Enterprise Dynamic Access Control
Version 2
Overview

Prepared for
Commander, U.S. Pacific Fleet
Pearl Harbor, HI 96860

Prepared by
Richard Fernandez
SSC San Diego
675 Lehua Ave, Building 992
Pearl City, HI 96782
(808) 474-9270, fax (808) 471-5837
fernandr@spawar.navy.mil

Revisions

Publication Debut	May 1, 2005	Richard Fernandez
EDAC version 2	Jan 1, 2006	Richard Fernandez

Acknowledgements

The author wishes to acknowledge the following personnel: Wallace Fukumae, Ryan Kanno, Dean Tanabe, Tuan Huynh and Wilfredo Alvarez.

Special thanks to Rick Kuhn and Mike Hogan from the National Institute of Standards and Technology (NIST) and Dr. Coyne from the Veterans Administration (VA).

Trademarks

Invention Disclosure

Resources

National Institute of Standards and Technology, Role Based Access Control: http://csrc.nist.gov/rbac/

Ravi Sandhu, David Ferraiolo, Richard Kuhn. American National Standard for Information Technology (ANSI) Role Based Access Control, 359-2004, 2004

OASIS Technical Committee, Extensible Access Control Markup Language 2.0 (XACML) Technical Committee http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml

Richard Fernandez. Enterprise Dynamic Access Control (EDAC) Compliance with the Role-Based Access Control (RBAC) Standard ANSI/INCITS 359-2004, May 2005. EDAC Compliance with the NIST RBAC Standard ANSI/INCITS 359

Marlin Pohlman, LDAP Metadirectory Provisioning Methodology, 2001.

David Ferraiolo, Richard Kuhn, Ramaswamy Chandramouli, Role-Based Access Control, 2003.

CONTENTS

Abstract

The Enterprise Dynamic Access Control (EDAC) represents an authorization model that adheres to the core specifications in the Role-Based Access Control (RBAC) standard (ANSI/INCITS 359-2004) authored by the National Institute of Standards and Technology (NIST). The EDAC accommodates complex and scalable access control situations many government and civilian organizations are experiencing when managing resource access.

Authorization is the process that evaluates resource access. Resources can represent software applications, web services and even facility access. Currently, access control lists (ACL) and groups represent static listings of individual names or identifiers allowed access to resources. This per person approach of establishing resource access becomes unmanageable as the number of users requiring resources access grows. EDAC automates the complexities and labor-intensive tasks of assigning users to resources. This makes EDAC a scalable solution that can accommodate a growing customer base without increase to resource management workload or sacrifice to security. EDAC establishes an effective security policy and accommodates enterprise implementations among regions.

Unlike other RBAC systems, the EDAC powerful role and permission assignment technology is capable of evaluating resource access based on the following criteria:

Attributes
Environmental
Business rules
Questionnaire
Workflow

This type of meta-database access control (MDAC) evaluation capability is quickly growing as a necessary requirement. Currently customers are encountering problems with niche access control solutions that only satisfy portions of their requirements. EDAC offers a comprehensive solution with an extensible framework.

Static listings offer little in the way of hierarchal considerations or inheritance of permissions but the EDAC can evaluate inheritance on every user characteristic and environmental. EDAC can also account for corporate and user profile attribute changes on a real-time basis to determine resource access. Static listings are incapable of altering resource access based on changes due to security threats, such as Homeland Security advisory changes but the EDAC can accommodate such changes with pre-configured conditions under respective security threats.

Another challenge facing access control systems is the necessity to effectively establish policy among an enterprise. Such a task involves the participation by various resource managers (RM) working on an interactive interface where policies can be edited and reviewed before implementation.

Enterprise Dynamic Access Control Objectives

An effective authorization system such as the EDAC will offer the following capabilities:

1) Comprehensive access control features that satisfy many prevailing customer authorization requirements.
2) Web service modularity that offers customers a choice of standard interchangeable access control components from various vendors.
3) Role-based capabilities that can automate the assignment of users into proper roles.
4) Separation of duties that avoids conflict of interest.
5) Role engineering mechanism to effectively manage large scale authorization systems.

To assist the reader in understanding this section a simple authorization model is shown in figure 1. The model illustrates the basic access control procedures that determine resource access.

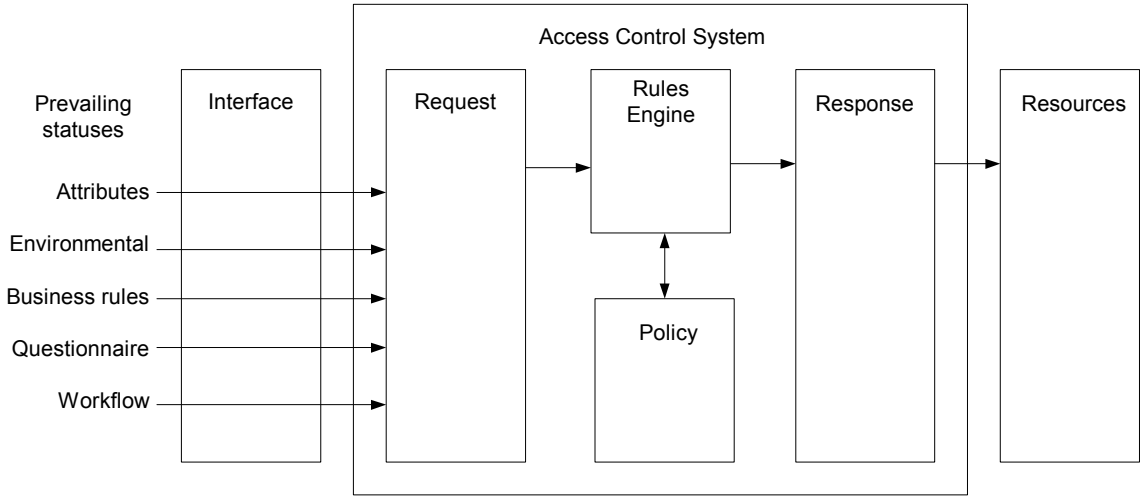


Figure 1

The *interface* represents a sensor that collects and presents the authorization system with input data. The *request* is a formatting/conditioning service that bundles input data for processing. This document assumes that request data has been properly authentication. A *rules engine* evaluates the request input against conditions inside a *policy* to produce a *response*. The response is delivered to a *resource* for user access.

Comprehensive Access Control Features

General

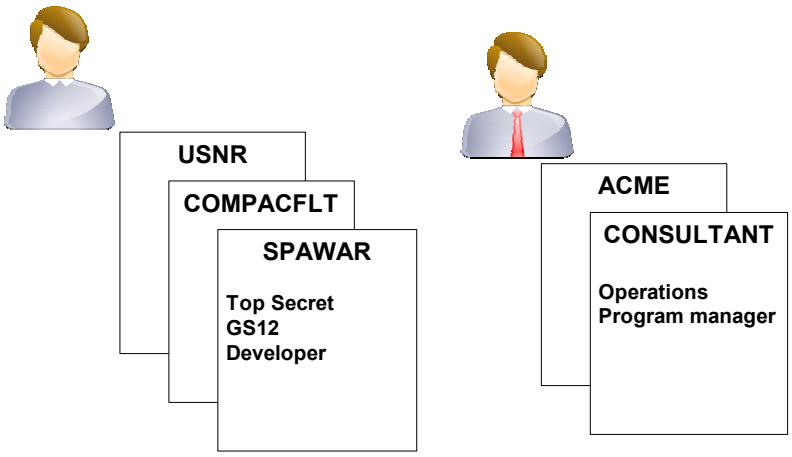
The objective of the EDAC is to furnish customers with a comprehensive access control framework that is extensible. An authorization system should be capable of evaluating a request with the following features:

User and corporate attribute changes
Environmental time constraints and security threats
Customizable business rules
Answers to questionnaires or surveys
Workflow progress

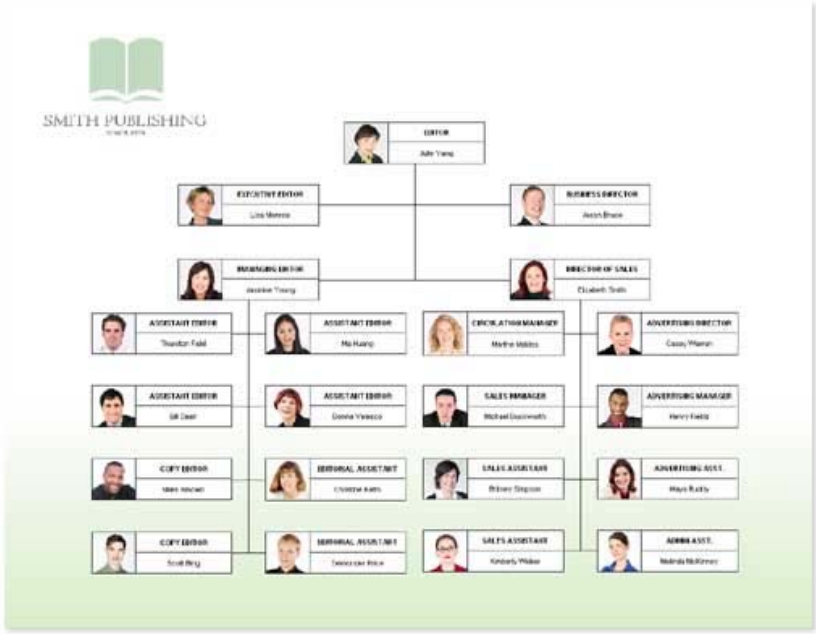
Attributes

An object will be classified as a user or thing that requires access to a resource. An **object or user profile** contains a compilation of characteristics identifying the object such as: corporate assignment, security clearance, job description and/or salary. If there is a corporate reassignment or security clearance change access to resources may be affected. Unfortunately static listings cannot accommodate such critical changes unless resource managers (RM) constantly monitor personnel records and implement immediate changes. Such a task can become unmanageable as the number of users and resources grow. Limitations to personnel records by RM enterprise-wide could compound the problem. In the EDAC model, a RM is not required to query personnel records. Instead, a RM simply establishes conditions based on user characteristics.

An effective access control system evaluates resource access based on different user profile selections and changes that can occur on a real time basis on user attributes.



Access to resources by many users can also be affected due to corporate re-structures such as: organization, job titles, relocation etc. An effective access control system should be able to monitor such changes on a real-time basis.



Environmental

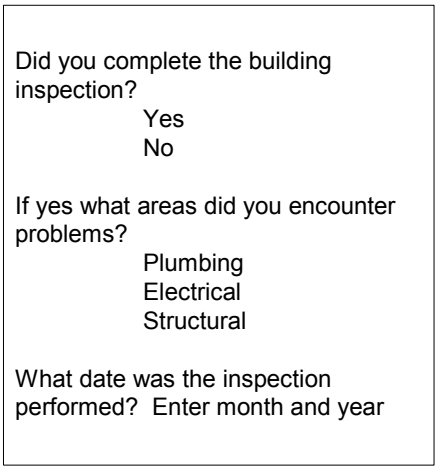
Another significant access control consideration is the evaluation of **environmental** conditions. Environmentals are non-object related events that can change over time such as: security advisories and time. Homeland Security and regional Information Assurance agencies are authorized to impose security warnings that may affect access to a wide range of resources by many personnel. Sudden changes in security conditions may not allow sufficient time to update static listings, thereby creating a possible security breach by unauthorized personnel. Finer granularity of resource access may be required during certain security levels. For example, during Homeland Security Advisories: *Severe* and *High*, only *administrator* and *superuser* account holders would be granted access to a particular resource, while all *guest* and *user* account would be denied access. An EDAC solution can accommodate these kinds of scenarios by pre-configured conditions for each respective security level. For example, if a Homeland Security Advisory changes, the EDAC only evaluates the conditions established for the prevailing security level. The EDAC can also accommodate corporate customized security advisories.

Time Constraints

Security Threat Levels

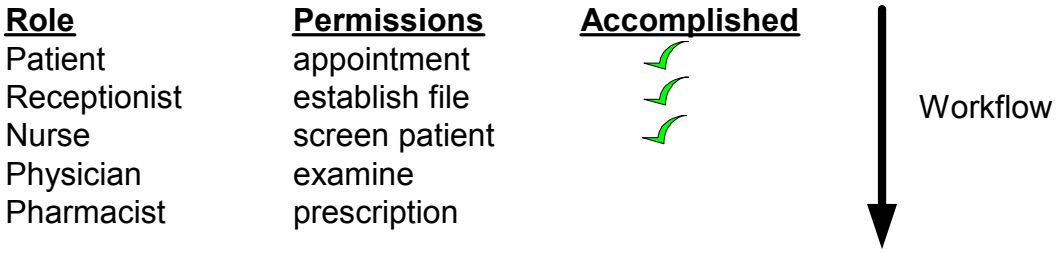
Questionnaire

Resource access can be contingent upon the answers to a survey, registration or questionnaire. Depending on the combination of answers the authorization system can direct the user to various different types of resources.



Workflow management

In integrated work environments the actions of co- workers could determine your permissions and/or resource access. For example, after certain medical personnel have processed a patient a pharmacist is allowed permission to fill a prescription.



Business rules

A business rule is an operation performed on any combination of attribute, environmental, questionnaire or workflow that produces a value used to determine resource access. The output is referred as a *complex*. A *complex* can represent a variable or Boolean value. A RM can use this output value as a condition to gain resource access.

For example, a logical and operation performed on an attribute, such as a job description and an environmental, such as a Homeland Security advisory could produce a risk assessment variable ranging from 1 - 10. The RM could select risk assessment 8 as a condition to access a resource.

	Homeland Security				
Job descriptions	Low	Guarded	Elevated	High	Severe
Program Manager	1	2	3	4	5
Developer	5	6	7	8	9
Accountant	6	7	8	9	10

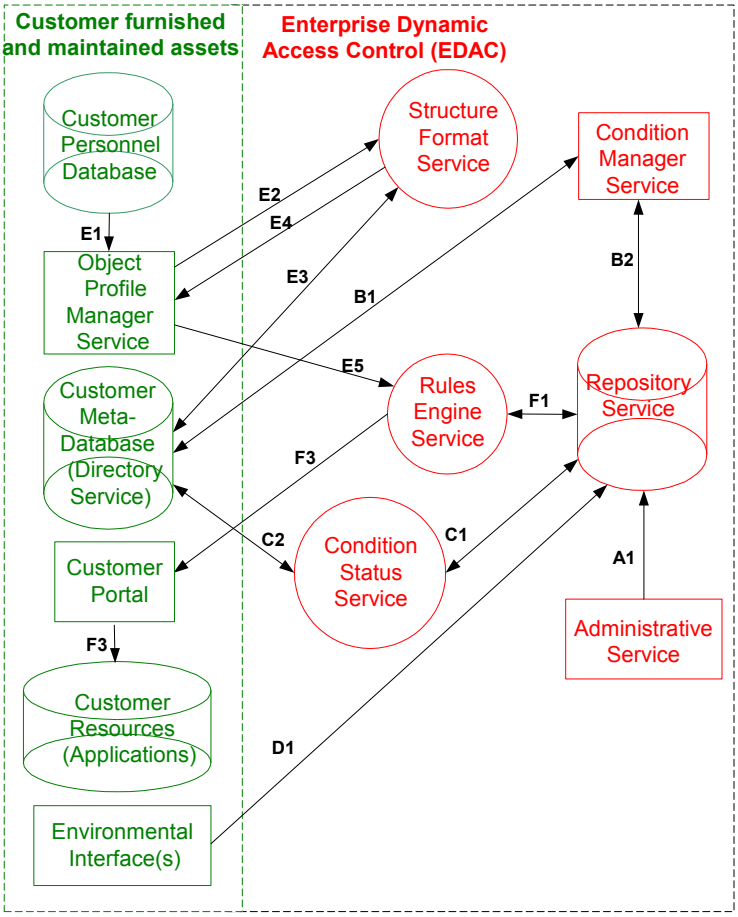
Modular Web Services

Modularity

Existing industry access control products currently require a proprietary commitment. A concern with access control proprietary solutions is the lack of standard tie-ins with customer assets. A standardless access control tie-in with customer assets leaves the customer at a disadvantage because proprietary solutions require some level of customization and maintenance. Customization also leaves the customer at risk if the servicing access control product can no longer be vendor supported. These unforeseen changes can quickly leave a customer's access control solution vulnerable. The consequences could be wide-ranging and significant since access control is tightly coupled with security. The alternative for the customer is to abandon the current access control infrastructure and replace it with another proprietary solution. This “fork-lift” approach leaves a customer with financial burdens and disruption of services. For this reason, some customer's have developed an in-house access control solution because it assures continued supportability but this option presents a significant cost.

EDAC is a modular access control framework with defined customer integration. If access control integration become standardized interchangeable access control components could seamlessly interface with customer assets. This approach would void the costs of custom coding interfaces and offer long-term support.

Figure 2 illustrates the components of an EDAC framework and how they relate with customer assets.



Processes

(A) Administrative Service- establishes resource containers, CMD referrals, RM accounts etc.

(B) Condition Manager Service - Establishes and edits conditions to access resources.

(C) Condition Status Service - listens for CMD content changes and flags unmatched reference conditions.

(D) Customer Environmental Interface - furnishes environmental updates.

(E) Customer Object Profile Manager Service - object characteristic compilation and selection.

(F) Rules Engine Service- evaluates object and conditions to determine object resource access.

Figure 2

Customer Personnel Database (CPD) represents a human resource database that contains corporate and personnel data such as employee: salary, job description, organization assignment etc.
Customer Object Profile Manager service (OPMS) queries the *customer personnel database* and presents a compilation of user characteristics referred as a user profile.
Customer Meta-Database (CMD) contains data used to establish conditions for resource access. The data content consists of structured corporate characteristics such as: salary, job descriptions, organization structures, environmental and complex data.
Customer portal interfaces with the EDAC to list accessible resources.
Customer resources represents software applications, web services, cipher locks, etc.
Customer environmental interfaces serves as an input for prevailing environmental statuses such as prevailing security levels (INFOCON, Homeland Security Advisory Levels, etc.), time, weather readings etc.
Condition Manager Service (CMS) is a web interface for RM to establish conditional access to *customer resources*.

Rules Engine Service (RES) evaluates object and conditions to determine resource access. Compares inputs such as object profiles and environmental statuses with pre-configured conditions.
Repository Service (RS) stores: resource access conditions, RM accounts and CMD connection parameters.
Administrative Service (AS) performs configuration management on content of *repository service*.
Structure Format Service (SFS) converts object profile and environmental status inputs to DN format by searching CMD.
Condition Status Service (CSS) evaluates policies to determine if a condition associated with a *customer meta-databases* should be flagged as deprecated or off-line in the *condition manager service*.

A modular concept offers an incentive for specialized markets to competitively develop high performance components at reasonable cost. Because the EDAC framework would consist of standard interfaces among components, customers would be able to select different components among various manufacturers in order to construct a turnkey access control system.

Web Service Technologies
The EDAC consists of web service components capable of seamlessly interfacing with customer assets such as: portals, human resource databases and resources. Web services simplify the communication process between components, therefore, reducing the integration and maintenance costs.

Another important requirement web services satisfy is the exchange of authorization traffic through firewalls. Since EDAC and customer assets can reside among disparate protected enclaves, a messaging protocol such as the Simplified Object Access Protocol (SOAP) is an ideal fit. SOAP is an XML messaging protocol and serves as an interface for applications to communicate over the Internet. The content exchange between components and customer assets consist of standard-based protocols. Table 1 illustrates the protocols employed between components in figure 2:

Links	**Protocols**
B1, C2, E3	LDAP v3 or DSML
D1, E5, F1, F3	SAML with a XACML payload.

Table 1

Directory services in the EDAC model communicate using the Lightweight Directory Access Protocol (LDAP) or Directory Service Markup Language (DSML). LDAP v3 is a TCP/IP network protocol used to query and edit directory service content. Directory services can be equipped with a DSML front-end that converts LDAP instructions into XML.

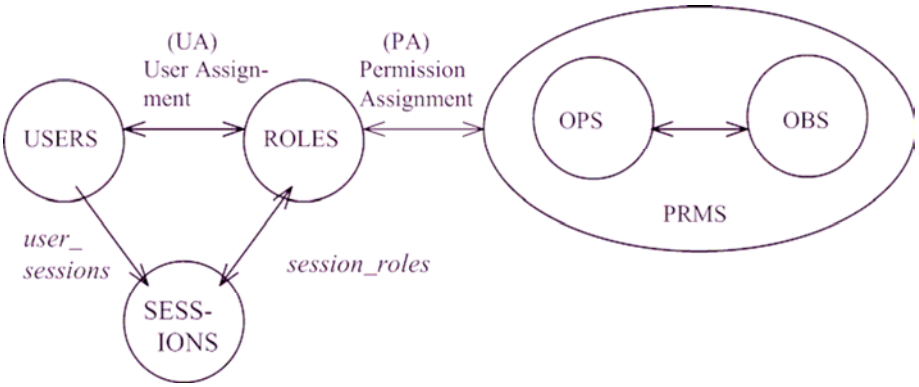
Security Assertion Markup Language (SAML) is an XML standard (as defined by the OASIS Technical Committee) for exchanging authentication and authorization data. In EDAC the use of SAML is used to transport user profiles (or other requests), policies and responses to the appropriate components for processing.

The Extensible Access Control Markup Language (XACML) is a standard (also defined by the OASIS Technical Committee) that defines an access control language. XACML is capable of generating and evaluating policies and requests to produce a response that allows or denies an object access to resources. XACML has been reviewed and designed by experts and users to make it easier to interoperate with customer resource assets. Besides a rich library of functions and attribute data type, XACML is extensible to accommodate special customer functions and attribute data types. The incorporation of XACML has made EDAC version 2 a reliable and flexible access control platform.

Role Capabilities

General
How users are assigned to roles and how roles are labeled, structured and mapped to resources is subjective and complex. This section discusses EDAC's interpretation and effective use of roles. Many figures in this section will be accompanied with a diagram of the Core RBAC model (contained in the RBAC standard) to illustrate a correlation with the EDAC design.

Flexible Resource Access
In EDAC version 1 resource access was only defined by the roles a resource contained. For example, a resource developer would imbed roles such as: administrator, user and guest within a resource. However, some resources may not have this functionality and can only be accessed by permissions such as: read, write execute operations on certain objects within the resource. XACML allows the flexibility for the user to access resources in the form of a role or permission. In figure 3 a door (classified as a resource) will be used to interpret the Core RBAC model's permissions.



Core RBAC model

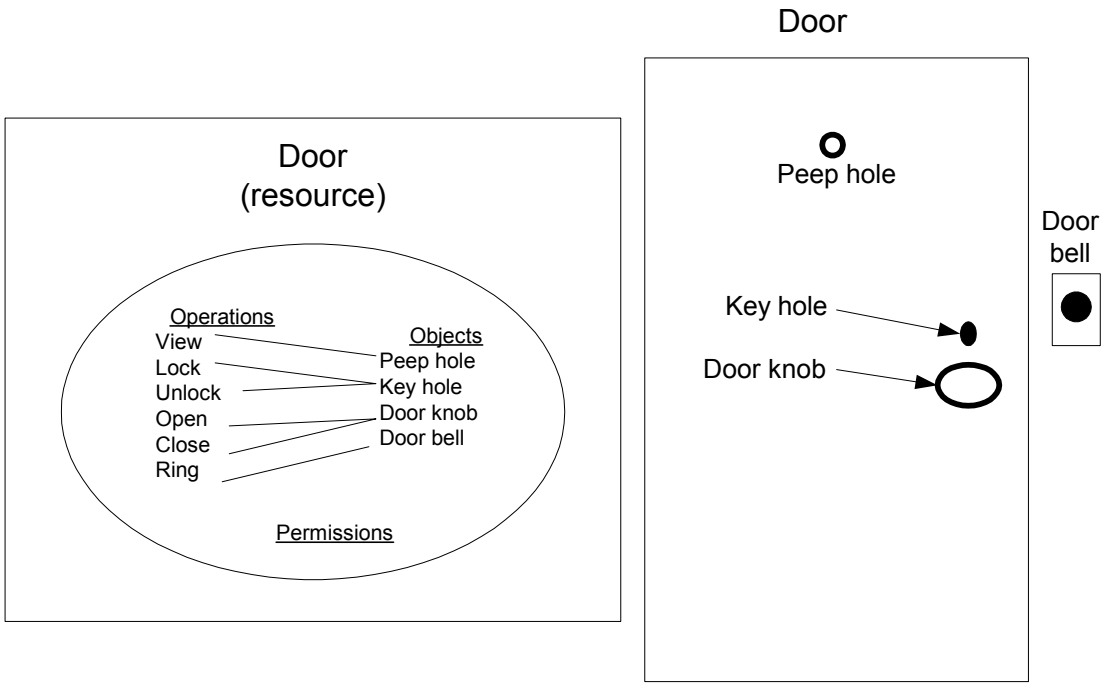
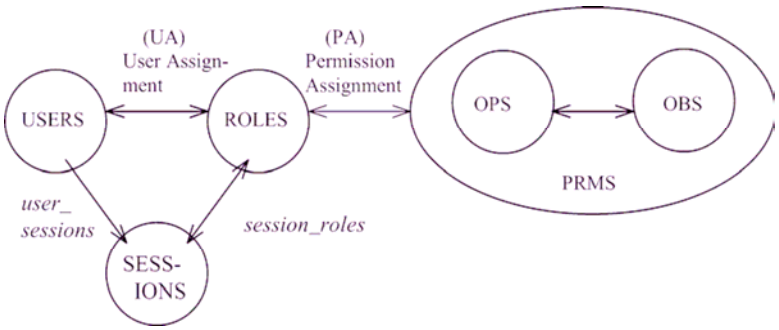


Figure 3

In this example, the door represents a resource that contains no roles only permissions. In order to assign access a RM would require the capability to associate conditions (in a policy) to specific sets of operations and objects, which make up a permission. The authorization system would perform the evaluation and automatically assign the user to the specific permission.

Most resources have embedded roles to abstract the complexity of operation/object assignments. These embedded roles that come with these resources will be classified in this document as *resource roles*. In figure 4, the resource can be access by three types of *resource roles*: admin, user and guest.



Core RBAC model

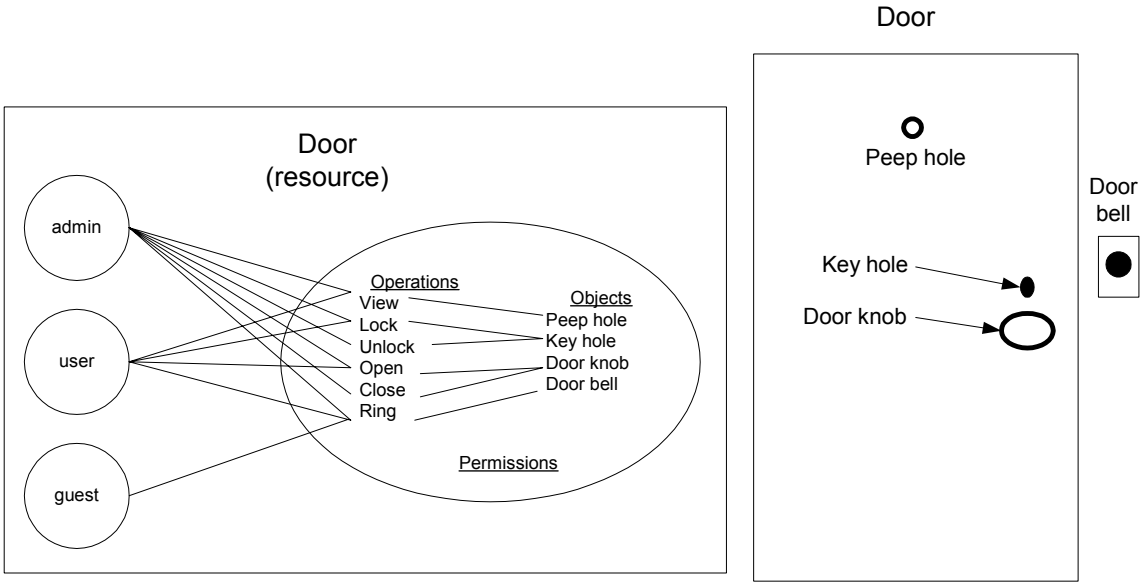


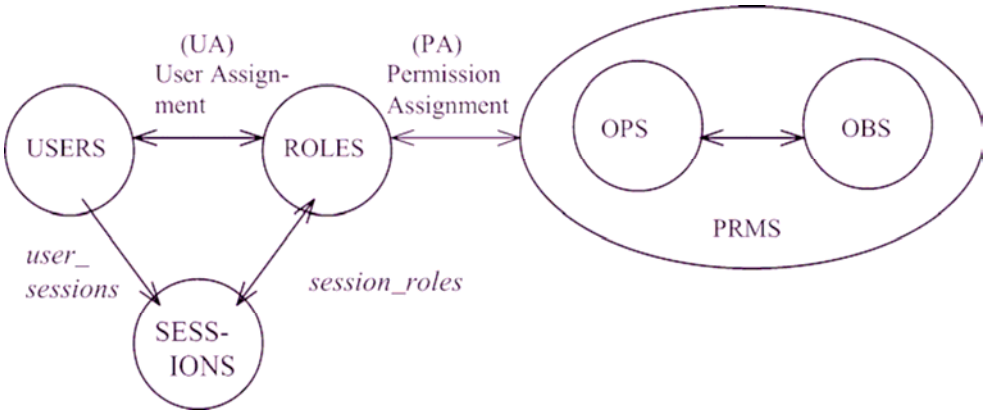
Figure 4

Access Control and Resource Roles

Continuing with the example in Figure 4, the RM should be allowed to create roles within the authorization system that map one-to-one to the *resource roles*. These roles created within the authorization system will be referred as *access control roles*. As a minimum the initial set of access control roles should:

1) always be mapped one-to-one to *resource roles*
2) be named the same as the mapped resource role

This important practice avoids confusion if access control roles are later extended within the authorization system. Refer to figure 5



Core RBAC model

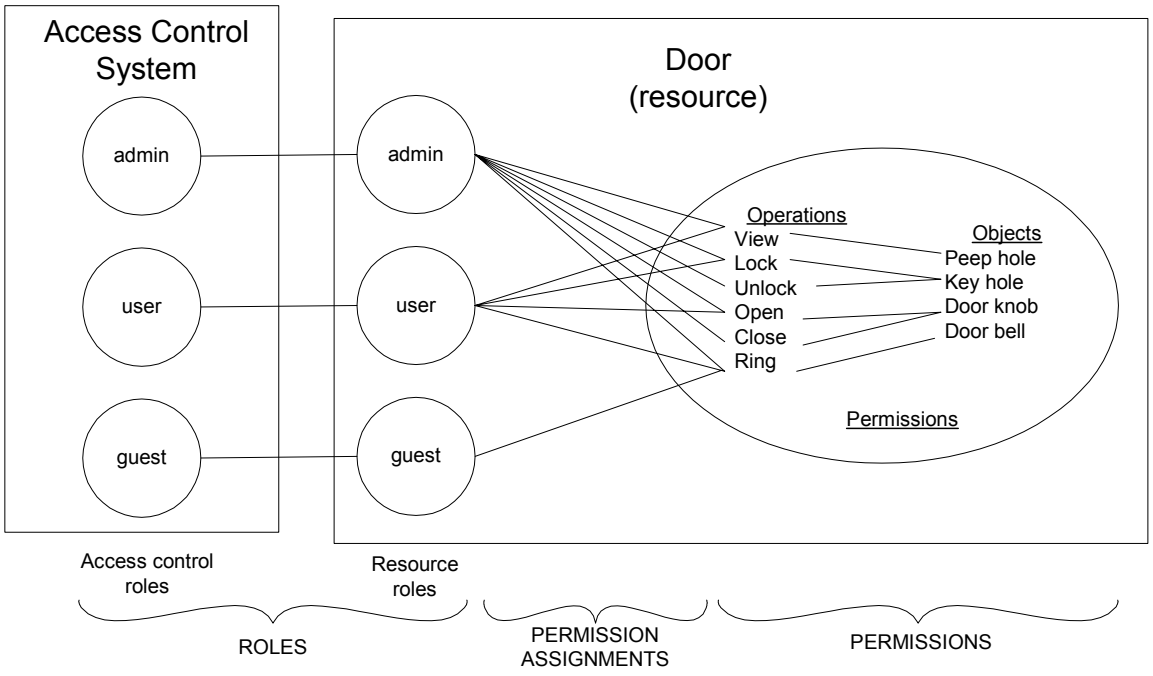
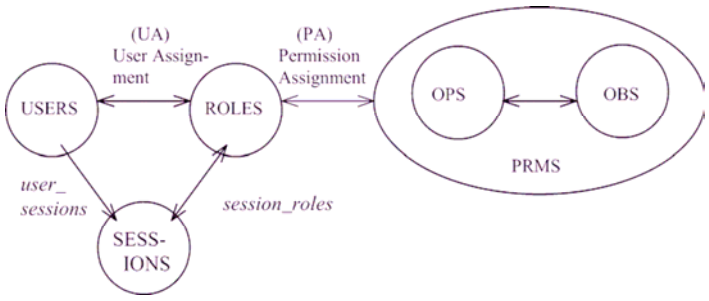


Figure 5

Extension of Access Control Roles

In a small customer base scenario the authorization system can assign users directly to the mapped access control roles. However, in large customer scenarios, involving many users an authorization system should be capable of extending the access control roles to accommodate customer business logic requirements. For example, a business logic layout offers the RM the capability to establish enterprise access among various resources. Refer to figure 6.



Core RBAC model

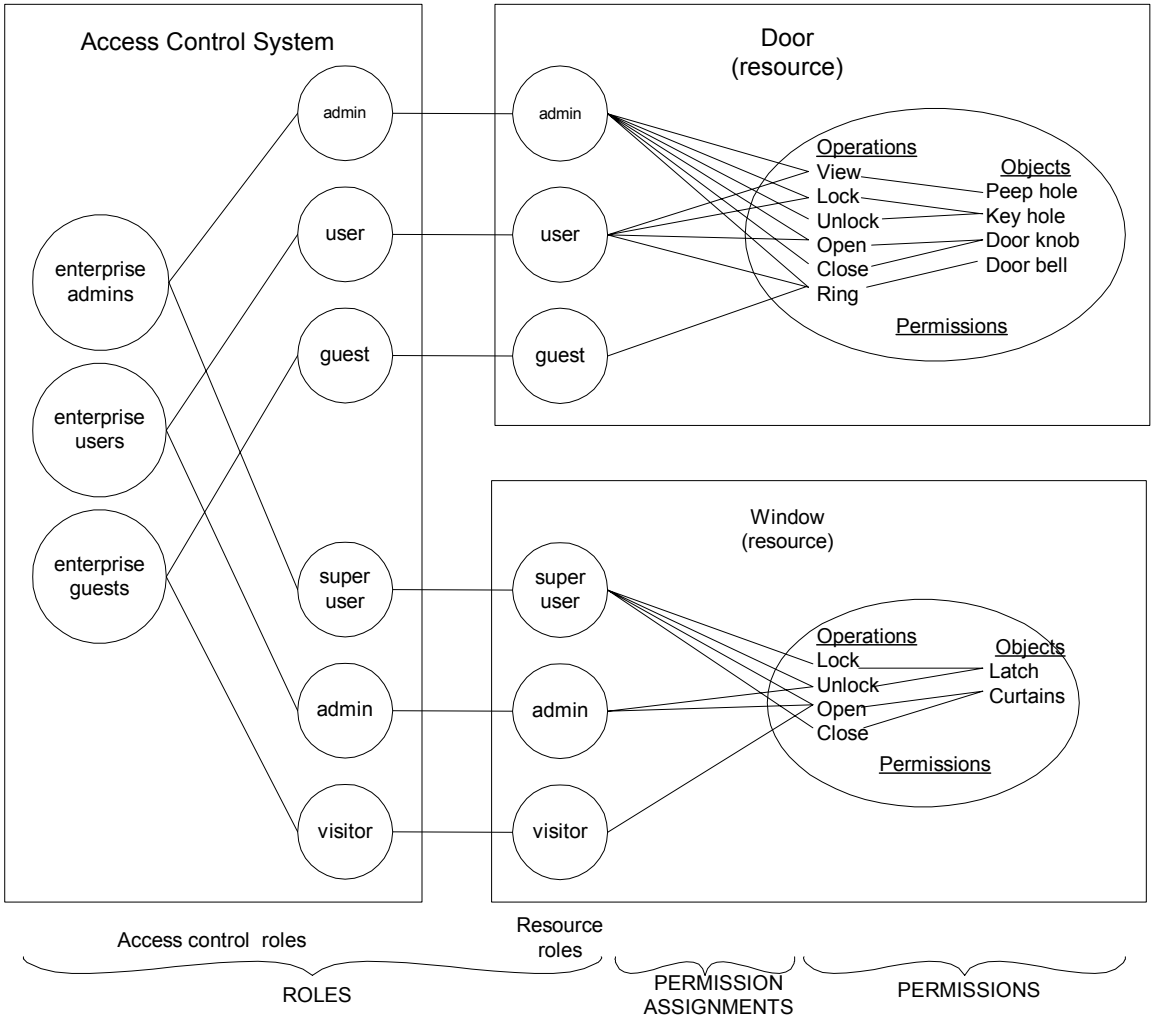


Figure 6

Role Hierarchy

The extension of access control roles may also involve role hierarchy as illustrated in figure 7. Although role hierarchy is not part of the core RBAC standard, a taxonomy of roles should be available for customers.

The RBAC standard defines role hierarchy as either *limited* or *general*. A limited role hierarchy consists of a train of roles with each parent containing only a single child. In a general role hierarchy a parent can have more than one child.

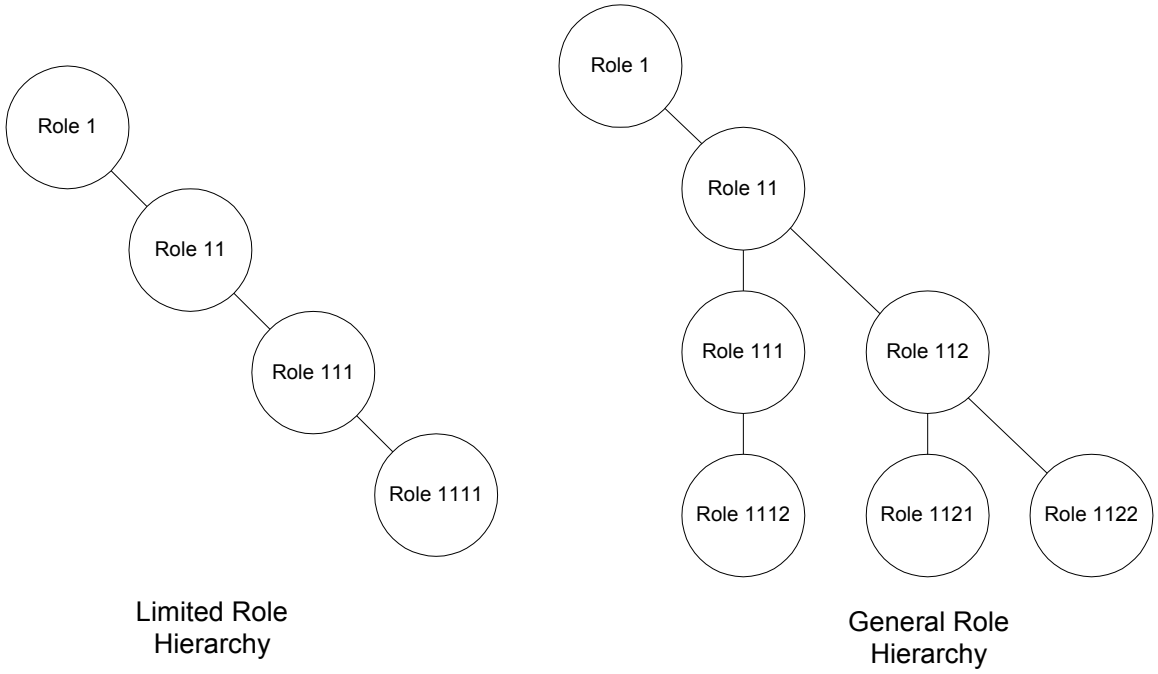
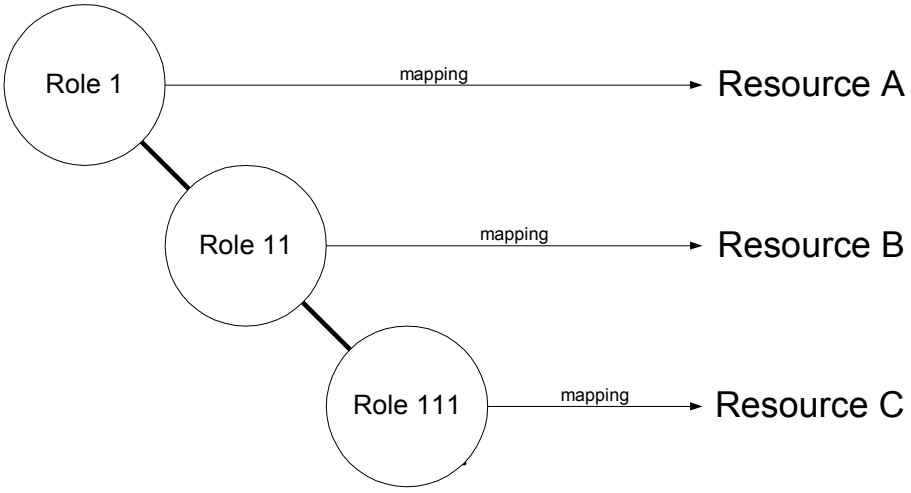


Figure 7

A general role hierarchy is the hierarchy of choice for enterprise environments with leaf roles statically mapped to resource role(s) or permissions. How rights are inherited in a hierarchy is subjective but the NIST suggest resource permissions consist of a subset of inherited roles. This means that resource access is based on a role's position in a hierarchy. Refer to figure 8.



Users	Assigned Roles	Accessible Resources
Bill	Role 111	C
Jane	Role 11	B, C
John	Role 1	A, B, C

Figure 8

User Assignments to Access Control Roles
It is important to clarify that assignment of users into access control roles can be performed by any combination of the following request-based criteria:

Attributes of users and corporations
Environmental
Customizable business rules
Answers to questionnaires or surveys
Workflow progress

The above list of request-based criteria is not all-inclusive and can be further extended to with a customer furnished criteria. Therefore, access control roles should not be labeled after a request-based parameter such as an attribute that describes an organization department or vocation. This common practice locks the authorization system to a specific request based criteria. A better practice labels access control roles based on resource mappings and role/permissions.

Continuing with our example in figure 6, the access control role *enterprise admin* is associated with the resource versus any user or request-based parameter. The word

enterprise refers to the mapping among various resources and the word *admin* alludes to a type of resource role or permission.

In the following example the assignment of John Doe to the *enterprise admin* role can now be based on any combination of request-based criteria such as: attributes, environmental and questionnaire. Refer to figure 9.

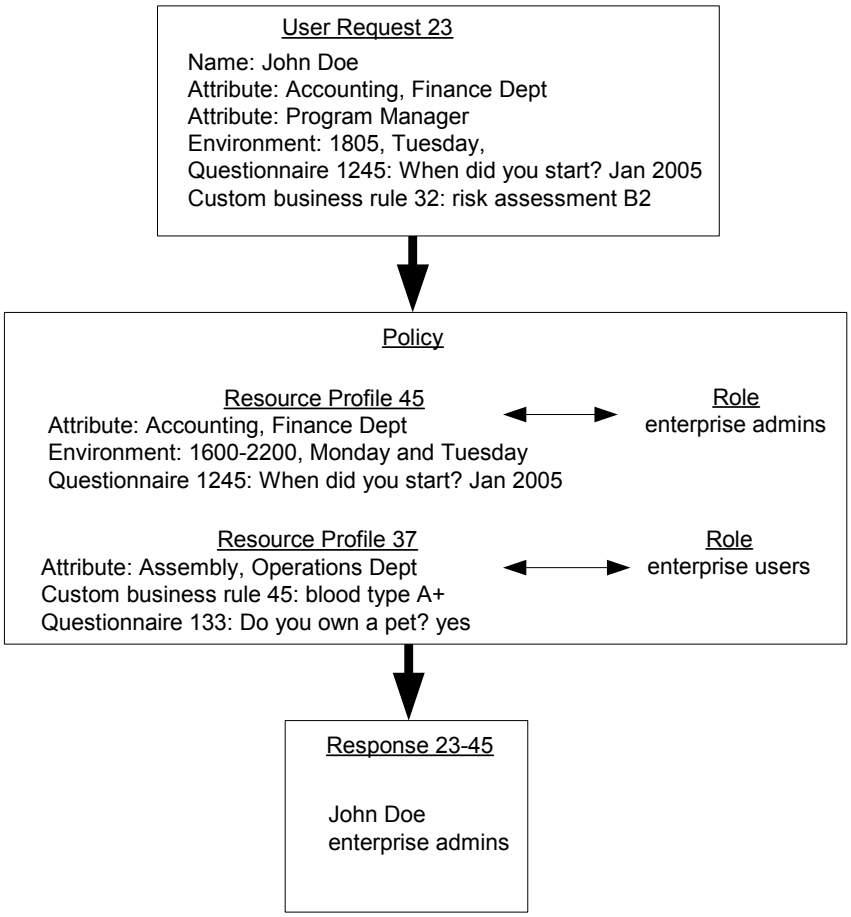
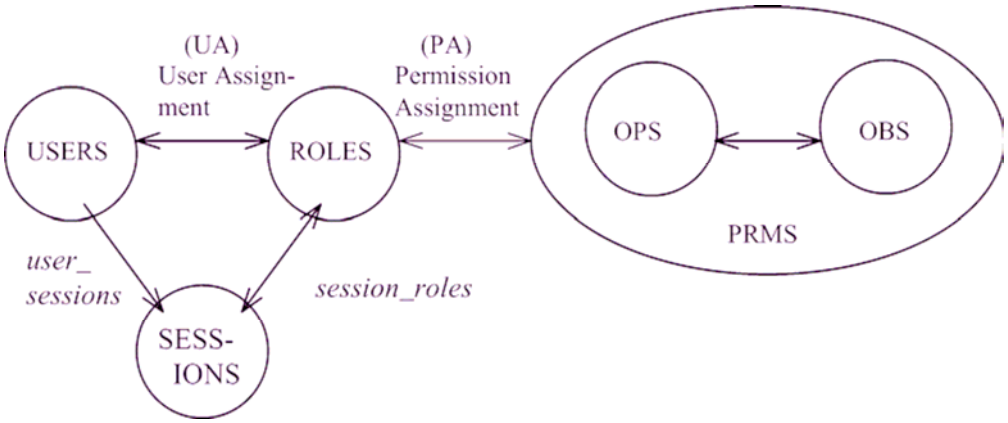


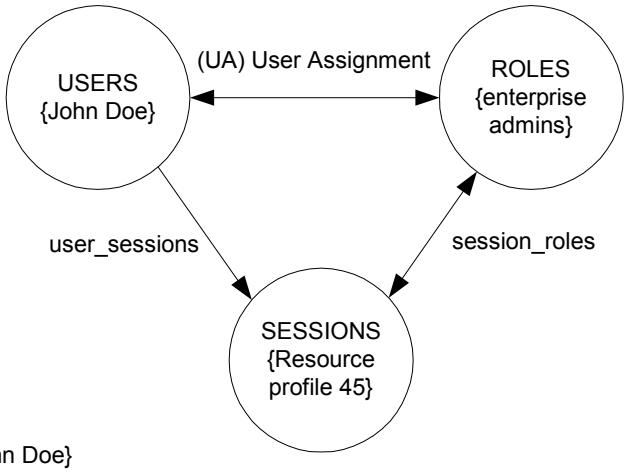
Figure 9

Therefore, naming the role *accounting* or *program manager*, which represent an attribute type request is misleading because it will not be descriptive of other request-based criteria such as environmental and questionnaire that also determine user assignment to the role. For this reason labeling a role based on the resource such as *enterprise admins* and not on a request would be appropriate.

Moving to the left portion of the core RBAC model we notice the importance of the sessions in the assignment of users to roles. The session simply implements the policies by filtering requests to determine role assignments of users. Continuing with the previous example refer to figure 10.



Core RBAC model



where:
USERS ∈ {John Doe}
SESSIONS ∈ {Resource profile 45}
ROLES∈ {enterprise admins}

user_session = John Doe/Resource profile 45
session_roles = resource profile 45/enteprise admins
user assignment = John Doe/enterprise admins

Figure 10

For a detailed explanation on this topic refer to the EDAC Compliance with the NIST RBAC Standard ANSI/INCITS 359 in NIST RBAC Standards Roadmap web site.

Case Study of Role Extension and User Assignments

This case study will demonstrate how access control roles are labeled and laid out in an effective authorization system. Recall that the resource mapping determines the role label and the customer's business logic dictates the structure (or hierarchy) of roles.

Lets suppose a smart home requires authorization to access doors, windows, alarms and a range/oven. In this scenario the following household users require various degree of access:

Children
Chef
Technician
Parents

The access control roles in this example will have enterprise (spanning more than one resource) and hierarchal capabilities in order to fulfill the following household business logic requirements:

Emergency access
Maintenance
Routine living
Overall administration

Refer to figure 11.

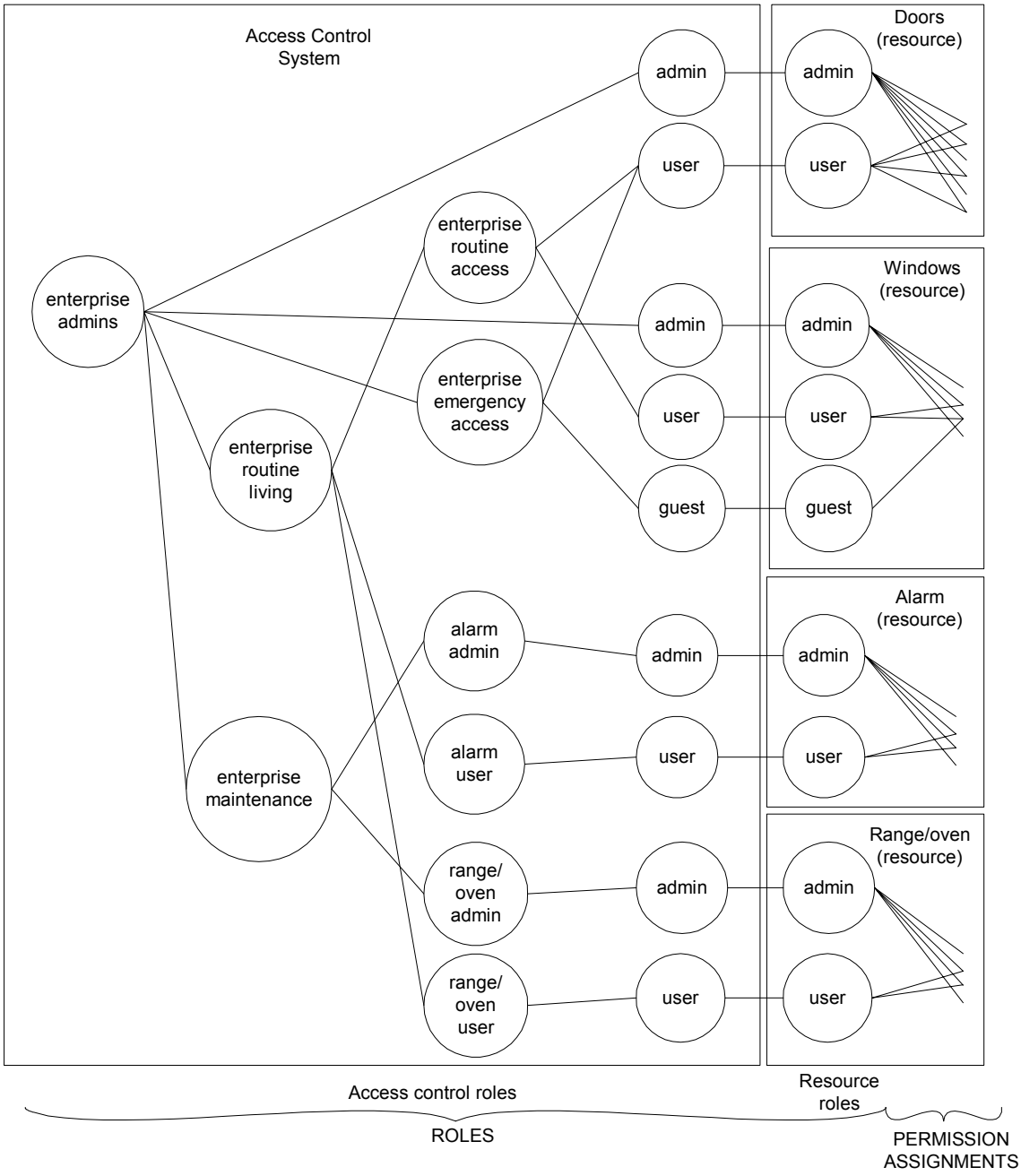


Figure 11

Notice how the classification of access control roles is resource based versus request based.

For example, table 2 shows a household policy, which assigns household users to the access control roles illustrated in figure 11. In this policy users are assigned to roles

based on various request-based parameters such as: attribute, environmental, workflow and questionnaire.

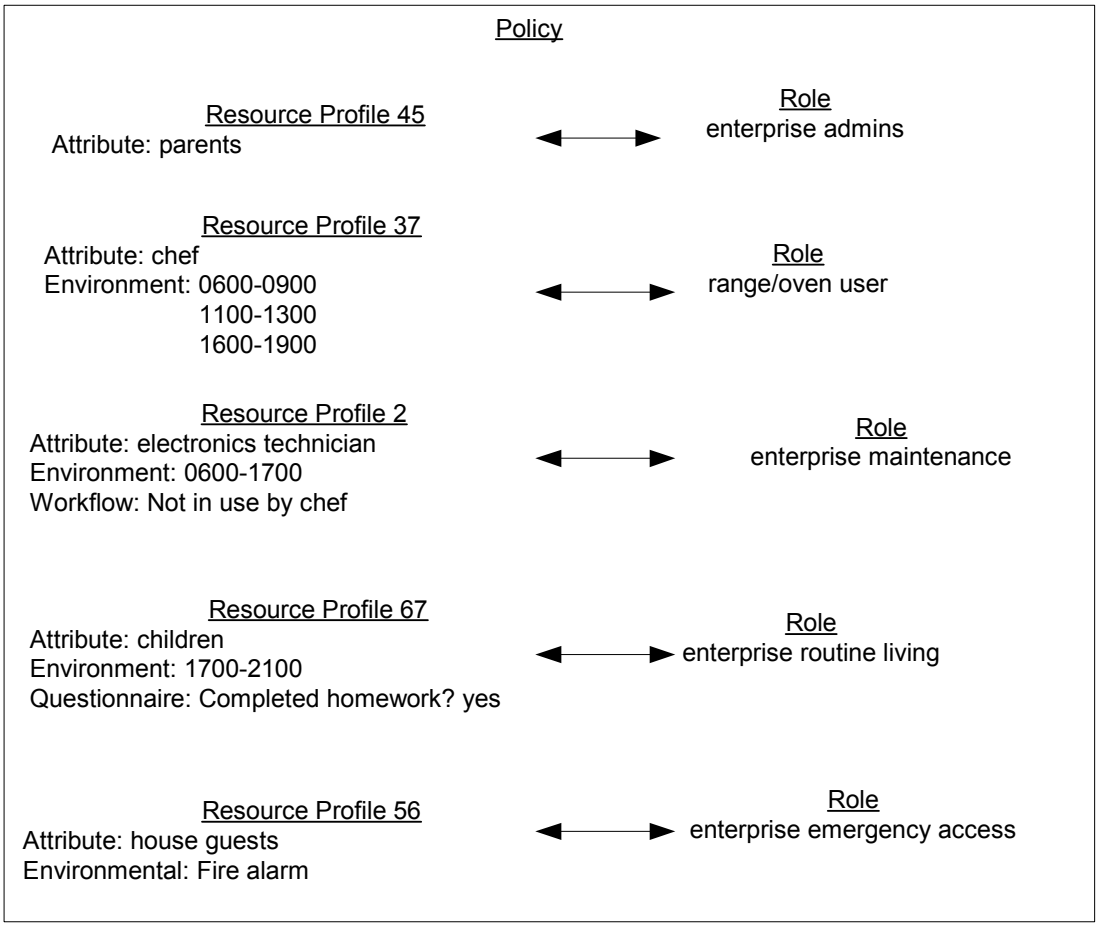


Table 2

In figure 11 and table 2, children are assigned the *enterprise routine living* role, which grants more privileges than houseguests, who only require entry and egress access of the home resource. In addition, the *enterprise routine living* role also grants children access to the range/oven and alarm resources. Technicians, who work with appliances and alarm systems are assigned the *enterprise maintenance* role. The parents, assigned to the *enterprise admin* role, require all forms of access on all home resources. This capability is automatically inherited by other roles such as the *enterprise routine living* and *enterprise maintenance*.

Request-Based and Role Hierarchy Distinction

Resource managers should understand the important distinction between role and request-based hierarchies. Although tempting, a resource manager should avoid the paradigm of creating a role hierarchy to resemble a corporate organization structure or job titles listing. Such request-based hierarchies used to organize roles limit the effectiveness of an authorization system. For this reason the labeling and/or classification of roles should be associated with resource permissions and mappings. In the previous example, roles were not classified by user attributes, such as: *parents, children, technicians, houseguest* and *chef*. Instead the roles were named as: *enterprise routine living* or *enterprise maintenance.* The words "*routine living*" and "*maintenance*" described resource permissions (in general terms) and the words "*enterprise*" describe the map scope to multiple resources.

To emphasize the distinction between request-based and role hierarchies refer to figure 12.

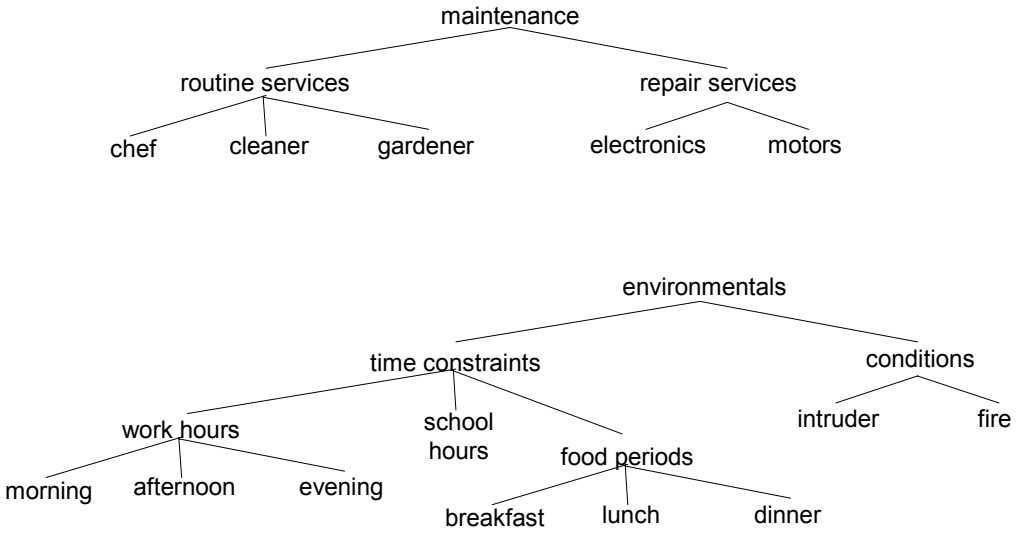


Figure 12

Figure 12 illustrates two request-based hierarchies: user attribute and environmental. Figure 13 illustrates an access control role hierarchy:

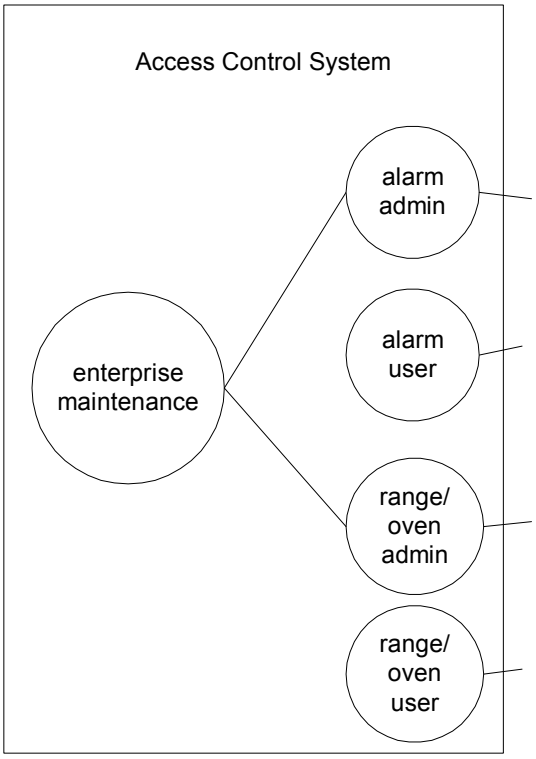


Figure 13

Notice the distinction between the two hierarchies. An effective authorization system treats **request-based hierarchies and access control roles hierarchies totally separate.**

The EDAC system uses request-based hierarchies to automate the assignment of users to access control roles and access control role hierarchies are statically mapped to resource role/permissions to match a customer's business logic.

Continuing with the previous example notice how users (a chef and repair person) are assigned to access control roles based on request-based hierarchies. Refer to figure 14.

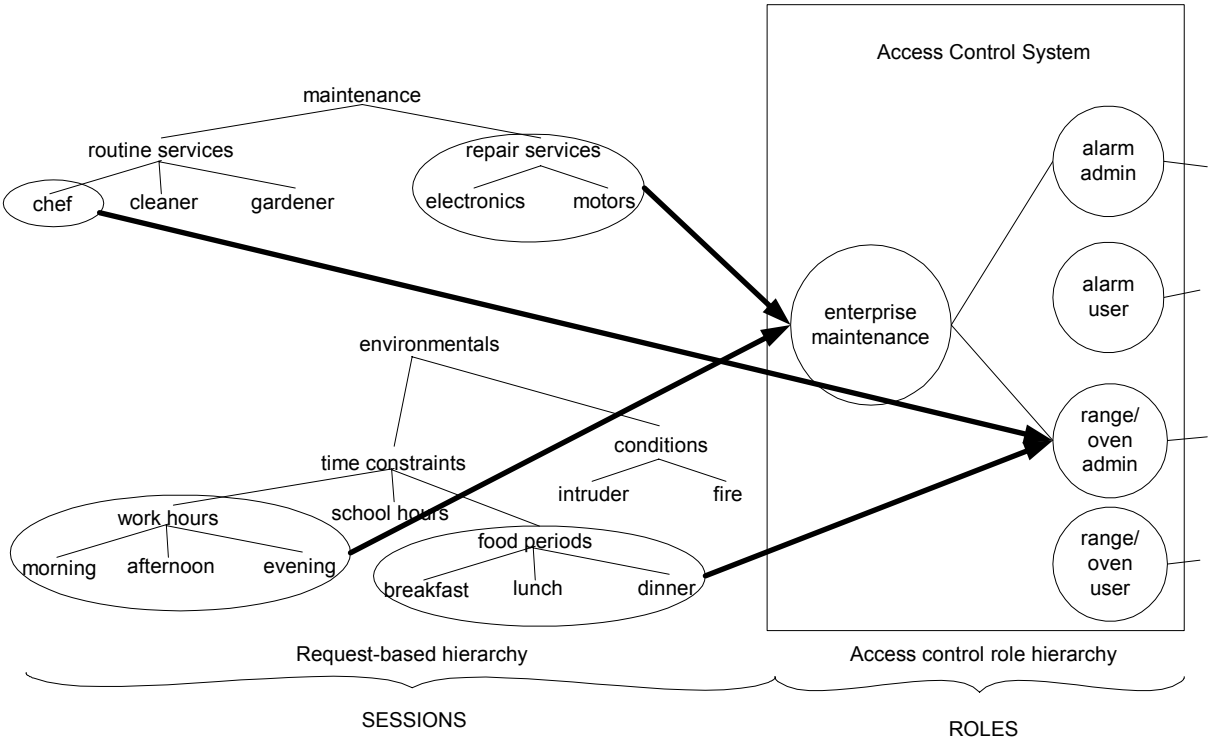


Figure 14

Separation of Duty (SoD)

The RBAC standard defines SoD as a critical operation(s) divided among two or more people, so that no single individual can compromise security. Essentially a SoD assures no conflict of interest occur among a user accessing resource(s). Although not part of the core RBAC standard, SoD feature should be available on authorization systems. The RBAC standard defines two types of SoD: Static SoD abbreviated (SSD) and Dynamic SoD (DSD). To differentiate between SSD and DSD assume there are two roles: A and B that are classified as SoD. With SSD a user assigned to role A cannot be assigned to role B and vice versa. In DSD a user can be assigned to role A or B but not both simultaneously. DSD offers more flexibility and is the recommended approach in EDAC.

The first step in establishing a SoD is to understand the resource and it's associated roles/permissions. A resource can contain multiple groups of SoD with each group of SoD consisting of multiple resource roles/permissions. Once the SoD(s) are identified the RM must annotate the corresponding roles mapped to the authorization system. Figure 15 illustrates a DSD.

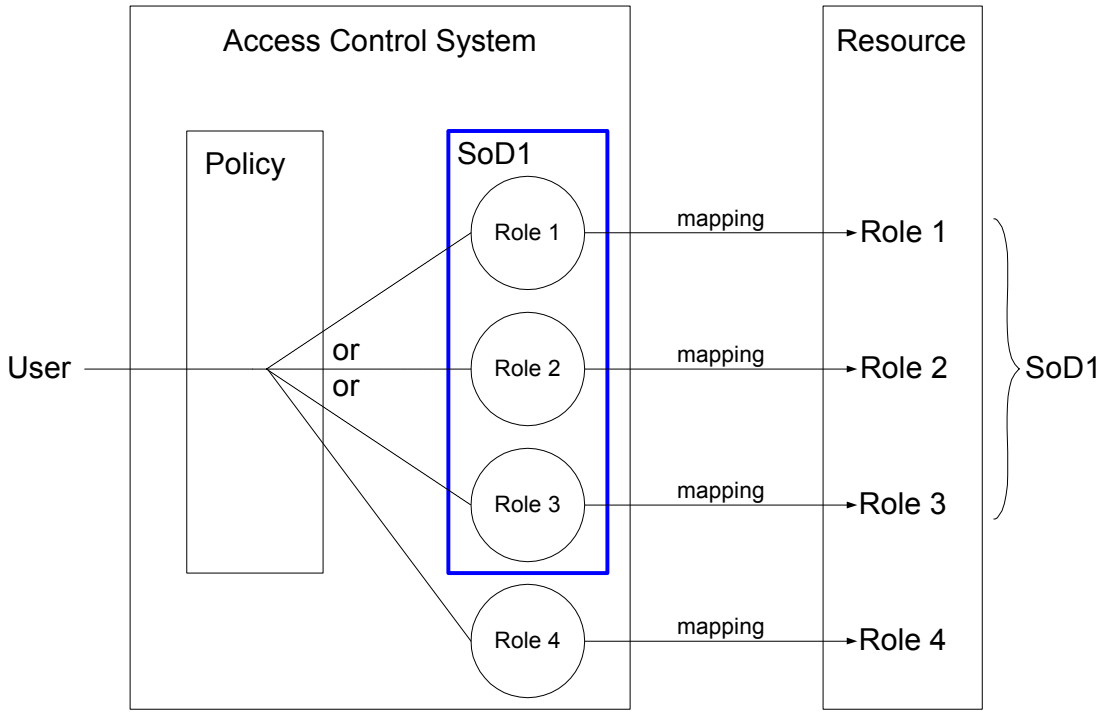


Figure 15

A DSD capability would require the resource to notify the authorization system whenever a user has terminated a session. This would allow the possibility for a user (depending on the policy) to be assigned to another resource roles/permissions within the same SoD group.

Complications can arise with access control hierarchies. If children within the hierarchy are mapped to a group of SoD resource roles/permissions a conflict of interest will occur. Refer to figure 16.

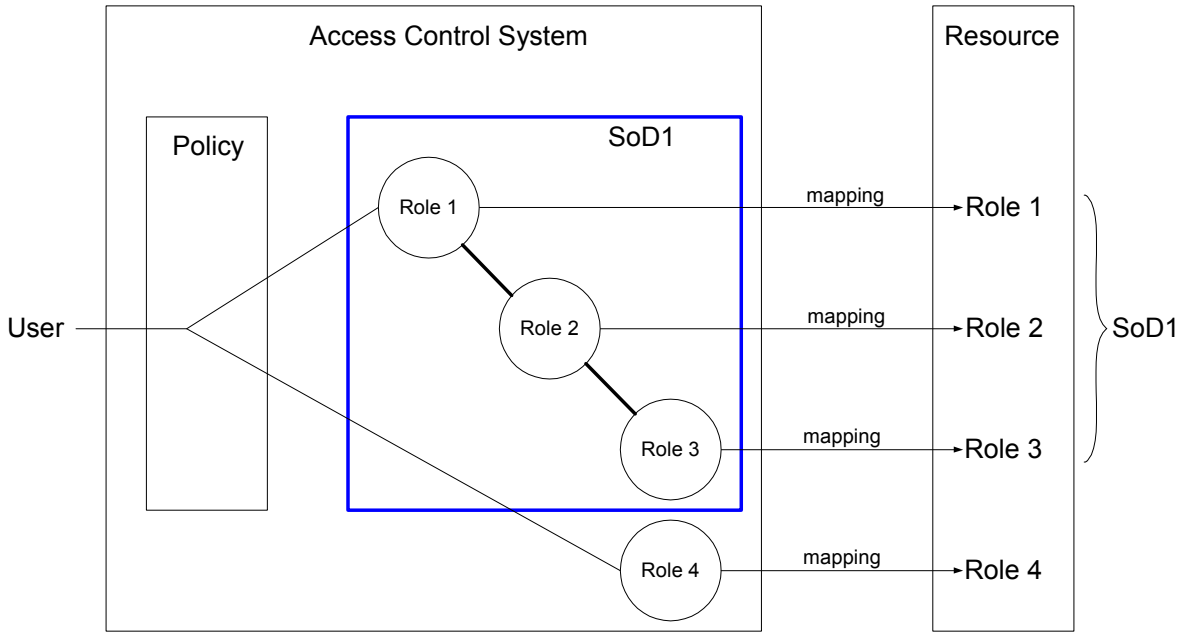


Figure 16

In a hierarchal scenario the authorization system should be capable of assigning only one role within the SoD group, preferably the directly mapped resource role/permission. Reference table 3 below for details.

Assigned Access Control Roles	Accessible Resource Roles w/o SoD	with SoD
Role 1	Role 1, 2, 3	Role 1
Role 2	Role 2, 3	Role 2
Role 3	Role 3	Role 3

Table 3

Role Engineering

Role engineering is defined as the capability by a staff of RM to delegate, review, execute and audit an access control policy. This staff of RM may consist of various backgrounds, such as information assurance, managers and developers. In addition the RM staff may work at different locations. The next generation of authorization must support a nexus of role engineering functionality.

The **condition manager service** (**CMS**) serves as the interface where RM can establish policy and where the role engineering functionalities will be required. In an enterprise environment a decentralized approach will be necessary to establish policy. The advantages of a decentralized scenario are three fold. First, in a centralized scenario RMs

may not co-located among users and de-coupled in understanding parochial access control requirements. Second, a decentralized scenario alleviates the purpose of dedicating a full-time staff of RMs from performing policy tasks. In a decentralized environment more personnel are involved in the resource management process but participation is part time. Third, because of a policy review capability in the CMS no single RM can be burdened with mistakenly implementing a security flaw. The review process spans the accountability among all levels of resource management thereby ensuring a sound policy execution.

In figure 17 a group of RM are accountable for establishing policy for the Accounting branch. Notice the level of granularity the further the policy gets delegated. Although the Accounting supervisor will understand more about access requirements, accountability extends up the chain of command due to a review process.

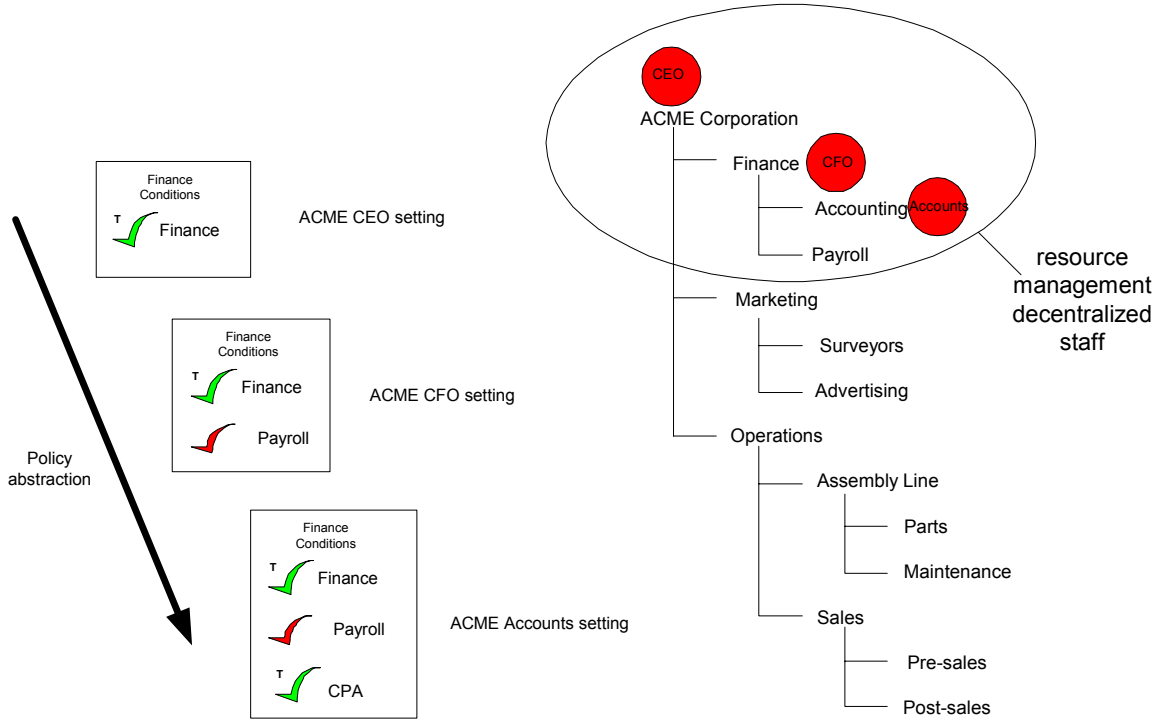


Figure 17

As part of the delegation capability, constraints can be placed on subordinate RM on how they can establish policy. For example, subordinate RM could be prevented from establishing certain types of conditions on a policy. In some instances, depending on the customer, role engineering can evolve into a complex administrative functionality.

Enterprise Dynamic Access Control (EDAC) Overview

References

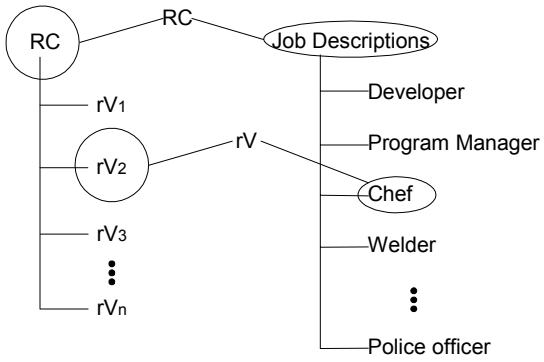
General

References are used to describe user and corporate attributes, environmental, business rules, questionnaires and workflow. Because references are used to construct requests, policies and responses, references should accurately identify an access control object's description, data types, structure and state. The level of effort provided to this task will reflect the usefulness of the access control system.

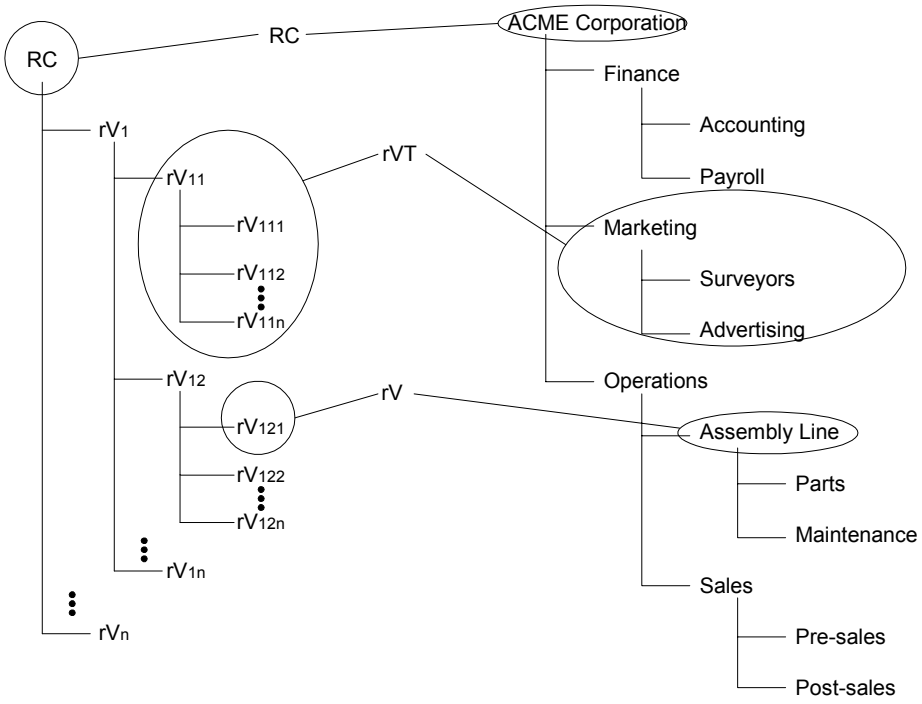
References consist of reference categories (RC) or attributes and corresponding reference values (rV) or attribute values.

Reference structure

References can be structured in the form of a list or hierarchy as shown in figure 18.



Listing



Hierarchal

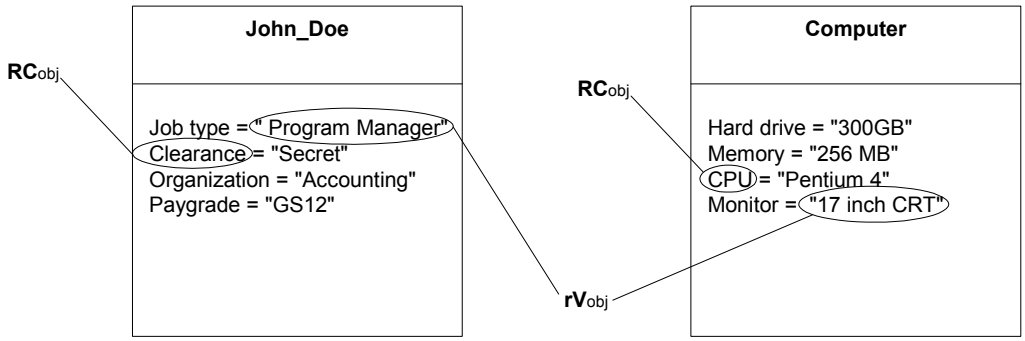
Figure 18

In a hierarchy a subtree of references is denoted by the rVT. For example, a rVT = Assembly Line will consist of the following rV:

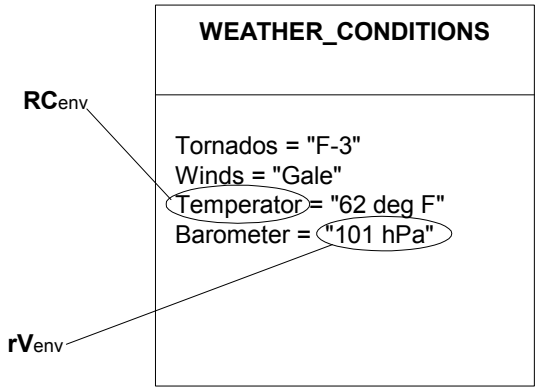
Assembly Line
Parts
Maintenance

References used as a request

A reference can be in the form of a request such as a user profile or an environmental status as shown in figure 19.



Object or User profile



Environmental status

Figure 19

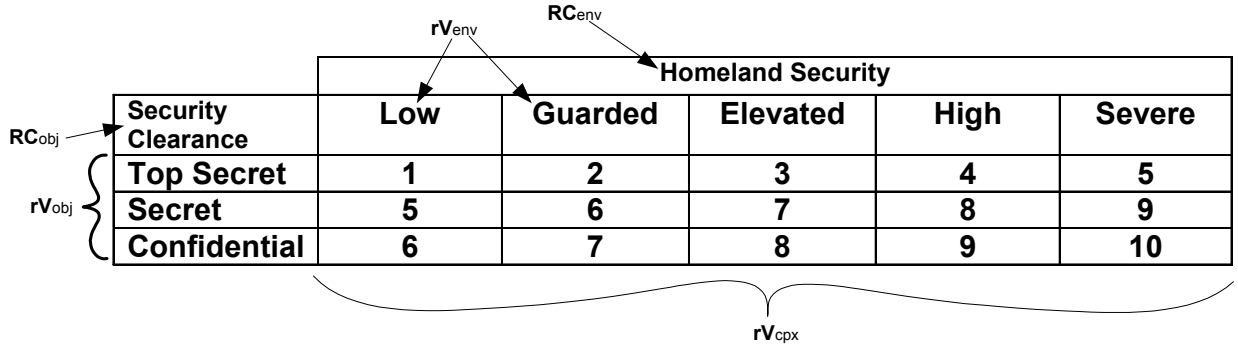
References used as a conditions

References are also used to describe conditions within a policy. References conditions are established by a RM(s) and evaluated by the EDAC **rules engine service** (**RES**) to determine resource access. Reference conditions can consist of: answers to questionnaires, business rule outcome, an organization department and a Boolean value for a particular workflow task.

Reference description

References are abbreviated by what they describe. In the example below a business rule condition contains environmental, object and complex references. The complex reference is used to describe the risk assessment outcome of a business rule and abbreviated "cpx". Figure 20 illustrates a risk assessment RCcpx containing various rVcpx ranging from 1 – 10.

COMPLEX output with a variable output. COMPLEX variable outputs do have reference categories and reference values. A resource profile that contains a COMPLEX points to the COMPLEX operation. The operation will return reference category and corresponding reference value(s).



Here is the COMPLEX operation that produces a variable output.

Figure 20

A complex operation could produce a Boolean outcome instead of a variable.

Customer Meta-Database (CMD)

CMDs offer RMs the capability to remotely select reference conditions from synchronized, centrally managed and reliable data stores. These data stores can consist of directory services, relational databases or an XML file. Most organizations store information about employees and corporation in relational database(s). These databases are referred as **customer personnel databases** (**CPD**) or human resource databases. This data can be transposed to a structured format and placed in a directory service referred as a CMD. In some instances CPD and CMD are one in the same, such as a directory service containing a corporate organization structure. CMD are own, maintained and operated by the customer. Refer to figure 21.

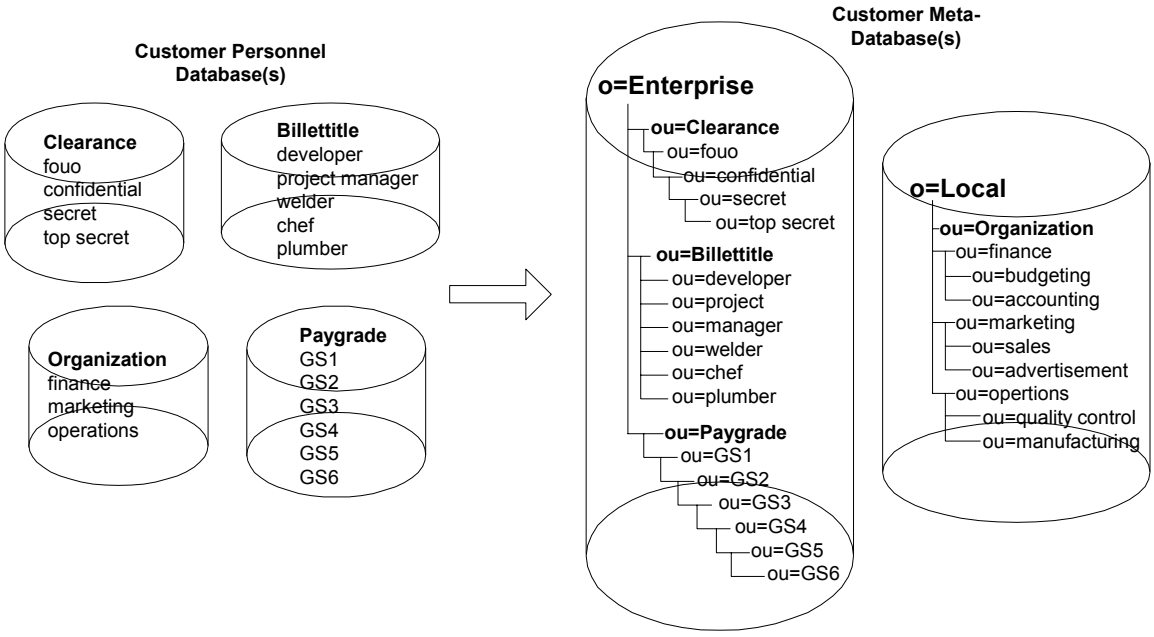


Figure 21

References can be distributed among many CMDs and managed by various organizations within a corporation or command. References can be maintained locally or globally within a community of interest. Figure 22 shows how a corporation could manage their own CMD domain. The corporation can consist of regional offices at different locations while the corporate headquarters can manage the global CMD(s).

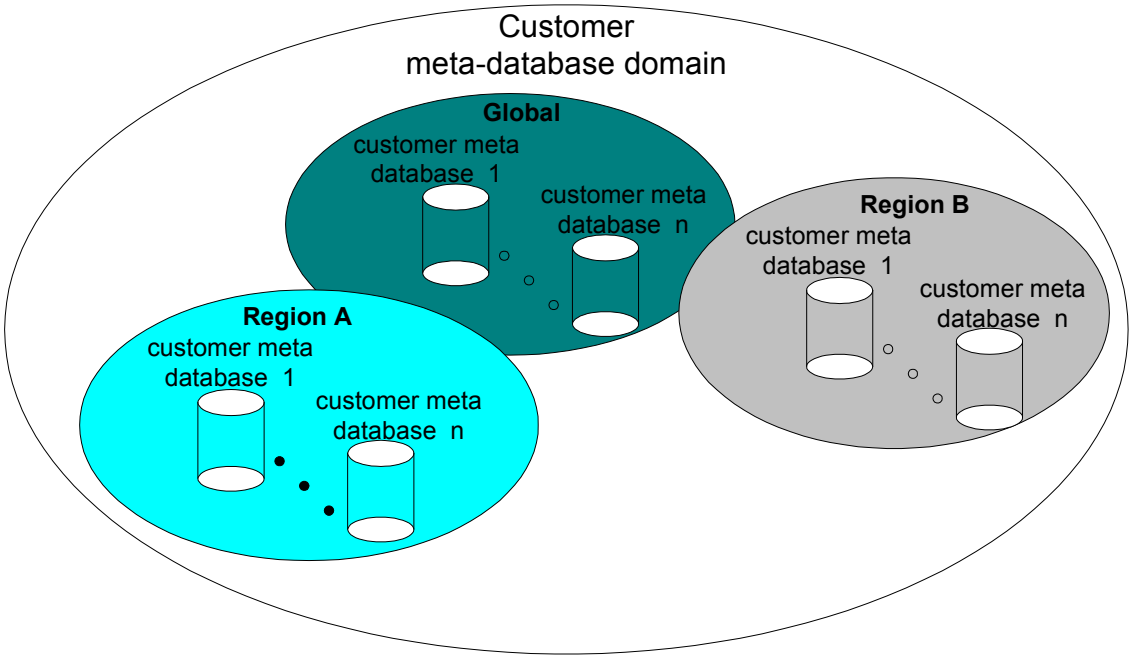


Figure 22

Resource Profiles

A **resource profile** is a container consisting of reference conditions that when evaluated with a reference request (such as object profile and/or environmental status) will determine if an object is allowed or denied access to a resource. There are two types of resource profiles: allow or deny. Before being evaluated all references (request or policy) must be exist in a structured format. For example, in figure 23, the selected reference conditions from a directory service CMD are structured in distinguished name (DN) format.

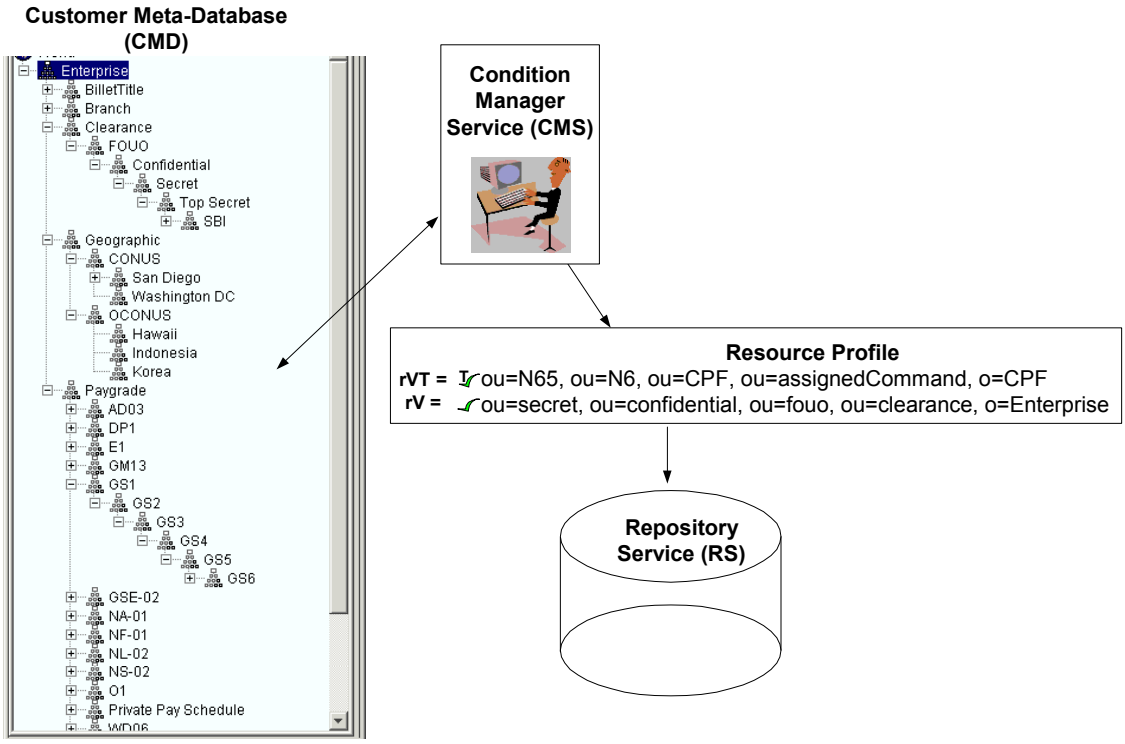


Figure 23

Resource access is granted if an object profile and/or environmental status match all reference conditions in an allow resource profile (ARP). A deny resource profiles (DRP) is optional and represents a filter for an ARP. This feature alleviates a RM from selectively establishing ARPs within a wide scope that excludes certain portions. For example, in figure 24 a RM allows resource access to all personnel who belong to the ACME Corporation but excludes the only the Finance and selected portion of the Sales departments.

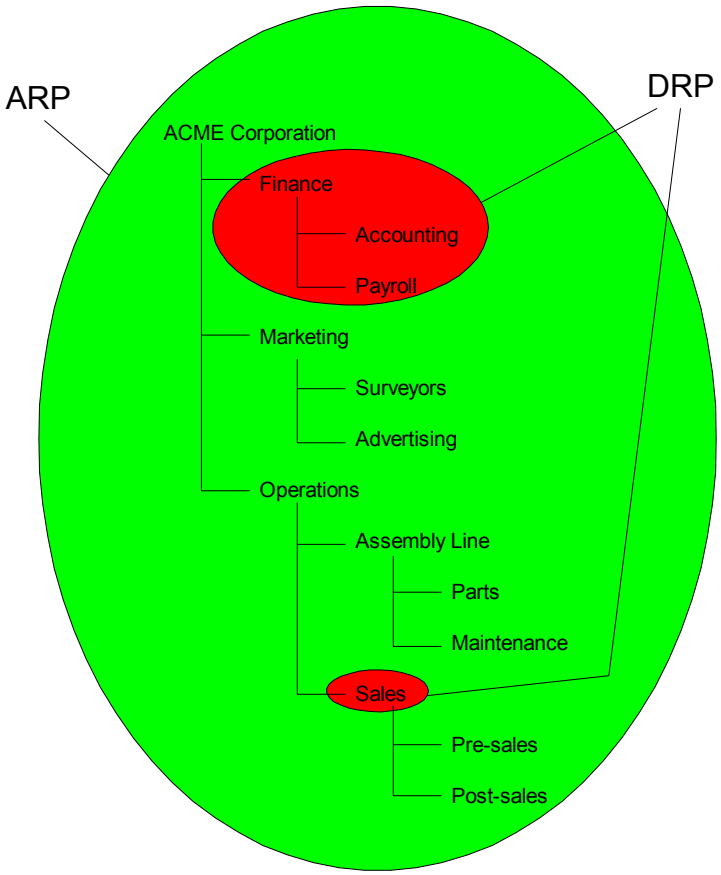


Figure 24

A RM can establish as many resource profiles as required to allow or deny access to a resource role. Deny resource profiles are evaluated first by the RES. An *access control role (ACR)* is a container consisting of a set of resource profiles. If any resource profile produces a match access can be granted or denied depending on type of resource profile. Figure 25 illustrates this point:

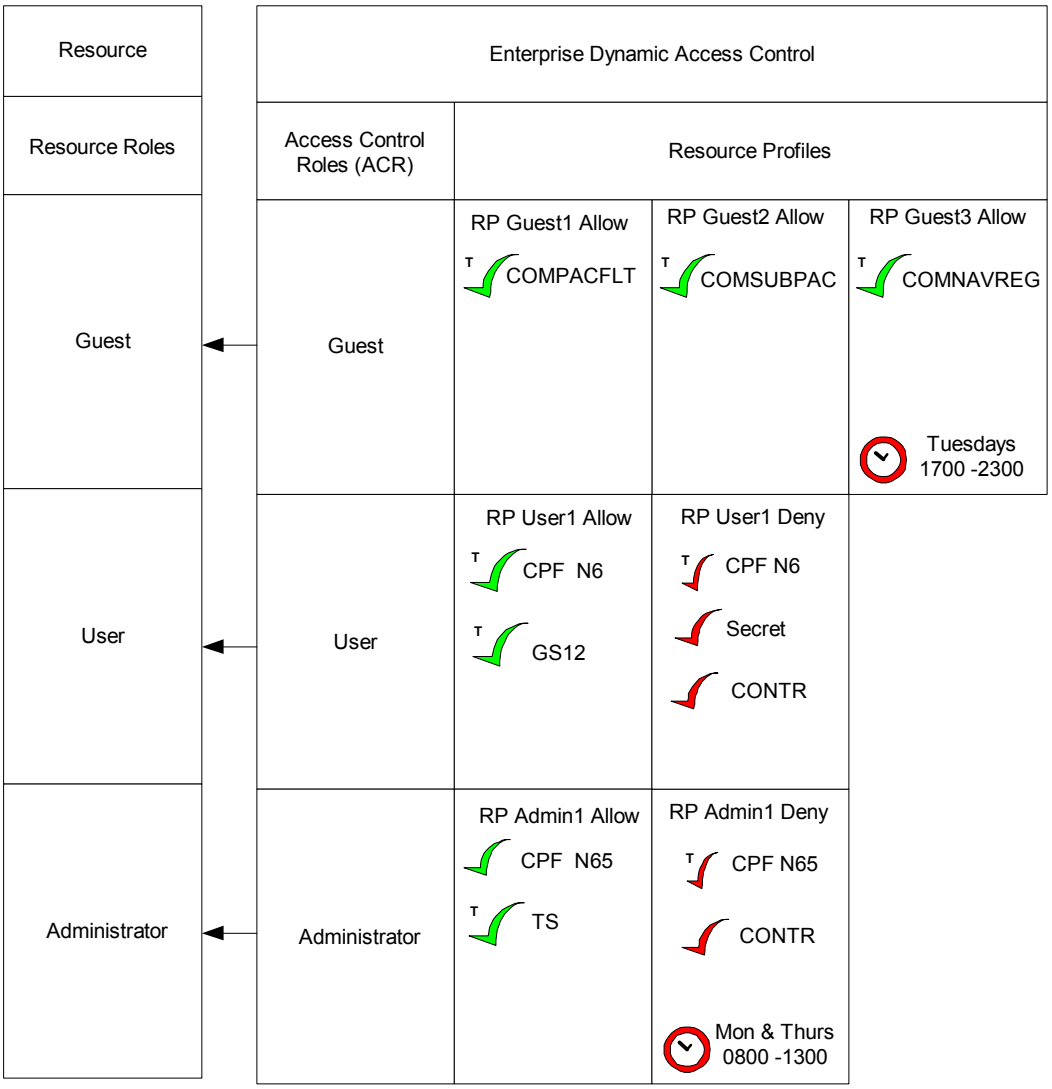


Figure 25

For example, if a user (or object) worked at CPF N6 and had a GS12 salary the *RP User1 Allow* resource profile (represented with green checkmarks) would produce a match and the object would be granted access to the *User* resource role. In another example, if an object worked at CPF N651, as a contractor the *RP Admin1 Deny* resource profile (represented with red checkmarks) would produce a match only during the hours of 0800-1300 and the object would be denied access to the *Administrator* resource role.
Note a "T" next to the checkmark represents a sub tree or "rVT" reference condition.

Structure Format Service (SFS)

The SFS converts reference request such as a user profile, environmental status and/or complex outputs into a structured format that can be compared with reference conditions stored in resource profiles. A CMD consisting of directory services requires the SFS to convert reference inputs into distinguished name format for evaluation by the RES. Refer to figure 26:

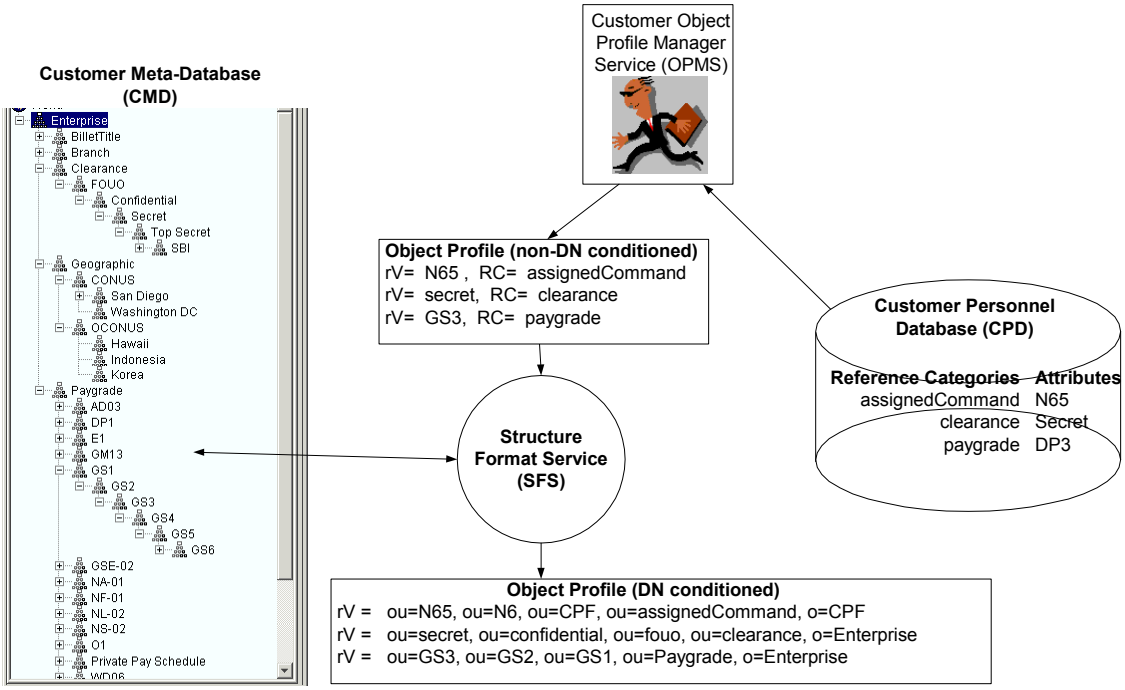


Figure 26

Since reference inputs are not in DN format the SFS queries domain CMDs and converts them into DN values. A DN reference represents a format that can be evaluated for inheritance purpose. For example, say a reference condition contained the following DN value:

rV: ou=secret, ou=confidential, ou=clearance

If an object profile contained the following reference input a match would not occur:

rV: ou=top secret, ou=secret, ou=confidential, ou=clearance

However, if the reference condition changed to a sub Tree a match would occur because it would represent all values equal or above *secret*, which includes *top secret*:

rVT: ou=secret, ou=confidential, ou=clearance

The capability to perform hierarchal evaluation on every: object, environmental and complex reference is essential for determining resource access and a significant aspect of the EDAC framework. ACL and groups are non-existent or limited in this capability.

Condition Status Service (CSS)

Reference changes in a CMD due to a re-organization, salary re-structure, etc., could affect resource access because a mismatch may occur between a newly created object profile and a previously established resource condition in a resource profile.
For example, say a RM establishes a reference condition from an organization structure contained in a CMD. Assume the selected reference condition requires *sales* permission to access a particular resource:

ou=sales, ou=operations, ou=ACME

Then a re-organization occurs and the *sales* department is placed under *marketing* in the CMD:

ou=sales, ou=marketing, ou=ACME

A user from the sales department will be processed with the latest reference input:

ou=sales, ou=marketing, ou=ACME

Because the reference condition was stored with the old structure (under *operations*) access will be denied. This automated constraint offers security and ensures RMs reconsider access control policies due to corporate changes. Content changes in the CMD triggers an event listener in the CSS to scan reference conditions in the repository service. Any mismatches are flagged as deprecated reference conditions in the CMS. The RM can easily identify the affected conditions and decide to edit or remove the deprecated reference condition(s). The CSS can also detect if a particular CMD is currently off-line and flags those associated conditions associated with that respective CMD.

Enterprise Interoperability

The EDAC model allows for easy expansion and interoperability among different regions. In an enterprise scenario, RMs are capable of establishing reference conditions for local and remote objects. For local objects the RM selects reference conditions from local CMDs and for remote objects the RM selects reference conditions from remote CMDs.

Figure 27 illustrates the concept.

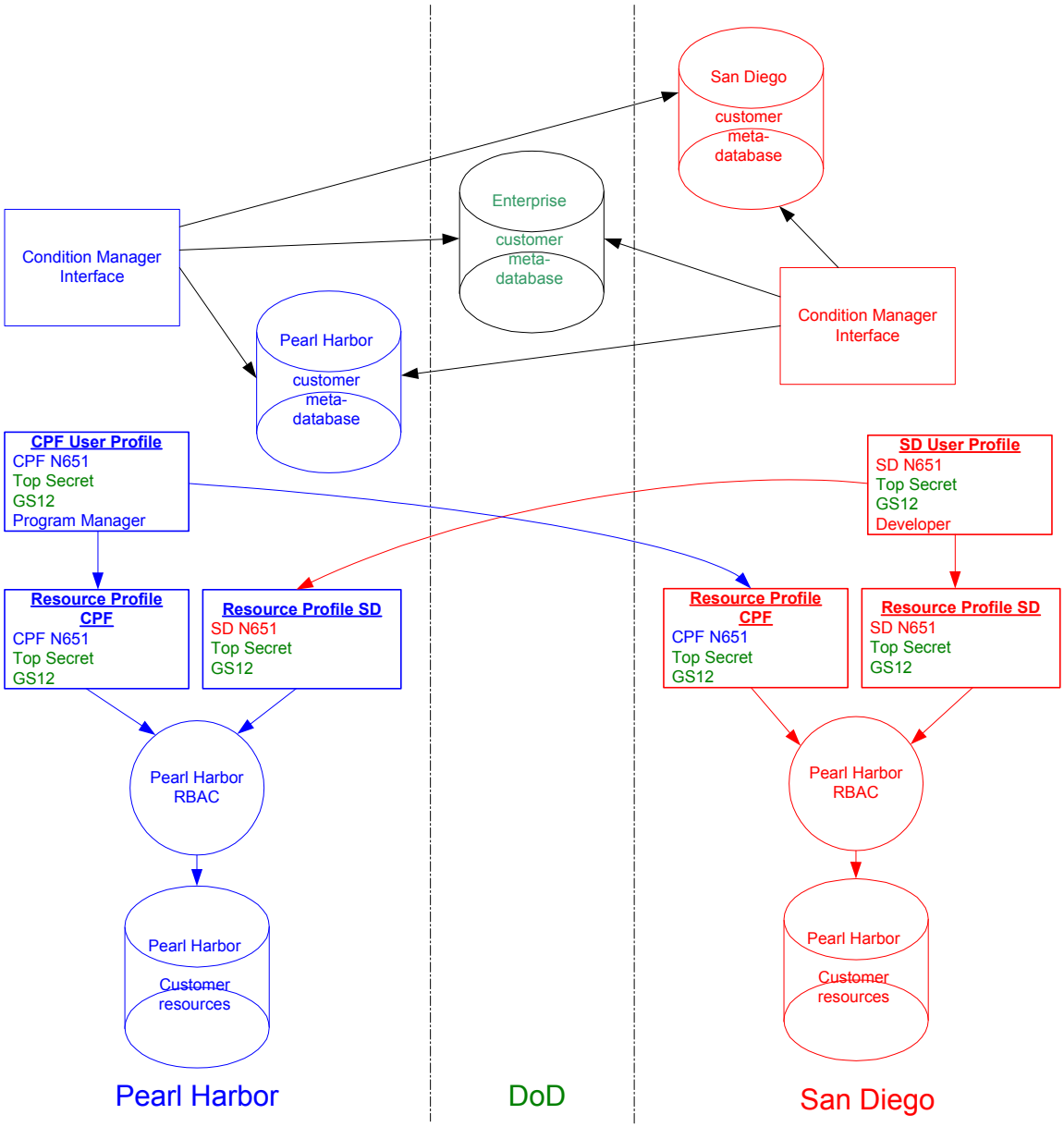


Figure 27

In the illustration above a Pearl Harbor RM has established two resource profiles:

Resource profile CPF
Resource profile SD

The CPF resource profile allows conditional resource access only for Pearl Harbor objects and the SD resource profile allows conditional resource access only for San Diego objects. Both CPF and SD resource profiles were created by the selection of reference conditions from various CMDs applicable to local and remote objects.

Enterprise Dynamic Access Control (EDAC) Summary

Niche authorization systems offer little in security by only evaluating a small spectrum of parameters. Refer to figure 28.

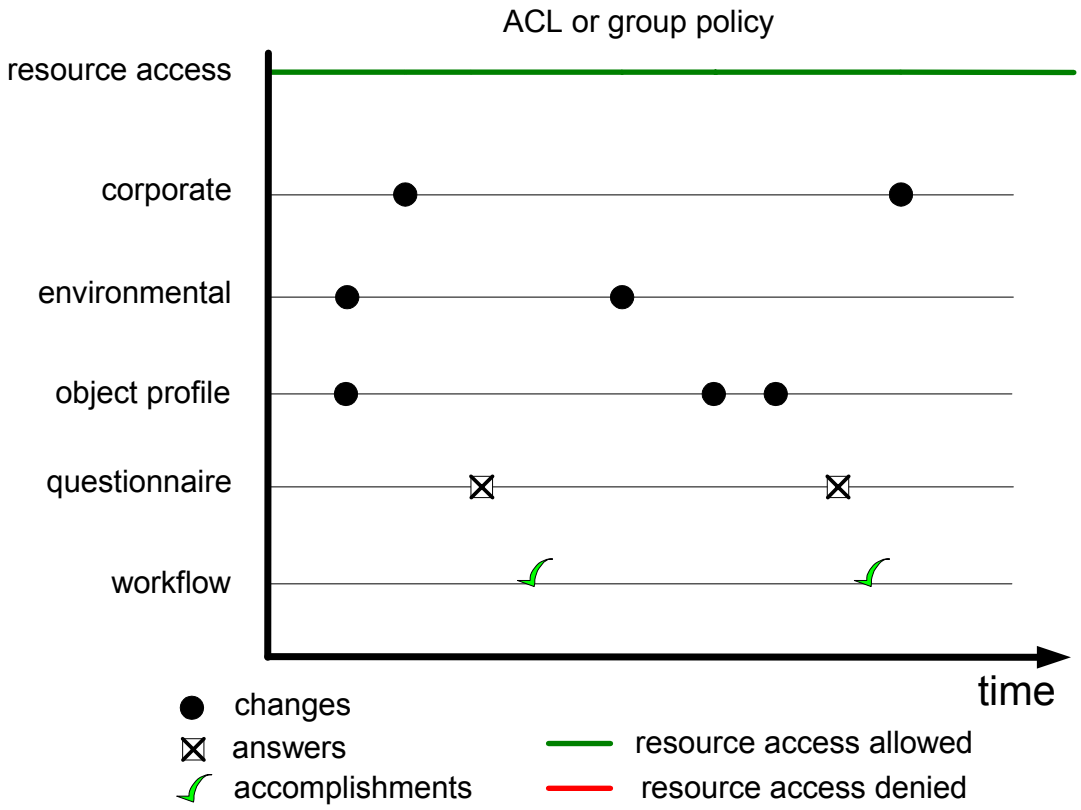


Figure 28

In contrast an authorization system that offers a comprehensive solution can improve security and offer the customer the capability to monitor and evaluate resource access on a real time basis. Refer to figure 29

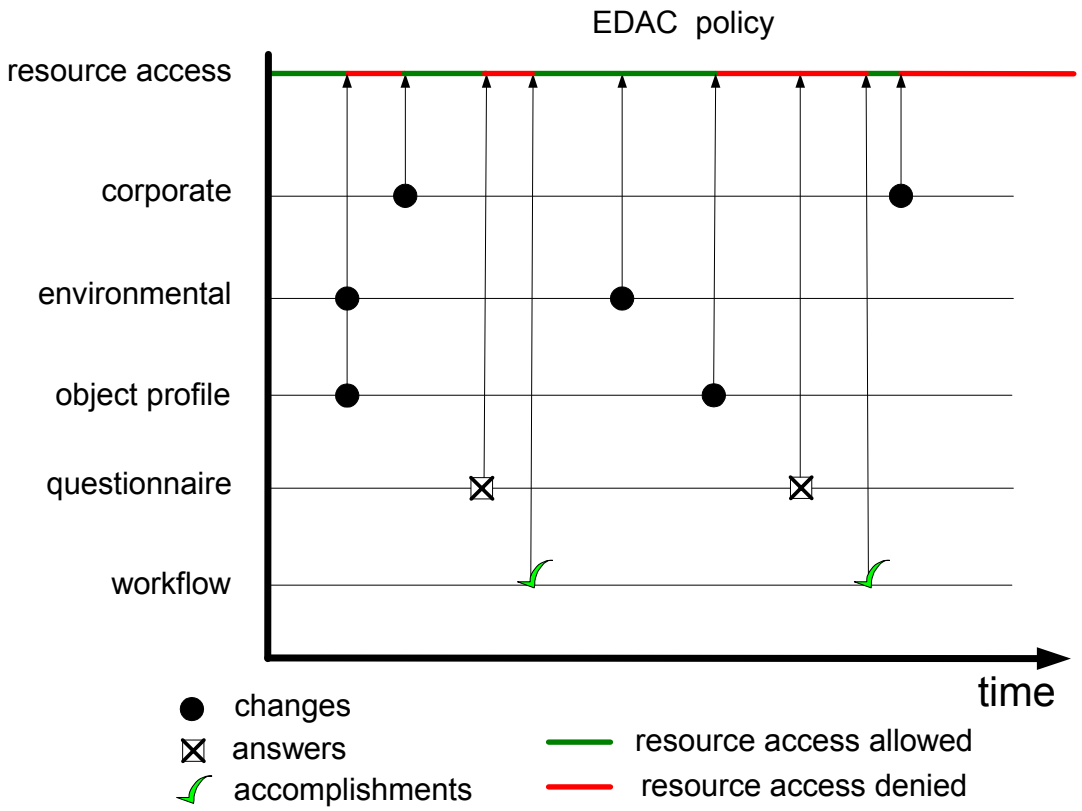


Figure 29

The objectives discussed in this paper emphasize the focus of the EDAC model. A comprehensive, standards-based and manageable access control system is vital to accommodate the growing authorization complexities confronting customers. The EDAC model requires little in equipment and operational costs because it leverages customer assets to produce an effective access control system reflective of a customer's environment. The amount of customer participation in maintaining a detailed and accurate CMD correlates on the usefulness of the access control service.