**User/Permission Assignments**

- $\text{URA} \subseteq U \times R$
- $\text{PRA} \subseteq A \times O \times R$

In an RBAC constraint program $\Pi$,

- $\Pi \models \text{ura}(u,r) \text{ iff } \langle u, r \rangle \in \text{URA}$;
- $\Pi \models \text{pra}(a, o, r) \text{ iff } \langle a, o, r \rangle \in \text{PRA}$;

**Denial-Role Assignments**

In the absence of overriding denials, the semantics of permission inheritance via an RBAC role hierarchy

$$\forall a_i \in A, \forall o_k \in O, \forall r_i \in R [\exists r_j \in R [\text{pra}(a_i, o_k, r_j) \land \exists \text{senior}_{\rightarrow}(r_i, r_j)]]$$

$\Rightarrow \text{pra}(a_i, o_k, r_j)$

**Inheritance Policies**

The authorization rule with restrictions on the upward inheritance of permission-role assignments is

- $\text{permitted}(U, A, O) \leftarrow \text{ura}(U, R_1), \text{active}(U, R_1), \text{senior}_{\rightarrow}(R_1, R_2), \text{pra}(A, O, R_2)$

Authorized rules include

- $\text{permitted}(U, A, O) \leftarrow \text{ura}(U, R_1), \text{active}(U, R_1), \text{senior}_{\rightarrow}(R_1, R_2), \text{pra}(A, O, R_2)$
**TRBAC**

- **URA**
  
  - ura(tom, r8, T) \[<\] 20040310 \[\leq\] T, T \[\leq\] 20041212, weekday(T,W), 2 \[\leq\] W, W \[\leq\] 5.

- **Authorization**
  
  - authorized(U,A,O) \[<\] date(T), ura(U,R1,T), active(U,R1), senior_to(R1,R2), pra(A,O,R2,T), not denied(R1,A,O,T).
  
  - denied(R1,A,O,T) \[<\] senior_to(R3,R1), dra(A,O,R3,T).

---

**Delegation Logic**

- **Authorization** = proof-of-compliance

- **Why delegation?**
  
  - The resource owner does NOT know the requester directly
  
  - The owner may use information from third parties who know the requester better

---

**D1LP**

- **Monotonic Version of Delegation Logic**

- **Example**
  
  - Bob delegates goodCredit(?X)^1 to Carl

  Bob trusts Carl about whether someone has good credit. If Carl says that someone has good credit, then Bob believes it.

  - Bob delegates goodCredit(?X)^2 to Carl

  Bob trusts Carl’s ability to judge other principals’ ability to judge whether someone has good credit.

---

**Example**

- **Determining Credit Status**

  ShopA says approveOrder(?U) if ShopA says creditRating(?U, good).

  ShopA delegates creditRating(?U,?R)^2 to BankB.

  BankB says creditRating(?U,good) if threshold(2, [cardW,cardX,cardY]) says accountGood(?U).
Conflicts

- Definition Conflicts
  - When authorization rules are created, conflicts may happen due to multiple designers can create rules or rules are created in distributed/remote sites

- Derivation Conflicts

- Propagation Conflicts

Derivation Conflicts

- When authorization rules are derived, conflicts may happen.

\[ \text{AR1: } \text{Auth} (O: \text{order.xml/order/item[position()=1]}, S:\text{engineer}, P:\text{write}, G:g33, \omega:+) \leftarrow \text{Auth} (O: \text{order.xml/order/item[position()=1]}, S:\text{sales}, P:\text{read}, G:g33, \omega:+) \]

\[ \text{AR1': } \text{Auth} (O: \text{order.xml/order/item[position()=1]}, S:\text{engineer}, P:\text{write}, \omega:-) \leftarrow \text{Auth} (O: \text{order.xml/order/item[position()=1]}, S:\text{sales}, P:\text{read}, \omega:+) \]

Propagation Conflicts

- When authorized subjects or objects are propagated, conflicts may happen.

\[ \text{AR1: } \text{Auth} (O: \text{order.xml/order/}, S:\text{engineer}, P:\text{write}, G:g33, \omega:+) \leftarrow \text{Auth} (O: \text{order.xml/order/}, S:\text{sales}, P:\text{read}, G:g33, \omega:+) \]

\[ \text{AR7': } \text{Auth} (O: \text{order.xml/order/}, S:u33, P:\text{write}, G:g33, \omega:+) \leftarrow \text{Auth} (O: \text{order.xml/order/}, S:\text{sales}, P:\text{read}, G:g33, \omega:+) \]

\[ \text{AR7''': } \text{Auth} (O: \text{order.xml/order/item[position()=1]}, S:\text{engineer}, P:\text{write}, G:g33, \omega:+) \leftarrow \text{Auth} (O: \text{order.xml/order/item[position()=1]}, S:\text{sales}, P:\text{read}, G:g33, \omega:+) \]

Authorization Conflicts

- DS = (OTH, UGH, RH, A, Rel)
  - DS: Data System; OTH: object type hierarchy; RH: role hierarchy; A: authorization mode (action); Rel: relationship (unary, binary,...)

- Authorization: \( (o,s,\langle\text{sign}\rangle a) \)
  - \( o \in \text{AO} \) (authorization object), \( s \in \text{AS} \) (authorization subject), \( a \in A \), \( \text{sign} \in \{+/-\} \)

- Propagation policy
  - No propagation, No overriding, Most specific overrides, Path overrides

Propagation

\[ G_1(\alpha, \pm a) \]

\[ (\alpha, -a) \quad G_2 \quad G_3 \quad G_4(\alpha, -a) \]

\[ G_5 \quad G_6(\alpha, -a) \]

\[ u_1 \quad u_2 \]