

Administrative Scope and Role Hierarchy Operations

Jason Crampton & George Loizou

School of Computer Science
& Information Systems

Birkbeck, University of London

Administration in Access Control

- Any practical access control system must admit changes
- We will refer to components of a model that can change as dynamic
- We view administration as the process by which changes (to the dynamic components of a system) are controlled

Role-Based Administration

- Centralized
 - NIST model
 - Role graph model
- Decentralized
 - Administrative permissions assigned to (administrative) roles
 - RBAC96
 - Use structural properties
 - ARBAC97

Hierarchy Operations

- Delete edge joining role c (child) to role p (parent)
 - **DeleteEdge**(a, c, p)
- Add edge from child role c to parent role r
 - **AddEdge**(a, c, p)
- Add role r with children $C \subseteq R$ and parents $P \subseteq R$
 - **AddRole**(a, r, C, P)
- Delete role r
 - **DeleteRole**(a, r)

Structure of Talk

- Administrative scope
- RHA₄ model
- Comparison of RHA₄ model and ARBAC97
- Potential applications and future work

Administrative Scope

- Let R be a partially ordered set of roles
- For all $r \in R$, define

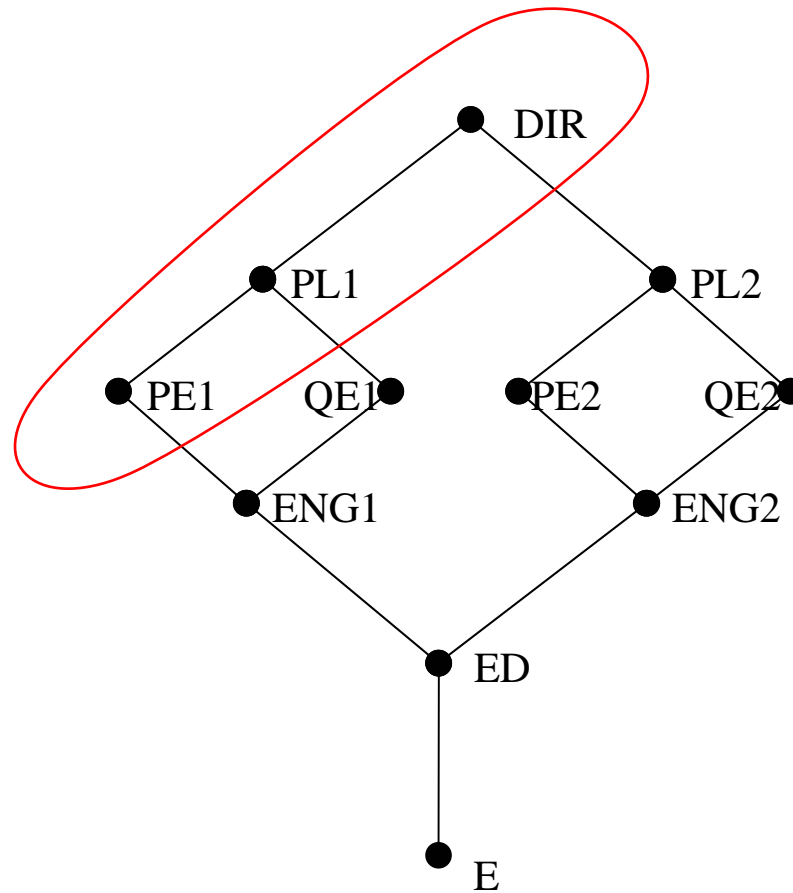
$$\uparrow r = \{s \in R : s \geq r\} \quad \downarrow r = \{s \in R : s \leq r\}$$

- For all $r \in R$, the *administrative scope* of r , denoted $S(r)$, is defined to be

$$\{s \in R : s \leq r, \uparrow s \setminus \uparrow r \subseteq \downarrow r\}$$

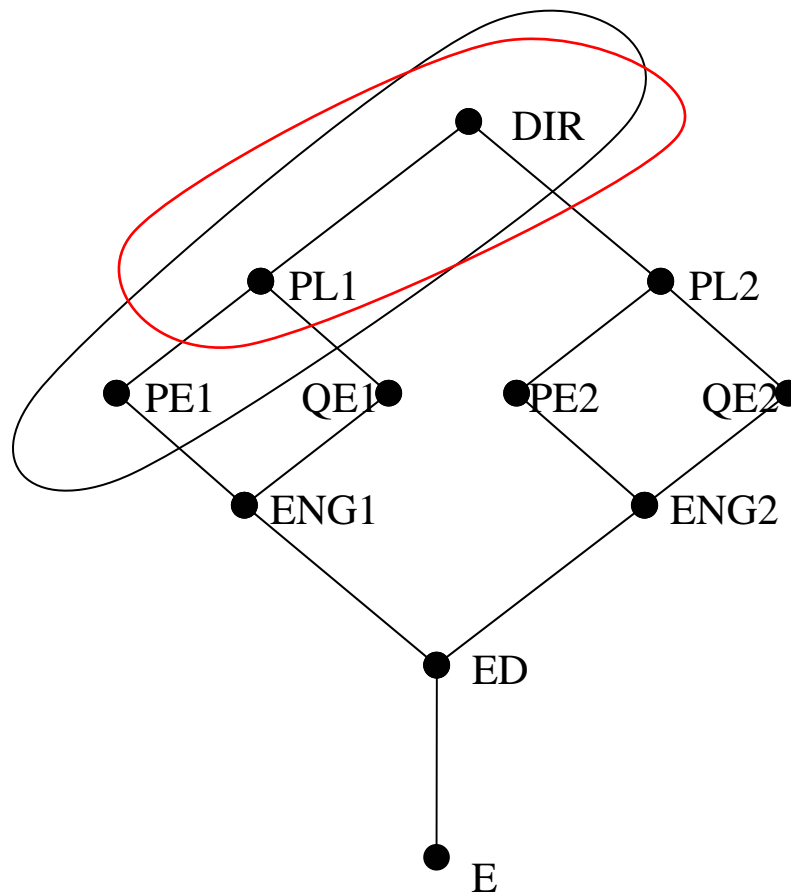
Administrative Scope

- \uparrow PE1



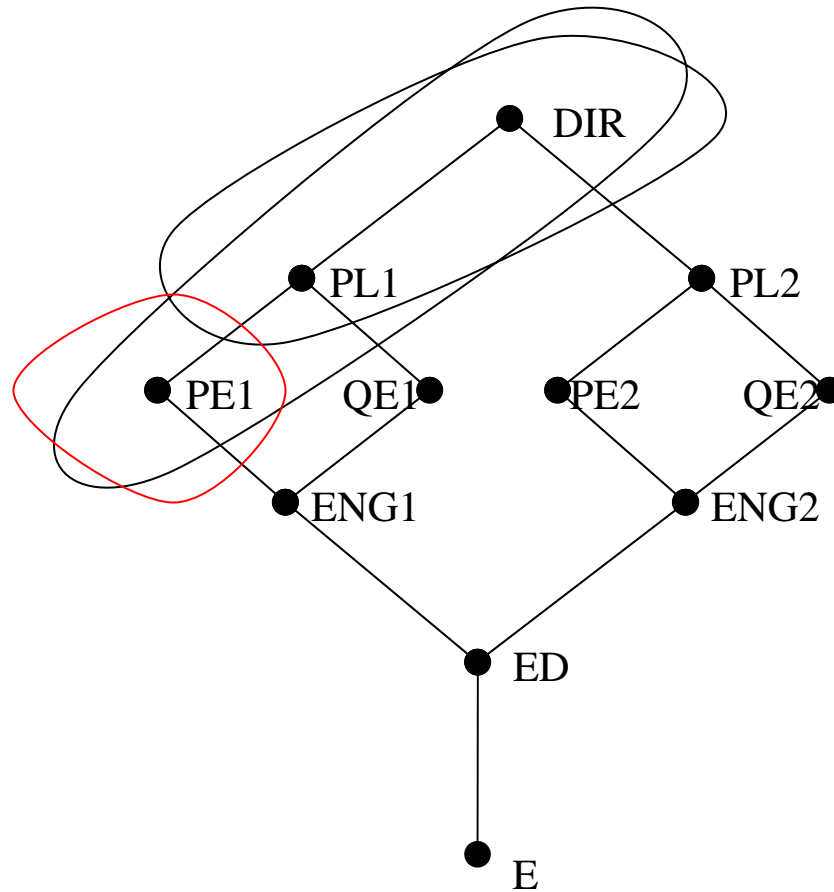
Administrative Scope

- \uparrow PE1
- \uparrow PL1



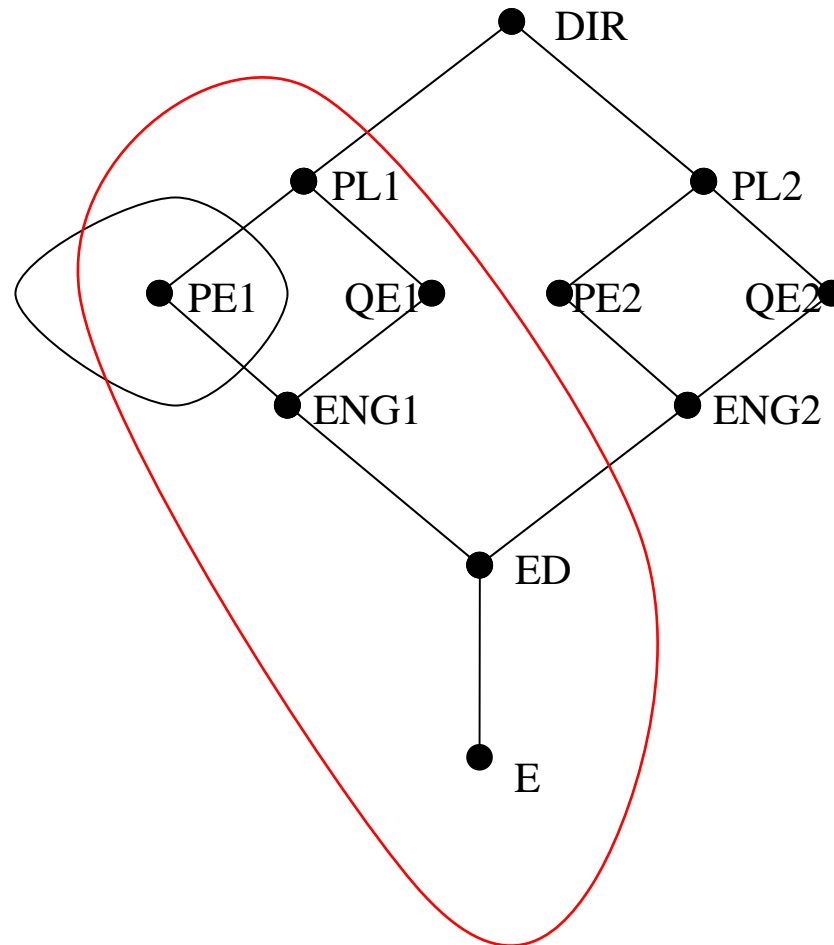
Administrative Scope

- $\uparrow PE1$
- $\uparrow PL1$
- $\uparrow PE1 \setminus \uparrow PL1$



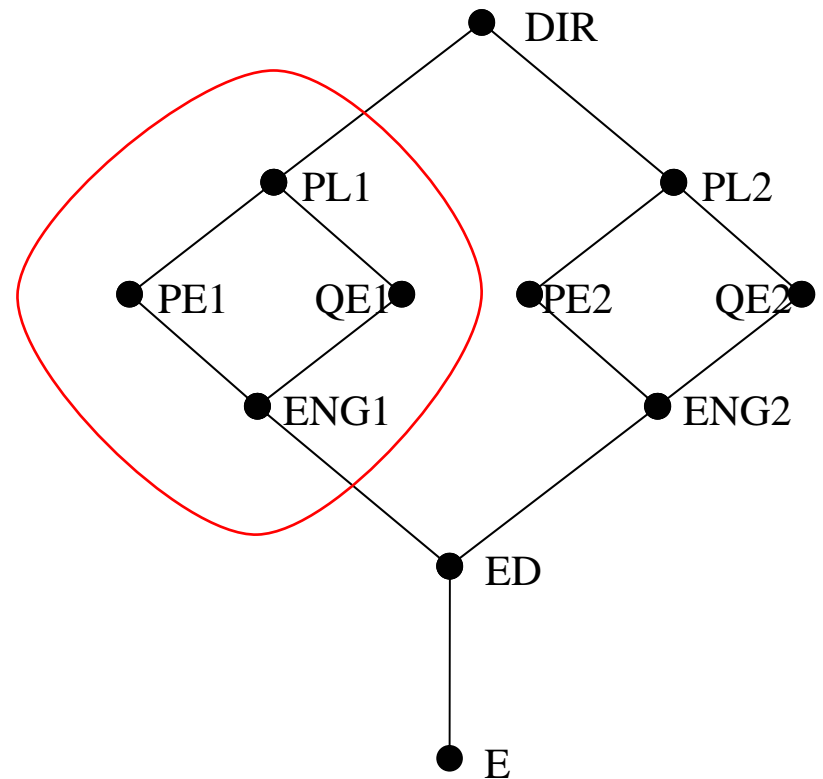
Administrative Scope

- $\uparrow PE1 \setminus \uparrow PL1$
- $\downarrow PL1$
- $PE1 \in S(PL1)$



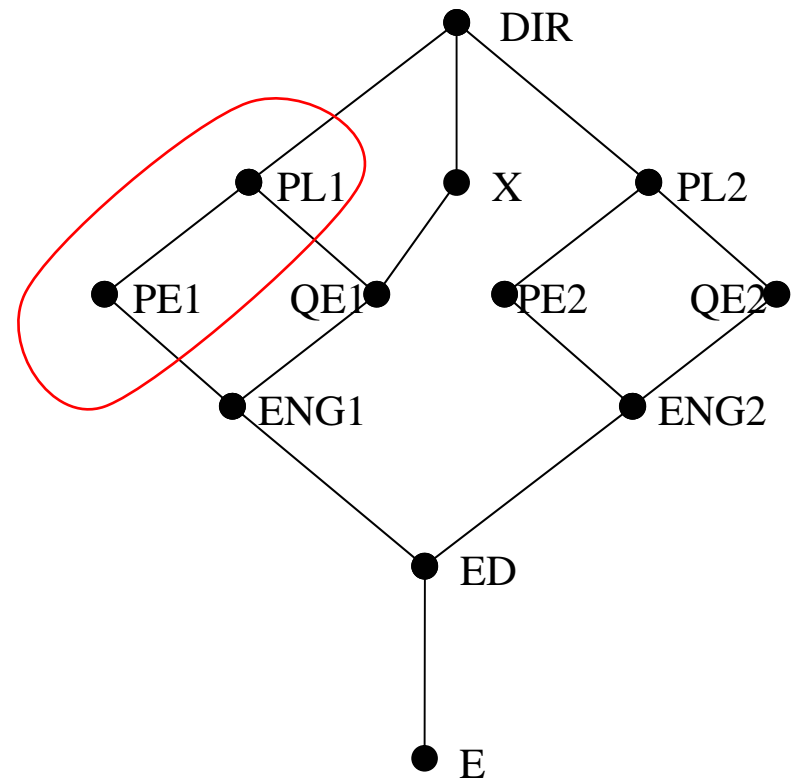
Administrative Scope

- $S(PL1) = \{ENG1, PE1, QE1, PL1\}$



Administrative Scope

- **AddRole**(?,X,{QE1},{DIR})
- $S(PL1) = \{PE1,PL1\}$



The RHA₄ Model

- Designed to interact with standard role-based models such as RBAC96
- Defines the relation
$$\mathbf{admin-authority} \subseteq R \times R$$
- If $(a,r) \in \mathbf{admin-authority}$, then we say
 - a is an *administrative role*
 - a *controls* r
- $C(a)$ denotes the set of roles controlled by a

The Extended Role Hierarchy

- (r,a) is an edge in the extended hierarchy if

(r,a) is an edge in the role hierarchy

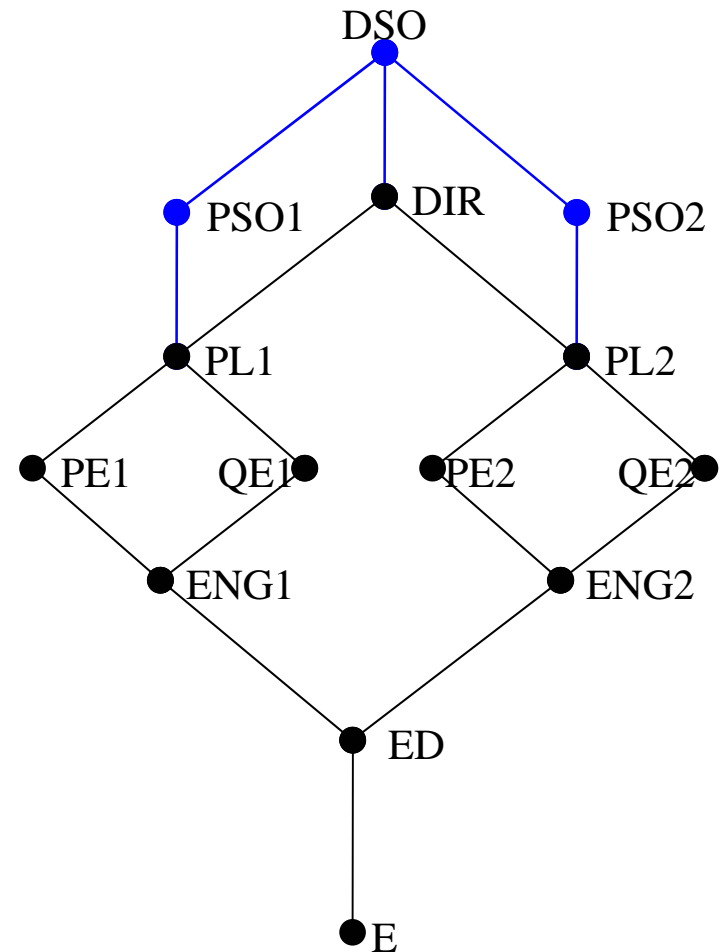
or

$(a,r) \in \mathbf{admin-authority}$

- Edges in the extended hierarchy **do not** imply inheritance

The Extended Role Hierarchy

- **admin-authority** =
{(DSO,PSO1),
(DSO,PSO2),
(DSO,DIR),
(PSO1,PL1),
(PSO2,PL2)}



Administrative Scope in RHA₄

- *Administrative scope* of a is

$$S(a) = \{s \in R : s \in \downarrow C(a), \uparrow s \setminus \uparrow C(a) \subseteq \downarrow C(a)\}$$

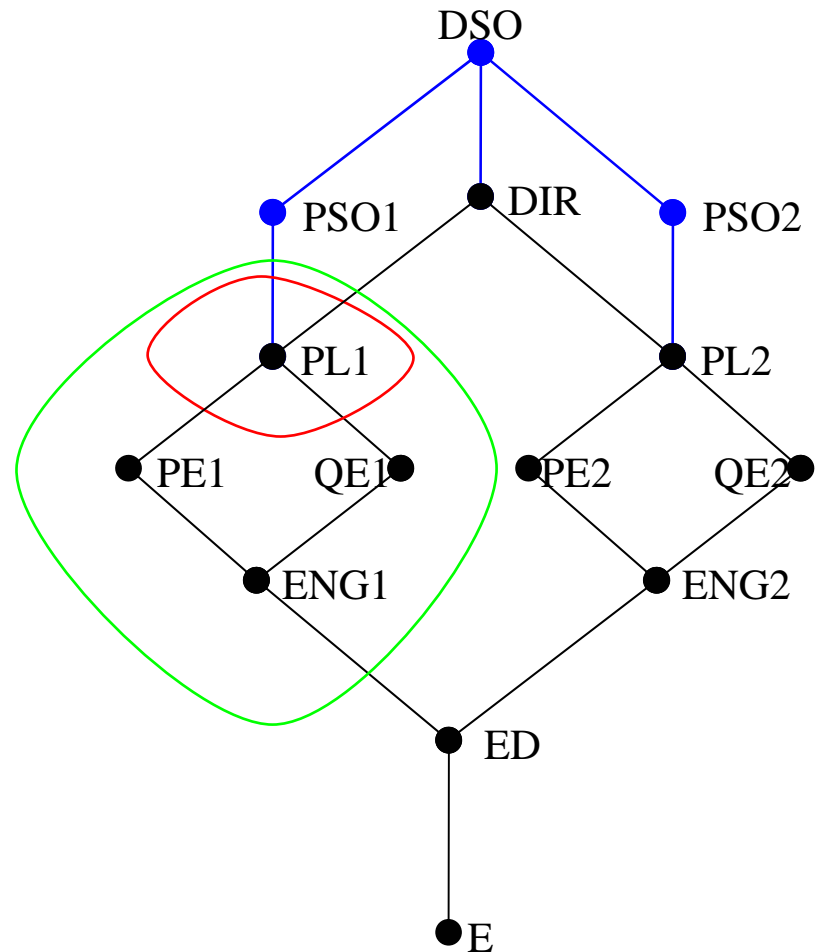
- *Proper administrative scope* of a is

$$S^+(a) = S(a) \setminus C(a)$$

- Evaluation of the up and down sets takes place in the extended hierarchy

Administrative Scope in RHA₄

- $C(\text{PSO1}) = \{\text{PL1}\}$
- $S(\text{PSO1}) = S(\text{PL1})$



Role Hierarchy Operations

- **AddEdge**(a, c, p) succeeds if
 - $c, p \in S(a)$
- **DeleteEdge**(a, c, p) succeeds if
 - $c, p \in S(a)$
- **AddRole**(a, r, C, P) succeeds if
 - $C \subseteq S^+(a)$ and $P \subseteq S(a)$
- **DeleteRole**(a, r) succeeds if
 - $r \in S^+(a)$

Updating the **admin-authority** relation

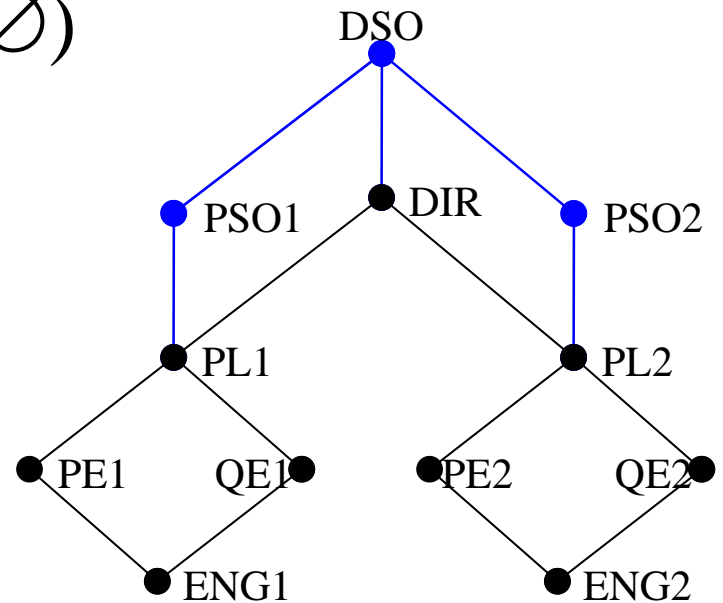
- (a,r) can be removed from **admin-authority** by b provided
 - $a \in S(b)$ and $r \in S^+(b)$
- (a,r) can be added to **admin-authority** by b provided
 - $a \in S(b)$ and $r \in S^+(b)$

Side Effects of Role Hierarchy Operations

- Hierarchy operations may have side effects on extended hierarchy
- **AddRole**(a, r, C, \emptyset)
 - Implies that r will not be in the administrative scope of any role because there are no roles greater than r
 - Hence (a, r) is added to **admin-authority**

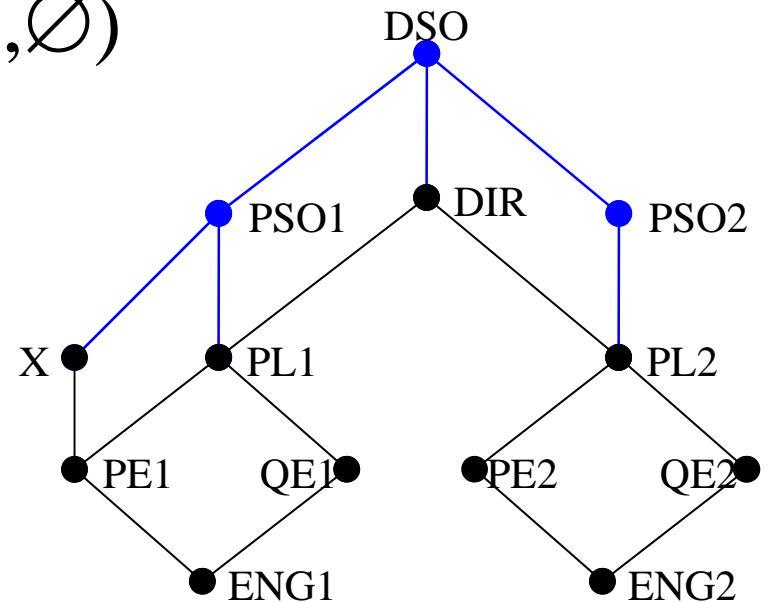
Side Effects of Role Hierarchy Operations

- **AddRole(PSO1,X,PE1,∅)**



Side Effects of Role Hierarchy Operations

- **AddRole(PSO1,X,PE1,∅)**



- (PSO1,X) is added to **admin-authority**

RHA₄ vs. ARBAC97

- Flexibility and simplicity
 - RHA₄ can be used for any hierarchy
 - ARBAC97 can only be used for hierarchies that contain encapsulated ranges
 - It is very easy to find role hierarchies that do not contain any encapsulated ranges
 - ARBAC97 requires that encapsulated ranges are preserved by hierarchy operations
 - For example, **AddRole**(?,X,{QE1},{DIR}) fails in ARBAC97
 - RHA₄ is considerably simpler and more intuitive than ARBAC97

RHA₄ vs. ARBAC97

- Dynamic aspects
 - Hierarchy operations in ARBAC97 controlled by **can-modify** relation
 - ARBAC97 assumes that **can-modify** is static
 - Administrative scope is a dynamic concept
 - **admin-authority** is dynamic; may be changed
 - Directly by administrative role
 - Indirectly as side effect of hierarchy operation
 - Constructing real hierarchies

RHA₄ vs. ARBAC97

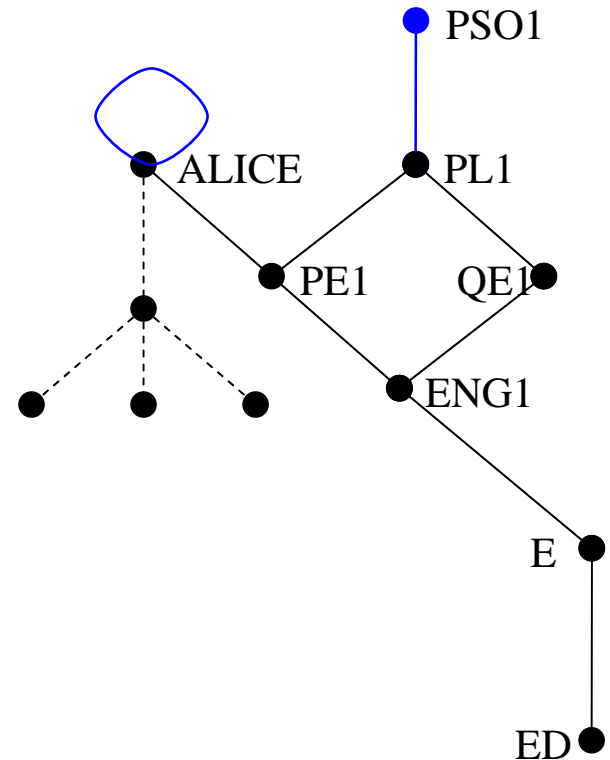
- Integration and extensibility
 - ARBAC97
 - URA97, PRA97 → RRA97
 - Hence the effect of hierarchy operations on URA97 and PRA97 relations is not always well defined
 - For example, hierarchy operations may change semantics of tuples in other ARBAC97 relations
 - RHA₄ deals with the difficult issue (ie, hierarchy administration) first
 - User- and permission-role assignment can be easily defined in terms of administrative scope

Future Work

- Role-based administration of user- and permission-role assignment
 - For example, **AssignUser**(a, r, u) is legitimate if r is in administrative scope of a
- Use of RHA_4 to model discretionary access control
 - Private hierarchy administered by “personal” role
- Use of RHA_4 to reduce inheritance in hierarchy

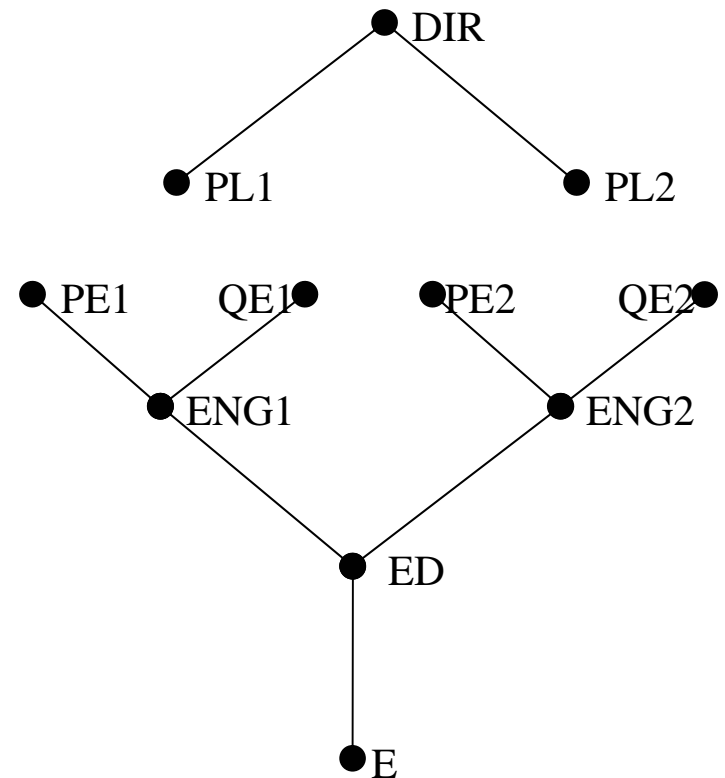
Private Hierarchies

- $(ALICE, ALICE) \in \text{admin-**authority**}$
- The role *ALICE* cannot administer *PE1*
- *ALICE* can administer the dotted (private) hierarchy
- Within private hierarchy, discretionary access control decisions can be taken by Alice (assigned to the *ALICE* role)



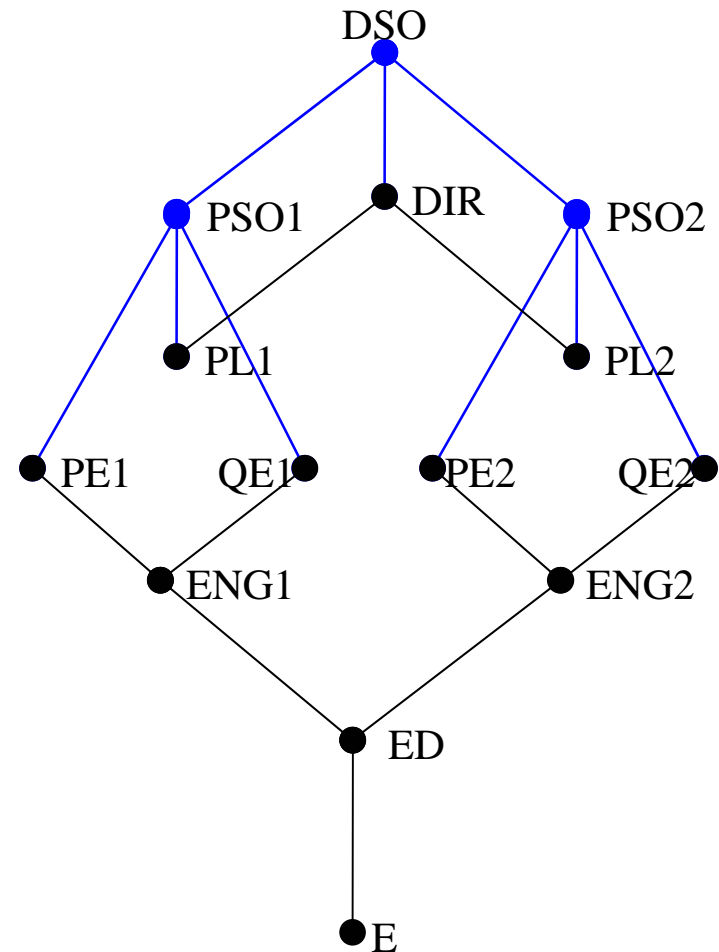
Reducing Inheritance

- Senior roles do not inherit the permissions of junior roles



Reducing Inheritance

- For a suitable **admin-authority** relation, it is possible to administer the role hierarchy, even though it is the disjoint union of two sets
- $S(\text{PSO1}) = \{\text{ENG1}, \text{PE1}, \text{QE1}, \text{PL1}\}$



Conclusions

- Administrative scope is an intuitive concept that identifies the set of roles that a given role can make changes to
- RHA_4 is dynamic, powerful model for role-based administration of the role hierarchy
- RHA_4 compares favourably with ARBAC97
- RHA_4 has several potential useful applications