ARPACAL

REGIONE CALABRIA

REPUBBLICA ITALIANA

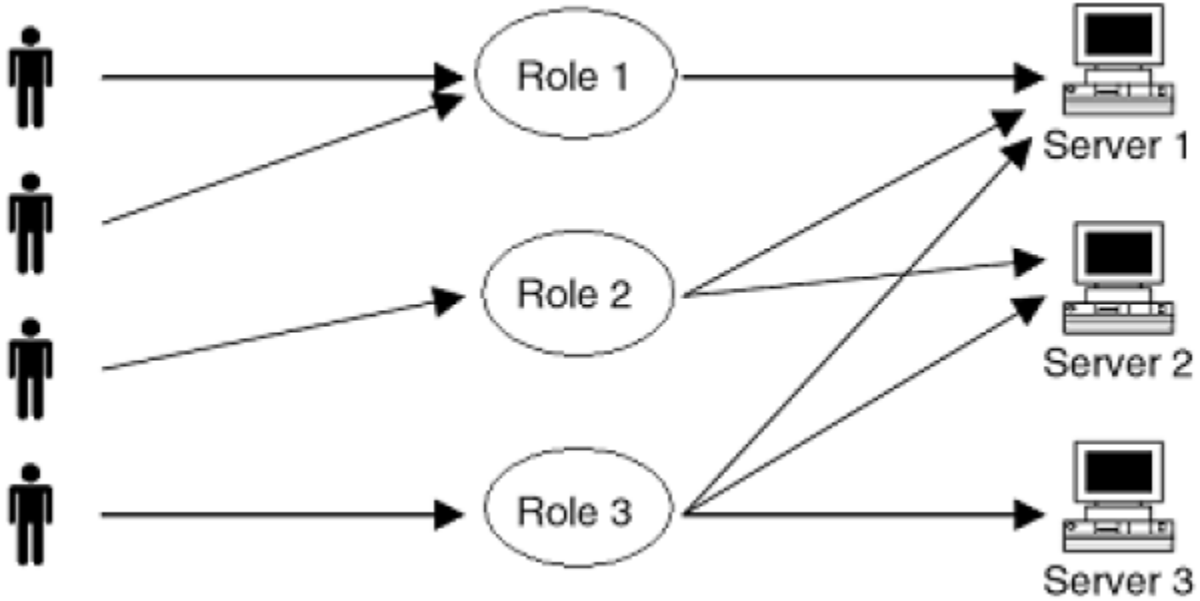
ALLEGATO 018

MODELLO RBAC (ROLED BASED ACCESS CONTROL)

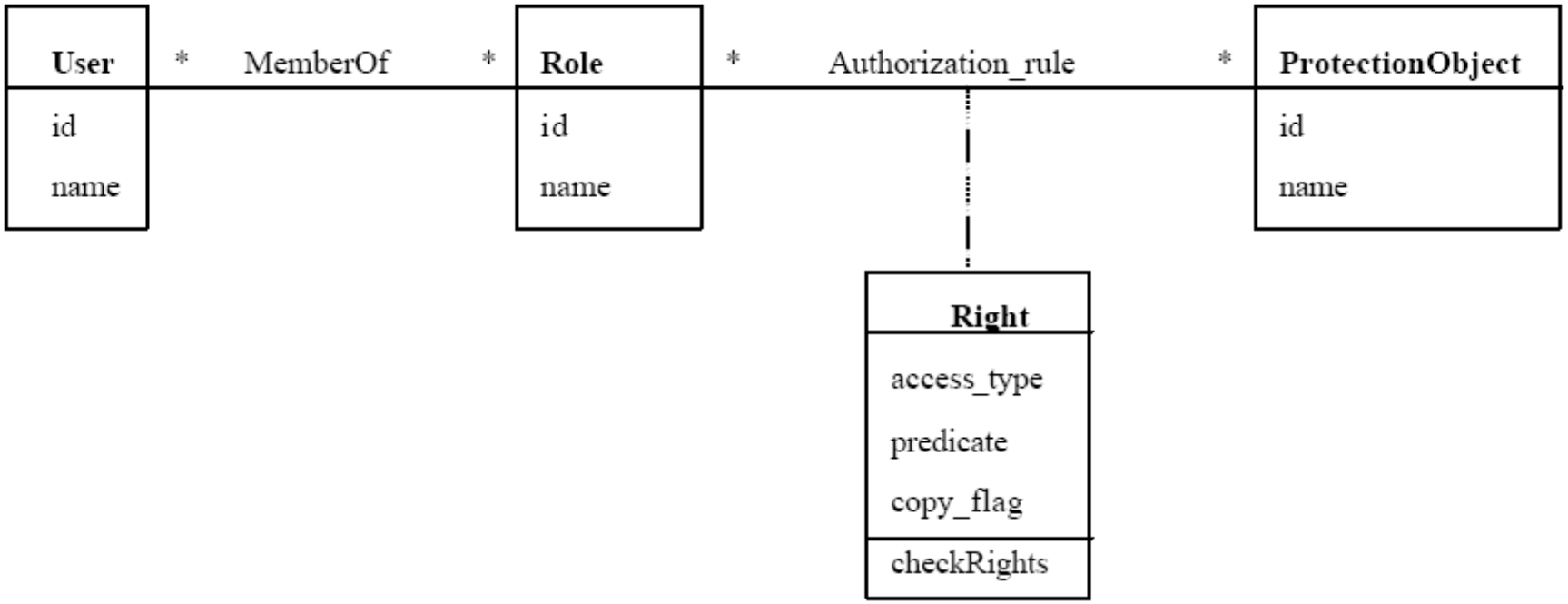
Modello di Controllo dell’Accesso basato sui ruoli (RBAC)

POLITICHE RBAC

- Sistemi di tipo Role Based Access Control (RBAC) assegnano i privilegi non agli utenti, ma alla funzione che questi possono svolgere nel contesto di una certa organizzazione
- L'utente acquista quindi privilegi assumendo uno o più ruoli



POLITICHE RBAC



RBAC

- RBAC consente di supportare facilmente i ben conosciuti principi di sicurezza:
  - *minimo privilegio (Least Privilege)*
  - *separazione dei compiti (Separation of duties):* l'utilizzo di ruoli mutuamente esclusivi potrebbe essere necessario in certe situazioni critiche (evitare che il "controllato" sia anche il "controllore")

  Inoltre RBAC permette di avere:

  - *astrazione dei dati (Data abstraction)*: invece dei tipici permessi "read", "write", "execute" utilizzati nei sistemi operativi, possono essere stabiliti permessi astratti come per esempio "crediti" e "debiti"

RUOLO

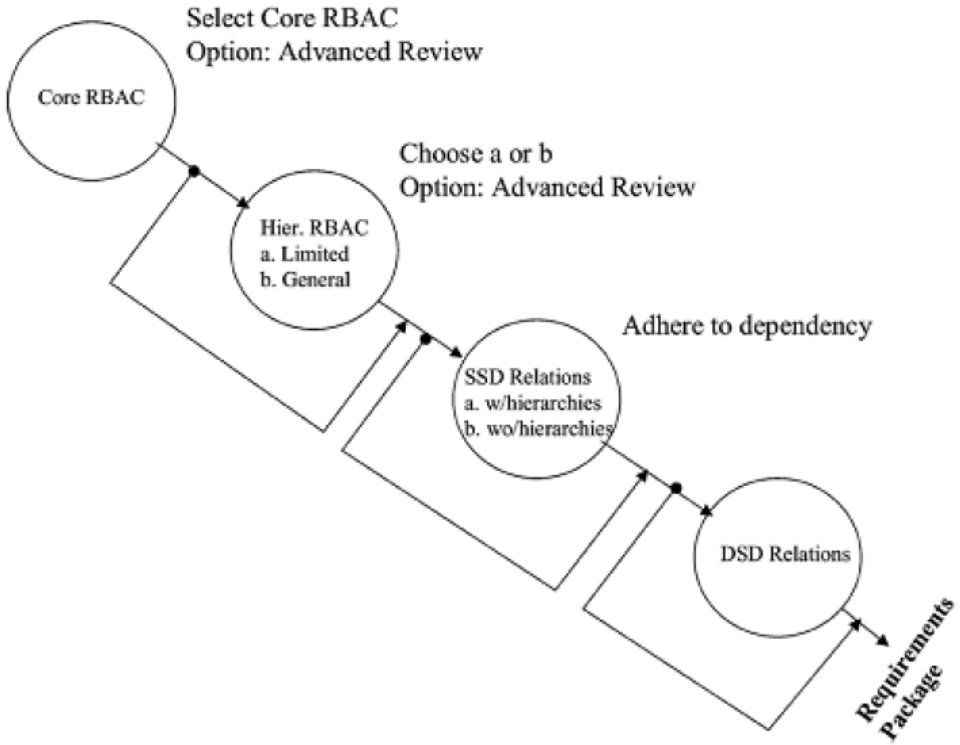
- In RBAC un *ruolo* è visto come un costrutto semantico attorno al quale vengono formulate le politiche di controllo d'accesso
- Il concetto di ruolo ha diversi significati:
  - **rappresenta la *competenza* nel compiere una specifica attività (per esempio "farmacista" o "fisico")**
  - **incorpora sia l'*autorità* che la *responsabilità* (per esempio "responsabile di progetto")**

RBAC

- Un sistema RBAC correttamente amministrato fornisce una *grande flessibilità* agli amministratori di sistema con il minimo sforzo:
  - i ruoli nell'organizzazione sono praticamente fissi, o variano molto raramente
  - stabiliti inizialmente i permessi per ogni ruolo...
  - tutto quello che deve fare l'amministratore è gestire l'assegnazione degli utenti ai ruoli
- Questo si traduce in un grande vantaggio rispetto alle politiche DAC e MAC

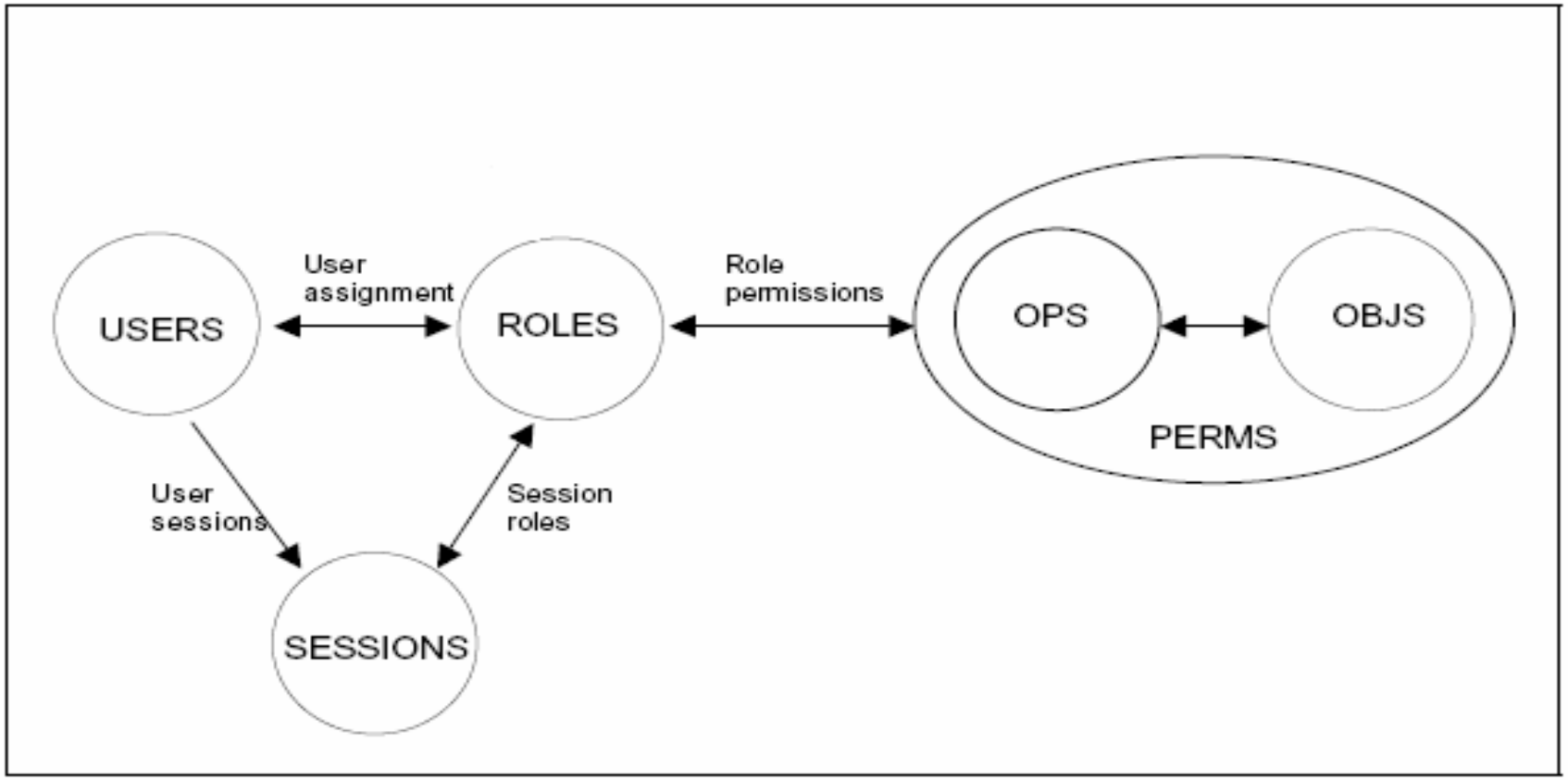
LO STANDARD

- RBAC è stato standardizzato come ANSI/INCITS 359-2004
- In questo documento vengono proposti quattro modelli RBAC in modo incrementale



CORE RBAC

- In questo primo modello vengono definiti solo gli elementi essenziali senza i quali non è possibile implementare un controllo d'accesso.



CORE RBAC: DETTAGLI

- Utente: tipicamente è un umano ma potrebbe per esempio essere anche un agente software
- Ruolo: è una “funzione lavorativa” all’interno di un sistema (organizzazione) che descrive le autorità e le responsabilità conferite al “membro” del ruolo
- Permesso: è l’approvazione di un particolare modo di accesso ad uno o più oggetti (risorse) del sistema (organizzazione)
- Sessione: l’utente stabilisce una sessione durante la quale può attivare un sottoinsieme dei ruoli che gli appartengono. Ogni sessione mappa un utente sui possibili ruoli che può attivare

CORE RBAC: DETTAGLI

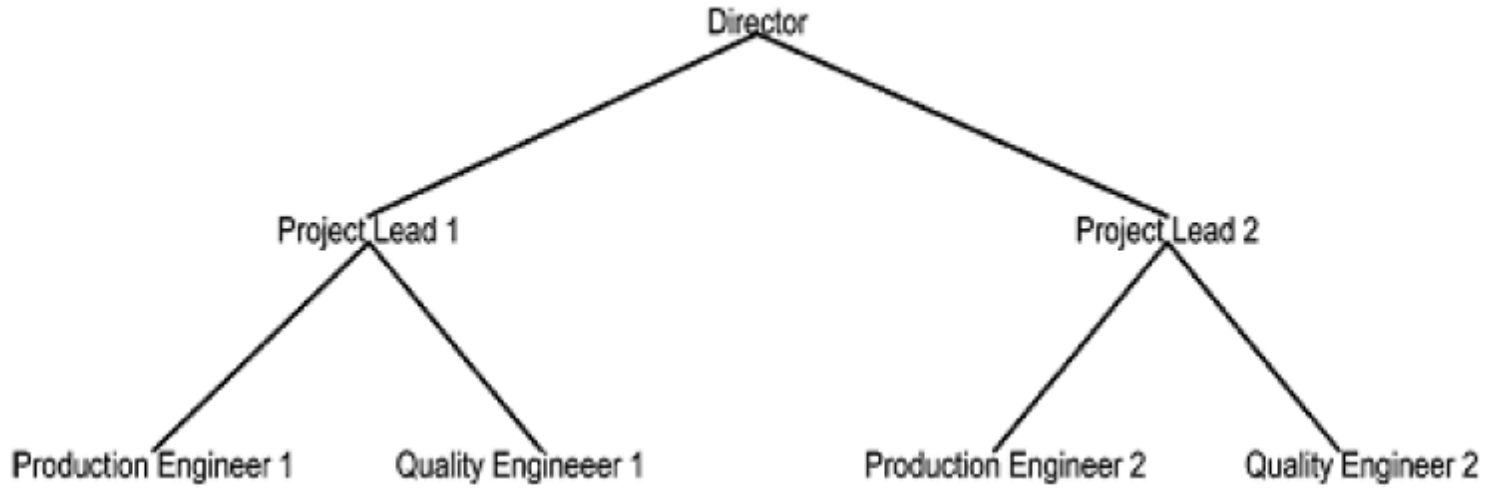
- Se per un dato utente esiste una relazione con un ruolo, non vuol dire che l'utente appartenga a quel ruolo sempre, ma solo che in generale gli è permesso di farne parte
- L’utente assumerà quel ruolo solo attivando la corrispondente sessione (e lo abbandonerà disattivandola)
- Un utente può creare una o più sessioni, all'interno di ognuna delle quali può attivare uno o più ruoli, scegliendo tra quelli che gli sono stati messi a disposizione
- I permessi disponibili all’utente sono l’unione dei permessi associati ai ruoli che sono attualmente attivi nella sessione

CORE RBAC: DETTAGLI

- Non esiste il permesso di compiere un'operazione in generale, ma, per ogni risorsa, esiste un permesso per ogni singola operazione che è possibile eseguire su di essa
- È evidente che la stessa operazione può essere associata a risorse diverse
- Implementare un modello Core RBAC vuol dire fornire un sistema con cui è possibile interagire mediante:
  - funzioni di amministrazione (creare e rimuovere utenti, ruoli, risorse permessi...)
  - funzioni di supporto (creare sessioni, aggiungere/rimuovere ruoli attivi)
  - funzioni di monitoraggio (controllare il sistema)

RBAC GERARCHICO

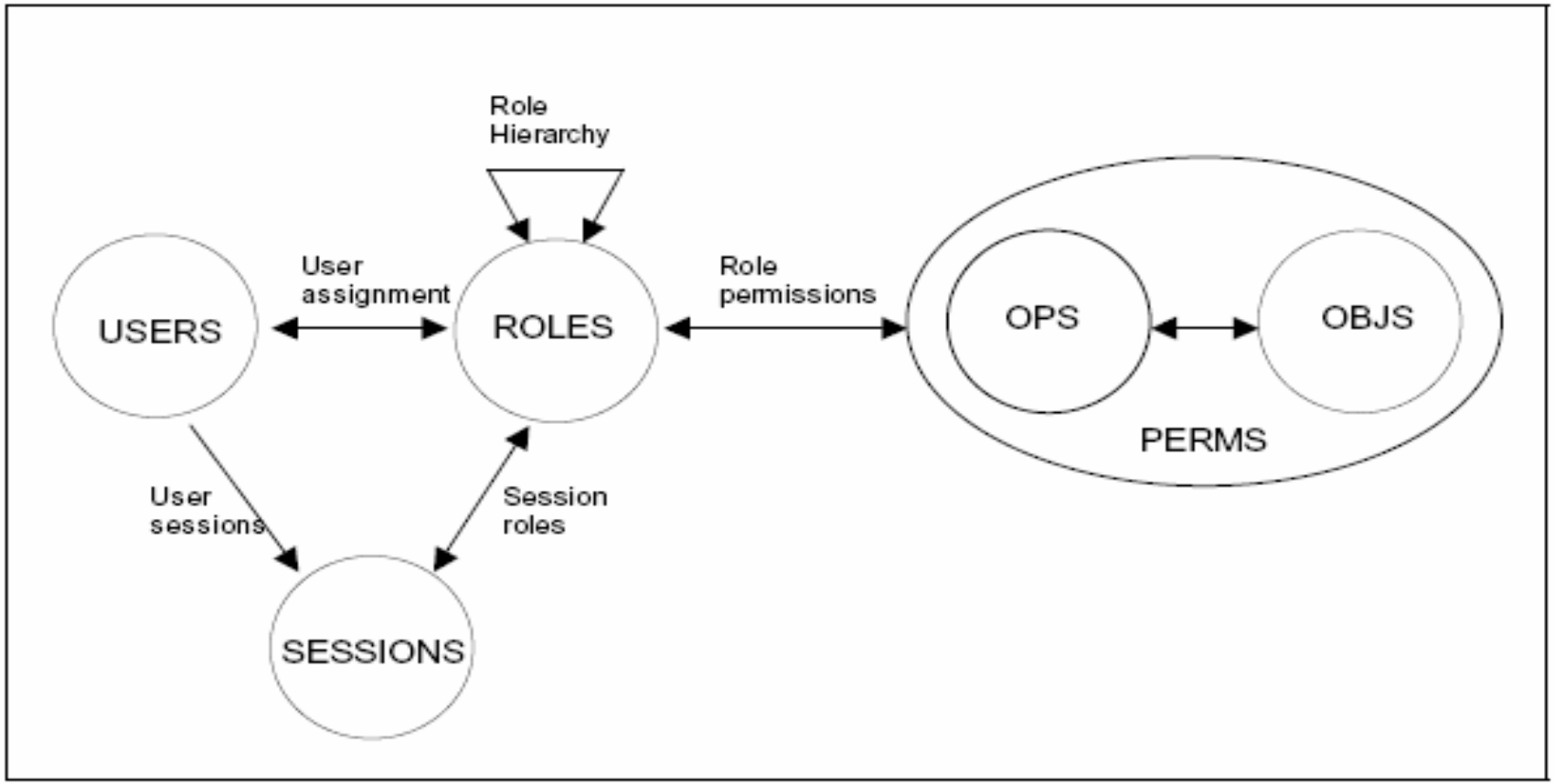
- Una gerarchia è un modo naturale di strutturare i ruoli all'interno di una organizzazione che rispecchia l'effettiva responsabilità e autorità di ognuno



- A questa gerarchia di ruoli corrisponde di solito una effettiva ereditarietà di permessi, ovvero, salendo nella gerarchia, i vari ruoli possiedono tutti i permessi dei ruoli sottoposti, oltre ai propri

RBAC GERARCHICO

- Per tenere conto di queste situazioni molto comuni è stato introdotto il modello RBAC Gerarchico che estende il Core RBAC introducendo una relazione gerarchica tra ruoli



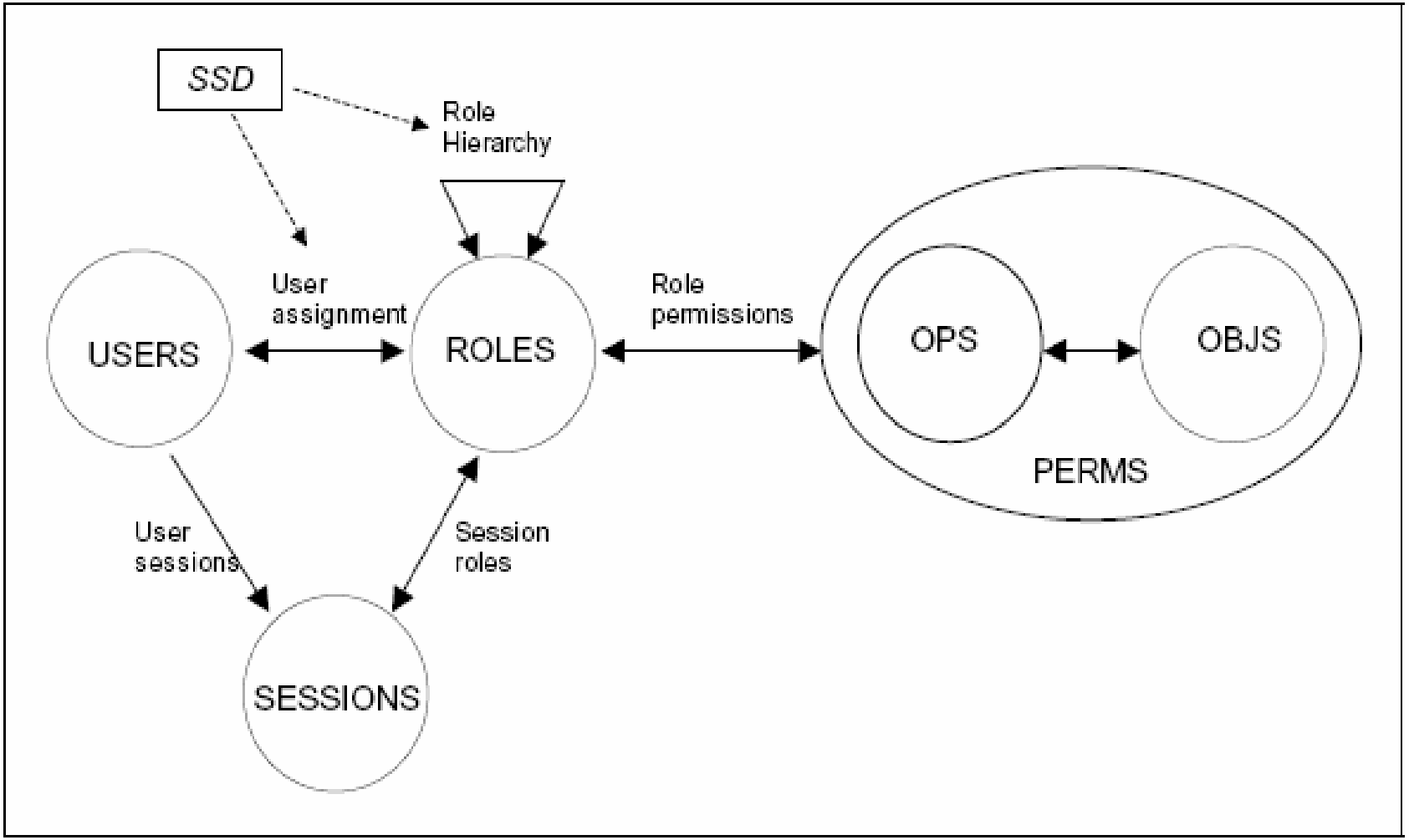
RBAC GERARCHICO: DETTAGLI

- Il significato di tale relazione è:
  - dati due ruoli r1 ed r2, r1 "*eredita*" il ruolo r2 se ogni permesso del ruolo r2 è anche un permesso del ruolo r1
- La relazione è di tipo molti-a-molti per cui
  - un ruolo può *ereditare* i permessi da più ruoli
  - un ruolo può *essere ereditato* da più ruoli
- Lo standard non specifica nulla sul tipo di gerarchia ammessa, qualsiasi ordine parziale imposto sull'insieme dei ruoli è valido
- Vengono estese le funzioni viste in precedenza al fine di poter gestire le relazioni gerarchiche

RBAC CON VINCOLI SSD

- In organizzazioni in cui gli utenti possono assumere ruoli diversi è possibile che sorgano problemi di conflitti di interessi
- Esempio: un utente può assumere sia il ruolo di controllore che di controllato su una certa transazione
- Spesso vengono quindi imposti a priori dei vincoli di separazione dei compiti (SSD- Static Separation of Duty)

RBAC CON VINCOLI SSD



RBAC CON VINCOLI SSD

- In RBAC è possibile definire vincoli SSD sia sulla relazione utente-ruolo che sulla relazione gerarchica tra ruoli: è possibile escludere a priori dei ruoli sia tra quelli assegnabili direttamente ad un certo utente, sia tra quelli da cui può ereditare dei permessi
- Un vincolo SSD è espresso come un insieme di coppie (rs,n)
  - rs: è un sottoinsieme di ruoli
  - n: è un numero intero maggiore di 1
- Una coppia di questo tipo specifica che nessun utente può essere assegnato (direttamente o tramite ereditarietà) a n o più ruoli nel sottoinsieme **rs**

VINCOLI TIPICI

- Ruoli Mutuamente esclusivi
  - allo stesso utente deve essere assegnato al massimo un ruolo di quelli presenti in un insieme mutuamente esclusivo
  - questo supporta il principio della “Separazione dei compiti”
- Esiste anche il vincolo duale che riguarda i permessi
  - lo stesso permesso deve essere assegnato ad al massimo un ruolo di quelli presenti in un insieme mutuamente esclusivo
  - questo permette di limitare la distribuzione del “potere”

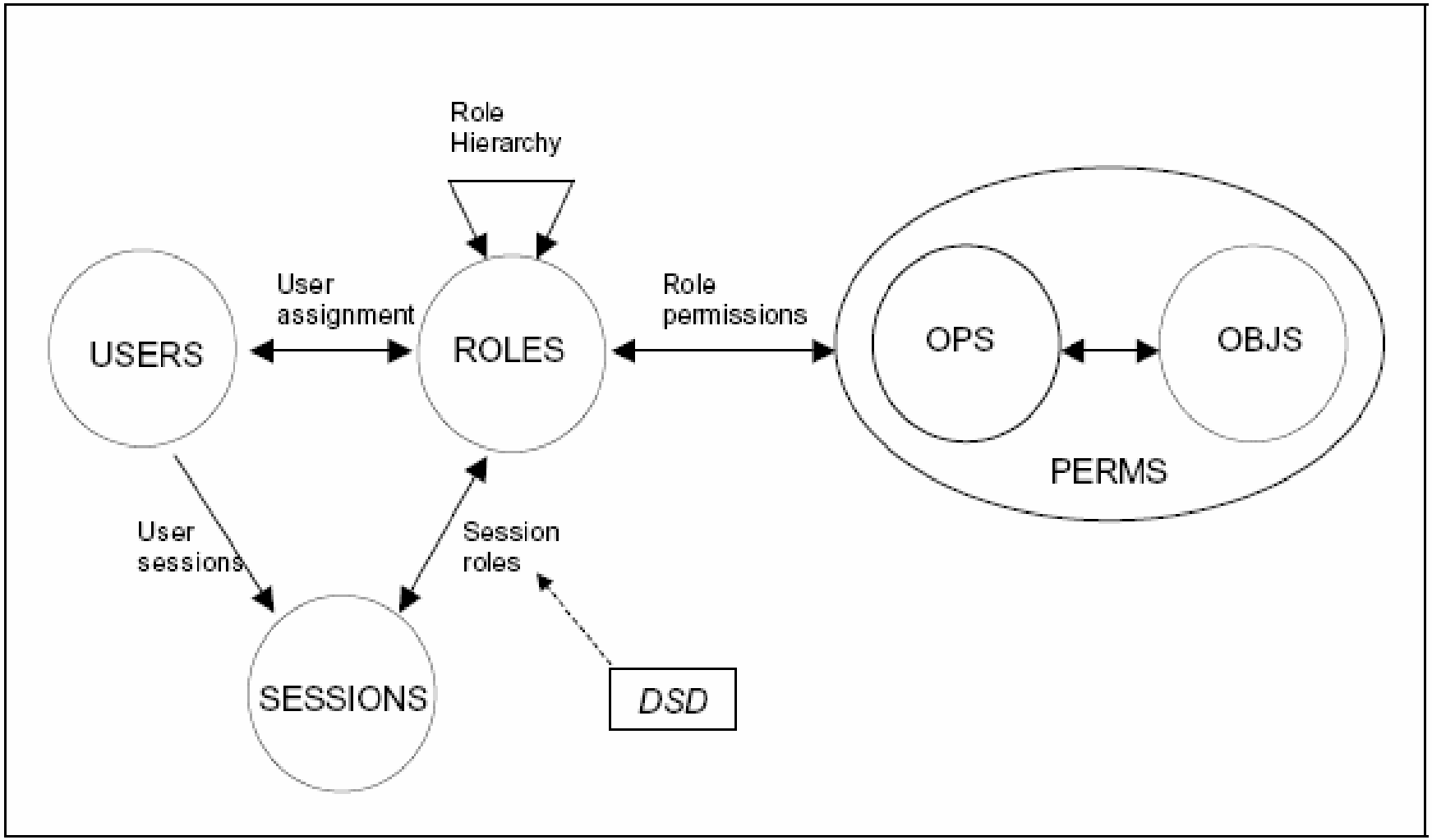
VINCOLI TIPICI

- Cardinalità
  - riguarda il massimo numero di membri appartenenti ad un ruolo
  - la cardinalità minima in genere non viene considerata poiché è molto difficile da controllare
- Ruoli Prerequisiti
  - ad un utente può essere assegnato il ruolo A se gli è già stato assegnato il ruolo B
  - questo vincolo è basato sulla competenza e appropriatezza
- Dualmente per i permessi
  - il permesso P può essere assegnato ad un ruolo solo se questo ruolo è già in possesso del permesso Q

RBAC CON VINCOLI DSD

- Spesso i vincoli imposti staticamente sono troppo restrittivi per il sistema...
- Non si vuole limitare a priori il numero di ruoli che i vari utenti possono assumere, ma solo quelli che posso essere attivati contemporaneamente
- Oppure, i vincoli imposti staticamente potrebbero non bastare ed occorre introdurne altri in fase di esecuzione
- Per questo è stata introdotta una versione di RBAC con vincoli dinamici di separazione dei compiti (DSD- Dynamic Separation of Duty)
- Questi coinvolgono ovviamente il concetto di sessione, unico elemento che lega l'utente al ruolo che assume

RBAC CON VINCOLI DSD



RBAC CON VINCOLI DSD

- Questa versione di RBAC supporta pienamente il principio del “minimo privilegio”
- Attraverso vincoli dinamici è possibile impedire che un utente attivi contemporaneamente più ruoli (e quindi acquisisca più privilegi) di quelli che gli sono strettamente necessari al momento