****

**University of Western Sydney**

June 24,2015

Secure Access Control Framework for Big Data

**Confirmation of Candidate**

**Mohammed Al-Zobbi**

**Principal Supervisor: Dr. Seyed Shahrestani**

**Co-Supervisor: Dr. Chun Ruan**

Contents

[Abstract 3](#_Toc423040846)

[1 Introduction 3](#_Toc423040847)

[2 Access Control Models 5](#_Toc423040848)

[2.1 MAC Model and Lattice 6](#_Toc423040849)

[2.2 Role-Based Access Control Model (RBAC) 7](#_Toc423040850)

[2.2.1 Commercial products examples on RBAC 8](#_Toc423040851)

[2.3 Current RBAC Concerns 12](#_Toc423040852)

[2.3.1 Management Complications in RBAC 13](#_Toc423040853)

[2.3.2 Big Data Analytics and Concerns 15](#_Toc423040854)

[2.4 Attribute Based Access Control Model 17](#_Toc423040855)

[2.5 Level of Assurance (LoA) 18](#_Toc423040856)

[3 Data Analytics 19](#_Toc423040857)

[3.1 Security Issues in Hadoop 21](#_Toc423040858)

[4 Research Questions 22](#_Toc423040859)

[5 Research Boundaries and Scopes 23](#_Toc423040860)

[6 Research Method 24](#_Toc423040861)

[6.1 Stage One: 24](#_Toc423040862)

[6.2 Stage Two: 24](#_Toc423040863)

[6.3 Stage Three 27](#_Toc423040864)

[7 Research Outcome 29](#_Toc423040865)

[7.1 Access Control Framework for Big Data 30](#_Toc423040866)

[7.2 DBMS and Access Model 30](#_Toc423040867)

[7.3 Attribute Value Access Control Model 32](#_Toc423040868)

[7.3.1 General Definitions 33](#_Toc423040869)

[7.3.2 *Attribute Value-Based Access Control (AVBAC)* 37](#_Toc423040870)

[7.3.3 Comparing this Framework with the Conventional Frameworks 47](#_Toc423040871)

[8 Budget Estimation during the PhD study 49](#_Toc423040872)

[9 Research Time-Line 50](#_Toc423040873)

[10 Publication Plan 52](#_Toc423040874)

[11 Conclusion 53](#_Toc423040875)

[12 References 54](#_Toc423040876)

[13 Appendix A (List of Abbreviations and Acronyms) 55](#_Toc423040877)

# Abstract

*The Big Data concept constitutes a need for a secure access control model. This CoC presents the Big Data security concerns. Managing the access to a large size of data, with multi-domain users is intractable. Data may be analyzed to reveal pattern trends and association. Securing data access over the cloud is essential, to overcome the Big Data ubiquitous use, and intensive access. Role-Based Access Control Model (RBAC) is inefficient in Big Data access management, due to it is implementation complexity, and security decrepitude. Attribute-Based Access Control Model (ABAC) is still under development and is still an active area of research. An Attribute Value-Based Access Control (AVBAC) framework is proposed to resolve the big data concerns on using either RBAC or ABAC. The non-conventional framework assigns numerical values to data attributes. The attribute values are pre-defined for each object, to control user’s access privileges. Each role is assigned a numerical access value. User’s access is then determined using this value. That is, if the access value matches the attribute, access to the object is permitted. For this, an experimental methodology that evaluates the Attribute-Based Value framework is proposed. A tentative experiment starts with a small scale example. The experiment will gradually raise the scale latitude to cover more access control scenarios. The achievement will thoroughly participate in finding the proper framework. That is able to mitigate the Big Data complexity and security breach in system analytics.*

# Introduction

Big Data is the future trend of virtual environments. Human create an average of 4 Zettabyte (1,000,000 Petabyte) of data worldwide. The term big data refers to the massive amount of digital information [[1](#_ENREF_1)]. Many factors help data fast growth, such as; Internet speed, cheap storage, mobiles, tablets, cameras, cloud technology, and social websites.

Big Data can be distinguished from ordinary data by two factors; these are Online Transaction Processing scaling or called OLTP scaling, and Online Analytical Processing or called OLAP. These two differences urged the network engineers to search for alternative technical solutions to handle both OLTP and OLAP in a better performance. Conventional database stores structured data with hundreds or even few thousands of terabytes, scaling up database with SQL database packages like Oracle, Sybase, and DB2 is possible to a certain size. However, this classical database is relatively slow in Big Data, Big Data performs better with NoSQL solution, or not only SQL, which is a combination of SQL and Object Orient implementation [[2](#_ENREF_2)].

Prior to Big Data concerns, it is worth mentioning that Big Data nature is public, in the sense that organizations claim its complete ownership not a straight-forward matter. Individuals, partners, strategic partners, contractors, clients, and co-workers access their data from everywhere at any time. Organizations may delegate the access control to other agencies. Personal information is commonly shared between organizations. This prevalence demand, eventually, causes an increasing number of Big Data access, from everywhere [[3](#_ENREF_3)].

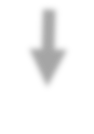
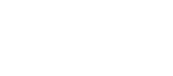
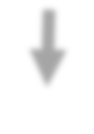
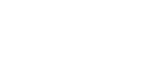
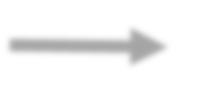
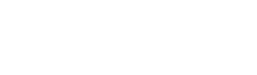
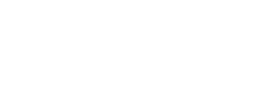
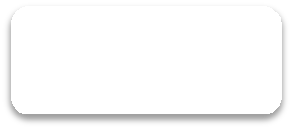
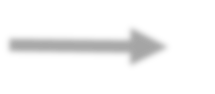
Data analytics is the key to examine raw data with a purpose of drawing refined data for decision makers. Some analytics practitioners transfer Big Data into an internal private environment and then transfer data back to the public environment, once the analytics is completed [[4](#_ENREF_4)]. However, this solution does not provide accurate results for Big Data with a massive size of dynamic data changes.

One of the security concerns in Big Data is the complexity of the current access frameworks. The increasing number of multi-domain users may add a cumbersome to system administrators on managing user’s access and permissions. Also, delegating other companies may participate in a security breach, since the current access model does not provide any security enforcement for delegators. Big Data tools are different from the ordinary tools that are being used in the market for years. Database tools, such as NoSQL, have changed, File systems tools, such as HDFS, have also changed, storage techniques, analysis techniques, even application and coding techniques have changed. Many changes have occurred to compromise the new Big Data approach, with large scales of OLAP and OLTP.

# Access Control Models

Current Access Control frameworks authorize and authenticate users over the network. Users are first authenticated before being given certain permissions for specific network devices and services. The permissions are based on the rules and procedures of each organization. Access Control frameworks are divided into three different types: Mandatory Access Control (MAC), Discretionary Access Control (DAC) and Role-Based Access Model (RBAC). DAC security is frail, hard to manage and control over the cloud. MAC provides a high-security access, for a small-scale size of data. MAC cannot be applied to Big Data [[5](#_ENREF_5)]. RBAC represents the entire structures of the organizations that are between MAC and DAC. However, it is essential to examine RBAC in Big Data environment.

Most Access Control Models use the principle of object and subject or a combination of both of them. The subject is usually a user or a program that wishes to access an object. A “reference monitor” controls the relationship between “subject” and “object”, by reading from Access Control Database. Figure 1 illustrates this model, any access or access attempts are recorded in the log file or audit file. The “reference monitor” must be implemented correctly, tamperproof, and be able to force the access policy [12].



Access Control Database

Subject

Reference

Monitor

Object

Audit file

Figure 1. Access Control Model components, and relationship between the subject and object.

## MAC Model and Lattice

Early MAC model used the mathematical Lattice model and was developed by Bell and Lapadula in beginning of the 1970s, for the US Defence Department and Aliases. The model describes the “No Read Up” and “No Write Down” based on the level of confidentiality, known as (Top Secret, Secret, Confidential, and Unclassified). For instance unclassified users cannot read data at a confidential level (No read up) and Secret data cannot be written into the unclassified file level (No Write Down) [[6](#_ENREF_6)].

Lattice uses the terms subject and object; the subject may be a user while the object is a file, a computer or an application. All subjects are objects, but not all objects are subjects. The values of the access table are presented by [Classification, Categories] where each object contains a current and a maximum security level, which may change dynamically. One level of security may dominate other levels. For instance, if the classification of A > classification of B, then security level A dominates security level B. Any file is presented by [subject,object] in a fine-grained format. Models with fine-grained access tables are an example of Discretionary Access Model DAC. Therefore, Bell-Lapadula represents both MAC and DAC [[7](#_ENREF_7)].

Bell-Lapadula model is technically difficult to enforce or apply in Big Data; this is because the non-classified subject cannot access any object for read or write. Hence, all objects and subjects must be pre-classified. The classification must include the network and the processing nodes. For instance, if data analytics process attempts to run a top-secret data while the computer node was classified as confidential; then the computer will not be able to read this data. Moreover, MAC access is totally controlled by system administrators, and object owners are not permitted to add any security privileges. This feature creates a cumbersome overhead management on administrators, and difficulties in implementing it in Big Data.

For the two previously mentioned obstacles; it is hard to implement such a framework in an enterprise environment. Big Data denotes a large number of processing machines, users, data, and applications.

## Role-Based Access Control Model (RBAC)

In 1992, Ferraiolo and Kuhn proposed a general RBAC model that is based on assigning roles to users. Subjects are assigned some role while ensuring that roles are allowed to access only the authorized data. This explains the current RBAC model since each subject (user) may belong to one or many roles. Only one active role at a time may be authorized to perform one or more transaction (s) [[8](#_ENREF_8)].

RBAC supports both MAC and DAC models. Many RBAC frameworks were developed. However, the core RBAC remains the same for most developed frameworks. RBAC structure consists of four main components; Static constrained, dynamic constrained, core, and hierarchical. The basic core of RBAC is the many-to-many relationship presented between users and groups. Groups represent the role in an organization like administrator role, engineer, designer, nurse, doctors and alike. Each role may have many users, and one user may belong to many groups or roles. As a result the core of RBAC can be defined by five administrative elements; users, roles, permissions, and permissions are operations applied to objects. Most operating systems assign access policies and group to ACL [[9](#_ENREF_9)].

In RBAC model, an administrator initiates the delegation to other users. An Administrator or a System Security Officer (SSO) is the first role that can delegate. The SSO grants the privileges for delegations [[10](#_ENREF_10)]. Other users must be given a role within the organization. Entity Relationships present many-to-many relationships between entities in RBAC, this can be confined by normalization. As shown in figure 2, normalization places entity “Subject” between “Users” and “Roles” to resolve the (n-n) relationships. One active operation (op) is permitted per object (o), and one role deals with one active operation.

Users u1,u2,u3,….

Roles r1,r2,r3,….

Subject

Perm. 1

Perm. 2

O1

Op1

Op2

O2

Figure 2. Resolving the many-to-many relationships in RBAC core.

### Commercial products examples on RBAC

Most cloud applications run in multi-tenant mode, one software can be used by many organizations. In cloud application software, three programming methods are used to implement the security access controls; access control using bit masking, access control using logic tier processes, or access control using URL locations. Most Cloud applications use web-based software programs, which makes the bit masking less desired; especially that bit masking method limits the permissions up to 32 or 64.

The bit masking uses the zero and one bits to assign permissions for objects, it reads permissions from the ACL, and stores this information in a 32-bit format. Usually, the first 16 bits determine the special rights; such as (view directory, change ownership). While the bits from 16 to 23 determine the standard rights; which are (read, write, delete, execute). The rest of the bits are used for generic execute, write and read. Programmers need to be aware of these format on programming permissions [[11](#_ENREF_11)].

The n-tier logical programming is the most use in programming permissions. Developers use their roles on programming cloud applications, instead of using organization’s roles. Each role can run specific actions, the actions and roles are tightly built-in and cannot be modified. Users are forced to use the available programmed roles, instead of using their own. Programmers use object oriented classes to control the objects methods, for instance methods could be (getAllPassengers(), addNewFlight(), suspendBooking(), viewFlightDetails(), assignFlightCost()).

Moreover, procedures and actions assigned to roles may change from one organization to another. Organizations use the workflow to control these procedures, the workflow should be taken into developers considerations. Workflow does not assign work processes and tasks only, but also it controls the access permission for each role.

Access control using URL locations is the easiest and least secure method. This method uses the URL address to control users’ access. Administrators control the roles that are permitted to access these interfaces.

A need for inspecting the current cloud applications is essential, to unveil the current market use of RBAC over the cloud. The inspection should explore the methods of assigning access policies, and the structure of the framework used to sustain the security policy. There are many cloud applications in the market. For this research, several popular products that follow the most common RBAC frameworks are chosen. They include, EmpowerID, and Microsoft Authorization Manager or AzMan.

#### EmpowerID

This application implies a set of services like Identity Management, SSO, Identity Provider and cloud security platform. EmpowerID focuses on Microsoft products Management for large scales and enterprise management. General definitions should be addressed before delving into EmpowerID processes. Firstly, it is essential to distinguish between Resource (known as Resource Type), Resource Role, and Resource Operation. The resource can be any asset type in the cloud such as; software, user account, workflow, web pages or Windows shared folder. Each Resource type is a bundle of Resource Operations, and Resource Operations are managed by Resource Role.

Access Level, known as Resource Role, is a bundle of operations (read, write, delete) and native system rights such as; User Account, AD Security Group, and Exchange Mail Box. Both operations and system rights are defined by “Resource Role Definition”. The definition can be assigned to any number of actors. Each Resource type has a set of Resource Role Definition, and each Resource Role Definition includes many Resource Operations, as shown below.

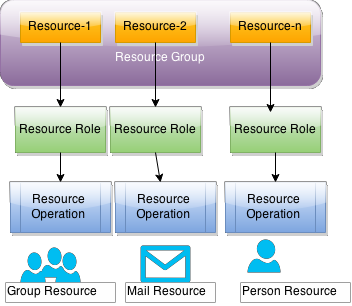


Figure 3. The Resource Role and operations in EmpowerID.

The term “Protected Resources” is defined as resources within a resource system that are managed and protected, such as Groups or Mailbox. The Protected Resources are managed by Protected Operations, such as change email quota, add a member to the group, edit account name and similar. Protected Operations inherit the permissions from RBAC security assignment users or groups.

For example, the Help Desk resource role, for person resource type is responsible for a bundle of protected operations integrated from the organization workflow. Edit Person expiration, Edit Person, attributes, Disable Password, and Remove Person, are examples of protected operations.

Resource Roles (Access Levels) are grouped together to conclude “Management Role,” so that multiple resource access levels can be assigned as a single unit to a person. To create Management Role, a group of assignments is put together to create a template that can be used to create a management role. The Management Role Assignment can be IT Administrator, Group Administrator, IT Help Desk, RBAC Author, and so on. Each object of the Management Role Assignment is defined by a bundle of Resource Role Assignments of different types, such as Workflow type, Pages and Reports type, and User Interface type [8].

#### Microsoft Authorization Manager

This Microsoft native framework uses RBAC. It provides a high-end authorization solution for .NET and COM applications. Authorization Manager (AzMan) is an alternative to custom authorization solutions. Other solutions may tend to be limited in features, poorly integrated with the system, or very expensive to design and maintain.

AzMan uses five main concepts. These are: operations, tasks, role, scope and authorization scope. The operations operate on the lower level like “add\_user, edit\_accoun” and alike, the operations collections make a task. The Role is created, and a group of operations is assigned. The fourth concept is the scope, which is the boundary of objects assigned to roles.

In AzMan, the administrator creates roles, and assigns a set of operations to that role, then uses “MS Active Directory Users and Groups” to assign groups to the created role. For ASP.NET developers, they need to switch AzMan console to “developers” mode and follow three main steps. Firstly, create an “AzMan Policy Store”, the policy store should use the same admin account that manages ASP.NET. Usually, the policy store is an XML stored file. Secondly, define tasks, and operations definitions, then assign these tasks and operations to users and/or groups. Finally, set up the ASP.NET “web.config” file to read from the created xml file.

This structure is suitable for internal organization processes and operations, and best match the single domain software. In cloud applications, implementing such an access framework and structure does not support the multi-domain, and multi-platform applications.

## Current RBAC Concerns

As introduced before, current RBAC and ABAC products provide stable solutions for controlling resource access. However, two main concerns have recently appeared with the cloud world revolution, these two concerns are:

### Management Complications in RBAC

One of the Big Data concerns is the storage up-scale, which requires access control up-scale as well. RBAC was designed to serve a large number of users, and to reduce the overhead management on administration. In reality, a careful study, for the current RBAC use, indicates that RBAC is inefficient in resolving the scale-up issue. Moreover, it does not provide a real solution for management complexity.

The current use of RBAC roles does not reflect the actual organization roles. Instead, it reflects the application roles needed and found by developers. This critical confusion leads to another accumulative problem with the increasing number of roles. Each new application, used by the organization, may add another set of new roles that are not available in the actual organization structure. As shown in figure 4 the original RBAC design and it is a relationship to users and resources, comparing the current RBAC use. It is clear that newer RBAC requires more management than before. RBAC users are confronted with different terminology and implementation.

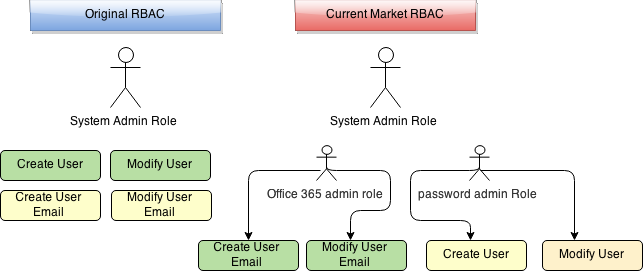


Figure 4. The original RBAC, and the current market RBAC.

This complication creates role duplication RBAC structure. For example ‘Help Desk’ role is usually assigned to any software application, as a built-in role. Therefore, Help Desk role should be identified individually for each application to determine the proper permissions.

Another example, of management complications, is the roles names similarities between different organizations. The multi-domain environment creates parity, which may cause permissions breach. Suppose that an organization A uses a cloud application, as shown in figure 5. Organization A has delegated organizations B and C to use the same application. Imagine that the three organizations contain a role called “teller”; then teller cannot be employed in this application, this is to avoid permissions similarity. Hence, another conventional name may be used like tellerA, tellerB, and tellerC. This looks easy for a limited size of organizations and roles names. One can imagine the management complexity on large scales when hundreds of organizations access the same application with tens or even few hundreds of roles. Moreover, organizations may be overwhelmed by unnecessary number of redundant roles names, which point to the same role. In our example, Organization A may create redundant names like tellerA, teller, and tellerMain.

File Group A

**Organization A App. owner**

Organization B B

Organization C C

Permission A

Permission B

Permission C

File Group B

File Group C

Organization

Organization

Role B

Figure 5. Redundant role names in a multi-tenant cloud application.

The current market applications products create their built-in roles that meet the functions and operations of the application, for instance, Backup role is a built-in role in MS- SQL, while “Modify User Email” is a built-in role in MS Exchange service.

This complexity also leads to programming complexity; developers are supposed to add security to their UI and operations. Creating own security permissions and access models is expensive, hard to manage, and may not be secure enough.

One of the previously mentioned examples is EmpowerID solution, this example shows the current access control management applications with SSO. The authorization is based on assigning operations to resources while the large number of operation for each application makes the management even harder. Also, such solutions do not integrate roles and authorization management in one place. The current available solutions in the market may provide a limited solution over the cloud, like providing federation and SSO. However, these products may increase the management complexity, and makes it even harder to assign the proper permissions for many vendors in a multi-domain environment.

### Big Data Analytics and Concerns

Cloud applications use n-tier architecture, multi-tiers of programming and connections to the database are used. In data analytics, programmers use programmed scripts to process data analytics, scripts are directly executed in database or files. This kind of access will skip the access control tiers, and eliminate any access control process. As a result, data will be exposed to any unauthorized access.

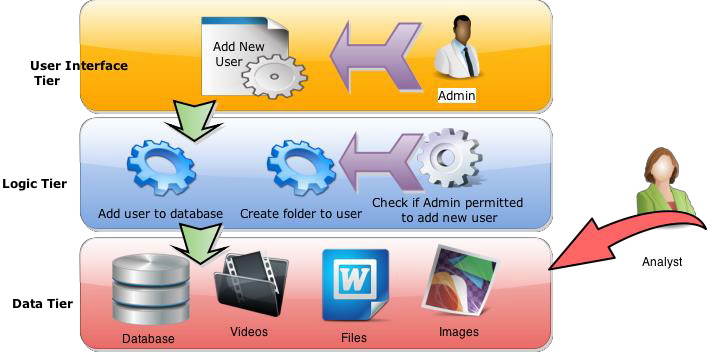


Figure 6. The n-tier permissions architecture and data access using data analytics.

This concern may also appear on using files, images, videos, and audios. Files access can be done using direct access or through applications. Applications use access control to manage access permissions to these files. However, users may skip the n-tier level on a direct access. Hence, they will not be controlled by any control management commands.

Moreover, each cloud application uses its own access control management system. This approach duplicates the management methods between a native belonging management, and any third party management system, that is being used by the organization.

The Data Base Management System (DBMS) may use RBAC or any other access control models. DBMS prevents some access attempts to database system, but this is not always the case. DBMS authorizes database administrators, and developers to access database, using (create, alter,dump,backup and others). Other users are still able to access database by using (select, insert, update, and delete). This principle can be applied when using MapReduce and analytics. Since the “MapReducer” is able to access the database by skipping the DBMS and any n-tier permission level. A need for a security technique that connects the database security to the organization RBAC is essential.

## Attribute Based Access Control Model

ABAC is suggested as an alternative for RBAC, as RBAC is impractical and hard to implement in the multi-domain environment. The model suggests using attributes instead of roles. ABAC model is developed based on RBAC model, such as separating users from permissions. ABAC has the capability of using DAC or MAC, it can be used in the multi-domain environment and the Internet. Enterprise data can benefit greatly from this model. Hence, it can be scaled-up to include Big Data environment. The scale-up can be implemented by increasing the size of subjects that can access a large size of objects, and without assigning access control for each subject and object. Also, ABAC systems can enable Risk-Adaptable Access Control (RAdAC) solutions, with risk values as variable attributes.

A formally defined implementation of ABAC in Big Data is not available. A research was conducted by IBM Centres for Advanced Studies (CAS) presents a framework for ABAC [[12](#_ENREF_12)]. Users can provide the idP with the required attributes and metadata, each attribute contains a level of confidence. The idP must send these attributes to the SP, so the SP must be aware of the current induced attributes on any access request. However, the SP have the right to keep these attributes or reject them, this depends on the trust level between idP and SP. For example, the provider may investigate whether the user age is 18+ or not, then the idP provides the age attribute, and the level of confidence (18) to the SP to decide.

The previous study didn’t show other actions that should be done by both idP and service provider, the study ignored the followings: The study does not show how the attributes are assigned to the object, and how the analytics process is protected and integrated by the user’s authentication data. Also, the study does not distinguish between the data owners and other users and does not take the multi-domain environment into considerations.

* Didn’t show how the security level is added to the data before transferring them to the cloud, for example risk management analysis should be taken into the security level consideration of the transferred data
* Didn’t show how the data owner able to track his/her data and information, and make sure that data is protected and security integrated with his/her policies and rules.

As noticed a need for a more comprehensive study is needed, procedures and details should be clearer to enable a proper system implementation. The framework design should handle the organizational data since data is created whether created in the cloud or outside the cloud, the idP should be given more roles and decisions, to determine the permitted attributes of both object and subject. On the other side, the service provider should be able to understand the enclosed attributes with data and able to enforce any conditions requested by the data owner, not only by the service provider.

## Level of Assurance (LoA)

The National Institute for Standards and Technology (NIST) has provided an identity relevant frameworks for evaluating authentication assurance [[13](#_ENREF_13)], users need to be authenticated to use cloud services such as; government and commercial services. NIST identifies four different levels of access, starting from level 1, which is the lowest, to level 4, which is the highest.

Access levels are determined by the data importance and confidentiality. For example top secret data needs a higher authentication Level of Assurance or (LoA), this can be denoted by LoA4. The higher level of access may promote more complicated technologies on authentication such as; authentication challenges, tokens, mobiles, and biometrics. US GSA has endorsed openID as a valid level 1. Many examples are currently available in the market for LoA1 like Yahoo, Google, Facebook, Twitter, and many others. LoA1 is available for the least important services as it is credential has little value. Therefore, doubts and risks about individuals that cannot be held accountable for the information [[14](#_ENREF_14)].

LoA2 uses a higher security level, more confidential features are added to manage the user verification. Also to the password, a proof of identity is added in this level, a token pin or mobile pin is required on each time access. The user identity represents one of the user attributes such as; his id number, last name, first name, or a combination of these attributes.

LoA3 provides a multi-factor remote network authentication, a verification of identity is required at this level. Level 3 authentication requires cryptographic strength mechanisms that protect the primary authentication token. Secret key, private key, or one-time password, to prevent eavesdropper, replay, on-line guessing, verifier impersonation and man-in-the-middle attacks. The security factors can be hard or soft cryptographic tokens.

LoA4 is similar to LoA3 with extra factors of security; these factors must be hard cryptographic tokens such as smart card, hardware token (commercial examples WAP, RSA, and AES) or biometric [[15](#_ENREF_15)].

# Data Analytics

As mentioned before, the most two concerns in Big Data are OLAP and OLTP. OLTP promotes scaling up data, and OLAP promotes Data analytics. Data Analytics is an important task for organizations. Organizations need to analyze their own data, and other organizations data as well, which helps taking the right decision, to develop their products and services. Business forecasting, investment and finance, healthcare, governments and media need to know public opinions and track fraud, terrorist and criminals.

A popular MapReduce [open-source](http://en.wikipedia.org/wiki/Open-source_software) implementation is [Apache Hadoop](http://en.wikipedia.org/wiki/Apache_Hadoop), figure 7 describes the Hadoop tasks processes. As shown in figure 7, Hadoop has its file system HDFS. Usually both MapReduce and Hadoop File System (HDFS) are tightly integrated and physically collocated within the same server in a cluster, to provide the shortest path between data and processing machines. HDFS keeps records of split data location within the cluster, the data might be structured database or just files. HDFS clusters do not use Redundant Array of Independent Disk or RAID, RAID redundancy performance is slow in Big Data. RAID uses a considerable amount of computation time to split and allocate data, RAID 0 performance is better than RAID 1 and so on [[16](#_ENREF_16)].

Load Balancers

HDFS

MapReduce (hadoop)

Nodes in Cluster 1

….

3

Storages in Cluster 2

Storages in Cluster 1

Cluster 2

Extendable









....

scalable

Figure 7. Big Data process structure, HDFS processes and MapReduce processes

Technically, MapReduce starts by breaking any process into two sub-processes, the map phase and reduce phase. Each phase has key-value pairs as input and output. The programmer who intends to map and reduce the data, has to program the function of both maps and reduce. The idea of MapReduce is using some servers in a distribution manner, then pass certain tasks to each server, so the tasks are completed in a shorter time. Usually, the target data are distributed around many clusters; each cluster includes servers for MapReduce. Each cluster completes the mapping process and transfers results to the home server, which receives many maps results and reduces them into a smaller size [[17](#_ENREF_17)].

Hadoop performs better when running MapReduce on the same local server that resides in HDFS, this is called the “Data Locality Optimization.” The optimal split size is the same as the block size; it is the largest size of input that can be guaranteed to be stored on a single node. If the split spanned two blocks, it would be unlikely that any HDFS node stored both blocks, so some of the split would have to be transferred across the network to the node running the map task. This is less efficient than running the whole map task using local data [[18](#_ENREF_18)].

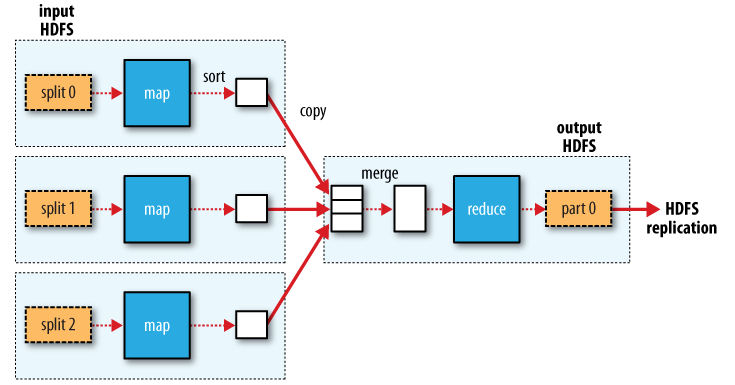


Figure 8. Map and Reduce processes, and HDFS interaction.

The most popular Hadoop package is called “Cloudera,” it is a stable Apache Hadoop with continuous patches, backports and updates. Cloudera contains main Hadoop features like self-heal HDFS, parallel MapReduce, HBase Hadoop database, Hive NoSQL database, Pig dataflow language, Oazie workflow for independent jobs, and other.

## Security Issues in Hadoop

Many companies use Hadoop in data analytics and scaling up Big Data. There is no clear analysis of companies names or numbers that use Hadoop features, but some enterprise companies have announced and advertised Hadoop use, like Yahoo!, Facebook, NetFlex, JBMorgan, Amazon, Microsoft, Adobe, eBay, Hulu, Twitter, and Ning,.

HDFS does not provide any security features, all security features are added to HDFS by third party applications. Also, HDFS can be integrated by using the operating system, for example, Windows users can apply their own ACL on files and folders. Encrypted files can be applied by using other third party plug-in programs. HDFS files can be protected by the UNIX operating system that HDFS servers operate on.

Another security concern with Hadoop was the management access, such as developer’s access, and console access. This access distinguisher is currently not implemented. Moreover, there is no any method of discovering rogue services running in parallel with authorized services. Is there any way to implement Attribute-Based Access Control (ABAC) or Role-Based Access Control (RBAC) is there any method to better control MapReduce process, for example; who can run, edit, stop tasks on the taskTracker. Also, data encryption requires some revive like encryption data in transit and at rest, and keeping a track of data provenance [14].

Obviously, the current available analytics tool for Big Data is Hadoop. The previously mentioned security issues may provide security solutions for Hadoop, but they never mention the data authentication and authorization integrity. The fact that it is hard to stop any rogue processes within Hadoop framework, hence, it is better to develop some techniques that protect data privacy in a hierarchy structure level.

# Research Questions

The following question is raised and derived from the Big Data concerns.

1. How a framework of Access Control Model can enforce the organizational business roles over Big Data, and without affecting the current framework performance. The framework:
   1. Should resolve the data analytics security concern in Big Data.
   2. Should not affect the efficiency of Online Transaction Process scaling (OLTP)
   3. Can resolve the big data complexity concern
   4. Can enforce the organizational policy on delegating access to any external user.
   5. Suitable for multi-tenant and multi-domain environments.
   6. Manage structured or unstructured data copyrights, and legitimate use.
   7. Integrate both RBAC and ABAC within the same framework.

# Research Boundaries and Scopes

Researchers have developed many access models with RBAC core. The most popular model was proposed by Professor Ravi Sandhu named as RBAC96, this model gained popularity by assigning constraints, and developing role hierarchy [[19](#_ENREF_19)].

Nevertheless the popularity of the model, it is essential to experiment the RBAC model in Big Data environment, for finding the best model and practice over the cloud. This includes the following boundaries and scopes of cloud services: multi-tenant software as in SaaS and PaaS, delegating other companies, web services, and files services and data analytics.

This research will focus on finding a conceptual framework for Big Data security protection. The framework should be able to resolve some Big Data concerns about data privacy, integrity, and authority.

The current access control models cannot be applied within a global domain, all control models were designed for on house access methods, i.e. within the borders of the company, or partnership companies. The need for global objects identifiers whether for users or cloud data will manage the access control in a better fit model for Big Data.

# Research Method

To evaluate the proposed framework and identify its gaps for rectifying, a research methodology with three main stages will be incorporated. The first stage will investigate the proposed framework in details, but using small examples. The next two stages will evaluate the framework within a bigger range of technologies and protocols. MATLAB simulator will be employed in certain stages. Figure 9 illustrates the three experimental stages and their objectives.

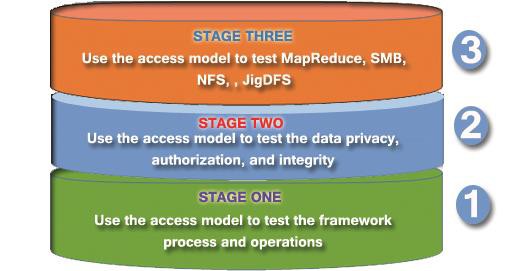


Figure 9. Three experimental stages and the aim for each stage.

## Stage One:

In this stage, a theoretical framework will be introduced. Varieties of use case scenarios will be proposed. A small software application will be coded using php programming language, with MySQL database, this application is essential to amend the proposed framework. Also, this program will be used within the next coming stages, with a continuous development.

## Stage Two:

Stage two will experiment the access control model for both of file system and database system. The experiment will evaluate the data upload and security parameters assigned for the file metadata. Two main assigned points will be configured for the experiment; these are the idP and the SP. The integrated framework will be implemented using Java and PHP for web programming.

SP: provides

* + - Data storage
    - Applications

**idP1: provides**

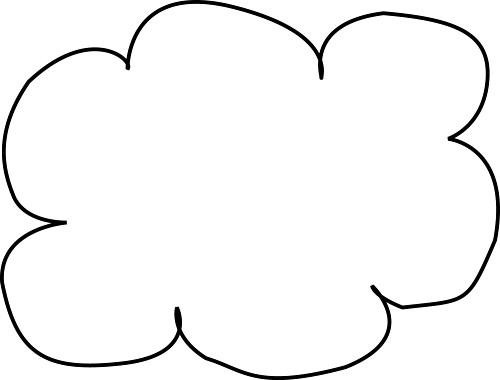
**-File Upload**

**idP2: provides**

**- User Auth.**

Client

Figure 10. The second stage of the experiment, using only three VMs.



In order to simplify the experiment requirements, three virtual machines are enough at this stage. The 3 VM’s with UNIX based Operating systems can be within the same LAN network, private IP addresses may replace the public unique IP addresses.

As shown in Figure 10, two idP servers and one Service Provider are used in this experiment. Both idP servers will be provided by SAML v2.0 standard SSO application, and a popular open source application is called JOSSO (Java Open Single Sign-On project home).

The first two servers act as a federation service and contain the following services:

* Apache Server
* Java Open Single Sign On or (JOSSO)
* Software Application program using Java and PHP, which implements the access model framework
* OpenSSL to encrypt XML and metaData

The third server (SP) is provided by different services, such as:

* NAS services and files sharing. The service should provide different types of files and folders, such as; text, images, audio, video and web pages.
* Apache Server
* Open source CRM application
* Software Application program using Java and PHP.

## Stage Three

The third stage is also experimental. This stage delves in the Big Data analytics, and MapReduce. Hadoop should be installed and configured with its HDFS. Figure 6-3 shows the third stage requirements, it is obvious that the third stage is identical with the second, but extra two SP side servers will be added. SP servers will examine the MapReduce procedures and other.



idP servers: SOAP, developed App.

SP servers: HDFS, MapReduce, NFS



Figure 11. The third stage of the experiment. Using five VM’s.

Moreover, Network File System (NFS) will be tested within the new framework environment. NFS is a consistent application that is used by UNIX to allow users access files across the network. NFS protocol version 4 contains more security features than before. The new framework will be compatible with NFS and does not overlap the access permissions.

Also, JigSaw Distributed File System (JigDFS), will be tested by using three nodes connected in one cluster. JigDFS is a distribution method to protect the high security level of data. Storing expensive data is usually accomplished by splitting data into parts, and then storing these parts in different locations within the cluster. Usually, data is stored in three redundant nodes within the cluster.

The following tools and applications will be installed in the five available servers: The first two servers act as Federation services. Moreover, the following services will be installed:

* Apache Server
* Java Open Single Sign On or (JOSSO)
* Software Application program using Java and PHP, which implements the access model framework
* OpenSSL to encrypt XML and metaData

The three servers are provided by the following services:

* NAS services, and files sharing. The service should provide different types of files and folders, such as; text, images, audio, video and web pages.
* Apache Server
* Open source CRM application
* Software Application program using Java and PHP.
* MapReduce application Cloudera

This Lab. Can be setup at UWS university campus. Alternatively, Microsoft Azure can be used to experiment stage 3. Hadoop (HDInsight) can be used to increase the number of nodes in a cluster. MS Azure provides App Services, Virtual Machines, Mobile Services, Cloud services and data management. More cluster nodes will provide a better outcome and more accurate results.

MS Azure prices are flexible and “As you go”, Researchers may add and /or omit any service after the completion of the experiment. The pricing methods is calculated per minute use, for example; if 6 hadoop nodes per cluster were used, as shown in figure 12 and the expected experiment period was around 5 days; then a total cost of (24\*5\*1.938 = $ 232.6) will occur.

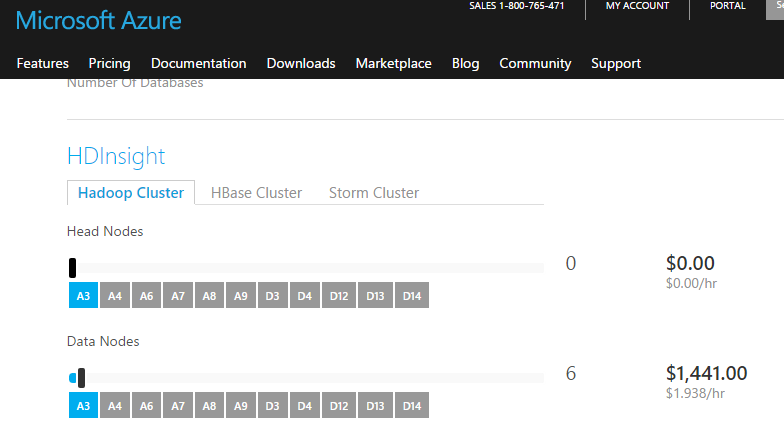


Figure 12. MS Azure pricing method per hour, for HDInsight / hadoop Cluster.

# Research Outcome

A proper access control framework for Big Data is the main aim and outcome of this research. The framework will provide practical solutions for Big Data privacy, integrity and authority.

The access control framework solution will support the followings:

1. Compatibility with structured and unstructured data.
2. Compatibility with the current access control RBAC and ABAC.
3. Resolves the Big Data analytics security tamper.
4. Reduces the complexity of the current RBAC, and ABAC frameworks.
5. Integrated with the organization workflow, and business structure.
6. Keeps the confidentiality and integrity of data.
7. Shows a high-performance level in Big Data, and easy to scale-up the framework.
8. Categorize identities and identity providers based on the Level of Assurance or (LoA)

## Access Control Framework for Big Data

A novel access control framework will be introduced for Big Data, which is expected to present a proper solution for users, administrators, and developers. The framework may share the access control management between two parties, developers (SP side), and system administrators (users side).

The framework will be compatible with the current available access control models such as RBAC and ABAC. Current RBAC design is perceived as being inappropriate for Big Data solutions, for its complexity and data analytics problems. RBAC is available in most SQL products, but it only controls the access for database developers and system administrators. Current access control frameworks are attached to the software applications, they are developed and programmed in n-tier logical level, in order to control the application authorization only. Repository storage, are eventually kept away from any access control except with the Database Management System (DBMS). For example; a user is able to read “employees” salaries, if he/she was able to manipulate any access to the database.

Also, the access model should assign an ad hoc permission, by using pre-defined global policies. A decision should be automatically taken on the fly to grant or deny access, and with the access level permitted. The access grants or denies permission is based on the mathematical algorithm. An application should use RBAC and ABAC with different core method. The access framework should not interrupt the multi-DBMS, as it can be considered as another level of security in the multi-level DBMS.

## DBMS and Access Model

A relational database is a combination of columns and rows, each table contains a set of columns (attributes), and these attributes combine tables with each other. The combination is completed using relational algebra. The horizontal values are called tuples, while the vertical values are called attributes. SQL syntax are limited to: select, insert, update, delete, alter, drop, create, use, show, describe, grant, commit, and rollback. Seven main operators work with the “select” queries these are: selection, projection, union, set difference, Cartesian products, join or inner join [[20](#_ENREF_20)].

Software engineers use the term DBMS to express the access control on the database dictionary and management only. All DBMS systems are isolated from the software application management system that is connected directly to the relational database. The connection is usually conducted by using SQL queries. The term “Relational Database” represent the actual SQL database. Both classical analysis and design phases, or object oriented analysis and design phases should output relational database. ERD, normalization, process diagram, and others are analysis and design tools used in the classical development. While UML diagrams are analysis and design tools used in object-oriented development. However, both classical or object oriented analysis and design must deploy a relational database schema.

Two main available access controls for DBMS, these are: DAC, and MAC. Most DBMS use DAC, and few of them use MAC. RBAC is a middle access role, and currently used in the most popular DBMS. Both DAC and MAC are proposed by two different protection classes. With DAC, Discretionary Protection Class C is divided into two subclasses. With MAC, Mandatory Protection Class B is divided into three subclasses, as shown below:

* DAC class C:

1. Class C1: manages data ownership and sharing with the others.
2. Class C2: manages login, auditing, and resource isolation.

* MAC class B:

1. Class B1: label each data object with a classification level, Top Secret, Secret, Confidential, and Unclassified.
2. Class B2: High-security level of inferring the answer of the illegal query, by using the answer of a legal one.
3. Class B3: supports security administrator, audit, and recovery.

* Verified Protection class A: The highest security level, it requires a proof of mathematical calculations [[21](#_ENREF_21)].

Many DBMS products support DAC – C2 and MAC - B1, such these products use a terminology of multi-DBMS. Multi-level of security can be applied in traditional databases. Some security filtering system occurs within the DBMS kernel and below the SQL engine, while others are applied on higher levels.

## Attribute Value Access Control Model

A non-conventional framework is proposed in this document. The novel framework uses attribute-based values, assigned to each object’s attribute. The attribute values can be attached to database schemas or files.

Let us start with general definitions that identify the major principles.

### General Definitions

General definitions are proposed, to outline the access framework:

* **Definition 1**: object (o) is defined as any file or instance of a class.
* **Definition 2**: subject (s) is defined as any user requesting access to objects
* **Definition 3**: procedure (pr) is a low-level operation such as write\_to\_file, sql\_select, create\_xml, add\_to\_ldap\_schema.
* **Definition 4**: method (m) is a collection of procedures such as addUser(), returnBook(), getDueDate() and so on. each object is managed by a set of methods.
* **Definition 5**: each object is uniquely identified by two identities these are unique ownerID, and idP, which represents the federation service that the subject was authenticated with, or attached to the object. Both IDs are attached to the object metadata. This denotes that each idP is presented by a global unique id, while the owner of the object is also given a global unique id, which usually represents the organization global number.

ownerID=orgID 🡪 where ID=10 digits number

idP=[[17](#_ENREF_17)]

*Note: organizations may have more than one idP, and each idP may provide a different Level of Assurance*.

* **Definition 6**: each object attribute (oa) is given an access level (al), and access level value (alv), these parameters are given during the application analysis and design:

∀ oa: al ∈ oa 🡪 al={TS,S,C,U} ~ {1,2,3,4}

where: TS=Top Secret, S=Secret, C=Confidential, U=Unclassified

and

∀ al: [alvx - alvy]=[alvx ∧ alvy ∈ alv: 0 ≤ alv[al]action ≥ 1000]

Where x: minimum, y: maximum

And action={read,write,delete}

* **Definition 7**: the ownership (ow) is defined as the object security access level that the users gain, for instance if the user is the object owner; then s/he can gain full access to the object. The ownership is determined by two parties, which are usually the SP and an organization, or between an organization and it is a partner.

∀ o: ow ∈ o 🡪 ow=={owner, co-owner, partner, customer, public}

~ {1,2,3,4}

Where owner: is the highest security access, and public is the lowest.

The object owners are allowed to control fully the metadata, security levels, and any other object parameters. For example in database objects, only owners can access data dictionary for database creation, alteration and others. Table 8 shows the ownership database permission.

Table 2. The permitted SQL commands and ownership level.

|  |  |
| --- | --- |
| **SQL COMMAND** | **MANAGMENET LEVEL BY OWNERSHIP** |
| AUDIT SYSTEM | OWNER |
| PROCESS | OWNER |
| RELOAD | OWNER |
| SHOW DATABASES | OWNER |
| SHUTDOWN | OWNER |
| USE | OWNER |
| DUMP (BACKUP) | OWNER |
| SHOW TABLES | OWNER |
| ALTER | OWNER |
| DROP TABLE | OWNER |
| CREATE | OWNER |
| SELECT | OWNER, CO-OWNER, PARTNER, CUSTOMER, PUBLIC |
| INSERT | OWNER, CO-OWNER, PARTNER, CUSTOMER, PUBLIC |
| UPDATE | OWNER, CO-OWNER, PARTNER, CUSTOMER, PUBLIC |
| DELETE | OWNER, CO-OWNER, PARTNER |

* **Definition 8**: the owner is the only member who is permitted to access metadata or database schema for any object.
* **Definition 9**: each role belongs to an organization falls to one access level (al), higher access levels roles start from 1. The security inheritance in the hierarchy structure of an organization starts from the root to the leaves. This implies that if role B as a part of role A; suppose role A(al=1), and role B(al=2), this denotes:

If B ∈ A AND B can read oa1 🡪 A can read oa1

If B ∈ A AND B can write in oa2 🡪 A can read oa2

* **Definition 10**: permission level (pl) for an object is a fixed security access value (alv) between 0 – 1000, where read permission denotes the lowest security value, the edit denotes medium security value, and the delete is the highest. Higher security value for a method requires equal or higher security value for a subject to be granted access. Security value presents the importance of the action intended by the user.
* **Definition 11**: each user requiring access is given four main values; these are: user ID, permission level, idP number that the user was authenticated by, and Level of Assurance (LoA), these variables can be represented as:

S 🡺 {id, pl, idP,loa}

With considering

id= [user id is 10 digits number]

pl=[0-1000]

idP=[id provider is 10 digits number]

loa={4,3,2,1}

loa is used for alv access calculation, the values were given are:

loa1=0, loa2=30, loa3=50, loa4=70

* **Definition 12:**

Suppose a group of roles (R(1),R(2),R(3)….R(n)) is belong to one department (D), and these roles constitute a branch of the company’s structure. The branch starts from R(1), which is highest managerial level down to the lowest managerial level (Rn). If R(1) access level is (x); then R(2) access level is (x+1), and R(n) access level is (x+(n-1).

* 1. If R(n) is permitted to access an attribute of an access level (x) for (read(r),write(w),delete(d)); then R(n+1) is permitted to access this attribute for (read,write) only; if R(n) was permitted by the workflow (w).
  2. If R(n) is permitted to access an attribute of an access level (x) for (read(r),write(w),delete(d)); then R(n+2), R(n+3),…R(n+x) are permitted to access this attribute for (read) only; if R(n) was permitted by the workflow (w).
  3. If R(n) is permitted to access an attribute of an access level (x) for (read(r),write(w)); then R(n+1), R(n+2), R(n+3)… R(n+x) are permitted to access this attribute for (read) only; if R(n) was permitted by the workflow (w).
* **Definition 13**: each attribute contains a permanent security value for read, write, and delete. While each user contains a temporal security value for read, write, and delete. Access Level Value (alv) for each attribute is mathematically calculated using the following factors:
  1. RBAC
  2. RBAC + ABAC

User Level Value (ulv) is mathematically calculated using the following factors:

* 1. RBAC

1. ABAC
2. RBAC + ABAC

The attributes can be successful if and only if ulv ≥ (alv -10)

### *Attribute Value-Based Access Control (AVBAC)*

Referring to definition 13, either factor (1) or (2) must be available, or both of them. Using both RBAC and ABAC supports a better decision making, and more accurate access level value. The federation service provides global values for each organization, this includes own organization members, and the other delegated organizations members. The global values determine the ownership level with the SP, the maximum access value, and the access method needed for each organization. For example organization A may be required to fulfill both ABAC and RBAC on authorization to gain any access. While organization B can be authorized by ABAC only, and so on. The calculations then are decided by the access method. Many factors can be considered with ABAC, known as environmental attributes, let us consider the following environmental attributes that can be used as default attributes:

1. Users have authenticated from own idP
2. Users have logged-in from certain IP addresses
3. Users have logged-in after n-trials
4. Users have logged-in inside/outside work hours
5. Users have logged-in using Assurance Level n
6. The last know log-in was from the country (y)

The above environmental variables gain a mark between [0-200], and is calculated using the following equation:

*ABACalv = ……………………(1)*

*ENv:* is the mark gained for each factor in the environmental variable, and

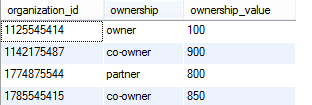
*n:* is the number of ABAC factors.

*Notice that the maximum alv for ABAC is only 800*

The above equation participates in 5% out of the total access level value (alv), if both ABAC and RBAC access methods are used, this denotes:

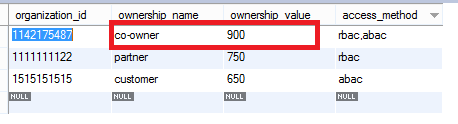
* ABAC + RBAC 🡪 alv(ABAC)=5% [max] alv(RBAC)=95% [max]
* ABAC only 🡪 alv(ABAC)=100% [max]
* RBAC only 🡪 alv(RBAC)=100% [max]

The SP participates in assigning the global variables, by giving the maximum possible alv based on the ownership type. The “owner” type can access a certain level of the data dictionary and other backup, and dump managerial functions. The SP keeps a record of organizations and their ownership type and alv, then each organization is given its ownership, and alv. The below tables show an example of SP global variables

Table 3. Global maximum ownership values provided by SP for each organization.   
 

The above global table defines the ownership values for organizations. Table 3 is stored in the SP side to provide the maximum ownership value for each organization. Table 4 is available in the federation service side. The organization pledges to use the maximum SLA agreed ownership value, as shown in table 4.The SP co-owner ownership value is 900, while the other delegated organizations are given lower ownership values. Moreover, this table describes the required access method for each organization.

Table 4. Global values provided by the federation service.



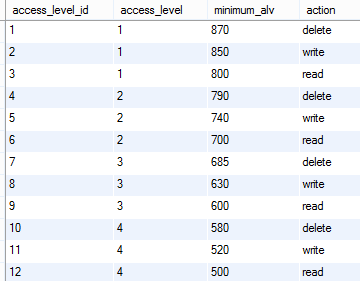
Moreover, other global data are pre-defined for access level and access level value, this data remains in the federation service to provide alv(al) parameters. Table 8-4 contains access level and minimum alv as per action. Notice that differences between access levels increases from TS to U, for example:

alv(TS)read – alv(S)delete=10

alv(S)read – alv(C)delete=15

alv(C)read – alv(U)delete=20

Table 5. Access level and the minimum alv with the permitted action.



The RBAC is determined on attribute based, this process starts on the SP side, and during the development phases. Developers usually assign use cases, then abstract objects. Referring to definition 1, the object is a part or instance of the class, and each object is defined by three parameters: object name, attributes, and methods. These attributes and methods, as shown below, are defined within the object-oriented analysis and design phases.

- Attribute 1

- Attribute 2

- Attribute 3

Object\_Name

- Method 1

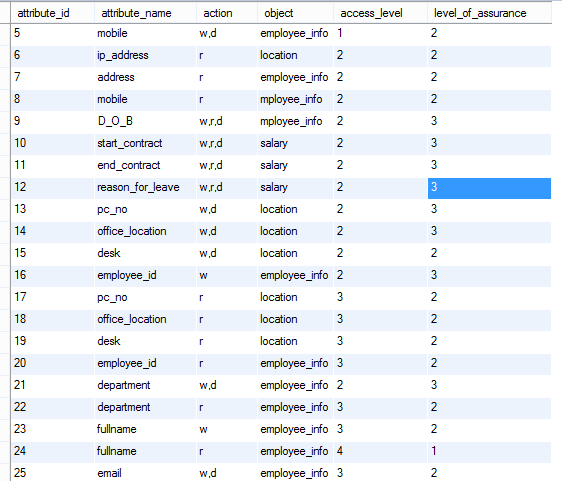
- Method 2

- Method 3

Figure 13. The three main components for an object.

SP developers define a ranking process for each attribute, this process will be discussed later, as it depends on different factors and variables. However, the ranking process will create the highest security attribute level for read, write, and delete. The access level can use definition 6 for object attribute and alv. Table 6 illustrates an example of an application software and attributes ranks, the table shows the access level using TS=1, S=2, C=3, U=4, notice that same attributes may carry different access levels for different action methods of read (r), write (w), and delete (d).

Table 6. SP Attributes and access levels.



To calculate RBAC and ABAC, let us consider a simple demo. With the following scenario of an employee management system provided by “Blue Sky Company”, and used by the BCB Company. The federation service is owned by the BCB Company. The following diagram shows BCB hierarchy structure.

Diagram 1. The BCB internal organization hierarchy structure.

The SP presents three 8s in the ER diagram, as shown in diagram 2.

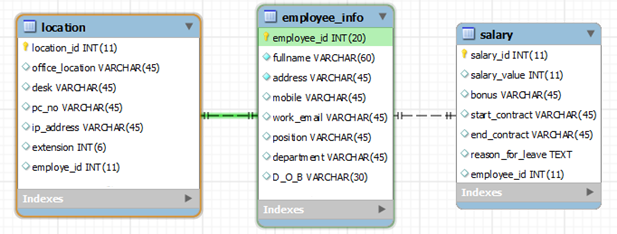


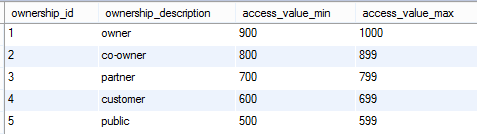
Diagram 2. Entity Relationship Diagram for Blue Sky Employee System.

Referring to the previous definitions and scenarios, the following abstracted processes identify the computational steps to calculate RBAC and ABAC alv. General permanent alv are stored in the SP side, while federation service generates an ad hoc access level value for any access attempt. Three main parties take a part in access control of RBAC and ABAC; these are the SP developers, the federated organization administrators, and the users who wish to access the SP application. Each one of the three parties determines the followings:

#### Developers

SP determines general global definitions for ownership access values. The access values are given as intervals [min.,max.] as shown in table 7.

Table 7. SP definition for general global access values intervals for each ownership.



The SP uses table 7, as a reference, on signing the SLA with the organizations. The following diagram illustrates the SP global data that are pre-defined for any organization access:

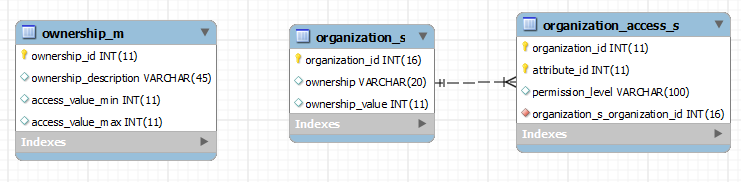


Diagram 3. The Sp global data, and the link between each organization and its ownership value. As shown in (table 3 - ownership\_m, and table 6 - organization\_s).

As shown in diagram 3, organization\_access\_s table is available and pre-defined for each organization to store it is own alv for each attribute. Different organizations have different attribute alv. Developers define three main tables during the development phase, these are

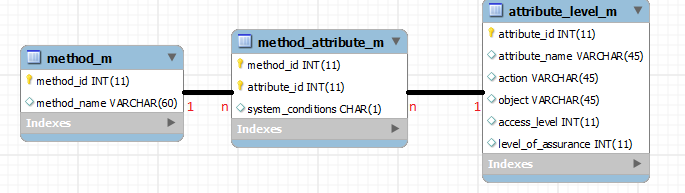


Diagram 4. Relationship between tables in the SP side (table 7- method\_m, table 5- attribute\_level\_m).

Diagram 4 is developed during the analaysis and design phases of the system. Objects are firstly defined during the analysis phase, and concludes the design phase with the attributes and methods. The above tables are developed by the SP, and can be given for federation services as an xml format file.

#### Federation Administrators

Administrators in BCB federation services need to assign security access policies to the Blue Sky System. Therefore, they need to use tables in diagram 4. The federation service provides tools to manage the business workflow. A simple workflow is shown in figure 14. The figure shows the steps of building a simple workflow based on the business needs, this enables a better management for each business role by controlling tasks, boundaries, and sequential procedures.

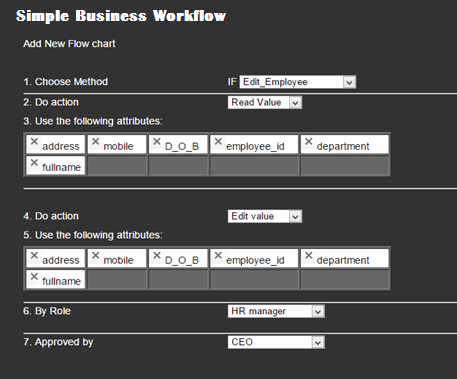
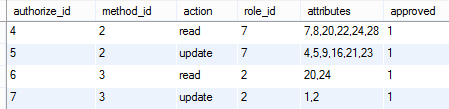


Figure 14. Simple Business Workflow example.

The flowchart helps processing alv for each attribute, firstly, the chosen method collects all available attributes connected to that method from database. Then it filters these attributes depending on the chosen action, some attributes are assigned for read, write, and delete, or a combination of them.

Secondly, the administrator keeps the required attributes and deletes the rest, then chooses the role that is supposed to action that method, and then assigns the approved authority. The results then will be stored in database as shown in Table 8.

Table 8. The business workflow database example.



Thirdly, the alv should be calculated for each attribute, and stored on the SP side, the calculation process follows these steps:

* *Read from table 4 and find the global variables for the home organization*
* *Read the ownership value and the access method from table 4*

*If RBAC only use the following:*

*…………………(2)*

*Example 1: attribute=salary\_value*

*Owv=900, loa=3*

*TOTalv(read)=900\*800/1000 + 50 = 785*

*TOTalv(write)=900\*850/1000 + 50 = 831*

*TOTalv(delete)=900\*870/1000 + 50 = 850*

* *If RBAC + ABAC use the following:*

*RABACalv =…..(3)*

Where *ENv* is the mark gained for each factor in the environmental variable, and *EN* is the number of ABAC factors.

Equation (3) uses both RBAC and ABAC, the exponential value used in ABAC provides a higher security level on accessing values with a large alv. The original equation is the summation of RBAC and ABAC percentage, as shown in Equation (4).

*RABACalv =(4)*

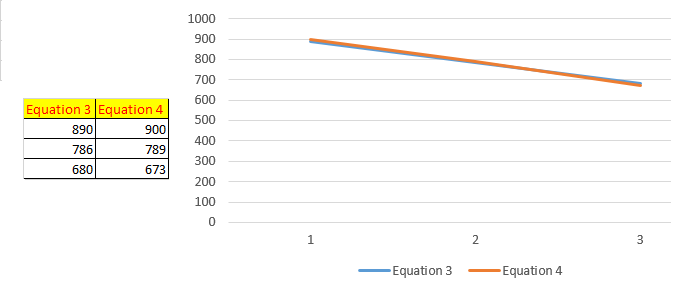


Figure 15. The slight difference on comparing between equation (3) and equation (4).

Notice that both equations reserve 5% ABAC weight of the total ARBACalv, while 95% weight is reserved for the RBAC.

#### Users

As mentioned before in definition 12, User Level Value (ulv) is mathematically calculated using the following factors: RBAC, ABAC, or RBAC + ABAC, the SP access control application triggers any action regarding database, and evaluates the attributes alv, by comparing it with the ulv.

* The access method is determined by the federation service
* If RBAC is used then an equation similar to equation (2) is used:

*…………………(5)*

Where ulv is read from the flowchart database.

* If ABAC is used then an equation similar to equation (1) is used:

*ABACulv = ……………………(6)*

*ENv:* is the mark gained for each factor in the environmental variable, and

*n:* is the number of ABAC factors.

If both of RBAC and ABAC are used then an equation similar to equation (3) is used:

*RABACulv =…..(7)*

### Comparing this Framework with the Conventional Frameworks

A brief comparison between this framework and the conventional used frameworks in the market can be established. In order to support a reasonable comparison, a small application was programmed as a demonstration. Figures 16, and 17 show two examples of web-application screenshots, which demonstrate the created application. The first figure shows the default user interface, and before adding any security permissions. While the second figure shows the interface after adding security permissions, using the access control workflow. The second figure clearly illustrates the company’s roles tasks. The application developers do not bother customizing special interfaces suit organization workflow.

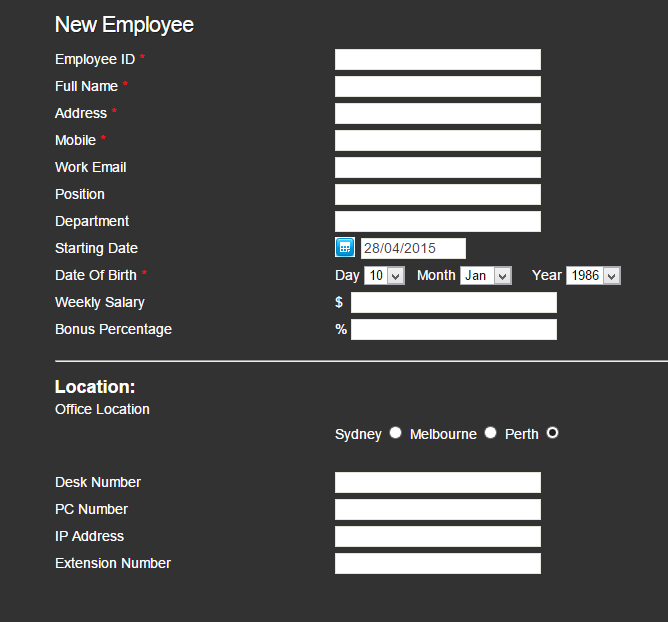


Figure 16. The default user interface for “Add a new user”, before assigning any access control.



Figure 17. The HR manager interface for “add a new user”, after assigning an access control.

The interface in figure 17 can be created using the administrative tool in figure 14, which demonstrates a simple business workflow. Similar interfaces can be created for each role, any authorized manager can create many workflow instances for any role. The authorized manager must be aware of the business needs and the workflow activities. However, the authorized person needs to assign a higher level role of the approvers group.

Table 9. Security comparison between conventional RBAC and AVBAC.

|  |  |
| --- | --- |
| **SECURITY COMPARISON** | |
| **Convectional RBAC** | **Attribute Value-Based Access Control (AVBAC)** |
| Separate the access control fine-grained access control for the application from the access control for database | Integrate both application and database access control in one data |
| Duplicate management tasks between DBMS-RBAC and application RBAC | Single point of management for database and application |
| Pre-defined roles, to suit the system tasks and functions. Created roles do not necessarily represent the actual roles and departments | No roles definition within the application system, therefore, roles represent the actual organization designation. The same role can be assigned for many applications. |
| Access control is connected tightly with the application system n-tier programming level. | Access control is independent of the application system n-tier programming level. |
| Unable to control Big Data Analytics | Can control Big Data Analytics by forcing the RBAC and ABAC on the database level. |
| Unable to control the delegating issues with the cloud applications, delegated users can not apply their roles | Permissions are not connected with the roles. Therefore, any user can access and without the administrator interaction |
| Hard to implement by developers, and hard to manage by administrators | Easy to implement and manage by both developers and administrators |
| **PERFORMANCE COMPARISON** | |
| Possibly better performance on large size of data, extra experiments are needed | Possibly slow especially on scanning all flowcharts to determine the alv, and ulv. |

# Budget Estimation during the PhD study

Table 1. Budget estimation for the expected expenses during the PhD study.

|  |  |  |  |
| --- | --- | --- | --- |
| **Item** | **Estimation for 2014 - 2015** | **Estimation for 2015 - 2016** | **Estimation for 2016 - 2017** |
| MATALB and a Laptop | Provided by UWS | | |
| Conference papers (registration and attendance) | $1000 (registration + local attendance) | $4600 (registration + overseas attendance) | $4600 (registration + overseas attendance) |
| Opnet IT Guru simulation software (without support tools) | Free for education use | | |
| Opnet Guru support | - | $2000 | $2000 |
| Thesis proof reading | - | - | $2000 |
| Five Virtual Machines (VMs), using VMware | - | Provided by UWS | Provided by UWS |
| MS Azure, Hadoop(HDInsight),  With 6 nodes for 40 days | - | - | $1860 |
|  | | | |
| Total per year | $1000 | $6600 | $10460 |
| Total (2015-2017) | $18060 | | |

# Research Time-Line

The below table shows the research time-line starting from the last July 2014.

|  |  |
| --- | --- |
| Time period | Task |
| Jul 2014 -Feb 2015 | * Literature review. * Identify research gaps * Establish research questions. * Prepare two reports for discussions with the supervisors (on Big Data, SAML, and access control models.) * Investigate the suitability of some simulation programs (MATLAB, OpNet, and GreenCloud). * Presentation of works. |
| Feb-Jun (2015) | * Literature review. * Complete the research methodology. * Revise the research questions and research outcomes. * Work on, revise, and complete the Confirmation of Candidate. * Presentation of the works during the CoC meeting. * Complete the required changes in the CoC document. |
| Jun-Dec (2015) | * Work on Stage one of the methodology. * Develop a theoretical framework. * Develop varieties of use case scenarios on access control for big data. * Write a small software application, the program will be coded using php programming language, with MySQL database, this application is essential to amend the proposed framework. * Start the preparation for the second stage. * Participate in a conference for Big Data security. * Continue developing the software while developing the framework. * Use MATLAB to test and modify the mathematical equations that are used in the proposed framework. * Complete Stage One, and start Stage two of the methodology |
| Jan-Jun (2016) | * Work on stage two of the methodology by setting up the three servers with the required software programs on each servers as follow:   The first two servers act as a federation service and contain the following services:  • Apache Server  • Java Open Single Sign On or (JOSSO)  • Software Application program using Java and PHP, which implements the access model framework  • OpenSSL to encrypt XML and metadata  The third server (SP) is provided by different services, such as:  • NAS services and files sharing. The service should provide different types of files and folders, such  as; text, images, audio, video and web pages.  • Apache Server  • Open source CRM application  • Software Application program using Java and PHP.   * Complete stage two of the methodology * Modify the access control framework for the big data * Evaluate stage two results and modify stage three consequently, if needed. * Write a paper about the “Attribute Value-Based Access Control for Big Data” |
| Jun-Dec (2016) | * Work on stage three of the methodology.   Start with 5 VMs, then move to a higher scale. The 5 servers are divided into two federation service servers, similar to the previous stage, and 3 hadoop servers, with the following applications:   * + NAS services, and files sharing. The service should provide different types of files and folders, such as; text, images, audio, video and web pages.   + Apache Server   + Open source CRM application   + Software Application program using Java and PHP.   + MapReduce application Cloudera * Evaluate and examine the proposed framework algorithms for further improvement using MATLAB, this evaluation will be used for performance wise. * Analyze the results and finding of stage three. |
| Jan-June (2017) | * Scale-up the range by using more hadoop servers, this can be done by using MS Azure. Buy 6 hadoop nodes for 45 days. * Write a paper about the proposed framework and social network. * Evaluate the results and improve the framework on need. * Identify improvements and potentially amend the research projects. * Participate in a conference for Big Data security. |
| Jun-Dec (2017) | * Work on any identified improvements or project amendments * Presentation of the results and discussions * Thesis writing, discussions, and revisions. * Write a paper about “Resolving the Analytics Problem Using AVBAC framework”. |
| Jan – Apr (2018) | * Finalize any loose ends, simulations, and any potential gaps * Thesis finalization and proofreading * Thesis submission |

# Publication Plan

|  |
| --- |
| 1. First Conference Publication (Jun – Dec 2015) |
| Title: A non-conventional Framework for Big Data Analytics. |
| Possible conference publisher:   * Wessex Institute for conference. Publish WitPress – Information and Communication Technology – Spain- (3-5 May 2016) * International Conference on Computing, Networking and Communications (ICNC) – IEEE USA - (15-18 Feb. 2016) * IC2E 2016 Home – IEEE – Berlin/Germany (April 4 - 8, 2016) |

|  |
| --- |
| 1. Second Journal Publication (Jan – Jun 2016) |
| Title: Attribute Value-Based Access Control for Big Data. |
| Possible publishers:   * Wiley Online Library Journal. * Intelligent Data Analysis Journal (IOS Press). * IEEE Online Plus. * Journal of Big Data - Springer Open Journal |

|  |
| --- |
| 1. Third Publication (Jun – Dec 2016) |
| Title: Attribute Value-Based Access Control Framework for Social Network. |
| Possible publisher:   * Wiley Online Library Journal. * Intelligent Data Analysis Journal (IOS Press). * IEEE Online Plus. * Journal of Big Data - Springer Open Journal |

|  |
| --- |
| 1. Fourth Conference Publication (Jan – Jun 2017) |
| Title: A framework for Big Data Analytics. |
| Possible conference publisher:   * To be decided later |

|  |
| --- |
| 1. Fifth Publication (Jun – Dec 2017) |
| Title: Resolving the Analytics Problem Using AVBAC framework. |
| Possible publisher:   * Wiley Online Library Journal. * Intelligent Data Analysis Journal (IOS Press). * IEEE Online Plus. * Journal of Big Data - Springer Open Journal |

# Conclusion

Two main concerns in Big Data were introduced; these are management complexity and data analytics security issues. A research question was introduced to rectify these two concerns. A proper methodology was introduced to tackle the research question. An experimental method was proposed, with three stages.

An Attribute Value-Based Access Control framework (AVBAC) was briefly introduced to highlight the next coming steps. The framework uses the principle of assigning numerical values to attributes, then attributes values are compared to any user’s value on access attempt. The non-conventional model resolves the direct access to the database using analytics method. This solution integrates both DBMS with the n-tier level Role-Based Access Control.

More experiments are needed to compromise a practical and a proper solution that reduces management complexity and prevent data analytics tamper. However, the proposed solution will be flexible enough to provide an ease of use for both administrators and developers. The model depends substantially on the organization flowchart, and the real organization structure and roles. These approaches prevent business roles fluctuation and changes to match the used software application. Instead, this model anneals organization roles and structure within the workflow framework.

# References

[1] J. K. Jinzhu Kong, "A Practical Approach to Improve the Data Privacy of Virtual Machines," ed, 2010, pp. 936-941.

[2] M. a. Alam, *Oracle NoSQL database : real-time big data management for the enterprise*: New York : McGraw-Hill Education, 2014, 2014.

[3] S. Chaudhuri, "What next?: a half-dozen data management research goals for big data and the cloud," in *PODS '12*, M. Lenzerini, M. Benedikt, Kr, and M. tzsch, Eds., ed: ACM, 2012, pp. 1-4.

[4] P. Russom, "BIG DATA ANALYTICS," *TDWI RESEARCH/IBM,* vol. 4, 2011.

[5] X. Jin, R. Sandhu, and R. Krishnan, "A unified attribute-based access control model covering DAC, MAC and RBAC," vol. 7371, ed, 2012, pp. 41-55.

[6] D. Rountree, *Federated identity primer*: Federated identity primer, Syngress,, 2013.

[7] D. E. B. a. L. J. LaPadula, "Secure Computer Systems: Unified Exposition and Multics Interpretation,," *Technical Report MTR-2997,* 1976.

[8] Klaus Rebensburg, Charles Youman, and V. Alturi, *RBAC '00: Proceedings of the fifth ACM workshop on Role-based access control*. Germany: ACM, 2000.

[9] D. Ferraiolo, *Role-based access control*, 2nd ed. ed. Boston: Boston : Artech House, 2007.

[10] C. Ruan and V. Varadharajan, "Dynamic delegation framework for role based access control in distributed data management systems," *An International Journal,* vol. 32, pp. 245-269, 2014.

[11] B. Tanaka, *Take Control of Permissions in Snow Leopard*. Sebastopol: Sebastopol : TidBITS, 2010.

[12] M. Shtern, B. Simmons, M. Smit, and M. Litoiu, "Toward an ecosystem for precision sharing of segmented Big Data," in *Cloud Computing (CLOUD), 2013 IEEE Sixth International Conference on*, 2013, pp. 335-342.

[13] William E. Burr, Donna F. Dodson, and W. Timothy Polk, "Electronic authentication guideline.

Technical report, National Institute for

Standards and Technology," 2006.

[14] S. C. Aliance. (2013). *Assurance Levels Overview and*

*Recommendations*. Available: <http://www.smartcardalliance.org/resources/pdf/identity_council_assurance_levels_brief.pdf>

[15] A. E. A. Maya I. Chehab, "Assurance in Identity Management Systems," *IEEE,* 2010.

[16] T. White, *Hadoop : the definitive guide*, 3rd ed. ed. Farnham: Farnham : O'Reilly, 2012.

[17] Y. Kwon, "Hadoop Internals," 2014.

[18] Apache, "MapReduce Tutorial," 2013.

[19] L. C. L. Dancheng, L. Binsheng,, "H-RBAC: A Hierarchical Access Control Model

for SaaS Systems " 2011.

[20] G. D. J. Patricia A. Dwyer, Bhavani M. Thuraisingham, "Multi-level Security in Database Management Systems " *Honeywell Compufer Scrences Center, ,* 1990.

[21] S. E. S. Sumathi, *Fundamentals of Relational Database Management Systems*: Springer, 2007.

# Appendix A (List of Abbreviations and Acronyms)

* 1. ABAC: Attribute-Based Access Control
  2. ACL: Access Control List
  3. AD: Active Directory
  4. AES: Advance Encryption Standard
  5. AVBAC: Attribute Value-Based Access Control
  6. AzMan: MS Authorization Manager
  7. CoC: Confirmation of Candidate
  8. COM: Component Object Model
  9. CRM: Customers Relationship Management
  10. DAC: Discretionary Access Control
  11. DBMS: Database Management System
  12. HDFS: Hadoop File System
  13. IaaS: Infrastructure as a Service
  14. JigDFS: JigSaw Distributed File System
  15. JOSSO: Java Open Single Sign On
  16. LoA: Level of Assurance
  17. MAC: Mandatory Access Control
  18. NAS: Network-Attached Storage
  19. NFS: Network File System
  20. NIST: National Institute for Standards and Technology
  21. NoSQL: Not Only SQL
  22. OLAP: Online Analytical Processing
  23. OLTP: Online Transaction Processing
  24. PaaS: Platform as a Service
  25. RAdAC: Risk-Adaptable Access Control
  26. RAID: Redundant Array of Inexpensive Disk
  27. RBAC: Role-Based Access Control
  28. RSA: Extensible Markup Language
  29. SaaS: Software as a Service
  30. SAML: Security Assertion Markup Language
  31. SP: Service Provider
  32. SSO: Single Sign On
  33. UI: User Interface
  34. UML: Unified Modelling Language
  35. URL: Unified Resource Locator
  36. WAP: Wireless Application Protocol.
  37. XML: Extensible Markup Language