**SCALE AND RATE OF CHANGE**

- **roles**: 100s or 1000s
- **users**: 1000s or 10,000s or more
- Frequent changes to
  - user-role assignment
  - permission-role assignment
- Less frequent changes for
  - role hierarchy
ARBAC97 DECENTRALIZES

- user-role assignment (URA97)
- permission-role assignment (PRA97)
- role-role hierarchy
  - groups or user-only roles (extend URA97)
  - abilities or permission-only roles (extend PRA97)
  - UP-roles or user-and-permission roles (RRA97)
ADMINISTRATIVE RBAC

Example Role Hierarchy

Director (DIR)
  ├── Project Lead 1 (PL1)
  │    ├── Production 1 (P1)
  │    └── Quality 1 (Q1)
  │         └── Engineer 1 (E1)
  ├── Engineering Department (ED)
  └── Project Lead 2 (PL2)
      ├── Production 2 (P2)
      └── Quality 2 (Q2)
            └── Engineer 2 (E2)

Employee (E)

Project 1

Project 2
EXAMPLE ADMINISTRATIVE ROLE HIERARCHY

Senior Security Officer (SSO)

Department Security Officer (DSO)

Project Security Officer 1 (PSO1)

Project Security Officer 2 (PSO2)

URA97 GRANT MODEL:

can-assign

<table>
<thead>
<tr>
<th>ARole</th>
<th>Prereq Role</th>
<th>Role Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSO1</td>
<td>ED</td>
<td>[E1,PL1)</td>
</tr>
<tr>
<td>PSO2</td>
<td>ED</td>
<td>[E2,PL2)</td>
</tr>
<tr>
<td>DSO</td>
<td>ED</td>
<td>(ED,DIR)</td>
</tr>
<tr>
<td>SSO</td>
<td>E</td>
<td>[ED,ED]</td>
</tr>
<tr>
<td>SSO</td>
<td>ED</td>
<td>(ED,DIR)</td>
</tr>
</tbody>
</table>
URA97 GRANT MODEL:  
**can-assign**

<table>
<thead>
<tr>
<th>ARole</th>
<th>Prereq Cond</th>
<th>Role Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSO1</td>
<td>ED</td>
<td>[E1,E1]</td>
</tr>
<tr>
<td>PSO1</td>
<td>ED &amp; ¬ P1</td>
<td>[Q1,Q1]</td>
</tr>
<tr>
<td>PSO1</td>
<td>ED &amp; ¬ Q1</td>
<td>[P1,P1]</td>
</tr>
<tr>
<td>PSO2</td>
<td>ED</td>
<td>[E2,E2]</td>
</tr>
<tr>
<td>PSO2</td>
<td>ED &amp; ¬ P2</td>
<td>[Q2,Q2]</td>
</tr>
<tr>
<td>PSO2</td>
<td>ED &amp; ¬ Q2</td>
<td>[P2,P2]</td>
</tr>
</tbody>
</table>

- “redundant” assignments to senior and junior roles
  - are allowed
  - are useful

© Ravi Sandhu 1999
URA97 REVOKE MODEL

- **WEAK REVOCATION**
  - revokes explicit membership in a role
  - independent of who did the assignment

- **STRONG REVOCATION**
  - revokes explicit membership in a role and its seniors
  - authorized only if corresponding weak revokes are authorized
  - alternatives
    - all-or-nothing
    - revoke within range
URA97 REVOKE MODEL:
can-revoke

<table>
<thead>
<tr>
<th>ARole</th>
<th>Role Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSO1</td>
<td>[E1,PL1)</td>
</tr>
<tr>
<td>PSO2</td>
<td>[E2,PL2)</td>
</tr>
<tr>
<td>DSO</td>
<td>(ED,DIR)</td>
</tr>
<tr>
<td>SSO</td>
<td>[ED,DIR]</td>
</tr>
</tbody>
</table>

PERMISSION-ROLE ASSIGNMENT

- dual of user-role assignment
- can-assign-permission
  can-revoke-permission
- weak revoke
  strong revoke (propagates down)
### PERMISSION-ROLE ASSIGNMENT

**CAN-ASSIGN PERMISSION**

<table>
<thead>
<tr>
<th>ARole</th>
<th>Prereq Cond</th>
<th>Role Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSO1</td>
<td>PL1</td>
<td>[E1,PL1)</td>
</tr>
<tr>
<td>PSO2</td>
<td>PL2</td>
<td>[E2,PL2)</td>
</tr>
<tr>
<td>DSO</td>
<td>E1 ∨ E2</td>
<td>[ED,ED]</td>
</tr>
<tr>
<td>SSO</td>
<td>PL1 ∨ PL2</td>
<td>[ED,ED]</td>
</tr>
<tr>
<td>SSO</td>
<td>ED</td>
<td>[E,E]</td>
</tr>
</tbody>
</table>

### PERMISSION-ROLE ASSIGNMENT

**CAN-REVOKE PERMISSION**

<table>
<thead>
<tr>
<th>ARole</th>
<th>Role Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSO1</td>
<td>[E1,PL1]</td>
</tr>
<tr>
<td>PSO2</td>
<td>[E2,PL2]</td>
</tr>
<tr>
<td>DSO</td>
<td>(ED,DIR)</td>
</tr>
<tr>
<td>SSO</td>
<td>[ED,DIR]</td>
</tr>
</tbody>
</table>

© Ravi Sandhu 1999
ARBAC97 DECENTRALIZES

- user-role assignment (URA97)
- permission-role assignment (PRA97)
- role-role hierarchy
  - groups or user-only roles (extend URA97)
  - abilities or permission-only roles (extend PRA97)
  - UP-roles or user-and-permission roles (RRA97)

Range Definitions
Authority Range

- **Range:**
  - \((x, y) = \{ r : \text{Roles} | x < r < y \}\)

- **Authority Range:**
  - A range referenced in can-modify relation

- **Partial Overlap of Ranges:**
  - The ranges Y and Y' partially overlap if
    - \(Y \cap Y' \neq \emptyset\) and
    - \(Y \not\subseteq Y' \land Y' \not\subseteq Y\)

- **Partial Overlap of Authority Ranges is forbidden**

Authority Range

- **Encapsulated Authority Range:**
  - The authority range \((x, y)\) is said to be encapsulated if
    - \(\forall r_1 \in (x, y)\) and \(\forall r_2 \notin (x, y)\)
    - \(r_2 > r_1 \iff r_2 > y\)
    - \(r_2 < r_1 \iff r_2 < x\)

© Ravi Sandhu 1999
Non-encapsulated Range (x, y)

Encapsulated Range (x, y)
Encapsulated Range \((x, y)\)

- New roles are created one at a time
- Creation of a role requires specification of immediate parent and child
  - Immediate parent and child must be a create range
Role Creation

**Create Range:**
- The range \((x, y)\) is a create range if
  - (a) \(\text{AR}_{\text{immediate}}(x) = \text{AR}_{\text{immediate}}(y)\)
  - (b) \(x = \text{End point of AR}_{\text{immediate}}(y)\)
  - (c) \(y = \text{End point of AR}_{\text{immediate}}(x)\)
- Note: only comparable roles constitute a create range.

Create Range

Authority ranges:
\((x, y)\) and \((B, A)\)
Role Deletion

◆ Roles in the authority range can be deleted by administrator of that range.
◆ End points of authority ranges cannot be deleted.

Inactive Roles

◆ End points of authority ranges can be made inactive.
◆ Inactive Roles:
  ● A user associated to it cannot use it.
  ● Inheritance of permissions is not affected.
  ● Permissions and users can be revoked.
Other Restrictions on deletion of roles

- Roles can be deleted only when they are empty.
- Delete the role and at the same time:
  - assign permissions to immediate senior roles.
  - Assign the users to immediate junior roles.

INSERTION OF AN EDGE

- Inserted only between incomparable roles (No Cycles)
- Inserted one at a time.
- The edge AB is inserted if
  - (a) $AR_{\text{immediate}}(A) = AR_{\text{immediate}}(B)$ and
  - (b) For a junior authority range $(x, y)$:
    - $(A = y \land B > x)$ or $(B = x \land A < y)$ must ensure encapsulation of $(x, y)$.  
  
© Ravi Sandhu 1999

29

30
DELETION OF AN EDGE

- Deleted one at a time.
- The edges in transitive reduction are candidates for deletion.
- Edges connecting the end points of an authority range cannot be deleted.
- Implied edges are not deleted

Example: Before deletion
(SQE1, JQE1)
Example: After deletion (SQE1, JQE1)

**Conclusion**

- RRA97 completes ARBAC97
- RRA97 provides decentralized administration of role hierarchies.
- Gives administrative role autonomy within a range but only so far as the side effects of the resulting actions are acceptable.