are included.

The aim of this research is to construct a conceptual dynamic access control model for an investment bank. The objectives of the research are to identify the gaps in the Access Control Management existing literature, to build upon existing knowledge through an inductive approach, and solve a real-life problem with a chosen institution.

3.1 Research Methodology

The research methodology is a structured approach that specifies how research is to be conducted and by which the research goes about the process of describing, explaining and predicting hypothesis (Almalki (2016)). This thesis has used descriptive methods. This is because descriptive research is helpful in identifying variables that can be tested, subjects or participants are observed in a natural and unchanged environment, and the data collection allows for gathering in-depth information that is qualitative (observations or case studies) in nature. This allows for a multifaceted approach to data collection and analysis. Descriptive studies result in large amounts of rich data.

3.2 Research Design

The research begins with the challenges faced by a real financial institution. A review of the related literature is performed to consider what the existing industry and research practices are, and where there are gaps in the literature. This research focuses on the descriptive theory and the development of related concepts and procedures to facilitate dynamic access control features for and existing access control model.

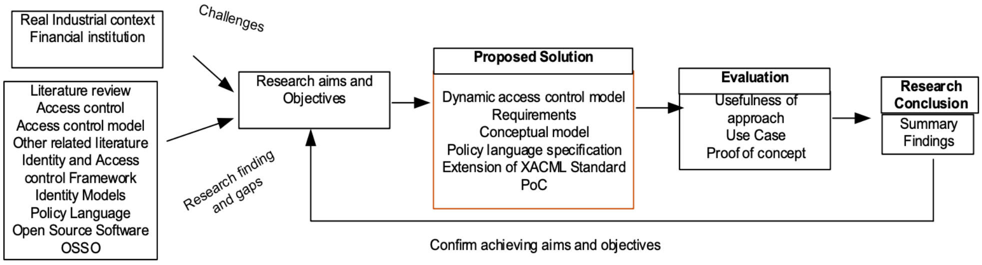


Figure 3.1: Summary of Research Design

When establishing research design, the following criteria were considered:

- Understand the problem from a real use case
- Establish a sound basis of knowledge about existing access control models
- Understand the requirements of a new dynamic access control model
- Verification of the proposed model using specification language
- Implementation of Proof of Concept (PoC) for Authorisation access control model
- Testing usefulness of the proposed model

The aims and objectives are considered based on the existing problems and research gaps as shown in Fig 3.1. The proposed solution attempts to address these problems through evaluation of the solution in order to achieve the research aims.

The analytical technique used for the research is that of a mixed empirical study. It contains a use case, evidence pertaining to which has been obtained via personal experience, observation and industrial experts opinion and survey. The information gathered has been analysed and grouped into three parts: access control model, access governance, and policy enforcement.

Another approach which has been used to analyse gaps in research is through a systematic literature survey of the Identity Access Control Management (IAM), as shown in Fig 3.2. This approach is common amongst the medical research field where large volume of data and this method of analysis assist in narrowing down to niche area, similarly IAM is well researched are and have attracted vast interests from academia and industries due to its importance. The systematic review has followed the quality reporting guidelines set by the Preferred Reporting items for Systematic reviews and Meta-analysis (PRISMA) group. A review was carried out with three research questions in mind;

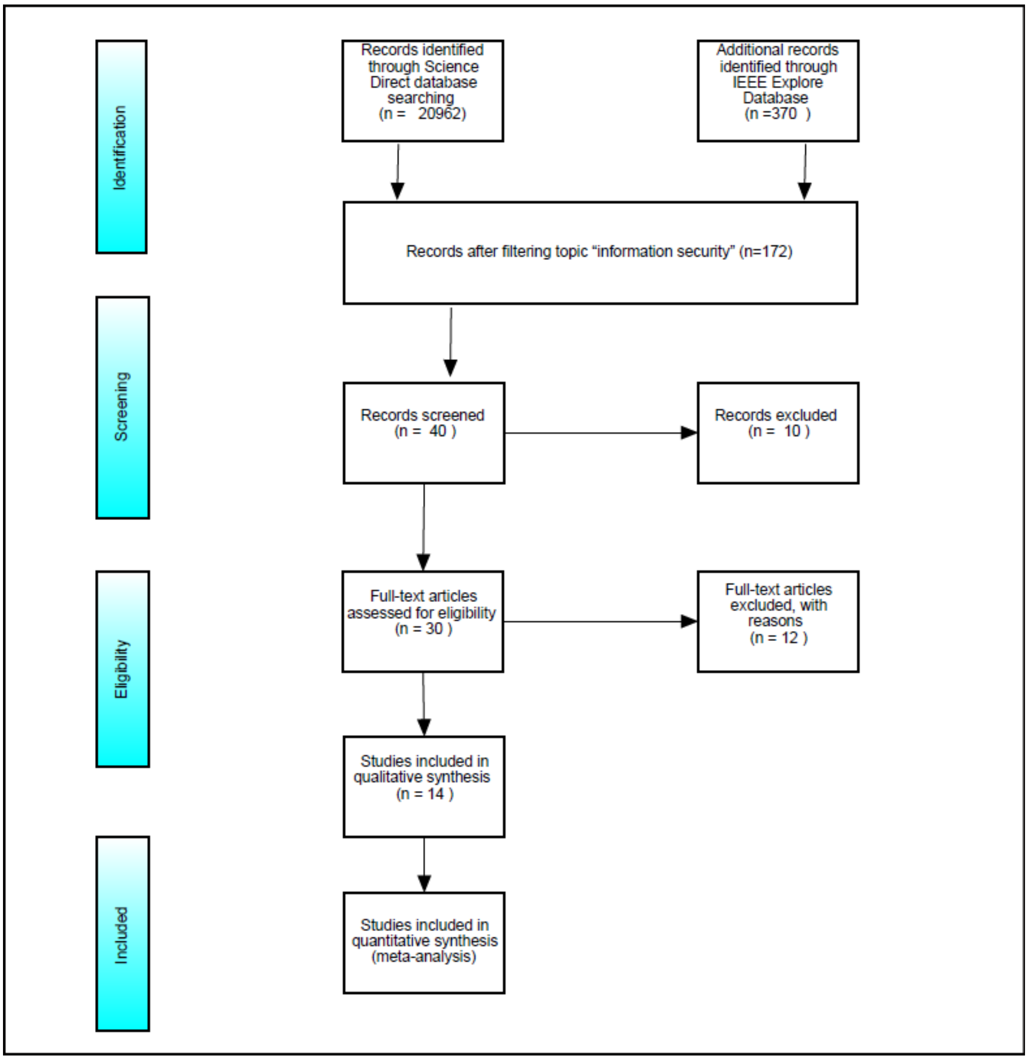

Identification
Screening
Eligibility
Included
Records identified through Science Direct database searching (n = 20962)
Additional records identified through IEEE Explore Database (n =370)
Records after filtering topic "information security" (n=172)
Records screened (n = 40)
Records excluded (n = 10)
Full-text articles assessed for eligibility (n = 30)
Full-text articles excluded, with reasons (n = 12)
Studies included in qualitative synthesis (n = 14)
Studies included in quantitative synthesis (meta-analysis)

Figure 3.2: Data Analysis approach through systematic review method

1. What is the intensity of research activity in framework/model/best practices for an IAM Solution in Information Security?

2. What IAM security functional taxonomy is being addressed in IAM development?

3. Which IAM Security taxonomy has been under-researched?

The literature review followed a systematic review process to ensure that the search and the retrieval processes were accurate and impartial.

3.2.1 Inclusion Criteria

The following inclusion criteria were used: articles that are written in English, articles related to the research topics: Identity access management, Identity access control within the information security and security management field. The search terms were applied to Google Scholar, IEEE explore, Science Direct, ACM Digital Library, with a date between 2010-2019, and the references included in the articles were also scanned to obtain to ensure the review is fully comprehensive.

3.2.2 Study Selection

The study selections were organized using the four phases:

Phase one - Research publication related to Identity Access Management and Access Control. This phase was searched using the string ("Identity access management") AND ("Access Control" OR "solution"), which was adapted to the search engine.

Phase Two - Exploration of Title, abstract, identified key words and selection based on the eligibility criteria.

Phase Three - Complete and partial articles that had not been eliminated were read to identify whether they were related to the eligibility criteria.

Phase Four - The reference lists were scanned to identify any further studies provided that met the eligibility criteria.

Excluded Publication - These were publications on the topics of Role engineering models, architectural details of IAM solutions.

3.2.3 Data Collection Process

An evaluation of articles revealed search engine Science direct and IEEE explore were the best sources for the purpose of the topic in this report; after the initial search 4 steps were followed:

- Query selection and search engine, initial search on identity access management revealed 20,962 articles in various sectors from the date ranging from 2010-2019.
- Manual refinement revealed that not all articles were related to the objectives of this report; some were identified as being related to social sciences, medicine, tourism, computers in human behaviour, IDM software development, and various role-based models of information security.
- Verification. The title, abstract and content of each article was checked in order to include or exclude based on the eligibility criteria.
- Classification of relevant publications. Classification of the publication was based on functional areas of security, using the following categories: data security, auditing, assurance, provisioning, compliance, policy and governance.

3.3 Research Phases

The research was conducted in six phases as shown in Fig 3.3

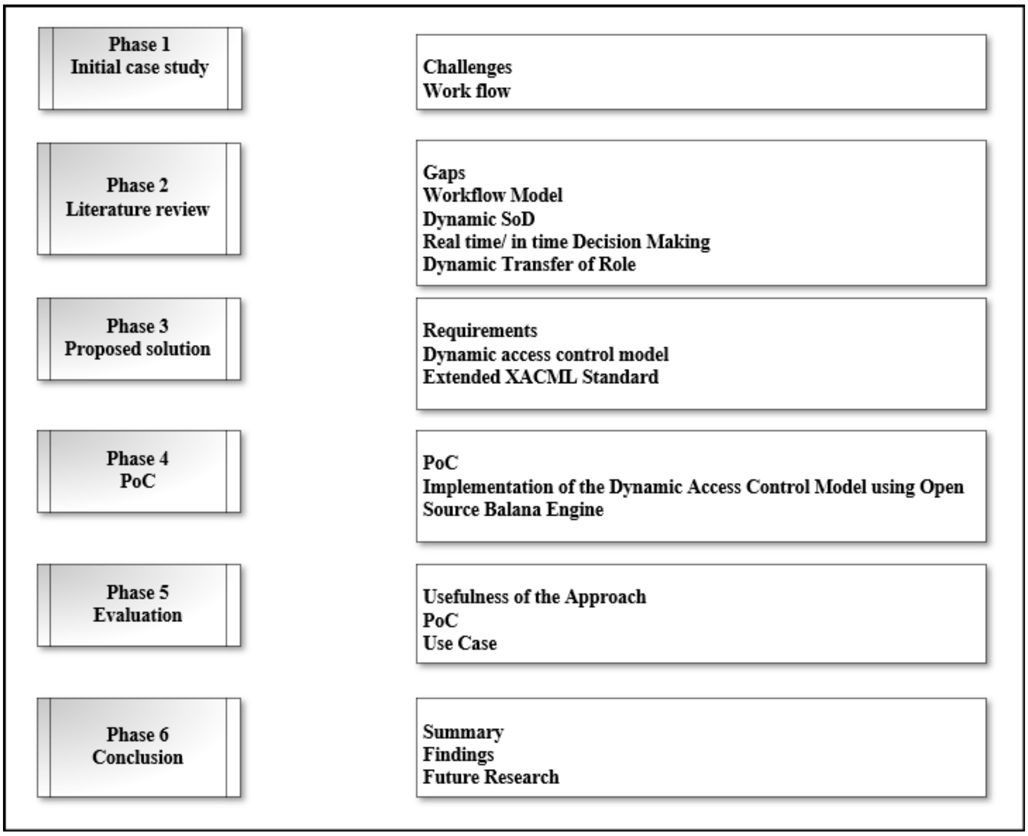


Figure 3.3: Research conducted in six Phases

3.3.1 Phase 1 Initial Use Case - Identifying the Challenges

Having worked as a consultant for various Investment Banks over ten years, noticed the challenges within the Access Management of complex banking applications setup, and observed various data breaches, this led me to choose the use case as part of my Professional Doctorate study to solve a real life challenge. Access Control Management has been a challenge in organisations and especially vital in banking industries due to high risk applications that support its investment decisions and operation. It is crucial that access to resources are permitted to authorised users only, and that access is granted in a timely manner. Access governance ensures that access to resources is authorised appropriately, and accountability and traceability is retained. Access control and access governance systems were in silos and it was necessary to have one streamlined process and system to support that process. Such a system should be dynamic, with intelligence to make a real-time decision based on policy rules which support both role-based and task-based instance restrictions to meet Binding of Duties and Segregation of Duties. An authorisation access control model was required, which could enforce role changes, ensuring old entitlements were revoked prior to granting new access, and ensuring access has been authorised and governed appropriately, whilst SoD & BoD constraints were retained. Case study has been appended in the appendix A.1, workshop has been held with key stakeholder, Information Security Dept Head, Security Administrator and Information Owner (Business approval director), to derive access governance process workflow.

3.3.2 Phase 2 Review of Literature - Identifying the Research Gaps and Limitations

Access control and identity models (such as such as SSO, Authorisation Model, and Authentication model) have received attention from a vast amount of research. However, despite this research into identity and access control both in academia and industry, identity access management remains challenging and the risk of unauthorised access to resources or exploitation of vulnerabilities due to escalation of privilege remain high-level threats for organisations. Although these gaps have been researched individually, they have never been considered as a whole to develop a new access control model.

3.3.3 Phase 3 Proposed Solution - The development of the Dynamic Access Control Model

The proposed solution AW-TRBAC (Authorising Workflow Task Role Based Access Control) has been developed using the requirements identified in the use case and the gaps in the literature review for such an access control model. AW-TRBAC used the concepts of task and role from the widely adopted access control model RBAC and ABAC. Although ABAC has received vast attention from academia, to date there hasn't been much adoption within industry. ABAC is a framework of concepts and has been used in various other access control models such as policy-based access control, and context-based access control. Another element in AW-TRBAC is the workflow task authorisation, which has received little attention within academia, however AW-TRBAC has been able to incorporate both the role and task concepts. AW-TRBAC has met the challenge to build a dynamic access control model which supports dynamic segregation of duties (instance level restriction), real time activation of permission (task instance activation), workflow authorisation (governance), binding of duties (restriction of roles) and timely decision-making enforcement based on the policy rules of the organisation through policy language (XACML).

To fulfil the policy enforcement a policy language was required, which supports the Role, Task, Operation notation. XACML was chosen as the policy language which would work well with the dynamic access control due to its cohesion and coupling ability, however, it was lacking Task notation capability. To meet the Task requirements and the dynamic capability of role change and SoD, the XACML standard was extended to support the additional functionalities. This extended XACML will add a Task authority function for task related queries and a role enabling function combined with two additional data stores which will be utilised during decision making and retaining audit trails.

The proposed solution has been implemented using open source XACML Engine (Chen et al. (2013)). It was extended using an additional five functions to be used as a policy engine to query an XACML request and retrieve information from the data store and compare against the policy before forwarding it to the Policy Decision Point (PDP) for decision making.

3.3.4 Phase 4 Evaluation –Usefulness of the Proposed Model

The evaluation of research findings is related to the extent to which the data can be generalised and how they are relevant and applicable to frameworks (Bryman (2016)), The evaluation will be carried out using the use case requirements to run end-to-end access request processes within the PoC system, and the results will be recorded to check whether they will align with the use case policy rules and the task constraints. Tests will be carried out using Java test scripts to run against the AW-TRBAC engine for decision making.

The evaluation will be in two phases: the first phase will test satisfaction of the use case requirements, and the second phase will test applicability in real life.

- The First phase will analyse the requirements by running a test script, which will record the results in a SQL Database and will use the SQL datastore to retrieve information in other test cases.

- The second phase to analyse and record the system performance; results from the tests will evaluated for applicability in real life.

3.3.5 Phase 5 Research Conclusion

The conclusion phase will summarise the findings and limitations within the thesis. It will summarise challenges, constraints, success and future work within the thesis and summarise the journey of the thesis to reach its conclusion. It will also propose future research into dynamic access control and how this could be used within the industry as a fully developed solution.

3.3.6 Limitations

Identity access management is very critical in every sector, including government, finance, healthcare, retail, and defence. As security functions are interrelated it was difficult to sepearate functions into a security taxonomy; however, the security functions have been grouped in this systematic review into IAM solutions, data security, provisioning, and compliance. Limitations within this exploratory systematic review are as follows:

- The search was conducted on various Databases and the choice of search strings may have led to accidental exclusion of relevant articles.
- Literature were limited to only English language literature, there may have been other non-English literature that was not included.
- Literature searches were manual and prone to error, there might have been literature which was incorrectly eliminated in the initial identification phase.
- Evaluation criteria used might not have been appropriate.

The limitations and disadvantages of the descriptive approach are:

- Descriptive studies cannot be used to correlate variables or determine cause and effect.
- Confidentiality can be an issue.
- Researcher bias may play a role in many ways. For example, the choice and wording of questions for the questionnaire may be influenced by the bias of the researcher, or subjective choices could be made about which information to record and emphasize in the findings.
- No variables are manipulated, therefore statistical analysis is not possible.
- The results are not repeatable and typically the study cannot be replicated. Findings may be open to interpretation.

3.4 Summary

The methods used to conduct this research is mixed method, an industrial use case workshop was setup to understand the process of access governance through workshop with the key stakeholder with framing question to connect the dots in a complex environment and to analyse the gaps. Several experts opinion survey report and security breach resulted due to access control adversaries were studied to understand impact of the gaps.

The systematic review has followed the quality reporting guidelines set by the preferred reporting items for Systematic reviews and Meta-Analysis (PRISMA) group. This is to ensure that the search and the retrieval process have been accurate and impartial. It is an approach that is typically used in medical clinical research where volume of data is large in quantity and needed to be narrowed down to specific niche area, similarly IAM field has received vast amount of interest both from academia and industries due to it importance in security and risk mitigation.

The next chapter is the main contribution of the thesis, which exhibit the characteristic requirements into the development of the AW-TRBAC dynamic access control model, it concepts and extension of the XACML Oasis Standard to meet the functionality requirements.

Chapter 4

AUTHORISING WORKFLOW-TASK ROLE BASED ACCESS CONTROL (AW-TRBAC)

This chapter presents the main contribution of the thesis, the dynamic access control model. In chapter two an overview of the use case and a critical analysis of existing identity and access control models has been studied. It has been observed there are several limitations relating to workflow governance and segregation of duties, especially at instance level. Several silo access management systems are required to maintain access control to resources.

This chapter exhibits the characteristics and requirements of a dynamic access control model. Then it defines the concepts that are used to develop a conceptual model of dynamic access control to bridge the gaps of dynamic SoD, governance and mitigates broken access control risk. Later in the chapter, the concepts of dynamic constraints have been proved mathematically using set theory for assertion on test scenarios.

Finally, in the chapter, rationalisation is provided for policy requirements and explain the XACML standard that has been extended to meet the requirements for policy enforcement of the conceptual model.

4.1 Motivation for Dynamic Access Control Model

After reviewing the use case of the investment bank in conjunction with a review of the access control model literature, it was apparent that there were certain aspects of access control that were not being studied or which did not have fully implemented solutions. This provided the motivation to propose a dynamic access control model which will provide real-time decision making intelligence.

RBAC is widely adopted access control model which supports passive access control and have incorporated session concept of its dynamic separation of duties, which has not fully met the requirement of for constraint within task instance enforcement that is required in dynamic environments (Oh and Park (2000)). For example, "issuer of a task" cannot be an "approver" for a task instance. However, it is possible for a "user" to be an approver for one instance and be an issuer for another instance; this level of enforcement is not possible within the role concept . The model of ABAC is a framework and can combine with other access control models. However, its flexibility is limited. DAC, MAC, RBAC, and ABAC all provide security control from the point of the user but can't achieve dynamic authorisation. Therefore, they are not suitable to meet the business workflow tasks constraints (JING and YANG (2006)). ABAC on the other hand separates access right assignment for users and access right activation. The ABAC model has limitations in the enterprise environment, ABAC does not deal with passive access control such as role to permission mapping. Therefore, additional access control methods need to be added to the ABAC model.

For an authorisation model to be able to work within a dynamic environment, it requires the ability to support a specific set of characteristics, of which instance-level restrictions, dynamic segregation of duties and binding of duties (BoD) are notable examples. This thesis contributes to addressing the limitations of both RBAC and ABAC models, with regards to tasks and the sequence of executing processes.

4.1.1 Requirements for Dynamic Access Control Model

From the analysis of the use case and the review of the existing access control models, primarily RBAC (Ravi Sandhu et al. (1996)) and ABAC (Biswas et al. (2016)), a set of characteristics and specifications have been derived for the functional requirements of the dynamic access control model. These requirements are necessary for 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 of 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 et al. (2001)), which are well-known and highly used in the security field. There are many variations of constraints of SoD, and despite various research approaches, it remains challenging to implement in a dynamic environment. This thesis will focus on requirements 4, 5, and 6, which are unique functional requirements for a dynamic access control model to meet a dynamic borderless network environment. Requirement 5, which allows for role changes, implies constraints of SoD, BoD, and revocation of access. These are integral requirements which have dependencies on the functionalities associated with the other requirements such as 2,3,4. This thesis focuses on the role change process and associated dependencies with the other processes.

4.2 Conceptual View of the Dynamic Access Control Model

After defining the requirements, next component of the model is the concepts necessary to define the AW-TRBAC. It is based on existing identity and access control concepts such as user, role, and permission and considers new concepts such as task, and IT workflow.

4.2.1 Conceptual View of the Dynamic Access Control Model

After defining the requirements, next component of the model is the concepts necessary to define the AW-TRBAC. It is based on existing identity and access control concepts such as user, role, and permission and considers new concepts such as task, and IT workflow.

4.2.1.1 AW-TRBAC Concepts

The concept "role" focuses on an actor, and "task" focuses on an activity, and therefore "task" is not sub-concept of "role". It is possible to group permissions by role and task but grouping permissions by role leads to role explosion. The key concept of AW-TRBAC is that each role has specific tasks assigned to it. AW-TRBAC extends the standard RBAC model by defining task elements and its relationship with role. With these extensions, AW-TRBAC can support the additional expressibility requirements and remain compatible with the RBAC standard. As in Fig 4.1 shows a high level abstract diagram of the two models: the standard RBAC model (bottom) and the AW-TRBAC (top). RBAC is used as a base and it uses the notions of role, user, permission and session. AW-TRBAC expands on the notion of tasks to support workflow active access control to provide more for dynamic access control behaviour, while retaining compatibility with RBAC. AW-TRBAC supports workflow tasks and dynamic role changes. Workflow tasks are distinguishing on the execution on the instances of the task via activation conditions and the task execution list. The function task execution list assists in the instance level restriction by using history-based information and provides an auditing function. It uses the workflow task notion to activate access rights using activation conditions, which ensure that priority tasks are completed in a sequence of tasks before activating access.

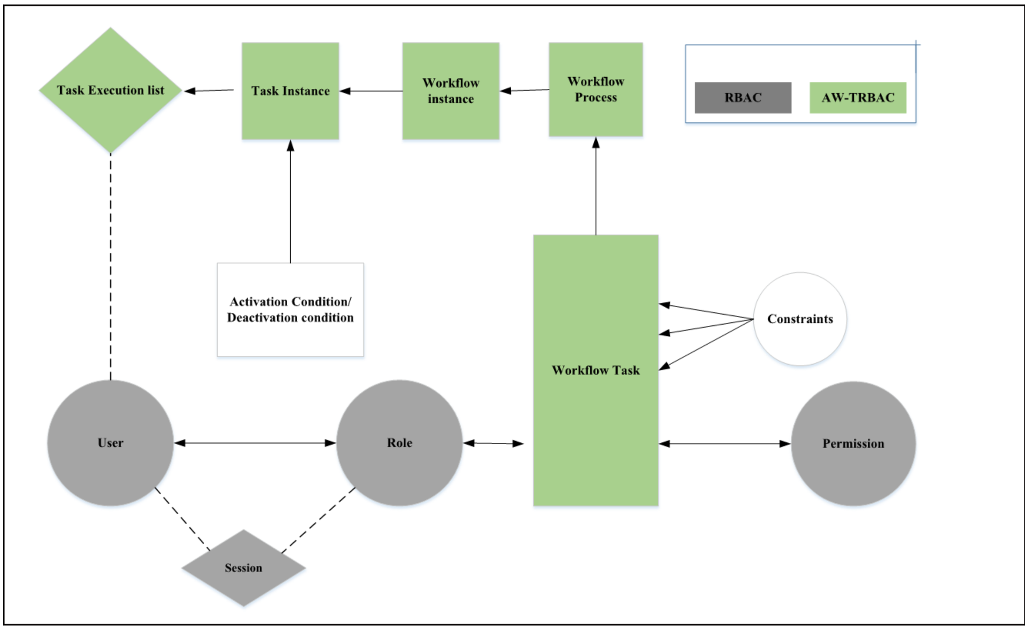


Figure 4.1: AW-TRBAC Dynamic Access Control Models shaded in green represent the notion of task to support workflow active access control for dynamic access control behaviour

A workflow authorisation model is important in information governance to keep track of all the activities involving controlled access to resources. This information can provide information security with a holistic view of who, what, how access to resources was authorised. Having such information supports an organization's immediate and future regulatory, legal, risk, environmental and operational requirements.

4.3 AW-TRBAC Conceptual View

The class diagram in Fig 4.2 that has been used in this thesis for the conceptual modelling of workflow authorisation models, describes the attributes and operations of a class and the constraints imposed on the system. The class diagram shows a collection of classes, associations, collaborations and constraints. Class diagrams are widely used in the modelling of object-oriented systems because they are the only Unified Modelling

Language UML diagrams which can be mapped directly with object-oriented languages.

User: Users are the subjects of 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 of a 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. From 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.

Workflow: This is an IT term describing a business process. In general, it means a product or method for supporting business processes in the enterprise environment. The task "approve customer orders" belongs to the process "receiving customer order". Execution of tasks in the business process should proceed in a defined order and take a defined amount of time. Although the task "approve customer orders" is assigned to the user, they can activate their access rights only when the prior tasks "check customer credit" and "check product stock" are completed. In this case, authorisation (access right assignment) is separated from activation of access rights. This type 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 the execution of tasks within workflow resources.

Business Process: This 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 organisational goal. A business process is a function of access control management in information security.

Execution List: An execution list is a record of all users who performed certain tasks, this will contain names, roles and tasks that have been performed by a user. It lists transaction logs of an event that has been actioned by a certain user, which can be used for incidence response root cause analysis and compliance. This is a critical control within information security for data analytic as well.

As shown in Fig 4.2 class model of AW-TRBAC, it shows the class user, which has a direct 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 has three generalised 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.

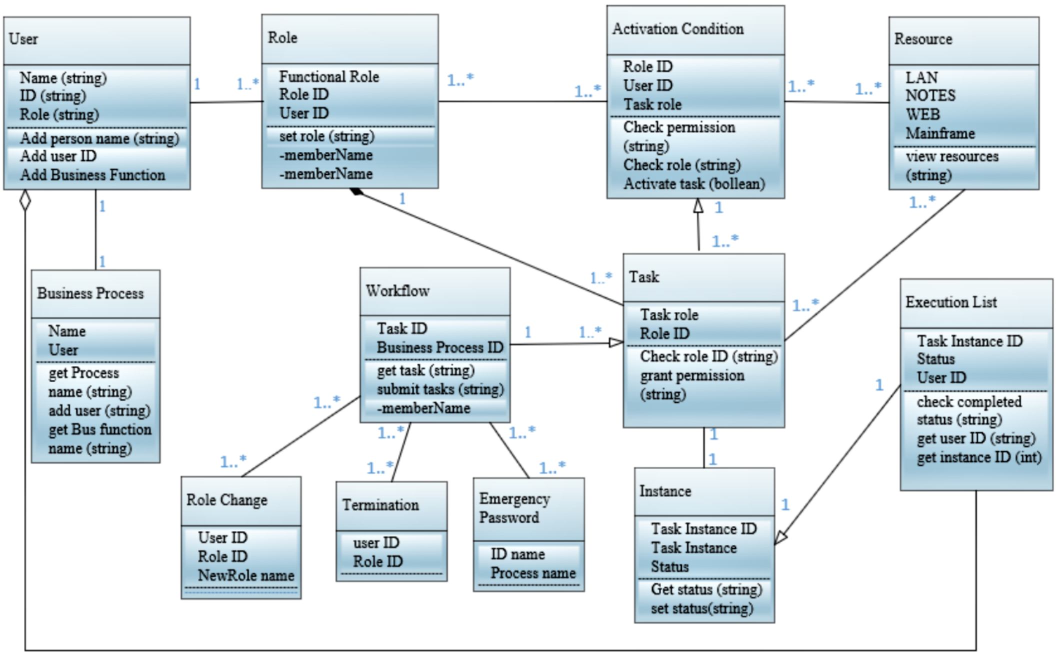


Figure 4.2: Static structure diagram which describes the structure of the AW-TRBAC system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects

4.4 Mathematical Model of Task Entailment Constraints

In this section of the thesis, the task constraints and task entailment constraints are analysed against the use case requirements using theory algorithms.

4.4.1 Task Constraints

Firstly, task approval constraints are described based on binary relations defined on the set of users. Such relations are expressive, intuitive and can be manipulated algebraically, enabling to derive new constraints that simplify the analysis of workflows.

$$O = \{o\}, A = \{a\}, \text{ and } P = \{o \cap a\} \tag{4.1}$$

Let "O" denote the set of all objects subject to access control, "A" the set of all actions that can be performed on those objects, and "P" the permission, is the set of all OBJECTS and ACTIONS:

Set Theory Label	
"O"	Denotes the set of all objects subject to access control
"A"	The set of all actions that can be performed on those objects
"P"	The permission
"U"	Denote all the authenticated subject users
"R"	Is the role, function
"URA"	User role assignment is many to many user-to-role assignments
"S"	A list of all sessions
"TY"	Task type
"RTA"	Role-to-task permission
"T"	Is the set of all tasks in the system
"TPA"	Task-to-permission
"W"	Denotes workflow
"AC"	Denotes activation condition
"BW"	Denotes the business workflow
"AU"	Auditing of task instance
"PI"	Denotes a process instance
"RTA"	Role-to-task permission
"TI"	Task instance
"DC"	Termination of account
"IR"	Denote an instance level restriction

Table 4.1: Set Theory Label

Let "U" denote all the authenticated subject users and "R" is the role, function, or position, that somebody has, or is expected to have, in an organization. "URA" user role assignment is many to many user-to-role assignments

$$\{u\}R = \{r\} \text{ and } URA \subseteq UxR. \tag{4.2}$$

Let denote "S" a list of all sessions, it is a function that return all activated roles for a user. A user can only activate assigned roles thereby an activated role (session) is a subset of all assigned roles=s,

$$s(u) = \{ \text{ active session for the user u} \} \text{ , } s(u) = \{r\} \text{ and } \{r\} \subseteq (R \cap r \text{ active for user u}). \tag{4.3}$$

A user can obtain permission through activating role u=uc, s(uc) $\subseteq$URA and s(u) $\neq$.

Let "TY" denote Task Type "W" Workflow tasks and "N" NON-WORKFLOW tasks. "T" is the set of all tasks in the system, a unique task "ID" which is Natural no, "p" is the permission and task type.

$$\begin{gathered} TY = \{ty\}, W = \{w\}, N = \{n\}, ty = \{w \cap n\}, ID = \{N\}, \\ P = \{p\}, T = (TY \cap N \cap P) andt = \{ty \cap n \cap p\}. \end{gathered} \tag{4.4}$$

"TPA" task-to-permission, which is many to many mapping and "RTA" role-to-task permission, which has many to many mappings.

$$TPA \subseteq T \cap P, RTA \subseteq R \cap T, r \in s\{u\} \text{ and } \{t \cap r\} \subseteq RTA \tag{4.5}$$

Proposition 1. Let "BW" denote the business workflow, it contains the name "N" of the Business workflow and "T", the task. TPA task-process-assignment is one-to-many mappings.

$$BW = \{bw\}, bw = (n \cap t) \text{ and } TPA \subseteq T \cap bw. \tag{4.6}$$

Let "PI" denote a process instance of a business workflow; process instances can be executed many times for a business workflow. "TI" denotes task instances for this particular process instance. TI have "ST" (status assignment active, unassigned, and completed).

$$PI = \{pi\}, pi = (bw \cap n), TI = \{ti\} \text{ and } ti = \{t \cap st \cap n\}. \tag{4.7}$$

"AU" is the auditing of task execution, and is a function mapping between completed task instances and the user who performed it.

$$AU : ti \to u \subseteq Uandti \in TI \cap ti(st) = completed \tag{4.8}$$

Proposition 2. Let "AC" denote the activation condition task which will activate if satisfied and not activate otherwise; it can take the values TRUE or FALSE.

$$\begin{aligned} AC : ti \in TIv \in \{true, false\} andAC(tin) = true \text{ iff } St(tin-1) = completed. \\ 'DC' \text{ Termination of account DC} : ti \in TIv \in true, False \text{ and} \\ AC(tin) = true \text{ iff } St(tin-1) = completed. \end{aligned} \tag{4.9}$$

SoD Segregation of Duties, NIST RBAC identifies pair of roles cannot be assigned at the same time (static), r1, r2 ∈ R,

$$SoD - type \in static, dynamic and SoD = (r1, r2, sod - type), \tag{4.10}$$

SoD and a pair of roles cannot be activated both session simultaneously (dynamic) if r1 ∈s(u) → r2 ∉ s(u).

Proposition 3. Let "IR" denote an instance level restriction. There are two types of instance level restriction, segregation of duties (SoD) and binding of duties (BoD), These segregation are at the instance level as opposed to role level in RBAC.

"AU" connects the task-instance with the user who performed the task for that specific instance, before allowing the user to activate a 'ti' task instance. The AW-TRBAC authorisation engine checks the 'ir's that apply to this 'ti'. It then uses the AU function to identify if the requesting user would violate an 'ir' instance restriction for those tasks that have SoD restriction on the process i.e. issuer and approver, if the task were performed by them.

$$\begin{aligned} ir \in IR, type \in \{SoD, BoD\}, ti1 \cap ti2 \in TI, ir = (ti1, ti2, type) \\ \text{and } ir = (t1i, t4j, SoD) \rightarrow u \in AU(t1i) \text{ and } u \in AU(t4j) iff i \neq j \end{aligned} \tag{4.11}$$

4.4.2 Task Entailment Constraints Within the Workflow

This section depicts the task constraints within the workflow, to activate certain permissions to perform a task, it requires completion of previous tasks within the workflow. Using set theory algorithms to describe the task entailment within the requirements identified, for the dynamic access control model proposed in this thesis.

- Access request can only be submitted by the "Security Coordinator". (Security request, coordinator) ∈ TRA.

- Security requests can only be approved by the authorised "Managers".

$$\begin{aligned} (Ir = (Securityrequest, Approve, SoD)), \\ \text{If u } \in AU(Securityrequest) \rightarrow u \not\subset AU(Approve) \end{aligned} \tag{4.12}$$

- Only requests approved by the approving managers can obtain access to resources.

$$\begin{aligned} AC(accessrights) = true\ iff(st(Securityrequest) = completed) \cap \\ (Security\_request\_approval) = completed. \end{aligned} \tag{4.13}$$

- Termination of employability requires revocation of access.

$$\begin{aligned} AC(Termination\_access\_rights) = true \\ iff(st(Termination\_request\_approval) = completed). \end{aligned} \tag{4.14}$$

- A transfer request requires two levels of management approval: from the department the user is transferring from and the department they are transferring to.

$$\begin{aligned} \text{AC (Role change access rights) = true if (st (Transfer request)} \\ = completed) \cap (transferrequestapproval1) = completed \cap \\ (transferrequestapproval2) = completed \end{aligned} \tag{4.15}$$

- Closing request requires completion of security request and access rights granted/revoked

$$\begin{aligned} AC(closeSecurityRequest) = trueiff(st(Securityrequest) \\ = completed \cap st(accessrightsgranted) = completed). \\ AC(closeTerminationRequest) = trueif(st(Deactivateaccessrights) \\ = completed \cap st(Terminateusersession) = completed). \end{aligned} \tag{4.16}$$

- Security Coordinator cannot be a security administrator.

$$\begin{aligned} (SoD = (coordinator, Administrator, static)). \\ if(Administrator, u) \in URA \cap (coordinator, uc) \in URAu \neq uc \end{aligned} \tag{4.17}$$

- Emergency password issuer and the checker should be Security Administration team.

$$\begin{aligned} (Ir = (issueEmergencypassword, check - inEmergencypassword, BoD) \\ AU(issueEmergencyPassword) = \{SecurityAdministrationTeam\} \end{aligned} \tag{4.18}$$

4.4.3 Analysis of Use Case Requirement Constraints

This section revisits the use case to analyse the requirements using set theory algorithms. A set of case scenarios have been constructed to mathematically test the constraints.

First test case: Termination of access rights assume a change role request received from HR, security coordinator “Amy” would like to log the request.

Proposition 1. The first thing the system will do is check to see if Amy can log a request (she has a Security coordinator role assigned to her):

$$(Amy, coordinator) \in URA \tag{4.19}$$

Is there violation of SoD rules by activating this role? If not, the system will then allow Amy to activate this role,

$$s(Amy) = coordinator? \tag{4.20}$$

After receiving the notification of a Change Role (transfer) request through the email, Amy (as a coordinator) wants to submit “Transfer request”.

$$(t(id) =) forinstancenumbere.g.1(n = 2) \tag{4.21}$$

First the system identifies if this role can perform this task.

$$(Transferrequest, coordinator) \in TRA. \tag{4.22}$$

The system then identifies the instance number and knows that Amy wants to perform a task-instance. It will also check the IR restriction to show that status is “unassigned” and no one has performed this task instance.

$$The = (unassigned, Transferrequest, 2).Ir(TerminationRequest) = null). \tag{4.23}$$

The activation condition of this task instance is ‘Transfer e-mail received’

$$AC(TransferRequest2) = trueiffst(Transfere - mailreceived) = completed, \tag{4.24}$$

The Activation condition of the task ‘Role change access rights’ will only be true if the task ‘Transfer_request_approval’ has been completed. Otherwise the Transfer Request condition will be false, and no one can perform the task yet.

Proposition 2. Role Change (within the same department): User remains in the same department, but moves on to a new business function.

$$\begin{aligned} AC(Rolechangeaccessrights) = trueiff(st(Transferrequest) = completed)\cap \\ (transferrequestapproval) = completed \end{aligned} \tag{4.25}$$

Complex role change (departmental transfer). Departmental transfer requires two levels of approval, the first approval from the department that the user is transferring from shows that the user is leaving this department, the second approval from the department that the user will be transferred to is an official approval that user will be working in this new department.

$$\begin{aligned} AC(Rolechangeaccessrights) = trueif(St(Transferrequest) = completed)\cap \\ (transferrequestapproval1) = completed \cap (transferrequestapproval2) = completed \end{aligned} \tag{4.26}$$

As the restrictions state that it should not be the same person to submit a request and approve it,

$$(ir = (Securityrequest, Approve, SoD)). \tag{4.27}$$

The system will check the Auditing execution list of the task ‘Transfer request’:

$$AU(TransferRequest) = Amy \tag{4.28}$$

Activation condition for the Closing Transfer request would only be true if both tasks, role change access rights activated, and existing access rights revoked, are completed.

$$\begin{aligned} AC(closeTransferRequest) = trueiff(st(RoleChangeaccessrights) = completed\cap \\ st(existingaccessrightsrevoked) = completed). \end{aligned} \tag{4.29}$$

Proposition 3: Security request to be closed by the Coordinator Closing a security request follows the rule of instance level restriction, and the Security Coordinator who submits the request will need to close the request.

$$(Ir = (submitrequest, closetherequest, BoD)AU(submitrequest) = Coordinator, \tag{4.30}$$

Similarly, for Privileged access management

$$\begin{aligned}(Ir = (issueEmergencypassword, check - inEmergencypassword, BoD)\\ AU(issueEmergencyPassword) = SecurityTeam\end{aligned} \tag{4.31}$$

4.5 Policy Specification Requirements

When data is flowing in the workflow, the user performing the task is changing and the user's permissions are changing too. This is related to the context of the data processing, due to the characteristics of the workflow system, the workflow is not only to correctly simulate the steps of the execution, but also to properly simulate rules to be followed and constraints maintained during the execution of the business. An authorisation policy language which can provide how access control policies are expressed in a manner that can be enforced in an information system. One authorisation policy language that has become widely used and accepted is the extensible Access Control Mark-up Language (XACML) (Leitner et al. (2011)), an XML defined standard language for authorisation policies. (Celino et al. (2007)) showed that XACML by itself is not enough to support all types of authorisation models. The XACML standard has been further extended to incorporate the Role notion to support RBAC policies; this extended XACML is known as "XACML-RBAC" (Celino et al. (2007)). Neither XACML nor the XACML-RBAC standard can accommodate the notion of tasks or task instances, therefore instance-level restrictions are not supported. This motivated me to extend the XACML de facto standard to support workflow processes. There is currently no published work or implemented to the knowledge of the author that extends the XACML language to support authorisation policies for IT workflow processes.

4.5.1 Access Control Policy Enforcement

As stated previously, this research extends the XACML standard to support the implementation of dynamic access control, to meet the use case requirements and enforce the rules of the dynamic access control model through policy language. XACML supports the notation of the proposed dynamic access control model, such as role, task, and operation, so that it can act as policy enforcement, which interacts with the access control model to make decisions. The focal point of this research is on dynamic role changes, SoD, BoD functionality and security requirements to enhance the risk posture and visibility. To satisfy the use case requirements, five new functions and two new data stores have been introduced: SoD check, BoD check, Role check, Role change check and Role change approve checks, which are utilised by the XACML policy engine in the decision making. These extended functions enable the dynamic access control model to provide real time history-based instance-level segregation to mitigate the risks of broken access control and insufficient logging of events. XACML is an OASIS standard that defines a general-purpose access control and authorisation system (Rissanen et al. (2013)). 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 whereby each policy lists the requirements to access a resource in a protected environment.

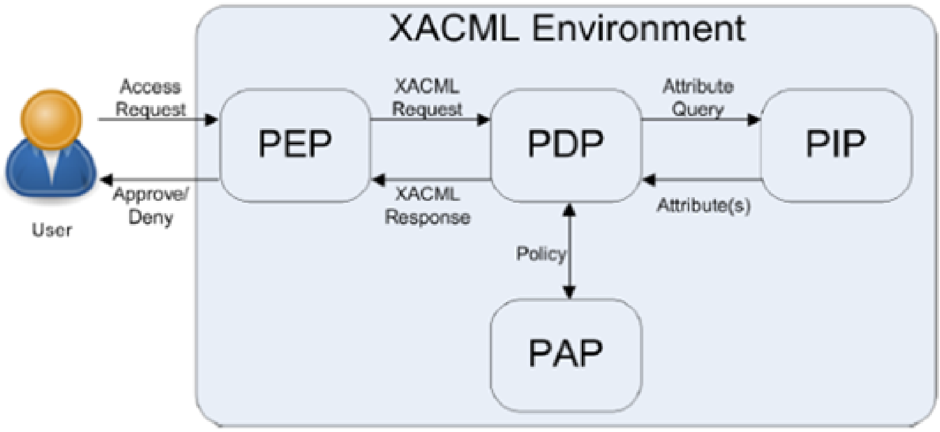


Figure 4.3: XACML Architecture and its interactions with various components PEP, PDP, PIP and PAP for a request evaluation OWASP (2017)

As depicted in Fig 4.3, 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.

4.5.2 Extension of XACML Standard

This research extends the RBAC XACML OASIS standard and introduces two new repositories 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 check (coloured in blue) in Fig 4.4. Each function is utilised for a different security request, for example SoD Check will be utilised for requests that require segregation of duty constraints on the submitter and approver roles; functions also contain conditional obligations to enforce policy rules.

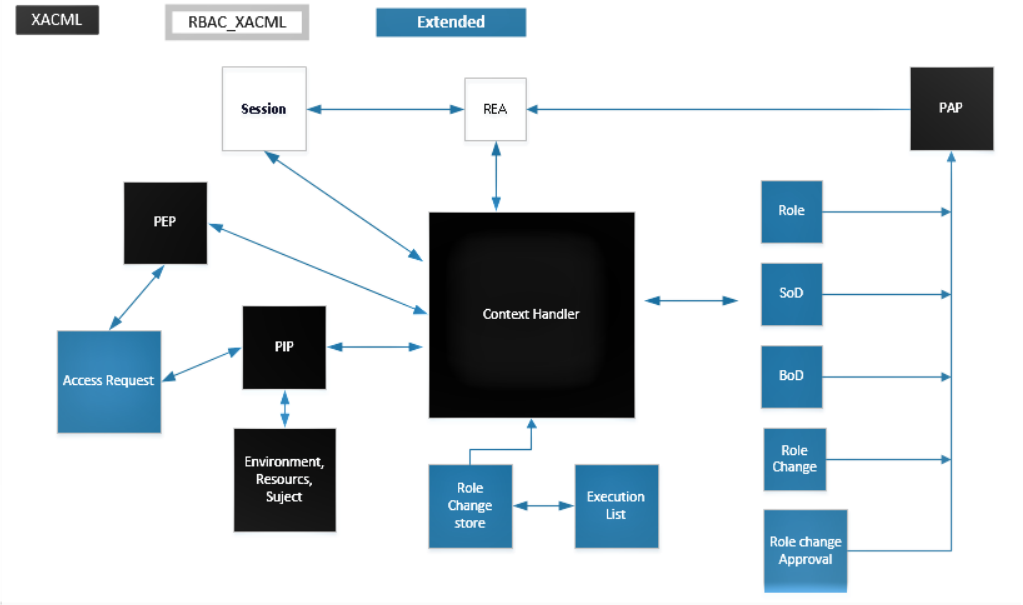


Figure 4.4: Showing Extended XACML Standard, shaded in blue components developed to meet AW-TRBAC dynamic access control behaviours

The context handler in Fig 4.5 is responsible for translating received requests into the XACML context and translating the results back to the native language of the other system. It is also responsible for communicating between the other components. In XACML the Policy Decision Point (PDP) is responsible for making decisions on the authorisation requests based on the policy sets. With RBAC-XACML (OWASP (2017)) there is a new type of request that deals with role activation; it was decided that role activation should be out of the scope of PDP. For this reason, the Role Enablement Authority (REA) was introduced as part of the standard development. REA is a specialised repository that will have a policy store to support the decision making in role activation. AW-TRBAC has a new type of request, to perform a workflow task. To deal with such a request, new functions have been added to provide input to the PDP for decision making.

As shown in the Fig 4.5, a sequence of events of task execution and authorisation are required before a decision is made in response to a request, and the outcome is access to authorised resources. This sequence of event needs to be executed in an orderly manner and a just in time decision need to be made based on the evaluation of various

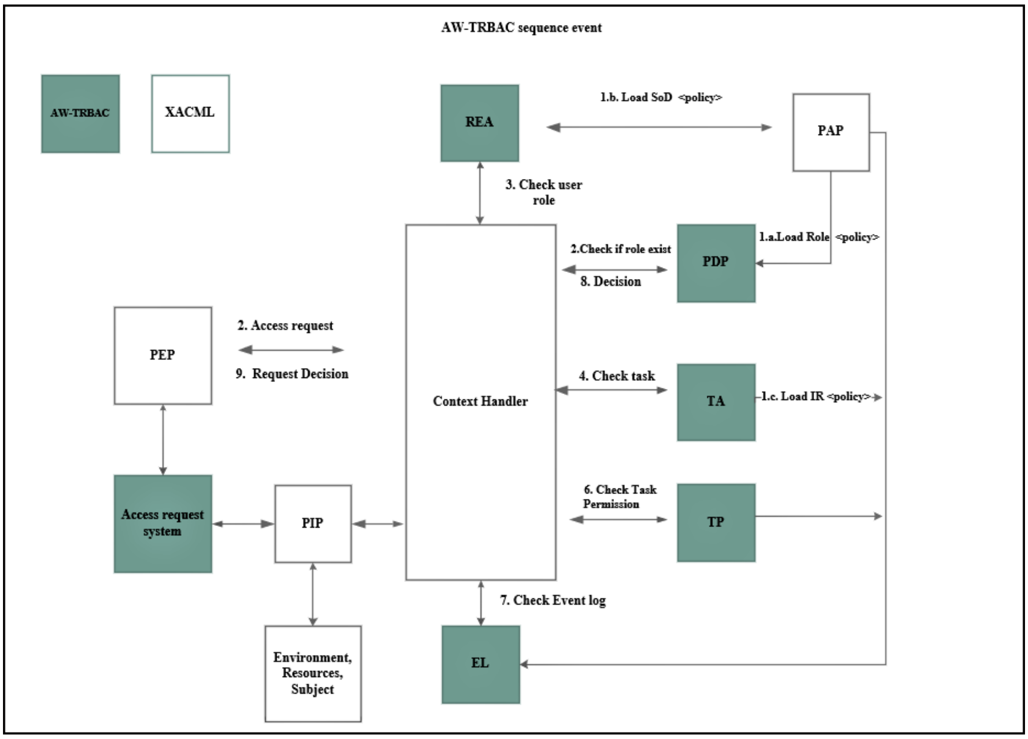


Figure 4.5: A sequence of events of task execution and authorisation are required to be executed in an orderly manner and a just in time decision need to be made based on the evaluation of various components of XACML in AW-TRBAC access control

components of XACML for the dynamic access control to function as required. PAP loads the SoD Policy set by the REA. When Context Handler (CH) receives a request for role activation from the PEP, it will forward the request to the Role function to query on SoD function to retrieve rules related to this request in Fig 4.5. In parallel to this, CH will query the Role function to get the user's active roles and will forward the information to the REA and a decision will be made based on evaluation of the SoD policy set and the Role Task policy set. the decision will be sent to the CH which will forward it to the PEP. CH will also update the user's role to add the new role if it was activated in the Role data store. The task activation sequence proceeds as follows in Fig 4.5. The Role Policyset will be loaded into the PDP and the IR Policy set by the TA Authority via Policy Administration Point (PAP). When a task activation request is sent to the CH by PEP, it will forward the request to the TA to check instance-level rules for any related instance-level restrictions. CH will use the Executioner List (EL) data store to get the historical information about the user completed the task instance. After retrieving this information, CH will forward it to the TA to decide on the request. The decision will then be sent to the CH. If the decision was 'deny', then CH will send 'deny' to Policy Enforcement Point (PEP). If it was "allow", then CH will query the Role Task repository to get all the permissions related to this task. CH will create a resource request for each permit and send it to the PDP. Finally, a decision on the permit will be made using the combining algorithm "deny Override will be sent to the CH", and the final decision will be sent to PEP. CH will also update the Executioner list to add the new activated task instance

Task activation sequence entails in Fig 4.5, Role Policyset will be loaded to the PDP and the IR Policy set by the TA (Task Authority) via PAP. When a task activation request is sent to the CH by PEP, it will forward the request to the TA to check Instance level rules for any related in-stance-level restrictions. CH will use the Executioner List (EL) data store to get the historical information about the user completed the task instance. After retrieving the information CH will forward the information to the TA to decide on the request. The decision will then be sent to the CH If the decision was 'deny' then CH will send 'deny' to PEP. If it was "allow" then CH will query Role-Task repository to get all the permissions related to this task. CH will create a resource request for each permit and send it to the PDP. Finally, a decision on the permit using the combining algorithm "deny Override will be sent to the CH" and the final decision will be sent to PEP. CH will also update the Executioner list to add the new task instance activated.

4.5.3 Structure of XACML Policy Request

Requests from various users are sent to the policy engine to be authorized by one or more policies. The 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, attribute and action.

Subject: A subject is defined as the user (whom the request originated from) and is implemented in XACML as User.

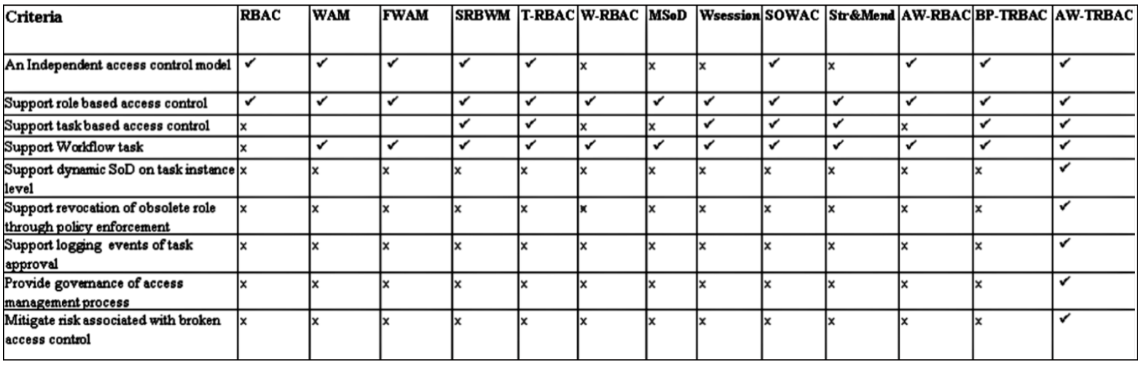
Objects: Objects are expressed using XACML Resources such as files, or web services. Operations are ex-pressed using XACML Actions.

Permission: Permission gives the ability or right to perform some action on some resource, possibly only under certain specified conditions.

Attribute: In this Profile, the term "attribute" refers to an XACML <attributes>. 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 Attribute), the protected resource (Resource Attribute), the action to be taken to the resource (Action Attribute), or the environment of the Request (Environmental Attribute). Attributes are referenced in a policy by using an <AttributeSelector> (an XPath expression) or one of the following:<SubjectAttributeDesignator>, <Re-sourceAttributeDesignator>, <ActionAttributeDesignator>, or <EnviornmentAttributeDesignator>.

4.6 Comparison with Another Workflow Model

Other authorisation models have been compared to AW-TRBAC in Fig 4.6. The criteria used for comparison of AW-TRBAC with other authorisation models are gaps identified in the development of the access control model in Fig 4.6. AW-TRBAC is a unique authorisation model which supports dynamic access control model characteristics such as dynamic SoD, governance and mitigation of broken access control risk. AW-TRBAC is an independent model that has policy enforcement components to support access control in a dynamic environment, such as a de-parameterised environment that requires access to networks from various endpoints. It has the ability to restrict access based on dynamic role changes through policy enforcement using a policy engine and a data store. This ensures that users have access to resources based on the most up-to-date role assigned to them, preventing escalation of privilege; this is an enhancement in comparison to all other access control models. Additional focus has been on risk mitigation through activity logs and accurately restricting sequences of task activation based on task constraints, ensuring tasks are followed through in a sequence. This has provided governance of the access management, which was identified as lacking in the literature review. With its architecture based on the XACML standard, AW-TRBAC is designed to form the basis of an enterprise-wide access control system that can integrate with existing architecture and applications.



Criteria	RBAC	WAM	FWAM	SRBWM	T-RBAC	W-RBAC	MSoD	Wsession	SOWAC	Str&Mend	AW-RBAC	BP-TRBAC	AW-TRBAC
An Independent access control model	✓	✓	✓	✓	✓	x	x	x	✓	x	✓	✓	✓
Support role based access control	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Support task based access control	x			✓	✓	x	x	✓	✓	✓	x	✓	✓
Support Workflow task	x	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Support dynamic SoD on task instance level	x	x	x	x	x	x	x	x	x	x	x	x	✓
Support revocation of obsolete role through policy enforcement	x	x	x	x	x	x	x	x	x	x	x	x	✓
Support logging events of task approval	x	x	x	x	x	x	x	x	x	x	x	x	✓
Provide governance of access management process	x	x	x	x	x	x	x	x	x	x	x	x	✓
Mitigate risk associated with broken access control	x	x	x	x	x	x	x	x	x	x	x	x	✓

Figure 4.6: Comparison of AW-TRBAC with other authorisation models based on gaps identified in the development of the access control model

4.7 Summary

This is the first and second contributions of the thesis. First Contribution, The development of the AW-TRBAC model to solve an industrial access control problem. It's characteristics that have been designed, based on the gaps in the literature and expert opinions derived from the various industrial surveys. The conceptual model has been tested mathematically through set theory algorithm and shown to prove the constraint requirements.

The Second contribution of this thesis is the extension of the XACML standard to meet the additional functional requirements, addition of the customised functions such as SoD Check, BoD Check, Role Check, Role change check and Role approve check. To the author's knowledge there has been no such extension exist to date and it first of it's kind.

Finally AW-TRBAC have been compared against the existing access control models based on the criteria which are the gaps identified in the development of the access control model, to illustrate it's unique characteristics and functionality toward the contribution to the knowledge in dynamic access control model to mitigate risk.

Next chapter describes the design & implementation of the AW_TRBAC Policy engine that has been developed to enforce the policy constraints through custom functions that have been implemented using open source engine to meet the requirements of complete AW-TRBAC dynamic access control model.

Chapter 5

DESIGN AND IMPLEMENTATION OF AW-TRBAC MODEL

So far in the chapter four, a dynamic access control model has been defined for the on premise, which can be used for cloud deployment based on Workflow Task Access Control. This work has been extended using XACML-based policy language that can express the requirement as policies and can validate the access control model using concepts of IT workflow task. This chapter expands all this work by implementing an AW-TRBAC policy engine, which provides the necessary APIs for integration of authorising model and policies.

This chapter presents the architecture, design and implementation of the AW-TRBAC, which is designed to be scalable and distributed in nature. It also provides a high-level overview of the implementation of the solution, and its integration with the Access Control Service.

5.1 System Architecture

XACML architecture is distributed in nature, less dependencies (low coupling) and reduced module complexity (high cohesion), which makes a viable solution to use as the base for our task-based access control architecture. Figure 5.1 shows the AW-TRBAC architecture, which extends the five core components of the XACML architecture to support Instance restriction for enforcement of use case logic.

To facilitate the additional functionality capabilities, dynamic SoD and IR (instance level Restriction), we are extending the XACML framework with five new functions: Role Check, Check SoD, Check Bod, Role Change, Role Approval and two new data stores Role Change and Executioner List. The PDP functionalities are extended by using Context Handler to query the additional functions which then forwarded to PDP for decision making.

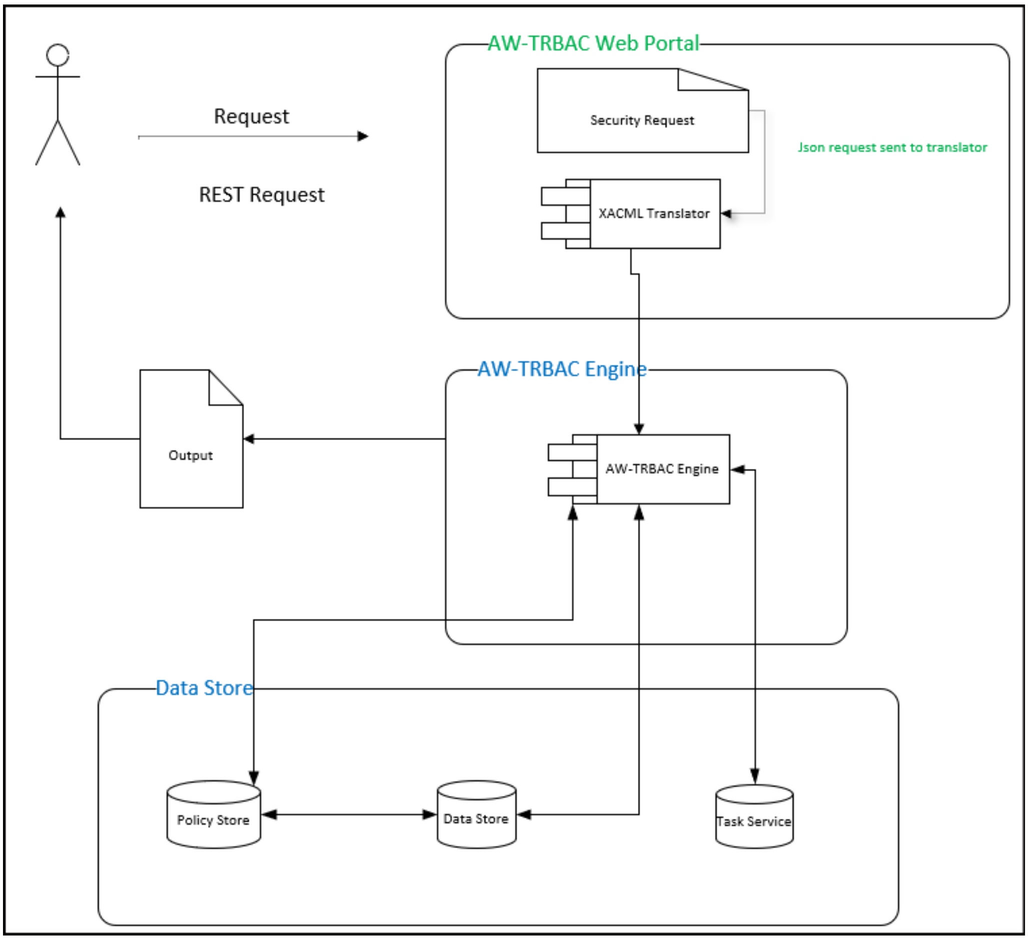


Figure 5.1: Shows the AW-TRBAC architecture, which extends the five core components of the XACML architecture to support Instance restriction for enforcement of use case logic

The choice of technology implementation is crucial for the adoption of the framework of the industry. It is paramount to use mature technology and widely adopted standard to reduce the complexity and ease of integration. A model of AW-TRBAC has been designed to illustrate the illustrate the framework proposed for access control model. The architecture in Fig 5.1 various components of the AW-TRBAC. Each component is discussed in turn.

In this thesis, the Web portal is the main entry point to the AW-TRBAC using REST API request which is then processed by extracting various information from the 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 which then executes the AW-TRBAC engines (extends the 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 a service.

5.2 System Design

This section explains the design of the AW-TRBAC policy engine, which entails various API interfaces between AW-TRBAC portal and XACML translator, AW-TRBAC engine and the Database connectivity. API connectivity that will be implemented is with REST API and the Connectivity API which will be Java Spring Framework (Perez et al. (2019)).

5.2.1 Access Portal API

This is the initial point of access to resources in invoking various authorization requests. The invocation is made through a series of REST calls. There is a total of three REST URI's for the portal: Role_change, SOD and BOD.

5.2.1.1 Role Change URI

This section describes the URI for the Role Change Request within the AW-TRBAC portal, for illustration purpose this thesis explains the Role Change Requirements design below and subsequent requirements appended in Appendix A.

The URI for the role change is as follows:

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

The following operations are supported: **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
```

The request body contains a JSON document in Table 5.8, that consists of: UserId, RoleId, TaskId and ResourceId. These values are extracted and used by the AW-TRBAC engine to validate the request.

Response:

A successful response would return a 201 HTTP response as shown in Table 5.2, with JSON body containing the TaskInstanceid and status. An unsuccessful response would contain HTTP 400 code.

Request Parameter	
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

Table 5.1: REST - Role Change Initial Request, the URI for the Role Change Request within the AW-TRBAC portal

Response Parameter	
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

Table 5.2: REST- Role Change Initial Response extracted and used by the AW-TRBAC engine to validate the request from the value of UserId, RoleId, TaskId and ResourceId

An example of REST HTTP Request/Response for Role change, as shown in the Fig 5.2 "Bob" is making an initial role change request.

5.2.2 First Manager Approval REST URI

This section describes the URI for the Role Change Request within the AW-TRBAC portal, for illustration purpose this thesis explains the Role Change Requirements design below and subsequent requirements appended in Appendix.

The URI for the role change is as follows:

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

The following operations are supported: 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/first_approval
```

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

{ "request": {

               "UserId": "Bob",
        "TaskInstanceid": "t001",
        .....
              }
}

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

Figure 5.2: A Rest HTTP Request/Response for a role change

Request Parameter	
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
	RsourceId - Unique service/object id
Request format	JSON
Action	POST / awtrbac/portal/auth/task/rolechange/first_approval

Table 5.3: REST - Role Change First Approval Request the URI for the Role Change Request within the AW-TRBAC portal

The first_approval is only executed after successful role change initial request. The first_approval REST request also has a JSON body, which takes the UserId, RoleId, Taskid and RsourceId as inputs. The request is submitted as a HTTP POST.

Response:

A HTTP response of 201 or 400 is returned depending on the outcome of the response. For a successful response, a JSON body with TaskInstanceid and status is returned.

Response Parameter	
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

Table 5.4: REST - Role Change First Approval Response is executed after successful role change initial request, AW-TRBAC engine validate the request from the value of UserId, RoleId, TaskId and ResourceId

5.3 Dynamic Role Change

Dynamic role change is another requirement identified during characteristics and requirement analysis of Dynamic Access Control (AW-TRBAC). In addition to SoD check as tested above, Role change requires additional conditional statements, it requires two levels of verification firstly terminating the existing role then provisioning the onboarding role. Second level verification is the governance, which is approved by the departing role manager and onboarding role manager. In both verification simulation will be using the below scripts, policies, functions, data stores and security request submitted via REST API.

5.3.1 Second Manager Approval REST URI

As shown below operations within the "Second Manager Approval" request via the REST end-point.

```
http://profdoc.uel.ac.uk/awtrbac/portal/auth/task/rolechange/second\_approval
```

Request Parameters

Request Parameter	
Resource Information	Description
Operation	Second_approval
Request Body	UserId - Unique identifier of a subject (user)
	Taskid- Unique identified of an IT workflow
	TaskInstanceId - Unique identifier of task instance
Request format	JSON
Action	POST /awtrbac/portal/auth/task/rolechange/second_approval

Table 5.5: REST - Role Change Second Approval Request, the URI for the Role Change Request within the AW-TRBAC portal

If the first approver has been successful, the system will then validate the second approval request.

Response

Response Parameter	
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

Table 5.6: REST - Role Change Second Approval Response is executed after successful role change first approval, AW-TRBAC engine validate the request from the value of UserId, RoleId, TaskId and ResourceId

A successful validation of second approver update the system data store and responses with HTTP 200 code, otherwise a HTTP 400 code is returned.

5.3.2 Close Request REST URI

The final REST operation is to close the request, the is performed by the following URI.

```
http://profdoc.uel.ac.uk/awtrbac/portal/auth/task/rolechange/close\_request
```

Request Parameters

Request Parameter	
Resource Information	Description
Operation	Close_request
Request Body	UserId - Unique identifier of a subject (Manager)
	Taskid- Unique identified of an IT workflow
	TaskInstanceId - Unique identifier of task instance
Request format	JSON
Action	POST /awtrbac/portal/auth/task/rolechange/close_request

Table 5.7: REST - Role Change Close Request, the URI for the Role Change close out Request within the AW-TRBAC portal

The close request takes in the UserId, TaskId and TaskInstanceId as the JSON input.

Response

The HTTP 200 response code, indicates a successful close of the request.

To summarise, the SoD URI will support the following operations: initial_request, role_ manager, manager_approval and for BOD: initial_request, role_ coordinator, close_request operations.

Response Parameter	
Resource Information	Description
Response Code	HTTP/1.1 201 created
Response Error	HTTP/1.1 400 bad request
Response Body	status
Response format	JSON

Table 5.8: REST - Role Change Close Request after successful validation of second approver update on the system data store and responses with HTTP 200

5.4 System Implementation

The choice of language for implementation is Java (Gosling (2000)), as there are widely available technologies, framework and open sourced project in Java, which are mature and secure. Some of the technologies such as JAXB (Fialli and Vajjhala (2003)) has wide community support. JAXP (Sun Microsystem), JAX-RS (Li (2011)) are used to develop the backbone of the framework, that includes processing and handling of XML (Bray et al. (1998)), which has REST based API interaction amongst the framework components with Tomcat back-end Server. Spring framework is typically used which is implemented in Java.

The core part of the system is the AW-TRBAC engine. Approach to implementing the engine has been to leverage on the latest industry standard XACML 3.0 (Rissanen (2010)). The open source implementation of this standard is the Balana (Chen et al. (2013)) by WS02. The XACML 3.0 is currently the widely-supported standard in the industry.

5.4.1 Task Workflow Support for XACML

This section describes the implementation of the function in AW-TRBAC engine as part of the extended XACML Standard. We describe each function in turn and explain the mechanism involved in the coupling of policy assertion within policies and data store. Workflow task has been supported by xacml 3.0 by extending some of the core functionality as shown in the in Fig 5.3. This research added dynamic access control requirement within XACML 3.0 by extending some of its core functionalities. Figure 5.3 & 5.4 shows the extended functionalities required for the XACML engine to execute an AW-TRBAC policy and request.

5.4.2 Coupling of Policy Assertions

The policies within the XACML reliant on the requests, as it contains conditional statement and target which are derived from the requests to make a decision 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. A new function of target is introduced to meet the additional requirement to provide the dynamic SoD instance level restriction.

5.4.3 Function Role Check

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 in Fig 5.3 below).

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

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.

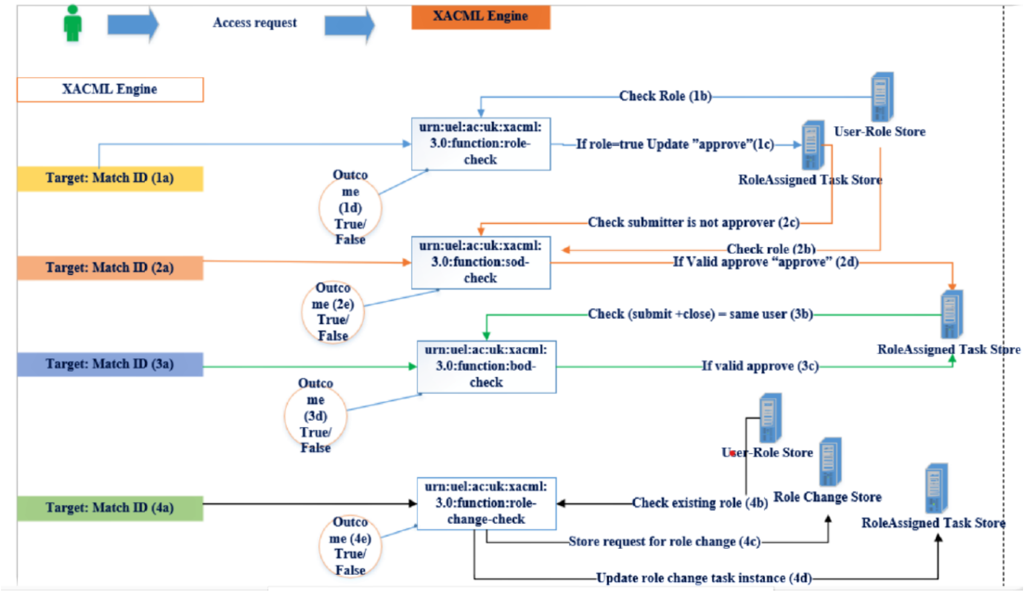


Figure 5.3: Implementation of the function in AW-TRBAC engine as part of the extended XACML Standard to support dynamic access control requirement for SoD and BoD.

5.4.4 Static & Dynamic Segregation of Duties Check (SoD)

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

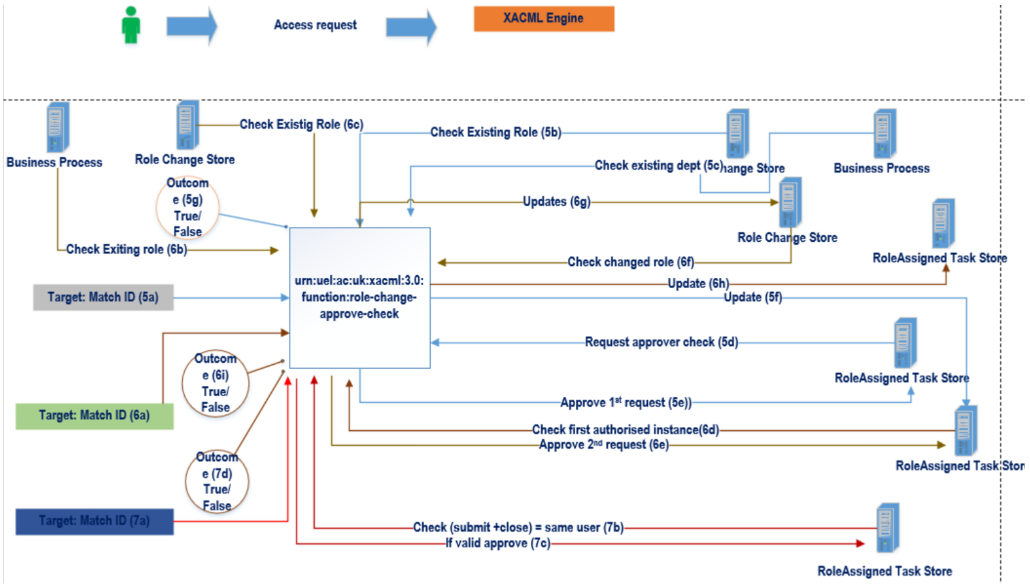


Figure 5.4: Implementation of the function in AW-TRBAC engine as part of the extended XACML Standard to support dynamic access control requirement for role change

statement. Such a concept is not present in XACML. To address this issue, this research introduced 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 values are 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 5.3. For a 'new' variable it creates an entry in the store for the statement, assertion see 3a-3b in Fig 5.3. The content is of a new variable populated and used by the conditional statements, see below Section 5.4.6.

5.4.5 Static & Dynamic Binding of Duties (BoD)

IT workflow task requires Binding on Duties (BoD), which entails match ID against the instance ID in the target 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 in-stance is stored in the role assigned task store see 3a-3c in Fig 5.3.

5.4.6 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, we have created a fourth new function:

```
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 in Fig 5.3.

5.4.7 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 it the most complex function, 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
```

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 in Fig 5.4. 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 Role Assigned Task store, if all successful, it will return true, otherwise false see 7a-7c in Fig 5.4. 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.

5.5 AW-TRBAC Model Integration

The Service has been integrated with the AW-TRBAC system. The service can generate and store its task instance audit trail data and apply task-based enforcements

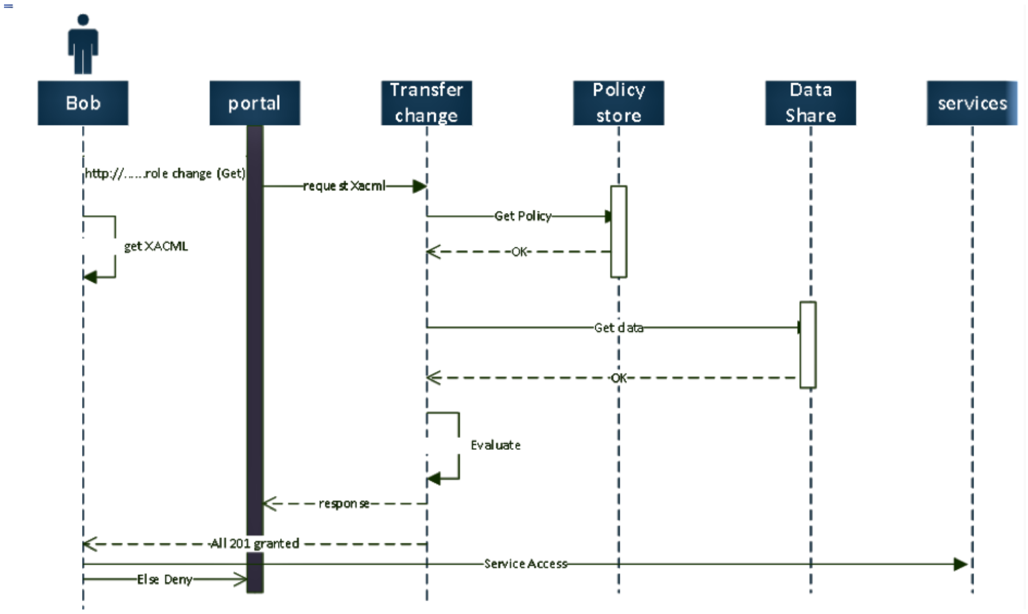


Figure 5.5: AW-TRBAC portal Dynamic Access Control Model Integration, demonstrates the interactions between the system components and the service to allow authorised access to resources

The sequence diagram in Fig 5.5 demonstrates the interactions between the system components and the service. It shows a user, Bob, invoking a resource a task request on the AW-TRBAC portal. The portal using API generates xacml request for this to interact with the AW-TRBAC engine and policy store to validate the request with information from data stores, after evaluating the conditional obligation with the task instance and a decision is made to grant or deny. The user can then have access to the authorised resources.

5.6 summary

This chapter details high level design and implementation of the AW-TRBAC, it illustrates the API interfaces that are used to connect to different component of the AW-TRBAC model, such as URI for invoking a security request and translation into an XACML request, which then used by the extended AW-TRBAC policy engine to make a decision on the request. Also in this chapter the mechanism behind extension of Oasis standard explained specifically to highlight involvement of each function in security request such as Role change or dynamic SoD.

This is the third contribution of this thesis and first of it's kind to be implemented, to bridge the gaps within the dynamic access control, to the best of the author's knowledge there has been no such implementation of policy engine to fulfil these requirements of a dynamic access control with the requirements such as dynamic segregation of duties at task instance level to mitigate broken access control risk of OWASP OWASP (2017)top 10 risk of the application security.

In the next chapter AW-TRBAC functionality and scalability will be experimented and evaluated with number of user simulated tests for functional validation and system performance to understand viability in real life industrial case.

Chapter 6

EXPERIMENTAL SETUP AND RESULTS

So far in Chapter Four AW-TRBAC model that have been proposed to include task instance level restriction of a dynamic access control, this is an improvement from all the previous access control model. In addition,to enforce the additional dynamic functionality of the AW-TRBAC model, the Oasis standard (XACML) have been extended with additional components. The AW-TRBAC engine have been implemented using open source to include the custom functionality of the XACML in chapter Five. This chapter presents the experimental setup of the research and describes tests that have been carried out to determine the use case requirements satisfaction and applicability of the dynamic access control model solution. Therefore, the identified requirements are traced and validated through evaluation.

The evaluation is performed in two phases. The first phase validates the access control model requirements against the implemented AW-TRBAC system. The requirements are pre-requisite for the proposed access control model, therefore it is essential to confirm that the dynamic access control model addresses these requirements. The second phase demonstrates the applicability of the dynamic access control model in a simulated industrial context, using the use case requirements to run an end-to-end process using the AW-TRBAC engine of the dynamic access control model. The results were used to determine whether it has aligned with the policy rules and the task constraints.

6.1 Requirements Satisfaction

The implemented AW-TRBAC model has been tested against the use case for the requirements satisfaction and applicability of the solution. Six test cases have been tested to determine whether the dynamic access control model meets the required constraints and characteristics. A script simulate a user making a security request using REST URI which invoked the AW-TRBAC system, implemented as a microservice architecture, that performs the dynamic access control. Access requests were converted to XACML equivalent which was then validated through the XACML policy, implemented by the AW-TRBAC engine. The engine in turn made various data assertions on the data store before allowing access to the resource (access request system). A total of six scripts were used to test the requirements identified in section 4.1.1 and the execution of the constraints were recorded in a backend SQL database in Fig 6.6.

6.1.1 Test Case

The first test case is the key policy requirements of dynamic segregation of Duties (SoD). A test script in Fig 6.1 was used to simulate a human user invoking a REST API request against the AW-TRBAC System. The request was validated against the policy, see Fig 6.3, in the AW-TRBAC engine, using the policy conditional statements, and the parameters from the input request, see Fig 6.4. If the conditions and attributes were met then user was authorised to access the resource, as shown in Fig 6.5.

6.1.2 Invoking a SOD Security Request Through REST API

The below script is then executed by the AW-TRBAC engine, composes of a REST request for the user "Bob" invokes a SoD request.

```
.....
 String intitalRequestUrl = "http://profdoc.uel.ac.uk/awtrbac/portal/auth/task/rolechange/initial_request";
RestTemplate restTemplate = new RestTemplate();
UserRequest initial_request = new UserRequest();

initial_request.setTask_id("security-request");
initial_request.setSubject_id("bob");
initial_request.setEnvironment_id("SEG001");
initial_request.setResource_id("PC");

HttpEntity<UserRequest> request = new HttpEntity<>(initial_request);
TarbacResponse response = restTemplate.postForObject(intitalRequestUrl, request, TarbacResponse.class);
assertEquals(response.getAction_status(), expected_outcome);
...
```

Figure 6.1: Invoking a SOD Security Request Through REST API

6.1.3 Policy for SoD Assertion in AW-TRBAC Engine

A SoD policy request is handled by the AW-TRBAC engine as shown below

```
PolicyEngineAW trbacEngine = new PolicyEngine()
File inputPolicyFile = new File('/response/xacml-security-response-policy.xml')
String policyRequestPath = "/request/xacml-security-response-req.xml"
Charset ascii = Charset.forName("US-ASCII")
byte[] encoded = Files.readAllBytes(Paths.get(policyRequestPath))
String policyRequestStr = new String(encoded, ascii)
String output = policyEngine.executePolicy(policyRequestStr, file)
```

Figure 6.2: SoD policy request handled by the AW-TRBAC engine

6.1.4 SoD policy to Match

A SoD policy, see Fig 6.3 typically consists of one target and four conditional statements.

Lines 4-10 is the target statement, which matches the SoD request to the SoD policy.

Lines 16-30 show the first two conditional statements. First it checks if the request is coming from a valid user, if it does not match one of the defined values in the policy, it would deny the request. If this is successful, it then checks to see if it is a valid task request.

Lines 32-47 define the third conditional statement, and uses an extended function "role check" (see Fig 5.3) to check the role against the role store. If this is successful, the SoD check is carried out.

Lines 49-65 consist of the fourth and the most important conditional statement. It carries out the SoD check (see Fig 5.3) using the task instance ID in the data store, and a successful check creates a new task instance which is updated in the data store.

6.1.5 XACML Request Generated for a Request Invoked Through API

XACML request

As mentioned earlier, the dynamic access control model has been developed on a microservice infrastructure which converts the request invoked via REST API into an XACML request. The parameters "Task and user" will be matched against the Balana engine policy to validate task instance ID and the user, to provide a decision "deny" or "permit" in Fig 6.4.

```
<Policy xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
  <Description>Task based access control policy to check for segregation of duties </Description>...
  <Target><AnyOf><AllOf> <!-- Check for the right environment -->
  <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
     <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">SEG001</AttributeValue>
    <AttributeDesignator
      AttributeId="urn:oasis:names:tc:xacml:1.0:environment:environment-id"
      Category="urn:oasis:names:tc:xacml:3.0:attribute-category:environment"
      DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true" />
    </Match>
              </AllOf>
     </AnyOf>
  </Target>
   <Rule Effect="Permit" RuleId="Rule-1">  <Condition> <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
     <!-- Check for a valid user -->
     <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
       <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
         <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">bob</AttributeValue>
         <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">phil</AttributeValue>
         <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">mat</AttributeValue>
         </Apply>
       <!-- Check for task, custom attribute check -->
       <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
         <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
           <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">security-request</AttributeValue>
           <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">security-request-approve</AttributeValue>
           <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">change-role</AttributeValue>
           <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">terminate-user</AttributeValue>
           <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">PAM</AttributeValue>
        </Apply>  ...
         <!-- Check for role -->
            <Apply FunctionId="urn:uel:ac:uk:xacml:3.0:function:role-check">
  <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
   <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
     Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
     DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true" />
  </Apply>
  <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
     <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:task:task-id"
       Category="urn:uel:ac:uk:xacml:3.0:task-category:access-task"
      DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true" />
  </Apply>
  <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
    <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
      Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-resource"
                    DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true" />
  </Apply>
              </Apply> ....
  <!-- check for segregation of responsibility, and update DB -->
  <Apply  FunctionId="urn:uel:ac:uk:xacml:3.0:function:sod-check">
    <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
      <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
        Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
        DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true" />
  </Apply>
  <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
    <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:task:task-id"
       Category="urn:uel:ac:uk:xacml:3.0:task-category:access-task"
       DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true" />
  </Apply>
  <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
  <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
      Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-resource"
      DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true" />
  </Apply>
             </Apply>
           </Apply>
       </Condition>
</Policy>
```

Figure 6.3: Policy for SoD assertion in AW-TRBAC Engine

```
<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="....#string">security-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">mat</AttributeValue>
      </Attribute>
    </Attributes>
    <!--  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:environment-id"
      ↪ IncludeInResult="false">
         <AttributeValue
         ↪ DataType="http://www.w3.org/2001/XMLSchema#string">SEG002</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">tif917803b</AttributeValue>
      </Attribute>
   </Attributes>
</Request>
```

Figure 6.4: XACML Request Generated for a Request Invoked Through API

6.1.6 Response for the Invoked Request Through REST API

As shown below 6.5, the response has been provided for the invoked EST API request (test case one). If all of the conditions are met then the decision is to permit, otherwise the decision is to deny.

Response to the Request

```xml
<Response xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17">
  <Result>
    <Decision>Permit</Decision>
    <Status>
      <StatusCode Value="urn:oasis:names:tc:xacml:1.0:status:ok"/>
    </Status>
  </Result>
</Response>
```

Figure 6.5: Response for the Invoked Request Through REST API

6.1.7 Binding of Duties (BoD)

BoD dependencies are when security request to be closed off by the same user who submit the request, this is enforced using the policy and the custom function defined in the policy engine, full list of policies have been appended at the appendix A. As shown below the policy which has been used by the AW-TRBAC engine to carry out BoD constraints against the data store Role Assigned Store task instance, in this instance is Bob, who is the role Security coordinator can close the request, the policy will use the parameters Role, Task Instance ID to evaluate the request to provide decision to user, see Appendix A.

6.2 Requirement Validation

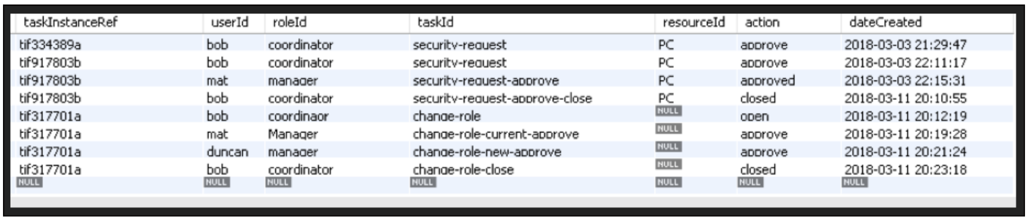
For validation purposes, the implemented solution has been run and associated outputs were recorded in Fig 6.6. For Example, UserId “Mat” has role permission “manager” which allows him to approve the security request submitted by UserID “Bob” who has the role “coordinator”, as shown in row one of the Fig 6.6. This satisfies the requirement that “only authorised users access the resources” and “second row in the Fig 6.6, showing SoD constraints followed through role and task instance” through restriction on task instance ID “Tif917803b” row one of the table. The Manager Role acted on the task that is on 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”.

In the action column in row two of the Fig 6.6, the status is changed to “approved”, this records the role that performed and the action on a task (security request) and resources (PC) that have been authorised by the role manager (Mat) on a task instance (Tif917803b) at a point in time. This satisfies the requirement of “adequate event logging and access availability in real time ensuring governance”.

Dynamic segregation of duties at instance level is also shown in row four, “tf317701a” is an instanceId for a role change request, submitted by the role “Coordinator”, which is shown in the task column as “Role Change”. This is approved by the role “manager”, a new task instance is recorded “change-rolecurrent-approve”, and action status is set to “approve”. This proves that SoD is enforced at the levels of task instance Id, subject and role. Also shown in row six is the subject “Duncan” who is the second level of approver for the onboarding service, and who approves the same task instance Id.

The task instance reference “tf317701a” as shown in Fig 6.6, ensures task contingency and sequence flow of tasks approval maintained and role change data store is updated. This will enable existing role will no longer remain active for the subject. This rule of enforcement in the policy allows revocation of existing entitlements and provision of new credentials.

Results in Fig 6.6, demonstrate that the 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 through instance level restriction and validation through the function in the policy engine, meeting dynamic access control requirements.



taskInstanceRef	userId	roleId	taskId	resourceId	action	dateCreated
tif334389a	bob	coordinator	security-request	PC	approve	2018-03-03 21:29:47
tif917803b	bob	coordinator	security-request	PC	approve	2018-03-03 22:11:17
tif917803b	mat	manager	security-request-approve	PC	approved	2018-03-03 22:15:31
tif917803b	bob	coordinator	security-request-approve-close	PC	closed	2018-03-11 20:10:55
tif317701a	bob	coordinaor	change-role	NULL	open	2018-03-11 20:12:19
tif317701a	mat	Manager	change-role-current-approve	NULL	approve	2018-03-11 20:19:28
tif317701a	duncan	manager	change-role-new-approve	NULL	approve	2018-03-11 20:21:24
tif317701a	bob	coordinator	change-role-close	NULL	closed	2018-03-11 20:23:18
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 6.6: Audit Logs of the Tests Based on Requirement Satisfaction Experiment

6.3 Applicability of the AW-TRBAC

To measure the system performance for the use case, and to check whether AW-TRBAC is a sustainable in the real-life solution, this research experiment benchmarked against similar work carried out by Ali and Moreau (2013) whereby the author extended the Balana engine to translate the provenance-based policy language into an XACML request for provenance based data. To the author’s knowledge there have been no other attempts to extend the Oasis standard for dynamic access control requirements, thereby there is no other existing data available for comparison. 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 executions were recorded. Requests were executed in a sequence of ten thousand batches and each result (containing mean value with error bar at 95% confidence level) was recorded against the two hypotheses:

1. System performance will not degrade with the dynamic access control request.

2. Increased conditional statement with a role change will affect the processing time.

6.3.1 Hypothesis

Hypothesis One: System performance will not degrade with the dynamic access control request.

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

Methods: The requirement one from (section 4.1.1) of the use case in relation to SoD has been tested for policy evaluation, generated using REST API client. The REST client would make a query to the AW-TRBAC engine, which executed the XACML policy and response back to the REST client as shown in Fig 6.7. The engine uses data from the data store from previous methods to evaluate the policy. The experimental setup ran 1 million end-to-end runs in a batch of 10,000 security request executions in Fig 6.8, then calculated the mean which is depicted in Fig 6.10.

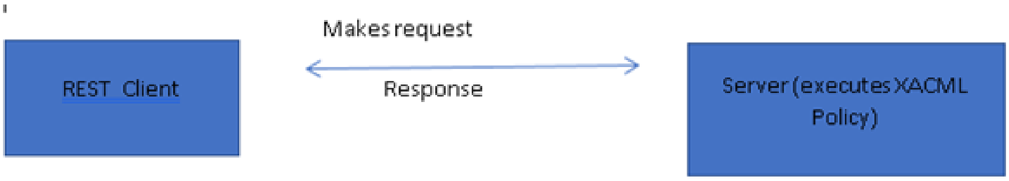


Figure 6.7: Security Request Executed via API Client Call to AW-TRBAC Engine

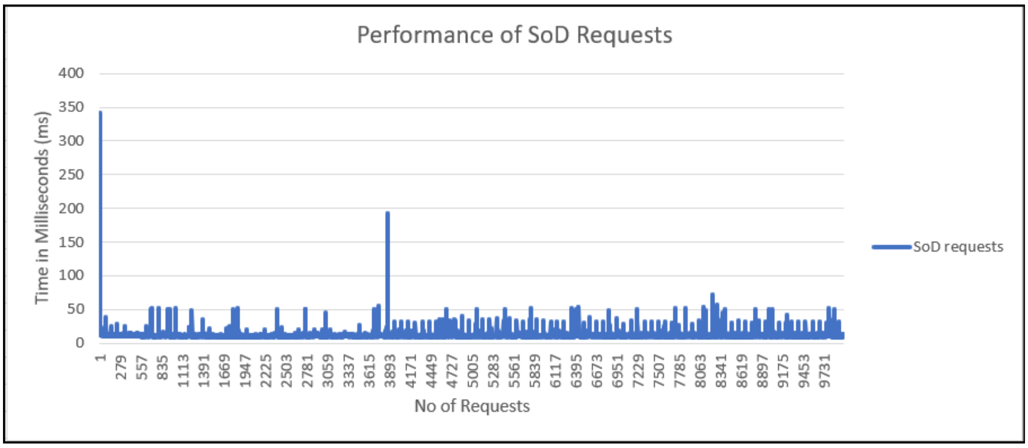


Figure 6.8: Security Request Result for the API client to AW-TRBAC Engine Performance Test

Results: As shown in Fig 6.10, there is a relatively high execution time at the beginning, this is due to a number of factors such as: class loader initialising of classes, loading OS level resources, fragmented hard disk. There has a sudden spike in execution in the middle of the experiment, this is believed to be due to the running of the JVM

(Java Virtual Machine) garbage collector, which is a program that manages memory automatically wherein de-allocation of objects is handled by Java. When there are no references to an objects, it is assumed to be no longer needed and the memory occupied by the objects is reclaimed and deleted resulting in execution jitter (delay). However, results were consistent throughout the experiment for execution of end to end process of SoD request, it took an average of 0.12 (S) for the end-to-end request to complete, with a standard error of 0.02(S) with the confidence interval set at 95%. This indicates that despite the additional overheads of AW-TRBAC engine, benchmarked against (Ali and Moreau (2013)) which took 0.42 (S) for end-to-end request of similar experiment. The system performance is commercially viable.

Hypothesis two: Increased conditional statement with a role change will affect the processing time.

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

Methods: Using REST API client, role change requests were made to the XACML server, which executed the XACML policy by the AW-TRBAC engine. The experimental setup ran 1 million ends to end runs of a security request execution, as shown in Fig 6.9, then calculating the variance for simple security request against the complex role change request the variance this is presented in Fig 6.10.

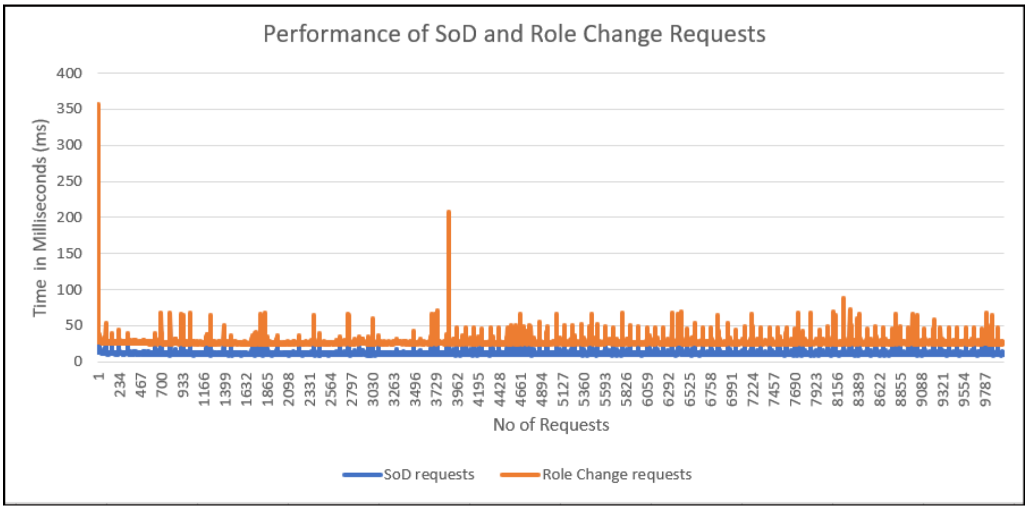
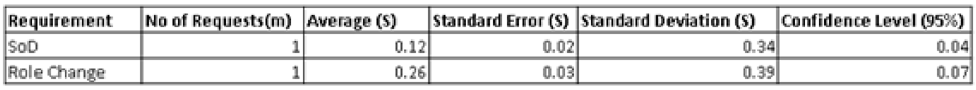


Figure 6.9: Comparison of System Performance Against Complex Role Change Request

Results: As shown in Fig 6.9, there has been a spike of executions at the beginning and in the middle of the experiment on execution of the policies. This is due to several factors; at the beginning class loader initialisation of the classes, loading os level resources, fragmented hard drive, which contributed to the spike with few initial policies, however it has been consistent thereafter. There has been a sudden spike in execution in the middle of the experiment, this is believed to be due to the running of the JVM garbage collector, which is a program that manages memory automatically wherein deallocation of objects is handled by Java, when there are no references to an objects, it is assumed to be no longer needed and the memory occupied by the objects is reclaimed and deleted, this is an automated standard component of Java programming language. However, there has been recent research in Simão et al. (2019) memory management for real-time Java VM (virtual Machine), a Self-adaptive approach for memory-performance efficiency through a learning phase and an execution phase (a training phase where it collects, with different heap (data Structure) resizing policies, representative execution metrics during the lifespan of a workload; and an execution phase where the execution parameters of new workloads against those of already seen workloads, and enforces the best heap resizing policy)to improve the realtime task execution jitter. It has been noted that complex role change request have additional conditional statements which require evaluation of variable parameters coupled with retrieval of information from the data store, which delays the decision output. In comparison to SoD requests, which require few statements analyses by the engine and have a mean request time of 0.12(S), Role Change requests in this experiment had a mean request time of 0.26 (S) in Fig 6.10. The time taken to process the Role Change request with complex conditions is almost double. However, the performance remained consistent during the performance test. As show in Fig 6.10 the standard deviation of 0.039(S) from the mean for Role Change requests remained fairly constant with a 0.039(S) s margin of error and a 95% confidence interval of 0.076, which indicates that if the system performance was 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. Although complexity within the statement has increased the required processing power, this will be scalable in industry with enhanced processing power. In this thesis relational data store is used, which means performance is better that XML data store used by Ali (Ali and Moreau (2013)).



Requirement	No of Requests(m)	Average (S)	Standard Error (S)	Standard Deviation (S)	Confidence Level (95%)
SoD	1	0.12	0.02	0.34	0.04
Role Change	1	0.26	0.03	0.39	0.07

Figure 6.10: Summary of Results for SoD and Role Change Request

- N - number of items (request unit)
- SD - Standard deviation (s)
- SE - Standard error (s)
- CI - Confidence interval (set to 95%)

6.4 Summary

This chapter presents the experimental setup and results for the AW-TRBAC system evaluation against the functional requirements of the use case. The first phase of the experiment has been setup using six test cases that simulate a user invoking a security request against the AW-TRBAC engine to satisfy the requirements constraints and policy enforcement of the dynamic access control model. AW-TRBAC have been successfully able to meet the requirements of dynamic SoD through restriction of instance level SoD, dynamic role change through role change at the policy level, preventing the risk of escalation of privilege, inhibiting the access to previous role and credentials, adequate logging through sequence of events, providing governance of access management. The second phase of the experiment was carried out to learn about system performance under a stress test; this has been carried out by measuring the security request end to end response time from client to server, to learn about the solution performance when a complex security request is invoked. There were total of one million requests tested in batches of 10,000 and the mean time of each batch was recorded to calculate the final mean time taken for each type of request to be processed by the system. To the author's knowledge no such policy engine have been developed to meet the requirements of AW-TRBAC model, thereby no experimental data is available, however experiments that has been benchmarked against similar experiment carried out by (Ali and Moreau (2013)) and the results indicate that it is commercially viable.

In the next chapter, thesis is concluded with summary of the findings and its limitations and opportunities for future research.

Chapter 7

CONCLUSIONS

This thesis focused primarily on resolving a real-world problem, using academic research for the specific industry problem and contribute a solution which is viable within the borderless security in the new technology era. The research was carried out to resolve an 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 the overall risk posture of the firm. The research developed a dynamic access control model leveraging on existing RBAC and ABAC access control models to provide the capability of task instance segregation coupled with role level 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 an audit trail of processes from owner approvals, through the sequence of tasks being followed and enforced, to role, task, process and task instances. The research extended the OASIS standard, introducing five new functions and two repositories to enhance the functionality through further development of the AW-TRBAC engine using open source Balana engine. This extension helps to fill a gap within current access control models to enable real-time decision making in a dynamic borderless environment, such as adoption of cloud and Robotic process automation. The research also focused on mitigating the critical web application risk highlighted by the OWASP standard, preventing broken access control through policy/rule enforcement and a dynamic access control model incorporating dynamic SoD and governance. The research also provides a solution to mitigate the risks associated with insufficient logging and monitoring 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. The AW-TRBAC model framework was able to meet the requirements for borderless network perimeter access control that requires dynamic and real time decision making for providing resource access to authorized users. It was noted that simple security requests took, on average, 0.12 s to process, while the complex request such as a change in service role with additional conditional statements and targets doubled this time to 0.26 s; this, in comparison to the benchmark experiment by (Ali and Moreau (2013)), is commercially viable.

7.1 Contributions of this Thesis

Key Contributions of this thesis are:

7.1.1 First Contribution

The development of a dynamic access control model. This was developed using the use case requirements of an investment bank and based on filling gaps identified in the literature review. In Banking industries, due to high risk applications that supports its investment decisions on operational. It is crucial that access to resources is permitted only to authorised users and is permitted in a timely manner when access is required. Access governance ensures access to resources is authorised appropriately and accountably, and traceability is retained. In the use case, access control and access governance systems were operating in silos and it was necessary to have one streamlined process and a system to support that process. This system needed to be dynamic, with intelligence to make a real time decision based on policy rules which support both roles based and task based instance restrictions, meet BoD and SoD requirements, and provide access governance and logs (audit logs for compliance requirements in forensic analytic). The access control model needed to meet the challenge of role changes, ensuring entitlements were revoked from the previous level prior to granting new access, and ensure access has been authorised and governed appropriately to promote governance and risk mitigation.

7.1.2 Second Contribution

An extended XACML based open source balana engine that facilitates and enforces the dynamic access control rules and additional requirements to fulfil the gaps in currently available access control models: to enforce SoD at the task instance, remediate broken access control risk, and log instance task events to enhances access governance to provide visibility of unmanaged data.

To fulfil the policy enforcement, a policy language was required which supports the Role, Task and Operation notation. XACML was chosen as the policy language which would work well with dynamic access control due to its cohesion and coupling ability. The XACML Oasis standard has previously been extended to support RBAC, however it was lacking task notation capability. To meet the task requirements and the dynamic capability of the AW-TRBAC model, and enable data assertion within the data store, which could be utilized during decision making and retaining audit trails for forensic analytic (Compliance), XACML standard has been extended with customised function for policy enforcement.

7.1.3 Third Contribution

An Implementation of the AW-TRBAC engine, leverages on banking solution, to ensure for greater control of existing information security management and information privacy. The AW-TRBAC has been built on the open source Balana policy engine by developing functions to support additional functionality; it was extended to support an additional five functions for real-time decision making capability.

The evaluation was carried out, using the use case requirements, by simulating six test cases to meet constraints and characteristics defined in the dynamic access control model. A script simulated a user making a security request using the REST URI, which invoked the TR-BAC system, implemented as microservice architecture that performs the dynamic access control. The result showed that the AW TRBAC model successfully met the requirement constraints of the case study and mitigated the risk of escalation of privilege to prevent data disclosure to unauthorised user.

1. Requirement 1: Access request shall only be submitted by the role "Security Coordinator".
2. Requirement 2: Authorised Business Process Owner should approve the Security request.
3. Requirement 3: Only Authorised users shall be permitted access to resources.
4. Requirement 4: User Access to be revoked after termination of service and service change.
5. 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.
6. Requirement 6: Sufficient logging of events to be retained and monitored.

The AW-TRBAC model has been tested for applicability against the use case by measuring system performance under stress test of executing 1 million security requests of a complex nature. AW-TRBAC has minimal impact on overall system performance despite changing user access requests dynamically and mitigating the risk of escalation of privilege to prevent data disclosure.

7.2 Future Research

This thesis has achieved the goals and objectives set for the research, however, there are areas which could be explored further in future research, such as:

Development of Tool: Although AW-TRBAC model has undergone performance test, due to time constraints and capacity, it was not possible to test the dynamic access model to its full potential in real life. It is, however, difficult to predict performance on an industrial scale. Further research could be to develop AW-TRBAC model at full industrial scale to further understand its limitations and benefits. There are many complex policies within the industries with complex rules which may need to be explored further to test for usability.

Adaptation within the Federated Environment: AW-TRBAC model has been developed and designed to be adopted in a dynamic environment; this could be applied in a federated environment such as a cloud, as it has the capability to evaluate requests in real-time decision making. It could well suit an environment that requires accountabilities and compliance, such as professional services or aviation industries due to its dynamic nature and analysis capability.

CASB (Cloud Access Security Broker) Environment: AW-TRBAC model could be extended to support a broker in a borderless security setup. This would be a service that sits between an organization's on-premises infrastructure and a cloud provider's infrastructure. Acting as a gatekeeper, it would allow the organization to extend the reach of their security policies beyond their own infrastructure.

Authorisation Engine: AW-TRBAC model is effectively an authorisation model which could be integrated with SSO technology such as SAML to provide an identity access management solution both on premise and in a cloud, as part of the 3rd wave of borderless IAM.

Appendix A

Appendix

Test Case: A Coordinator can only submit a security request

Test Script

```java
public void xacml_task_security_request_test() throws IOException {
  ...
  PolicyEngine policyEngine = new PolicyEngine();
  File file = new File("target/classes/policies/xacml-security-request-policy.xml");
  ....
  //XACML Policy to use
  String policyRequestPath = "target/classes/policies/xacml-security-request-req.xml";

  Charset ascii = Charset.forName("US-ASCII");
  byte[] encoded = Files.readAllBytes(Paths.get(policyRequestPath));
  String policyRequestStr = new String(encoded, ascii);
  String output = policyEngine.executePolicy(policyRequestStr, file);
}
```

XACML Request for This Test

```
<Request xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
  CombinedDecision="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">security-request</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">bob</AttributeValue>
    </Attribute>
  </Attributes>

  <!--  Policy to match  -->
  <Attributes
    Category="urn:oasis:names:tc:xacml:3.0:attribute-category:environment">
    <Attribute At-tributeId="urn:oasis:names:tc:xacml:1.0:environment:environment-id"
      IncludeInResult="false">
      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">SEG001</AttributeValue>
    </Attribute>
  </Attributes>

  <!--  Resource to access -->
  <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">PC</AttributeValue>
    </Attribute>
  </Attributes>
</Request>
```

XACML Policy to Match

```
<Policy xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
  PolicyId="MyPolicy"
  RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:first-applicable"
  Version="1.0">
  <Description>Task based access control policy to check for segregation
    of responsibility
  </Description>
  <Target>
    <AnyOf>
      <AllOf> <!-- Check for the right environment -->
        <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">SEG001</AttributeValue>
                                        ....
                                       </Match>
      </AllOf>
    </AnyOf>
  </Target>
  <Rule Effect="Permit" RuleId="Rule-1">
    <Condition>
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
        <!-- Check for a valid user -->
        <Apply
          Func-tionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
          <Apply Func-tionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">bob</AttributeValue>
                                                ...
                                                </Apply>
                                        ....
        </Apply>

        <!-- Check for resource access type -->
        <Apply
          Func-tionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
          <Apply Func-tionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
            <AttributeValue
                                                ...
                                                DataType="http://www.w3.org/2001/XMLSchema#string">Network-drive</AttributeValue>
          </Apply>
          ...
        <!-- Check for task, custom attribute check -->
        <Apply
          Func-tionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
          <Apply Func-tionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">security-request</AttributeValue>
            <AttributeValue

                                                ↪  DataType="http://www.w3.org/2001/XMLSchema#string">security-request-approve</AttributeValue>
                                                ....
                                              </Apply>
          <AttributeDesignator At-tributeId="urn:oasis:names:tc:xacml:1.0:task:task-id"
            Category="urn:uel:ac:uk:xacml:3.0:task-category:access-task"
            DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true" />
        </Apply>

        <!-- Check for role (user, task, resource)-->

        <Apply FunctionId="urn:uel:ac:uk:xacml:3.0:function:role-check">
          <Apply Func-tionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
            <AttributeDesignator
              At-tributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
              Catego-ry="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
              DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true" />
          </Apply>
          <Apply Func-tionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
            <AttributeDesignator At-tributeId="urn:oasis:names:tc:xacml:1.0:task:task-id"
              Catego-ry="urn:uel:ac:uk:xacml:3.0:task-category:access-task"
              DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true" />
          </Apply>
                                        ...
          </Apply></Apply></Apply></Condition>
                                        ...
  </Rule>

  <Rule Effect="Deny" RuleId="Deny-Rule" />
</Policy>
```

Test Case for SoD

```
\textbf{Test Script}

public void xacml_task_security_approve_test() throws IOException {

  ...
  PolicyEngine policyEngine = new PolicyEngine();

  File file = new File("target/classes/policies/xacml-security-response-policy.xml");
  String policyRequestPath = "target/classes/policies/xacml-security-response-req.xml";
  Charset ascii = Charset.forName("US-ASCII");
  byte[] encoded = Files.readAllBytes(Paths.get(policyRequestPath));
  String policyRequestStr = new String(encoded, ascii);
  String output = policyEngine.executePolicy(policyRequestStr, file);
  Assert.assertTrue(output.contains("Permit"));
}
```

XACML Request: for This Test

```
<Request xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
  CombinedDecision="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">security-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">mat</AttributeValue>
    </Attribute>
  </Attributes>

  <!--  Policy to match -->
  <Attributes
    Category="urn:oasis:names:tc:xacml:3.0:attribute-category:environment">
    <Attribute At-tributeId="urn:oasis:names:tc:xacml:1.0:environment:environment-id"
      IncludeInResult="false">
      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">SEG002</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">tif917803b</AttributeValue>
    </Attribute>
  </Attributes>
</Request>
```

XACML Policy to match:

```
<Policy xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
  PolicyId="MyPolicy"
  RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:first-applicable"
  Version="1.0">
  <Description>Task based access control policy to check for segeration
    of responsibility
  </Description>
  <Target>
  ....
        </Target>
  <Rule Effect="Permit" RuleId="Rule-1">
    <Condition>
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
        <!-- Check for a valid user -->
        <Apply
          Func-tionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
          <Apply Func-tionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">bob</AttributeValue>
          ...
                                                </Apply>
                                                ...
                                        </Apply>

        <!-- Check for task, custom attribute check -->
        <Apply
          Func-tionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
          <Apply Func-tionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">terminate-user</AttributeValue>
            <AttributeValue
                                                DataType="http://www.w3.org/2001/XMLSchema#string">PAM</AttributeValue>
                                                ..
          </Apply>
          <AttributeDesignator At-tributeId="urn:oasis:names:tc:xacml:1.0:task:task-id"
            Category="urn:uel:ac:uk:xacml:3.0:task-category:access-task"
            DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true" />
        </Apply>

        <!-- Check for role -->

        <Apply FunctionId="urn:uel:ac:uk:xacml:3.0:function:role-check">
          <Apply Func-tionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
            <AttributeDesignator
                                                ...
          </Apply>
          <Apply Func-tionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
                                        ...
                                        </Apply>

          <Apply Func-tionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
          ...
                                        </Apply>

        </Apply>

        <!-- check for segeration of responsibility, and update DB -->
        <Apply  FunctionId="urn:uel:ac:uk:xacml:3.0:function:sod-check">

          <Apply Func-tionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
            <AttributeDesignator
              At-tributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
              Catego-ry="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
              DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true" />
          </Apply>

          <Apply Func-tionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
                                        ...
          </Apply>
                                        ....
          </Apply></Apply></Apply></Condition>
  </Rule>

  <Rule Effect="Deny" RuleId="Deny-Rule" />
```

Test Case for BoD

Test Script

```
public void xacml_BOD_task_security_response_close_test() throws IOException {

  ...
  PolicyEngine policyEngine = new PolicyEngine();
  File file = new
  ↪  File("target/classes/policies/xacml-security-response-close-policy.xml");
  String policyRequestPath =
  ↪  "target/classes/policies/xacml-security-response-close-req.xml";

  Charset ascii = Charset.forName("US-ASCII");
  byte[] encoded = Files.readAllBytes(Paths.get(policyRequestPath));
  String policyRequestStr = new String(encoded, ascii);
  String output = policyEngine.executePolicy(policyRequestStr, file);
  ...
}
```

XACML Request: for This Test

```xml
<Request xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
  CombinedDecision="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">security-request-approve-close</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">bob</AttributeValue>
    </Attribute>
  </Attributes>

  <!--  Policy to match -->
  <Attributes
    Category="urn:oasis:names:tc:xacml:3.0:attribute-category:environment">
    <Attribute At-tributeId="urn:oasis:names:tc:xacml:1.0:environment:environment-id"
      IncludeInResult="false">
      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">SEG003</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">tif917803b</AttributeValue>
    </Attribute>
  </Attributes>
</Request>
```

XACML Policy to Match:

```
<Policy xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
  PolicyId="MyPolicy"
  RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:first-applicable"
  Version="1.0">
  <Description>Task based access control policy to check for segeration
    of responsibility
  </Description>
  <Target>....</Target>
  <Rule Effect="Permit" RuleId="Rule-1">
    <Condition>
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
        <!-- Check for a valid user -->
        <Apply
          Func-tionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
                                        ...
                                        </Apply>
          <AttributeDesignator
            At-tributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
            Catego-ry="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
            DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true" />
        </Apply>

        <!-- Check for task, custom attribute check -->
        <Apply
          Func-tionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
          <Apply Func-tionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">terminate-user</AttributeValue>
            <AttributeValue
                                                DataType="http://www.w3.org/2001/XMLSchema#string">PAM</AttributeValue>
                                                ...
          </Apply>
          <AttributeDesignator At-tributeId="urn:oasis:names:tc:xacml:1.0:task:task-id"
            Category="urn:uel:ac:uk:xacml:3.0:task-category:access-task"
            DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true" />
        </Apply>


        <!-- check for segeration of responsibility, and update DB -->
        <Apply  FunctionId="urn:uel:ac:uk:xacml:3.0:function:bod-check">

          <Apply Func-tionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
            <AttributeDesignator
              At-tributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
              Catego-ry="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
              DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true" />
          </Apply>

          <Apply Func-tionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
            <AttributeDesignator
              At-tributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
              Catego-ry="urn:oasis:names:tc:xacml:1.0:subject-category:access-resource"
              DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true" />
          </Apply>
        </Apply>
      </Apply>
    </Condition>
  </Rule>

  <Rule Effect="Deny" RuleId="Deny-Rule" />
```

Test Case for Role Change: Department Transfer 1st Approval Test Script

```
public void xacml_SOD_task_change_role_request_approve_one_test() throws IOException {
  ..
  PolicyEngine policyEngine = new PolicyEngine();
  File file = new
  ↪  File("target/classes/policies/xacml-change-role-current-approve-policy.xml");
  String policyRequestPath =
  ↪  "target/classes/policies/xacml-change-role-current-approve-req.xml";
  Charset ascii = Charset.forName("US-ASCII");
  byte[] encoded = Files.readAllBytes(Paths.get(policyRequestPath));
  String policyRequestStr = new String(encoded, ascii);
  String output = policyEngine.executePolicy(policyRequestStr, file);
  Assert.assertTrue(output.contains("Permit"));
}
```

XACML Request: for this test

```
<Request xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
  CombinedDecision="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">change-role-current-approve</AttributeValue>
    </Attribute>
  </Attributes>

  <!--  User approving the role change-->
  <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">mat</AttributeValue>
    </Attribute>
  </Attributes>

  <!--  Policy to match  -->
  <Attributes
    Category="urn:oasis:names:tc:xacml:3.0:attribute-category:environment">
    <Attribute At-tributeId="urn:oasis:names:tc:xacml:1.0:environment:environment-id"
      IncludeInResult="false">
      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">SEG005</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">tif317701a</AttributeValue>
    </Attribute>
  </Attributes>

</Request>
```

XACML Policy to match

```
<Policy xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
  PolicyId="MyPolicy"
  RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:first-applicable"
  Version="1.0">
  <Description>Task based access control policy for role change re-quest</Description>
  <Target>
  ....
        </Target>
  <Rule Effect="Permit" RuleId="Rule-1">
    <Condition>
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
        <!-- Check for a valid request user for requesting the role change-->
        <Apply
          Func-tionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
                                        ....
                                        <AttributeDesignator
            At-tributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
            Catego-ry="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
            DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true" />
        </Apply>


        <!-- Check for task, custom attribute check -->
        <Apply
          Func-tionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
          <Apply Func-tionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">change-role-current-approve</AttributeValue>
          </Apply>
          <AttributeDesignator At-tributeId="urn:oasis:names:tc:xacml:1.0:task:task-id"
            Category="urn:uel:ac:uk:xacml:3.0:task-category:access-task"
            DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true" />
        </Apply>

        <!--  Check for valid existing role -->

        <!-- Change role request approve 1st (user, task, instance )-->

        <Apply FunctionId="urn:uel:ac:uk:xacml:3.0:function:role-change-approve-check">
          <Apply Func-tionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
            <AttributeDesignator
              At-tributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
              Catego-ry="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
              DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true" />
          </Apply>

          <Apply Func-tionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
            <AttributeDesignator
              At-tributeId="urn:oasis:names:tc:xacml:1.0:task:task-id"
              Catego-ry="urn:uel:ac:uk:xacml:3.0:task-category:access-task"
              DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true" />
          </Apply>

          <Apply Func-tionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
            <AttributeDesignator
              At-tributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
              Catego-ry="urn:oasis:names:tc:xacml:1.0:subject-category:access-resource"
              DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true" />
          </Apply>

        </Apply>
      </Apply>
    </Condition>
  </Rule>

  <Rule Effect="Deny" RuleId="Deny-Rule" />
```

Onboarding Manager (2nd) Approval for Role Change:

Test Script

```
public void xacml_SOD_task_change_role_request_approve_two_test() throws IOExcep-tion {

  ...
  PolicyEngine policyEngine = new PolicyEngine();

  File file = new
  ↪  File("target/classes/policies/xacml-change-role-new-approve-policy.xml");
  String policyRequestPath =
  ↪  "target/classes/policies/xacml-change-role-new-approve-req.xml";

  Charset ascii = Charset.forName("US-ASCII");
  byte[] encoded = Files.readAllBytes(Paths.get(policyRequestPath));
  String policyRequestStr = new String(encoded, ascii);
  String output = policyEngine.executePolicy(policyRequestStr, file);
  Assert.assertTrue(output.contains("Permit"));
}
```

XACML Request: for the test

```
<Request xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
  CombinedDecision="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">change-role-new-approve</AttributeValue>
    </Attribute>
  </Attributes>

  <!--  User approving the role change-->
  <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">duncan</AttributeValue>
    </Attribute>
  </Attributes>

  <!--  Policy to match  -->
  <Attributes
    Category="urn:oasis:names:tc:xacml:3.0:attribute-category:environment">
    <Attribute At-tributeId="urn:oasis:names:tc:xacml:1.0:environment:environment-id"
      IncludeInResult="false">
      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">SEG006</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">tif317701a</AttributeValue>
    </Attribute>
  </Attributes>

</Request>
```

XACML Policy to Match

```
<Policy xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
  PolicyId="MyPolicy"
  RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:first-applicable"
  Version="1.0">
  <Description>Task based access control policy for role change re-quest</Description>
  <Target>
        ...
  </Target>
  <Rule Effect="Permit" RuleId="Rule-1">
    <Condition>
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
        <!-- Check for a valid request user for requesting the role change-->
        <Apply
          Func-tionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
          <Apply Func-tionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">bob</AttributeValue>
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">duncan</AttributeValue>
                                                ...
          </Apply>
          <AttributeDesignator
            At-tributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
            Catego-ry="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
            DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true" />
        </Apply>


        <!-- Check for task, custom attribute check -->
        <Apply
          Func-tionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-one-member-of">
          <Apply Func-tionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">change-role-new-approve</AttributeValue>
          </Apply>
          <AttributeDesignator At-tributeId="urn:oasis:names:tc:xacml:1.0:task:task-id"
            Category="urn:uel:ac:uk:xacml:3.0:task-category:access-task"
            DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true" />
        </Apply>

        <!--  Check for valid existing role -->

        <!-- Change role request approve 1st (user, task, instance )-->

        <Apply FunctionId="urn:uel:ac:uk:xacml:3.0:function:role-change-approve-check">
          <Apply Func-tionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
            <AttributeDesignator
              At-tributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
              Catego-ry="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
              DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true" />
          </Apply>

          <Apply Func-tionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
            <AttributeDesignator
              At-tributeId="urn:oasis:names:tc:xacml:1.0:task:task-id"
              Catego-ry="urn:uel:ac:uk:xacml:3.0:task-category:access-task"
              DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true" />
          </Apply>

          <Apply Func-tionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
            <AttributeDesignator
              At-tributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
              Catego-ry="urn:oasis:names:tc:xacml:1.0:subject-category:access-resource"
              DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true" />
          </Apply>

        </Apply>
      </Apply>
    </Condition>
  </Rule>

  <Rule Effect="Deny" RuleId="Deny-Rule" />
```

A.1 Case Study

This is a brief outline of the case study and the workshop conducted with the key stakeholder from an Investment Bank in London.

A.1.1 Introduction and Justification

Information is the key driver for business strategy development, to deliver world class and innovation client service and to remain relevant and competitive. It is imperative to govern and manage access to the information with adequate information security and access governance. Information access governance present challenges, due to sheer volume of complex systems, processes with diverse policies.

Investment Bank are highly regulated environment with critical (High Risk) application that require robust information security controls to protect from malicious threat and to maintain its, availability, integrity and confidentiality. This require security controls to be embedded into the system development to include both technical and procedural controls. Control such as Segregation of Duties (SoD) are as an example which is to prevent fraud, needs to be included into system level and at the process level to ensure a single individual or business process cannot submit and approve a request.

A.1.2 Aim

The aim of this case study is to collect relevant information in a workshop with key stakeholder in a real-life investment organisation to understand how Access governance are executed. This will provide insight into how access governance is carried out in multiple business processes that run across multiple systems that deal with different authorisation management policies.

A.1.3 Methodology

Having worked in the Information assurance department for the organisation, it was observed there were processes which were inconsistent, lack of visibility of attestation reports, access request process was complex and multiple silos access system and policy applied for management of access control. As banking organisations are regulated and require robust security policies to protect its asset such as information.

There were no of workshops which has been used to connect the dots of various access governance process, including new joiner, leaver, role transfer and emergency maintenance process requiring privileged account management. Questions were framed to allow adequate information can be collected without deep diving into the process to stay relevant.

A.1.4 Choosing the process

Based on the complexity of the environment, run across multiple systems, includes sensitive access control requirements,information flows across number of business processes. Access governance has been grouped into three process: access request, access authorisation and access Administration.

A.1.5 Key Stakeholder

As the three chosen processes are operational and deal with the assets, systems and process of the organisation, and because information security through access governance is the focus, thereby the following three key stakeholder were invited to the workshop;

1. Information Security Dept Head
2. Security Administrator
3. Information Owner (Business side Director/approver)

A.1.6 Workshop

Workshop was held with the key stakeholder in the office to understand the current process of Access Management Lifecycle to include the tool, people and process involved in the governance of access request approval.

Access Request Process

1. Who are the people involved in the process?
2. What are the roles of the people involved?
3. What is the role hierarchy of these roles (chain of command)?
4. What are the systems involved in this process and who is responsible for these systems?
5. What kind of information security concerns do you have in relation to this process?
6. Are there any access control restrictions related to systems involved in this process?

Access Authorisation Process

1. What are the steps for access authorisation of a request?
2. What are the different types of request?
3. Who are the people involved in the process?
4. What are the roles of the people involved?
5. What is the role hierarchy of these roles (chain of command)?
6. What are the systems involved in this process and who is responsible for these systems?
7. What kind of information security concerns do you have in relation? to this process?
8. Are there any access control restrictions related to systems involved in this process?

Access Administration Process

1. What are the steps involved in Security administration?
2. Who are the people involved in the process?
3. Are there task constraints involved in actioning any of the process?
4. What are the challenges around security administration of a request?
5. What are the systems involved in this process and who is responsible for these systems?

There were number of follow up workshop conducted and the below workflow of access governance process were presented to ensure requirements were captured accurately.

Workflow of New User Request

Workflow of Role Transfer

Workflow of Termination

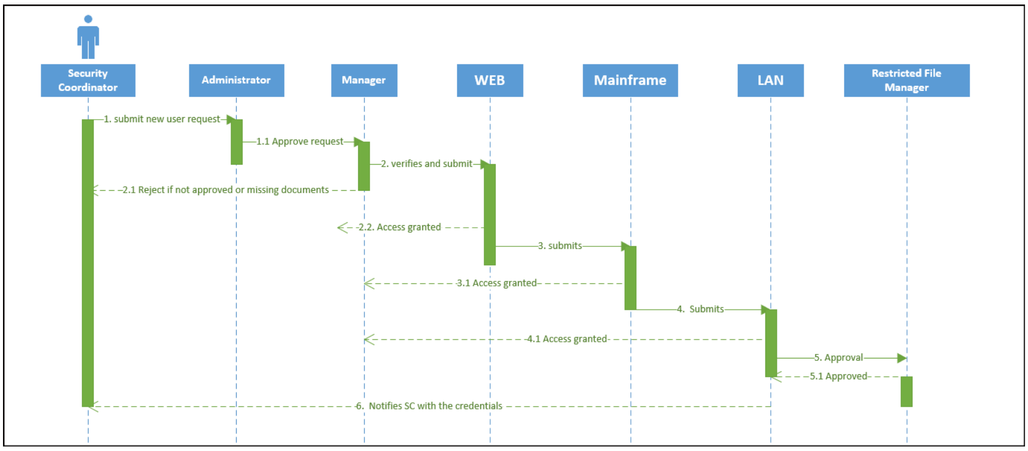


Figure A.1: Sequence diagram showing new user request workflow

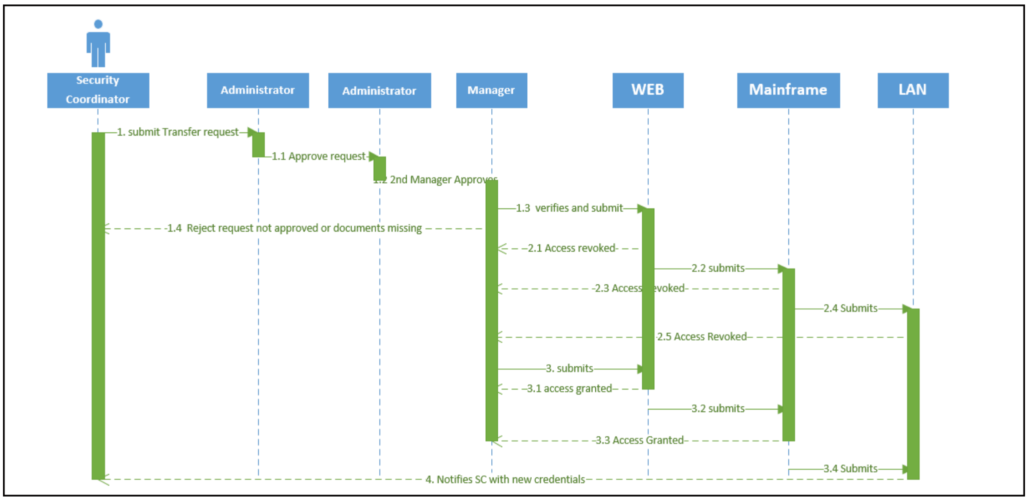


Figure A.2: Sequence diagram showing role transfer request workflow

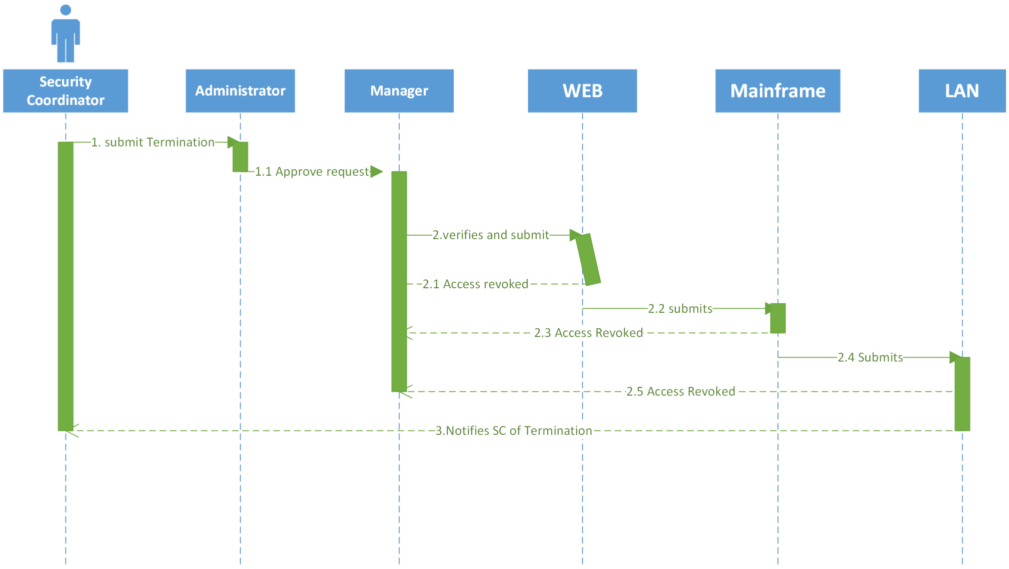


Figure A.3: Sequence diagram showing role termination request workflow

Bibliography

Mohamad Raja Gani Mohamad Abdul and Gregg Wilson. User authentication using kerberos with identity cloud service, October 22 2019. US Patent 10,454,915.

AmirHosein Adavoudi-Jolfaei, Maede Ashouri-Talouki, and 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*, 12(1):43–59, 2019.

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.

Gail-Joon Ahn. User-portable device and method of use in a user-centric identity management system, July 9 2019. US Patent 10,348,769.

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.

Sami Almalki. Integrating quantitative and qualitative data in mixed methods research–challenges and benefits. *Journal of Education and Learning*, 5(3):288–296, 2016.

Gergely Alpár, Jaap-Henk Hoepman, and Johanneke Siljee. The identity crisis. security, privacy and usability issues in identity management. *arXiv preprint arXiv:1101.0427*, 2011.

Waleed A Alrodhan and Chris J Mitchell. Addressing privacy issues in cardspace. In *Third International Symposium on Information Assurance and Security*, pages 285–291. IEEE, 2007.

Carol Beaumier. Identity and access management in financial services – staying ahead of the curve. *Protiviti Blog. https://blog.protiviti.com/2019/04/01/identity-and-access-management-in-financial-services-staying-ahead-of-the-curve/*, 2019.

Elisa Bertino, Elena Ferrari, and Vijayalakshmi Atluri. A flexible model supporting the specification and enforcement of role-based authorization in workflow management systems. In *ACM Workshop on Role-based Access Control*, pages 1–12, 1997.

Prosunjit Biswas, Ravi Sandhu, and Ram Krishnan. Label-based access control: An abac model with enumerated authorization policy. In *Proceedings of the 2016 ACM International Workshop on Attribute Based Access Control*, pages 1–12. ACM, 2016.

Reinhardt A. Botha and Jan H. P. Eloff. Separation of duties for access control enforcement in workflow environments. *IBM Systems Journal*, 40(3):666–682, 2001.

Marco Brambilla, Piero Fraternali, and Carmen Vaca. A notation for supporting social business process modeling. In *International Workshop on Business Process Modeling Notation*, pages 88–102. Springer, 2011.

T. Bray, J. Paoli, C.M Sperberg-McQueen, E. Maler, and F. Yergeau. Extensible markup language (xml). *World Wide Web Consortium Recommendation REC-xml-19980210*, 1998. URL http://www.w3.org/TR/1998/REC-xml-19980210.

Alan Bryman. *Social research methods*. Oxford university press, 2016.

Tracey Caldwell. Security at the data level. *Network Security*, 2013(5):6–12, 2013.

Jan Camenisch, Manu Drijvers, Petr Dzurenda, and Jan Hajny. Fast keyed-verification anonymous credentials on standard smart cards. In *IFIP International Conference on ICT Systems Security and Privacy Protection*, pages 286–298. Springer, 2019.

Scott Cantor and T Scavo. Shibboleth architecture. *Protocols and Profiles*, 10:16, 2005.

Scott Cantor, Internet2 Jahan Moreh, Sigaba Rob Philpott, and Eve Maler. Metadata for the oasis security assertion markup language (saml) v2. 0, 2005.

Irene Celino, Ana Karla Alves de Medeiros, Gernot Zeissler, Michael Oppitz, Federico Michele Facca, and Stefan Zoeller. Semantic business process analysis. In *SBPM*, 2007.

David W Chadwick, Wensheng Xu, Sassa Otenko, Romain Laborde, and Bassem Nasser. Multi-session separation of duties (msod) for rbac. In *2007 IEEE 23rd International Conference on Data Engineering Workshop*, pages 744–753. IEEE, 2007.

Liang Chen, Luca Gasparini, and Timothy J Norman. Xacml and risk-aware access control. *Resource*, 2(10):3–5, 2013.

Jason Crampton. On the satisfiability of constraints in workflow systems, 2004.

Ferdinand Damon and Marijke Coetzee. Towards a generic identity and access assurance model by component analysis-a conceptual review. In *Proceedings of the First International Conference on Enterprise Systems: ES 2013*, pages 1–11. IEEE, 2013.

Yuri Demchenko, Canh Ngo, Cees de Laat, and Craig Lee. Federated access control in heterogeneous intercloud environment: Basic models and architecture patterns. In *2014 IEEE International Conference on Cloud Engineering*, pages 439–445. IEEE, 2014.

Shlomi Dinoor. Privileged identity management: securing the enterprise. *Network Security*, 2010(12):4–6, 2010.

Quang Do, Ben Martini, and Kim-Kwang Raymond Choo. The role of the adversary model in applied security research. *Computers & Security*, 2018.

Marlon Cordeiro Domenech, Eros Comunello, and Michelle Silva Wangham. Identity management in e-health: A case study of web of things application using openid connect. In *2014 IEEE 16th International Conference on e-Health Networking, Applications and Services (Healthcom)*, pages 219–224. IEEE, 2014.

Cath Everett. Identity and access management: the second wave. *Computer Fraud & Security*, 2011(5):11–13, 2011.

Mohammad Faraji, Joon-Myung Kang, Hadi Bannazadeh, and Alberto Leon-Garcia. Identity access management for multi-tier cloud infrastructures. In *2014 IEEE Network Operations and Management Symposium (NOMS)*, pages 1–9. IEEE, 2014.

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.

Md Sadek Ferdous and Ron Poet. Dynamic identity federation using security assertion markup language (saml). In *IFIP Working Conference on Policies and Research in Identity Management*, pages 131–146. Springer, 2013.

D.F Ferraiolo, R. Sandhu, S. Gavrila, D.R Kuhn, and R. Chandramouli. Proposed nist standard for role-based access control. *ACM Trans. Inf. Syst. Secur.*, 4(3):224–274, August 2001. ISSN 1094-9224. doi:10.1145/501978.501980. URL `http://doi.acm.org/10.1145/501978.501980`.

Joseph Fialli and Sekhar Vajjhala. The java architecture for xml binding (jaxb). *JSR Specification, January*, 2003.

Ludwig Fuchs, Günther Pernul, and Ravi Sandhu. Roles in information security–a survey and classification of the research area. *computers & security*, 30(8):748–769, 2011.

Gartner. Gartner forecasts worldwide information security spending to exceed $124 billion in 2019. `https://www.gartner.com/en/newsroom/press-releases/2018-08-15-gartner-forecasts-worldwide-information-security-spending-to-exceed-124-b` 2019.

J. Gosling. *The Java language specification.* Addison-Wesley Professional, 2000.

Umme Habiba, Abdul Ghafoor Abassi, Rahat Masood, and Muhammad Awais Shibli. Assessment criteria for cloud identity management systems. In *2013 IEEE 19th Pacific Rim International Symposium on Dependable Computing*, pages 188–195. IEEE, 2013.

Eran Hammer-Lahav. The oauth 1.0 protocol. Technical report, 2010.

Dick Hardt. The oauth 2.0 authorization framework. 2012.

Jason Hart. Why the traditional approach to information security is no longer working. *Network Security*, 2013(1):12–14, 2013.

Hao Jiang and Shengye Lu. Access control for workflow environment: The rtfw model. In *International Conference on Computer Supported Cooperative Work in Design*, pages 619–626. Springer, 2006.

Dong-sheng JING and Ji-wen YANG. A model of task-role based access control and its application [j]. *Computer Technology and Development*, 2, 2006.

Rieks Joosten, Diane Whitehouse, and Penny Duquenoy. Towards a meta model for identity terminology. *Pre-proceedings of the IFIP/FIDIS Internet Security & Privacy Summer School*, pages 1–7, 2008.

Audun Jøsang, John Fabre, Brian Hay, James Dalziel, and Simon Pope. Trust requirements in identity management. In *Proceedings of the 2005 Australasian workshop on Grid computing and e-research-Volume 44*, pages 99–108. Australian Computer Society, Inc., 2005.

Anthony D JoSEP, RAnDy KAtz, AnDy KonWinSKi, LEE Gunho, DAViD PAttERSon, and ARiEL RABKin. A view of cloud computing. *Communications of the ACM*, 53 (4), 2010.

Balachandra Reddy Kandukuri, Atanu Rakshit, et al. Cloud security issues. In *2009 IEEE International Conference on Services Computing*, pages 517–520. IEEE, 2009.

Laura Kankaala et al. Identity federation using shibboleth identity provider. 2015.

Ahammad Karim and Muhammad Abdullah Adnan. An openid based authentication service mechanisms for internet of things. In *2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS)*, pages 687–692. IEEE, 2019.

A. Keith and A. Ole. A comparison of software and hardware techniques for x86 virtualization. In *Proceedings of the 12th international conference on Architectural support for programming languages and operating systems*, ASPLOS XII, pages 2–13, New York, NY, USA, 2006. ACM. ISBN 1-59593-451-0. doi:10.1145/1168857.1168860. URL http://doi.acm.org/10.1145/1168857.1168860.

Salim Khamadja, Kamel Adi, and Luigi Logrippo. Designing flexible access control models for the cloud. In *Proceedings of the 6th International Conference on Security of Information and Networks*, pages 225–232. ACM, 2013.

Konstantin Knorr and Henrik Stormer. Modeling and analyzing separation of duties in workflow environments. In *IFIP International Information Security Conference*, pages 199–212. Springer, 2001.

Michael Koch and Wolfgang Wörndl. Community support and identity management. In *ECSCW 2001*, pages 319–338. Springer, 2001.

Srinivasan Madhan Kumar and Paul Rodrigues. A roadmap for the comparison of identity management solutions based on state-of-the-art idm taxonomies. In *International Conference on Network Security and Applications*, pages 349–358. Springer, 2010.

Michael Kunz, Matthias Hummer, Ludwig Fuchs, Michael Netter, and Günther Pernul. Analyzing recent trends in enterprise identity management. In *2014 25th international workshop on database and expert systems applications*, pages 273–277. IEEE, 2014.

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.

Nick Lewis. Access rights–protect access to your data or lose it: serious misconceptions about information security. *Computer Fraud & Security*, 2012(11):8–10, 2012.

Hongjun Li. Restful web service frameworks in java. In *2011 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*, pages 1–4. IEEE, 2011.

Fei Liu, Jing Wang, Hongtao Bai, and Huiping Sun. Access control model based on trust and risk evaluation in idmaas. In *2015 12th International Conference on Information Technology-New Generations*, pages 179–184. IEEE, 2015.

Lianzhong Liu, Xuanyu Liu, and Xiaomei Tang. A resource-oriented model for identity provisioning and its application. In *IEEE Conference Anthology*, pages 1–4. IEEE, 2013.

Shengye Lu and Hao Jiang. Rtfw: an access control model for workflow environment. In *2006 10th International Conference on Computer Supported Cooperative Work in Design*, pages 1–5. IEEE, 2006.

Gang Ma, Kehe Wu, Tong Zhang, and Wei Li. A flexible policy-based access control model for workflow management systems. In *2011 IEEE International Conference on Computer Science and Automation Engineering*, volume 2, pages 533–537. IEEE, 2011.

Eve Maler and Drummond Reed. The venn of identity: Options and issues in federated identity management. *IEEE Security & Privacy*, 6(2):16–23, 2008.

MarketWatch. Identity and access management (iam) market size to surpass usd 24.52 billion by 2025. *Brandessence Market Research via COMTEX. https://www.marketwatch.com/press-release/*, 2019.

James A. Martin. What is access control? a key component of data security. *CSO. https://www.csoonline.com/article/3251714/what-is-access-control-a-key-component-of-data-security.html*, 2019.

Calif Menlo Park. Does cybersecurity have an identity crisis? the iam annual report card answers. https://cybersecurityventures.com/identity-and-access-management-report/, 2017.

Nkosinathi Mpofu and Wynand Jc Van Staden. A survey of trust issues constraining the growth of identity management-as-a-service (idmaas). In *2014 Information Security for South Africa*, pages 1–6. IEEE, 2014.

Gerry Murray. Idc marketscape: Worldwide ai in enterprise marketing clouds 2019–2020 vendor assessment. *IDC MarketScape. https://www.idc.com/getdoc.jsp?containerId=US45719919*, 2019.

Pinki Nida, Harsh Dhiman, and Shahnawaz Hussain. A survey on identity and access management in cloud computing. *Int. J. Eng. Res. Technol*, 3(4), 2014.

M Noureddine and R Bashroush. A provisioning model towards oauth 2.0 performance optimization. In *2011 IEEE 10th International Conference on Cybernetic Intelligent Systems (CIS)*, pages 76–80. IEEE, 2011.

A. O'Connor and R. Loomis. Economic analysis of role-based access control. Technical report, RTI International, 2010.

Sejong Oh and Seog Park. Task-role based access control (t-rbac): An improved access control model for enterprise environment. In *International Conference on Database and Expert Systems Applications*, pages 264–273. Springer, 2000.

Top OWASP. Top 10-2017 the ten most critical web application security risks. *URL: owasp. org/images/7/72/OWASP_ Top_ 10-2017_ % 28en*, 29, 2017.

Christian Paquin. U-prove technology overview v1. 1 (revision 2). *Microsoft, Apr*, 2013.

Quentin Perez, Alexandre Le Borgne, Christelle Urtado, and Sylvain Vauttier. An empirical study about software architecture configuration practices with the java spring framework. 2019.

Andreas Pfitzmann and Katrin Borcea-Pfitzmann. Lifelong privacy: Privacy and identity management for life. In *IFIP PrimeLife International Summer School on Privacy and Identity Management for Life*, pages 1–17. Springer, 2009.

PWC. 2015 iformation security breaches survey. *Technical Report. https://www.pwc.co.uk/assets/pdf/2015-isbs-technical-report-blue-digital.pdf*, 2015.

Dan Rafter. 2019 data breaches: 4 billion records breached so far. *Emerging Threats https://us.norton.com/internetsecurity-emerging-threats-2019-data-breaches.html*, 2019.

Qasim Mahmood Rajpoot, Christian Damsgaard Jensen, and Ram Krishnan. Integrating attributes into role-based access control. In *IFIP Annual Conference on Data and Applications Security and Privacy*, pages 242–249. Springer, 2015.

HF Ravi Sandhu, E Coyne, and Charles Youman. Role-based access control models. *IEEE Comput*, 29(2):38–47, 1996.

David Recordon and Drummond Reed. Openid 2.0: a platform for user-centric identity management. In *Proceedings of the second ACM workshop on Digital identity management*, pages 11–16. ACM, 2006.

Hyeun-Suk Rhee, Young U Ryu, and Cheong-Tag Kim. Unrealistic optimism on information security management. *Computers & Security*, 31(2):221–232, 2012.

E. Rissanen. extensible access control markup language (xacml) version 3.0 (committe specification 01). Technical report, Technical report, OASIS, 2010. URL http://docs.oasisopen.org/xacml/3.0/xacml-3.0-core-spec-cd-03-en.pdf.

Erik Rissanen et al. extensible access control markup language (xacml) version 3.0. *OASIS standard*, 22, 2013.

John W Rittinghouse and James F Ransome. *Cloud computing: implementation, management, and security*. CRC press, 2017.

Draft N Sakimura, J Bradley, Ping Identity, M Jones, B de Medeiros, and C Mortimore. Openid connect basic client implementer's guide 1.0-draft 37. *Specification, OpenID Foundation, August*, 2015.

Ravi Sandhu, David Ferraiolo, Richard Kuhn, et al. The nist model for role-based access control: towards a unified standard. In *ACM workshop on Role-based access control*, volume 10, 2000.

R.S. Sandhu, E.J. Coyne, H.L. Feinstein, and C.E. Youman. Role-based access control models. *Computer*, 29(2):38–47, 1996. ISSN 0018-9162. doi:http://doi.ieeecomputersociety.org/10.1109/2.485845.

José Simão, Sérgio Esteves, Andre Pires, and Luís Veiga. Gc-wise: A self-adaptive approach for memory-performance efficiency in java vms. *Future Generation Computer Systems*, 100:674–688, 2019.

Sebastian Stein, Nicholas R Jennings, and Terry R Payne. Provisioning heterogeneous and unreliable providers for service workflows. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, page 78. ACM, 2007.

Mark Strembeck and Jan Mendling. Modeling process-related rbac models with extended uml activity models. *Information and Software Technology*, 53(5):456–483, 2011.

Gregory Tassey, Michael P Gallaher, Alan C O'Connor, and Brian Kropp. The economic impact of role-based access control. *Economic Analysis*, 2002.

Roshan K Thomas and Ravi S Sandhu. Task-based authorization controls (tbac): A family of models for active and enterprise-oriented authorization management. In *Database Security XI*, pages 166–181. Springer, 1998.

Mumina Uddin and David Preston. Systematic review of identity access management in information security. *Journal of Advances in Computer Networks*, 3(2), 2015.

R. Uhlig, G. Neiger, D. Rodgers, A.L. Santoni, F.C.M. Martins, A.V. Anderson, S.M. Bennett, A. Kagi, F.H. Leung, and L. Smith. Intel virtualization technology. *Computer*, 38(5):48–56, May 2005. ISSN 0018-9162. doi:10.1109/MC.2005.163.

Varonis. Varonis global data risk report. *World Wide Web Consortium Recommendation REC-xml-names-19990114. https://www.varonis.com/2019-data-risk-report/*, 2019.

Verizon. 2019 data breach investigations report. *verizon. https://enterprise.verizon.com/en-gb/resources/reports/dbir/*, 2019.

J. Voas and J. Zhang. Cloud computing: New wine or just a new bottle? *IT Professional*, 11(2):15–17, 2009. ISSN 1520-9202. doi:http://doi.ieeecomputersociety.org/10.1109/MITP.2009.23.

Jacques Wainer, Paulo Barthelmess, and Akhil Kumar. W-rbac—a workflow security model incorporating controlled overriding of constraints. *International Journal of Cooperative Information Systems*, 12(04):455–485, 2003.

Lingyu Wang, Duminda Wijesekera, and Sushil Jajodia. A logic-based framework for attribute based access control. In *Proceedings of the 2004 ACM workshop on Formal methods in security engineering*, pages 45–55. ACM, 2004.

Michael Waters. Evaluating identity and access management (iam) as a cloud service. 2016.

Barbara Weber, Manfred Reichert, Werner Wild, and Stefanie Rinderle. Balancing flexibility and security in adaptive process management systems. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 59–76. Springer, 2005.

Peter White, Irfan Altas, Jason Howarth, and John Weckert. An internal enterprise framework for identity based management. In *Australian Partnership for Advanced Computing 07*. 2007.

Phillip J Windley. *Digital Identity: Unmasking identity management architecture (IMA)*. " O'Reilly Media, Inc.", 2005.