NIST Special Publication 800-162

Guide to Attribute Based Access Control (ABAC) Definition and Considerations (Draft)

Vincent C. Hu
David Ferraiolo
Rick Kuhn
Arthur R. Friedman
Alan J. Lang
Margaret M. Cogdell
Adam Schnitzer
Kenneth Sandlin
Robert Miller
Karen Scarfone

COMPUTER SECURITY

NIST
National Institute of
Standards and Technology
U.S. Department of Commerce

NIST Special Publication 800-162

Guide to Attribute Based Access Control (ABAC) Definition and Considerations (Draft)

Vincent C. Hu
David Ferraiolo
Rick Kuhn
Computer Security Division
Information Technology Laboratory

Arthur R. Friedman
Alan J. Lang
Margaret M. Cogdell
National Security Agency

Adam Schnitzer
Booz Allen Hamilton

Kenneth Sandlin
Robert Miller
The MITRE Corporation

Karen Scarfone
Scarfone Cybersecurity

April 2013

U.S. Department of Commerce
Rebecca Blank, Acting Secretary

National Institute of Standards and Technology
Patrick D. Gallagher, Under Secretary of Commerce for Standards and Technology and Director

Authority

This publication has been developed by NIST to further its statutory responsibilities under the Federal Information Security Management Act (FISMA), Public Law (P.L.) 107-347. NIST is responsible for developing information security standards and guidelines, including minimum requirements for Federal information systems, but such standards and guidelines shall not apply to national security systems without the express approval of appropriate Federal officials exercising policy authority over such systems. This guideline is consistent with the requirements of the Office of Management and Budget (OMB) Circular A-130, Section 8b(3), *Securing Agency Information Systems*, as analyzed in Circular A-130, Appendix IV: *Analysis of Key Sections*. Supplemental information is provided in Circular A-130, Appendix III, *Security of Federal Automated Information Resources*.

Nothing in this publication should be taken to contradict the standards and guidelines made mandatory and binding on Federal agencies by the Secretary of Commerce under statutory authority. Nor should these guidelines be interpreted as altering or superseding the existing authorities of the Secretary of Commerce, Director of the OMB, or any other Federal official. This publication may be used by nongovernmental organizations on a voluntary basis and is not subject to copyright in the United States. Attribution would, however, be appreciated by NIST.

National Institute of Standards and Technology Special Publication 800-162
Natl. Inst. Stand. Technol. Spec. Publ. 800-162, 54 pages (April 2013)
CODEN: NSPUE2

Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by NIST, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

There may be references in this publication to other publications currently under development by NIST in accordance with its assigned statutory responsibilities. The information in this publication, including concepts and methodologies, may be used by Federal agencies even before the completion of such companion publications. Thus, until each publication is completed, current requirements, guidelines, and procedures, where they exist, remain operative. For planning and transition purposes, Federal agencies may wish to closely follow the development of these new publications by NIST.

Organizations are encouraged to review all draft publications during public comment periods and provide feedback to NIST. All NIST Computer Security Division publications, other than the ones noted above, are available at http://csrc.nist.gov/publications.

Public comment period: *April 22, 2013* through *May 31, 2013*

National Institute of Standards and Technology
Attn: Computer Security Division, Information Technology Laboratory
100 Bureau Drive (Mail Stop 8930) Gaithersburg, MD 20899-8930
Email: vincent.hu@nist.gov

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the Nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analyses to advance the development and productive use of information technology. ITL's responsibilities include the development of management, administrative, technical, and physical standards and guidelines for the cost-effective security and privacy of other than national security-related information in Federal information systems. The Special Publication 800-series reports on ITL's research, guidelines, and outreach efforts in information system security, and its collaborative activities with industry, government, and academic organizations.

Abstract

This document provides Federal agencies with a definition of attribute based access control (ABAC). ABAC is a logical access control methodology where authorization to perform a set of operations is determined by evaluating attributes associated with the subject, object, requested operations, and, in some cases, environment conditions against policy, rules, or relationships that describe the allowable operations for a given set of attributes. This document also provides considerations for using ABAC to improve information sharing within organizations and between organizations while maintaining control of that information.

Keywords

access control; attribute based access control (ABAC); authorization; privilege; access control model; access control policy; access control mechanism;

Acknowledgments

The authors, Vincent C. Hu, David Ferraiolo, and Rick Kuhn of the National Institute of Standards and Technology (NIST); Arthur R. Friedman, Alan J. Lang, and Margaret M. Cogdell of the National Security Agency (NSA); Adam Schnitzer of Booz Allen Hamilton; Kenneth Sandlin and Robert Miller of The MITRE Corporation; and Karen Scarfone of Scarfone Cybersecurity, wish to thank their colleagues who reviewed drafts of this document, including the following: Kevin Miller (NSA), Anne P. Townsend (The MITRE Corporation), Jennifer Newcomb (Booz Allen Hamilton), John W. Tolbert (Boeing), Jeremy Wyant (General Dynamics), Ian Glazer (Gartner), Scott C. Fitch (Lockheed Martin), Tim Schmoyer (Jericho Systems), Dave Coxe (Criterion Systems), Don Graham (Radian Logic), and Ronald Ross (NIST).

The authors also gratefully acknowledge and appreciate the comments and contributions made by government agencies, private organizations, and individuals in providing direction and assistance in the development of this document.

Trademark Information

Table of Contents

List of Figures

Executive Summary

The concept of Attribute Based Access Control (ABAC) has existed for many years. It represents a point on the spectrum of logical access control from simple access control lists to more capable role-based access, and finally to a highly flexible method for providing access based on the evaluation of attributes.

In November 2009, the Federal Chief Information Officers Council (Federal CIO Council) published the Federal Identity, Credential, and Access Management (FICAM) Roadmap and Implementation Plan v1.0 [FEDCIO1], which provided guidance to federal organizations to evolve their logical access control architectures to include the evaluation of attributes as a way to enable access within and between organizations across the Federal enterprise. In December 2011, the FICAM Roadmap and Implementation Plan v2.0 [FEDCIO2] took the next step of calling out ABAC as a recommended access control model for promoting information sharing between diverse and disparate organizations.

Despite the clear guidance to implement ABAC, to date there has not been a comprehensive effort to formally define or guide the implementation of ABAC within the Federal Government. This document serves a two-fold purpose. First, it aims to provide Federal agencies with a definition of ABAC and a description of the functional components of ABAC. Second, it provides planning, design, implementation, and operational considerations for employing ABAC within a large enterprise with the goal of improving information sharing while maintaining control of that information.

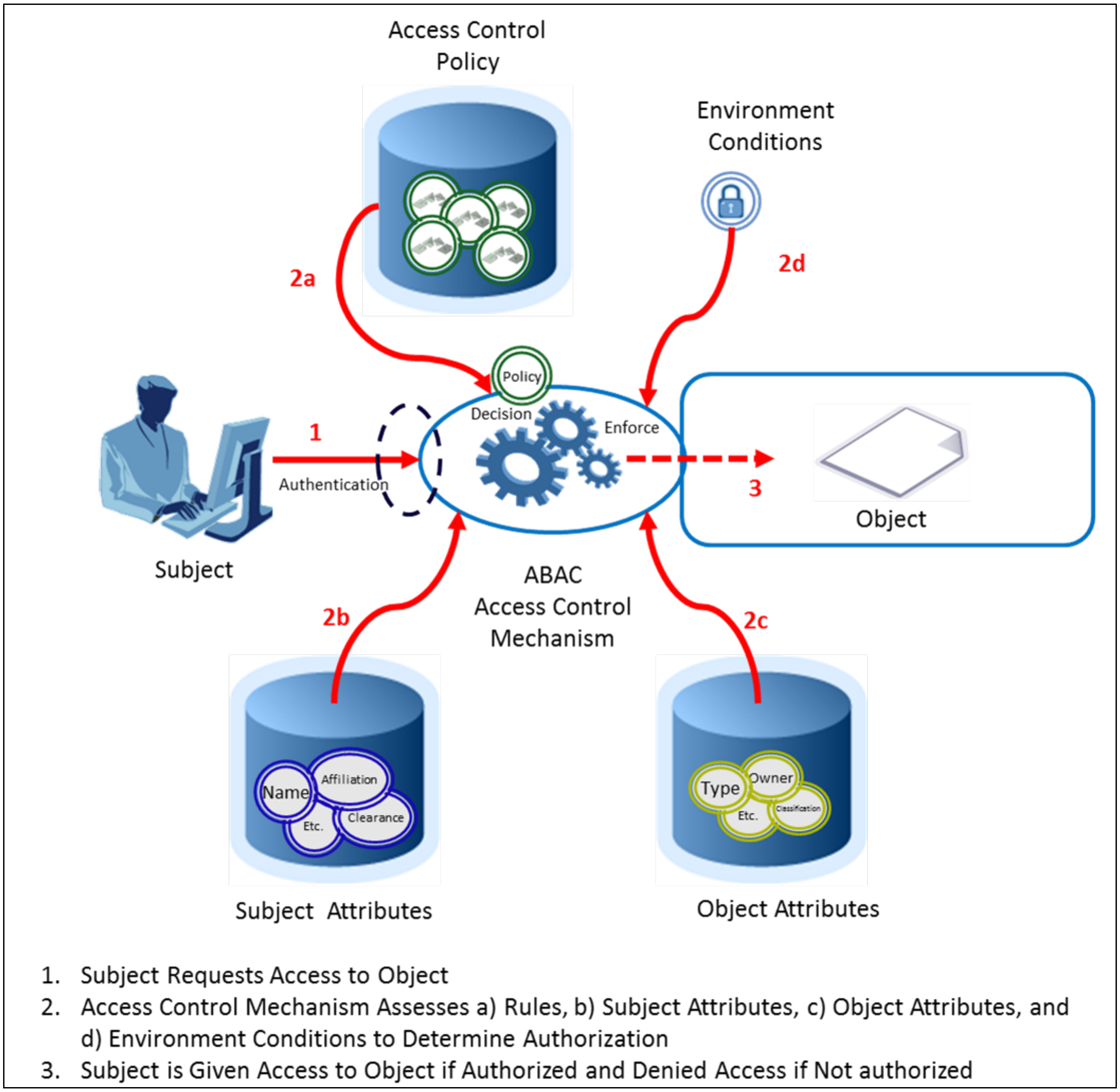
ABAC is a logical access control model that is distinguishable because it controls access to objects by evaluating rules against the attributes of the entities (subject and object) and the environment relevant to a request. Attributes may be considered characteristics of anything that may be defined and to which a value may be assigned. In its most basic form, ABAC relies upon the evaluation of attributes of the subject, attributes of the object, and a formal relationship or access control rule defining the allowable operations for subject-object attribute combinations. All ABAC solutions contain these basic core capabilities to evaluate attributes and enforce rules or relationships between those attributes. ABAC systems are capable of enforcing both Discretionary Access Control (DAC) and Mandatory Access Control (MAC) models. Moreover, ABAC systems can enable Risk-Adaptable Access Control (RAdAC) solutions, with risk values expressed as variable attributes. For more information on RAdAC, see http://csrc.nist.gov/news_events/privilege-management-workshop/presentations/Bob_McGraw.pdf.

The rules or policies that can be implemented in an ABAC model are limited only to the degree imposed by the computational language. This flexibility enables the greatest breadth of subjects to access the greatest breadth of objects without specifying individual relationships between each subject and each object. For example, a subject is assigned a set of subject attributes upon employment (e.g., Nancy Smith is a *Nurse Practitioner* in the *Cardiology Department.*). An object is assigned its object attributes upon creation (e.g., a folder with *Medical Records* of *Heart Patients).* Resources may receive their attributes either directly from the creator or as a result of automated scanning tools. The object owner creates an access control rule to govern the set of allowable operations (e.g., all *Nurse Practitioners* in the *Cardiology Department* can *View* the *Medical Records* of *Heart Patients).* Adding to the flexibility, attributes and their values may then be modified throughout the lifecycle of subjects, objects, and attributes without modifying each and every subject/object relationship. This provides a more dynamic access control capability as access decisions can change between requests when attribute values change.

Provisioning attributes to subjects and objects governed by an access control ruleset that specifies what operations can take place enables object owners or administrators to apply access control policy without prior knowledge of the specific subject and for an unlimited number of subjects that might require access. As new subjects join the organization, rules and objects do not need to be modified. As long as the subject

is assigned the attributes necessary for access to the required objects (e.g., all Nurse Practioners in the Cardiology Department are assigned those attributes), no modifications to existing rules or object attributes are required. This benefit is often referred to as accommodating the external user and is one of the primary benefits of employing ABAC.

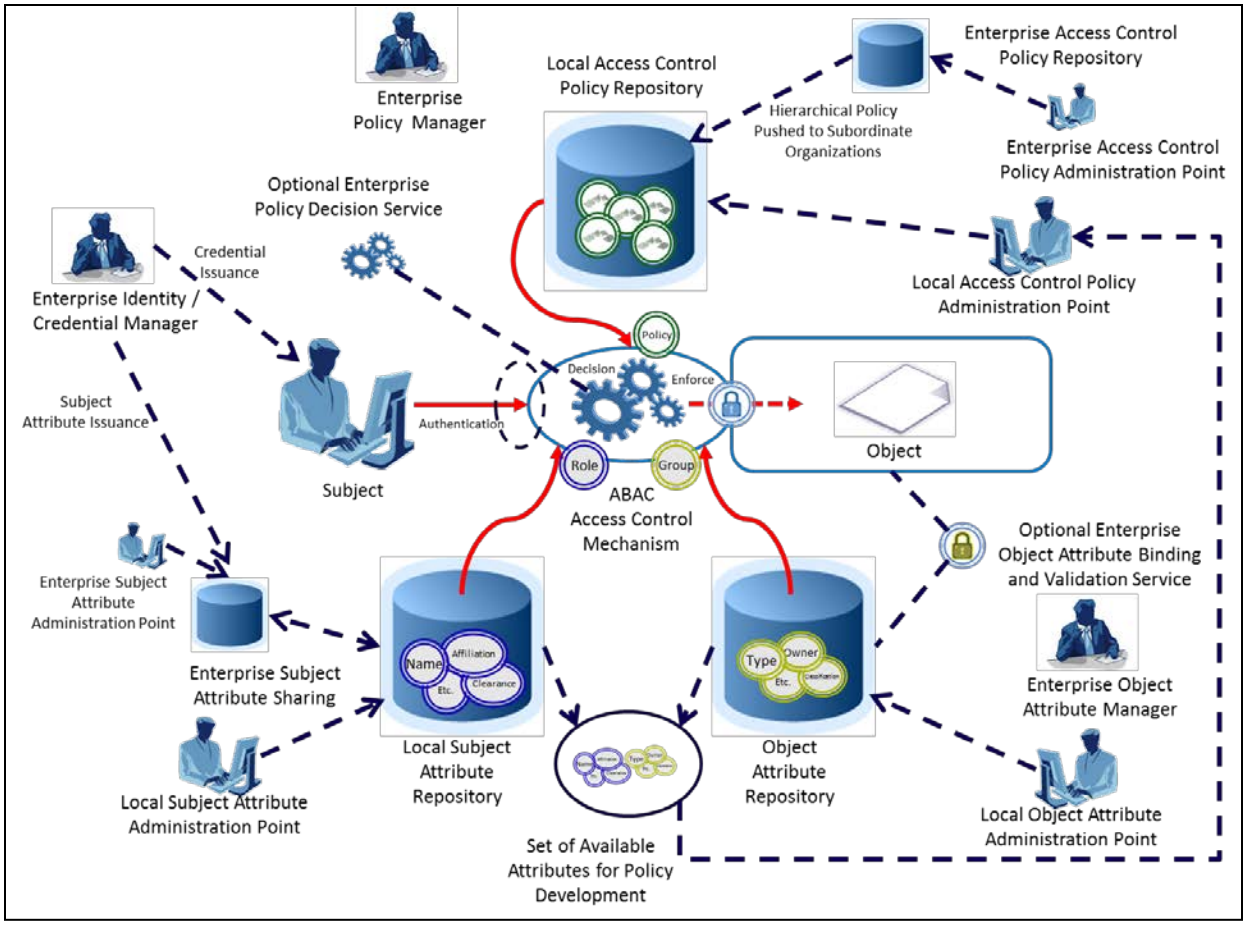
Figure 1 illustrates a simple ABAC access control scenario where a subject requests access to an object through some access control mechanism. This mechanism assembles policy, subject attributes, and object attributes to determine and enforce a set of allowable operations by the subject upon the object.



1. Subject Requests Access to Object
2. Access Control Mechanism Assesses a) Rules, b) Subject Attributes, c) Object Attributes, and d) Environment Conditions to Determine Authorization
3. Subject is Given Access to Object if Authorized and Denied Access if Not authorized

Figure 1: Basic ABAC Access Control Scenario

When deployed across an enterprise for the purposes of increasing information sharing between diverse organizations, ABAC implementations become much more complex—relying on the existence of extensive attribute management infrastructures, machine readable policy ontologies, and interoperable access control mechanisms deployed to uniquely diverse networks. Added to the basic ABAC scenario is an array of functions that must be present before the first access decision can be rendered.



Enterprise Policy Manager
Local Access Control Policy Repository
Hierarchical Policy Pushed to Subordinate Organizations
Enterprise Access Control Policy Repository
Enterprise Access Control Policy Administration Point
Optional Enterprise Policy Decision Service
Credential Issuance
Enterprise Identity / Credential Manager
Local Access Control Policy Administration Point
Policy
Decision
Enforce
Subject Attribute Issuance
Authentication
Subject
Object
Role
Group
ABAC Access Control Mechanism
Optional Enterprise Object Attribute Binding and Validation Service
Enterprise Subject Attribute Administration Point
Enterprise Subject Attribute Sharing
Name
Affiliation
Clearance
Etc.
Type
Owner
Classification
Etc.
Enterprise Object Attribute Manager
Local Subject Attribute Administration Point
Local Subject Attribute Repository
Object Attribute Repository
Local Object Attribute Administration Point
Set of Available Attributes for Policy Development

Figure 2: Enterprise ABAC Access Control Scenario

Figure 2 illustrates the complexity of ABAC within an enterprise. In addition to the basic policy, attribute, and access control mechanism requirements, the enterprise must support management functions for enterprise policy development and distribution, enterprise identity and subject attributes, subject attribute sharing, enterprise object attributes, authentication, and access control mechanism deployment and distribution. The development and deployment of these capabilities requires the careful consideration of a number of factors that will influence the design, security, and interoperability of an enterprise ABAC solution. (Additional information on enterprise ABAC concepts can be found in Section 3 of this document.) These factors can be summarized around a set of principles:

Establish the Business Case for ABAC Implementation. What are the costs of developing/acquiring new capabilities and transitioning away from old capabilities? What are the hidden costs and risks in exposing sensitive information to external users in the enterprise, the new governance structures required to manage shared capabilities, and the infrastructure to document and manage policies that were previously human-in-the-loop decisions? How are privileges managed, monitored, and validated for compliance? Which datasets, systems, applications, and networks need ABAC capabilities?

Understand the Operational Requirements and Overall Enterprise Architecture. What objects will be exposed to the enterprise for information sharing? How will subject attributes be shared? How will object attributes be used consistently? What are the access control rules and how are they captured, evaluated, and enforced? How is trust managed within the enterprise?

Establish or Refine Business Processes to Support ABAC. How are access rules documented? How are required attributes identified and assigned? How are policies applied in a hierarchy and deconflicted? How are access failures handled? Who creates new policies? How are common policies shared?

Develop and Acquire an Interoperable Set of Capabilities. What standards and specifications apply to policies, attributes, and management of ABAC capabilities? How is interoperability measured and enforced? How are subject attribute capabilities integrated with identity management capabilities? How are diverse or special needs for identities handled? How are subject attributes shared and maintained? Is there any benefit to a central authentication, authorization, attribute management, decision, or enforcement capability? How are environment conditions used in access decisions? How is the confidence in security, quality, and accuracy measured, conveyed, and used in access decisions? How are subject attributes mapped between organizations? How are policies developed to incorporate the latest set of available subject, object, and environment condition attributes?

Operate with Efficiency. How are subject attributes managed for disconnected and disadvantaged users? How available are interface specifications for new participants to the enterprise? How is quality and timeliness of data measured and enforced? How is liability for data loss or misuse of data managed?

The remainder of this document provides a more detailed explanation of ABAC concepts and considerations for employment of enterprise ABAC capabilities. This document serves as the first in a series of access control publications designed to help planners, architects, managers, and implementers fulfill the information sharing and protection requirements of the U.S. Federal Government.

1. Introduction

1.1 Purpose and Scope

The purpose of this document is to provide Federal agencies with a definition of **Attribute Based Access Control** (ABAC) and considerations for using ABAC to improve information sharing while maintaining control of that information. This document describes the functional components of ABAC, as well as a set of considerations for employing ABAC within a large enterprise without taking into account Identity Management[1], thus assuming subjects are bound to trusted identities or identity providers. These considerations, although considered important, should not be deemed comprehensive. Before selecting and deploying an ABAC product or technology, the hosting organization should augment these considerations with testing and independent product reviews.

This document brings together many previously separate bodies of ABAC knowledge in order to bridge gaps between available technology and best practice ABAC implementations. ABAC implementations have tended to be inconsistent across organizations, so this document strives to provide guidelines that can be consistently applied throughout organizations. Given the broad depth of ABAC knowledge brought together in this document, this document can best be used as an informational guide for organizations that are considering to deploy, planning to deploy, or are currently deploying ABAC systems.

This document extends the information in NIST IR 7316, *Assessment of Access Control Systems* [NIST7316]; NIST IR 7665, *Proceedings of the Privilege Management Workshop* [NIST7665]; NIST IR 7657, *A Report on the Privilege (Access) Management Workshop* [NIST7657]; and NIST IR 7874 *Guidelines for Access Control System Evaluation Metrics* [NIST7874], which demonstrates the fundamental concepts of policy, models, and properties of AC systems.

1.2 Audience

This document assumes that readers are interested in understanding access control capabilities that utilize attributes to determine authorization decisions. These readers may also want to enhance the flexibility of access decisions without the need for a predetermined set of explicit privileges being defined between every subject (also known as a user) and every object (also known as a resource).

This document is intended to benefit and address the needs of two specific audiences:

- Persons who have a basic understanding of access control concepts and desire a general understanding of ABAC concepts

- Access control subject matter experts or managers experienced in access control concepts who are seeking detailed deployment or operational information on ABAC.

1.3 Document Structure

The rest of this document is divided into the following sections and appendixes:

- Section 2 provides a basic understanding of ABAC. It gives readers an overview of the current state of logical access control, a working definition of ABAC, and an explanation of core and

[1] See NIST SP 800-63-1 at http://csrc.nist.gov/publications/nistpubs/800-63-1/SP-800-63-1.pdf and NIST SP 800-63-2 at http://csrc.nist.gov/publications/drafts/800-63-2/sp800_63_2_draft.pdf.

enterprise ABAC concepts. Readers can gain a general understanding of ABAC concepts from just completing Section 2.

- Section 3 discusses ABAC enterprise employment considerations during the initiation, acquisition/development, implementation/assessment, and operations/maintenance phases. Readers with an interest in access control and/or project management will benefit most from this section.
- Section 4 contains a conclusion for the document.
- Appendix A provides an ABAC example.
- Appendix B defines various acronyms and abbreviations related to ABAC.
- Appendix C lists the references for the document.

Because of the constantly changing nature of the IT industry, readers are strongly encouraged to take advantage of other resources, including those listed in this document.

1.4 Notes on Terminology

The terminology used in this document is not meant to be authoritative, merely consistent within the confines of the document itself. Where possible, terminology that is used elsewhere within NIST publications and across the Federal Government was adopted to maintain consistency. Where terms were found to be used inconsistently or where multiple terms were being used throughout the Federal Government and the Identity and Access Control community to address a common concept, the simplest and most concise terms and definitions were used to explain ABAC concepts.

It is assumed that the reader understands the basic concepts of logical access control. That is, a logical **object**—sometimes referred to as a **resource**—has inherent value and must be protected by the object's owner from unauthorized use by others. The **subject** represents the entity requesting to perform the operation upon the object and is often called the **user** or **requestor**. Sometimes the subject is meant to be the logical representation of the user, in that the user does not actually access anything. It is, rather, a machine acting on behalf of the user that accesses and performs operations on the object. For the purposes of this document, it is assumed that the subject and user are synonymous, and the term **subject** is used throughout.

The subject is most often assumed to be a human; however, there is some debate over whether or not the subject must be human. Some contend that a **non-person entity (NPE)**, such as an autonomous service or application could fill the role of the subject. Others contend that every operation performed by a computer must be done on behalf of some person or organization with the authority to perform the operation. For the purposes of this document, the term **subject** is used to denote any entity (human or non-human) requesting access to an object and, for the sake of simplicity, is often referred to as a human person in the examples and illustrations.

There are traits or **attributes** about this person such as name, date of birth, home address, eye color, and social security number that may, either individually or when combined, comprise membership in a group or a unique **identity** that distinguishes that person from all others. These traits are often called **identity attributes** or **subject attributes.** The term **subject attributes** is used consistently throughout this document.

In the course of this person’s life, he or she may work for different organizations, within different roles, and may inherit different **authorities** tied to those roles. The person may establish different **personas** for each organization or role and amass different identity attributes related to each persona. For example, an individual may work for Company A as a Gate Guard during the week and may work for Company B as a shift manager on the weekend. The identity attributes and authorities are different for each persona and for each role. Although trained and qualified as a Gate Guard for Company A, while operating in her Company B persona as a shift manager on the weekend she does not have the authority to perform as a Gate Guard for Company B.

A subset of subject attributes are typically captured in a **credential**, a trusted token—something the subject has, knows, or is—that can be used to **authenticate** the subject or verify that the subject is who they say they are. **Authentication** can be performed using a variety of credentials—typically something that is unique to that subject (username and password, public/private key pair in a PKI, biometrics, or other unique characteristic that only the subject has, knows, or is). These credentials logically bind the subject to their digital identity, which is made up of subject attributes or information about that subject that, when put together, form a unique set of characteristics about that subject Credentials are usually issued by an authoritative agency and are bound to a unique identity or persona. Often, they contain a unique subject attribute or **unique identifier** along with additional subject attributes that help prove the identity of the bearer and the authority they have. Additionally, there are elements of the credential that prove its authenticity as belonging to a specific organization. Information about authentication can be found in NIST SP 800-63-1 at http://csrc.nist.gov/publications/nistpubs/800-63-1/SP-800-63-1.pdf and NIST SP 800-63-2 at http://csrc.nist.gov/publications/drafts/800-63-2/sp800_63_2_draft.pdf.

Authentication is not the same as access control or authorization. **Authentication** is the act of verifying that the subject performing an operation is actually who they say they are. **Access control** or **authorization**, on the other hand, is the decision (implicit or explicit) to permit or deny a subject access to a specific object (network, data, application, service, etc.) The terms **access control** and **authorization** are used synonymously throughout this document.

Privileges represent the authorized behavior of a subject; they are defined by an authority and embodied in **policy** or rules. For the purposes of this document, the terms **privileges**, **rights**, **authorizations**, and **entitlements** are essentially identical and are meant to convey one’s authority and implicit approval to access a resource. Many would argue that there are fundamental distinctions between each. Rights are inherent to every member of society (e.g., the right to life, liberty, and the pursuit of happiness). Privileges are granted for a specified time period or indefinitely by an authority and may be revoked (e.g., driving privileges given in a driver’s license). Authorizations are granted only when requested and for a specific timeframe (e.g., a work visa grants temporary authorization to work in a foreign country). Entitlements are attributes or tokens that represent predetermined authorization decisions that the subject may take with them to the point of enforcement (e.g., food stamps or a voter registration card).

Policy, rules, and relationships govern allowable behavior within an organization, based on the privileges of subjects and how resources or objects are to be protected under which environment conditions. Throughout this document, the term **policy** is used to convey these rules and relationships. Policy is typically written from the perspective of the **object** that needs protecting and the privileges available to subjects.

Like subjects, each object has a set of attributes that help describe and identify it. These traits are called **object attributes** and are sometimes referred to as **resource attributes**. This document uses the term **object attributes** consistently throughout. Object attributes are typically bound to their objects through reference, by embedding them within the object, or through some other means of assurance such as **cryptographic binding.**[2]

Information about policy, such as author, policy effective date, deconflict methods, etc. are sometimes called **meta-policy**. Information about attributes such as attribute authority, attribute creation date, etc. are sometimes called **meta-attributes**. Meta-policy and meta-attributes may be used in the development of policy sets and the identification of the appropriate attribute sets needed for authorization. A good example of the use of a meta-attribute is assigning an assurance level or **measure of confidence** to the attribute—a composite score for an attribute that could combine subjective ratings like a confidence score for the authority behind the attribute, a freshness score of the information in the attribute, and a level of accuracy score for how often the information is validated. At times, these measures of confidence may even be used as input to the access decision.

These policies must be enforced through some type of **access control mechanism**. The access control mechanism must assemble authorization information, which may include information about the object being protected, the subject requesting access, the policies governing access to the resource, and any contextual information needed to make a decision. By evaluating each policy element against the available information, the access control mechanism often employs a **policy decision point (PDP)** to render a decision, a **policy enforcement point (PEP)** to enforce the decision, and some sort of **context handler** or **workflow coordinator** to manage the collection of attributes required for the decision. For the purposes of this document, it is assumed that the term **access control mechanism** incorporates all of this functionality, and the term is used throughout.

[2] Cryptographic binding is a methodology for providing integrity and authenticity to data and data relationships using well-known cryptographic techniques. Cryptographic binding works by determining the hash value of each object attribute associated with a specific object and digitally signing the collection of hashed values. When the object is accessed, if the object signature fails, the attribute hash values are then compared to determine which element was modified since the last binding operation.

2. Understanding ABAC

Fully understanding ABAC requires understanding of the basic principles of logical access control. The purpose of logical access control is to protect objects—be they data, services, executable applications, network devices, or some other type of information technology—from unauthorized operations. These operations may include discovering, reading, creating, editing, deleting, and executing objects. These objects are owned by an individual or organization and have some inherent value that motivates those owners to protect them. As owners of the objects, they have the authority to establish a policy that describes what operations may be performed upon those objects, by whom, and in what context those subjects may perform those operations. In some cases, the owners are required to enforce a policy imposed upon them by higher authorities (Mandatory Access Control, MAC) and in others, the owners have the discretion to determine the policy themselves and can delegate this authority to others (Discretionary Access Control, DAC). If the subject satisfies the access control policy established by the object owner, then the subject is authorized to perform the desired operation on that object—better known as being granted access to the object. If the subject does not satisfy the policy, then it is denied access to the object.

Computer security architects and administrators deploy access control mechanisms (ACM) in logic aligned to protect their objects by mediating requests from subjects. These ACMs can use a variety of methods to enforce the access control policy that applies to those objects.

> ***Access Control Mechanism (ACM):*** *The logical component that serves to receive the access request from the subject, to decide, and to enforce the access decision.*

How these ACMs function can be described in terms of various logical access control models. These access control models provide a framework and set of boundary conditions upon which the objects, subjects, operations, and rules may be combined to generate and enforce an access control decision. Each model has its own advantages and limitations but it is important to note the evolution of these models to fully appreciate the flexibility and applicability of the ABAC model.

MAC/DAC

The earliest application of logical access control occurred in Department of Defense (DoD) applications in the 1960s and 1970s with the emergence of the concepts of Discretionary Access Control (DAC) and Mandatory Access Control (MAC). These terms are further defined in the DoD Trusted Computer System Evaluation Criteria (TCSEC) or "Orange Book" [TCSEC]. The definition of DAC and MAC can be also found in NIST SP 800-53 at http://csrc.nist.gov/publications/drafts/800-53-rev4/sp800_53_r4_draft_fpd.pdf.

IBAC/ACLs

As networks grew, the need to limit access to specific protected objects spurred the growth of identity based access control (IBAC) capabilities. IBAC employs the use of access control lists (ACLs) to capture the identities of those allowed to access the object. If a subject presents a credential that matches the one held in the ACL, the subject is given access to the object. Individual privileges of the subject to perform operations (read, write, edit, delete, etc.) are managed on an individual basis by the object owner. Each object needs its own ACL and set of privileges assigned to each subject. In the IBAC model, the authorization decisions are made prior to any specific access request and result in the subject being added to the ACL. For each subject to be placed on an ACL, the object owner must evaluate identity, object, and context attributes against policy governing the object and render a decision. This decision is static and a notification process is required for the owner to reevaluate and perhaps remove a subject from the ACL to represent subject, object, or contextual changes. Failure to remove or revoke access over time leads to users accumulating privileges, also known as authorization creep.

RBAC

In 1992, D.F. Ferraiolo and D.R. Kuhn published a paper that presented the Role-Based Access Control model (RBAC) [FK92]. RBAC employs the use of pre-defined roles that carry a specific set of privileges associated with them and to which subjects are assigned. For example, a subject assigned the role of Manager will have access to a different set of objects than someone assigned the role of Analyst. In this model, access is implicitly predetermined by the person assigning the roles to each individual and explicitly by the object owner when determining the privilege associated with each role. At the point of an access request, the access control mechanism evaluates the role assigned to the subject requesting access and the set of operations this role is authorized to perform on the object before rendering and enforcing an access decision. Note that a role may be viewed as a subject attribute that is evaluated by the access control mechanism and around which object access policy is generated. As the RBAC specification gained popularity, it made central management of enterprise access control capabilities possible and reduced the need for ACLs.

ABAC

In 2003, with the emergence of Service Oriented Architecture (SOA), a new specification was published through the OASIS standards body called Extensible Access Control Markup Language (XACML) [XACML]. The specification first presented the elements of what would come to be known as ABAC. The XACML model employs the use of elements such as rules, policies, rule- and policy-combining algorithms, attributes (subject, (resource) object, action and environment conditions), obligations, and advice. The reference architecture includes functions such as Policy Decision Points (PDPs), Policy Enforcement Points (PEPs), Policy Administration Points (PAPs), and Policy Information Points (PIPs) to control access. Furthermore, XACML provides a request/response protocol which can be used to mediate communications between the components.

Unfortunately, without a formal definition and implementation guidance, the user and technology communities started implementing ABAC solutions and defining new versions of advanced access control models based upon the XACML model without a common understanding or definition of ABAC. This document complements the OASIS XACML specification by providing a basic definition, concepts, and components that make up an ABAC model.

2.1 The Benefit of ABAC

Traditionally, logical access control solutions have been based primarily on the identity of a subject requesting execution of an operation (e.g., read) upon an object resource (e.g., a file). Examples include IBAC or RBAC where access to an object has been individually granted to a locally identified subject, or when access to an object has been granted to locally defined roles that the subject is a member of. This approach to access control is often cumbersome to manage. In this traditional (non-ABAC) multi-organizational access method example (illustrated below in Figure 3), authenticated access to resource objects outside of the subject’s originating organization would require the subject’s identity to be pre-provisioned in the target organization and pre-populated on an access list.

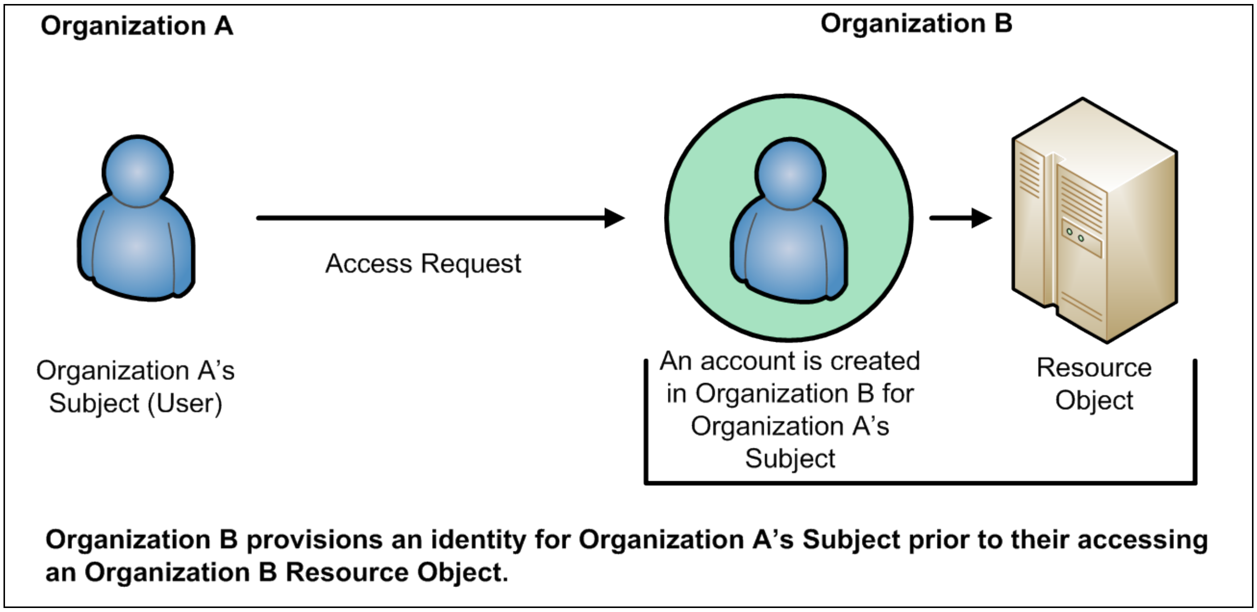

Organization A
Organization B
Access Request
Organization A's Subject (User)
An account is created in Organization B for Organization A's Subject
Resource Object
Organization B provisions an identity for Organization A's Subject prior to their accessing an Organization B Resource Object.

Figure 3: Traditional (Non-ABAC) Multi-Organizational Access Method

Additionally, the subject qualifiers, such as identity and roles, are often insufficient in the expression of real-world access control needs. RBAC makes a decision based on the subject's association with a role. RBAC does not easily support multi-factor decisions (for example, decisions dependent on rank, organization, physical location, and specialized training such as for Health Insurance Portability and Accountability Act (HIPAA) records access; recent training on HIPAA data protection may be a prerequisite to view medical records.) RBAC role assignments tend to be based upon more static organizational positions, presenting challenges in certain RBAC architectures where dynamic access control decisions are required.

A method is needed to make access control decisions without previous knowledge of the object by the subject or knowledge of the subject by the object-owner. By relying upon the concepts of subject and object attributes consistently defined between organizations, ABAC avoids the need for explicit authorizations to be directly assigned to individual subjects prior to a request to perform an operation on the object. Moreover, this model enables flexibility in a large enterprise where management of access control lists or roles and groups would be time consuming and complex.

Leveraging consistently defined attributes, authentication and authorization activities can be executed and administered in the same or separate infrastructures, while maintaining appropriate levels of security. For example, a subject can authenticate within a hospital's access management infrastructure, and then be able (or authorized) to access objects within the same or different hospital's access management infrastructure based upon their attribute values. It is not unusual to see subjects authenticating locally within one organization, and then securely accessing objects in a different organization, when appropriate organization-to-organization data sharing agreements and infrastructures are established.

2.2 A Working Definition of ABAC

ABAC has been described in various ways. For example, one early paper on web services states that ABAC "grants accesses to services based on the attributes possessed by the requester" [WWJ04], while a discussion of security in geographic information systems describes ABAC as an approach in which "attribute values associated with users determine the association of users with privileges" [CGLO09].

Still another paper summarizes ABAC as a model that is “based on subject, object, and environment attributes and supports both mandatory and discretionary access control needs” [YT05]. In these and other definitions, there is a reasonable consensus that ABAC determines access (i.e., operations upon system objects) by matching the current value of subject attributes, object attributes, and environment conditions with the requirements specified in access control rules. Thus, the following represents a high-level definition of ABAC:

Attribute Based Access Control (ABAC): *A logical access control methodology where authorization to perform a set of operations is determined by evaluating attributes associated with the subject, object, requested operations, and, in some cases, environment conditions against policy, rules, or relationships that describe the allowable operations for a given set of attributes.*

Attributes are characteristics that define specific aspects of the subject, object, environment conditions, and/or requested actions that are predefined and preassigned by an authority. Attributes are composed of an optional category that indicates the class of information given by the attribute, a name, and a value (e.g., Class=HospitalRecordsAccess, Name=PatientInformationAccess, Value=MFBusinessHoursOnly).

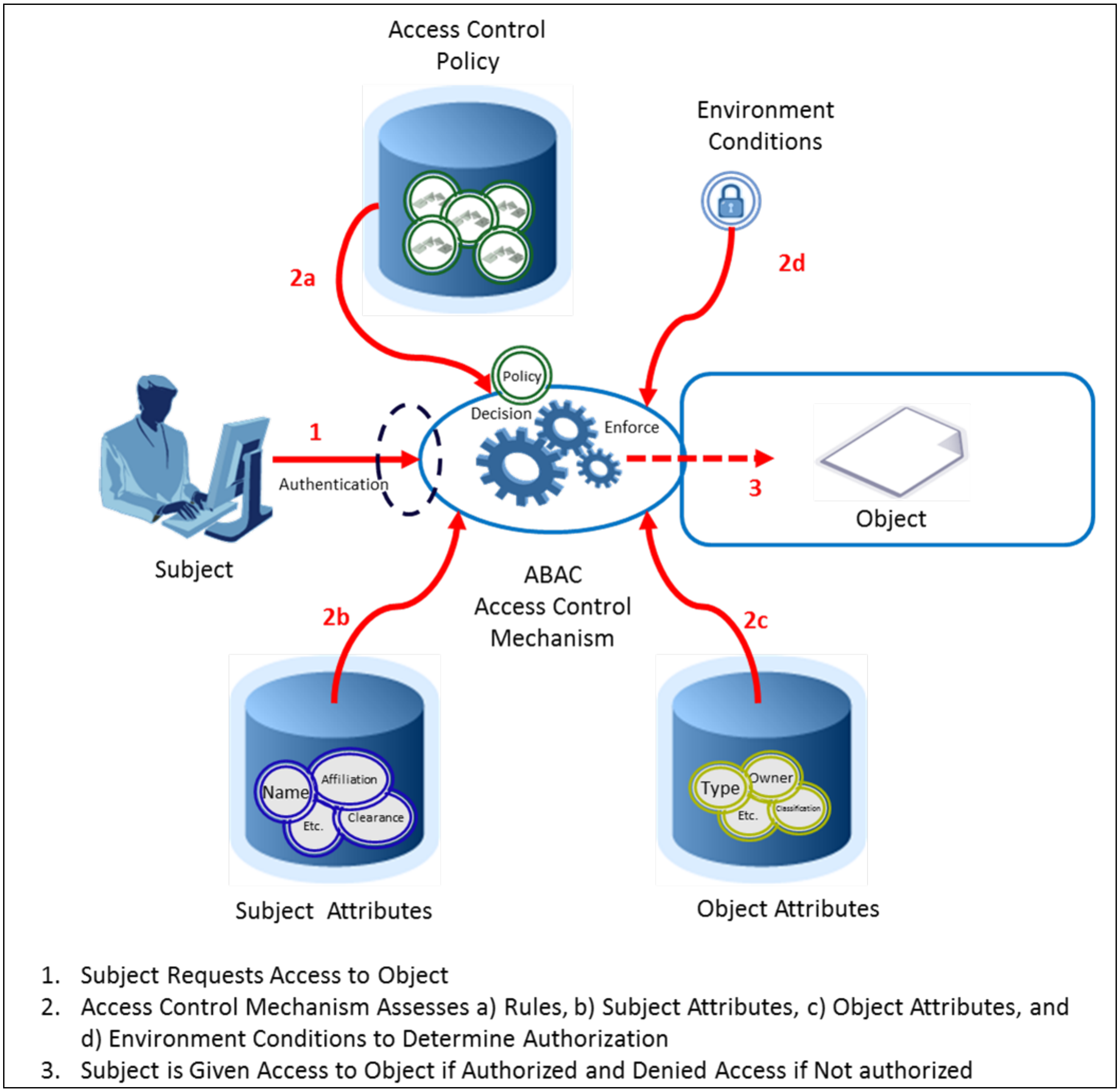
A **subject** is an active entity (generally an individual, process, or device) that causes information to flow among objects or changes the system state. It can be the user, requestor, or mechanism acting on behalf of the user or requestor. A subject may be a non-person entity such as a system or process, rather than a human. Subjects often act on behalf of a specific human or organization. Subjects may be assigned attributes that describe their name, organization affiliation, citizenship, etc.

An **object** is a passive information system-related entity (e.g., devices, files, records, tables, processes, programs, networks, domains) containing or receiving information. Access to an object implies access to the information it contains. It can be the resource or requested entity, as well as anything upon which an operation may be performed by a subject including data, applications, services, devices, and networks. Objects usually require some form of protection from unallowable operations by unauthorized subjects.

An **operation** is the execution of a function at the request of a subject upon an object. Operations include read, write, edit, delete, author, copy, execute, and modify.

Policy is the representation of rules or relationships that define the set of allowable operations a subject may perform upon an object in permitted environment conditions.

The high-level ABAC definition is visually depicted in the following diagram where the ABAC ACM receives the subject’s access request, then examines the subject’s and object’s attributes against a specific policy. The ACM then determines what operations the subject may perform upon the object.



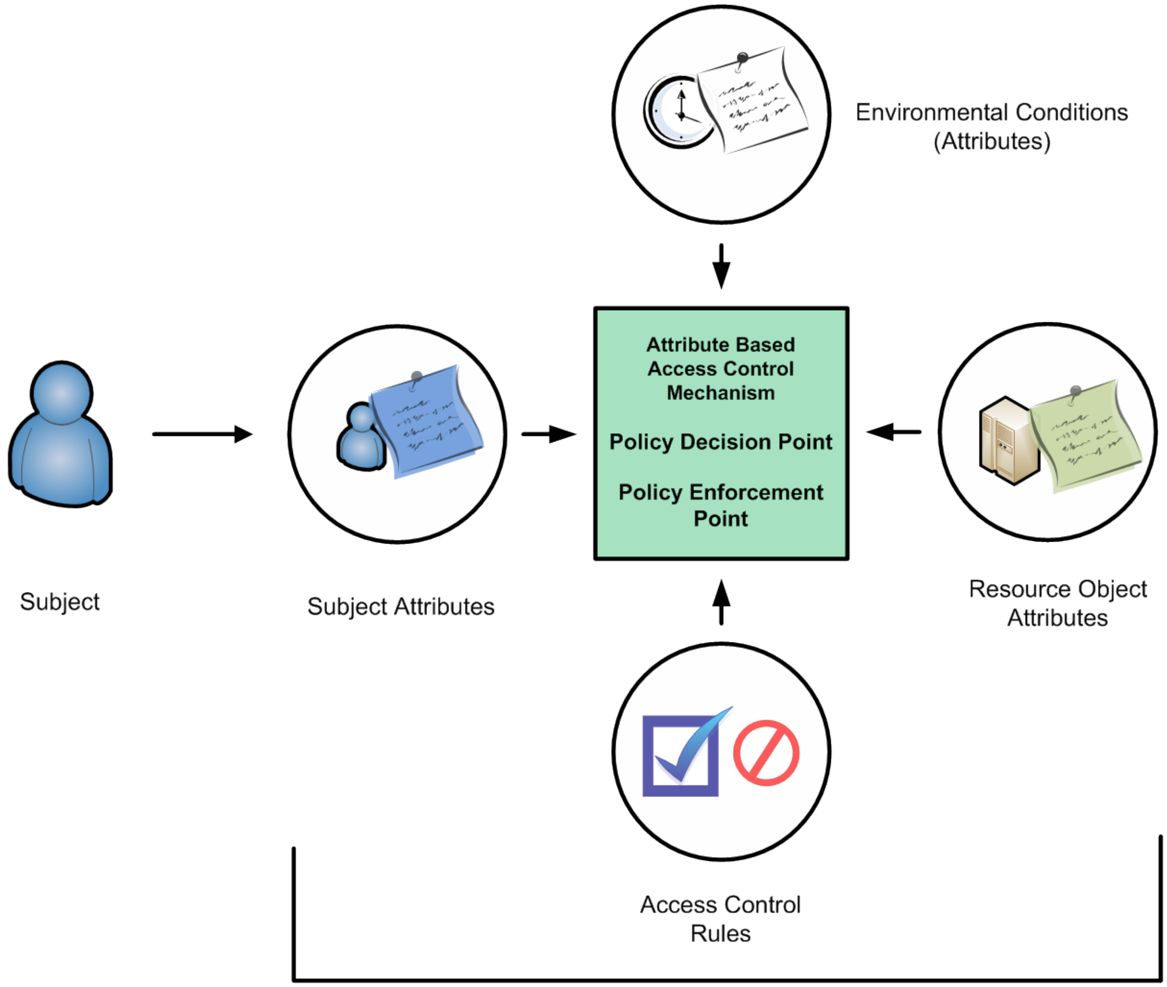
1. Subject Requests Access to Object
2. Access Control Mechanism Assesses a) Rules, b) Subject Attributes, c) Object Attributes, and d) Environment Conditions to Determine Authorization
3. Subject is Given Access to Object if Authorized and Denied Access if Not authorized

Figure 4: Basic ABAC Access Control Scenario

While a set of mature infrastructure capabilities is required for an enterprise ABAC implementation, Section 2.3 of this publication focuses on the rudimentary combination of subject attributes, object attributes, and policies within the access control mechanism. Section 2.4 introduces the fundamental functions needed for enterprise ABAC. Section 3 provides high-level guidelines aligned to the System Development Life Cycle (SDLC) on implementing ABAC for a large enterprise. Subsequent publications will explore the infrastructure needs for attribute management and policy management, give more detailed guidelines for enterprise ABAC implementation, and examine advanced complex implementations including hierarchical decisions, risk-based decisions, use of environment conditions in access decisions, and the use of measures of confidence to increase access decision assurance.

2.3 Core ABAC Concepts Explained

In its most basic form, ABAC relies upon the evaluation of attributes of the subject, attributes of the object, and the formal relationship or access control rule or policy defining the allowable operations for subject-object attribute combinations. All ABAC solutions contain these basic core capabilities to evaluate attributes and enforce rules or relationships between those attributes (see Figure 5 below).



When an access request is made, Attributes and Access Control Rules are evaluated by the Attribute Based Access Control Mechanism to provide an access control decision. In ABAC's basic form, the Access Control Mechanism contains both a Policy Decision Point, and a Policy Enforcement Point.

Figure 5: Core ABAC Concept

Even within a small isolated system, ABAC relies upon the assignment of subject attributes to subjects and object attributes to objects, and the development of policy that describes the access rules for each object. Each object within the system must be tagged or assigned specific object attributes that describe the object. For example, consider a document residing in a directory within a file management system. This document has a title, an author, a date of creation, and a date of last edit—all object attributes that are determined by the creator, author, or editor of the document. Additional object attributes may be assigned such as owning organization, intellectual property characteristics, export control classification, or security classification. Each time a new document is created or modified, these object attributes must be captured. These object attributes are often embedded within the document itself, but they may be captured in a separate table, incorporated by reference, or managed by a separate application.

Each subject that uses the system must be assigned specific subject attributes. Consider a user accessing a file management system. The user is established as a subject within the system by an administrator and characteristics about that user are captured as subject attributes. This subject has a name, a role, and an organization affiliation. Other subject attributes may include US Person status, nationality, and security clearance. These subject attributes are assigned and managed by an authority within the organization that can maintain the subject identity information for the file management system. As new users arrive, old users leave, and characteristics of subjects change, these subject attributes must be updated.

Every object within the system must have at least one policy that defines the access rules for the object. This policy is normally derived from documented or procedural rules that describe the business processes and allowable actions within the organization. For example, in a hospital setting, a rule may state that only approved medical personnel shall be able to access a patient's medical record. If a subject has a PersonnelTypeAttribute with a value of NonMedicalSupportStaff and they are trying to perform the operation Read upon a document with a RecordAttribute of PatientMedicalRecord, access will be denied and the operation will be disallowed.

The rules that bind subject and object attributes indirectly specify privileges (i.e., which subjects can perform which operations on which objects). Allowable operation rules can be implemented through many forms of computational language such as:

- A Boolean combination of attributes and conditions that satisfy the authorization for a specific operation, or

- Specified lists of attributes or similar methods of explicitly relating specific subjects to specific objects and the allowable set of operations.

Once object attributes, subject attributes, and policies are established, objects can be protected using ABAC. Access control mechanisms guard access to the objects by limiting access for allowable operations by allowable subjects. The ACM assembles the policy, subject attributes, and object attributes, then renders and enforces a decision based on the logic provided in the policy. ACMs must be able to manage the workflow required to make and enforce the decision, including determining what policy to retrieve, which attributes to retrieve in what order, and where to retrieve attributes. The ACM must then perform the computation necessary to render a decision.

The policies that can be implemented in an ABAC model are limited only to the degree imposed by the computational language. This flexibility enables the greatest breadth of subjects to access the greatest breadth of objects without having to specify individual relationships between each subject and each object. For example, a subject is assigned a set of subject attributes upon employment (e.g., Nancy Smith is a *Nurse Practitioner* in the *Cardiology Department.*). An object is assigned its object attributes upon creation (e.g., a folder with *Medical Records* of *Heart Patients).* The object owner creates an access control rule to govern the set of allowable operations (e.g., all *Nurse Practitioners* in the *Cardiology Department* can *View* the *Medical Records* of *Heart Patients).* Adding to the flexibility, attributes and their values may then be modified throughout the lifecycle of subjects, objects, and attributes.

Provisioning attributes to subjects and objects governed by a ruleset that specifies what operations can take place enables an unlimited number of subjects to perform operations on the object—all without prior knowledge of the specific subject by the object-owner or rule-maker. As new subjects join the organization, rules and objects do not need to be modified. As long as the subject is assigned the attributes necessary for access to the required objects (e.g., all Nurse Practioners in the Cardiology Department are assigned those attributes), no modifications to existing rules or object attributes are required. This benefit is often referred to as accommodating the external user and is one of the primary benefits of employing ABAC.

2.4 Enterprise ABAC Concepts Explained

While ABAC is a critical enabler of enterprise information sharing, when deployed across the scale of an enterprise, the set of capabilities required to implement ABAC gets more complex. At a system level the focus is on the access control mechanism and the logic within. At the enterprise level the increased scale requires complex and sometimes independently established management capabilities necessary to ensure consistent sharing and use of policies and attributes and the controlled distribution and employment of access control mechanisms throughout the enterprise. The following represents a definition of enterprise for this document.

> ***Enterprise:*** *Collaborated or federated organizations, or a single organization with multiple operational units, which require sharing of information to perform business operations.*

Figure 6 below presents a high-level representation of the major components required to enable enterprise ABAC capabilities. Most enterprises have existing capabilities that can be leveraged to complete this picture. For example, most enterprises have some form of identity and credential management to manage population of subject attributes, such as name, unique identifier, role, clearance, etc. Similarly, many enterprises may have some form of policy management to establish and apply rules authorizing subjects' access to enterprise objects. However, these rules are often documented in human-readable form and hard-coded into individual applications; they are usually not written in a machine-readable format. For enterprise ABAC to achieve its full potential, digital policies must be made available in machine-readable format, then stored in repositories and published for ACM consumption. From these digital policies, subject and object attributes required to fully render access control rules can be identified. These enterprise subject attributes must be created, stored, and shared across organizations within the enterprise through a subject attribute management capability. Likewise, enterprise object attributes must be established and bound to the objects they define through an object attribute management capability. Finally, ABAC-enabled access control mechanisms must be deployed or provided as an enterprise service to protect enterprise objects. The remainder of this section provides more detail on each of these major components of enterprise ABAC.

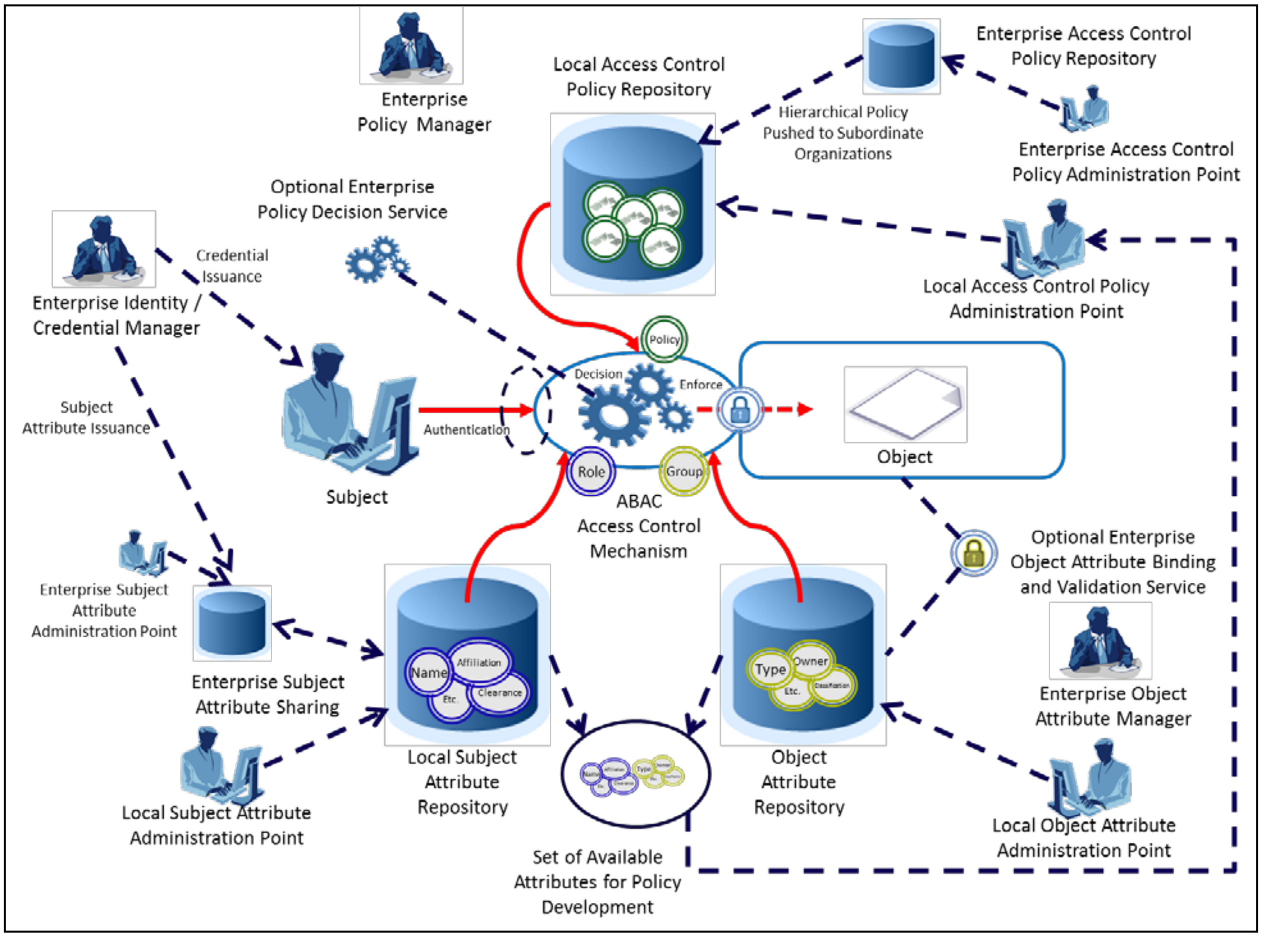

Enterprise Policy Manager
Local Access Control Policy Repository
Enterprise Access Control Policy Repository
Hierarchical Policy Pushed to Subordinate Organizations
Enterprise Access Control Policy Administration Point
Optional Enterprise Policy Decision Service
Credential Issuance
Enterprise Identity / Credential Manager
Local Access Control Policy Administration Point
Policy
Decision
Enforce
Subject Attribute Issuance
Authentication
Subject
Object
Role
Group
ABAC Access Control Mechanism
Optional Enterprise Object Attribute Binding and Validation Service
Enterprise Subject Attribute Administration Point
Enterprise Subject Attribute Sharing
Name
Affiliation
Etc.
Clearance
Type
Owner
Etc.
Classification
Enterprise Object Attribute Manager
Local Subject Attribute Repository
Object Attribute Repository
Local Subject Attribute Administration Point
Local Object Attribute Administration Point
Set of Available Attributes for Policy Development

Figure 6: Enterprise ABAC Scenario

2.4.1 Policy Use in Enterprise ABAC

First, examine policy or the rules or relationships that define allowable operations for subject and object pairs.

Natural Language Policies (NLPs) are high-level requirements that specify how information access is managed and who, under what circumstances, may access what information. NLPs are expressed in human understandable terms and may not be directly implementable in an ACM. While NLPs can be application-specific and thus taken into consideration by the application vendor, NLPs are just as likely to pertain to subject actions within the context of enterprise policies. For instance, NLPs may pertain to object usage within or across organizational units or may be based on need-to-know, competence, authority, obligation, or conflict-of-interest factors. Such policies may span multiple computing platforms and applications. Therefore, NLPs are defined as follows:

> ***Natural Language Policy (NLP):*** *Statements regarding the managing and accessing of enterprise objects. NLPs are abstract concepts that can be translated to machine-enforceable access control rules.*

Given that all relevant NLPs are comprehensive and exist for each organization in an enterprise, the next step is to translate those into a common set of rules that can be enforced equally and consistently within the ACMs across the enterprise. In order to accomplish this, it is necessary to identify all required subject-object attribute combinations and allowable operations. Often these values will vary from organization to organization and may require some form of consensus or mapping to each organization's existing attributes to accommodate enterprise interoperability. The agreed-upon list of subject and object attributes, the allowable operations, and all mappings from existing organization-specific attributes are then translated into machine-readable format.

NLPs are required to codify into Digital Policy (DP) algorithms or mechanisms. For efficiency of performance and simplicity in specification, an NLP may require to be decomposed and translated into different versions of DPs that suit the infrastructure of operation units in the enterprise. Thus in the implementation of NLP, DPs are defined as:

> ***Digital Policy (DP):*** *Contains access control rules or other DPs that compile directly into machine executable codes or signals such as access control language. Subject and object attributes are the fundamental elements of DP, the building blocks of DP rules, which are then enforced by an access control mechanism.*

These different versions of DPs may then require Meta Policies (MPs), or policies dictating the use and management of DPs to handle DP hierarchical authorities, DP deconfliction, and DP storage and updates. Thus, MPs are used for managing DPs. Depending on the level of complexities, hierarchical MPs may be required based on the structures for the priority and combination strategies specified by NLP. Thus, in the usage of NLP and DPs, an MP is defined as:

> ***Meta Policy (MP):*** *Regulates how to assign priorities and mediate conflicts between DPs or other MPs. An MP is a policy about policies, or policy for managing polices.*

Once DPs and MPs are developed they need to be managed, stored, validated, updated, deconflicted, shared, retired, and enforced. Each of these operations requires a set of capabilities that will often be distributed across the enterprise and may be termed Digital Policy Management (DPM). There may be multiple policy authorities and hierarchies within organizations that will have variations on enterprise policy. Common enterprise policies should be shared by an authoritative source while subordinate policies should be managed locally. The rules for how DPs are managed should be determined by a central authority like an Enterprise Policy Manager.

Proper DP definition and development are critical to the identification of subject and object attributes that are needed to render an access control decision. Remember that a DP statement is comprised of the subject and object attribute pairings as well as environment conditions needed to satisfy a set of allowable operations. Once the full set of subject and object attributes needed to satisfy the entire set of allowable operations for a given set of enterprise objects is identified, this set of attributes comprises the entire set of attributes needed to be defined, assigned, shared, and evaluated for enterprise ABAC access decisions. For this reason, identifying the NLP and DP must be accomplished first when implementing an enterprise ABAC capability. Additional considerations for management of digital policy can be found in Section 3 of this document.

2.4.2 Attribute Management in Enterprise ABAC

Next, consider the lists of attributes developed while examining the NLPs and DPs. Without a sufficient set of object and subject attributes, ABAC does not work. Attributes need to be named, defined, given a set of allowable values, assigned a schema, and issued to subjects and objects. Subject attributes need to be established, issued, stored, and managed under an authority. Object attributes need to find a way to be bound to the objects they describe. Attributes shared across organizations need the ability to be published, validated, assured, updated, modified, and revoked.

Subject attributes are provisioned by attribute authorities—typically authoritative for the type of attribute that is provided and managed through an attribute administration point. Often, there are multiple authorities—each authoritative over a different attribute. For example, Security might be the authority for Clearance attributes, while Human Resources might be the authority for Name attributes. Subject attributes that need to be shared to allow subjects from one organization to access objects in another organization must be consistent, comparable, or mapped to allow equivalent policies to be enforced. For example, a member of Organization A with the role Job Lead wants to access information in Organization B, except Organization B uses the term Task Lead to denote the equivalent role. This problem also applies to mapping between an enterprise attribute schema and an application-specific schema, particularly ones built before the enterprise schema is defined and/or COTS products that come with their own schema built in. Organizations must normalize subject attribute names and values, or maintain a mapping of equivalent terms for all organizations. This should be managed by a central authority like an Enterprise Identity/Credential Manager.

Object attributes need to be established, maintained, and assigned to objects as objects are created or modified. While it may not be necessary to have a common set of object attributes in use across the enterprise, object attributes should be consistently employed to fulfill enterprise policy requirements, and available sets of object attributes should be published for those wishing to mark, tag, or otherwise apply object attributes to their objects. At times, it might be necessary to ensure that object attributes are not tampered with or altered to satisfy an access request. Objects can be cryptographically bound to their object attributes to identify whether objects or their corresponding attributes have been inappropriately modified. Mechanisms must be deployed to ensure that all objects created are assigned the appropriate set of object attributes to satisfy the policy being employed by the ACM. It may be necessary to have an Enterprise Object Attribute Manager to coordinate these requirements.

In the course of managing attributes, the concept of “meta attributes”—or characteristics of attributes—arises. Meta attributes apply to subjects, operations, objects, and environment conditions as extended attribute information useful for enforcing more detailed policy that incorporates information about the attributes and for managing the volumes of data needed for enterprise attribute management. Thus, meta attributes can simply be stated as:

Meta Attributes: *Data descriptors about attributes that are necessary to implement MP and DP processing within an ACM.*

Additional considerations for attribute management can be found in Section 3 of this document.

2.4.3 Access Control Mechanism Distribution in Enterprise ABAC

Finally, consider the distribution and management of ACMs throughout the enterprise. Depending on the needs of the users, size of the enterprise, distribution of the resources, and sensitivity of the objects that need to be accessed or shared, the distribution of ACMs can be critical to the success of an ABAC implementation. The functional components of an ACM may be physically and logically separated and distributed within an enterprise rather than co-located as described in the system-level view of ABAC.

Within the ACM are several functional "points" that serve as the service node for retrieval and management of the policy, along with some logical components for handling the context or workflow of policy and attribute retrieval and assessment. The Figure 7 example shows the main functional points: the Policy Enforcement Point (PEP), the Policy Decision Point (PDP), the Policy Information Point (PIP), and the Policy Administration Point (PAP). When these components are in an environment together they must function in harmony to provide access control decisions.

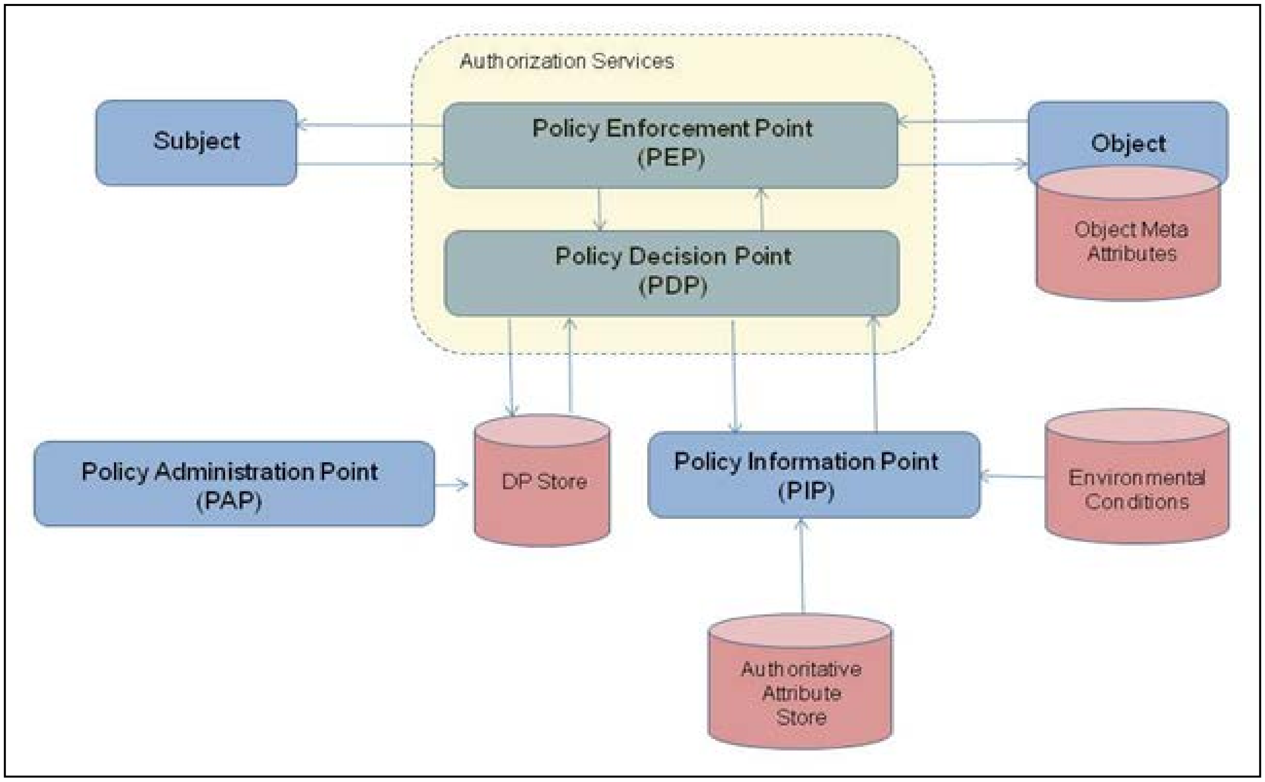

Authorization Services
Subject
Policy Enforcement Point (PEP)
Object
Object Meta Attributes
Policy Decision Point (PDP)
Policy Administration Point (PAP)
DP Store
Policy Information Point (PIP)
Environmental Conditions
Authoritative Attribute Store

Figure 7: An Example of ACM Components

A PDP performs an evaluation on DPs and MPs in order to produce an access control decision. Therefore, it can be stated that:

> ***Policy Decision Point (PDP):*** *Makes the access decisions by evaluating the applicable DPs and MPs. The PDP implements the decision procedures according to the ACM's computational languages. One of the main functions of the PDP is to mediate or deconflict DPs according to MPs.*

The next function to perform within these components is to enforce these decisions made by the PDP. This role belongs to the PEP. The PEP can be defined as

> ***Policy Enforcement Point (PEP):*** *The logical entity or place on a server that enforces policies for authorization and policy decisions in response to a request from a subject wanting to access a protected object; it executes the appropriate access decisions, as determined by the PDP, which shall either allow or deny user access to the requested protected object.*

The PDP and PEP may be physically and logically separated in an enterprise. For example, an enterprise could establish a centrally controlled enterprise decision service that evaluates attributes and policy and renders decisions which are then passed to the PEP as assertions. This allows for central management and control of subject attributes and policy, but grants partial control of access to the object from the local object owner. The design and distribution of ACM components requires a management function to ensure coordination of ABAC capabilities.

In order for the PDP and the PEP to perform their roles, they must be able to have information about the attributes and the policies to be enforced. These functions are performed by the PIP.

Policy Information Point (PIP): *Serves as the retrieval source of attributes, or the data required for policy evaluation to provide the information needed by the PDP to make the decisions.*

Before these policies can be enforced, they must be thoroughly tested and evaluated to ensure they meet the intended need. This action is carried out by the PAP.

Policy Administration Point (PAP): *Provides a user interface for creating, testing, and debugging MPs, and storing these policies in the appropriate repository.*

Finally, as a recommended additional component within the ACM, the Context Handler manages the order in which policy and attribute retrieval and assertion is performed. This is most crucial when time critical or disconnected access control decisions must be made. For example, attributes may be retrieved in advance of an access request, or cached to avoid the delay inherent in retrieval and assertion at the time of the access request. The Context Handler also coordinates with PIPs to add attribute values to the request context, and converts authorization decisions in the canonical form (e.g., XACML) to the native response format.

Context Handler: *Executes the workflow logic that defines the order in which policy and attributes are retrieved and enforced.*

3. ABAC Enterprise Employment Considerations

Many factors must be considered before deploying an ABAC system across an enterprise. This section attempts to consolidate available guidelines based on the state of the technology to date and lessons learned through multiple attempts within the Federal Government to deploy ABAC capabilities throughout a large enterprise. The guidelines are presented according to the phases of the NIST System Development Life Cycle (SDLC) illustrated in Figure 8. For more general information regarding the definitions of the phases and expected outputs, please refer to NIST SP 800-100: *Information Security Handbook: A Guide for Managers*. Most considerations for employment of enterprise ABAC fall within the first four phases: Initiation, Acquisition/Development, Implementation/Assessment, and Operations/Maintenance. As such, this section focuses on those phases exclusively.

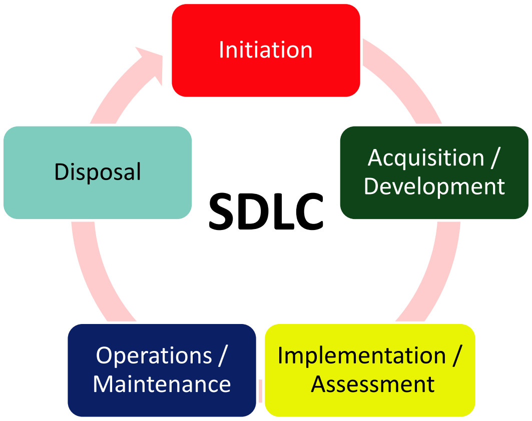


Figure 8: ACM NIST System Development Life Cycle (SDLC)

The development and deployment of an enterprise ABAC capability requires the careful consideration of a number of factors that will influence its design, security, and interoperability. These factors can be summarized around a set of principles:

- **Establish the Business Case for ABAC Implementation.** What are the costs of developing/acquiring new capabilities and transitioning away from old capabilities? What are the hidden costs of risk exposure, the new governance structures required to manage shared capabilities, and documenting policies that were previously human-in-the-loop decisions? How are privileges managed, monitored, and validated for compliance? Which datasets, systems, applications, and networks need ABAC capabilities?

- **Understand the Operational Requirements and Overall Enterprise Architecture.** What objects will be exposed to the enterprise for information sharing? What ACM will be used? How will subject attributes be shared? How will object attributes be used consistently? What are the access control rules and how are they captured, evaluated, and enforced? How is trust managed within the enterprise?

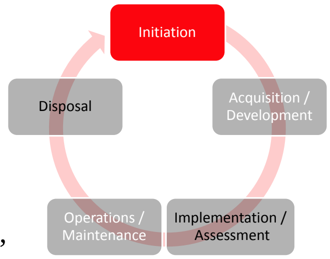
- **Establish or Refine Business Processes to Support ABAC.** How are access rules documented? How are required attributes identified and assigned? How are policies applied in a hierarchy and deconflicted? How are access failures handled? Who creates new policies? How are common policies shared?

- **Develop and Acquire an Interoperable Set of Capabilities.** What standards and specifications apply to policies, attributes, and management of ABAC capabilities? How is interoperability measured and enforced? How are subject attribute capabilities integrated with identity management capabilities? How are diverse or special needs for identities handled? How are subject attributes shared and maintained? Is there any benefit to a central authentication, authorization, attribute management, decision, or enforcement capability? How are environment conditions used in access decisions? How is confidence in security, quality, and accuracy measured, conveyed, and used in access decisions? How are subject attributes mapped between organizations? How are policies developed to incorporate the latest set of available subject, object, and environment condition attributes?

- **Operate with Efficiency.** How are subject attributes managed for disconnected and disadvantaged users? How available are interface specifications for new participants to the enterprise? How is quality and timeliness of data measured and enforced? How is liability for data loss or misuse of data managed?

The following sections address these principles and questions in more detail.

3.1 Considerations During the Initiation Phase



During the initiation phase, the organization establishes the need for a particular system and documents its purpose. It is often determined whether the project will be an independent information system or a component of an already-defined system. Once these tasks have been completed and a need has been recognized for a new or enhanced capability, several processes must take place before the project is approved, to include clearly defining project goals and defining high-level requirements. Typically, during this phase, the organization defines high-level business and operational requirements as well as the enterprise architecture.

3.1.1 Building the Business Case for Deploying ABAC Capabilities

As with any major system deployment, the deployment of enterprise ABAC capabilities should be preceded by significant requirements evaluation, trade studies, and planning activities to include the determination of whether ABAC is the right type of access control capability needed and feasible given the application portfolio. Before any technical requirements are generated or deployment decisions are made, it is important to evaluate and establish a business case for the deployment of ABAC capabilities as well as define the scope of the enterprise targeted for these capabilities. Enterprise ABAC carries with it significant development, implementation, and operations costs as well as a paradigm shift in the way enterprise objects are shared and protected. It may be more practical to take an incremental approach and implement ABAC protections for a limited set of well-understood objects. This implementation would establish and utilize, to the maximum extent possible, policies and attributes appropriate for the enterprise as a whole. Feedback from incrementally building out this ABAC capability will refine policy and attribute definitions and exercise the governance and configuration management capabilities necessary to support broader ABAC use throughout the enterprise. It should be noted that without addressing the issues presented in the following subsections, an enterprise will incur significant delay and cost in its ABAC deployment.

3.1.2 Scalability, Feasibility, and Performance Requirements

Scalability, feasibility, and performance are important considerations when choosing the deployment of an ABAC product or technology. When ABAC is implemented within a single operating environment (e.g., operating system, database management system) all of the requisite components are typically found within that environment, well within the network and system boundaries. Enterprise ABAC—allowing an organization within an enterprise to have unimpeded access to authorized resources owned and possessed within another organization within the same enterprise—requires a complex level of interaction between ABAC components. Often these components are distributed throughout the enterprise across organization boundaries and sometimes on different networks. The larger and more diverse the enterprise, the more complex these interactions become, forcing what may have been a simple request to access a document within a repository to now require a policy request from an enterprise service, multiple attribute assertions from numerous logically and geographically dispersed attribute sources, a third-party validation of the integrity of the object attributes bound to the document, and a decision made at one point in the enterprise while the enforcement of that decision is performed at a completely different point within the enterprise. Feasibility evaluation checks application support of ABAC, for some applications might not be able to support ABAC (or might be able to only by using a third-party plug-in). All of these potential interactions have a performance cost that must be evaluated when determining the scope of potential objects that will be shared through an enterprise ABAC implementation. To mitigate potential performance and scalability concerns, it is best to deploy PDPs in close proximity to PEPs. In addition to minimizing network latency, enterprises should only distribute relevant policies and policy sets to PDPs.

3.1.2.1 Budget for Development vs. Budget for Maintenance

While ABAC provides many important new features when deployed across an enterprise, the cost of development, deployment, and maintenance of ABAC components is significant and may not provide cost savings over existing solutions in the long term. In addition, the cost of retrofitting applications to use ABAC is wholly separate from procuring, setting up, and maintaining an authorization infrastructure. While cost savings can be incurred through no longer having to maintain existing solutions, it is suspected that a large portion of that maintenance cost will be offset by the cost of managing and maintaining subject attributes and the policies needed for ABAC, as well as additional system support required. The benefits of having more fine-grained,[3] consistent, and flexible security must be quantified and used to determine the right balance between cost of risk and cost of security. Given these considerations, ABAC is not the right solution for every logical access control problem and should be applied only when needed for requirements such as fine-grained control of objects, ability to provide access without prior knowledge of or information about the subject, and large-scale enterprise information sharing of a limited set of mission or business critical objects.

3.1.2.2 Cost of the Paradigm Shift

For many organizations, resources are often protected solely by network access privileges—where access to the network equates to access to network resources. Other resources within the same organization may

[3] Fine-grained access control allows for a larger number of discrete inputs into an access control decision, providing a larger set of possible combinations of those variables to reflect a larger and more definitive set of possible rules, policies, or restrictions on access. ABAC allows an unlimited number of attributes to be combined to satisfy any access control rule imaginable. As long as the attributes are available to evaluate at the time the access decision is rendered, the rule can be as complex and definitive as it needs to be to satisfy the protection requirements of the object. Thus, fine-grained AC allows access to be more detailed or flexibly partitioned when compared with coarse-grained AC, for example: coarse: employees can read file X, fine: employees working on project A can read file X, and finer: employees working on project A during office hours can read file X.

employ group policies where roles and rudimentary policies or stovepiped authentication protections such as IBAC or RBAC dictate access. The vast user population is accustomed to the business processes related to these legacy access control methods. The governance and business process changes that must accompany the shift to ABAC represent a significant paradigm shift from a model where resources are controlled and protected by the local owner, to one where resources are exposed to the enterprise and controlled and protected by enterprise-governed rules and enterprise-controlled attributes, and sometimes local control as well. New enterprise resources are no longer solely the responsibility of the creator, but adopt the significance of the organization from which they are being exposed. These resources must now adhere to an additional set of interoperability and quality specifications that have not needed to be defined until now. Users accustomed to logging onto their network and having unlimited access to resources will no longer have that luxury. While policy makers will do their best to reflect current mission and business needs in policies, there will be unexpected but inevitable denials of access to those with critical mission or business functions.

As ABAC products are implemented and an organization's access control paradigms shift, new processes and capabilities will need to be integrated into the users' day-to-day business processes. During the transition it will be important to ensure that users understand why these access control changes are being implemented and what impact they will have on the way business is done. These users will need to be trained in the new ABAC systems and processes. These changes need to be properly communicated to show the benefits of an enhanced user experience, the enhanced security and safeguarding of critical information, the requirements of the new ABAC system, and the legacy access control systems, if replaced, that will be phased out. Users may be comfortable with existing processes and may not see an immediate value in switching to an ABAC capability. It will be important to emphasize areas in which ABAC enhances the security posture of the enterprise as well as areas where it can function as not necessarily a replacement but as a complement to existing access control mechanisms.

3.1.2.3 Need to Review Privilege and Monitor Authorizations

A desired feature of many enterprises is the ability to review the capabilities associated with subjects and their attributes and the access control entries associated with objects and their object attributes. More succinctly, there are some requirements to know what access each individual has before the requests are made. This is sometimes referred to as "before the fact audit" . Before the fact audit is often necessary to demonstrate compliance to specific regulations or directives. A concept that is closely related to being able to review the capabilities of a subject is data discovery. When an object is provisioned, how do subjects become aware of the fact that they can now access that object? Another commonly desired review feature is determining who has access to a particular object or to the set of resources that are assigned to a particular object attribute. ABAC does not lend itself well to efficiently conducting these audits. Rather, a key feature of ABAC is the ability of the object owner to protect and share the object without any prior knowledge of individual subjects. Evaluating the set of subjects that have access to a given object requires a significant data retrieval and computation effort—essentially requiring every object owner to run a simulation of the access control request for every known subject in the enterprise. Limiting the scope of ABAC implementation can help in predetermining access authorizations, but other methods of ensuring the validity of access authorizations should be explored if the enterprise requires such validation.

Additionally, enterprise authorization services should be tightly integrated with security audit, data loss prevention, security configuration management, continuous monitoring, and cyber defense capabilities. Authorization services alone are not enough to ensure the security needed to protect the mission-critical objects resident on the networks. Comprehensive and cohesive enterprise security capabilities are needed to establish the desired level of assurance, and they must be tightly integrated to seamlessly feed the security information needed for making security decisions. Efforts should be undertaken to fully understand enterprise security requirements and the impacts an ABAC implementation will generate. For example, when using a distributed ACM architecture there are consequences to the ability to centrally audit access control decisions.

3.1.2.4 Maturity and Type of Rules to Enforce

Within the various operating environments of an enterprise there are a number of different operation and object types over which policy needs to be enforced. These operating environments may include operating systems, applications, data services, and database management. While some NLPs may exist to help determine authorized access, access to most objects is controlled through local group policy governed by local business rules, undocumented evaluation factors, and inherited non-standard doctrine. Implementing ABAC requires, first and foremost, a thorough understanding of the objects and their protection requirements. Without that understanding, the cost to develop and implement the technology required for enterprise ABAC increases dramatically. It is recommended that enterprise ABAC implementations be initially applied to mission or business critical objects that are well defined, controlled, and documented.

3.1.2.5 Enterprise Governance and Control

Successful enterprise ABAC requires the centralized coordination and determination of several business process and technical factors as well as establishment of enterprise responsibilities and authorities. Without the proper governance model in place, organizations will develop stovepiped solutions and enterprise interoperability will be delayed significantly. It is recommended that an enterprise governance body be formed to manage all identity, credential, and access management capability deployment and operation and that each subordinate organization maintain a similar body to ensure consistency in managing the deployment and paradigm shift associated with enterprise ABAC implementation. Additionally, it is recommended that the centralized governance body develop a “trust model” that can be used to illustrate the trust chain and help determine ownership and liability of information and services, needs for additional policy and governance, and requirements for technical solutions to validate or enforce trust relationships. The trust model can be used to help influence organizations to share their information with clear expectations of how that information will be used and protected and to be able to trust the information and attribute and authorization assertions coming from other organizations.

When managing the risk inherent in information sharing, two perspectives of risk must be addressed when deploying an enterprise ABAC solution. First, an ABAC solution may be considered one of many security controls that help protect an enterprise from risk. Second, use of ABAC capabilities may increase or decrease operational risk of an enterprise by exposing protected resources to access by unknown entities. By assuming that attributes are issued appropriately, the true access decision is being made by the attribute-issuing authorities, not the object owner. This deferral of risk and shared liability presents a number of challenges that must be managed through governance and a formal trust model.

A comparison of representative trust chains for legacy ACL use and ABAC use (Figures 9 and 10) shows that there are many more complex trust relationships required for ABAC to work properly. Ignoring the commonalities in both diagrams, one can observe that with ACLs the root of trust is with the object owner, who ultimately enforces the object access rules by provisioning access to the object through addition of a user to an ACL. In ABAC, the root of trust is derived from many sources of which the object owner has no control, such as Subject Attribute Authorities, Policy Developers, and Credential Issuers.

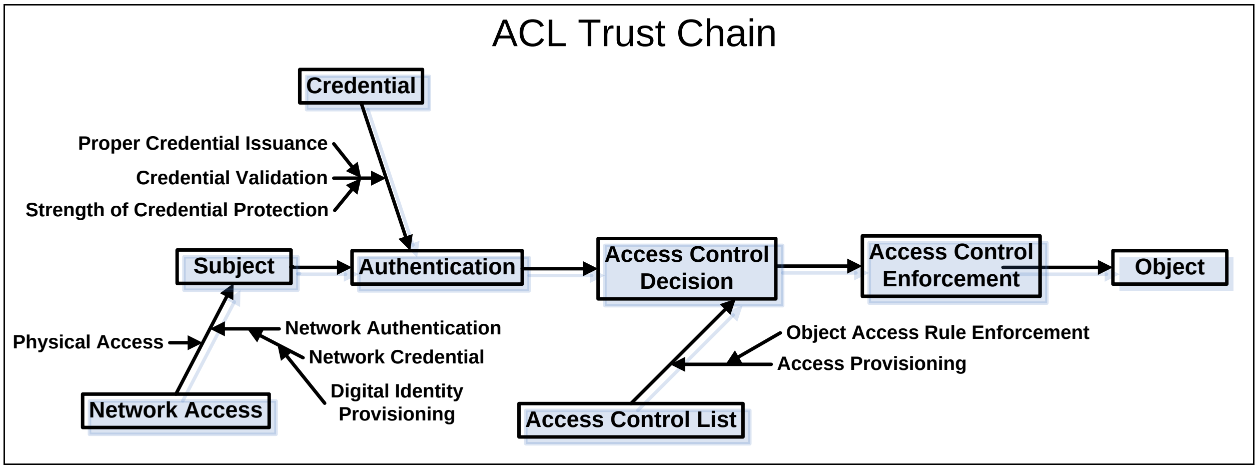


Figure 9: ACL Trust Chain

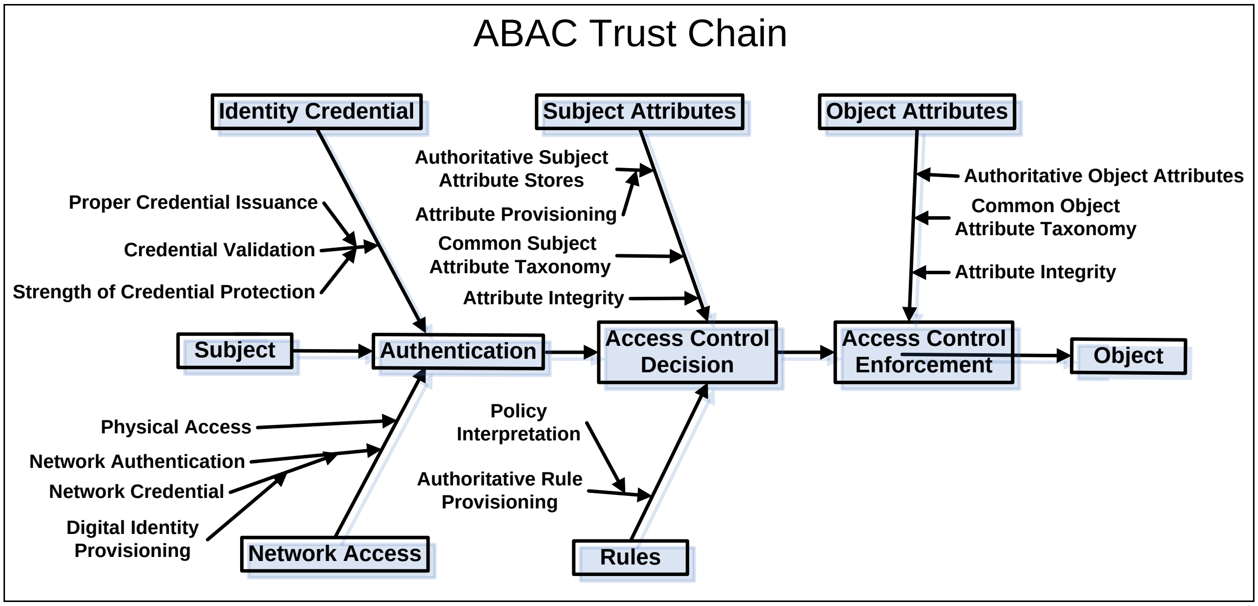


Figure 10: ABAC Trust Chain

When establishing a governance model for managing the risks inherent in ABAC, it is important to ensure there are mechanisms and agreements in place with each responsible organization to monitor and manage these roots of trust and any liabilities that occur as a result of unwarranted access.

3.1.3 Developing Operational Requirements and Architecture

Several high-level operational and architecture planning requirements must be satisfied before implementing an ABAC solution.

- First, identify the objects that will be shared and protected by ABAC, and define the rules or policies that govern their protection.

- Second, identify and formally define the subject and object attributes in coordination with the access control rule developers.

- Third, develop processes regarding how the access control policies are written, validated, and managed.

- Finally, determine how the ACMs will be segmented or distributed throughout the enterprise and how attribute, policy, and decision requests and responses will be rendered.

3.1.3.1 Identification of the Objects that Will Be Shared through ABAC

The objects selected to be shared and protected by the ABAC solution will vary based upon organizational requirements. Each object or class of object must be identified and the policy or rules protecting each must be documented in NLP. A set of business processes need to be established to identify, class, and assign policy to each new object created within the scope of the ABAC implementation.

3.1.3.2 Attribute Architecture

All attributes, whether subject or object, must be established, defined, and constrained by allowable values. The schema for these attributes and allowable attribute values must be published to all participants to help enable object owners with rule and relationship development. Once attributes and allowable values are established, methods for provisioning attributes and appropriate attribute values to subjects and objects need to be established as well as an architecture for any attribute repositories, retrieval services, or integrity checking services. Interfaces and mechanisms must be developed or adopted to enable sharing and authoritative assertion of these attributes.

3.1.3.3 Subject Attributes

Many subject attributes are typically provisioned upon employment with the organization and may be provisioned by several different authorities (human resources, security, organization leadership, etc.) For these, approaches to obtaining authoritative data are well known. As an example, only security authorities should be able to provision and assert clearance attributes and attribute values based on authoritative personnel clearance information; an individual should not be able to alter his own clearance attribute value. Other subject attributes may involve the subject’s current tasking, physical location, and the device from which a request is sent; processes need to be developed to assess and assure the quality of such subject attribute data.

Authoritative subject attribute provisioning and assertion capabilities should be appropriately dependable in regards to quality, assurance, and service expectations if they are to be relied upon for access control decisions. These expectations may be defined in an Attribute Practice Statement (APS). An APS can provide a listing of the attributes that will be used throughout the enterprise, and may identify authoritative attribute sources for the enterprise. Still further network infrastructure capabilities (including the ability to maintain attribute confidentiality, integrity, and availability) are required to share and replicate authoritative subject attribute data within and across organizations.

The Joint DoD/IC Attribute and Authorization Services Committee (AASC) [AASC] has made significant progress in identifying and establishing governance around several enterprise authoritative attributes. While this group is authoritative for the entire Federal Government, their work includes establishing models for sharing attributes and schemas for describing the attribute information, and developing APSs that may be used to define a service level assertion of quality and accuracy of authoritative subject attribute information.

3.1.3.4 Object Attributes

Standards do not exist for labeling objects. Object attributes are typically provisioned upon object creation and may be bound to the object, applied to an attribute that is bound to the object, or externally stored and referenced. It is to be expected that access control authorities cannot closely monitor all data acquisitions. Frequently, this information is driven by non-security processes and requirements. Good data that supports good access decisions is in the interests of the object owner, and measures must be taken to ensure that object attributes are assigned and validated by processes that the object owner considers appropriate for the application and authoritative. For example, object attributes must not be modifiable by the subject to manipulate the outcome of the access control decision. The object attributes must be made available for retrieval by access control mechanisms for access control decisions. Additional considerations for creating object attributes include:

- Most users will not be exposed to all potential values of an object attribute (e.g., to which sensitive compartment is a given user authorized). This should be accounted for in authoring tools, so that users only see the values that are applicable to them.

- As with subject attributes, a schema is required for object attributes defining attribute names and allowed values. Often plug-ins are required to surface document attributes to the user interface in an intuitive fashion.

- Attributes need to be kept consistent in DP, MP, and NLP.

There have been numerous efforts within the Federal Government and commercial industry to create object attribute tagging tools that provide not only data tagging, but also cryptographic binding of the attributes to the object and validation of the object attribute fields to satisfy access control decision requirements.

3.1.3.5 Access Control Rules

In ABAC, all data protection rules must include some combination of attributes and allowable operations. They may also include conditions, couplings, hierarchical inheritance, and complex logic. Together these provide a rich array of options when implementing ABAC. Rulesets and the application of rulesets to objects must be governed and managed appropriately. It is not enough to have a rich set of attributes without the rules that bind them to the allowable operations. Rules must accurately and completely reflect the NLP, and be authoritatively developed (some by organizations, some by resource owners)[4], applied, maintained, shared, and asserted. In some settings, one might limit the visibility of which rules apply to which objects to limit the likelihood of unauthorized subjects manipulating attributes to obtain authorization. In other circumstances, subjects that are denied access should have a method to verify or rectify the circumstances that caused the denial. Some organizations may wish to track the denials to see if the rules were appropriate. Similarly, rule definition and employment mechanisms and processes should include a robust rule deconfliction capability to determine rule conflicts and resolution processes.

An example of an important authorization-related standard is the Organization for the Advancement of Structured Information Standards (OASIS) eXtensible Access Control Markup Language (XACML).

[4] ABAC allows multiple rules from multiple stakeholders. New techniques are needed to coordinate and obtain the proper balance of sharing and protection

XACML is an XML-based special-purpose language used to describe policies, requests, and responses for DP. XACML provides a flexible and system-independent representation of access rules or policy that vary in granularity, allowing the combination of policies for different authoritative domains into one policy set for making access control decisions in a widely distributed system environment. More information about XACML can be found at http://www.oasis-open.org/committees/xacml/.

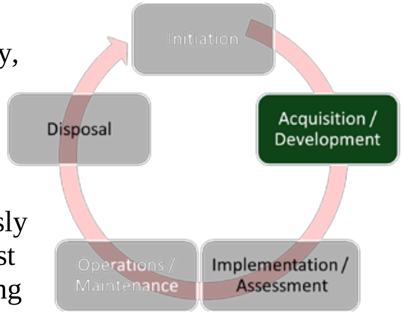
3.1.3.6 Access Control Mechanism and Context Handling

The distribution and orchestration of ACM must be predetermined to avoid conflicts and weaknesses in object protection. For example, if an identical object is held by two different organizations, an unauthorized subject should not be able to access the version held by the organization with lesser restrictions. ACMs should be managed, maintained, and employed in a consistent manner to ensure interoperability and comprehensive security.

The order in which the ACM retrieves information, evaluates for a decision, and enforces the decision can differ greatly based on the specific requirements of the implementation, and may even take into account environment conditions during access control decision rendering. This is referred to as Context Handling and simply refers to the workflow the ACM undertakes when gathering the data needed for a decision.

Additionally, where and how policy, attribute, and decision information is stored and exchanged throughout the enterprise is an important consideration. Note that there is no specific requirement that the PDP and PEP exist on the same system, though they are often co-located for performance and scalability benefits; the PDP and PEP may reside on separate devices or be managed as enterprise services.

3.2 Considerations during the Acquisition/Development Phase



During the acquisition/development phase, the system is designed, purchased, programmed, developed, or otherwise constructed. Typically, during this phase, the organization prepares the business processes needed for enterprise-wide execution and defines the systems to be deployed and integrated. This phase often consists of other defined cycles, such as the system development cycle or the acquisition cycle. During the first part of this phase, the organization should simultaneously define the system's security and functional requirements. During the last part of this phase, the organization should perform developmental testing of the technical and security features/functions to ensure that they perform as intended prior to launching the implementation/assessment phase.

3.2.1 Business Process Generation and Deployment Preparation

3.2.1.1 Documentation of Rules

For each of the types of objects controlled by an organization, there should be an accompanying set of access control rules documented in plain English or NLP. (Use cases might provide the easiest means for enterprise participants to define NLP for a set of objects.) These rules should dictate who can and cannot create, view, modify, delete, forward, and interact with data and services controlled by the organization and under what context or environment conditions they have those privileges. Documenting these rules incorporates the organization's interpretation of applicable policies and guidance, the specific sensitivities of applicable objects, and knowledge of appropriate user communities that will need the objects.

Documenting NLP facilitates the development of DP and provides traceability back to the written policy. For example, many organizations have difficulties transitioning their authorization capabilities from ACLs into a more robust ABAC infrastructure because no corresponding NLP exists. Many organizations still operate on ACLs that are maintained by a data owner who does not have documentation that specifies the required criteria for being granted access. As an example, consider that when a request for access is received, the data owner evaluates a set of criteria—usually undocumented—such as, "Is this person a member of the working group?" or "Am I familiar with this person or his or her organization?" and then renders a decision before adding the requestor's name to the appropriate ACL. Clearly documented access control rules provide the ability for an organization to define who should be allowed access to specific objects, establish the logic for the decision, and transition traditionally human-generated decision-making to an automated capability that can make consistent fine-grained access control decisions in real time.

3.2.1.2 Customizing Policy

Unless required by higher authorities or obligations, subordinate organizations should not make local policies less stringent. If subordinate organizations in an enterprise are able to independently relax the restrictions established for enterprise policy, the security inherent in the system is undermined, possibly allowing local access to enterprise objects where it would otherwise be forbidden.

3.2.1.3 Agreement and Understanding of Attributes

A consistent set of valid values must be defined and applied for enterprise subject and object attributes. This allows authorization decisions to be based on known values that are consistent throughout the enterprise. The lifecycle management of attributes is the responsibility of the provisioning organization, whether the attributes are used exclusively within an organization or across organizations.

3.2.1.4 Understanding Meaning of Attributes

Attribute service providers need to describe attributes and their relationship with other attributes so that consumers may properly and effectively use attributes in DP. Attribute service providers must document the definitions and meanings of enterprise authorization attribute values and provide guidance on the use of the attributes. In some cases, attributes must be used in combination with other attributes to establish a valid context, such as the combination of role and organization—a role has no meaning unless it is defined within the context of an organization. For example, the Director of Operations for an entire organization, whose responsibilities may encompass the Finance, Human Resources, Legal, and many other departments, has an entirely different contextual meaning from the Director of Operations within the Web Services branch of the IT Department. Without the understanding of the guidance related to the attribute, its context, and the knowledge that these attribute values are required together to render a decision, the DP—and hence the decision—may be generated on insufficient information or using faulty logic. The confusion caused by role overlap can be effectively addressed with ABAC. For example, in addition to a role attribute, a "unit" or "group" attribute can be used to provide scope.

3.2.1.5 Processes and Procedures for Object Access and Authorization Service Failures

A set of procedures and requirements for communicating exception handling, access denials, and errors should be established to provide users a means to remediate access decisions given mission, role, and need-to-know imperatives. As authorization services mature from the traditional method of provisioning an account and populating an ACL to an automated decision process, it will be more difficult for system users to understand and remedy access denials. A well-established process for properly discovering and obtaining the attributes needed for access approval will help ease the transition to a new paradigm of access control. This can be expanded to address dropped connections to any authorization service component.

In a mission-critical role, the user should be able to understand the limitations and request an exception, be pointed to an authoritative source of help, or attempt an alternate path to access equivalent information or services.

3.2.1.6 Attribute Privacy Considerations

ABAC capabilities should be developed to comply with all applicable privacy laws, directives, and policy. Due to the personal and descriptive nature of subject attributes, implementing attribute sharing capabilities may increase the risk of privacy violation of Personally Identifiable Information (PII) due to inadvertent exposure of attribute data to untrusted third parties or aggregation of sensitive information in environments less protected than the originator's. Organizations engaged in attribute sharing should employ trust agreements to ensure the proper handling of PII and enforcement of PII regulations. These trust agreements should detail authorized PII use and handling for all components in the trust chain as well as methods for validating, remediating, and adjudicating liability for regulatory infractions.

3.2.1.7 DP Creation

Every DP should be written to satisfy the requirements of a non-digital NLP. Only authorized individuals, who understand the limitations on sharing the object, know how to write DP that correctly reflect non-digital NLP, and have authority to write the digital policies, should write these policies. The digital rules or policies that are developed to protect objects must meet the objectives of relevant laws, organizational policies and mandates, and business and mission requirements. Without clear object ownership and accompanying authorities, policy deconfliction, traceability, and auditing of decisions may be difficult or impossible.

3.2.1.8 Distribution of Digital Rules and Policies

To reduce redundancy and inconsistencies, a single enterprise organization should be charged to develop digital rules and policies reflecting federal, department, agency, and enterprise policy. Enterprise-applicable policies should be written at the highest level in the enterprise and be promulgated to subordinate organizations. Individual organizations should develop local policy and unique policy that applies only to their constituent or subordinate organizations.

3.2.2 System Development and Solution Acquisition Considerations

3.2.2.1 Standardization within the Enterprise

Implementers of ABAC should strongly consider using a comprehensive standards-based approach that enables current day interoperability and future deployment flexibility by making use of products or capabilities that are built upon widely accepted standards and that employ commonly used interoperability enablers (such as XACML) endorsed by large enterprises. A beneficial way to achieve interoperability and achieve cost-efficient ABAC deployments is to establish and enforce a series of standards, specifications, and profiles that address the functionality, interfaces, and infrastructure required for enterprise ABAC capabilities.

Although numerous authorization solutions exist, in instances where a comprehensive standards-based approach is not used, they can be limited in their range of abilities, may be components of a suite of products with proprietary interfaces, or may be able to only partially meet available standards and specifications. Standards that have optional elements may be implemented inconsistently by vendors and developers, making it possible for two services or applications that are fully compliant with a standard to be non-interoperable. For this reason, well-defined and standardized profiles should be strongly encouraged, especially in cross-organizational environments. When acquiring ABAC solutions, implementers should use commonly agreed-upon tailored profiles as well as leverage the standards and profiles contained within existing standards registries.

Individual authorization service components (e.g., access control point, policy decision point, policy enforcement point, policy retrieval point, attribute retrieval point, meta attribute retrieval point) should be developed with standard, open interfaces so that products from multiple vendors can be employed while ensuring interoperability.

3.2.2.2 Interoperability Requirements

A set of requirements addressing functionality, interfaces, infrastructure, and product support should be developed and employed as a filter within the procurement process for all acquisitions regardless of categorization or affiliation. Often, enterprise service authorization components are procured outside of the system acquisition process—either as a service developed under existing contract vehicles or as a small set of functionality within a larger mission system procurement. Without a common set of requirements and an enforcement process, organizations may purchase a wide variety of authorization capabilities such as RBAC, ABAC, and others that meet stovepiped mission and budget requirements but can fail to meet interoperability expectations for the enterprise as a whole.

3.2.2.3 Identity Management Integration

A request for access to an object must be authenticated as originating from a unique subject through the use of identity credentials before an access decision can be made. The ABAC system needs to support the prevalent and strategic authentication mechanisms and credentials used by the organization. This may mean the organization needs to make enhancements to its authentication infrastructure, if its current state impedes ABAC adoption. The subject attributes conveyed in these credentials should uniquely determine the subject via an identity vetting process recognized throughout the enterprise. Strong authentication methods should be used that are of sufficient assurance for the request (see NIST SP 800-63-1 and SP 800-63-2). Once the request is authenticated, subject attributes associated to the unique subject can be used to determine an access decision, and access decisions can be captured in required audit records/systems to provide attribution of the request to the unique subject. For example, a request transferred via a TLS 1.2 session with client authentication (see IETF RFC 5246) depending on X.509 certificates issued by a trusted certificate authority is associated to the subject bound by the certificate authority to the distinguished name.

3.2.2.4 Support of Diverse Identities

Identity or subject attribute designations should not be limited to human actors. Authorization services use identities and attributes associated with entities in any form. The attributes bound to the identity not only help define the unique identity but also reflect the context of that entity within an organization and establish the individual's persona. An individual may have more than one persona but uses only one at a time. Therefore, in this context identity is defined as:

Identity: *The set of attribute values (i.e., characteristics) by which an entity is recognizable and that, within the scope of the identity manager's responsibility, is sufficient to distinguish that entity from any other entity.*

In some cases, access control decisions may be associated to NPE subjects acting on behalf of one or more individuals. These NPEs can authenticate a request using their own identity credentials. For example, a Watch Officer gains access to resources via group account representing the "Watch Officer" role. Resources accessible by policies that only depend on group or role membership can support access authorization using the NPE's identity credential to authenticate the request. Of course, the access control system basing an access decision on an NPE credential will not be able to attribute the request to the individual or individuals who may be acting in that role, or logged into the group account, at the time of the request.

In addition, NPEs need to be supported as allowable subjects by authorization services. An NPE may act either independently or on behalf of an authenticated individual. Examples of NPEs include network devices (e.g., switches, routers), processes running on servers (e.g., portals), workstations, and other endpoint devices. As mission and security functions are increasingly automated, NPEs will play a larger role as actors in authorization service interactions.

3.2.2.5 An Authentication Service for Mutual Authentication

Within the authorization service, authentication at each point of information exchange for retrieval of policies, attributes, and meta attributes as well as assertion of policy decisions is necessary to ensure the validity of the information being used for access decisions. For each exchange, proof of origin, data integrity, and timeliness is required. Mutual authentication may be required when authorization service components exchange sensitive information, or to support quality of service or performance requirements. When the authorization service needs to obtain attributes from an authoritative attribute service, mutual authentication must be used between the two services to protect message integrity (assuring that the attribute request that was received by the attribute service matches what the decision service sent) and message origin (the attribute service receiving the request is assured that the sender is a valid policy decision service). Authentication protocols based on strong methods (e.g., X.509 authentication) should be used to provide the level of assurance needed by both parties involved in the attribute exchange.

3.2.2.6 Enterprise Authorization Services Integration with Security Controls

Authorization services alone are not enough to ensure the security needed to protect the mission-critical objects resident on the networks. Comprehensive and cohesive enterprise security capabilities are needed to establish the desired level of assurance, and they must be tightly integrated and able to seamlessly feed the security information needed for making security decisions. A set of integrated authentication, authorization, security audit, security configuration management, continuous monitoring, and cyber defense capabilities will provide the desired level of confidentiality, integrity, availability, non-repudiation, and situational awareness needed to holistically protect the information and services needed to support the enterprise.

3.2.2.7 Establishment and Accessibility of Subject Attribute Sources

Subject attribute repositories and provider services need to be formally established, accessible to participating object owners, and accountable to a standard quality of service for attribute provisioning. Authorities should be clearly identified so object owners know which attribute service to solicit as the authoritative source when an object request is received and attributes must be collected for policy evaluation. Ideally each subject should have a single attribute service to identify and provision the appropriate authoritative source for each subject attribute.

3.2.2.8 A Shared Repository for Subject Attributes

Direct use of shared repositories for subject attributes is encouraged for consumers who have sufficient network connectivity to take advantage of economies of scale, increased quality control, and standard interfaces. Another advantage of using shared attribute repositories is that they provide a single access point for data that is from multiple sources. Building and managing a connection to a single access point is much less complex than managing multiple connections. In some cases, limited connectivity, insufficient bandwidth, or intermittent connections may prevent service providers from being able to use shared repositories reliably. Consumers that must maintain local copies of data that cannot synch with service providers will not be able to use a shared attribute repository and thus will not have access to the most current and highest quality data.

3.2.2.9 Minimum Standard Sets of Object Attributes

Just as a minimum set of subject attributes should be defined for the user population to promote enterprise interoperability, a minimum set of object attributes should be defined for objects. Objects being made available for access outside the owning organization will need to have the minimum set of attributes to be eligible for discovery and access. With a standard set of enterprise subject attributes and object attributes, DP applying to all enterprise objects can more easily be developed and modified to reflect changes in policy. A good example of where this methodology has been employed is with classification and compartmentalization markings within classified networks. In most cases, an object cannot be placed on the network without proper marking, and access control policies are written to address the finite and well-known set of classification and compartmentalization markings.

3.2.2.10 Object Attribute Management

Objects must have a complete and valid set of object attributes for subject access decisions to be accurate and appropriate. As objects are created or modified, their attributes need to be generated or modified accordingly. Without a comprehensive and accurate set of object attributes, access decisions will be made on faulty information or denied simply because the object attributes are not complete. Additionally, some form of validation, integrity, provenance mechanism (to verify the completeness, allowable values, integrity, and change history of object attributes) should be integrated into the mechanism or framework used to manage object attributes.

3.2.2.11 NLP Traceability

A comprehensive and coherent traceability between high-level enterprise written policy/NLP and low-level enterprise or local DP should be maintained by an appropriate authority. This will enable changes to written policy to be evaluated and subsequent DPs to be altered accordingly. With this policy traceability, the plethora of DPs resident in local organizations will be auditable, verifiable, and alterable given any change to requirements.

3.2.2.12 Digital Rules or Policies Based on the Agreed Attributes

If an organization has an agreement with one or more organizations to grant authorization to access objects based on a defined list of attributes, the organization that owns the objects must ensure that it writes access control policies based only on those attributes. Every effort should be made to use any accepted common set of shared enterprise attributes, no matter how limited, to ensure basic interoperability if only to effect a limited secure information sharing capability. As new requirements arise, the enterprise may choose to introduce new enterprise attributes and rules for sharing them.

3.2.2.13 Externalization of Policy Decision and Enforcement Services

Where practical for enterprise solutions, it is recommended that PDPs and, to a lesser degree, PEPs be implemented as services, separate from individual enterprise services and applications. Doing so removes the burden and expense of providing similar decision and enforcement services for every enterprise service or application, since a single PDP or PEP can support multiple enterprise services. Allowing service providers to simply use PDP and/or PEP services that are provided by the larger enterprise or by the organization greatly simplifies service/application development; saves money that would otherwise be spent on licensing, training, configuring, and deploying disparate instances of these services; and moves operations and maintenance away from individual programs.

3.2.3 Considerations for Advanced Enterprise ABAC Capabilities

As the enterprise embarks on developing and implementing ABAC enterprise authorization capabilities, architects and program managers must keep in mind that there will inevitably be a long transition from the legacy access control methods in use now to the desired end state. As standards and technology mature, organizations will need to embrace concepts that enhance interoperability and promote higher assurance solutions while discarding proprietary, stovepiped solutions.

3.2.3.1 Incorporation of Environment or Contextual Condition Attributes

Environment or contextual information must be fed into the access control process based on the level of assurance necessary. The level of assurance is the degree of certainty or confidence in the subject when presenting a credential. The use of environment or contextual attributes enables usage of existing infrastructure technologies and properly distributes risk across identity providers and relying parties. Access control decisions leveraging context, such as time of day, authenticator time, and transaction value, will increase the level of assurance. Just like subject attributes, it is important to identify the relevant environment or contextual attributes for authorization, standardize the attribute data, and assess the availability of this data. The environment or contextual attributes will evolve over time; as the supporting technologies change so will the measurement of the environment and contextual attributes. There must be a process in place that will audit the relevancy of the attributes and update the associated policies, and there must be authoritative governance of this attribute management process.

3.2.3.2 Measuring the Confidence of Access Control Decisions

Ideally, an access control decision is made by using the most accurate, timely, and relevant data gathered from the most authoritative source(s) possible. As accuracy, timeliness, relevance, authority, and quality suffer from incomplete information, inattention to detail, and inability to update, the overall confidence in the access control decision must proportionately suffer. Measures of confidence concepts are fairly new and are not found in most privilege management products available today. Substantial research, requirements analysis, policy definition, and proof-of-concept work is required to further define the mechanisms and policies that can achieve the goal of computing a measure of confidence value. The value is computed by establishing levels of confidence associated with a requestor's identification and authentication processes (e.g., strength of authentication mechanism, identity vetting, credential issuance and proofing, attestation, source Internet Protocol [IP] address), and the confidence with the corresponding ABAC implementation, and then using computed measures of confidence values as real-time derived attributes that can affect the authorization decision process.

3.2.3.3 Mapping Attributes between Organizations

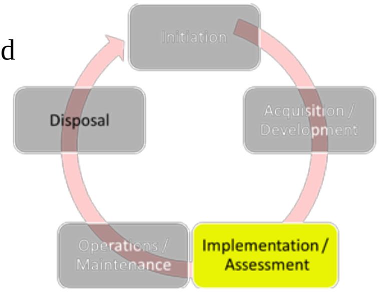
Most organizations name attributes and attribute values differently. At some point, it will be important to implement solutions that provide attribute mapping between enterprise organizations to minimize the need for a special class of attributes called “enterprise attributes.” Attribute mapping serves as a translation between attributes or attribute values that are named differently. For example, one organization may use the name Title and another may use the name Salutation to refer to the same set of attribute values.

3.2.3.4 Integrating Attribute Sets into Policy Development Capabilities

Automated policy generation capabilities should incorporate the current available set of enterprise attributes to assist in generation of the DP. As new DPs for enterprise shared objects are being generated, the only attribute options that should be available to the policy creator are those that have been agreed upon for enterprise data sharing. If policy creators are allowed to create or designate their own attributes, policies may not be interoperable. By enforcing adherence to a specific set of attributes, the policies will be uniform and easily understood. Having this capability built into the policy generation interface will make policy creation easier while at the same time ensure compliance with attribute standards.

3.3 Considerations during the Implementation/Assessment Phase

In the implementation/assessment phase, the organization configures and enables system security features, tests the functionality of these features, installs or implements the system, and finally, obtains a formal authorization to operate the system. Most of the considerations during this phase are focused on optimizing performance and ensuring security features work as expected.



3.3.1 Attribute Caching

What has been typically observed when an ABAC solution moves from the prototype/pilot to implementation is that attribute caching becomes necessary due to the number of requests for attributes. Stated another way, performance of the ABAC solution can be negatively affected if each access decision requires an across-the-network attribute request. This is especially apparent in low-bandwidth, high-latency environments.

When designing the ABAC implementation, the organization will need to make a decision regarding the caching of attributes. In addition to performance issues regarding attribute caching, the organization will need to evaluate and address a tradeoff regarding the freshness of attributes and its impact upon security. Attributes that are not refreshed as often will ultimately be less secure than attributes that are refreshed in real time. For example, a subject’s access rights may have changed since the last refresh, but those updates will not be reflected in their available access rights until the next refresh.

In disconnected environments, attribute availability at the local (disconnected) location will be mandatory. The security ramifications of using cached attributes at the local level will need to be decided upon within the implementing organization at a policy level, and then addressed with appropriate technical controls. In these disconnected environments, administrators may employ risk-based analysis as a basis for access decisions, as some attributes at the local (disconnected) level may change or be removed before the system refreshes its attributes. The local (and disconnected system) possible use of stale or removed cached attributes could introduce a level of risk to the system, as the local system is not making use of the most recently available attributes. Therefore, a risk-based analysis may be warranted as to whether or not to deploy this type of solution.

3.3.2 Attribute Minimization

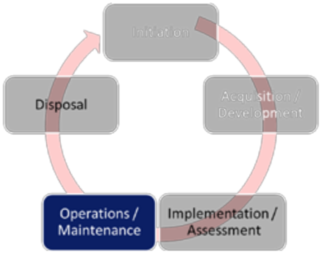
Keeping to a minimum the number of attributes used in authorization decisions will improve performance and simplify the overall security management of the ABAC solution.

Organizations that are planning to deploy an ABAC solution will benefit from establishing a close working relationship among all of the organization's stakeholders who will be involved in the solution's deployment.

3.3.3 Availability of Interface Specifications

In order to help ensure that consistently reliable availability to ABAC services occurs, all organizations that will participate in information sharing through enterprise ABAC capabilities should fully understand the interface, interaction, and precondition requirements for all types of requests. Requests may include the more commonly described attribute requests, as well as object attribute and DP requests. It is also important to ensure that as changes occur in the infrastructure and interface requirements, all relying parties are provided notification of updates so they can plan to modify their components accordingly.

3.4 Considerations during the Operations/Maintenance Phase



In the operations/maintenance phase, systems and products are in place and operating, enhancements and/or modifications to the system are developed and tested, and hardware and/or software is added or replaced. During this phase, the organization should continuously monitor performance of the system to ensure that it is consistent with preestablished user and security requirements, and needed system modifications are incorporated.

3.4.1 Availability of Quality Data

As the information needed to render access control decisions, and in some cases the decisions themselves, is externalized from the objects and consumers, access to information and services will become more dependent on an outside service's ability to provide timely and accurate data used for access decisions. The infrastructure used to support attribute services, attribute stores, policy stores, policy and attribute generation and validation components, decision engines, and meta attribute repositories as well as the conduits through which all of those requests and information must pass must be robust, resilient, well-tested, of high quality, and able to scale to the needs of the missions and functions supported. Service agreements should detail availability, response time, and data quality and integrity requirements. For example, failover, redundancy, and continuity of operations must be considered for data and services that are considered mission critical. Maintaining high availability of quality data will require that the addition, updating, and deleting of attribute values is performed by trained, authorized individuals, typically organized by workflows with appropriate approvals, and regularly audited.

Providers and consumers of attributes for authorization services should publish and adhere to a set of formal agreements within the enterprise to meet a minimum standard of service, quality, availability, protection, and usage. Various laws and regulations establish responsibilities, liabilities, and penalties related to the appropriate protection of information such as classified, sensitive, private, or proprietary information, as well as personally identifiable information. The agreements should capture these requirements as well as those related to liability of data ownership/possession. Data ownership refers to both the possession of and responsibility for information.

The control of information includes not only the ability to access, create, modify, package, derive benefit from, sell, or remove data but also the right to assign these access privileges to others. It is incumbent upon the data owner to adhere to applicable laws and regulations and to ensure proper policies are in place to pass applicable restrictions to external entities accessing and using the data.

One of the most difficult hurdles to information sharing is the ability of one organization to “trust” another organization with its data. These agreements would serve to formalize that trust relationship with a series of requirements and, possibly, penalties for nonconformance. APSs and MOUs/MOAs for attribute services and authoritative and accountable attribute sources can also serve to translate organizational policy into operational procedures. The purpose, usage, participants, responsibilities, and administration of these services are described in these formal agreements.

3.4.2 Distribution of Timely and Accurate Subject Attributes

Implementing an authorization service that relies on subject attributes depends on a high level of availability and consistently reliable access to enterprise attribute services. Users in austere environments may not have reliable on-demand access to enterprise services. To support users with disconnected operations, intermittent connectivity, and limited communications, alternative methods for obtaining data and allowances for caching or local storage of enterprise data may be necessary and a formal strategy for providing this support should exist.

An example of an austere environment is the deployment of a seagoing vessel. The deployed ship will have a semi-static user population with only intermittent but non-ideal connection to enterprise network fabrics. Because the deployed user population will have only minor changes throughout their transit, supporting the “unanticipated” system user is less of a concern. In this case, a bulk download and local storage of subject attributes may be sufficient for most local access control decisions. Therefore, subject attribute data could be stored locally on the ship throughout a deployment, and local applications and services could use the data from the local store without the need to reach to an authoritative enterprise attribute source. While this is one example of a solution to an austere environment problem, it should not be inferred that this is the only solution.

4. Conclusion

This document brings together many previously separate bodies of ABAC knowledge in order to bridge existing gaps between available technology and best practice ABAC implementations and to address the emerging demand for ABAC employment within the Federal Enterprise.

This document defines general concepts necessary to understand ABAC. It defines subject and object attributes, and the generic features of an ABAC mechanism that allows further dialogue about the merits of specific implementation mechanisms. It brings to light numerous considerations aligned to the SDLC that must be factored in the planning, design, development, implementation, and operation of ABAC capabilities within an enterprise. The advantages and common pitfalls of ABAC mechanisms are discussed, especially for large or federated enterprises.

ABAC capabilities will allow an unprecedented amount of flexibility and security while promoting information sharing between diverse and often disparate organizations. It is vital that these capabilities be developed and deployed using a common foundation of concepts and functional requirements to ensure the greatest level of interoperability possible. ABAC is well suited for large and federated enterprises. An ABAC system can implement existing role-based access control policies and can support a migration from role-based to a more granular access control policy based on many different characteristics of the individual requester. It supports the unexpected user and provides a more efficient administration. However, an ABAC system can be more complicated, and therefore more costly to implement and maintain, than simpler access control systems.

Future publications will address more formal definitions of ABAC as a family of access control models, highlight standards and specifications available for use, and address in greater detail the complexities of enterprise ABAC deployment and operation.

Appendix A - ABAC Examples

Example 1

Various Government Organizations have synergized efforts that yielded the successful demonstration of ABAC systems that realize IdAM capabilities. Through integration of evolving, commercially available technologies and products into the ABAC system for a web information portal, this example provides evidence that ABAC systems provide fine-grained access control functionality. Fine-grained access control uses integrated security mechanisms such as built-in row level security (RLS) and parametric views to support the principle of least privilege, in which the levels of access are managed down to the smallest discrete element of protected data, resource, or data/resource subset. RLS essentially rests on setting an application role automatically when the user logs in via a web application server, and then the web application server sets an appropriate structured query language (SQL) predicate based on the role. Parametric views generated from a web server provide fine-grained access control by performing the following functions: 1) to transfer the users' identities to the databases that house the requested and protected resources in question, and 2) upon successful authorization, to display the relevant data to the requesting users in question. A web information portal is an information technology (IT) framework for integrating information, data, enterprise applications, people, processes, and other enterprise resources and assets across Government Organizations. It provides a secure unified access point, often in the form of a web-based graphical user interface (GUI) or web-based client application, and is designed to aggregate and personalize information through pluggable user interface software components that are managed and displayed in a web information portal, called portlets. Through this example of an ABAC system integrated into a web information portal, the realized capabilities of assured information sharing and collaboration among workers across various Government Organizations helps them to perform daily business operations, as well as critical tasks in the event of an emergency.

The paragraphs that follow introduce the basic system overview and objectives for the ABAC system integrated into a web information portal. This section then concludes with discussion on the lessons learned from the demonstration, along with details highlighting the best industry practices for ABAC implementation.

Figure 11 is the system overview for an ABAC system. The web information portal demonstrates ABAC capabilities that provide, for workers at various Government Organizations (i.e., users), assured information sharing and secured access via authorization to protected resources that reside in and are managed by the Government Organizations. In addition to PDP and PEP, the basic architecture of the ABAC system that protects a web information portal includes the following major components:

- Authoritative Attribute Store(s)—provides collections of data (usually housed with data cluster or set of databases) that are official sources of attribute data that is authorized by the Government Organization and/or Data Custodian responsible for the custodianship and/or ownership of the attribute data and that overrides all other attribute sources

- Policy Store(s)—provides the Storage Area Network (SAN) for all policies that govern access to objects, which include the web information portal and the protected enterprise resources.

When the user from a Government Organization makes a request for a resource (e.g., resource request A, B, or C), the request transcends from the user's terminal at the Government Organization to the web information portal server. The PEP on the back end of the web information portal server forwards the request to the PDP. The PDP performs the back-end access decision processing (i.e., authorization) to determine if the PEP shall grant or deny access to the requested resource. Upon the successful PDP ingest and processing of authoritative attributes, environment conditions, and associated policies for the requested, protected resource in question, the PDP makes the appropriate access decision(s) and the PEP executes the results of the authorization decision(s) as such. Upon successful authorization by the PDP and PEP, the PEP provides access to the resource (i.e., responses to resource requests A, B, and C). The resource is then forwarded via the web information portal, and the resource is shared among known and authorized users with provisioned (i.e., preregistered and established) web information portal user accounts, as well as known, but unanticipated, users without pre-provisioned web information portal user accounts. External users, by definition, are authenticated users that do not necessarily possess established web information portal user accounts registered within the Government Organization that manages and controls a web information portal.

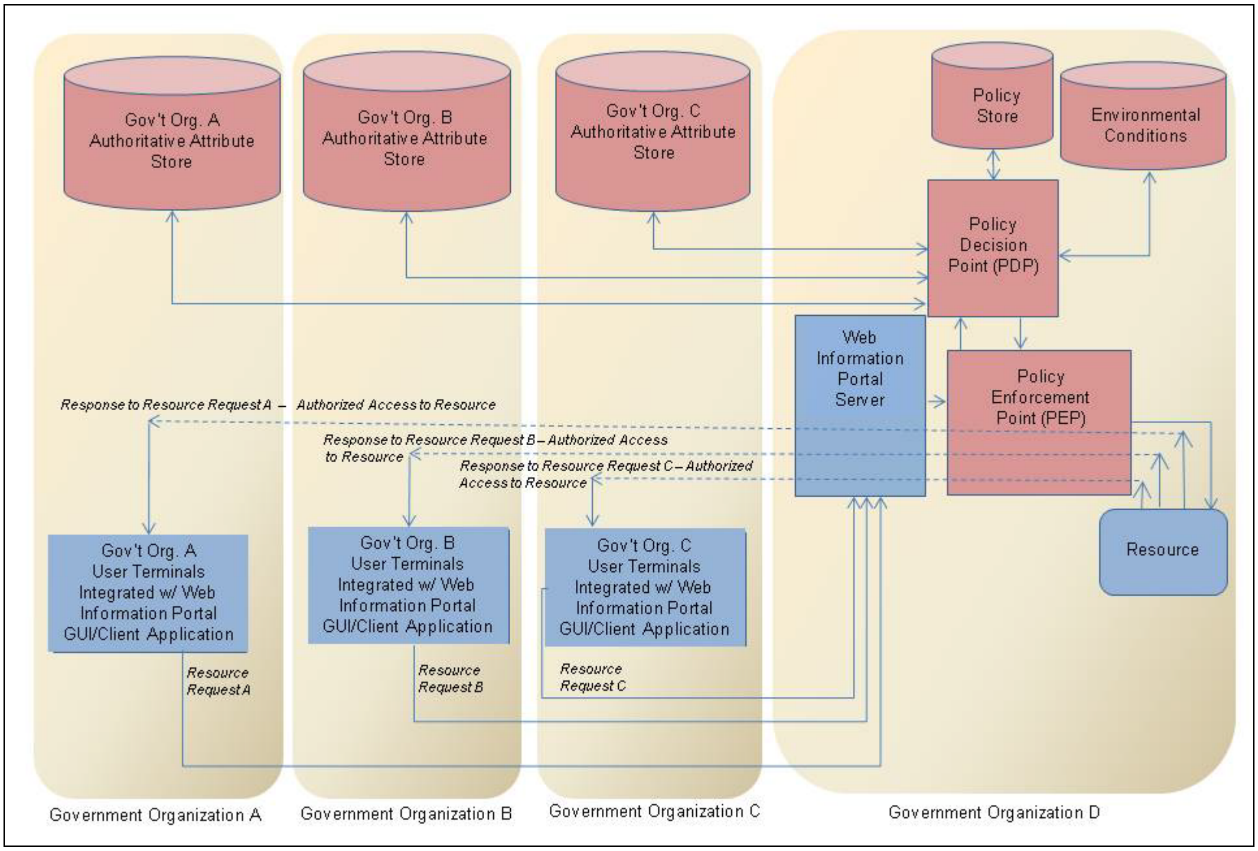

Gov't Org. A Authoritative Attribute Store
Gov't Org. B Authoritative Attribute Store
Gov't Org. C Authoritative Attribute Store
Policy Store
Environmental Conditions
Policy Decision Point (PDP)
Web Information Portal Server
Policy Enforcement Point (PEP)
Response to Resource Request A – Authorized Access to Resource
Response to Resource Request B – Authorized Access to Resource
Response to Resource Request C – Authorized Access to Resource
Resource
Gov't Org. A User Terminals Integrated w/ Web Information Portal GUI/Client Application
Gov't Org. B User Terminals Integrated w/ Web Information Portal GUI/Client Application
Gov't Org. C User Terminals Integrated w/ Web Information Portal GUI/Client Application
Resource Request A
Resource Request B
Resource Request C
Government Organization A
Government Organization B
Government Organization C
Government Organization D

Figure 11: System Overview of a Web Information Portal Using ABAC

The various Government Organizations had developed the proposed, verified design for an ABAC system integrated into a web information portal for the following reasons:

- To enable dynamic ABAC capabilities that allow or deny access to all or part of a web information portal based on a single environmental attribute or external condition change to support surge demands associated with an emergency. Whenever users need to acquire access to resources, whether centralized at one Government Organization or distributed among various Government Organizations, the required processes that support this need are static, manual, very coarse-grained access control, with complicated and manpower-intensive account and information management. Coarse-grained access control uses traditional access control models for two absolute modes of access control to protected resources during an event or operational scenario: allow access to all protected resources or deny access to all protected resources. Also, coarse-grained access control ignores external and environment conditions; context (or implied usage for the protected data/resource in question); and access level granularity for protected resources with respect to making appropriate access decisions. Typically, security mechanisms such as access control lists (ACLs) and role-based access control (RBAC) provide coarse-grained access control. Therefore, when an emergency arises, the time and energy that system administrators require to provide the appropriate levels of access to the resources for particular user groups (i.e., granular access control) may create a precarious situation where the system administrators may decide to ignore access control during surge periods in order to provide availability for the current task or scenario. This is because current legacy environments do not have access control policies or environment conditions established for user groups whose domains are outside of that particular environment. If these environments could seamlessly support shared access to its policy stores for all users and shared access to environment conditions that are mapped to a particular user group, then the legacy environment could expeditiously allow the ABAC system to acquire the appropriate policies and required external conditions that are affiliated with primary user groups. ABAC could then perform the required processing and determination of appropriate access decisions (i.e., authorization). This feature would temporarily allow waivers for external users to acquire access to shared objects (e.g., resources) within a particular legacy environment or domain in the event of an emergency.

- To enable assured sharing of sensitive information to a restrictive set of external users by using ABAC. To protect resources in compliance with applicable laws, regulations, policies, etc., access controls must remain in place to provide confidentiality, integrity, and availability of the protected resources for authorized user groups only. Therefore, the ABAC system demonstrates security controls that enable fine-grained access control policies based on: 1) subject (or user group) attributes; 2) object (i.e., resource) attributes; or 3) environment conditions. With these functions, the ABAC system for a web information portal can allow policy managers and system administrators to dynamically make instantaneous or near real-time granular changes to business rules (policies) and access control parameters. This will maintain the appropriate levels of access to the protected resources for the appropriate user groups and allow for dynamic changes to these levels of access when needed.

The lessons learned from ABAC implementation by organizations for the web information portal include the following:

- Established formal agreements for the development, integration, and deployment of future ABAC and IdAM implementation and deployment projects should state the objectives clearly and commitments explicitly among the appropriate and official Government Organization leaders. Acquiring the appropriate funding commitments and formal agreements early, including access to authoritative attribute stores and other enterprise resources managed and housed at various Government Organizations, shall ensure that there exist low ABAC implementation and sustainability risk in the event that role/job assignments for the Government Organization Operations Security (OPSEC) personnel or the Chief Information Officer (CIO) change.

- Stakeholders for the future IdAM-ABAC implementation and deployment projects should establish a Stakeholder Consortium to create the initial high-level concept for the IdAM-ABAC implementation and deployment project. The Stakeholder Consortium should also define the initial set of high-level policies; required subject, object, and environment attributes; and other desired capabilities for ABAC implementation and deployment. This is to ensure that through various system engineering artifacts, such as the Concept of Operations (CONOPS), all initial stakeholders would have early concurrence and buy-in for the execution of the technology development (TD) phase of the development efforts toward ABAC implementation.

All users should perform early security requirements definition for concurrence and buy-in to ensure that the specific security requirements for each target environment in question are satisfied. The selected commercial products that satisfy these security requirements and establish the architecture for the Web information portal, as appropriate, should not adversely affect the mission effectiveness and performance afforded by their integrated ABAC capabilities, as originally advertised and validated.

Example 2

Data Loss Prevention (DLP) tools are used to prevent data from being copied, modified, transferred, or sent to unauthorized users or systems. DLP tools are effectively fine-grained access control systems. They operate on networks, servers, and endpoints to protect data in motion, data in storage, and data in use. Most DLP tools today utilize proprietary mechanisms to identify and categorize information. Migrating DLP to the ABAC model will promote interoperability between the DLP systems and ABAC systems that currently control access to applications, web services, and file repositories.

From an architectural perspective, DLP client components (endpoint and server software) function as ABAC PEPs. The DLP policy decision point could be instantiated as an ABAC PDP. The DLP administrative console would then be an ABAC PAP. The various components could be co-located as needed for performance purposes.

DLP tools apply object attributes to information objects in the discovery phase. If these DLP tools tagged and used standardized attribute name/value pairs, such as the OASIS XACML EC-US and IPC profile metadata, the entire enterprise could benefit from consistently applied attributes and policies for defined domains.

Moreover, DLP systems make access control decisions according to proprietary policies. If DLP systems standardized on the XACML policy format, enterprise policy authorities could use the same language to define access control policies for endpoints, networks, servers, applications, web services, and file repositories. The cost savings and improvements to security posture would be substantial.

Appendix B - Acronyms and Abbreviations

Selected acronyms and abbreviations used in the guide are defined below.

AASC	Attribute and Authorization Services Committee
ABAC	Attribute Based Access Control
AC	Access Control
ACL	Access Control List
ACM	Access Control Mechanism
APS	Attribute Practice Statement
CIO	Chief Information Officer
CONOPS	Concept of Operations
COTS	Commercial Off-the-Shelf
DAC	Discretionary Access Control
DLP	Data Loss Prevention
DoD	Department of Defense
DP	Digital Policy
DPM	Digital Policy Management
FICAM	Federal Identity, Credential, and Access Management
FISMA	Federal Information Security Management Act
GUI	Graphical User Interface
HIPAA	Health Insurance Portability and Accountability Act
IBAC	Identity Based Access Control
IdAM	Identity and Access Management
IETF	Internet Engineering Task Force
IP	Internet Protocol
IR	Interagency Report
IT	Information Technology
ITL	Information Technology Laboratory
MAC	Mandatory Access Control
MP	Meta Policy
NIST	National Institute of Standards and Technology
NLP	Natural Language Policy
NPE	Non-Person Entity
OASIS	Organization for the Advancement of Structured Information Standards
OMB	Office of Management and Budget
OPSEC	Operations Security
PAP	Policy Administration Point
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PII	Personally Identifiable Information
PIP	Policy Information Point
PKI	Public Key Infrastructure
RAdAC	Risk-Adaptable Access Control
RBAC	Role-Based Access Control
RFC	Request for Comment
RLS	Row Level Security
SAN	Storage Area Network
SDLC	System Development Life Cycle
SOA	Service Oriented Architecture

SP	Special Publication
SQL	Structured Query Language
TCSEC	Trusted Computer System Evaluation Criteria
TD	Technology Development
TLS	Transport Layer Security
XACML	Extensible Access Control Markup Language
XML	Extensible Markup Language

Appendix C - References

[AASC] "DOD Federated ABAC Symposium Summary Report", Department of Defense and Intelligence Authorization and Attribute Services Committee (AASC).

[CGLO09] Cruz, I. F., Gjomemo, R., Lin, B., & Orsini, M., "A constraint and attribute based security framework for dynamic role assignment in collaborative environments", Collaborative Computing: Networking, Applications and Worksharing, pages 322-339, 2009.

[FEDCIO1] Federal Identity, Credential, and Access Management (FICAM) Roadmap and Implementation Guidance Version 1.0, November 10, 2009.

[FEDCIO2] Federal Identity, Credential, and Access Management (FICAM) Roadmap and Implementation Guidance Version 2.0, December 2, 2011.

[FK92] Ferraiolo, D. and Kuhn, R., "Role-Based Access Controls," In Proceedings of 15th NIST-NCSC National Computer Security Conference, pages 554-563, http://csrc.nist.gov/rbac/ferraiolo-kuhn-92.pdf, October 13-16, 1992.

[NIST7316] Hu, V., Ferraiolo, D., and Kuhn, D.R., "Assessment of Access Control Systems", NIST IR 7316, http://csrc.nist.gov/publications/nistir/7316/NISTIR-7316.pdf, 2006.

[NIST7657] NIST/NSA Privilege (Access) Management Workshop Collaboration Team, "A Report on the Privilege (Access) Management Workshop," NIST IR 7657, 2010.

[NIST7665] "Proceedings of the Privilege Management Workshop", NIST IR 7665, September 1-3, 2009.

[NIST7874] Hu, V., and Scarfone, K., "Guidelines for Access Contol System Evaluation Metrics", NIST IR 7874, 2012.

[TCSEC] Trusted Computer System Evaluation Criteria, DOD 5200.28-STD. Department of Defense, 1985.

[WWJ04] Wang, L., Wijesekera, D., & Jajodia, S., "A logic-based framework for attribute based access control", in Proceedings of the 2004 ACM workshop on Formal methods in security engineering, pages 45-55, October 2004.

[XACML] OASIS, "eXtensible Access Control Markup Language (XACML)", http://www.oasis-open.org/committees/xacml/.

[YT05] Yuan, E. and Tong, J., "Attributed Based Access Control (ABAC) for Web Services," Proceeding ICWS '05 Proceedings of the IEEE International Conference on Web Services, pages 561 - 569, 2005.