Proposal for Fast-Tracking NIST Role-Based Access Control Standard

David Ferraiolo

Rick Kuhn

National Institute of Standards and Technology

Gathersburg, Maryland

Ravi Sandhu

George Mason University

Fairfax, Virginia

Agenda

- Why an RBAC Standard?
- Is the Standard Ready to Go?

Some of the Vendors Offering RBAC Products

Sun microsystems

Microsoft

Cisco Systems
Empowering the Internet Generation

Novell

IBM

SIEMENS

symantec

SYBASE

RSA

network associates

bmcsoftware

SECURE COMPUTING

Entrust
Securing the Internet

ORACLE

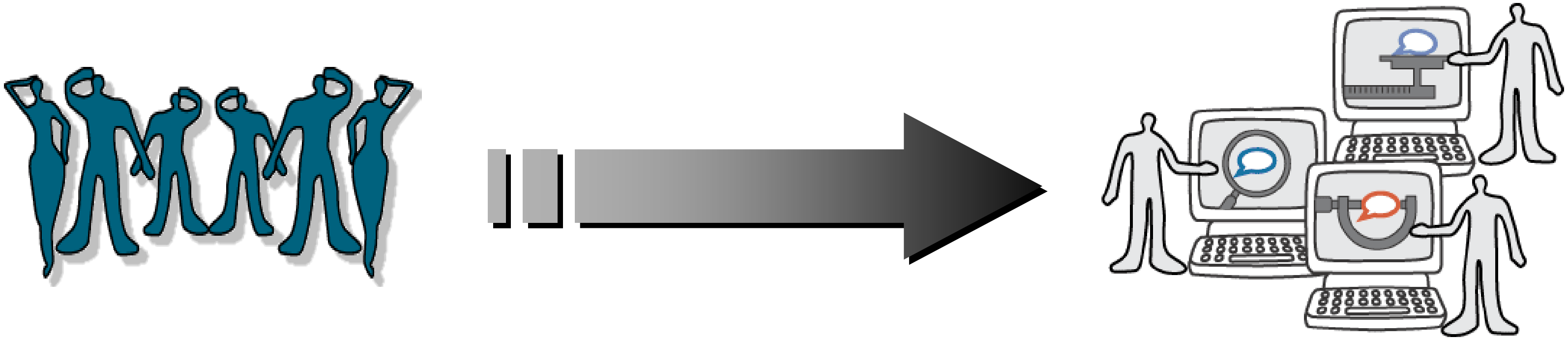
ADEXA

protegrity
We keep your secrets.

access360

Accurate Configuration Control Over User Privileges

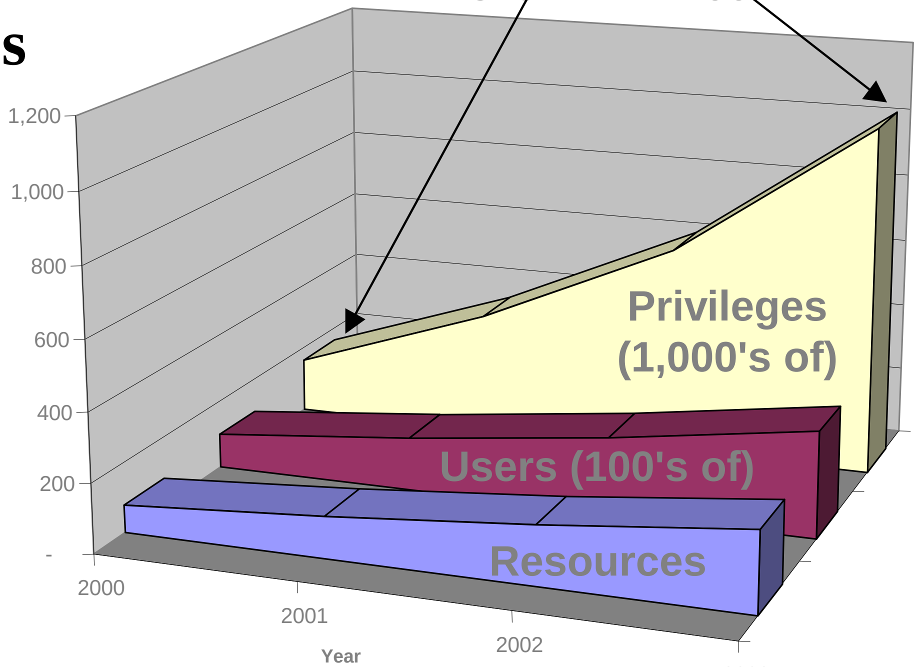
Lots of users and privileges scattered over many platforms and applications.
Who are the valid users?
What are they entitled to access?
How do you keep access rights up-to-date?
How do you specify and enforce policy?



Maintaining Access Configurations is Labor-Intensive

- **Adding IT Staff Scales Linearly**
- **Administering Privileges Scales Non-Linearly**
- **Symptoms of the problem**
  - **Unused accounts proliferate**
  - **Turn -on time rises for user privilege creation**
  - **Privilege review is impractical**
  - **Security audits fail**
  - **User down -time increases**
  - **Security admin requests staff increases**
  - **Help desk requests staff increases**

Privileges almost double yearly growing from less than 200k to over 1M in 2004

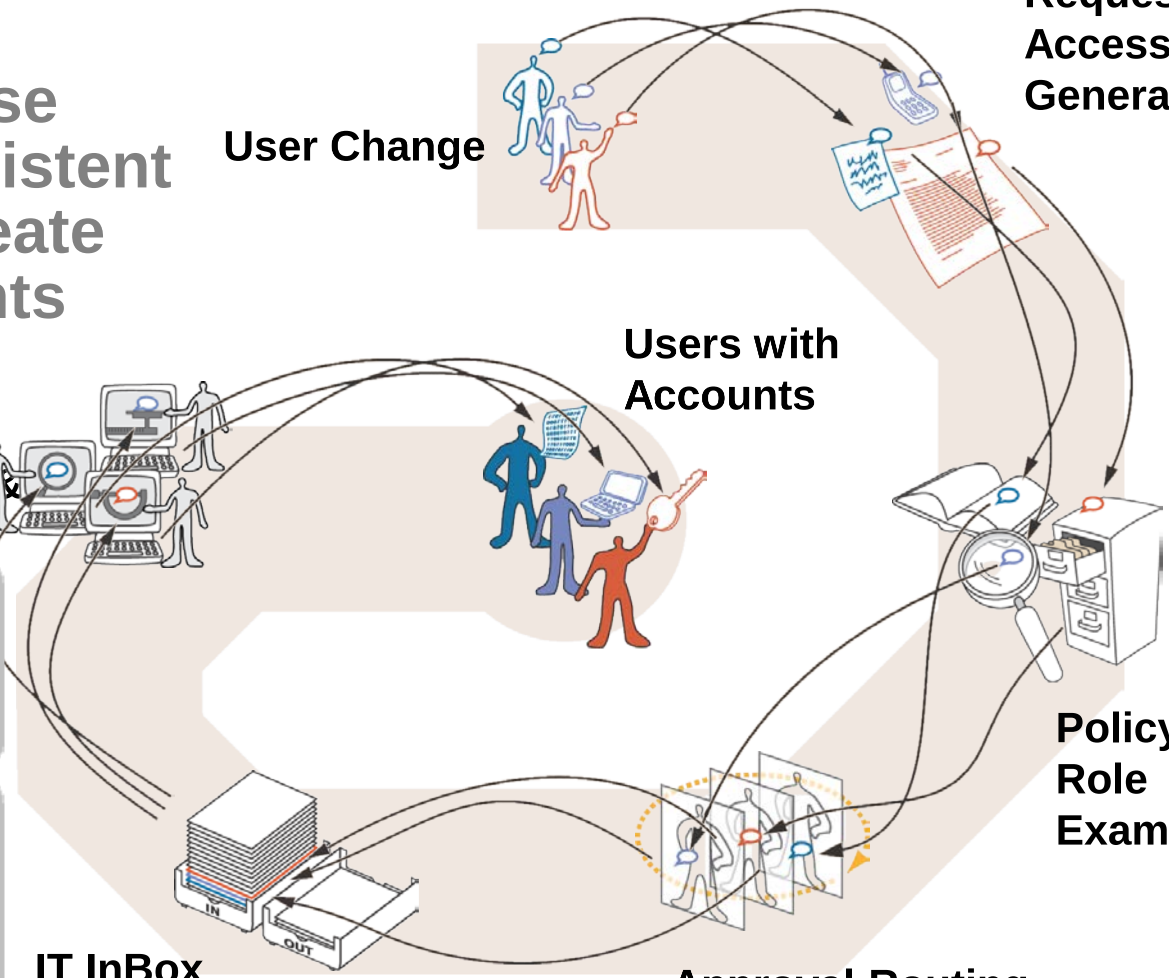


Source: IDC, 2001

Estimated Privilege distribution Activity in Typical Companies

Manually Configuring Privileges

Organizations use slow and inconsistent processes to create user access rights



User Change

Request for Access Generated

Users with Accounts

Policy & Role Examined

Approval Routing

IT InBox

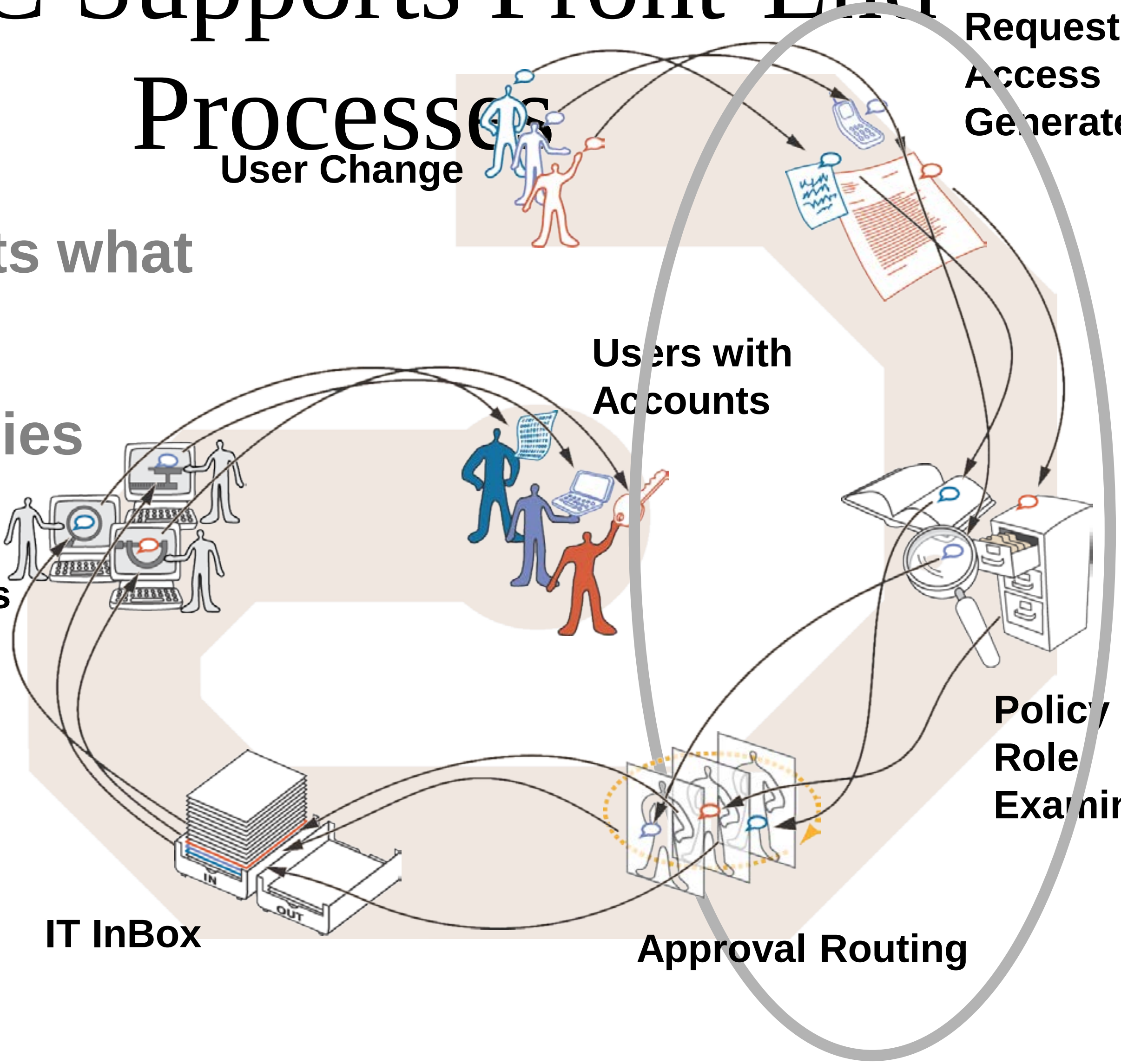
Administrators Create Accounts & Access Rights

Elapsed turn-on time: up to 12 days per user

Account turn-off performance: 30-60% of accounts are invalid

RBAC Supports Front-End Processes

Maintain who gets what based on your organization's operational policies



User Change

Request for Access Generated

Users with Accounts

Administrators Create Accounts

Policy & Role Examined

IT InBox

Approval Routing

Installed Technology Base

Access Control List (ACL) are the most common access control mechanism in use today

- Fine when end-users are viewed as “owners” of enterprise resources
- Resource Oriented: poorly organized to address many commercial and Government security policies
- Costly and difficult to centrally administrate
- At the wrong level of abstraction
- Platform Dependent with proprietary administrative tools

Role-Based Access Control – A Strategy for Security Policy Management

- Centrally administered and locally enforced role based access control policies
- Policy Rich: highly configurable (richer set of parameters)
- Enforces access control across the virtual enterprise
  - Employees
  - Suppliers
  - Consultants
- Role membership is based on Competency, Duty, Authority, giving the user's the potential to execute privileges
- Role centric (roles are global and persistent)

Motivations

- Simple and Intuitive Administrative Interface
- Administrative Efficiency
  - Automatic user privilege assignment
  - Automatic revocation of user privilege
  - Simple user functional re-assignment
- Administrative Flexibility
  - Static Separation of duty (SSD)
  - Fine granularity of resource/administration partitioning
- Scalability, Extensibility, Accuracy
- Agreement of core RBAC Features
- For Each RBAC feature in the standard there are one or more known implementations
- Broad industry involvement in ACM RBAC Workshops

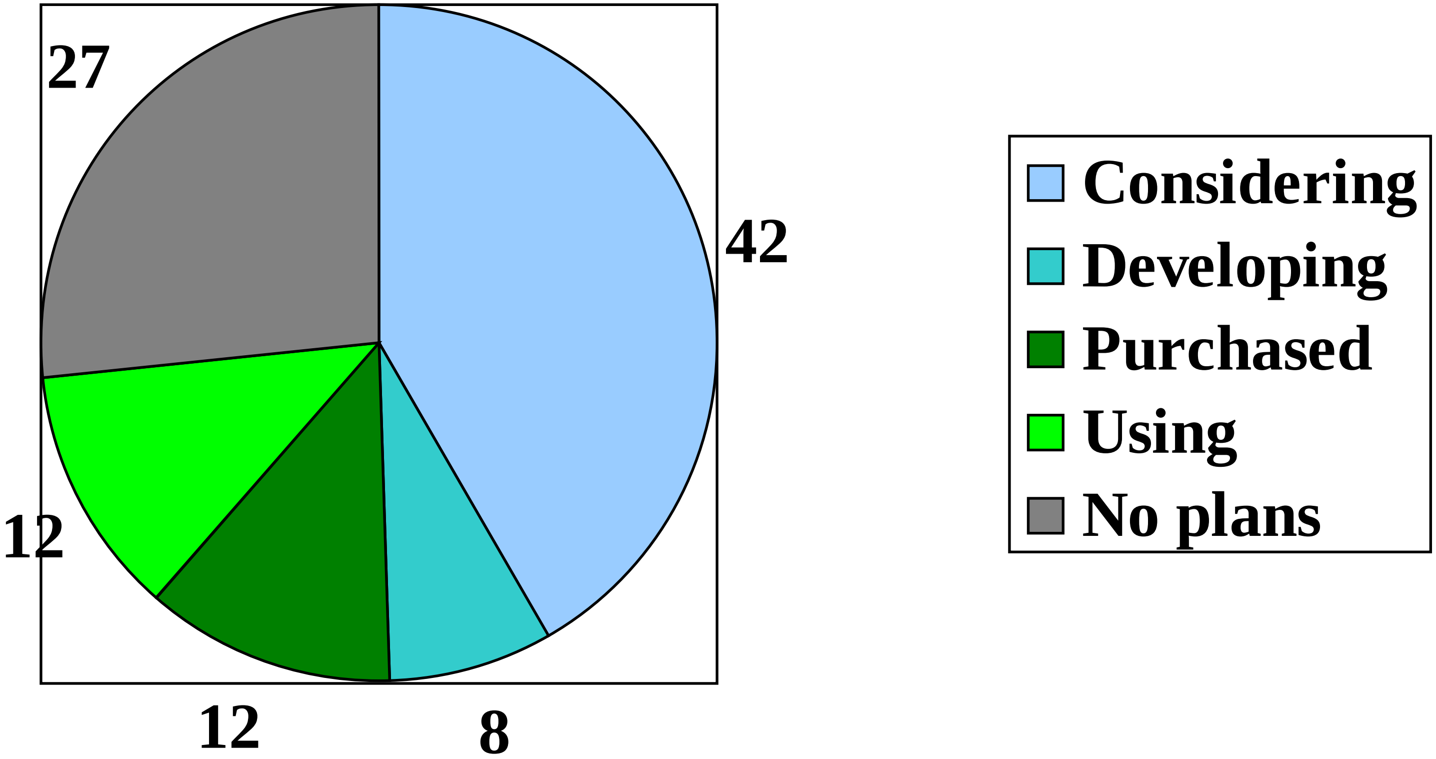
Background

- NIST study reviewed the access control practices of 30 large organizations
- First RBAC model published in 1992
  - Combined several existing and emerging concepts (OS user groups, DBMS privilege groups [Baldwin90], separation of duty [Clark-Wilson87, Sandhu88, Brewer-Nash89] into a single relational model [Ferraiolo-Kuhn92]
  - Reference implementation led to a revision [Ferraiolo-Cugini-Kuhn95]
- Annual ACM RBAC Workshop series started in 1995 with international vendor and researcher participation
- Sandhu et al, developed a well accepted comprehensive RBAC framework in 96
- Sybase implemented most of NIST RBAC model in 1996, DBMS survey showed other vendors have RBAC features
- Based on these efforts numerous other models have been proposed that have often included reference implementations

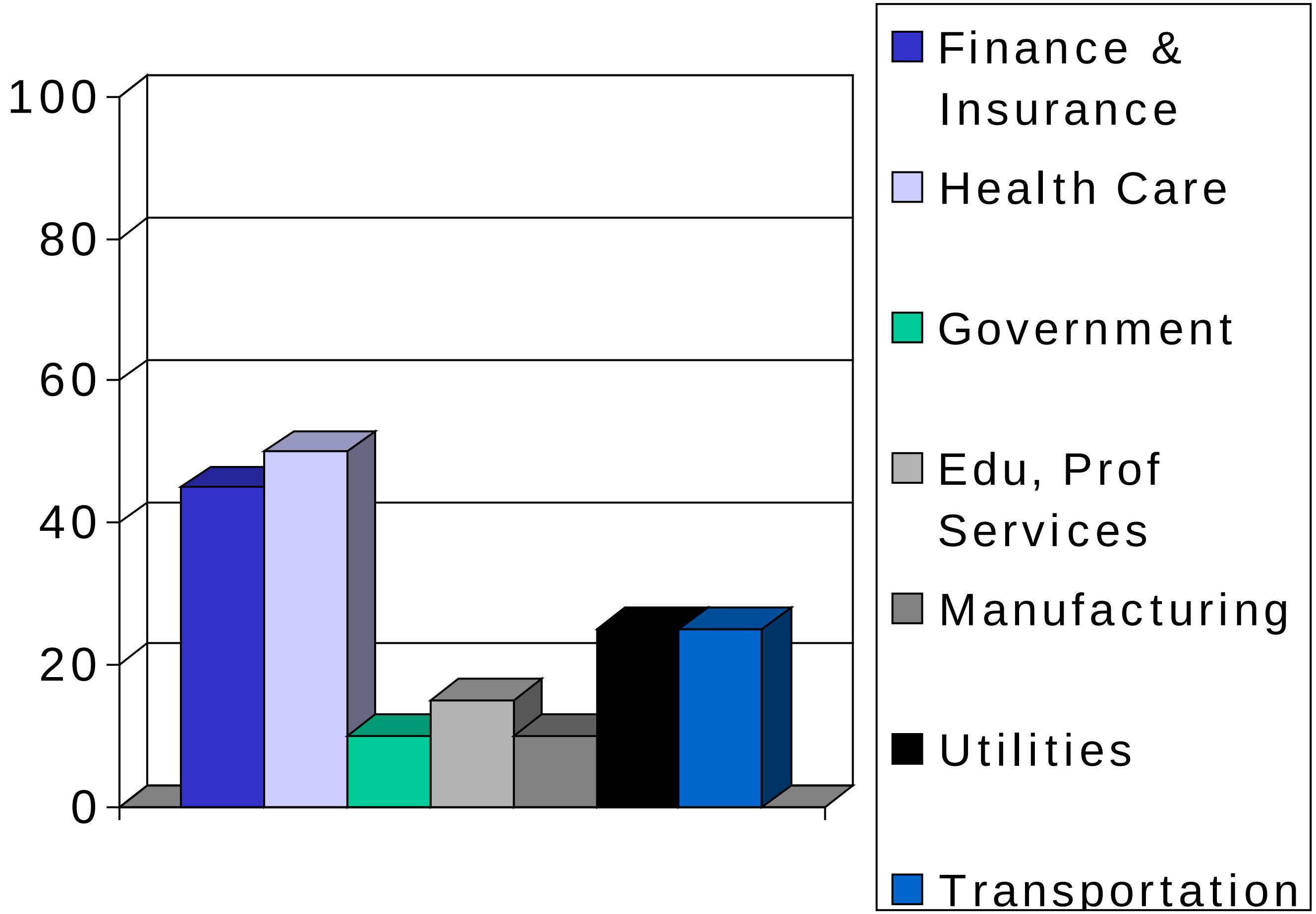
Background

- Since 1995 vendors, users, and researchers have gathered on an annual basis to present papers and discuss issues related to RBAC, in a formal ACM workshop setting
- RBAC has matured to the point where it is being consistently prescribed as a generalized approach to access control
  - “the most attractive solution for providing security in e-government” IEEE COMPUTER, Feb. 2001
  - “most relevant in meeting complex policy needs of Web-based applications” ACM COMMUNICATIONS, Feb. 2001
- First effort to define a consensus standard for RBAC was proposed in a special session at the 5th ACM Workshop on RBAC
- Published comments resulted in the existing proposed standard

Diffusion of RBAC - 2001



Estimated Use of RBAC in 2005 - by industry (mid-range est)



Timeliness & Appropriateness of RBAC Standard

- Need for consistent, universally understood semantics for RBAC
- Vendors value "establishing a taxonomy and a shared vocabulary for us, our customers, and the industry as a whole"

Is RBAC ready for a standard?

- Network Applications Consortium - $500,000,000,000 customer base says:

  “If RBAC is going to ‘move to the mainstream’, then there will have to be some sort of standard.”

Current Situation - Problem

- Although existing models and implementations use similar RBAC concepts, they differ in significant areas and use different terminology
- RBAC is a rich and open-ended technology, ranging from the very simple to the complex
  - Not all features are appropriate for all environments
  - No vendors implement all RBAC features
  - Research continues to promote its use in other applications and extended features

Solution - RBAC Standard

- Standardization over a collection of basic and well accepted RBAC features
- Features are divided into logical components and sub-components
- Sub-components can be combined into relevant packages giving:
  - IT consumers a basis for uniform acquisition specification and a basis for making purchasing decisions
  - Vendors a set of benchmarks use in the characterization and marketing of their products
- Each feature is known to be viable in that there exists at least one example commercial and/or reference implementation

Standard Organization

- Two Main Parts

  -- RBAC Reference Models

  -- Requirement Specification

- Four Components

  -- Core RBAC

  -- Hierarchical RBAC

  --- Limited Hierarchies

  --- General Hierarchies

  -- Static Separation of Duty Relations

  --- Without Hierarchies

  --- With Hierarchies

  -- Dynamic Separation of Duty Relations

Conformance

- Standard provides for conformance by vendor self-declaration
- Standard provides foundation for third-party conformance testing sought by vendors and customers

Requirement Specification

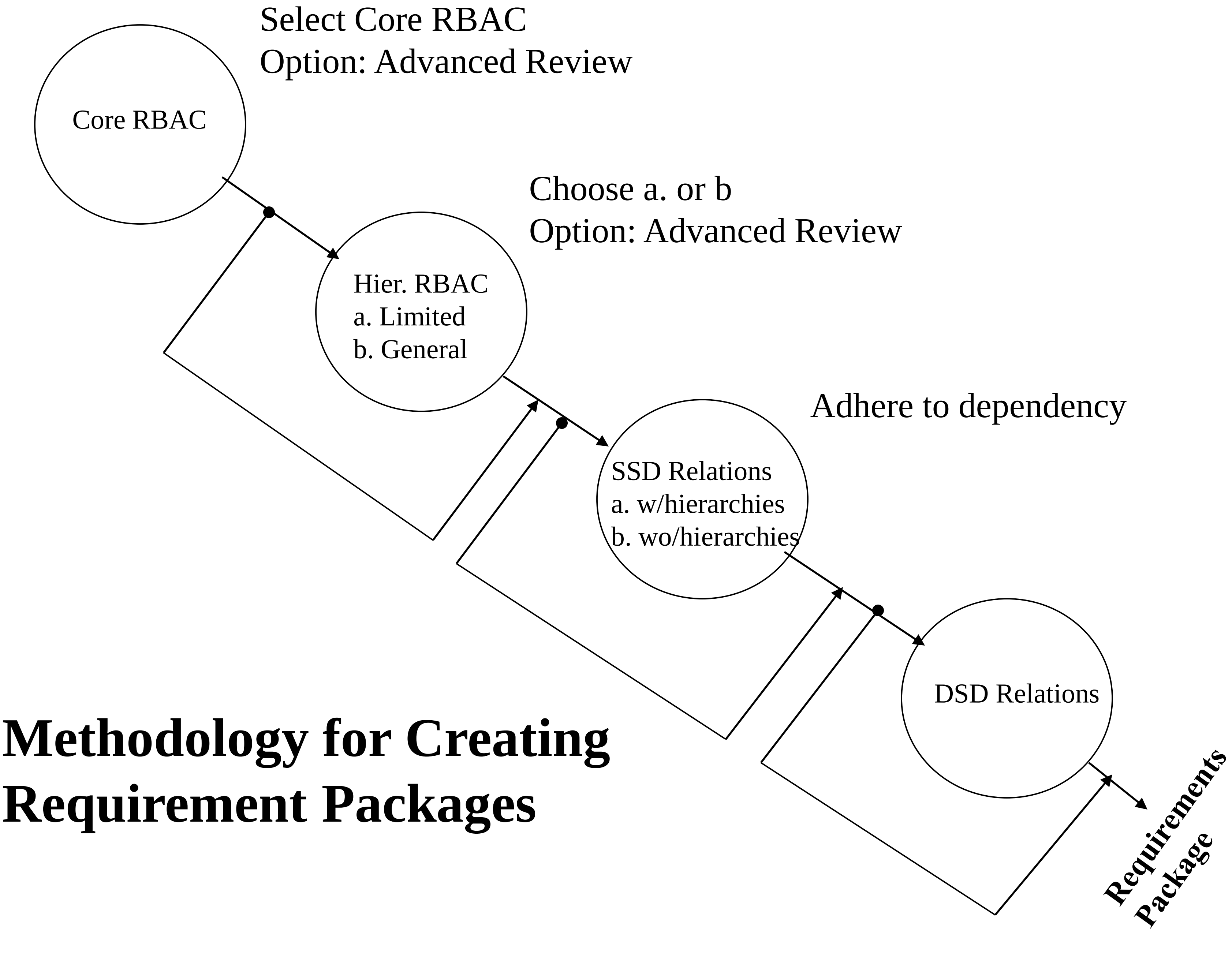
- Requirements are specified using the relations defined by the reference model
- Administrative Operations

  (e.g., create/delete role, create/delete user assignment, create/delete hierarchical relation)
- Administrative Queries and Review Functions

  (e.g., assigned users, assigned roles, authorized users, authorized permissions, separation of duty relations)
- System Functions

  (e.g., session management, access calculation)

Methodology for Creating Requirement Packages



Select Core RBAC
Option: Advanced Review

Core RBAC

Choose a. or b
Option: Advanced Review

Hier. RBAC
a. Limited
b. General

Adhere to dependency

SSD Relations
a. w/hierarchies
b. wo/hierarchies

DSD Relations

Requirements
Package

Conclusion

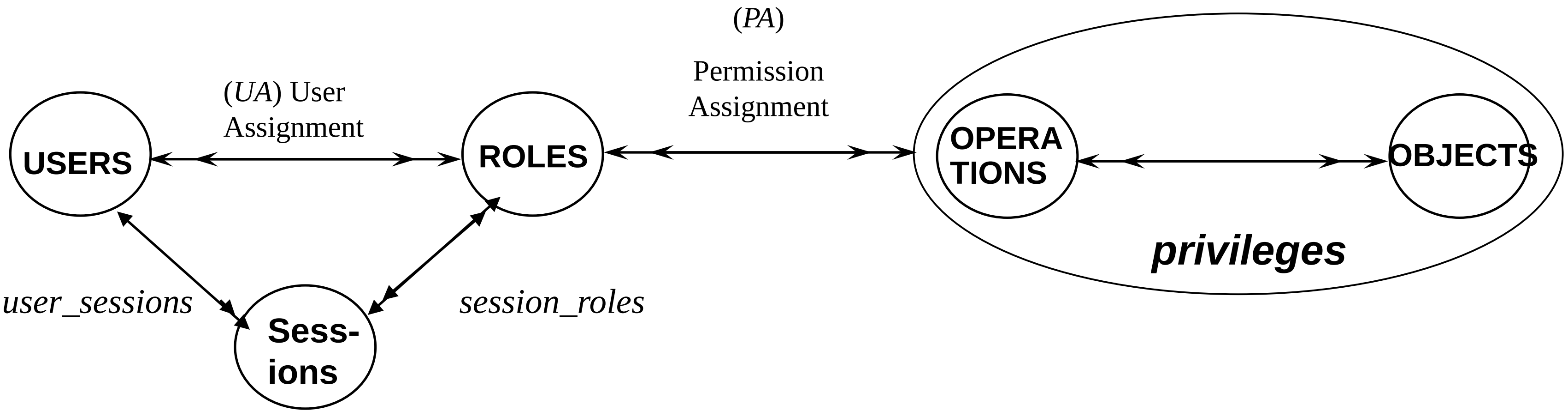
RBAC is ready for a standard

- User need - $500,000,000,000 customer base says: **"If RBAC is going to 'move to the mainstream', then there will have to be some sort of standard." – NAC**
- Vendors - At least 28 vendors offer some type of RBAC product
- Future solutions - "the most attractive solution for providing security in e-government" IEEE COMPUTER, Feb. 2001

Additional Information on Standard Components

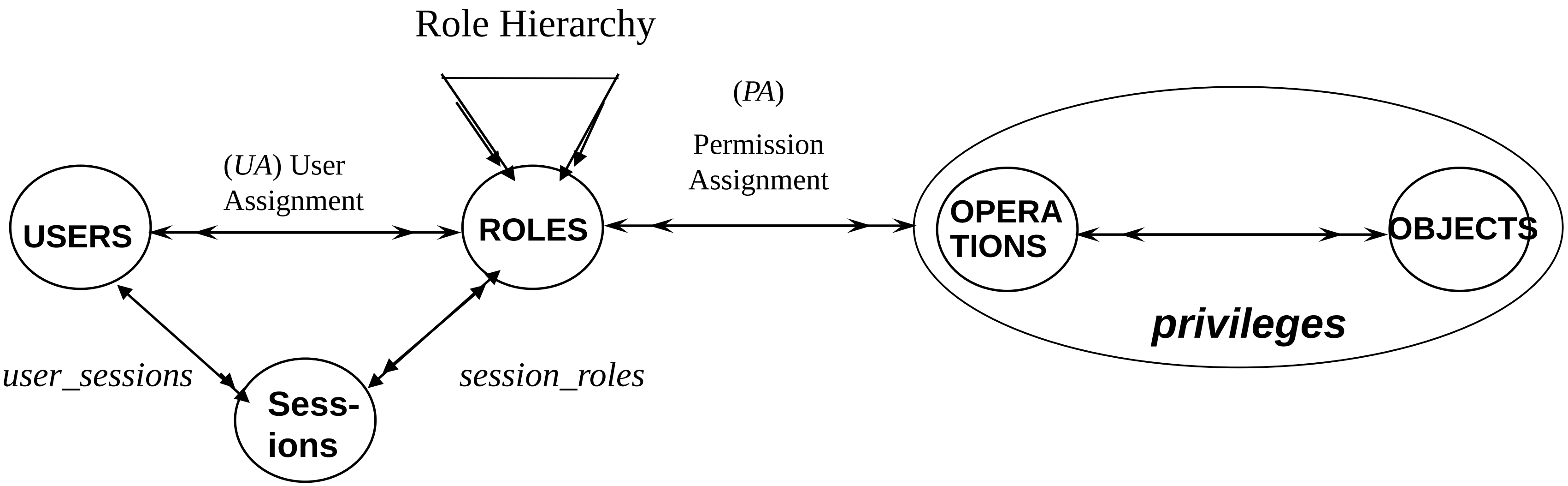
- Core RBAC
- Hierarchical RBAC
- Role Inheritance
- Static Separation of Duty
- Dynamic Separation of Duty

Core RBAC



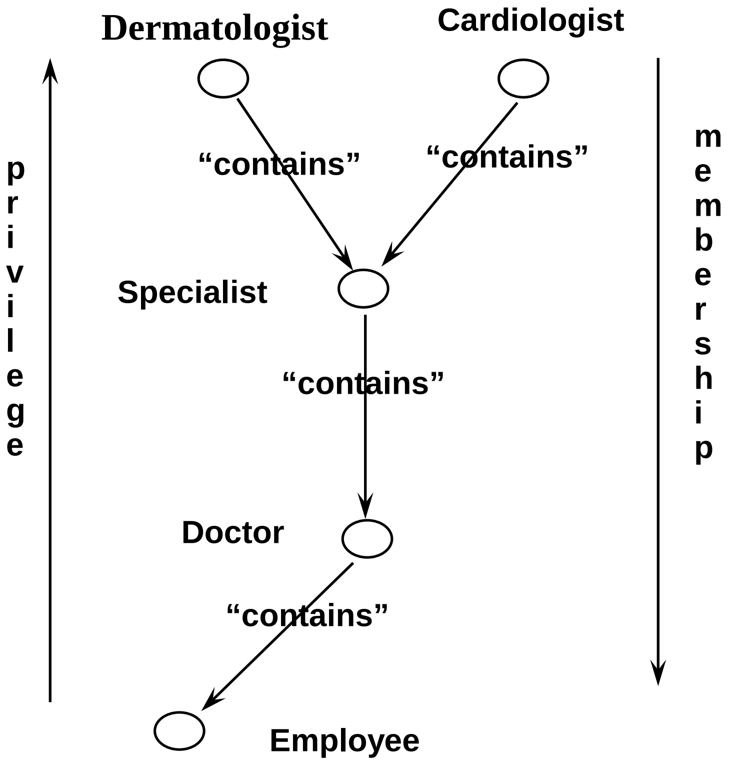
- Many-to-many relationship among individual users and privileges
- Session is a mapping between a user and an activated subset of assigned roles
- User/role relations can be defined independent of role/privilege relations
- Privileges are system/application dependent
- Accommodates traditional but robust group-based access control

Hierarchical RBAC

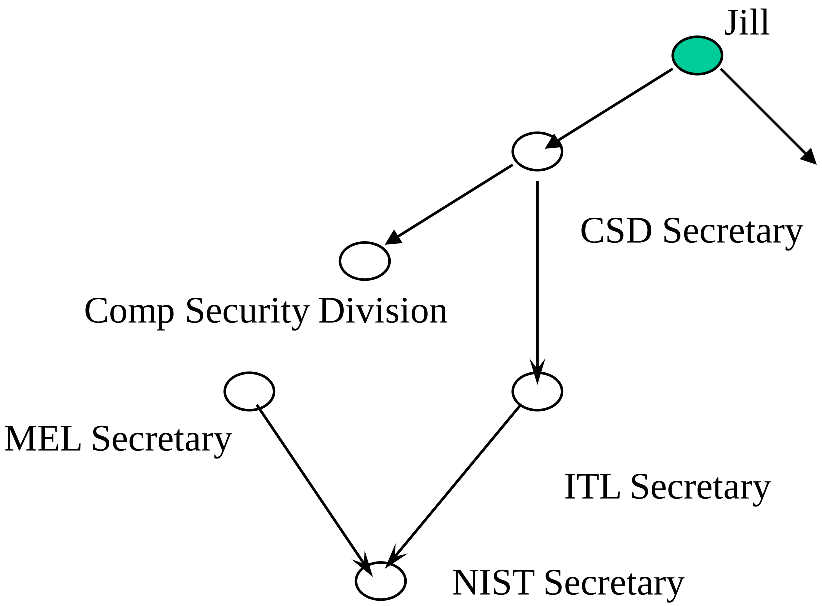


- Role/role relation defining user membership and privilege inheritance
- Reflects organizational structures and functional delineations
- Two types of hierarchies:
  - Limited hierarchies
  - General hierarchies

Role Inheritance



a-Limited Hierarchies

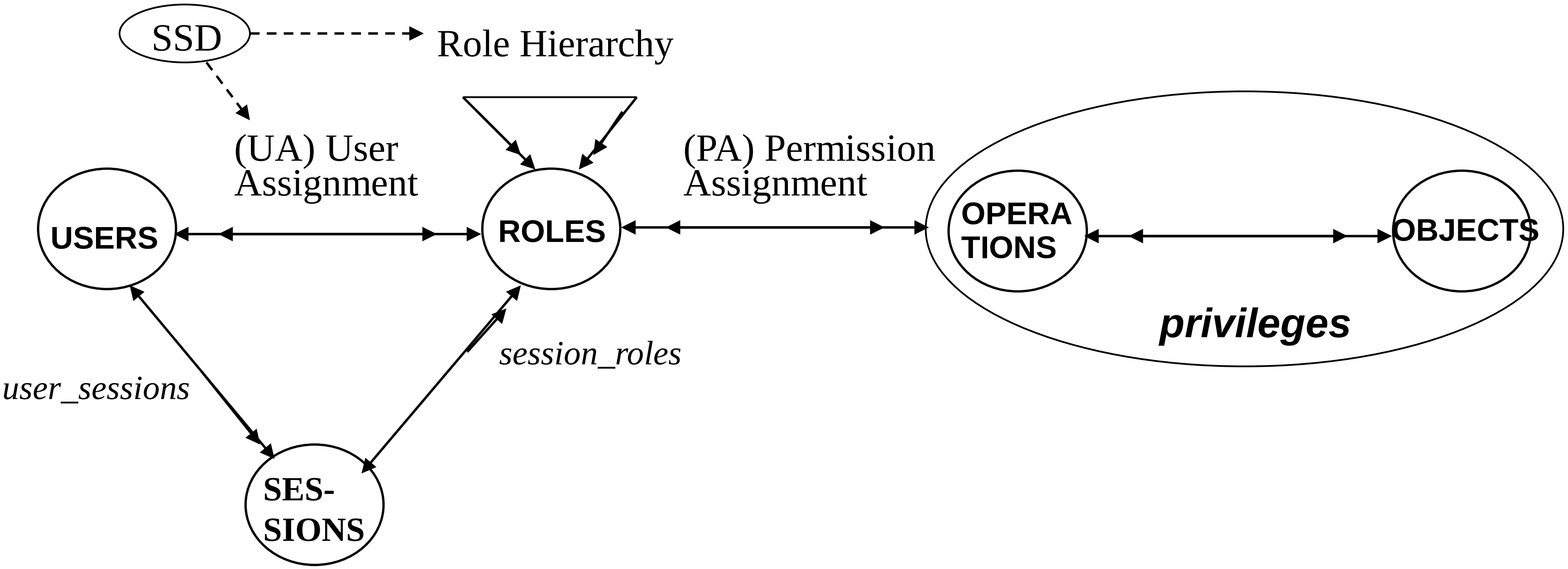


Added Advantages:

- User's can be included on edges of graph
- Role's can be defined from the privileges of two or more subordinate roles

b-General Hierarchies

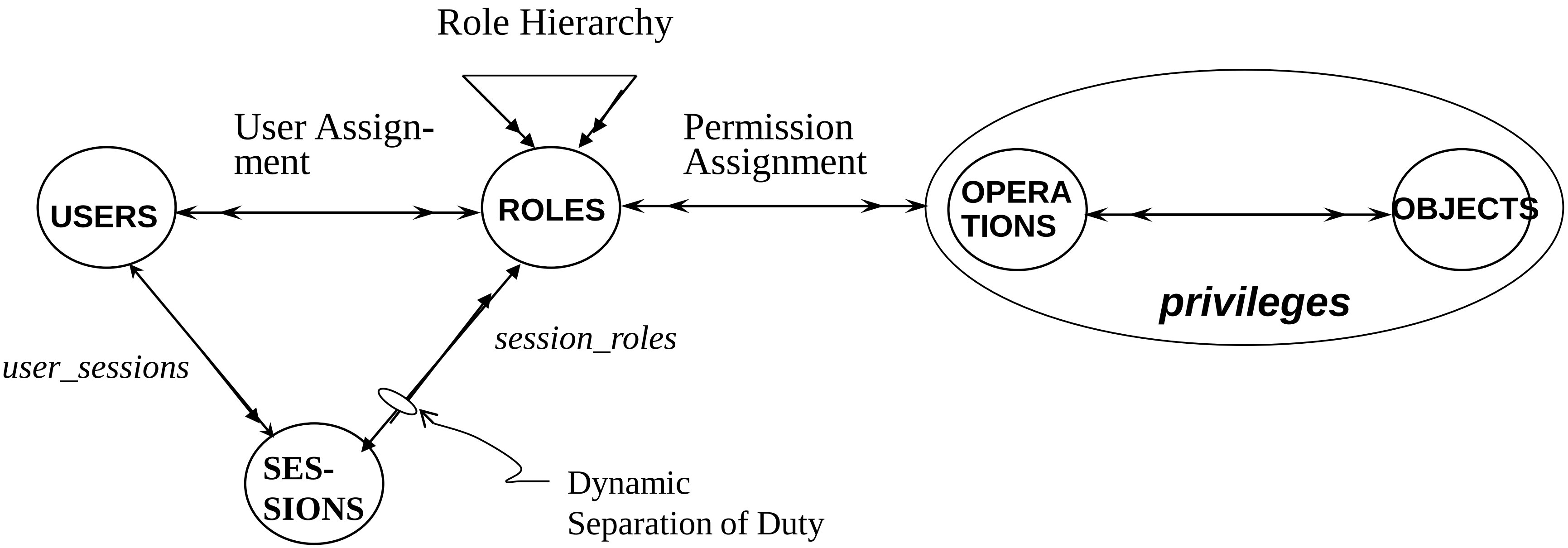
Static Separation of Duty



SoD policies deter fraud by placing constrains on administrative actions and there by restricting combinations of privileges that are available to users

E.g., no user can be a member of both Cashier and AR Clerk roles in Accounts Receivable Department

Dynamic Separation of Duty



DSoD policies deter fraud by placing constrains on the roles that can be activated in any given session there by restricting combinations of privileges that are available to users

E.g., No user can active both cashier and cashier supervisor role although the user maybe assigned to both

Valuable in the Enforcement of least privilege