

Designing Identity Access Mechanism for Broker-Based Cross-Cloud
Federation System

A Thesis
presented to
the Faculty of Natural and Applied Sciences
At Notre Dame University-Louaize

In Partial Fulfillment
Of the Requirements for the Degree
Master of Science

By
NADER A. GEMAYEL

MAY 2018

© COPYRIGHT

By

Nader A. Gemayel

2018

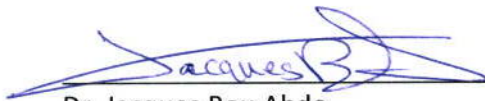
All Rights Reserved

Notre Dame University - Louaize
Faculty of Natural and Applied Science
Department of Computer Science

We hereby approve the thesis of

Nader A. Gemayel

Candidate for the degree of Master of Science in Computer Science



Dr. Jacques Bou Abdo

Assistant professor of Computer Science, Advisor



Dr. Marie Khair

Associate professor of Computer Science, Member of Committee



Dr. Hikmat Farhat

Associate professor of Computer Science, Member of Committee

Declaration

This thesis has been created solely by myself and that it has not been submitted, in whole or in part, in any previous application for a degree. Except where states otherwise by reference or acknowledgment, the work presented is entirely my own.

Acknowledgments

I would like to thank my thesis advisor Dr. Jacques Bou Abdo and chairperson Dr. Hoda Maalouf of the faculty of applied science at Notre Dame University;

Dr. Bou Abdo was a great supporter, teacher and guider who paved all the way to reach this great ending. He gave lot of enhancements on all academic and technical levels where all difficulties and obstacles were found but solved at last!

I would also like to thanks all the experts who were involved in the technical and advisory phase in this project:

Mr. Osman Omer – IBM lead technical support EMEA

Mr. David Brossard – AXIOMATICS Int'l USA

Mr. Paul Colmer - AXIOMATICS Int'l USA

Dr. Daniel Dos Santos - SAP Labs France

Finally, I would like to express my profound gratitude to my parents for providing me their gentle and amazing support throughout the hard and failing times I had during this research.

THANK YOU

Nader Gemayel

Table of contents

Declaration	3
Acknowledgment	4
Table of contents	5
List of figures	8
List of tables	9
List of code listings	10
List of abbreviations	11
Abstract	14
CHAPTER 1. INTRODUCTION	15
1.1 Problem definition	15
1.2 Research Objectives	17
1.3 Thesis organization	19
CHAPTER 2. STATE OF THE ART	20
2.1 Introduction	20
2.2 Cloud computing	20
2.2.1 Types and models of Cloud computing	21
2.2.1.1 Infrastructure as a service (IaaS)	21
2.2.1.2 Platform as a service (PaaS)	21
2.2.1.3 Software as a service (SaaS)	22
2.3 Cloud federation.....	22
2.3.1 Cloud federation architecture	23
2.3.2 Cloud federation benefits	24
2.4 Broker-based services	25
2.4.1 Broker-based server architecture	25
2.4.1.1 Conversations (dialogs)	26
2.4.1.2 Message Ordering and Coordination.....	26
2.4.1.3 Transactional Asynchronous Programming.....	27
2.4.1.4 Loosely Coupled Applications.....	28

2.4.1.5 Service Broker Components	28
2.4.2 Broker-based server benefits.....	29
2.4.3 Broker-based service in cloud computing	30
2.4.3.1 Definition of Cloud Services Brokerage	30
2.4.3.2 Types of Cloud Brokers	30
2.4.4 Broker-based cross-cloud federation manager (BBCCFM)	31
2.5 Firewall and cloud-based firewall.....	32
2.5.1 Proxy firewall.....	33
2.5.2 State-full inspection firewall	34
2.5.3 Next-generation firewall	34
2.5.4 Threat-focused Next Generation firewall	34
2.5.5 Cloud based firewalls	35
2.6 Proxy servers	36
2.7 Security Token Services (STS)	37
2.8 Authentication and authorization	38
2.9 Chapter Summary.....	39
CHAPTER 3. Access Control and Identity Management Mechanisms	41
3.1 Introduction.....	41
3.2 Access control and identity management mechanisms.....	42
3.2.1 Mandatory Access Control (MAC)	42
3.2.2 Discretionary Access Control (DAC)	44
3.2.3 Role Based Access Control (RBAC)	45
3.2.3.1 Definition and main components	45
3.2.3.2 RBAC in Operating systems: UNIX based AIX.....	48
3.2.3.3 RBAC architecture.....	52
3.2.3.4 Benefits of RBAC.....	55
3.2.4 Attribute Based Access Control (ABAC)	57
3.2.4.1 Definition and main components	57
3.2.4.2 Benefits of ABAC.....	60
3.2.4.3 Implementing ABAC.....	62
3.2.4.4 Deploy the architecture.....	67
3.2.4.5 eXtensible Access Control Markup Language.....	69
3.2.5 Federated Identity Management (FIM)	71
3.2.5.1 Definition and main components	71

3.2.5.2 Benefits of FIM.....	75
3.2.5.3 FIM architecture	78
3.2.5.4 SAML and OAuth in federated identify.....	81
3.2.5.5 OAuth.....	90
3.2.6 Content Based Access Control (CBAC)	95
3.2.7 Policy Based Access Control (PBAC)	98
3.3 Chapter summary	99
CHAPTER 4. Evaluation of Access Control Mechanisms	101
4.1 Introduction.....	101
4.2 Evaluate and compare IAM mechanisms	101
4.2.1 Mandatory Access Control (MAC)	101
4.2.2 Discretionary Access Control (DAC)	102
4.2.3 Role Based Access Control (RBAC)	103
4.2.4 Policy Based Access Control (PBAC)	104
4.2.5 Attribute Based Access Control (ABAC)	105
4.2.6 Federated Identity Management (FIM)	122
4.3 Integration of FIM and TEMCM.....	125
4.4 Chapter summary	128
CHAPTER 5. Conclusion	130
5.1 Summary of main results.....	130
5.2 Main contribution of the thesis.....	130
5.3 Possible extension of future work	131
Bibliography	132

List of figures

Fig. 1 Broker Server inside cloud federation systems.....	24
Fig. 2 Message Ordering and Coordination inside broker server	27
Fig. 3 Traditional firewall architecture inside LAN network	33
Fig. 4 Traditional Cloud-based firewall architecture inside LAN network	36
Fig. 5 Traditional proxy server in action.....	37
Fig. 6 Identity Provider STS (IP-STS) and a Relying Party STS (RP-STS)	38
Fig. 7 Hierarchy of elements inside RBAC	47
Fig. 8 Role creation and authorization assignment inside AIX	51
Fig. 9 Authorization and role testing	52
Fig. 10 Interconnection of elements inside RBAC	52
Fig. 11 RBAC work flow and decision making	54
Fig. 12 Elements inside ABAC system	67
Fig. 13 Elements and message flow inside ABAC system.....	69
Fig. 14 Traditional SAML system sequence diagram.....	87
Fig. 15 Traditional SAML architecture flow diagram.....	88
Fig. 16 Traditional system sequence OAuth flow diagram	91
Fig. 17 Traditional SAML and OAuth combination’s system sequence diagram	94
Fig. 18 Traditional FIM inside BBCCFM	125
Fig. 19 Trust Establishment, Management and Continuous Monitoring (TEMCM)	126
Fig. 20 (TEMCM) combined with FIM inside BBCCFM	127

List of table

Table 1 Roles, groups and authorizations inside AIX	48
Table 2 Authorizations and definitions inside AIX	50
Table 3 ABAC model comparison - General architecture	118
Table 4 ABAC model comparison - Domain architecture (2005-2010)	120
Table 5 ABAC model comparison - Domain architecture (2010-2014)	121
Table 6 FIM model comparison based on multiple factors	124
Table 7 Comparison table: MAC, DAC, RBAC, ABAC and FIM	129

List of code listings

Code 1 Namespaces sample for customer and employee.....	63
Code 2 General SAML request in XML format	89
Code 3 General SAML response in XML format.....	90

List of abbreviations

CRM : Content Resource Management

ISP : Internet Service Provider

SLA : Service Level Agreement

CSP : Cloud Service Provider

IAM : Identity Access Management

BCCCFM : Broker-Based Cross-Cloud Federation Manager

IaaS : Identity as a Service

PaaS : Platform as a Service

SaaS : Software as a Service

QoS : Quality of Service

FLA : Federation Level Agreement

UDDI : Universal Description, Discovery, and Integration

NGFW : Next Generation Firewall

STS : Security Token Service

RP-STS : Relaying Party Security Token Service

IP-STS : Initiator Party Security Token Service

RP : Relaying Point

FIM : Federated Identity Management

IP : Internet Protocol

MAC address : Media Access Control

MAC : Mandatory Access Control

DAC : Discretionary Access Control

RBAC : Role-Based Access Control

ABAC : Attribute-Based Access Control

CBAC : Content-Based Access Control

PBAC : Policy-Based Access Control

NIST : National Institute of Standards and Technology

SAML : Security Assertion Markup Language

XACML : eXtensible Access Control Markup Language

PEP : Policy Enforcement Point

PDP : Policy Decision Point

PAP : Policy Administration Point

PIP : Policy Information Point

SoR : System of record

API : Application programming interface

OASIS : On-line Applicant Status and Information System

SSO : Single Sign On

LDAP : Lightweight Directory Access Protocol

SP : Service Provider

IdP : Identity Provider

CoT : Circle of Trust

AAL : Authentication Assurance Level

TSP : Trust Service Provider

ETIS : Efficient Trust and Identity Management System

WSP : Web Service Provider

OAuth : Open Authorization

XML : Extensible Markup Language

HTTP : Hyper Text Transfer Protocol

SOAP : Simple Object Access Protocol

JSON : JavaScript Object Notation

XEEE : XML External Entity

XSD : XML Schema Definition

AAS : Authoritative Attribute Source

PIN : Personal Identification Number

DB : Database

ACML : Asian Conference on Machine Learning

ABAM : Attribute-Based Access Matrix

ANAC : Application Network Access Control

ABCL : Attribute-Based constraint specification language

SOD : Separation of duties

TEMCM : Trust Establishment Management and Continuous Monitoring

Abstract

Cloud computing ecosystem is one of the trends currently being discussed in the domain of information technology and computing. Computing ecosystem allows smart usage of technology which means unneeded resources are saved such as power, computing resources, spaces for datacenters as well as saving huge amount of CO₂ and electric circuit boards and machines. Cloud federation and its underlying techniques, is considered role model in sharing and expanding cloud business to provide better and cheaper services for end-users. In this thesis, we will survey and compare existing Identity Access Management (IAM) mechanisms and select the IAM best suiting the cloud federation scenario. Our contribution is mainly survey-based where we have compared and selected the best IAM, and design-based in terms of proposing and designing new module: Trust Establishment, Management and Continuous Monitoring "TEMCM" to allow scalable, secure and efficient access control in Broker-Based Cross-Cloud Federation Manager "BBCCFM", one of the most developed Cloud Federation architecture found in literature. TEMCM will enforce trust management and continuous monitoring which are currently missing in the selected IAM models.

Keywords: Cloud, federation, security, IAM, Cross-Cloud, BBCCFM, TEMCM

CHAPTER 1. Introduction

1.1 Problem definition

With the wide categories of apps moving to the cloud, we can explain the reason why the future of the cloud is going to be federated: Social media, e-commerce, CRM, web, gaming, etc. All these applications need truly global coverage where ultra-latency is the major problem facing the end user. The cloud has always promised to be the solution for this problem. One single provider, no matter how large it is, can not satisfy the end users' needs. Market giants such as Amazon, invest with infrastructure where it's profitable for them. The limited geographic presence of many Cloud Service Providers makes the coverage from today's "global" cloud providers weak and reduced. The imbalance in infrastructure, between the cities in the same country and between the different countries as well, acts as a barrier for a global and instant coverage. For example, in the US, the closest access point to one's business is neither in the same city nor the same state. Actually, infrastructure exists in different and dispersed locations. There are data center operators, hosting providers and ISPs almost everywhere. Companies need not be worried about resources since resources can be easily found, rent or bought.

When IT companies have the way to pool all their capacities on one massive pool of cloud resources and make it available to anyone who needs it, we can say the target of “federated cloud” was reached.

As a definition, “federated cloud” takes advantage from this geographically dispersed infrastructure in order to finally deliver the promise of the cloud. These local infrastructure providers are connected to a global marketplace through the federated cloud. In this market, each participant is able to buy and sell capacity on demand. From a provider perspective, customers have an instant and wide access to the global infrastructure, and in case of a sudden need of few new hundred servers, they can afford it by buying just the required resources.

For example, if a marketing firm needs to accelerate a website in Tokyo or Hong Kong, it only has to subscribe to those locations and benefit from the existing infrastructure.

Without building a new huge infrastructure, small service providers as part of a cloud federation can offer an effective global service. Companies with spare resources such as platforms, infrastructures and other types of resources, can at any time transfer these available resources to the marketplace through federation to be used by end users. This simple way creates an additional source of revenue for the initial service provider where end users have immediate benefits. Users can receive data from their federated cloud

provider of choice; In other words, cloud users can choose a local host that fits their needs without referring to other “global” cloud providers on the market. End users have nothing to do with pricing, app support and Service Level Agreements (SLA). End users benefit from a wide range of cloud providers without having to manage multiple invoices and other supports. The federated cloud is a real example of the globalization in the cloud market. This model enables the businesses to use local cloud providers in order to connect with their third parties (customers, partners and employees...) all over the world. End users will finally realize the potential of the cloud, and data center operators and other service providers will be also able to compete with, and beat today’s so-called global cloud providers. Now, what makes federated solutions or any IT system secure and easily accessible? The answer is simple: The underlying backbone architecture, especially access management mechanisms.

1.2 Research Objectives

Designing and implementing access control mechanism is a very complicated and critical task which almost all organizations need to manage. For cloud-based systems, security access mechanisms and access control policies have another level of security in all underlying aspects and modules.

In other terms, cloud based systems require a well-designed and defined set of security rules and access control mechanisms built and tested in a very professional and rigorous way, because putting and getting data in the cloud is not an easy process, especially with the emerging security breaches and hacking activities that are spread all around the world.

Cloud based systems should have a very secure environment and mechanism to allow maximum level of integrity, scalability, privacy and availability to their clients and users. Existing IAM mechanisms will be surveyed and compared with highlights on the pros and cons of each access control, especially for cloud based systems. Later on, we will step forward to another level of access control needs that should meet the requirements of cross cloud federation integrated with broker-based system.

J. Bou Abdo et Al. [1] proposed a Broker-based cross-cloud federation manager (BBCCFM) which is considered an enhanced mode of normal federation managers.

Every (Cloud Service Provider) CSP maintains the identification, authentication, authorization and accountability of users trying to access its resources, even if these users belong to another federated CSP through Identity and Access Management (IAM).

Access controls currently used by cloud-based systems do not satisfy the needs of BBCCFM (Broker based cross cloud federation manager). Hence we, in this research, will

select the best IAM mechanisms to be used in BBCCFM model. An extension will be made to enhance the selected mechanism so that BBCCFM requirements are met.

1.3 Thesis organization

The research conducted in this thesis is organized as follows: Definition, architecture and benefits of brokers, cloud computing and federation mechanisms. The next chapter presents surveys on the latest IAM mechanisms currently used in information security especially in cloud solutions. Later on, IAM mechanisms evaluation is conducted where it shows the pros and cons of each access control mechanism. In the last chapter, we select the best IAM mechanism to be implemented in BBCCFM and state whether it meets the requirements of BBCCFM or another proposal should be implemented.

CHAPTER 2. STATE OF THE ART

2.1 Introduction

In this chapter we describe Cloud Computing, Cloud Federation and IAM mechanisms and other concepts necessary to make this manuscript self-contained. Finally, we go over previous work done in the subject.

2.2 Cloud computing

Cloud computing or the cloud, is the concept of delivering on-demand computing resources (Platforms, infrastructure or software) through the internet on a pay-for-use basis. [2]

Cloud computing features may be as follows:

- Elastic or scalable resources: Increase or decrease easily based on customers' needs.
- Pay-for-use: Customer pay for used resources only.
- Self-service: Full access for all the IT resources.

In the next sections, we will describe existing types and models of cloud computing.

2.2.1 Types and models of Cloud computing

Cloud computing can be divided into tens of models, but the major ones are three, as follows:

2.2.1.1 Infrastructure as a service (IaaS)

Infrastructure as a service (IaaS) provides clients with computing resources such as servers, networking, storage, and data center space on a pay-per-use basis. [3]

IaaS benefits are as follows:

- Investments are not made on clients' private hardware
- Workloads increase, Infrastructure scales accordingly
- New and trendy services are always available.

2.2.1.2 Platform as a service (PaaS)

Platform as a service (PaaS) provides clients with a cloud-based environment to start a full lifecycle of building and delivering web-based (cloud) applications. This lowers the cost and complexity of buying and managing the underlying hardware, software, provisioning, and hosting.

PaaS benefits are as follows:

- Fast and easy application development.
- Fast deployment of new web applications to the cloud.

- Use of middleware as a service

2.2.1.3 Software as a service (SaaS)

Software as a Service (or SaaS) run on distant servers in the cloud that are owned and operated by private companies. SaaS are mainly accessible through the internet, usually, web browsers.

SaaS benefits are as follows:

- Business applications are ready and easily accessible.
- Any connected computer can access applications and data.
- Data is in the cloud, safe and secure in case of any computer breakings.
- Scalable service based on usage needs.

2.3 Cloud federation

Cloud federation enable power-efficient, cost-effective, dynamic sharing of unused cloud resources and services [4]. End users ensure that services they get are stable in terms of quality of service (QoS) and availability by signing service-level agreements (SLAs)

Cloud federation main purposes are as follows:

- Define clear marketing system to describe the cost of using resources and services.

- Allocate resources based on location of users to decrease network problems that could interrupt service access.
- Follow rules in a Federal-Level Agreement (FLA) describing the collaboration and association between participating clouds service providers

In cloud federation, cloud service providers participate voluntarily after signing an FLA.

2.3.1 Cloud federation architecture

In cloud federation, heterogeneous CSPs must be able to cooperate between each other, which is difficult to achieve. For example, CSPs might describe services they offer in different techniques and users need a protocol to access available services. For this reason, cloud federation architecture must have interface standards, a service broker that communicates between CSPs and users to bring updates on offered services and users' status changes. For the federation to run properly, all CSPs and users must sign an FLA that specifies cooperation rules and defines each participant's responsibilities and permissible behaviors, with all penalties for violating terms such as financial and administrative penalties.

Figure 1 shows the broker inside cloud federation playing a central role in the communication between all cooperating CSPs.

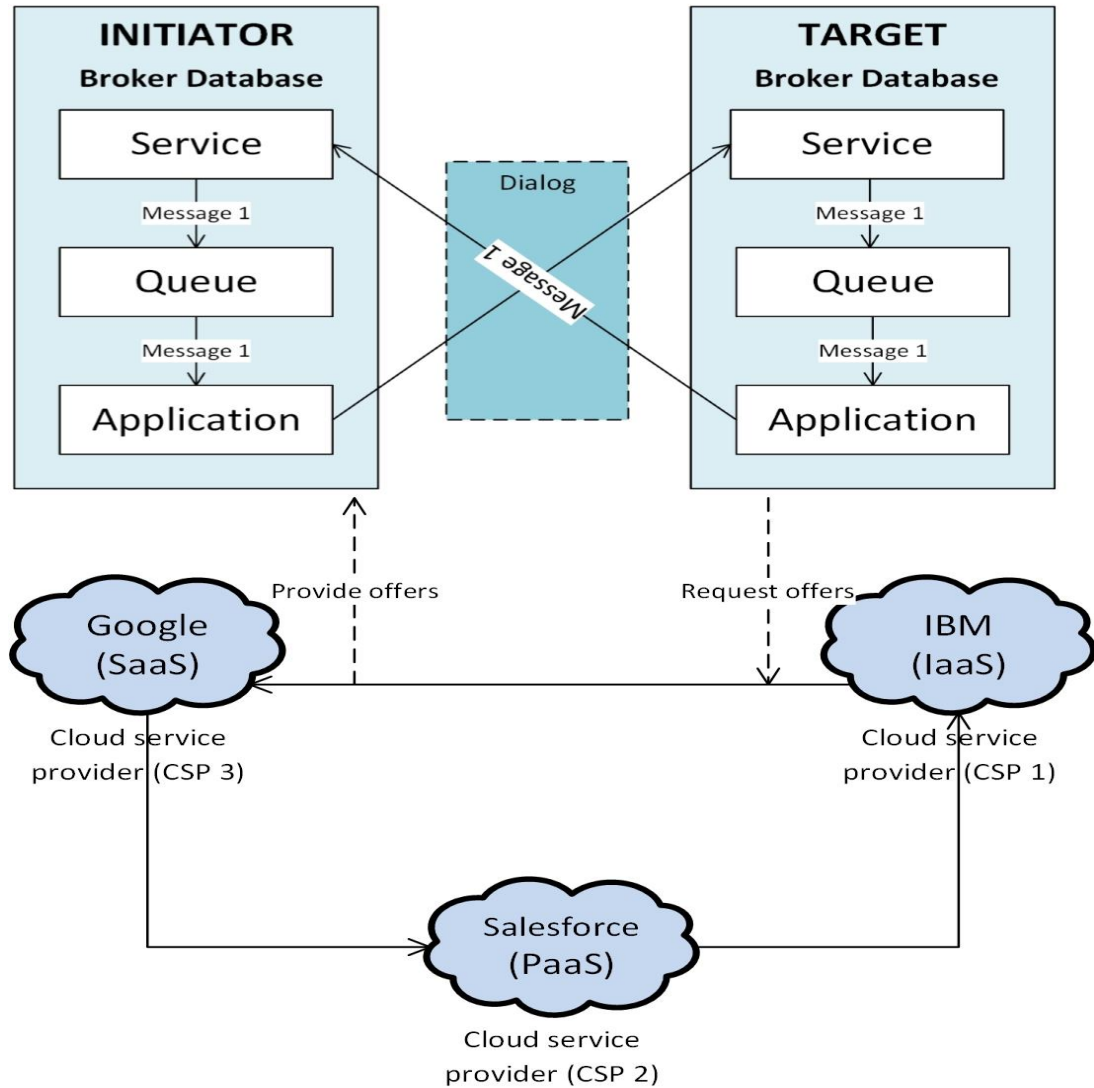


Fig. 1 Broker Server inside cloud federation systems

2.3.2 Cloud federation benefits

Cloud Federation performance is guaranteed by the dynamic resource allocation (or elasticity) which allows CSPs to coordinate between each other to share idle resources

[5]. This coordination allows smooth service delivery and resource scalability, which is based on the transparent operation between CSPs for the delivery of QoS level.

Cloud federation enables the geo-allocation of resources, so that users can get efficient resources based on their locations; In case of any shortage, distant resources are used to overcome the local shortage.

And because the FLA clearly describes what each participant is offering, as well as the federation's rules, it ensures the commitment of the involved parties to the operation's performance.

2.4 Broker-based services

Broker-Based server is considered as middle layer between clients and end-servers, for organizing messaging and conversations.

2.4.1 Broker-based server architecture

Message Service Broker

Service broker is used in building applications where independent components cooperate to exchange information needed to accomplish a task [6].

Service broker can be understood through 5 major aspects as follows:

2.4.1.1 Conversations (dialogs)

The main function of service broker is to exchange messages that establish a complete conversation that is considered a determined communication channel.

Service broker assures that all messages are sent and received only one time per transaction.

Inbound and outbound messages are protected by enhanced security with digital certification. The sent message to a service broker service will be isolated from the receiving application.

The receiving application can be rearranged, changed or even shut down. In this case, the service broker will keep on adding the messages to queue until we restart the receiving application.

2.4.1.2 Message Ordering and Coordination

Service broker database handles queuing, which has significant benefits over common databases: Service broker queues are integrated into the database for coordinating and ordering related messages.

The simple transact-SQL interface for exchanging messages is strongly combined with a set of guarantees for delivery, processing of messages and ensuring that messages are received no more than one time only (which is the most important part of the broker).

Figure 2 illustrates the exchange of messages in a typical dialog conversation inside the

Broker-Based server. The conversation has two sides as follows:

- The initiator side with initiator service and message queue
- The target side with target service and message queue

Each side has a service and each service has an associated message queue.

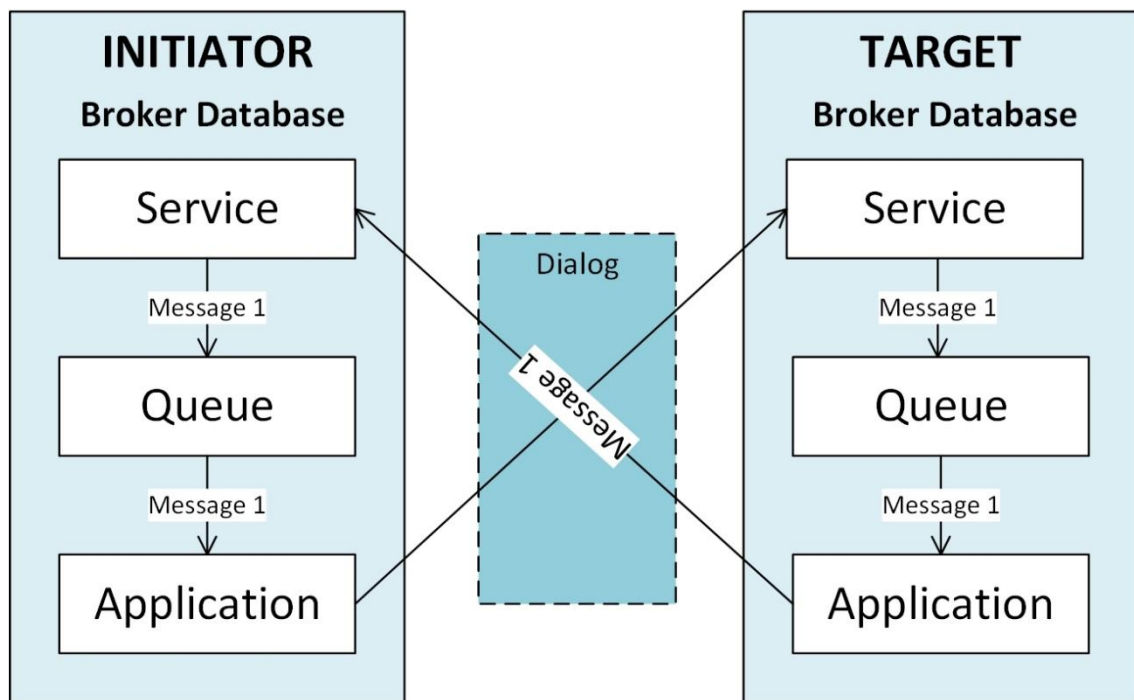


Fig. 2 Message Ordering and Coordination inside broker server

2.4.1.3 Transactional Asynchronous Programming

The message delivery between applications is:

- Transactional: if a transaction rolls back, all Service Broker operations in the transaction roll back

- Asynchronous: helps developers write applications with queuing

Queues can deliver two benefits to database applications:

When a user places the job request in a queue, the application responds immediately without waiting for all the jobs associated to be completed [7].

The job of a single request can be distributed into multiple units of work processed as separate transactions.

A request in a queue is placed by the database application for each unit.

2.4.1.4 Loosely Coupled Applications

Loosely coupled applications consist of multiple programs that exchange messages independently of each other. Exchanged messages in these applications hold the same definition during the transactions. Moreover, loosely coupled applications are not aware of the servers' physical location and database.

2.4.1.5 Service Broker Components

Service broker has three types of components:

- Conversation components

A service broker request structure is composed of Conversation groups, conversations, and messages. Messages are exchanged inside conversations and group of conversations. No message is processed alone.

- Service definition components

It specifies the basic structure needed for the application. Structure might contain attributes such as: message types, conversation flow and database specifications.

- Networking and security components

Define the organization to manage the messages between initiators, targets and their related database.

2.4.2 Broker-based server benefits

Benefits of service broker are as follows:

- Database integration: Integration on the database level improves performance and administration of application.
- Message ordering and coordination for simplified application development.
- Loose application coupling provides workload flexibility.
- Related message locking allows more than one instance of an application to process messages from the same queue without explicit synchronization.
- Automatic activation allows applications to scale with the message volume.

2.4.3 Broker-based service in cloud computing

2.4.3.1 Definition of Cloud Services Brokerage

Cloud servers consist of roles and business models where customers request specific services to be delivered on purpose and on time. Cloud brokering is done through aggregation, integration, and customization brokerage. In other words, cloud broker acts as middle layer between the client of a cloud computing service and the cloud service providers.

2.4.3.2 Types of Cloud Brokers

Cloud brokers are divided into three main types [8] as follows:

- **Cloud Aggregator.** The main function of cloud aggregator is packaging and integrating multiple service providers into one simple graphical user interface. Clients are then able to select services directly from the cloud aggregator with one single bill to the broker.
- **Cloud Integrator.** Integrators add value by enhancing workflows across hybrid environments through a single orchestration to increase performance and decrease business risk. Once the migration is complete, the integrator can continue to provide support to the organization on an ongoing basis as needed.

- **Cloud Customizer.** As the name suggests, customization involves modifying existing cloud services to meet business needs. In some cases, the broker may even develop additional features as requested by the organization. This function is critical to building a fully configured cloud with improved visibility, compliance, and integration of key IT processes.

2.4.4 Broker-based cross-cloud federation manager (BBCCFM)

J. Bou Abdo et Al. [1] proposed a new broker node inside Cross-cloud federation manager to enhance the mechanisms of discovery and delivery of all types of cloud services to end users.

A. Celesti et Al. [9] proposed, peer-to-peer group to allow CSPs interested in federation to join and subscribe, hence, the joined CSPs may share its information to a centralized file accessible by other clouds.

On the other hand, the broker model proposed by J. Bou Abdo et Al. [1] allows extra enhancement over A. Celesti et Al. [9] model on the discovery Peer-to-Peer (P2P) file used in cloud discovery.

The already proposed broker model contains information and technical details of all clouds and their resources. These information are stored in one table defined as UDDI (Universal Description, Discovery, and Integration).

Since the broker has an updated information list about all participating CSPs, it can save time and effort on the saturated cloud. In the previous mechanism described by A. Celesti et Al. the cloud has to get the P2P file and communication with all participating cloud to share or get resources.

However, the broker model requests one pair of messages only.

2.5 Firewall and cloud-based firewall

In information security and networking, a firewall is a system used to manage access inside private network. Firewalls can be hardware entities, software modules or combination of both.

Firewalls prevent end-users inside protected network to send or receive message directly to outside networks (Example: The internet).

Furthermore, all messages entering or leaving the network must first pass by the firewall to be examined against security criteria before allowing or denying the message flow [10].

Figure 3 shows a traditional firewall architecture inside LAN network connected to the internet.

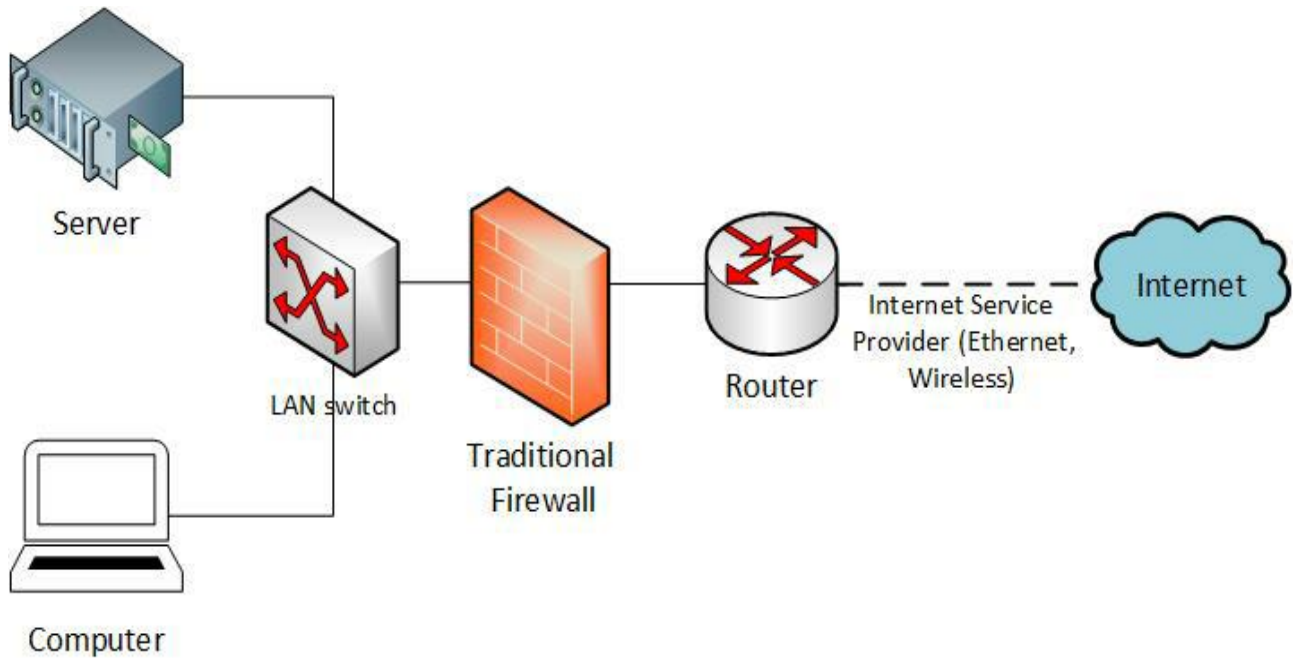


Fig. 3 Traditional firewall architecture inside LAN network

Firewalls exist in many types such as proxy, state full and next generation:

2.5.1 Proxy firewall

It acts as gateway from one network to another for specific applications. Proxy firewalls can provide content caching and security enhancement by preventing direct access from outside the network.

2.5.2 State-full inspection firewall

Mainly considered as traditional firewall which allows or blocks traffic data based on state, port and protocol. It provides continuous monitoring of all activities starting with the opening of connection until its closure. Defined rules are created by administrators to grant or deny access from and to outside networks.

2.5.3 Next-generation firewall

Or NGFW, which surpasses standard packet filtering and inspection provided by old generation firewalls. NGFW are used to prevent advanced malwares attacks held on the application layer inside the network. Gartner says NGFW must do the job of standard firewalls plus intrusion prevention, application monitoring and heuristic analysis for security threats.

2.5.4 Threat-focused NGFW

It includes all standard NGFW capabilities with advanced threat analysis and remediation. It allows administrators to secure assets that are at risk with context awareness. These firewalls provide intelligent security automation, endpoint event correlation, and continuous monitoring of suspicious activities.

2.5.5 Cloud based firewalls

Main advantages of cloud-based firewall are three:

- **Scalability:** Cloud-based firewall vendors provide services to many customers and in their backbone they use on-premise firewalls designed to scale up to meet increasing demand. Scalability is crucial when it comes to significant increase in bandwidth. On-premise firewall needs replacement when bandwidth is higher than firewall throughput. However, cloud-based firewalls can be scaled up based on bandwidth increase.
- **Availability:** Cloud-based firewall offer high availability (99.99%) through their backbone infrastructure enhanced by power redundancy, network services and replication architecture to allow maximum availability in comparison to on-premises firewalls.
- **Extensibility:** Cloud-based firewalls can be bought and used from anywhere in the world. On-premises firewalls are designed to be accessed only from same site location.

Figure 4 shows a typical architecture of cloud-based firewall:

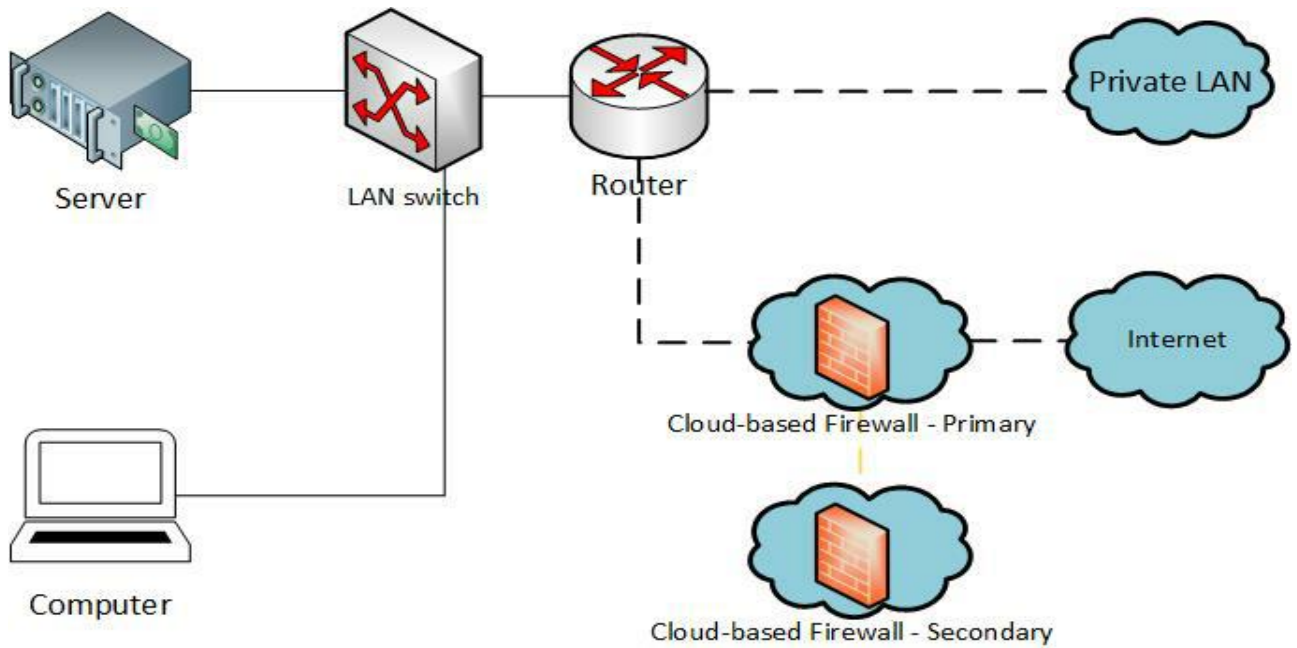


Fig. 4 Traditional Cloud-based firewall architecture inside LAN network

2.6 Proxy servers

Proxy servers are dedicated servers or software systems used as middle point between endpoints such as personal computers and servers to grant specific requests from end users. The proxy server can be embedded with firewall server or can be separate which sends requests to firewalls.

Proxy servers allows many features and advantages such as:

- Hiding customers' real IP address.
- Filtering requests from end users
- Protection from outside malwares

Figure 5 shows the default architecture of a proxy server:

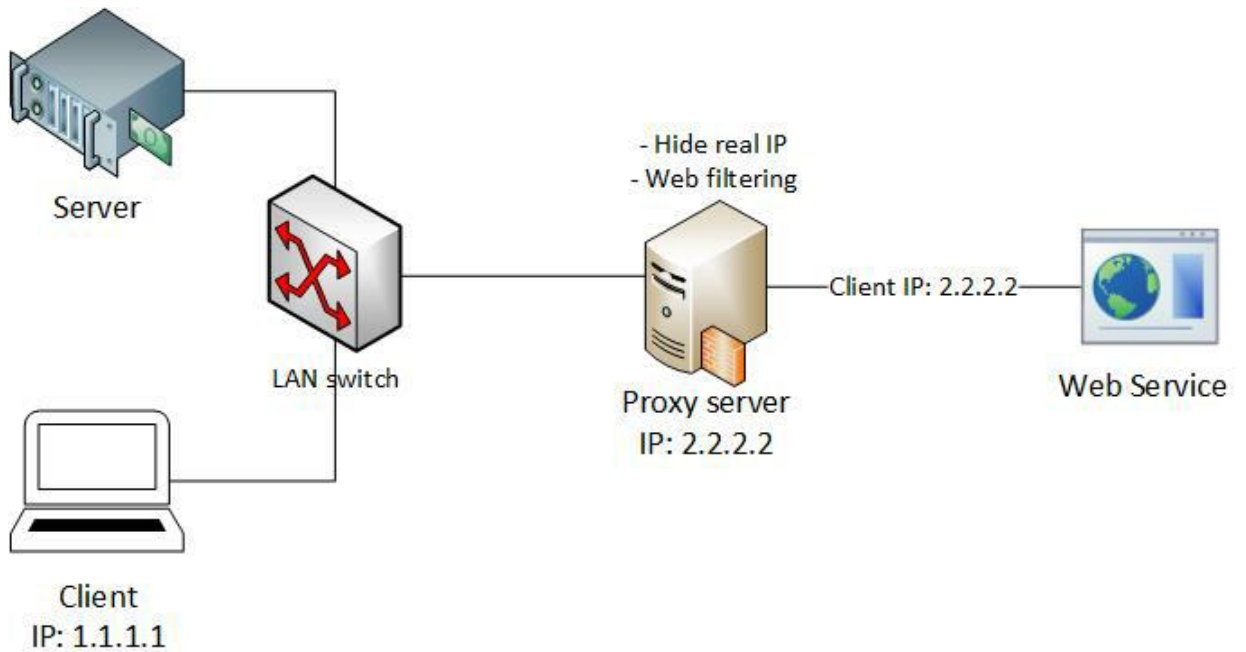


Fig. 5 Traditional proxy server in action

2.7 Security Token Services (STS)

There are two types of security token service: Identity Provider STS (IP-STS) and a Relying Party STS (RP-STS).

- IP-STS authenticates a client using integrated OS authentication (Windows integrated authentication). It creates a SAML token based on the claims sent by the client, and might add its own claim.

A relying party application (RP) receives the SAML token and uses the claims inside to decide whether to grant the client access to the requested resource.

- RP-STS uses SAML token provided by an IP-STS that it trusts. Typically, an IP-STS is found in the client's domain, whereas an RP-STS is found in the RP's domain.

This RP-STS and IP-STS connection can be shown in figure 6.

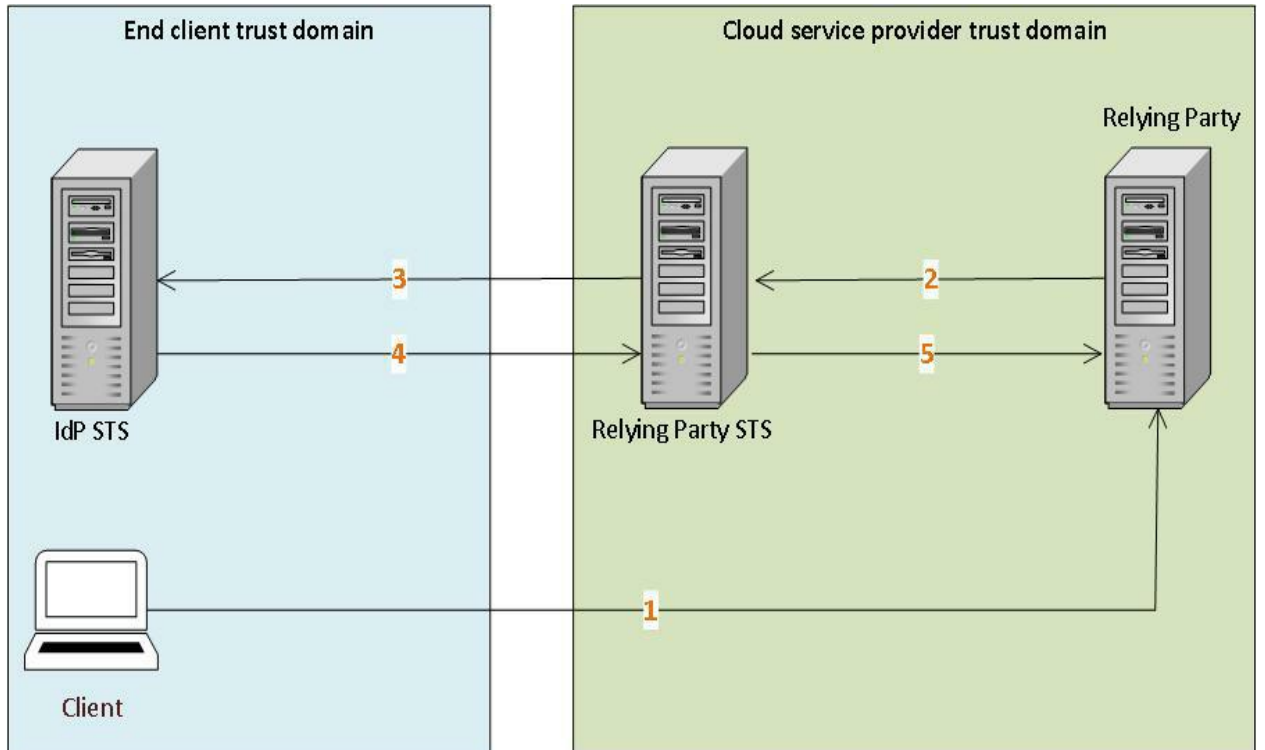


Fig. 6 Identity Provider STS (IP-STS) and a Relying Party STS (RP-STS).

2.8 Authentication and authorization

Authentication is the process of proving that somebody really is who he claims to be.

Example: Login credentials of a user trying to login to webpage.

Authorization refers to rules that determine who is allowed to do what. Example: the

user may be authorized to create and delete databases or only authorized to read.

The two concepts are completely orthogonal and independent, but both are essential in security design. The failure in achieving any of the two concepts makes any attempt to break the security possible.

2.9 Chapter Summary

Consumers and organizations have many different reasons for choosing to use cloud computing services. They might include: Convenience, Scalability, Low costs, Security, Anytime and anywhere access high availability.

As part of a cloud federation, even small SPs can offer global services without building new infrastructure. End users can host apps with their federated cloud provider of choice, instead of choosing from a limited “global” cloud providers on the market today and making do with whatever pricing, app support and SLAs they happen to impose. Cloud users can choose a local host with the exact pricing, expertise and support package that fits their needs, while still receiving instant access to as much local or global IT resources as they’d like. They get global scalability without restricted choice, and without managing multiple providers and invoices.

To achieve better cloud federation experience, solid and flexible settings should be implemented using broker-based servers. This architecture may enhance the work flow starting by users request to reach for the requested resources. IAM still missing in this

case and until now, most of our researches showed that FIM is the best candidate to be deployed in our architecture. Next chapter shows how FIM may be implemented and merged with Broker-Based systems.

CHAPTER 3. Access Control and IAM Mechanisms

3.1 Introduction

In this chapter, we discuss the most used and well known access control and IAM mechanisms in the domain of computing and cloud security. The identity and access management (IAM) mechanism includes the components and policies necessary to control and track user identities and access privileges for IT resources, environments, and systems.

Specifically, IAM mechanisms comprises four main components [11]:

- **Authentication:** Username and password remain the most common forms of user authentication credentials managed by the IAM system, which can also support digital signatures and certificates, biometric hardware, dedicated software and locking user accounts to reserved IPs or MAC addresses.
- **Authorization:** The authorization component uses attribute services to define attributes and access control rules and oversees the relationships between identities, access control rights, and IT resource access.
- **User Management:** Based on the administrative capabilities of the system, the user management program is responsible for creating new user identities and

access groups, resetting passwords, defining password policies, and managing privileges.

- **Credential Management:** The credential management system creates and manages identities through credential issuance.

IAM mechanisms stated are: Mandatory Access Control (MAC), Discretionary Access Control (DAC), Role Based Access Control (RBAC), Attribute Based Access Control (ABAC), Federated Identity Management (FIM), Content Based Access Control (CBAC) and Policy Based Access Control (PBAC).

We will give detailed architecture and work flow inside each of the mentioned mechanisms.

3.2 Access control and identity management mechanisms

3.2.1 Mandatory Access Control (MAC)

MAC is an access control policy owned by the system and not the data owner. In other words, users have no full access to resources they create. MAC policy defines the access rights and users cannot grant themselves higher level of permissions than the administrator allows.

MAC policy controls the access of information, processes or devices by authenticated users or processes with different levels.

MAC based systems have been coupled with security policy models, which acts as statement of the protection properties the system must have. There are levels of trust and sensitivity for users and information [12].

These levels can be summarized as follows: Unclassified - confidential - secret - top secret. Security levels is the term used for either clearance level or classification level.

Clearance level is the level of trust given to a person with a security clearance, or a computer with classified information or an area securing classified information.

Classification level tells the level of sensitivity linked with some information such as document or file. The level must also define the degree of damage if the information is disclosed to an enemy.

MAC has no boundaries. The MAC model places various restrictions on user actions that prevent dynamic manipulation of the underlying policies, which requires large parts of the OS and associated utilities to be "trusted", while assigning and enforcing secure levels by the system. Trusted components are usually a form of database and processes, such as releasing cryptographic processes that are placed outside of the MAC model due to their violation of MAC principles.

The code behind these components is assumed to be correct and conforming to the underlying policies of the system, in order to sustain the security policies and prevent

unauthorized access. In order to restrict access to these components additional measures should be taken. But practice shows that MLS can't be applied without implying MAC without changing the entire operating system and its various associated utilities outside of the MAC mode.

3.2.2 Discretionary Access Control (DAC)

DAC is a type of access control defined by the Trusted Computer System Evaluation Criteria "as a means of restricting access to objects based on the identity of subjects and/or groups to which they belong. DAC functions as a centralized security model as well as a distributed model. Distributing access to data by an administrator or a team of engineers define centralized security model. This can be time-consuming in large organizations especially if the admins are outsourced or off-site [13]. A knowledgeable personnel is allowed to distribute access to data and applications in a distributed model. This personnel can be a manager, supervisor or even lead by a team in large companies. While in smaller companies this role goes to the most computer-wise experienced member. Distributed models avoid delays because the administration of accounts is isolated. For example a manager wants to distribute records between for individuals based on location with the country such as records of home loans for a bank. The manager is in control of the data and can assign access to his employees to data because

the DAC is the security model. The manager can distribute the data the way he wishes, for example one member can have the eastern coast while the other can have access to the western coast... This way each employee can view their records only. This is due to the classification in the older MAC model.

DAC is implemented in a distributed security model, which reduces account access change turnaround, due to the removal of the "middle-man".

Some OS manipulated this DAC implementation to create new roles such as a "Workgroup Manager" which is implemented by Novell Netware, which can grant ability to modify access for accounts or even create them.

This access control has the potential to benefit human reasoning, and allows for variables that are not considered in the MAC model to be controlled by the administrator.

3.2.3 Role Based Access Control (RBAC)

3.2.3.1 Definition and main components

Originally designated for military purpose to secure and limit access to subtle data, RBAC was a great success in providing security model based on predefined roles having built in permissions.

Unlike roles implemented in OS like UNIX, RBAC developed by NIST (National Institute of Standards and Technology) is a secure control model which is scalable, logical, non-

system independent and economic on implementation. In 1992, Ferraiolo and Kuhn came up with a complete RBAC model solution. RBAC evolution was done under four main models:

- RBAC0 was the initial model consisting of separation of duties and least privileges. It does not contain a hierarchy hence users were assigned direct permissions.
- RBAC1 introduced the use of hierarchies based on level of responsibilities and job levels inside organizations [14].
- RBAC2 introduced the concept of constraints, acting as limiters to enforce policies and regulate access based on certain criteria. Moreover, constraints can ensure the separation of duties.
- RBAC3 covers all of the components in previous RBAC models, hence allowing full hierarchal structure and constraints.

RBAC has five elements: Users, roles, permissions, operations and objects used to create level of permissions and constraints. Users are entities wishing to access data resource or objects.

Unlike DAC models, users have no full access to resources but may inherit the access permission from roles associated with.

Fig 7 shows the default levels of participants in RBAC

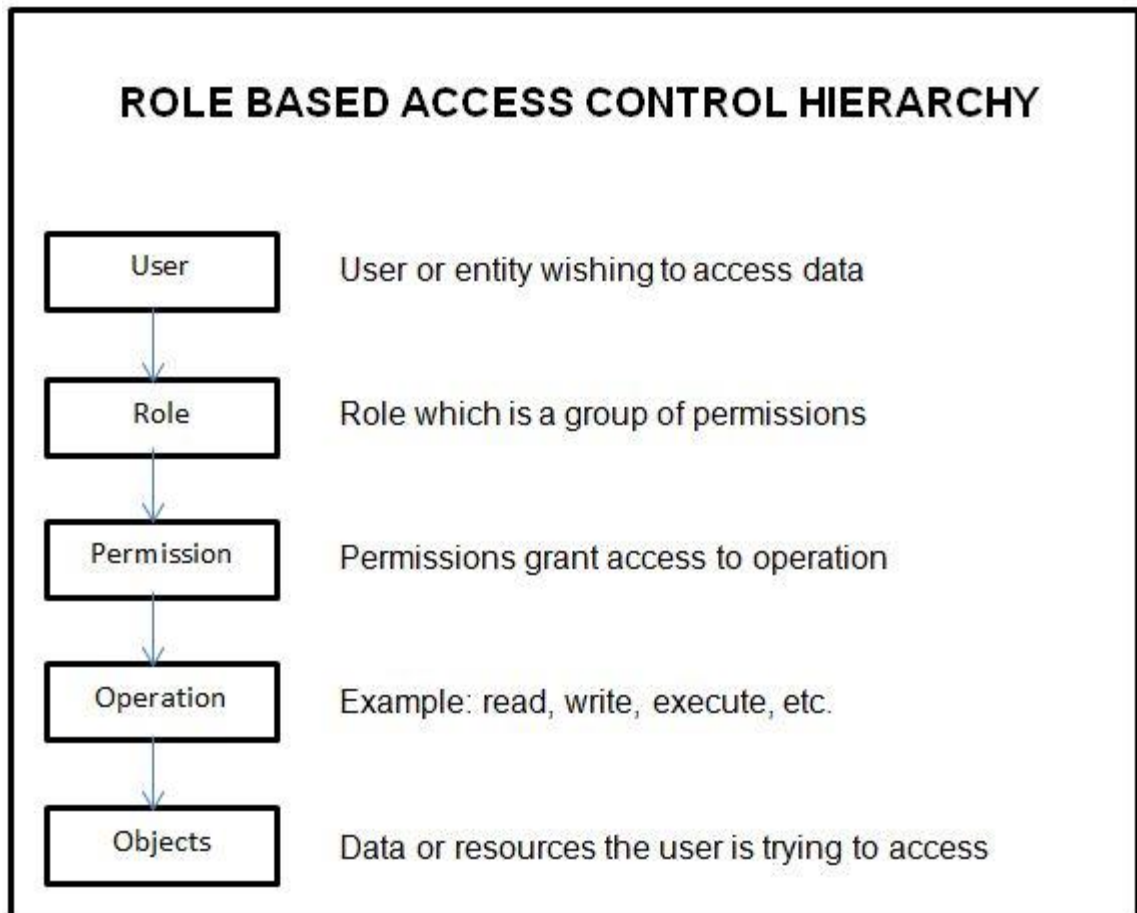


Fig. 7 Hierarchy of elements inside RBAC.

Role is a group of permissions based on a job function inside organizations. Users are assigned to single or multiple roles based on their position and function in the organization. Permissions are assigned to a role and granted access to operations. Levels of permissions are higher than operations (insert, delete and update) since limited functions are found inside operations. Objects are accessed through operations that users have permission to access through assigned role.

3.2.3.2 RBAC in Operating systems: UNIX based AIX

RBAC by default is not assigned in UNIX based operation systems. In the below example, we enabled RBAC and showed some of the many built-in roles and authorizations inside the system.

Table 1 shows some roles, groups and authorizations inside AIX.

Table 1 roles, groups and authorization inside AIX

Role	Description	Role List	Authorization
AccountAdmin	User and Group Account Administration	N/A	aix.security.group, aix.security.user
BackupRestore	Backup and Restore Administration	N/A	aix.fs.manage.backup, aix.fs.manage.restore
FSAdmin	File System Administration	N/A	aix.fs.manage.change,aix.fs.manage.create,aix.fs.manage.debug,aix.fs.manage.defrag,aix.fs.manage.dump,aix.fs.manage.list,aix.fs.manage.mount,aix.fs.manage.quota,aix.fs.manage.recover
SysBoot	System Boot Administration	N/A	aix.system.boot.create,aix.system.boot.halt,aix.system.boot.info,aix.system.boot.reboot,aix.system.boot.shutdown
sa	System Administrator	FSAdmin AccountAdmin	aix.system.config.acct,aix.system.config.cron,aix.system.config.src,aix.system.install
secadm	Security Administrator	N/A	aix.security.group.change,aix.security.role.assign,aix.security.domains.assign,aix.security.user.change,aix.security.role.change,aix.security.passwd.normal,aix.security.user.attr.acct_locked
so	System Operator	BackupRestore SysBoot	aix.proc.kill,aix.ras,aix.system.config.init,aix.system.config.wlm
Useradm	User Administrator	N/A	aix.security.user.create.normal,aix.security.user.remove.normal,aix.security.user.list,aix.security.user.change,aix.security.role.create,aix.security.role.list,aix.security.group.create.normal,aix.security.ldap,aix.security.nis,aix.security.kerberos,aix.security.pki

Table 2 shows all authorizations and their associated command or job.

In Fig 8 and Fig 9, we have shown how top and sub authorizations levels are created.

These authorizations hold commands for which specific task is encompassed. Roles and users were also simulated with role switching and testing. In Fig 8, we created high-authorization and sub-authorization levels called NDU, NDU.SYSTEM and NDU.SECURITY.

Another sub-authorizations were also created, with their assigned commands:

NDU.SYSTEM.OS.REBOOT assigned with “reboot” command and

NDU.SECURITY.AUTH.LIST assigned with “lsauth” command. New role called

“NDU_ROLE” was created, in which sub-authorization NDU.SECURITY.AUTH.LIST is

inserted. In Fig 9, we showed that user “NDU_admin” cannot perform “lsauth” task

before granting the permission by assigning the user “NDU_admin” to role “NDU_ROLE”.

After granting the required permissions, the command “lsauth” can be run as shown in

figures 8 and 9

Table 2 Authorizations inside AIX

Authorization name	Definition	Authorization name	Definition
aix	Operating System Administration	aix.lvm	Logical Volume Manager Administration
aix.device	Device Administration	aix.lvm.conc	Manage Enhanced Concurrent LVM Daemons
aix.device.config	Configure Devices	aix.lvm.debug	Debug Logical Volume Manager
aix.device.config.path	Configure MPIO Devices	aix.lvm.manage	Manage LVM Objects
aix.device.config.printer	Configure Printers	aix.lvm.manage.change	Change LVM Objects
aix.device.config.random	Configure the Random Device	aix.lvm.manage.create	Create LVM Objects
aix.device.config.tty	Configure TTY Devices	aix.lvm.manage.export	Export a Volume Group
aix.device.manage	Manage Devices	aix.lvm.manage.extend	Extend LVM Objects
aix.device.manage.change	Change Attributes of a Device	aix.lvm.manage.import	Import a Volume Group
aix.device.manage.create	Create a New Device	aix.lvm.manage.join	Join LVM Objects
aix.device.manage.list	List Attributes of a Device	aix.lvm.manage.migrate	Move LVM Objects
aix.device.manage.remove	Remove a Device	aix.lvm.manage.mirror	Mirror LVM Objects
aix.device.monitor	Monitor Devices	aix.lvm.manage.recreate	Recreate a Volume Group
aix.device.monitor.tty	Monitor a TTY Session	aix.lvm.manage.reorg	Reorganize a Volume Group
aix.device.stat	Device Status	aix.lvm.manage.remove	Remove LVM Objects
aix.device.stat.printer	Display Printer Status	aix.lvm.manage.scan	Scan LVM Objects
aix.fs	File System Administration	aix.lvm.manage.split	Split LVM Objects
aix.fs.chroot	Change the root directory	aix.lvm.manage.sync	Sync Logical Volume Mirrors
aix.fs.manage	Manage File Systems	aix.lvm.manage.unmirror	Unmirror LVM Objects
aix.fs.manage.backup	Backup Files and File Systems	aix.lvm.manage.varyoff	Vary Off a Volume Group
aix.fs.manage.change	Change Attributes of File Systems	aix.lvm.manage.varyon	Vary On a Volume Group
aix.fs.manage.create	Create New File Systems	aix.lvm.perf	Manage LVM Performance
aix.fs.manage.debug	Debug File Systems	aix.lvm.perf.stat	Query LVM Performance Statistics
aix.fs.manage.defrag	Defragment File Systems	aix.lvm.perf.tune	Modify LVM Tunable Performance Parameters
aix.fs.manage.dump	Dump File System Information	aix.lvm.readlvcopy	Read a Specific Copy of a Logical Volume
aix.fs.manage.export	Export File Systems	aix.network	Network Administration
aix.fs.manage.list	List Characteristics of File Systems	aix.network.config	Network Configuration
aix.fs.manage.mount	Mount File Systems	aix.network.config.arp	Configure Address Resolution
aix.fs.manage.quota	Manage Disk Quotas	aix.network.config.host	Configure Host
aix.fs.manage.recover	Recover Corrupted File Systems	aix.network.config.mail	Configure Mail Aliases
aix.fs.manage.remove	Remove File Systems	aix.network.config.no	Configure Network Tuning Parameters
aix.fs.manage.restore	Restore Files from a Backup	aix.network.config.route	Configure Routing Tables
aix.fs.manage.snapshot	Modify	aix.network.config.tcpiip	Configure TCPIP Network Interface Parameters
aix.fs.manage.unmount	Unmount File Systems	aix.network.daemon	Network Daemon Administration
aix.fs.object	File System Object Administration	aix.network.debug	Enable Network debug
aix.fs.object.acl	Read and Write Object ACL	aix.network.ndaf	NDAF Administration
aix.fs.object.create	Create File System Objects	aix.network.ndaf.admin	Manage NDAF Administrative Server
aix.fs.object.group	Read and Write Object Group Ownership	aix.network.ndaf.client	Manage NDAF Client
aix.fs.object.list	List Attributes of File System Objects	aix.network.nfs	NFS Administration
aix.fs.object.mode	Read and Write Object Mode	aix.network.nfs.export	Create and Remove NFS Exports
aix.fs.object.owner	Read and Write Object Owner	aix.network.nfs.manage	Start and Stop NFS
aix.fs.object.remove	Remove File System Object	aix.network.nfs.mount	Create and Remove NFS Mounts
aix.fs.object.time	Change Object Access and Modification Time	aix.network.nfs.set	Configure NFS
aix.fs.stat	File System Statistics	aix.network.rawsock	Create and Use Raw Sockets

```

10.99.129.22 - PuTTY
aixtesttime:/# mkauth NDU
aixtesttime:/# mkauth NDU.SYSTEM
aixtesttime:/# mkauth NDU.SECURITY
aixtesttime:/# lsauth NDU
NDU id=10030
aixtesttime:/# lsauth NDU.SYSTEM
NDU.SYSTEM id=10031
aixtesttime:/# █

aixtesttime:/# mkauth NDU.SYSTEM.OS.REBOOT
1420-004 Authorization hierarchy "NDU.SYSTEM.OS" does not exist.
aixtesttime:/# mkauth NDU.SYSTEM.OS
aixtesttime:/# mkauth NDU.SYSTEM.OS.REBOOT
aixtesttime:/# lsauth NDU.SYSTEM.OS.REBOOT
NDU.SYSTEM.OS.REBOOT id=10034
aixtesttime:/# █

aixtesttime:/# mkauth NDU.SECURITY.AUTH
aixtesttime:/# mkauth NDU.SECURITY.AUTH.LIST
aixtesttime:/#
aixtesttime:/# mkrole dfltmsg="THIS IS TEST ROLE FOR NDU - UNIV" NDU_ROLE
aixtesttime:/# lsrole NDU_ROLE
NDU_ROLE authorizations= rolelist= groups= visibility=1 screens=* dfltmsg=THIS I
S TEST ROLE FOR NDU - UNIV msgcat= auth_mode=INVOKER id=21
aixtesttime:/# █

aixtesttime:/# setsecattr -c accessauths=NDU.SECURITY.AUTH.LIST /usr/sbin/lsauth
aixtesttime:/# lssecattr -c -a accessauths ALL | grep NDU.SECURITY.AUTH.LIST
/usr/sbin/lsauth accessauths=NDU.SECURITY.AUTH.LIST

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

* Role NAME                                [Entry Fields]
Role ID                                    NDU_ROLE
AUTHORIZATIONS                             [21]
Role LIST                                   [NDU.SECURITY.AUTH.LIST]
GROUPS                                     []
Smit SCREENS                               [*]
VISIBILITY                                  [1]
Message CATALOG                             []
Message SET                                 []
Message NUMBER                              []
Description                                 [THIS IS TEST ROLE FOR NDU - UNIV]

aixtesttime:/home/NDU_admin> lsauth
ksh: lsauth: 0403-006 Execute permission denied.
aixtesttime:/home/NDU_admin> █

```

Fig. 8 Role creation and authorization assignment inside AIX

```

aixtesttime:/# chuser roles=NDU_ROLE NDU_admin
aixtesttime:/# setkst
Successfully updated the Kernel Authorization Table.
Successfully updated the Kernel Role Table.
Successfully updated the Kernel Command Table.
Successfully updated the Kernel Device Table.
Successfully updated the Kernel Object Domain Table.
Successfully updated the Kernel Domains Table.
aixtesttime:/# su - NDU_admin
aixtesttime:/home/NDU_admin> lsrole
ksh: lsrole: 0403-006 Execute permission denied.
aixtesttime:/home/NDU_admin> lsauth
ksh: lsauth: 0403-006 Execute permission denied.
aixtesttime:/home/NDU_admin> swrole NDU_ROLE
NDU_admin's Password:
aixtesttime:/home/NDU_admin> lsauth
Usage: lsauth [-R load_module] [-C | -f] [-a attr attr ...] { "ALL" | auth1,auth2 ... }
aixtesttime:/home/NDU_admin>
    
```

Fig. 9 Authorization and role testing

3.2.3.3 RBAC architecture

RBAC model consists of many elements such as: Users, Roles, Permissions and resources.

Figure 10 shows the inter-relationship of RBAC elements.

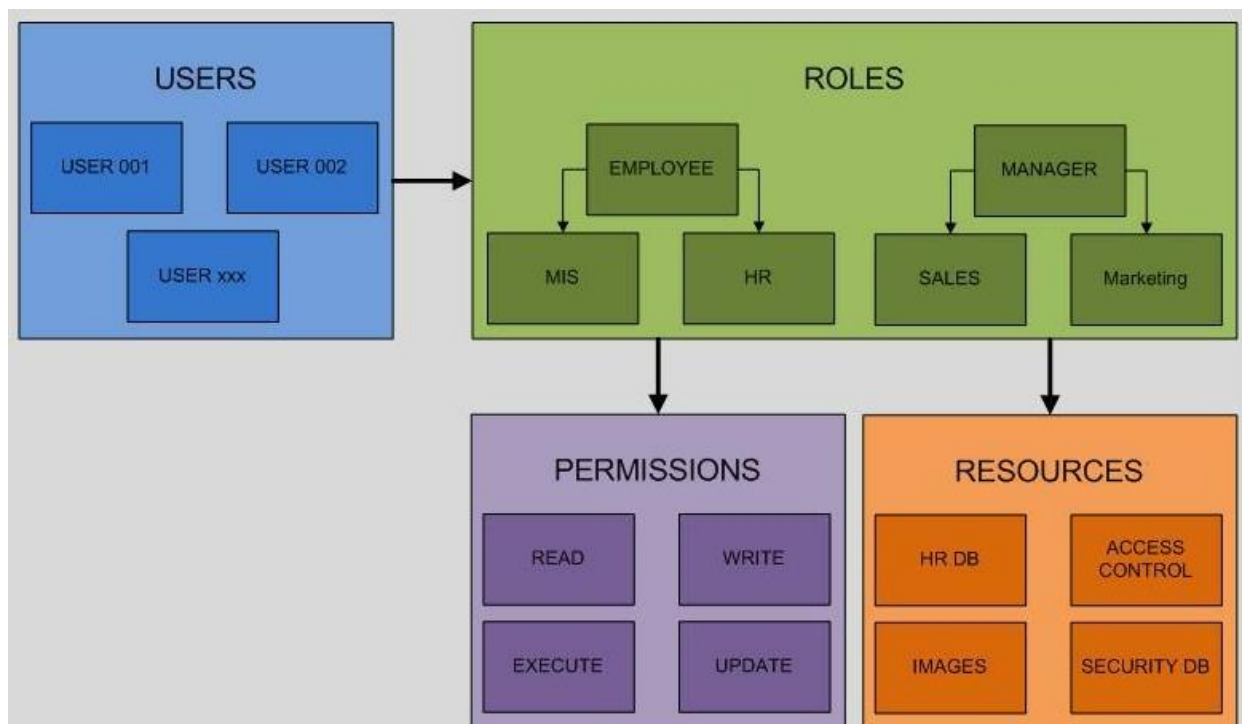


Fig. 10 Interconnection of elements inside RBAC.

RBAC basic implementation strategies can be summarized as follows:

- **Ease of implementation**

RBAC model needs to be deployed through roles engineering, to reflect all positions inside the organizational policy. This task requires lot of research and testing to ensure that the concept of least privileged is achieved through role design. Once the role is tested and implemented, administrators can benefit from the new design which allows less human intervention when updating access requests.

Moving user from current role to another is hence an easy task and it's done through the disassociation from user's old role and association of a new role.

Moreover, the disassociation of the user from his current role makes him unable to access the system and benefit from the assigned permissions.

Should the user leave the work, the task of disassociation of the user from the role is easy and makes the access impossible even if their account accidentally remain active.

Hierarchy and rights inheritance

RBAC3 supports a hierarchal framework, which can be used to ease association by allowing permissions to spill down to subsidiary objects. In Novell NetWare, an example would be the rights coming out of an authorization unit down into the users arranged underneath it. The other advantage of this comes into place as for role design, this

dynamic framework can immensely diminish the amount of roles made. Another advantage is that different roles can be associated with each other to allow greater functionality for the end-user.

- **Separation of duties**

RBAC3 permits and authorizes separation of duties through constraints, which implies user with a specific employment job role which can't be in another role at the same time.

This idea is remarkably useful and required particularly in health frameworks.

Figure 11 shows RBAC workflow, where both new RBAC decision and historic behavior are triggered.

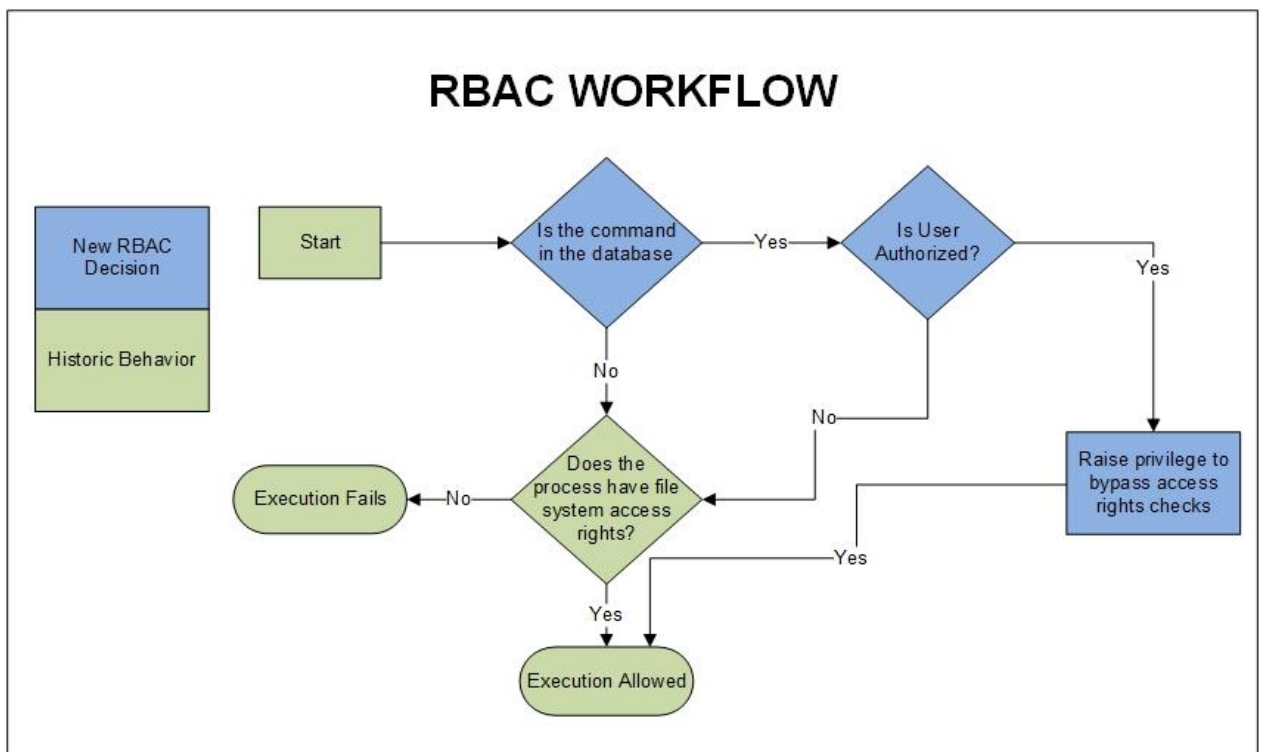


Fig. 11 RBAC work flow and decision making

3.2.3.4 Benefits of RBAC

- **Scalability**

RBAC3 is scalable. It allows well definition and documentation of policies and organizational structure within enterprises [14]. In an enterprise where this is the condition, roles can be made by “role engineering” and changed similarly as required. An advantage to this is to grant on a very basic level to users, where individual administration of accounts is reduced or eliminated. Since role engineering has developed the package of approvals for each user, the information required in DAC to observe appropriate rights for users is implemented after being built. As the organization grows, more roles may be required. However, since RBAC3 provisions a hierarchal flow allowing rights to flow down the tree and to be constrained, extra design and implementation tasks are reduced.

- **Security**

Planning roles before implementation permits some security vulnerabilities inherited from DAC, for example, administrator permission errors during logging in and out.

In connection with the security offered through the MAC security, conflicts have been made that RBAC is practically identical. The argument that RBAC is a sort of MAC comes from the fact that MAC is dependent on data and users grouping or labels, and RBAC

uses RBAC roles as a kind of classification. Through the hierarchies and rights inheritance, we can consider that RBAC is a "Multilevel Security Model". In any case, RBAC does not have the hard coded security courses of action that MAC offers, a huge thought for military security. Scalable RBAC3 offers the limit through role engineering and hierarchies to make roles granular to secure the structure. It is the practice at various relationship to oversee security issues similarly as they get the chance to be distinctly known. In RBAC security, officials must have propose data of how agreements are being discussed, why and what operations are associated with those approvals and parts.

3.2.4 Attribute Based Access Control (ABAC)

3.2.4.1 Definition and main components

ABAC enables access to objects by assessing rules against the attributes of elements (objects and subjects), operations and area connected to the entities. ABAC permits higher number of inputs than RBAC which allows higher mixture of factors to a bigger combination of parameters and rules to express policies only restricted by programming language [15]. This malleability empowers development of access control rules without considering individual connections between subjects and objects. For instance, a subject is assigned a group of subject attributes upon action. Here is an example to illustrate the above scenario: John Doe, a doctor in the emergency room division and an object "O"

was assigned its attributes upon creation. Let's say for example, a directory containing medical information about patients with heart disease where objects may get their attributes either from the creator or by automatic scanning software. The owner of an object creates an access control rule utilizing attributes of subjects and objects to represent the arrangement of suitable capabilities. ABAC deals with all identities, roles and all properties of users, objects and subjects as attributes, which means more complex policy compared to DAC, MAC and RBAC. But having attributes is a rich feature that allows capturing all the properties in access control systems such as DAC, MAC and RBAC. ABAC attributes allow: high level of flexibility by capturing identities and higher level of access list in DAC, security label and classifications in MAC and roles in RBAC. ABAC also imprisons more attributes like location, authentication level, qualifications, time, etc. NIST mentions that ABAC leads to a problematic and multi-layered charge which is design and implementation of role and attribute.

ABAC components includes users, subjects, objects, user attributes, subject attributes, and object attributes, permissions, authorization policies, and constraint checking policies. An attribute can take an element, for example, a user. An attribute range is given by a limited arrangement of atomic qualities and an atomic respected attribute will

return one incentive from the range, while a set esteemed attribute will give back a subset of the range.

Every user is related with a limited arrangement of user attributes works whose qualities are allotted by security administrators (outside the extent of the model). These attributes speak to the user properties, for example, name, clearance, roles and sex.

Subjects are made by users to perform out a few activities in the framework. The creating user is the special case who can end a subject and every subject is associated with a limited arrangement of subject attribute functions which require an underlying incentive at creation time.

Subject attributes are set by the creating user and are obliged by policies built up by security designers [15].

For instance, a subject attribute estimation might be acquired from a relating user attribute. Objects are assets that should be secured and related to a limited arrangement of object functions. Objects might be made by a subject for the benefit of its user. At creation, the objects' attributes might be set by the user by means of the subject. The values may be bound by the relating subject's attributes. For instance, the new object may acquire values from relating subject attributes.

Constraints are functions which return genuine when conditions are fulfilled and false otherwise. Constraints can apply at subject and object creation time, and in this way at subject and object property change time.

Permissions are benefits that a user can hang on objects and practice by means of a subject. Permissions empower access of a subject to an object in a specific mode, for example, read or write. Permissions definition is reliant on particular frameworks constructed utilizing this model.

Authorization policies are a Boolean functions which is assessed for every access decision. This is suitable in multi-approach systems. For example, in banking systems, policies define who is allowed to view, edit, delete and approve banking transactions.

An instance of positive policy would be like:

A manager can view banking financial transactions.

An instance of a negative policy would be like:

No person can approve a banking financial transaction above their approval level.

In large enterprises, policies may be a combination to achieve any relevant authorization scenarios.

ABAC covers all of the three access control models MAC, DAC and RBAC. ABAC can be used to configure all the above three models and all the extensions can be modelled;

For example, in MAC, subjects are categorized into read-only and read write for a better security and availability.

3.2.4.2 Benefits of ABAC

The main advantage of ABAC is ensuring the right information is only accessible by the right people and only when they need it. NIST mentions multiple advantages of ABAC but we will state the most important three [16]:

Single Point Provisioning of Users.

ABAC system administrator is not obliged to check users' account, assign roles or modify their access control list based on approval processes.

ABAC system is able to know what is accessible to the user based on policies assigned to the application. There is central management for the attributes and can be accessed from internal sources such as Active directory.

Dynamic Access Control.

Access control is dynamically made based on the most updated policies. Digital policies always change to address security alarms which include conditions such as nation level of security. ABAC model uses these updates as input data for policy decisions; which allows flexible control depending on the organization change.

Finer Grained Access Control.

RBAC might result in “roles explosion” when federal agencies administrators create roles for a small group of people despite the many updates on the access levels. ABAC allows accurate access control by extracting from a higher set of attributes to take decision, generating a bigger set of probable rules and choices without managing groups and roles.

New and Emerging Technology.

ABAC engineers are still working to understand the needs of customers. XACML and SAML should be used to exchange authentication and authorization to maintain agreement with federation and solution development over time.

Sensitive Federal Environments.

In the complex methodological environments that exist in federal agencies, many attribute based application types exist.

People Change, Process Change.

ABAC’s main idea is to provide centralized authorization decisions. When subjects send access request to create actions on objects, it is interrupted by the Policy Enforcement Point (PEP) which converts the request from a business procedure to an authorization request.

3.2.4.3 Implementing ABAC

The authorization policy lifecycle involves the below steps:

Step 1: Use case definition

This is the first step in ABAC implementation. A discussion is made to provide context and an achievable scope. For example, a broker based server wants its users to access resources and data provided. Users may be officers, auditors, managers, third party providers, etc.

Step 2: Authorization requirements gathering

In this step, a statement or natural language should state what should be allowed or disallowed. Once the use case is well defined, authorization requirements can be authored. These requirements may come from sources and stakeholders. For example, some requirements are related to business operation (e.g. working hours). Other are related to security guidelines (e.g. Data encryption). Example: Managers are allowed to view all records but employees can view records in their own department only.

Step 3: Attributes identification

Once the requirements have been listed, attributes required for those requirements should be identified. For each attribute, shortened name should be identified for a better and clearer policy implementation.

Shortened name: Part of name identifier and the most used inside a policy. Naming should be complied with ALFA naming convention.

Step 3: Namespace organization

An attribute namespace is the logical domain it belongs to. Namespace organizes attributes into different domains. For example, our Broker-Based Cross-Cloud Federation Manager (BBCCFM) may want to differentiate namespaces used from customers and employees:

Code 1 Namespaces sample for customer and employee

`com.bbccfm.user.customer`

`com.bbccfm.user.employee`

Step 4: Category management

All attributes fall into different categories which define the function the attribute plays in the authorization requirement. In ABAC, categories are divided into four sections:

Subject, Action, Resource and Environment

For example, `com.bbccfm.user.employee.approvalLimit` can belong to subject class.

Moreover, ABAC allows implementers to define their own categories.

Step 5: Data type specification

Each attribute should belong to a specific data type. When implementing ABAC requirements and the actual policy language e.g XACML, it is necessary to specify the data types. The most common data types are String, Boolean, Numerical (integer or double), Date and time types

For example, `com.bbccfm.user.employee.approvalLimit` is of integer type.

Step 6: Value constraints arrangement

Value constraints provide a value range. There are many ways to specify the constraints.

For example, a list of discrete values like country codes are defined by ISO 1366 or number ranging from 0 to 9.

Step 7: Cardinality definition

Defining cardinality for each attribute leads to better policies, access reviews, authorization requests and enforcement. Cardinality relates to the number of values any attribute can have at any given time. Example, `dateOfBirth` has one unique value whereas `citizenship` could hold multiple values.

Step 8: Source identification

ABAC attributes source is resolved from a System of Record (SoR) which is equivalent to (Policy Information Point) PIP in XACML terminology, and the application is referred to as the PEP. Both PIP and PEP may be the source of attribute values.

Step 9: Contact management

It is very important to keep track of the data used and of the attribute owner. This person is in charge of the integrity, availability and reliability of the attribute source.

Step 10: Writing the authorization policies

This step takes the natural language statements and convert them into machine-format statements. This allows the elimination of any doubt introduced by natural language. In the example below, we will define what the word “own” means. In natural language, a machine would not be able to make the right inference. This is why natural language should be broken into atomic attributes and attribute comparison. This step ends up with an implementable list of policies in order to be evaluated. Many types of tests can be run against ABAC policies:

- Binary testing: Easy test covered by XACML. Authorization requests and responses including decision are defined.

- Gap analysis and Reverse query testing

A policy tester must implement a test harness that iterates through the potential values and generates the relevant tests to guarantee test coverage.

Tests can then be run every time a policy is added, edited, deleted, and promoted to upper environments.

Figure 12 Shows elements inside ABAC systems

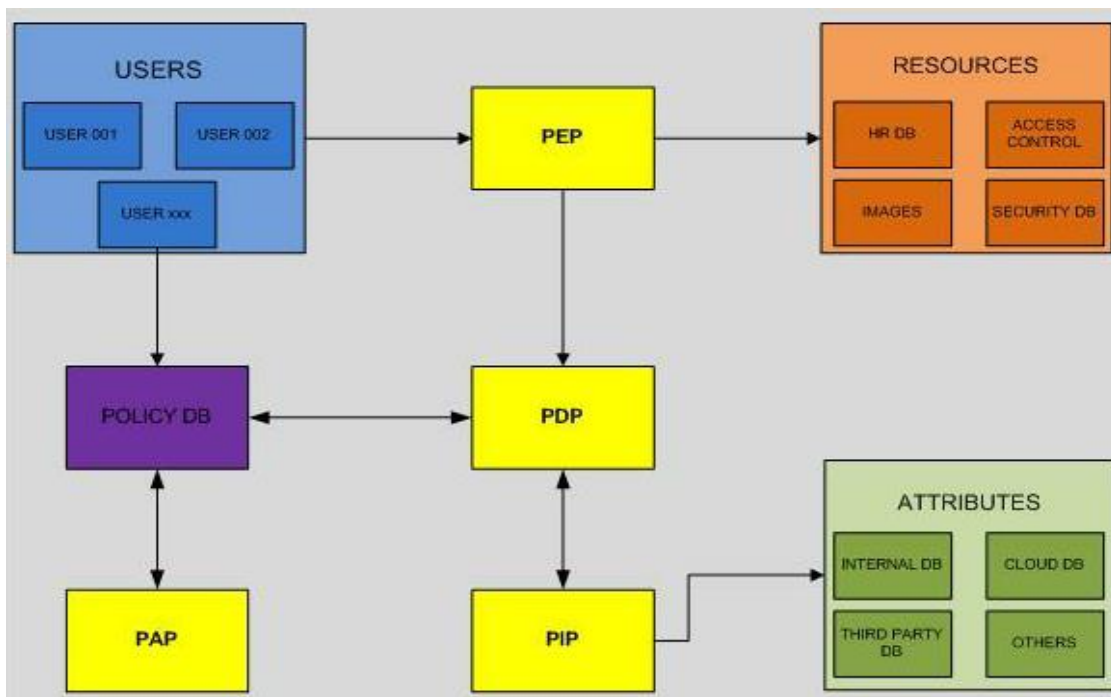


Fig. 12 Elements inside ABAC system

3.2.4.4. Deploy the architecture

In ABAC, we can choose where and how to deploy the PEP, which means we can determine what type of authorization can be achievable and how broad the protection is. Figure 13 shows the interaction of messages inside ABAC systems.

PEP can be implemented in many layers such as web SSO tier, presentation tier, API tier, business tier and data tier. The tier where PEP is implemented may impact the access granularity. For example, integrating PEP with Web SSO tier will be coarse-grained where PEP has no access to any message payload. Integrating PEP with API tier will be fine-grained as the PEP will have access to API messages in both ways. Multiple PEP can be deployed and it's the responsibility of application developer and security architect to define the most suitable place to integrate the PEP [16].

Actors involved in the implementation are:

- Application owner: defines the general use cases
- Business analysts, security architects and officers: define authorization requirements for the use case
- Business analysts, architects and data owners: define the required attributes
- Application developers, policy authors and business analysts: Define, implement and run policy tests.
- System architect and application owners: deploy the architecture and the policies
- Compliance and audit manager: Runs ABAC access reviews.

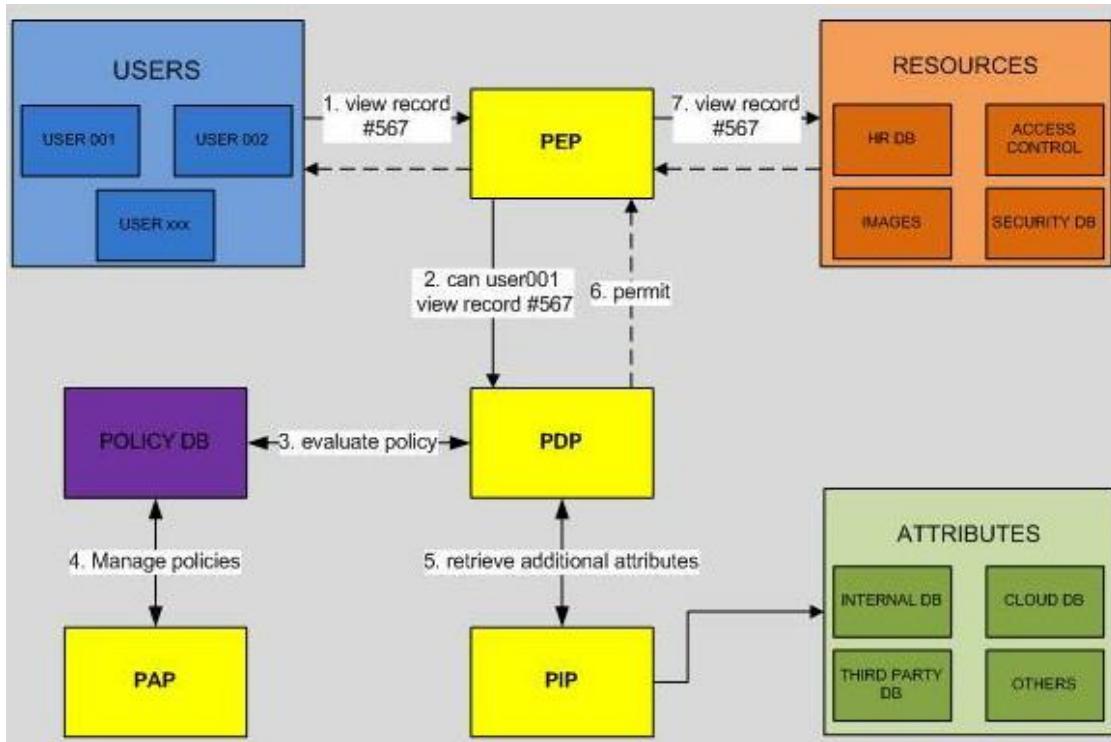


Fig. 13 Elements and message flow inside ABAC system

3.2.4.5 eXtensible Access Control Markup Language

“extensible Access Control Markup Language” is a security policy management standard used by the Organization for the Advancement of Structured Information Standards (OASIS) that is an easy and adaptable way to establish authorization policies in a complicated and active environments.

It is considered as a policy language and a request protocol to control decisions

XACML scenario

A user requests to access a resource that is protected by PEP (policy enforcement point).

The pep assigns an XACML request to the policy decision point (PDP) to check if the user should or should not be given the access. PDP issues a response to the PEP mentioning

is decision: permit, deny, intermediate or not applicable. This decision is applied by the PEP [17]. XACML is advantageous over using other proprietary and application-specific languages for access control.

Standardized Method for Authorization

Previously, application dealers put their rules to access control. These control policies are inputted in different languages by IT agents. While using XACML, the policy is written once and used for many applications.

Externalization/Centralization of Authorization

The policies will be handled centrally and changes are applied at once in all the organization by being created at an external PDP. This will allow the administrators to concentrate more on business issues.

Robust Standard

XACML 3.0 supports a wide range of access-control policy needs, data types, functions and rules. It can let different groups manage pieces of the policies and then merge the results to get one decision. XACML can operate with other standards like SAML and LDAP which has an operational and cost-saving advantage. Same PDP can be replicated through multiple servers for enterprise-level performance and scalability.

3.2.5 Federated Identity Management (FIM)

3.2.5.1 Definition and main components

Nowadays, with the diversity of sites and applications being used, global digital identities are spread over the web.

- Credentials should be created for each user visiting a new webpage.
- These credentials are stored on that website.
- Each time end users visit this website, they have to re-enter these credentials. This authentication should be done as well for each new accessed site.

This authentication process was an unwieldy system. Even when the different websites are managed by the same organization, end-users have to identify themselves on each login request. As the internet is becoming more complex and more interconnected, organizations are unable to carry over from one domain to another the user credentials. So, they should have a new system for authentication. Realizing this problem, researchers start to develop new authentication systems. Actually, the same origin policy was one of the most important principles. That made the concept of the “Federated Identity Management” very difficult to apply. The “same origin policy” forbids the access of the information stored on the end-user’s computer by another user except the “original creator” of that data. This principle states that each domain is

independent from the others. It means that domain Y cannot go through customer's data and transfer username and password to domain X. If so, the whole concept of internet security would be questionable. As a result, companies with multiple domains that want an easy transmission between its domains have to find a secure solution permitting this transfer of information. Here next, we explain the Federated Identity Management systems. These methods enable the secure transfer of data without violating the fundamental of the "same-origin-policy" [18].

- You, as a user, try to log into the application or website (the client).
- The client sends a request for authentication to an authorization server.
- The authorization server returns the authentication to the client.
- After this process, the user is permitted to access the application or not.

This triangular process enables the user to move freely between the different websites or applications.

If domains X and Y are related and if an external authorization server exists for both domains, the user can benefit from a smooth and secured experience of using data.

Nowadays the federated identity is applicable everywhere. That's what we call the "Single Sign On" (SSO). In this way, one can log into his Gmail and then open up Twitter in a different tab without logging out. All this is related to a central domain system

identifying the status of each user permitting him to move across the different sub-domains.

- End users need only to log in once.
- The central domain makes the Authentication.
- In order to move across other domains, a token or cookie is generated.

Many SSO providers have popped up in order to provide the webmasters this kind of service. Each one with its unique characteristics, strengths and weaknesses, such as: SAML, OpenID Connect, Facebook Connect, Microsoft Active Directory, Twitter.

Some SSO identity providers could be based on “enterprise-focused” systems and others use a “decentralized” system. The “enterprise-focused” system authenticates through social media and is better for personal use. The decentralized system with SAML accepts the authentication done by any of the nodes belonging to the network. Systems nowadays are “password-less”. Clients won’t need a set of credentials to access different applications. The possession of one tool such as cell phones or unique fingerprints enable users to move across different domains.

When it comes to choosing identity providers and the way of authentication, clients have to take into consideration the benefits and limitations of the selected provider and choose the one that satisfies their concerns.

The diversity of SSO services presents a dilemma. The bigger the organization, the worse the problem. In fact, the following scenarios are found:

A dozen of internal and third-party tools are used at once by any company.

Each tool is designed in a different way and each one connects to the others by a different protocol. Customers have their own tools provided by many other third-parties.

In the past days, organizations were not worried about integration. All corporate software was stored in one place. Now with the different social and enterprise options, clients need all systems to be integrated and interconnected.

This kind of integrated experience enables clients to deal with different identity providers:

- Authorization practices: define some access restrictions.
- Attributes exchange practices: when the different identity providers are integrated, users may find a way to avoid the duplication of their data.
- User management practices: Users may be able to manage their accounts (create, delete or update).

3.2.5.2 Benefits of FIM

As a definition, the federated identity is a system that sits between the organization (as well as users) and all its running applications. It certifies the authenticity of users by

confirming the username and the related password they have entered. By this function, we can refer to the federated identity provider as a “middleware”.

Users can access the application by using their existing Active Directory credentials through the federated identity. Actually, the related credentials are stored and managed by the Active Directory. As a result, the users’ authentication is done via “on-premises Active Directory services”.

Single Sign On

With the diversity of devices and applications that we use, we are obliged to create numerous login credentials for each one of them. Actually, it’s difficult from the user’s perspective to remember all these credentials. From one hand, IT teams have to spend a lot of time to resolve login problems. From another hand, it’s extremely time consuming for IT administrators to control these accesses. In fact, they have to manage multiple users’ identities across different applications and set a log of access right policies.

As a solution, we see the Federated Identity with what we call “Single Sign-On” (SSO). This latter could be implemented using existing Active Directory credentials. The model of “Federated Identity sign-in” allows a true Single Sign-On. Users can have the same password for all cloud applications (such as Office 365) and other third party cloud

applications. Users can access Office 365 without having to re-enter their credentials on their domain-joined computers once they are connected to the Active Directory domain.

In brief, The Single Sign On model makes the IT user experience more convenient, simpler and quicker.

Reduced Security Risks

Federated Identity increases security level. By identifying the authentication process within on-premises Active Directory, IT administrators don't have to synchronize different passwords existing on the cloud Active Directory. Actually, the authentication policy is stored on-premises, behind the firewall. Using a Single-Sign On model presents a win-win position for both users and IT admins. In fact, creating a multiple login credentials expose the organization to serious risks. In addition, it increases the potential use of weak passwords by the users. Setting a Single-Sign On policy, it's more convenient for both employees and IT teams and helps to create a strong security policy.

Increased Organizational Productivity

By switching to cloud-based applications, organizations can increase their productivity. Actually, if IT teams have to deal constantly with "multiple application logins and re-entering passwords", and if helpdesk receives always calls for password resets. This will

increase the administrative tasks within the organizations. The log in process can be simplified by using the Federated single sign on policy. As a result, the company's productivity will be improved.

The user has the benefit of having only to remember his "Domain Credentials." In a nutshell, FIM is cheaper and much more secure in the long run because it doesn't need to manage individual cloud based accounts; Usually It happens automatically.

Licenses for said cloud based applications are assigned or removed automatically.

Access to ALL cloud based applications is removed by one simple task through the unified interface where SaaS, PaaS and IaaS management take place. Moreover, the user only needs to remember ONE username and password combination. FIM allows IT to protect critical apps with Multi Factor Authentication

3.2.5.3 FIM architecture

Users

Users (subject or principal) are associated with a person. User "U" is represented by identity collection of attributes that represent properties about U. Attributes describe qualities (example Age), circumstances (example employer), behaviors (example shopping) or assigned values (example USERID). The number of attributes comprising an identity has no restrictions; identities can be small and simple (example, username and

password) or they can be large and complex (example, interconnection of qualities, circumstances and behaviors). A single user can be associated with multiple identities. Identity management systems allow users to choose among multiple digital identities.

Service Providers

Service providers (SP) authorize users based on authentication assertions. The authorization depends on the received attributes, the authentication assertion format, or properties of the party "P" that issued the authentication assertion. Service providers (for example, Google, IBM and Salesforce) implement their own identity management. In this case, users are responsible for managing a separate identity for each SP they are dealing with. Almost all users reuse the same authentication credentials with many service providers. But because only identity providers are able to manage authentication assertions, SPs are no longer in need to trust other SPs.

Identity Providers

IdPs can be standalone party or service provider itself. IdPs aim at authenticating users, storing and managing collections of attributes of these users. Users' authentication allows IdPs to determine if this particular user has its own private identity hence issuing authentication assertions. Moreover, IdPs have the ability to provision all identities

which means to create, update, release and delete any record whether it was attribute or identity.

Trust establishment in FIM

This section describes how trust is established between IdPs and SPs. We shall mention two methods of trust due to their wide usage and solid architecture: Static and dynamic trust establishment.

Static Trust Establishment

Trust is predefined between IdPs and SPs. This trust can be through negotiation between the two parties or during implementation phase where one party should - at runtime - communicate with other entities. Many models can be used to implement static trust establishment such as:

Model 1: By Chen et al [19], this model allows interoperation paths to be discovered inside IdPs based on different circles of trust (CoT). The model presents how the trust can be established between CoTs to allow path interoperation and discovery. Authentication Assurance Level (AAL) conversion is designed and role mapping is also implemented to improve level of interoperation security.

Model 2: By Jiang et al [20], this model implements a new entity called Trust Service Provider (TSP) which allows and at runtime, to establish and manage trust relationship

between federated entities. TSP requires registration in order to obtain certificate, hence parties can communicate through secure and private channel. TSP is considered third trust party where federated parties share their metadata.

Dynamic Trust Establishment

When it comes to huge number of IdPs and SPs, static trust establishment would not be the right choice, hence comes the concept of dynamic trust establishment which is dynamically made at runtime and not offline like in static model. Dynamic trust is based on metadata provided by IdPs and SPs along with their SLA and reputation. Dynamic trust can be one of the below models:

Model 1: By Bhonsle et al [21], suggest a model which implements Efficient Trust and Identity Management System (ETIS) where trust third party is not mandatory. ETIS allows SPs to establish trust between themselves without going through third party. User's attributes needed by SPs to authorize another user id defined as access control policy definition and can be automated to allow ease establishment between two SPs.

Model 2: Marmol et al [22], suggests a model called Trust and Reputation Model for Identity Management Systems (TRIMS) that offers an acceptable security level where multiple domains can decide about their reliability and exchange sensitive user attributes. When client requests web services from Web Service Provider (WSP), it

requests by his turn some information from the IdP. In this scenario, IdPs acts as basic role to hold identity information based on users requests.

Model 3: Kanwal et al [23], proposed a Trust Establishment Model that evaluates the trust level of Cloud Service Provider (CSP). The model has sub-modules as follows: Registration Management Module, SLA Management Module, Feedback Management Module and Trust Management Module. However, this model does not monitor or update trust score of CSP.

3.2.5.4 SAML and OAuth in federated identify

SSO inside federated identity management is critical but does not ensure a high level of security. Here comes the role of Security Assertion Markup Language (SAML) to enhance security while working inside cloud federation and OAuth to allow resources sharing inside application and users through web services.

Security Assertion Markup Language (SAML)

Federated authentication should be implemented to support web services communication hence integration between users and multiple partners can securely take place. Federated authentication allows the establishment of agreements, secure trust and user authentication to allow interaction between business domains.

SAML acts as a layer to support and standardize trust between different business domains.

SAML is an XML based framework for exchanging user authentication and attribute information [24]. SAML is developed by OASIS to allow entities such as users, resources and attribute information to make assertions about attributes and authorizations to another user.

SAML 2.0 provides protocols to define communication sequence during request and response messaging. Moreover, SAML supports HTTP and SOAP.

Benefits of SAML 2.0

Platform neutrality

SAML abstracts the security framework away from platform architectures and particular vendor implementations. Making security more independent of application logic is an important tenet of Service-Oriented Architecture.

Loose coupling of directories

SAML does not require user information to be maintained and synchronized between directories.

Improved online experience for end users for which SAML enables single sign-on by allowing users to authenticate at an identity provider and then access service providers

without additional authentication. In addition, identity federation (linking of multiple identities) with SAML allows for a better-customized user experience at each service while promoting privacy.

Reduced administrative costs

Using SAML to 'reuse' a single act of authentication (such as logging in with a username and password) multiple times across multiple services can reduce the cost of maintaining account information. This burden is transferred to the identity provider.

Risk transference

SAML can act to push responsibility for proper management of identities to the identity provider, which is more often compatible with its business model than that of a service provider.

SAML with SSO

SAML was introduced as domain model, which consists of Credential Collector, Authentication Authority, Session Authority, Attribute Authority, and Policy Decision Point. These are the key system entities in providing single sign-on service to service requesters.

Credential Collector

A system entity used to gather user credentials for authenticating with the associated Authentication Authority, Attribute Authority, and Policy Decision Point (PDP)

Authentication Authority:

A system entity used to produce authentication assertions.

Session Authority: A system entity (for example, identity provider) to maintain the state related to the session.

Attribute Authority: A system entity that produces attribute assertions.

Attribute Repository: A repository where attribute assertions are stored.

Policy Repository: A repository where policies are stored.

Policy Decision Point: A system entity that makes authorization decisions for itself or for other system entities that request authorization.

Policy Enforcement Point: A system entity that enforces the security policy of granting or revoking the access of resources to the service requester.

Policy Administration Point: A system entity where policies (for example, access control rules about a resource) are defined and maintained.

SAML Assertions

A SAML assertion is data information issued by SAML authority. It can be an authentication action performed on a subject (for example, service requester), attribute information about the subject, or an authorization request (for example, whether the service requester can access a resource).

SAML assertions can have three models:

- **Authentication Assertion:** It carries business data about successful authentication performed on a subject (for example, a service requester).
- **Authorization Decision Assertion:** An assertion that carries business data about an authorization decision. For example, the authorization decision may indicate that the subject is allowed to access a requested resource.
- **Attribute Assertion:** An assertion that carries business data about the attributes of a subject.

SAML flow diagram

Figure 14 shows SAML system sequence diagram between clients, IdPs and SPs. Clients are normal end-user, IdPs can be any third party entities with identity databases and SPs are service providers willing to rent or sell their cloud services such as SaaS in this case.

1- User log in to IBM.com (SP) where SaaS services are available to buy. IBM.com does not manage authentication itself.

2- IBM.com needs to authenticate the user, hence it builds SAML authentication request (Authnrequest), sign it (optional encryption) and finally encodes it. Then the user's web browser is redirected to the IdP for authentication. Here, the IdP receives the request, decodes it, decrypts it if necessary and verifies the attached signature.

3- Now the Authnrequest is valid, the IdP will redirect the user to login portal where he enters the credentials.

4- When the user logs in, the IdP generates a SAML token containing some user's information such as username, email, location, etc. Then the IdP redirects the user to IBM.com with the required SAML token.

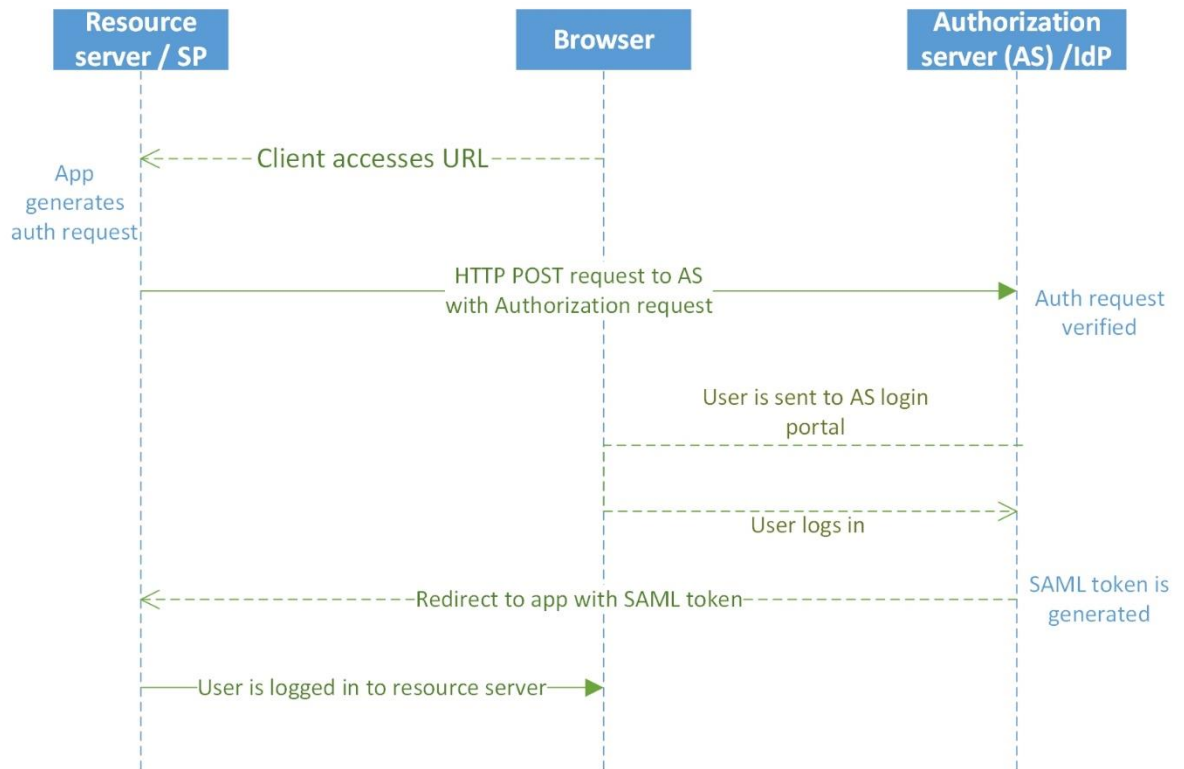


Fig. 14 Traditional SAML system sequence diagram

5- IBM.com in its turn, receives the SAML token and verifies it, decrypts it if necessary and extracts user's identity information such as userID and their permissions. The user now might log to IBM.com and perform any desired task. The IdP in this case does not hold user's credentials.

SAML architecture

Figure 15 shows how the SAML domain model is mapped to SAML logical architecture. The diagram shows how a user requests access to remote resources under an SSO environment.

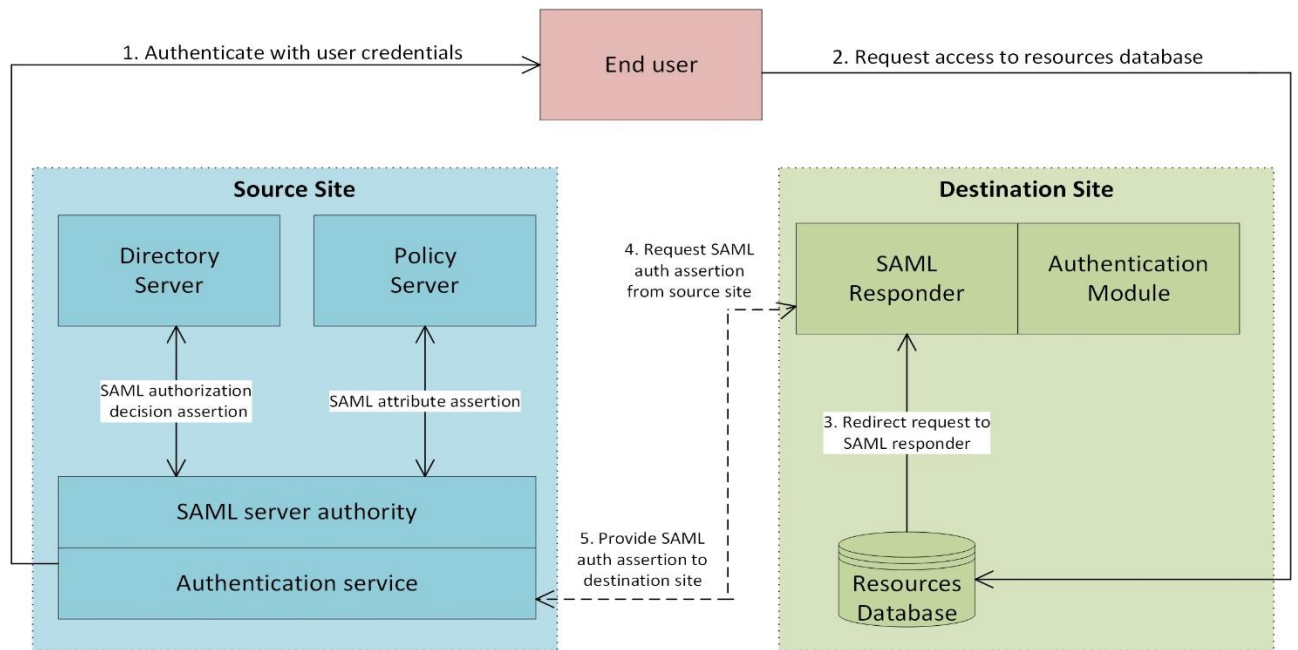


Fig. 15 Traditional SAML architecture flow diagram

The destination site has remote resources, cloud services and authentication system with authentication sub-system (Policy Enforcement Point). It also has SAML Responder that can manage user requests for resources and build SAML assertion requests. The source site has authentication service (Authentication Authority), directory server (Attribute Authority that stores the policy attributes such as Microsoft LDAP or UNIX BIND), and a policy server (Policy Decision Point). The SAML server (or authority) manages requests for SAML assertions and responds to the SAML Responder.

SAML sample code

In the following section, we will show few SAML codes and the logic behind them.

- **General SAML request**

The SAML request contains SOAP envelope and SOAP body that contain the SAML request `<samlp:Request>`.

The SAML request element may contain the elements `AuthQuery`, `AttributeQuery`, or `AuthDecisionQuery`. SOAP message might also contain a digital signature `<ds:Signature>`

Code 2 General SAML request in XML format

```
<env:Envelope
xmlns:env="http://www.w3.org/2001/11/soap/envelope/">
<env:Body>
<samlp:AuthnRequest
  xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
  ForceAuthn="true"
  AssertionConsumerServiceURL="http://www.IBM.com/"
  AttributeConsumingServiceIndex="0"
  ProviderName="string"
  ID="ID329740"
  Version="2.0"
  IssueInstant="2017-12-01T01:00:00Z"
  Destination="http://www.IBM.com/"
  Consent="http://www.IBM.com/">
<saml:Subject
  xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
<saml:NameID
  Format="urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress">
  nagemayel@ndu.edu.lb
</saml:NameID>
</saml:Subject>
</samlp:AuthnRequest>
</env:Body>
</env:Envelope>
```

General SAML response

SAML response includes the <Status> element with the SAML assertion statements, such as AuthenticationStatement, AttributeStatement, and Authorization DecisionStatement.

The below example shows SOAP message for a SAML response.

Code 3 General SAML response in XML format

```
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body>
  <samlp:Response xmlns:samlp="/" xmlns:saml="..."
    xmlns:ds="...">
    <Status>
      <StatusCode
value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
    </Status>
    <saml:Assertion>
      <saml:AuthenticationStatement>
        ...
      </saml:AuthenticationStatement>
    </saml:Assertion>
  </samlp:Response>
</SOAP-Env:Body>
</SOAP-ENV:Envelope>
```

3.2.5.5 OAuth

OAuth was developed for Twitter OpenID project. The second version OAuth 2.0 was released in 2012 [25].

OAuth main goal is for user sharing resources and information such as media, books and data without sharing their usernames and password but instead sending tokens.

OAuth general architecture

Figure 16 shows system sequence diagram of OAuth flow between a client and resource.

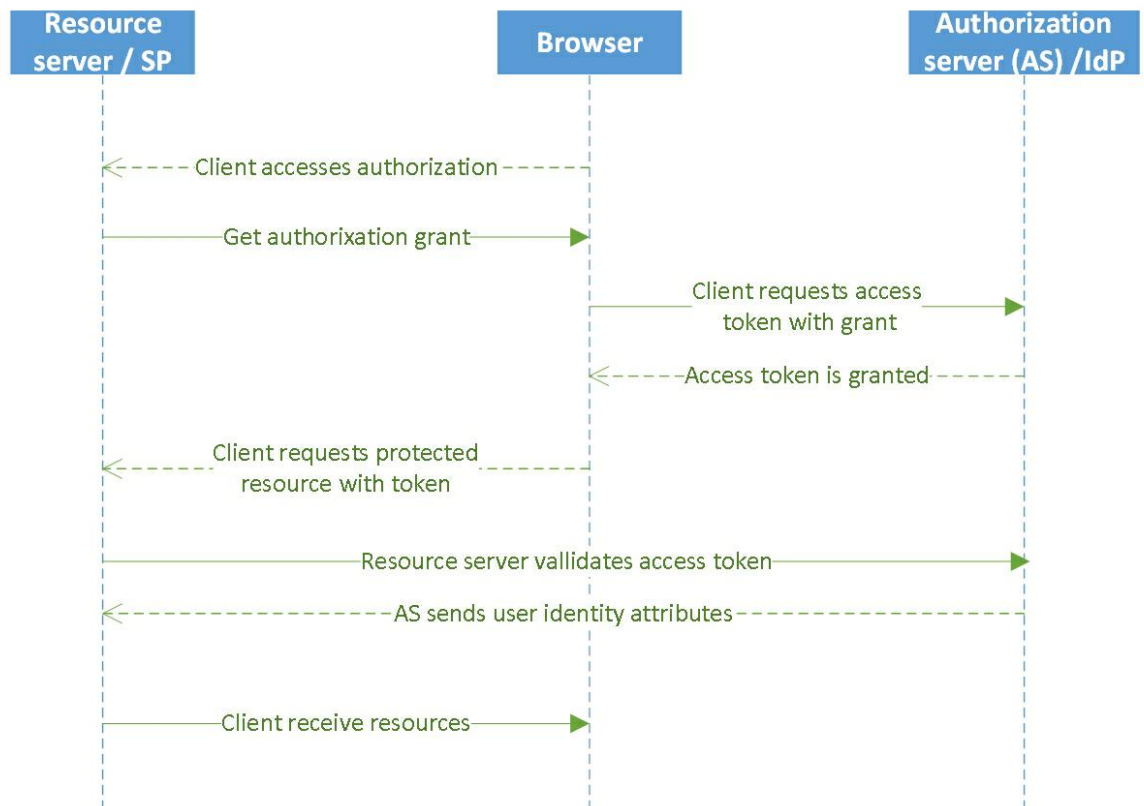


Fig. 16 Traditional system sequence OAuth flow diagram

The above scenario can be described as follows:

1- User goes to IBM.com which stores all services. IBM.com has no private authentication system, hence the user is redirected to the authorization server to get authorized. The user is presented with a login form to input his credentials, then after, he should accept terms and conditions of the resource server (IBM.com) to act on his behalf. User logs in and go to IBM.com.

2- The user gets an authorization grant code.

3- The Client then uses that authorization grant code to request an access token from the Authorization Server (AS).

4- If the authorization grant code is valid, then the AS grants an access token to be used by the client when requesting resources from IBM.com.

5- IBM.com receives the request for resources with the access token. In order to make sure it's a valid access token it sends the token directly to the AS to validate. If valid, the Authorization Server sends back information about the user.

6- Now the user's request is validated, IBM.com sends the requested resource back to the user.

To get OAuth tokens, users should get granted by the below grant types:

Authorization code: This allows the resource owner to grant access and an authorization code is issued which can be embedded inside URL. By his turn, the client exchanges the URL with an access token.

Implicit grant: Mainly for browser applications on a client side. The resource owner grants access to the resource and a new token is provided.

Client credentials: Allows applications to gain access to resources owned and managed by the client.

Device profile: This grant type supports devices with no browser. Game consoles are good examples.

SAML bearer assertion profile: It enables the exchange of SAML 2.0 assertions for an OAuth access token. This grant is mainly used for integration with cloud applications. It supports server-server communication which is mainly advantageous for cloud systems.

In our paper, we are interested in SAML bearer assertion profile to be used in cloud federation systems, with broker-based server acting as middle layer between clients and service providers.

SAML and OAuth for communication

The user sends request to access through OAuth 2.0 client application (mobile device or laptop). The client application will contact the IdP to obtain a SAML 2.0 access token. IdP can be a trusted third party service or a security token service (STS) inside IAM system. STS can generate token services in different formats. When the token is assigned, the client application calls the authentication server to exchange the SAML 2.0 token with its own OAuth 2.0 token. The client application is now able to access the desired resource server based on the resources identified in the token. Combining SAML and OAuth would have the below message flow as shown in Figure 17.

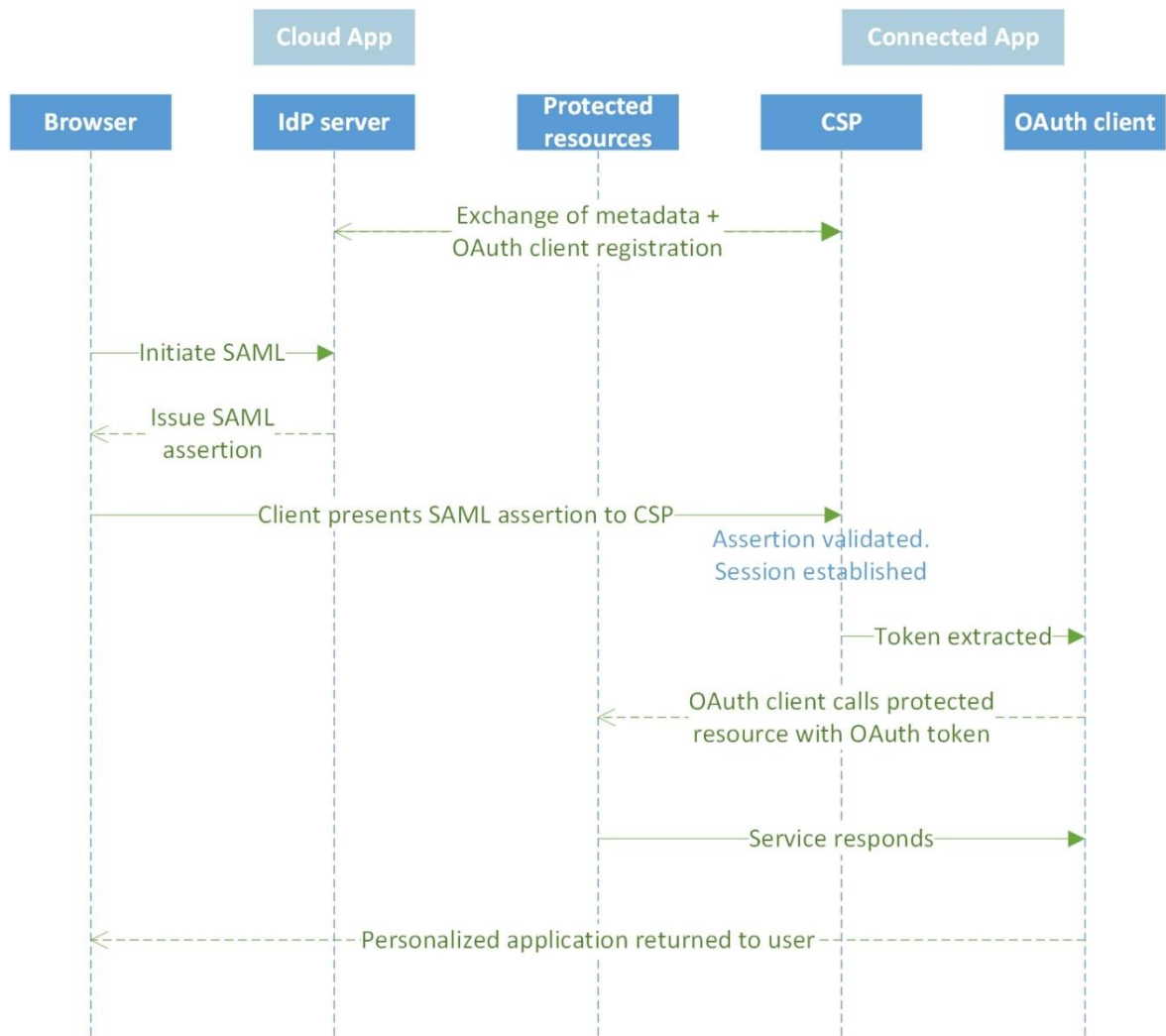


Fig. 17 Traditional SAML and OAuth combination's system sequence diagram

When a developer launches a new Connected Application in the marketplace, they've established the appropriate SAML metadata to allow for single sign-on between the IDP and the SP. When an admin installs the Connected Application into their tenant in the multi-tenant Cloud Application, they explicitly grant permission for the Connected Application to access protected resources. This delegates authority for all users to the application.

When a user logs into the Cloud Application, they see a list of Connected Applications they have access to. The user clicks on a link from this list, which kicks off a SAML exchange between the two parties. When a SAML response is sent to the Connected Application, it includes an OAuth Response as part of an Attribute Assertion.

This response includes an access token (and optional response token) is valid for the subject of the assertion, and for the resources which the admin granted permission for.

The Connected Application receives the SAML Assertion, processes it, establishes session for the user, and uses the OAuth token to call protected resources in order to personalize the experience for the user.

3.2.6 Content Based Access Control (CBAC)

It is an innovative access control model designed for content centric information sharing.

It is applied where RBAC will give more access right; on top of such model CBAC is deployed. The CBAC model takes access control decisions based on content similarity. In

CBAC, subject can use the underlying RBAC model to access all large set of objects but with additional restriction to subject where the subject could access subset of designated

record. Boundary of the subset is dynamically determined by the textual content of data objects. Content-based access control (CBAC) grants or denies a customer's request

based on the content that is sent. In most cases content-based access control is used

along with identity processing, but it can be used on its own for granting or denying API requests by identifying threats, verifying signatures or validating structure of the content and messages being sent.

To implement CBAC, one of the first things that need to be accepted is that within an API economy, cloud, mobile or B2B connections, each use different formats to send messages such as XML, JSON, SOAP, and HTML.

Any content-based access control solution must be capable of inspecting content across any of these message formats. Once this is established, engineers can set a content-based access policy to inspect through the following areas:

- **Threat Detection:** Perform deep content inspection to check for malware destined for the target application. One common attack where a CBAC policy can be effective is the XML External Entity (XEE) attack, where an off-the-shelf XML parser is poorly configured to allow XML that contains references to external entities. Detecting an XEE is critical for preventing corporate data breaches [26].
- **Signature Verification:** The message here is inspected to see that it has not been tampered with nor changed. Signature verification establishes the integrity of the messages.

- Schema Validation: Depending on the message format, an XML Schema Definition (XSD) may be available to check the structure of a message. In cases where schemas are not available, being able to define acceptable message structures and data types is recommended to ensure that the content is fully aligned with expected business semantics.

As part of an overall access control policy, and by setting the gateway in front of the application servers and allowing content-based access control, administrators can protect applications and application servers.

In many contexts, users are either unable or unwilling to specify their access control policies. In Data Loss Prevention, for example, users cannot fully express what is secret in rule-based formats.

Many users are unwilling to use access controls, particularly in the Web 2.0, because they are too draconian, leading to disastrous consequences in terms of privacy.

To address both of these issues, we have introduced the concept of Content-Based Access Control (CBAC). CBAC combines content recognition with policy acquisition and enforcement.

A CBAC-enabled system can be trained to recognize policy violations by learning what is secret from examples.

3.2.7 Policy Based Access Control (PBAC)

PBAC is an access model that helps companies in implementing solid access controls based on clear and well defined policy and requirements. PBAC is considered a harmonized and standardized form of ABAC model at an enterprise level. PBAC gathers attributes from resources, environments and requesters with specific information on the conditions under which the access request was made. PBAC also utilizes rule sets that tell whether, under organizational policy, the access is allowed for those attributes under those conditions [27].

Under the PBAC model, enterprises might have only one policy that manages access to sensitive or critical resources regardless the location or the owner of data.

PBAC is more complicated than ABAC; PBAC attributes should be designed, deployed and maintained in enterprise high level systems. Examples would be like databases, directory services, and other middleware and management applications, all of which must be integrated.

Moreover, PBAC requires complicated algorithm to manage access based on attributes and in the same time, a mechanism to build and manage policy rules in unambiguous way, else illegal access to information resources can be achieved. The extensible Access

Control Markup Language (XACML) is XML-based language developed to specify access control policy in a readable format. Policy creation is not easy even with the use of ACML. Attributes used across the enterprise must be the same and from authoritative sources only. That means Authoritative Attribute Source (AAS) is the one source of attribute data authorized by the enterprise which overrides all other attribute sources.

3.3 Chapter summary

To summarize the above details, we can deduce the following:

MAC access control is a policy, software or hardware module used to limit access to a resource. This could be a password or set of permissions granted to the resource. When applied, several levels of security must be passed. DAC is based on user identity and user must show identification. This might involve showing a badge or driver's license, entering a logon ID or swiping a card.

RBAC Authenticate: the user is authenticated to the network. This can be accomplished with a password, PIN, hand scan, or signature.

RBAC Authorize: The system restricts the user's access to a particular resource based on a predetermined set of policies. RBAC is scalable and secure comparing to older access control systems. ABAC enables access to objects by assessing rules against the attributes of elements (objects and subjects), operations and area connected to the entities.

ABAC permits higher number of inputs than RBAC which allows giving higher blends of factors to a bigger combination of parameters and rules to express policies only restricted by programming language. This malleability empowers development of access control rules without considering individual connections between subjects and objects. As a definition, the federated identity is a system that sits between any organization and all relative applications. It certifies the authenticity of users by confirming the username and the related password they have entered. By this function, we can refer to the federated identity provider as a “middleware”. Users can access the application by using their existing Active Directory credentials through the federated identity.

CHAPTER 4. Evaluation of Access Control Mechanisms

4.1 Introduction

In this section, we will evaluate each of the previous access control mechanisms and deeply talk about their drawbacks and limitations in terms of security, administration, complexity, policy design, etc.

4.2 Evaluation and comparison of IAM mechanisms

4.2.1 Mandatory Access Control (MAC)

MAC can induce over-classification of data due to high watermark principles which as a result deficits the productivity by setting a boundary on the ability to transfer data between systems and restring user control over data. MAC does not address the dynamic separation of duty or security or validation of security components.

MAC systems are usually high priced and difficult to use due to their reliance on the trusted components and their needs of applications for the mac labels and properties.

Rumor has it that Microsoft abandoned the idea of implementing MAC in their OS due to the issues that were emerged in the rewriting of the applications needed [28]. MAC models place restrictions on user access that do not allow for dynamic alterations. The associated utilities and OS have to be places outside the access control frame work. MAC

requires planning for it to be consistent and needs high system management due to the constant update object and account labels to collect new data.

4.2.2 Discretionary Access Control (DAC)

Even though DAC shows positive potential in various group sizes, it also holds a great deal of negative points to be considered. Since the end-users do not know what information their peers have access to, they cannot be sure if the other's data is the same as theirs. This could be a case if the manager is not well experienced and has no supervision by another member of greater experience to not allow for such incidents to happen. That is because the manager is the only one who has access to all of the resources and how they are distributed [29].

Moreover, DAC is susceptible to Trojan horse, since the system is open and allows users to control object access permissions. Additionally, DAC permits the access rights to owned objects, which leads to a more difficult maintenance and verification of security principles. DAC is also inherent to safety problems due to the lack of constraints and copy privileges. This lack of copy privilege prevents the verification of information theft and protection which is also an assistant to potential exploits for Trojan horses. DAC has trouble to ensure consistency since it grants users to decide access control policies on their policies which happen to be global. Malicious software/program: DAC is vulnerable

because it executes malicious programs that exploits the authorization of particular users on behalf of whom they are executing i.e. Trojan horse. Information flow: It is possible to get a copy of the original information if the DAC has no control on the flow of info.

4.2.3 Role Based Access Control (RBAC)

RBAC is coarse-grained. For example, role called "Doctor" is given permission to "view medical record". That would give the "doctor" the right to view all medical records including their own. This is what leads to role explosion.

RBAC is static. RBAC cannot use contextual information e.g. time, user location, device type, etc.

RBAC ignores resource meta-data e.g. medical record owner.

RBAC is hard to manage and maintain. Very often, administrators will keep adding roles to users but never remove them, which ends up with hundreds of users with huge number of roles and permissions

RBAC cannot cater to dynamic segregation-of-duty.

RBAC relies on custom code within application layers (API, apps, DB...) to implement finer-grained controls [30].

RBAC Access reviews are painful, error-prone and lengthy.

4.2.4 Policy Based Access Control (PBAC)

Under the PBAC model, enterprises might have only one policy that manages access to sensitive or critical resources regardless the location or the owner of the data.

PBAC is more complicated than ABAC; PBAC attributes should be designed, deployed and maintained in enterprise high level systems. Examples would be like databases, directory services, and other middleware and management applications, all of which must be integrated.

Moreover, PBAC requires complicated algorithm to manage access based on attributes and in the same time, a mechanism to build and manage policy rules in unambiguous way, else illegal access to information resources can be achieved. The extensible Access Control Markup Language (XACML) is XML-based language developed to specify access control policy in a readable format. Policy creation is not easy even with the use of ACML. Attributes used across the enterprise must be the same and from authoritative sources only. That means authoritative attribute source (AAS) is the one source of attribute data authorized by the enterprise which overrides all other attribute sources.

4.2.5 Attribute Based Access Control (ABAC)

There is a considerable list of problems related to the ABAC systems and applications due to the bigger complexity attribute used to increase the flexibility and generality of access control policies. The hybrid ABAC models and frameworks used to fix these problems are affecting the flexibility and the anonymity of the identity of ABAC.

Untraditional way of users' authorization.

Unlike traditional IAM access control mechanisms, permissions and roles assignments are handled by the security team. The process of provisioning and de-provisioning is made in place. Later on, access reviews are performed on a regular basis where user-role assignments are checked and approved by managers. However, in ABAC, user-role assignment are directly allocated through roles and permissions. Users' entitlements are the result of a runtime authorization request evaluated against a set of policies. This new form of assignment makes access reviews, provisioning and de-provisioning insufficient. In this case, ABAC requires a new process for the above actions therefore, new authorization requirements should be implemented.

Lack of requirements.

In traditional access control mechanisms, requirements are handled by applications developers who implement the requirements as codes inside the applications. In ABAC,

authorization requirements are gathered and coded as authorization policies centrally managed. Therefore, new steps should be implemented: use case definition and authorization requirements gathering.

Complex ownership of authorization

Most of the ownership and responsibility in traditional IAM lies in the central IAM team. This is done by defining coarse-grained access with RBAC system for instance, then allowing developers to implement fine-grained controls in the applications. In ABAC, the entire authorization logic is expressed inside the authorization policies. In other words, central IT team, the application owners and business analysts should work together to define requirements and agree on ownership.

Foundational Models

One of the problems is the absence of a reference and/or foundational model for ABAC. The many published ABAC models turned out to be domain specific and limited to a specific use case, and the hybrid models are missing the versatility of “pure” models. Only 3 of the generalized models (Jin et al. [31], Servos and Osborn [32], Zhang et al [33]) are both formal and complete yet none of them got qualified to be the standard model of ABAC. Even the most successful working models cited as “the model of ABAC” are problematic as foundational models. XACML, an access control policy language, is

missing the formal model of ABAC, even though it supports attributes. Wang et al [34] logic-bases framework focuses on modelling policies and their evaluation, and so cannot be considered a complete model of ABAC. The ABAC model provided by Yuan Tong, the basis for many models, is simplistic and specific to a limited domain. NIST is working on a formalized family of ABAC models and seems to be achieving promising results. Some details regarding “Framework of ABAC models” were presented at the NIST Attribute Bases Access Control Workshop that took place on July 17, 2013 by David Ferraiolo [35] that defined 4 families of ABAC models: ABAC rule, ABAC rule-hier, ABAC rel, and ABAC rel-history. However, there are still no formal definitions for these models, only a few details coming from a presentation slides. Barker suggested what could be a solution away from adopting or creating models. The solution lies in focusing on unifying meta-models and avoiding “developing the next 700 particular instances of access control models” [36]. A meta-model avoids creating new models for every small extension of the concept, and provides a unified model to describe and reason about ABAC.

Emulating and Representing Traditional Models.

ABAC’s ability to emulate the traditional models entitled it to be a more general model of access, but there is no real proof to support this claim. Jin et al.’s work proved how ABAC Alfa can be constrained to model DAC, MAC and hierarchical RBAC, but each model

has only one representation and there's no model for RBAC's separation of duty constraints. In order to develop the best practices to help in the transition to ABAC (e.g. converting existing traditional systems to ABAC systems) and formally prove that ABAC can model all possible DAC, MAC, and RBAC-based policies, a better exportation and evaluation of the different methods of representation are needed.

Hierarchical ABAC

In hierarchical RBAC, roles are related in a way similar to that of real organizations which simplifies the administration on both engineering and reviewability of existing role-based policies levels. The majority of "pure" ABAC models are missing this kind of inheritance and expressiveness. While a role can be easily modelled as a single attribute of a subject, this simplistic representation cannot emulate RBAC's hierarchical nature without allowing for complex data types in the value of an attribute (as is done in Jin et al.'s ABAC alfa [37]) or unmaintainable complex policies. "Pure" ABAC needs more simplistic means to provide hierarchical administration to be able to compete with RBAC and hybrid models. "Attribute users groups", which are hierarchical groups that inherit sets of attributes from their parent groups and allocate them to their members, may provide a solution. This technique could also work for objects and other access control entities onto which attributes may be assigned. Another technique is to allow attributes to have

direct inheritance relationships with other attributes, such that a child attribute supersedes the parent attribute in policies. However this leaves the attributes with no value and limits ABAC's usefulness.

Auditability

Being able to determine the set of users who can access certain resources or the set of resources that users can access is a major aspect of access control for legal and security reasons. This is easily done in RBAC by calculating the union of the set of effective privileges from each role assigned to the user. In ABAC, things get more complicated, ABAC being an identity-less access control system where users are only known after the access control requests are made. Even when the identities of all users and their assigned attributes are known, computing the set of permissions that results for a given user is difficult since all objects have to be checked against all relevant policies. Hybrid ABAC models use attributes for role assignment to put constraints on the permission assigned to a role addressed this issue. These models' use of hybrid strategies make them lose flexibility and identity-less access control. ABAM [38], one of the "pure" ANAC models, provides auditability to a certain extent using a predefined access matrix where subjects are assigned permission, but requiring hence the users' identity to be known and labelled in the access matrix. Complete and efficient "pure" ABAC systems auditing methods need

to be developed for administrators to comply with specific regulations and directives before the fact auditing, without which ABAC won't be usable where legal or industry regulations don't allow systems that rely on after the fact auditing techniques only.

Separation of Duties.

SoD is about many people completing a sensitive task to limit the potential error and fraud. In RBAC, people are not allowed to be given conflicting roles, which is provided by static SoD, and dynamic SoD keeps people from activating conflicting roles in the same session. In ABAC, applying this concept is yet to be explored. Applying SoD type constraints to ABAC is still an issue along with whether additional constraints through policy languages are necessary. Alipour and Sabbari [39], in an attempt to solve this problem, introduced "can't-perform" rules that keep the subject from doing certain actions on specified resources. But this solution requires knowing the subject and the possible conflicts of interest beforehand. Bijon et al. [40] suggest an attribute-based constraint specification language (ABCL) that allows constraints to be put on both attributes and attribute assignments. However, this merely defines a language to represent constraints and doesn't present a formal model or framework for their use. A solution would be the use of the SoD constraints from RBAC in hybrid ABAC models that

include roles. Yet this is achieved at the cost of flexibility and the anonymity nature of ABAC.

Delegation.

Delegation is a wanted access control highlight that enables one subject to incidentally appoint their entrance rights to a more junior subject. In Barka and Sandhu RBAC paper [41], this is regularly proficient by empowering delegation of allotted roles under certain predefined limitations and renouncement conditions, yet it has additionally been tackled from authorization appointment Wang and Osborn 2011[42]; Zhang et al. 2003 [43]; Wang and Osborn 2006 [44], in which a delegator makes and delegates a transitory role made out of a subset of their delegable consents. While delegation is considered as ABAC-based encryption [Waters 2011 [45]; Servos et al. 2013 [46]], a couple of endeavors to date have been made to apply a delegation model to ABAC. Such a model of delegation could be deployed on attributes between clients and delegation of permissions consents granted by policies. Delegation of attributes could not be fully supported using X. 509 attribute authentications [Farrell and Housley 2002 [47]; Farrell et al. 2010 [48]]. In all cases, this scenario requires extensive authentication chains to be transmitted as a major aspect of a user's attribute sensitive information and could also lead to privacy concerns. In addition, attribute certificates are an execution thing but not

a formal piece of a delegation model. Dynamic delegation of authorizations is more difficult because of the attributes may change bringing new permission assignments.

Attribute Storage and Sharing

While multiple attribute resources are utilized in an ABAC system (e.g. using attribute authorities from distinctive corporations in a distributed system) complications can arise in phrases of both comparing the trustworthiness of attributes and making sure that differing attribute resources are the usage of compatible attributes (e.g. the use of the identical namespace and information type for common attributes). The problem of trustworthiness is frequently treated through relying on pre-present trust relations negotiated among organizations earlier than get admission to manage takes place; but, in peer- to-peer situations this can be hugely more complex. Shafiq et al. [49] provide a potential answer in their hybrid ABAC model that consists of a trust assessment and negotiation framework that both presents a trust assessment of claimed attributes and a way to dynamically set up trust between participating organizations. But, in peer- to-peer situations this may be hugely complex. Shafiq et al. provided an ability solution in their hybrid ABAC version that consists of a trust assessment and negotiation framework that both present a trust assessment of claimed attributes and a way to dynamically set up trust among participating organizations. Lee et al. [Lee et al. 2008 [50] suggest an

“attribute aggregation structure” in which attributes are collected from neighboring peers and evaluated using a reputation-based trust scheme wherein “every peer decides its reputation about different peers primarily based on its personal experiences, and the trustworthiness of a peer is evaluated with the help of aggregated reputation”. It is possible that Shafiq’s, Lee’s different research in dynamic trust negotiation might be without problems implemented to “pure” ABAC fashions; but, most work in this area has assumed attributes are derived from a trusted source. Ensuring attributes from unique resources are well matched could probably require a typically regular namespace or ontology of attribute names or alternatively some method of mapping attributes to equivalent representations (as recommended in Hu et al. 2013 [51]). A second problem in attribute sharing is making sure the confidentiality of sensitive user attributes. This is particularly a challenge when ABAC architecture is utilized in domain names including health care where leaking attributes approximately a user or object could be potentially compromising. Contemporary images related to attribute privacy or confidentiality have in large part been constrained to attribute-based encryption applications however some efforts have been made closer to regular privacy keeping attribute sharing protocols [Camenisch et al. 2010 [52]; Ardagna et al. 2010 [53]; Esmaeeli and Shahriari 2010[54]; Zhang et al. 2013][55].

Scalability.

Before implementing ABAC as described in the NIST Guide [56] we should take into consideration ABAC scalability. Traditional access control architecture like RBAC have been highly adopted in complex systems but ABAC still not proven in terms of efficient scalability. ABAC needs complex interconnections between access control entities that may be distributed on different network resources. In complex systems with hundreds of users, permissions, and policies, it is not clear how ABAC solutions can be managed in terms of administration and computing resources required. Complex studies of large systems utilizing ABAC concepts are needed to calculate the feasibility and usability of ABAC.

Administration and User Comprehension

Lee and Winslett [57] tackled human factor difficulties in ABAC systems. They came up with open problems related to management and usability. These problems can be divided into three main categories as follows: “Access Control Comprehension”, “Technology Management” and “Policy Specification and Maintenance”. “Access Control Comprehension” describes user’s ability to understand the access decisions based on their access requests. Classical access control models along with their decisions are clear and straightforward, in the sense of users acting as members of role assigned

permissions; like in RBAC. In ABAC case, decisions are based on complex policies which contains unstable attributes of multiple users and many other entities. This insufficient understanding of ABAC architecture and its related policies, make user access comprehension difficult and chaotic. Yao et al. [58] worked on visualizing ABAC policies and decisions towards solid solution. “Technology Management” which means managing credentials by users working in ABAC environment; ABAC subject credentials are complex and based on algorithms such as cryptographic credentials, X.509 certificates and attribute sources. Users in ABAC have difficulties in managing PGP certificates for signing and encrypting emails. “Policy Specification and Maintenance” addresses the complexity in ABAC administration and policy engineering. Almost all ABAC models cannot provide complete (or even partial) administration models. Policies are based in XML format (such as XACML) and requires intervention made by administrators and engineers. Moreover, the distributed design of ABAC means that this model is not centralized but divided among multiple policy administration points and attribute stores. This issue requires lot of training and administrative users to enhance users' ability to understand and work with policies and security configurations.

XACML challenges

Challenge #1: it is complex to write policies

XACML relies on Policy set. It is written using XML which is so sensitive [59]. We have to use XML editor to avoid syntax errors. When there are many interdependent policies the development of XACML will be so complex and difficult.

Challenge #2: The full impact of XACML policies is difficult to understand

Because the XACML is complex it is easy to be misused. Mistakes can happen since the understanding of the dimensions of policies is complex.

Challenge #3: Performance

XACML solutions have performance problems related to the policy evaluation process and policy structure for the below reasons:

Real-time policy evaluation: policies are evaluated per request which creates a big load on the system and slowdown the system response

Approving each access request: Since each access request has to be evaluated this will result in a delay.

Policy matching and attribute retrieval: Because of the complex policies and attributes, the access has to be evaluates to give an adequate result which will cause a delay.

ABAC general domain and specific domain models comparison

ABAC models can be divided into two main parts: General architecture domain and domain specific model. In Table 3, we have surveyed six of the well-known general ABAC architecture for a complete and deep analysis and comparison, based on attributes like objects attributes, environment attributes, user attributes, policy language, delegation, formal and complete model, etc.

Table 3 contains all general ABAC models presented by Wang et Al. [60], Zhang et Al. [61], Jin et Al. [39], Carlos E Rubio-Medrano et Al. [62], Ferraiolo et Al. [63] and others.

The comparison made showed that almost all ABAC models had object and user attributes. In addition, they are considered as formal and complete model. However, none of the ABAC general models contained connection attributes, recursive rules, user and object groups, delegation and trust.

Table 3 ABAC model comparison - General architecture

ABAC model comparison - General architecture						
Attributes \ Authors	Wang - 2004	Zhang - 2005	Jin - 2012	Ferraiolo - 2011	Rubio-Medrano - 2013	Servos-Osborn 2014
Object Attributes	NO	YES	YES	YES	YES	YES
Environment Attributes	NO	NO	NO	NO	YES	YES
User Attributes	YES	YES	YES	YES	YES	YES
Functional Specification	NO	NO	YES	NO	NO	NO
Connection Attributes	NO	NO	NO	NO	NO	YES
Mutable Attributes	NO	NO	NO	NO	NO	NO
Hierarchical	NO	NO	NO	NO	NO	NO
Policy Language	NO	N/A	YES	NO	N/A	YES
Recursive Rules	YES	NO	NO	NO	NO	NO
User & Object Groups	NO	NO	NO	NO	NO	YES
Separation of Duties	NO	NO	NO	YES	NO	NO
Delegation	NO	NO	NO	NO	NO	NO
Trust	NO	NO	NO	NO	NO	NO
Formal Model	YES	YES	YES	YES	NO	YES
Administration Model	NO	N/A	YES	YES	N/A	YES
Complete Model	NO	YES	YES	YES	YES	YES

ABAC domain architecture models were also surveyed and a comparison was made based on same factors used to compare ABAC general models in Table 3. The results are

shown in Table 4. ABAC models were chosen based on their reputation and citations used to study and enhance current ABAC models. E. Yuan and J. Tong [64], Hai-bo Shen and Fan Hong [65], Jian Shu Lianghong et Al. [66], Haibo Shen [67], Florian Kerschbaum [68], Bo Lang et Al. [69] . Almost all models did not contain trust or administration model. The domain used inside these models were all almost dedicated for web services and few of them for mobile systems and grid systems. Table 5 contained the rest of ABAC models that have been compared: Daniel J Buehrer et Al. [70], Feng Liang et Al. [71], Jian Shu Lianghong et Al. [72], Mike Burmester et Al. [73], Waleed W Smari et Al. [74] and Yongsheng S Zhang et Al. [75].

Table 4 ABAC model comparison - Domain architecture (2005-2010)

ABAC model comparison - Domain architecture (2005-2010)						
Authors	Yuan and Tong - 2005	Shen and Hong - 2006	Xia and Liu - 2009	Shen - 2009	Kerschbaum - 2010	Lang - 2010
Domain	Web services	Web services	Web services	Web services	mobile systems	Grid systems
Object Attributes	YES	YES	YES	NO	YES	YES
User Attributes	YES	YES	YES	YES	YES	YES
Environment Attributes	YES	YES	YES	NO	NO	YES
Connection Attributes	NO	NO	NO	NO	NO	NO
Mutable Attributes	NO	NO	NO	NO	NO	NO
Policy Language	XACML	XACML	XACML	XACML	XACML	XACML
Hierarchical	NO	NO	NO	NO	NO	NO
Recursive Rules	NO	NO	NO	NO	NO	NO
Trust	NO	NO	NO	NO	NO	NO
User & Object Groups	NO	NO	NO	NO	NO	NO
Separation of Duties	NO	NO	NO	NO	NO	NO
Delegation	NO	NO	NO	NO	NO	NO
Functional Specification	NO	NO	NO	NO	NO	NO
Formal Model	YES	YES	YES	NO	YES	NO
Emulates Traditional Models	N/A	N/A	N/A	N/A	N/A	N/A
Administration Model	NO	NO	NO	NO	NO	NO
Complete Model	YES	YES	YES	NO	NO	YES

Table 5 ABAC model comparison - Domain architecture (2010-2014)

ABAC model comparison - Domain architecture (2010-2014)						
Authors	Buehrer and Wang - 2012	Liang - 2012	Dan - 2012	Burmester - 2013	Smari - 2014	Zhang - 2014
Attributes						
Domain	cloud computing	collaborative systems	Web services	realtime systems	collaborative systems	Web services
Object Attributes	YES	YES	YES	YES	YES	YES
User Attributes	YES	YES	YES	YES	YES	YES
Environment Attributes	YES	YES	YES	YES	NO	YES
Connection Attributes	NO	NO	NO	NO	NO	NO
Mutable Attributes	NO	NO	NO	NO	YES	NO
Policy Language	Class algebra	XACML	XACML	N/A	Undefined	XACML
Hierarchical	NO	NO	NO	NO	NO	NO
Recursive Rules	NO	NO	NO	NO	NO	NO
Trust	NO	NO	NO	NO	YES	NO
User & Object Groups	NO	NO	NO	NO	NO	NO
Separation of Duties	NO	NO	NO	NO	NO	NO
Delegation	NO	NO	NO	NO	NO	NO
Functional Specification	NO	NO	NO	NO	NO	NO
Formal Model	NO	YES	NO	NO	YES	NO
Emulates Traditional Models	N/A	N/A	N/A	N/A	N/A	N/A
Administration Model	NO	NO	NO	NO	NO	NO
Complete Model	NO	NO	NO	NO	NO	NO

4.2.6 Federated Identity Management (FIM)

Trust management

In every structure, some object is accountable to achieve the communication or trust creation between IdPs and SPs. This could be done by a centralized unit, or between IdPs and SPs themselves (Peer- Peer). User requirements allow us to choose which form to use which can be also affected by the number of IdPs and SPs. Many solutions have proposed centralized forms for organization of IdPs and SPs, but these solutions might not be possible in a large network of IdPs and SPs, as the central unit might have to tolerate a lot of data processing load, causing an incompetent structure. But if all the IdPs and SPs interconnect straightly (P2P), the resolution will become more scalable but trust establishment would be difficult to achieve.

Trust establishment

In FIM, trust is established offline through some trust cooperation procedure. IdPs and SPs might meet to work on a deal and sign an agreement for trust establishment. Sometimes IdPs or SPs have to register with the centralized unit so that other entities could trust it, but it is impossible to have one centralized unit to serve all IdPs and SPs at

the same time. The number of parties working in federation might be smaller than the number of IdPs and SPs combined together.

In this case, the user might use static trust establishment to provide more confidence and legitimate sense towards the SPs.

User privacy

User privacy can be a major concern when it comes to malicious SPs. Worst case scenario can be identity theft of users, password stealing, fraud activities and money laundry financial transactions.

Reliable access rights across Circle of Trusts (CoTs)

Users might be assigned roles and privileges in the CoT where they belong to, but when users are allowed to get services of an SP inside or outside the CoT, it will become unstable in terms of number and level of rights assigned to users. This scenario may lead to some sort of security attacks known as Escalation of Privileges attack which may lead to security compromise inside the system.

Continuous Trust Monitoring

Runtime trust monitoring can be done through multiple frameworks in order to keep evaluating the metrics and getting results of the trust relationship. Quality of services

provided by SPs might affect as well the trust which may lead to degradation in the trust relationship.

Adaptation to unexpected changes

Entities should work in dynamic environments where lot of changes might occur without previous notifications. Therefore, FIM systems should be adaptable to any potential changes or unwanted situations. Situations can be geo-locations problems or anything related to information system degradation. Table 6 shows FIM model comparison based on multiple factors [76] [77] [78] [79] [80] [81] [82] [83] [84] [85] [86].

Table 6 FIM model comparison based on multiple factors

Trust Based FIM models		Factors					
		Trust Management	Trust Establishment	User privacy	Reliable access rights across CoTs	Continuous Trust Monitoring	Adaptation to unexpected changes
Bhonsle et Al	Efficient Trust and Identity Management System	Peer to Peer	Dynamic	Yes	No	No	No
Alguliev et Al	IDM based security architecture of CC on MAS	Centralized	Dynamic	No	No	No	No
Chen et Al	Secure interoperation of IDM among different CoT	Peer to Peer	Static	No	Yes	No	No
Jiang et Al	FIM with Centralized Trust and Unified SSO	Centralized	Static	No	No	No	No
Chadwick et Al	CardSpace in the Cloud	Centralized	Static	Yes	No	No	No
Khattak et Al	Threat Model for Federated Identities in FIM Systems	N/A	Dynamic /Static	Yes	No	No	No
Chadwick et Al	Trusted Attribute Aggregation Service TAAS	Centralized	Static	Yes	No	No	No
Samlinson et Al	User Centric IDAAS	Centralized	Dynamic	No	No	No	No
Gao et Al	Dynamic Trust Model for FIM	Centralized	Dynamic	No	No	No	No
Marmol et Al	A privacy-aware trust and reputation model for IDMS	Centralized	Dynamic	Yes	No	Yes	No
Kanwal et Al	Evaluation and Establishment of Trust in Cloud Federation	Centralized	Dynamic	No	No	No	No

4.3 Integration of FIM and TEMCM

Figure 18 shows how FIM can be implemented inside BBCCFM WITHOUT implementing the above four mentioned modules.

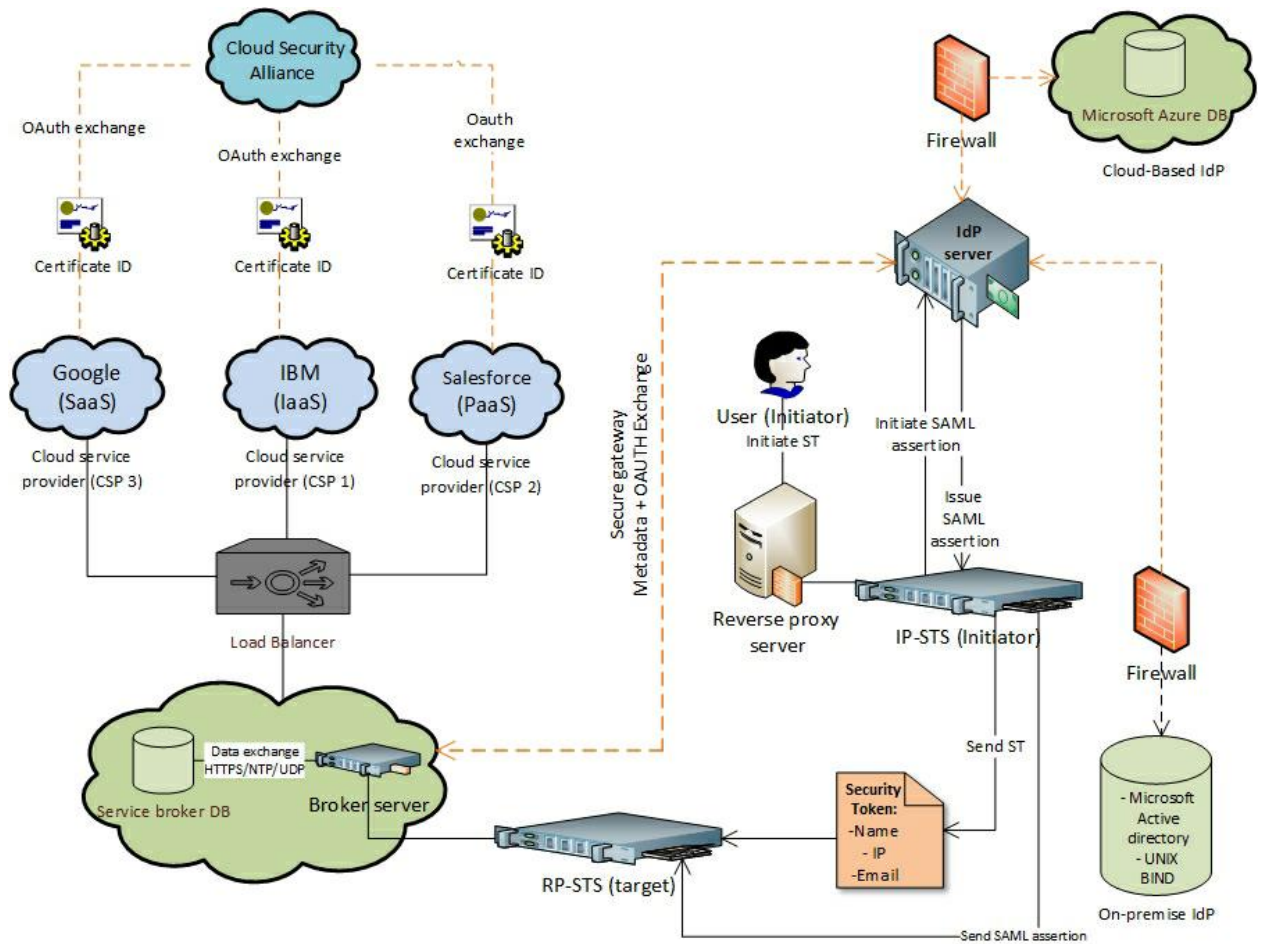


Fig. 18 Traditional FIM inside BCCFM

In this section, we will propose one solution to cover module 1 and 2: Trust Establishment, Management and Continuous Monitoring (TEMCM) acting as third-party centralized distributed system mainly used for establishing, managing and monitoring trust between IdPs and SPs in real time.

Peer-to-peer architecture is considered secure but may result in huge load when dealing with high number of SPs and IdPs. TEMCM will be based on multiple modules used to contribute in the trust establishment and monitoring:

- Reputation based trust (App 1)

- SLA verification based trust (App 2)
- Cloud transparency mechanisms (App 3)
- Trust as a service (App 4)
- Formal accreditation, audit, and standards (App 5)

TEMCM would have the following architecture as shown in figure 19.

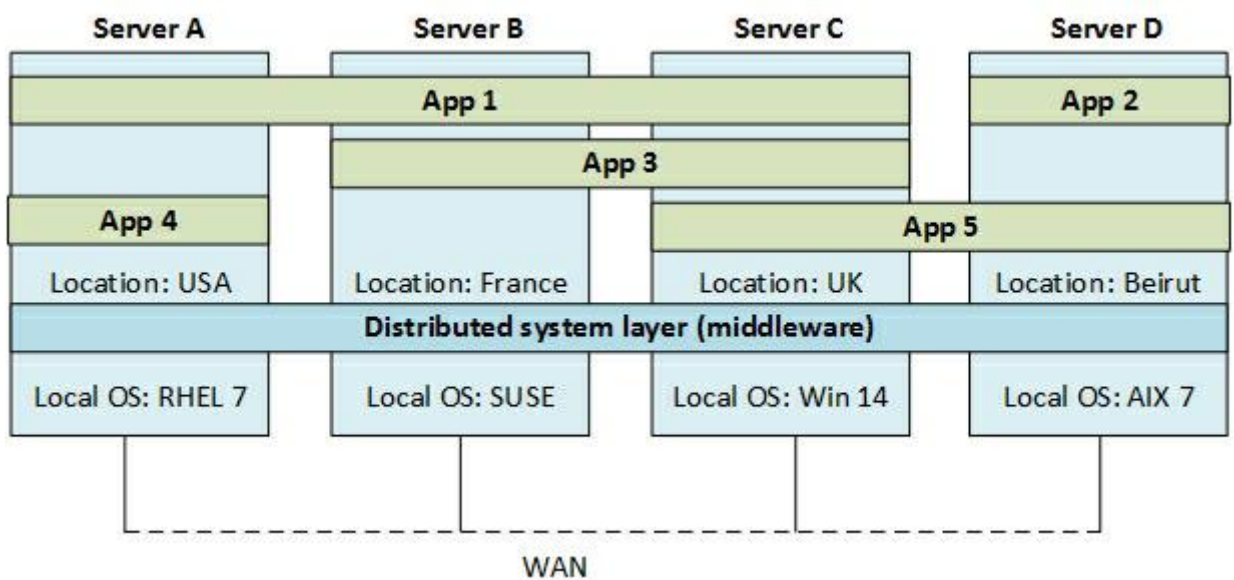


Fig. 19 Trust Establishment, Management and Continuous Monitoring (TEMCM)

Modules inside TEMCM will be distributed across different servers running in multiple geo-locations areas. This may bring more performance when dealing with huge numbers of SPs willing to manage their data.

Moreover, distributed systems bring extra scalability when it comes to adding resources on current running systems.

Figure 20 shows TEMCM inside FIM based BBCCFM.

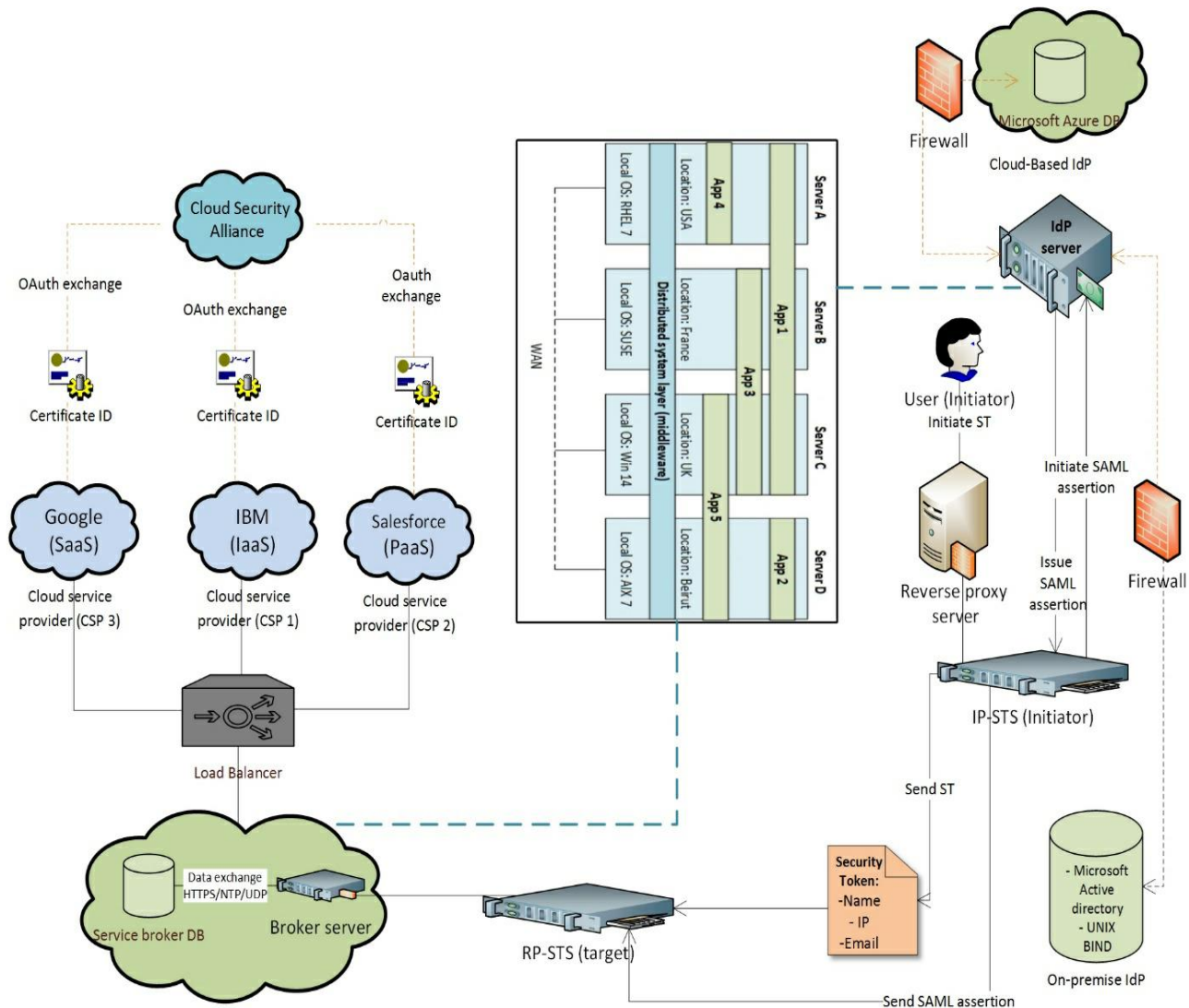


Fig. 20 TEMCM combined with FIM inside BBCCFM

4.4 Chapter summary

This section summarizes the drawbacks of RBAC, ABAC and FIM. MAC and DAC were not mentioned since they are automatically included inside RBAC access control system. In brief, we can deduce that RBAC is coarse-grained, static and ignore resources meta-data. It is hard to maintain and has no dynamic flow when dealing with multiple computer

systems. ABAC has an increasing number of problems that affect its way of user's authorization; Lack of requirements and complex ownership of authorization. Till present ABAC has no final and solid model that emulates and represent the traditional model. ABAC is hard to audit since number of users along their roles cannot be checked and controlled like in RBAC. ABAC induces XACML challenges like complexity in writing policies, full impact of understanding XACML policies and performance of XACML policies. The above tables show that almost all ABAC models has no connection attributes, no mutual attributes, no recursive rules, no SOD, no delegation of duties and no trust management. On the other hand, FIM has also few drawbacks that makes it incomplete in terms of trust management, establishment and continuous trust monitoring, user privacy, reliable access rights across circle of trusts and adaptation to unexpected changes. Table 7 shows a brief comparison between MAC, DAC, RBAC and ABAC. Moreover, we showed that FIM cannot satisfy the needs of BBCCFM without integrating it with TEMCM module.

Table 7 Comparison table: MAC, DAC, RBAC, ABAC and FIM

Access control mechanism	Accountability	Advantages	Disadvantages	Applications
Mandatory Access Control (MAC)	System	<ul style="list-style-type: none"> Scalable Secure Sensitivity labels Full control by admins only 	<ul style="list-style-type: none"> Implementation is hard Relies on system to control access Not flexible Limited number of users 	<ul style="list-style-type: none"> Government Military Critical missions
Discretionary Access Control (DAC)	Data Owner	<ul style="list-style-type: none"> Easy implementation Highly flexible Protect users from unauthorized data 	<ul style="list-style-type: none"> Relies on object owner to control access Not scalable Prone to mistakes Susceptibility to Trojan Horse attacks Difficulty in system maintenance and verification 	<ul style="list-style-type: none"> Data-based Web applications OS: UNIX, LINUX, Netware
Role Based Access Control (RBAC)	Roles in the system	<ul style="list-style-type: none"> Easy implementation Hierarchy and rights inheritance Separation of duties Scalable High level of security (higher than MAC and DAC) 	<ul style="list-style-type: none"> Coarse-grained Static in using contextual information Ignores resource meta-data Hard to manage and maintain Cannot cater to dynamic segregation-of-duty. Hard to implement fine grained access control Access reviews are painful, error-prone and lengthy 	<ul style="list-style-type: none"> Medical organizations Academic institutions Banking systems
Attribute Based Access Control (ABAC)	Attributes	<ul style="list-style-type: none"> Single Point Provisioning of Users Dynamic Access Control Finer Grained Access Control New and Emerging Technology Sensitive Federal Environments 	<ul style="list-style-type: none"> Untraditional way of users' authorization Lack of requirements Complex ownership of authorization Foundational Models Emulating and Representing Traditional Models Auditability is complex and lengthy Separation of Duties Delegation is complex Attribute Storage and Sharing Hard in scalability Administration and User Comprehension 	<ul style="list-style-type: none"> Government organizations Health Care Systems Airlines companies Insurance systems Telecommunications Carriers
Federated Identity Management (FIM)	User Identity	<ul style="list-style-type: none"> Reduced security risks Increased organizational productivity Delegated administration and self-services Easy auditability Increased organizational productivity Delegated administration and self-services Easy auditability 	<ul style="list-style-type: none"> Trust management Continuous trust monitoring Trust establishment User privacy Reliable access rights across CoTs Adaptation to unexpected changes 	<ul style="list-style-type: none"> Web applications SSO services Banking solutions Cloud applications Mobile applications

CHAPTER 5. Conclusion

5.1 Summary of main results

Based on section 4.3, we can say that only FIM has the highest credits to be used as access control mechanisms inside our BBCCFM. Gathering all of the advantages and drawbacks of FIM, we can consider FIM as main role model to be integrated with BBCCFM. Yet, many of FIM drawbacks discourage us from implementing FIM without designing and enhancing few technical issues.

By theory, we proposed a new model for FIM where it addresses all the drawbacks mentioned above. Our new model is composed of the main FIM standard model and extended modules used for:

Module 1: Trust management and establishment.

Module 2: Continuous trust monitoring.

Module 3: Reliability access rights across circle of trusts.

Module 4: Adaptation to unexpected changes.

5.2 Main contribution of the thesis

As part of the Broker-Based Cross-Cloud Federation Manager development and enhancement plan, this thesis proposed, discussed and proved a new FIM-extended technique to manage and control access and policies in cloud federation.

We have proven that old and traditional access control are not suitable in our case, as well as ABAC which is considered one of the most advanced and well-built access control in the world of cloud computing.

In brief, the below points were the main contributions inside our thesis:

- Proving that RBAC and ABAC cannot be deployed in cloud federation architecture.
- FIM is the best solution for IAM inside cloud federation
- FIM needs extra modification to cover all federation needs
- TEMCM was proposed and implemented to enhance FIM's main functionality.

5.3 Possible extension of future work

Future work might contain the below uncovered modules:

Module 3: Reliability access rights across circle of trusts.

Module 4: Adaptation to unexpected changes.

Reliability access rights across circle of trusts and adaptation to unexpected changes have large impact on any access control systems and especially on federated clouds.

TEMCM might as well be enhanced to allow more flexible and dynamic monitoring of trust between SPs and IdPs. The current version is still beta but future work might enhance the underlying architecture.

Bibliography

- [1] Jacques Bou Abdo, Jacques Demerjian, Hakima Chaouchi, Kabalan Barbar and Guy Pujolle, "Broker-based Cross-Cloud Federation Manager", 8th International Conference for Internet Technology and Secured Transactions (ICITST-2013), Year: 2013, Pages: 244 - 251
- [2] Joel Gibson, Robin Rondeau, Darren Eveleigh and Qing Tan, "Benefits and challenges of three cloud computing service models", 2012 Fourth International Conference on Computational Aspects of Social Networks (CASoN), Year: 2012, Pages: 198 - 205
- [3] Antonio Celesti, Francesco Tusa, Massimo Villari and Antonio Puliafito, "An Approach to Enable Cloud Service Providers to Arrange IaaS, PaaS, and SaaS Using External Virtualization Infrastructures", 2011 IEEE World Congress on Services, Year: 2011, Pages: 607 - 611
- [4] Wided Mathlouthi and Narjès Bellamine Ben Saoud, "2017 IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud)", Year: 2017, Pages: 358 - 365
- [5] Bojan Suzic and Andreas Reiter, "Towards Secure Collaboration in Federated Cloud Environments", 2016 11th International Conference on Availability, Reliability and Security (ARES), Year: 2016, Pages: 750 - 759
- [6] Thamarai Selvi, Somasundaram Kannan Govindarajan, M. R. Rajagopalan and S. Madhusudhana Rao, "A Broker Based Architecture for Adaptive Load Balancing and Elastic Resource Provisioning and Deprovisioning in Multi-tenant Based Cloud Environments", Part of the Advances in Intelligent Systems and Computing book series (AISC, volume 174)
- [7] Pamela Fong, "Asynchronous processing in WebSphere Process Server", IBM developerWorks, Year: 2009, https://www.ibm.com/developerworks/websphere/library/techarticles/0904_fong/0904_fong.html
- [8] Maciej Sztukiewicz, "What is a cloud broker?", IBM developerWorks, Year: 2013 <https://www.ibm.com/blogs/cloud-computing/2013/01/cloud-broker/>
- [9] Antonio Celesti, Francesco Tusa, Massimo Villari and Antonio Puliafito, "Three-Phase Cross-Cloud Federation Model: The Cloud SSO Authentication", 2010 Second International Conference on Advances in Future Internet, Year: 2010, Pages: 94 - 101
- [10] Alan Zeichick, "Understanding cloud-based firewalls", Hewlett Packard Enterprise, Year: 2017, <https://www.hpe.com/us/en/insights/articles/understanding-cloud-based-firewalls-1702.html>
- [11] <https://www.patecco.com/blog/why-iam-components-are-critical-for-managing-user-identities>
- [12] Fred B. Schneide, "Mandatory Access Control", Cornell Computer Science, Year: 2014, <https://www.cs.cornell.edu/fbs/publications/chptr.MAC.pdf>
- [13] z/OS Security Server RACF Security Administrator's Guide SA23-2289-00, "Discretionary access control (DAC)", Year: 2014 https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.1.0/com.ibm.zos.v2r1.icha700/icha700_Disc_retionary_access_control__DAC_.htm
- [14] David F. Ferraiolo and D. Richard Kuhn, "Role-Based Access Controls", 15th National Computer Security Conference, Baltimore, MD. October 13-16, Year: 1992, <https://arxiv.org/abs/0903.2171>
- [15] COMPUTER SECURITY RESOURCE CENTER NIST, "Attribute Based Access Control", Year: 2014, <https://csrc.nist.gov/Projects/Attribute-Based-Access-Control>
- [16] Vincent Hu, David Ferraiolo, Richard Kuhn, Adam Schnitzer, Kenneth Sandlin, Robert Miller, Karen Scarfone, "Guide to Attribute Based Access (ABAC) Definition and Considerations", Year: 2014 <https://csrc.nist.gov/publications/detail/sp/800-162/final>
- [17] Srijiith Nair, "XACML Reference Architecture", Year: 2013 <https://www.axiomatics.com/blog/xacml-reference-architecture/>
- [18] Eleanor Birrell and Fred B. Schneider, "Federated Identity Management Systems: A Privacy-Based Characterization", Year: 2013 <https://www.cs.cornell.edu/fbs/publications/idMgmt.SP.pdf>
- [19] Chen, Jianyong, Guihua Wu and Zhen Ji, "Secure interoperation of identity managements among different circles of trust", Standards & Interfaces, Year: 2011, Pages: 533-540
- [20] Jiang, Jian et al, "A federated identity management system with centralized trust and unified single sign-on", Communications and Networking in China (CHINACOM), 2011 6th International ICST, Conference on. IEEE, Year: 2011

- [21] Bhonsle, Makarand V., Nayot Poolsappasit and Sanjay Kumar Madria, "ETIS--Efficient Trust and Identity Management System for Federated Service Providers", *Advanced Information Networking and Applications (AINA)*, 2013 IEEE 27th International Conference, Year: 2013
- [22] Marmol, Felix Gomez, Joao Girao and Gregorio Martinez Perez, "TRIMS: a privacy-aware trust and reputation model for identity management systems", *Computer Networks* 54.16, Year: 2010
- [23] Kanwal, Ayesha, Rahat Masood and Muhammad Awais Shibli, "Evaluation and establishment of trust in cloud federation", *Proceedings of the 8th International Conference on Ubiquitous Information Management and Communication*, ACM, Year: 2014
- [24] John Hughes, Atos Origin and Eve Maler, "Security Assertion Markup Language (SAML) 2.0 Technical Overview", Sun Microsystems, Year: 2005
- [25] Nitin Naik and Paul Jenkins, "Securing digital identities in the cloud by selecting an opposite Federated Identity Management from SAML, OAuth and OpenID Connect", 2017 11th International Conference on Research Challenges in Information Science (RCIS), Year: 2017
- [26] Wenrong Zeng, Yuhao Yang and Bo Luo, "Content-Based Access Control: Use data content to assist access control for large-scale content-centric databases", 2014 IEEE International Conference on Big Data, Year: 2014
- [27] Faraz Fatemi Moghaddam, Philipp Wieder and Ramin Yahyapour, "An effective user revocation for policy-based access control schema in clouds", 2017 IEEE 6th International Conference on Cloud Networking (CloudNet), Year: 2017
- [28] [29] Ryan Ausanka-Cruces, "Methods for Access Control: Advances and Limitations", Year: 2016
- [30] Hazen A. Weber, "Role-Based Access Control: The NIST Solution", Security concept of Role-Based Access Control (RBAC) as proposed by the National Institute of Standards and Technology (NIST), Year: 2013
- [31] Xin Jin, Ram Krishnan and Ravi Sandhu, "A Unified Attribute-Based Access Control Model Covering DAC, MAC and RBAC", In *Data and Applications Security and Privacy XXVI*, Springer Pages: 41–55, Year: 2012
- [32] Daniel Servos, Sylvia L Osborn: "HGABAC: Towards a Formal Model of Hierarchical Attribute-Based Access Control", 7th International Symposium on Foundations and Practice of Security (FPS'2014), Springer, 187–204, Year: 2014
- [33] Guoping Zhang, Jing Liu and Jianbo Liu, "Protecting Sensitive Attributes in Attribute Based Access Control", *International Conference on Service-Oriented Computing (ICSOC)*, Springer, Pages: 294–305, Year: 2013
- [34] Guojun Wang, Qin Liu and Jie Wu, "Hierarchical Attribute-Based Encryption for Fine-Grained Access Control in Cloud Storage Services", *Proceedings of the 17th ACM Conference on Computer and Communications Security*, ACM 735–737, Year: 2010
- [35] David Ferraiolo, "Towards an ABAC Family of Models", National Institute of Standards and Technology (NIST), Year: 2013
- [36] E. Yuan and J. Tong, "Attributed based access control (ABAC) for Web services", *IEEE International Conference on Web Services (ICWS'05)*, Year: 2005
- [37] Xin Jin, Ram Krishnan and Ravi Sandhu, "A Unified Attribute-Based Access Control Model Covering DAC, MAC and RBAC", In *Data and Applications Security and Privacy XXVI*, Springer Pages: 41–55, Year: 2012
- [38] Xinwen Zhang, Yingjiu Li and Divya Nalla, "An Attribute-Based Access Matrix Model", *Proceedings of the 2005 ACM Symposium on Applied Computing ACM 59*, Page: 363, Year: 2005
- [39] Hadiseh Seyyed Alipour and Mehdi Sabbari, "Definition of Action and Attribute Based Access Control Rules for Web Services", *Proceedings of the 2012 International Conference on Industrial Engineering and Operations Management*, Year: 2012
- [40] Khalid Zaman Bijon, Ram Krishnan and Ravi Sandhu, "Constraints Specification in Attribute Based Access Control", Year: 2013
- [41] Ezedin Barka and Ravi Sandhu, "Framework for role-based delegation models", *Computer Security Applications ACSAC'00*, 16th Annual Conference, IEEE, Year: 2000
- [42] He Wang and Sylvia L Osborn, "Static and Dynamic Delegation in the Role Graph Model", *Transactions on Knowledge and Data Engineering*, Year: 2011
- [43] Xinwen Zhang, Sejong Oh and Ravi Sandhu: "PBDM: a Flexible Delegation Model in RBAC", In *Proceedings of the Eighth ACM Symposium on Access Control Models and Technologies*, ACM, Pages: 149–157, Year: 2003
- [44] He Wang and Sylvia L Osborn, "Delegation in the Role Graph Model", In *Proceedings of the Eleventh ACM Symposium on Access Control Models and Technologies*, ACM, Pages: 91–100, Year: 2006
- [45] Brent Waters, "Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization", In *International Workshop on Public Key Cryptography*, Springer, 53–70, Year: 2011
- [46] Daniel Servos, Sabah Mohammed, Jinan Fiaidhi and Tai hoon Kim, "Extensions to Ciphertext-Policy Attribute-Based Encryption to Support Distributed Environments", *International Journal of Computer Applications in Technology* 47, 2 215–226, Year: 2013

- [47] S. Farrell and R. Housley, "An Internet Attribute Certificate Profile for Authorization", RFC 3281, RFC Editor: <https://www.ietf.org/rfc/rfc3281.txt>, Year: 2002
- [48] S. Farrell and R. Housley and S. Turner, "An Internet Attribute Certificate Profile for authorization", RFC 5755, RFC Editor: <https://tools.ietf.org/html/rfc5755>, Year: 2010
- [49] Basit Shafiq, Elisa Bertino and Arif Ghafoor, "Access Control Management in a Distributed Environment Supporting Dynamic Collaboration", In Proceedings of the 2005 Workshop on Digital Identity Management, ACM 104–112, Year: 2005
- [50] Jaewon Lee, Heeyoul Kim and Joon Sung Hong, "An Attribute Aggregation Architecture with Trust-Based Evaluation for Access Control", In NOMS 2008-IEEE Network Operations and Management Symposium, 1011–1014, Year: 2008
- [51] Vincent C Hu, David Ferraiolo, Rick Kuhn, Arthur R Friedman, Alan J Lang, Margaret M Cogdell, Adam Schnitzer, Kenneth Sandlin, Robert Mille and Karen Scarfone, "Guide to Attribute Based Access Control (ABAC) Definition and Considerations (Draft)", NIST Special Publication 800, Page: 163, Year: 2013
- [52] Jan Camenisch, Sebastian Modersheim, Gregory Neven, Franz-Stefan Preiss and Dieter Sommer, "A Card Requirements Language Enabling Privacy-Preserving Access Control", In Proceedings of the 15th ACM Symposium on Access Control Models and Technologies, ACM Pages: 119–128, Year: 2010
- [53] Claudio Agostino Ardagna, Sabrina De Capitani di Vimercati, Gregory Neven, Stefano Paraboschi, F-S Preiss, Pierangela Samarati and Mario Verdicchio, "Enabling Privacy-Preserving Credential-Based Access Control with XACML and SAML", 2010 IEEE 10th International Conference on Computer and Information Technology (CIT), IEEE, 1090–1095, Year: 2010
- [54] Ali Esmaeeli, Hamid Reza Shahriari, "Privacy Protection of Grid Service Requesters through Distributed Attribute Based Access Control Model", In Proceedings of the 5th International Conference on Advances in Grid and Pervasive Computing, Springer 573–582, Year: 2010
- [55] Guoping Zhang, Jing Liu and Jianbo Liu, "Protecting Sensitive Attributes in Attribute Based Access Control", In International Conference on Service-Oriented Computing (ICSOC), Springer 294–305, Year: 2013
- [56] Vincent C Hu, David Ferraiolo, Rick Kuhn, Arthur R Friedman, Alan J Lang, Margaret M Cogdell, Adam Schnitzer, Kenneth Sandlin, Robert Miller, and Karen Scarfone, "Guide to Attribute Based Access Control (ABAC) Definition and Considerations (Draft)", NIST Special Publication 800 162, Year: 2013
- [57] Adam J Lee, Marianne Winslett, "Open Problems for Usable and Secure Open Systems", In Workshop on Usability Research Challenges for Cyber infrastructure and Tools Held in Conjunction With ACM CHI, Year: 2006
- [58] Danfeng Yao, Michael Shin, Roberto Tamassia and William H Winsborough, "Visualization of Automated Trust Negotiation", In IEEE Workshop on Visualization for Computer Security (VizSEC 05), IEEE 65–74, Year: 2005
- [59] Bernard Stepien, Amy Felty and Stan Matwin, "Challenges of Composing XACML Policies", 2014 Ninth International Conference on Availability Reliability and Security, Year: 2014
- [60] 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, ACM 45–55, Year: 2004
- [61] Xinwen Zhang, Yingjiu Li and Divya Nalla, "An Attribute-Based Access Matrix Model", Proceedings of the 2005 ACM Symposium on Applied Computing ACM 59–363, Year: 2005
- [62] Carlos E Rubio-Medrano, Clinton D'Souza and Gail-Joon Ahn, "Supporting Secure Collaborations With Attribute-Based Access Control", In 2013 9th International Conference Conference on Collaborative Computing: Networking, Applications and Worksharing (Collaboratecom), IEEE, 525–530, Year: 2013
- [63] David Ferraiolo, Vijayalakshmi Atluri and Serban Gavril, "The Policy Machine: A Novel Architecture and Framework for Access Control Policy Specification and Enforcement", Journal of Systems Architecture 57, Year: 2011
- [64] E. Yuan and J. Tong, "Attributed based access control (ABAC) for Web services", IEEE International Conference on Web Services (ICWS'05), Year: 2005
- [65] Hai-bo Shen and Fan Hong, "An Attribute-Based Access Control Model for Web Services", 2006 Seventh International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT'06), IEEE, Year: 2006
- [66] Jian Shu Lianghong, Shi Bing Xia and Linlan Liu, "Study on Action and Attribute-Based Access Control Model for Web Services", 2009 Second International Symposium on Information Science and Engineering, 213–216, Year: 2009
- [67] Haibo Shen, "A Semantic-Aware Attribute-Based Access Control Model for Web Services", In Proceedings of the 9th International Conference on Algorithms and Architectures for Parallel Processing, Springer, 693–703, Year: 2009

- [68] Florian Kerschbaum, "An Access Control Model for Mobile Physical Objects", In Proceedings of the 15th ACM Symposium on Access Control Models and Technologies, ACM, 193–202, Year: 2010
- [69] Bo Lang, Hangyu Li and Wenting Ni, "Attribute-Based Access Control for Layered Grid Resources", In Communication and Networking, Springer, Year: 2010
- [70] Daniel J Buehrer and Chun-Yao Wang, "CA-ABAC: Class Algebra Attribute-Based Access Control", In Proceedings of the 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology-Volume 03, IEEE Computer Society, 220–225, Year: 2012
- [71] Feng Liang, Haoming Guo, Shengwei Yi and Shilong Ma, "A Multiple-Policy Supported Attribute-Based Access Control Architecture within Large-Scale Device Collaboration Systems", Journal of Networks 7 2012, Year: 2012
- [72] Jian Shu Lianghong, Shi Bing Xia and Linlan Liu, "Study on Action and Attribute-Based Access Control Model for Web Services", 2009 Second International Symposium on Information Science and Engineering, 213–216, Year: 2009
- [73] Mike Burmester, Emmanouil Magkos and Vassilis Chrissikopoulos, "T-ABAC: An Attribute-Based Access Control Model for Real-Time Availability in Highly Dynamic Systems", 2013 IEEE Symposium on Computers and Communications (ISCC), IEEE 000143–000148, Year: 2013
- [74] Waleed W Smari, Patrice Clemente and Jean-Francois Lalande, "An Extended Attribute Based Access Control Model With Trust and Privacy: Application to a Collaborative Crisis Management System", Future Generation Computer Systems 31 (2014), 147–168, Year: 2014
- [75] Yongsheng S Zhang, Mingfeng F Wu, Lei Wu and Yuanyuan Y Li, "Attribute-Based Access Control Security Model in Service-Oriented Computing", In Proceedings of the 2012 International Conference on Cybernetics and Informatics, Springer, 1473–1479, Year: 2014
- [76] Bhonsle, Makarand V., Nayot Poolsappasit and Sanjay Kumar Madria, "ETIS--Efficient Trust and Identity Management System for Federated Service Providers", Advanced Information Networking and Applications (AINA), 2013 IEEE 27th International Conference, Year: 2013
- [77] Alguliev R.M. and Abdullayeva, "Identity management based security architecture of cloud computing on multi-agent systems.", Innovative Computing Technology (INTECH) IEEE, Pages: 123-126, Year: 2013
- [78] Chen, Jianyong, Guihua Wu and Zhen Ji: "Secure interoperation of identity managements among different circles of trust", Computer Standards & Interfaces 33.6, Pages: 533-540, Year: 2011
- [79] Jiang, Jian et al, "A federated identity management system with centralized trust and unified single sign-on", Communications and Networking in China (CHINACOM), 2011 6th International ICST Conference on IEEE, Year: 2011
- [80] Chadwick, David W., George Inman and Paul Coxwell, "CardSpace in the Cloud", Proceedings of the 17th ACM conference on Computer and communications security, ACM, Year: 2010
- [81] Khattak, Zubair Ahmad, Suziah Sulaiman and J. A. Manan, "A study on threat model for federated identities in federated identity management system", Information Technology (ITSim), 2010 International Symposium in. Vol. 2, IEEE, Year: 2010
- [82] Chadwick, David W. and George Inman, "The Trusted Attribute Aggregation Service (TAAS)", Availability Reliability and Security (ARES), 2013 Eight International Conference on. IEEE, Year: 2013
- [83] Sam linson, "User-Centric Trust based Identity as a Service for federated Cloud Environment.", Computing Communications and Networking Technologies (ICCCNT), 2013 Fourth International Conference on, IEEE, Year: 2013.
- [84] Gao, Hao, Jun Yan, and Yi Mu, "Dynamic trust model for federated identity management", Network and System Security (NSS), 2010 4th International Conference on IEEE, Year: 2010.
- [85] Marmol, Felix Gomez, Joao Girao and Gregorio Martinez Perez, "TRIMS: a privacy-aware trust and reputation model for identity management systems." Computer Networks 54.16 (2010) 2899-2912, Year: 2010
- [86] Kanwal, Ayesha, Rahat Masood and Muhammad Awais Shibli, "Evaluation and establishment of trust in cloud federation", Proceedings of the 8th International Conference on Ubiquitous Information Management and Communication, ACM, Year: 2014.