# فهرست مطالب

- مقدمه
- مدل‌های کنترل دسترسی اختیاری
- مدل‌های کنترل دسترسی اجباری
  - مدل حفظ محرمانگی
  - مدل حفظ صحت
- مدل‌های کنترل دسترسی نقش-مبنا

# تعاریف

- مدل کنترل دسترسی (مجازشماری)
  - تعریف: انتزاعی از خط‌مشی‌های کنترل دسترسی
  - بیانگر ساختار داده‌ای و زبان توصیف خط‌مشی‌های کنترل دسترسی
  - نوع خط‌مشی‌ها در کاربردهای مختلف، متفاوت است، لذا نوع مدل‌های کنترل دسترسی حاصله نیز متفاوت است.
- مکانیزم (اِعمال) کنترل دسترسی
  - تعریف: روش و سیستم اِعمال کنترل دسترسی بر اساس خط‌مشی‌های توصیف شده در قالب یک مدل کنترل دسترسی
  - مبتنی است بر یک مدل کنترل دسترسی

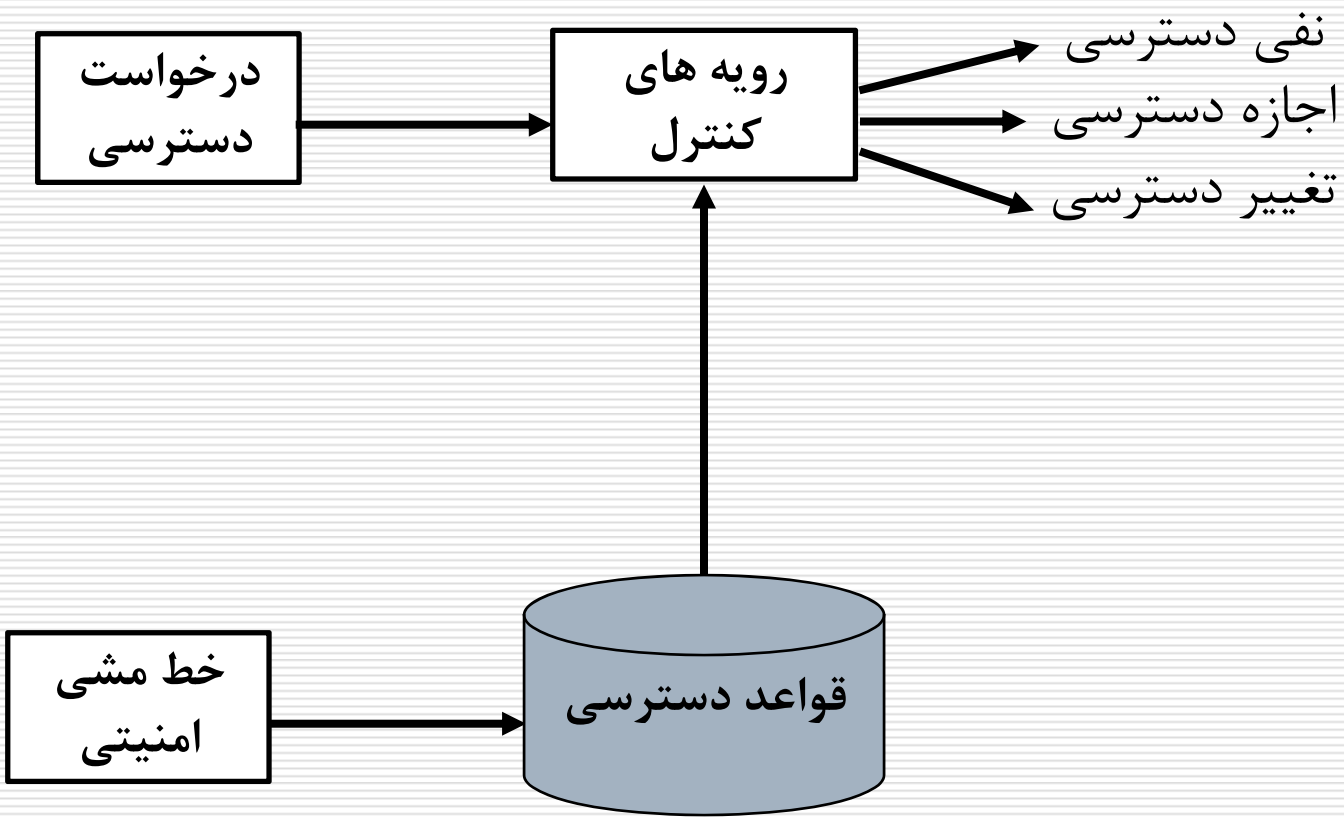
# عناصر اصلی

- موجودیت‌های اصلی دخیل در کنترل دسترسی:
  - عامل (Subject): هر آنکه متقاضی دسترسی است.
    - عامل انسانی، عامل ماشینی، پردازه، وب سرویس و . . .
  - شیئ یا منبع (Object or Resource): هر آنچه مورد دسترسی قرار می‌گیرد.
    - فایل، جدول پایگاه داده، پردازه، پردازنده، . . .
  - عمل (Action): عملی که توسط عامل بر روی شیئ یا منبع انجام می‌شود.
    - خواندن، نوشتن، تغییر، حذف، چاپ، . . .
- عامل عنصری فعال (Active) و شیئ عنصری منفعل (Passive) است.
  - یک عنصر می‌تواند هم نقش عامل را داشته باشد و هم نقش شیئ.
    - مثال: پردازه در سیستم عامل، وب‌سرویس در محیط وب

# خط‌مشی کنترل دسترسی

- خط‌مشی کنترل دسترسی: چه عامل‌هایی اجازه انجام چه اعمالی را بر روی چه اشیایی دارند و یا ندارند.
- در قالب مجموعه‌ای قاعده دسترسی بیان می‌گردد.
  - علی اجازه خواندن و تغییر به اطلاعات حقوق افراد را دارد.
  - کارمندان عادی اجازه خواندن قراردادهای شرکت را ندارند.
  - سیستم‌های درون سازمان (به غیر از سرورها) اجازه برقراری ارتباط با شبکه‌های بیرونی را ندارند.

# مکانیزم کنترل دسترسی



# انواع مدل‌های کنترل دسترسی

- بر اساس معیارهای مختلفی می‌توان مدل‌ها را دسته‌بندی کرد.
- انواع مدل‌های کنترل دسترسی بر حسب نحوه انتشار حقوق:
  - مدل کنترل دسترسی اختیاری (DAC)
  - مدل کنترل دسترسی اجباری (MAC)
  - مدل کنترل دسترسی نقش-مبنا (RBAC)

# فهرست مطالب

# مدل کنترل دسترسی اختیاری

- خصوصیات اصلی مدل‌های کنترل دسترسی اختیاری:
  - مبتنی بر شناسه و نیاز به شناخت از کاربر و احراز هویت آن
    - *مثال: حسن اجازه خواندن فایل x.doc را دارد.*
    - *مثال: علی اجازه تغییر جدول y را در پایگاه داده‌ها ندارد.*
  - نحوه انتصاب مجوزهای دسترسی به منابع در اختیار مالک است.
    - *مثال: مالک x.doc مجوز دسترسی خواندن را به حسن می‌دهد.*

# مدل کنترل دسترسی اختیاری

- **مزایا:**
  - سادگی، انعطاف‌پذیری
- **معایب:**
  - عدم کنترل جریان اطلاعات و کانال‌های مخفی، عدم کنترل استنتاج
  - سختی مدیریت: مدیر با حجم زیادی از مجوزها و افراد سر و کار دارد.
- **کاربرد:**
  - سیستم‌های کاربردی تجاری که فاقد طبقه‌بندی اطلاعات هستند.
  - سیستم‌های متمرکز با کاربران شناخته شده.

# مدل کنترل دسترسی اختیاری

- انواع مدل‌های کنترل دسترسی اختیاری بر حسب اینکه مجوز پیش‌فرض چه باشد:
  - **مدل‌های باز:** یک عامل به یک شیئ دسترسی دارد مگر آنکه خلاف آن در قواعد دسترسی بیان شده باشد.
  - **مدل‌های بسته:** یک عامل به یک شیئ دسترسی ندارد مگر آنکه در قواعد دسترسی، مجوز دسترسی به آن شیئ صادر شده باشد.

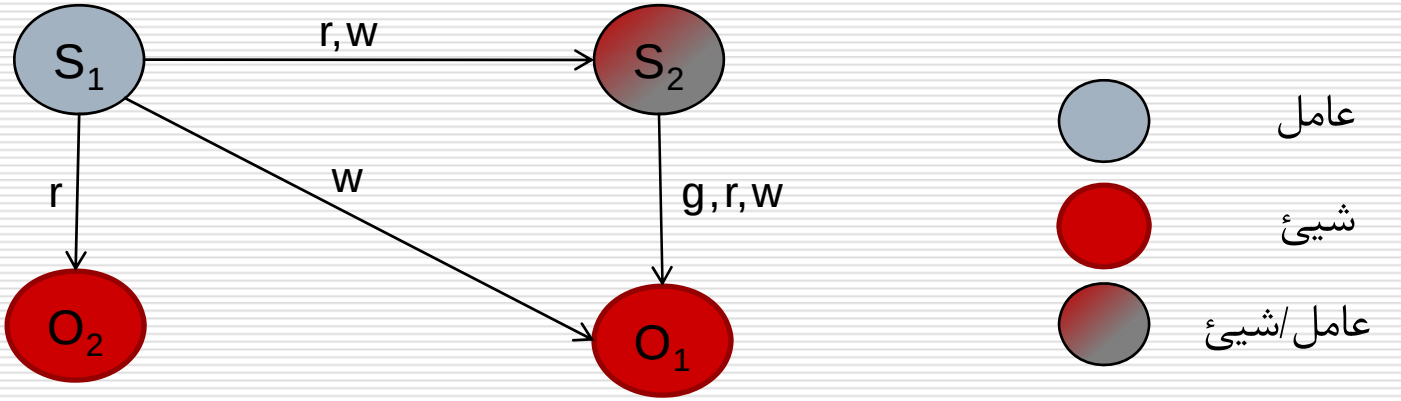
# مدل کنترل دسترسی اختیاری

- مدل‌های ماتریس-مبنا
  - اولین بار توسط لمپسون در سال ۱۹۷۱ ارائه شد.
  - هر سطر مربوط به یک عامل و هر ستون مربوط به یک شیئ است.
  - هر درایه ماتریس، مجوزهای دسترسی یک عامل را به یک شیئ نشان می‌دهد.

| Rights | Objects | | | |
|---|---|---|---|---|
| Subjects | | $O_1$ | $O_2$ | $O_3$ |
| | $S_1$ | +r | -r, +w | +r, +w |
| | $S_2$ | -w | +w | +r, -w |

# مدل کنترل دسترسی اختیاری

- مدل‌های گراف-مبنا
  - نمونه بارز آن مدل Take-Grant است.
  - رئوس گراف نمایانگر عامل‌ها و اشیاء است.
  - یال‌ها نمایانگر مجوزهای دسترسی است (g=grant، r=read، w=write).



# مدل کنترل دسترسی اختیاری

- مدل‌های قاعده-مبنا
  - بیان مجوزهای دسترسی با مجموعه‌ای از قواعد
  - در محیط‌های جدید محاسباتی بیشتر کاربرد دارد.
  - امکان تعریف شرایط دسترسی را نیز فراهم می‌نماید.
  - برای پیاده‌سازی از سیستم‌های خبره می‌توان استفاده کرد.

| |
|---|
| If 8am<time<4pm, then access($S_1$, $O_1$, +r) |
| If $S_1$.Location=Univ., then access($S_1$, Wireless, +connect) |

# مکانیزم‌های کنترل دسترسی اختیاری

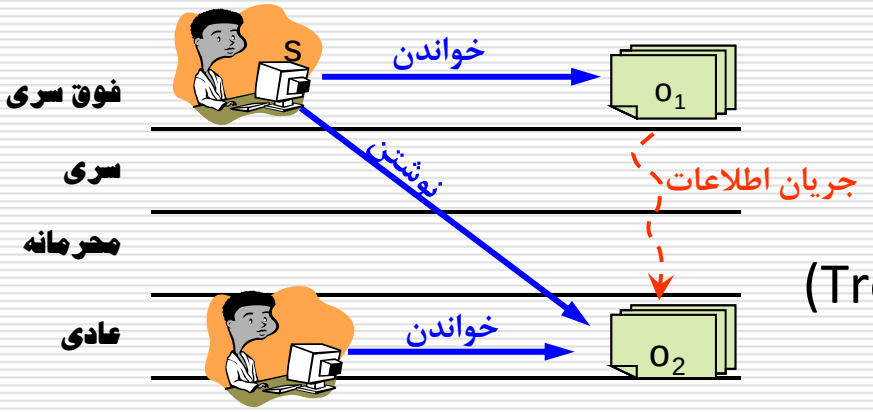
- پیاده‌سازی مکانیزم‌های کنترل دسترسی مبتنی بر مدل کنترل دسترسی اختیاری بر دو روش استوار است:
  - مبتنی بر توانایی (Capability based)
    - مجوزهای هر عامل را خود عامل به صورت یک مجموعه از توانایی‌ها در دست دارد و جهت دسترسی آنها را ارائه می‌نماید.
    - *مثال: کنترل دسترسی به سرویس‌ها در کربروس*
  - مبتنی بر لیست کنترل دسترسی (Access Control List)
    - لیست عامل‌ها و مجوزها آنها در کنار هر شیئ یا منبع قرار می‌گیرد.
    - *مثال: پیاده‌سازی کنترل دسترسی در لینوکس*

# فهرست مطالب

- مقدمه
- مدل‌های کنترل دسترسی اختیاری
- **مدل‌های کنترل دسترسی اجباری**
  - مدل حفظ محرمانگی
  - مدل حفظ صحت
- مدل‌های کنترل دسترسی نقش-مبنا

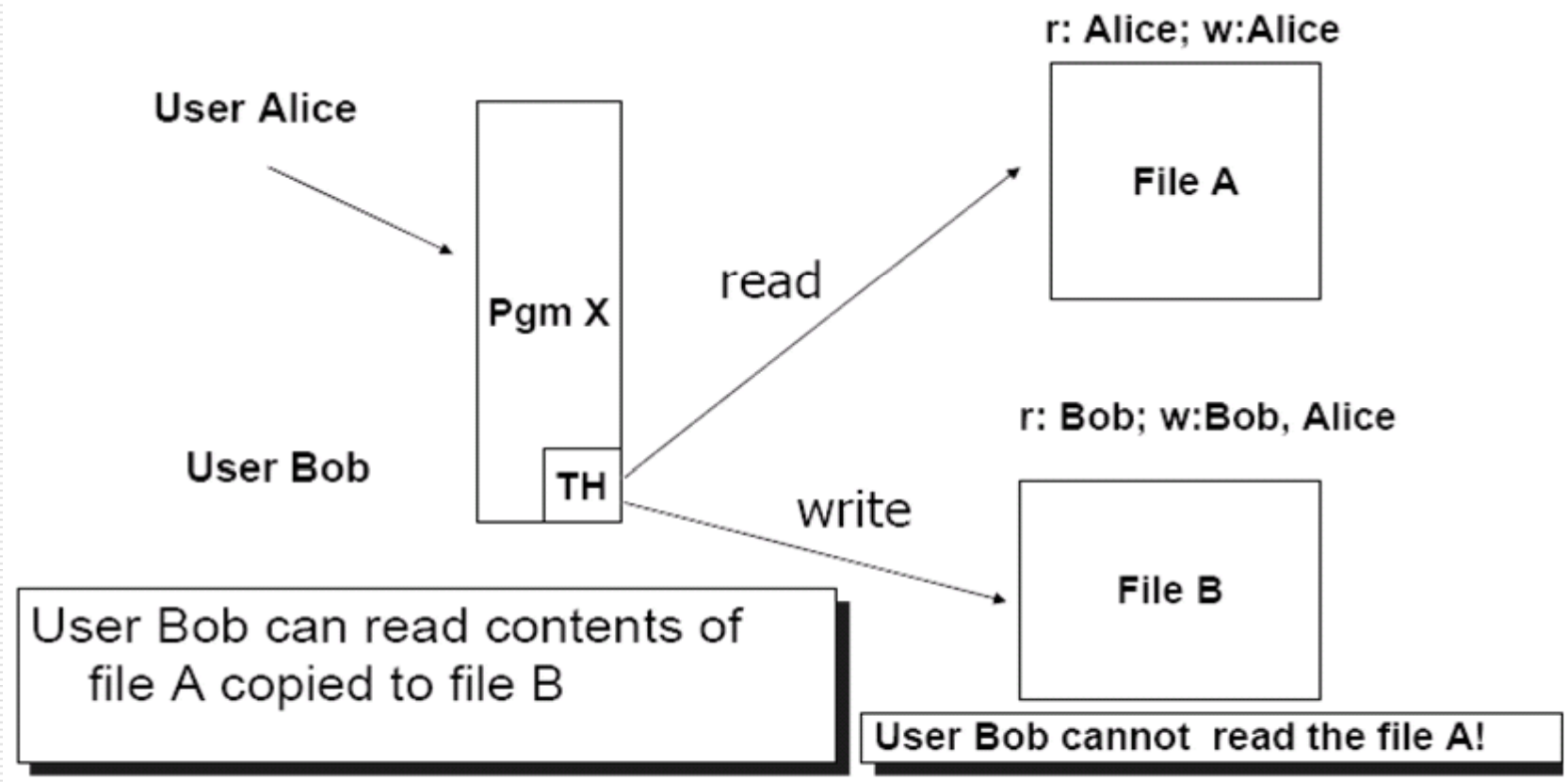
# ضعف مدل کنترل دسترسی اختیاری

- عدم امکان کنترل انتشار اطلاعات توسط عامل‌های دیگر
  - علی صاحب فایل A، اجازه خواندن را به حسن می دهد.
  - حسن فایل A را می خواند و در فایل B می نویسد.
  - علی دیگر هیچ کنترلی روی B (حاوی اطلاعات A) ندارد.
- عدم امکان کنترل جریان اطلاعات از یک شیئ به شیئ دیگر
- با فرض معتمد بودن عامل‌ها
  - به نرم‌افزارها نمی‌توان اعتماد کرد.
  - احتمال وجود اسب تروا (Trojan Horse)



# ضعف کنترل دسترسی اختیاری

- احتمال وجود اسب تروا (Trojan Horse)



# کنترل دسترسی اجباری

- کنترل دسترسی عامل‌ها به اشیاء بر اساس سطوح امنیتی آنها و قواعد ثابت
- مدل‌های حفظ محرمانگی
  - مثال: مدل BLP
- مدل‌های حفظ صحت
  - مثال: مدل Biba
- مدل‌های حفظ صحت و محرمانگی
  - مثال: مدل Dion

# فهرست مطالب

- مقدمه
- مدل‌های کنترل دسترسی اختیاری
- **مدل‌های کنترل دسترسی اجباری**
  - **مدل حفظ محرمانگی**
  - مدل حفظ صحت
- مدل‌های کنترل دسترسی نقش-مبنا

# مدل BLP

- ارائه شده به وسیلهٔ Bell و Lapadula در سال ۱۹۷۶
- توسعه‌یافتهٔ مدل ماتریس دسترسی برای حفظ امنیت چند سطحی
- مناسب برای محیط های نظامی
- عامل‌ها و اشیاء دارای سطح امنیتی (سطح محرمانگی)
- هر سطح امنیتی با دو جزء مشخص می‌گردد L=(C, S):
  - C: رده (classification): یک ترتیب از برچسب‌های امنیتی است مانند عادی، محرمانه، خیلی محرمانه، سری، بکلی سری
  - S: رسته (category): معرف سازمان یا کاربرد است (مانند ناتو، هسته‌ای).

# مدل BLP

- مبتنی بر مفهوم عامل / شئ است:
  - عاملها، اجزای فعال بوده و عملیات را اجرا می‌کنند.
  - اشیاء، اجزای غیرفعال بوده و دارای اطلاعات هستند.
- حالات دسترسی:
  - فقط خواندن (یا خواندن) = R
  - الحاق (نوشتن بدون خواندن) = A
  - اجرا (اجرای یک شئ یا برنامه) = E
  - نوشتن – خواندن (یا نوشتن) = W

| Observe / Modify | + | - |
|---|---|---|
| + | W | A |
| - | R | E |

# مدل BLP

- سطوح رده‌بندی عامل‌ها (یا برنامه‌ها) و اشیاء (یا منابع):
  - خیلی سری یا TS (مخفف Top Secret)
  - سری یا S (مخفف Secret)
  - محرمانه یا C (مخفف Confidential)
  - بدون رده یا U (مخفف Unclassified)
- **رابطه تفوق (dominance)**: مقایسه دو سطح امنیتی
  $L1 = (C1, S1)$ و $L2 = (C2, S2)$

$$L1 \geq L2 \Leftrightarrow C1 \geq C2 \wedge S1 \supseteq S2$$

- سطوح امنیتی بر اساس رابطه ترتیب جزئی تفوق (≥) تشکیل یک شبکه (Lattice) می‌دهند.

# مدل BLP

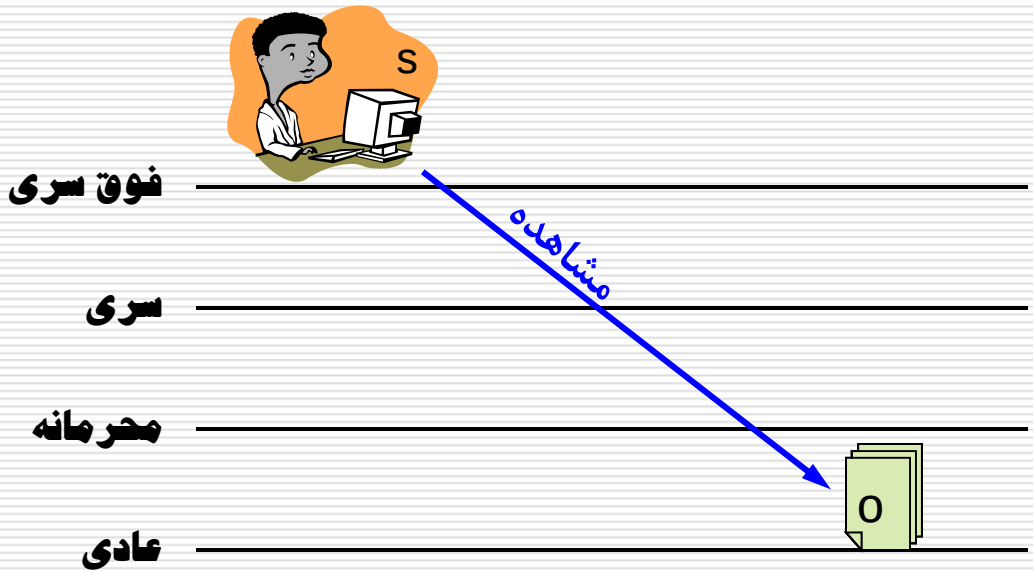
- هر شیئ دارای یک سطح امنیتی است که با تابع $f_o$ مشخص میشود.
  - مثال: $f_o(doc_1)=(secret,nato)$
- هر عامل یک سطح امنیتی اصلی ($f_s$) دارد و یک سطح امنیتی جاری ($f_c$) که هنگام ورود به سیستم مشخص می‌شود و پایینتر از سطح امنیتی اصلی است.

# قواعد مدل BLP –۱

**(۱) قاعدهٔ سادهٔ امنیتی (SS):**

یک عامل، مجوز مشاهده (خواندن R یا نوشتن W) یک شئ را دارد فقط اگر سطح امنیتی اصلی عامل، بزرگتر یا مساوی سطح امنیتی شیئ باشد.

$$f_s(s) \geq f_o(o)$$



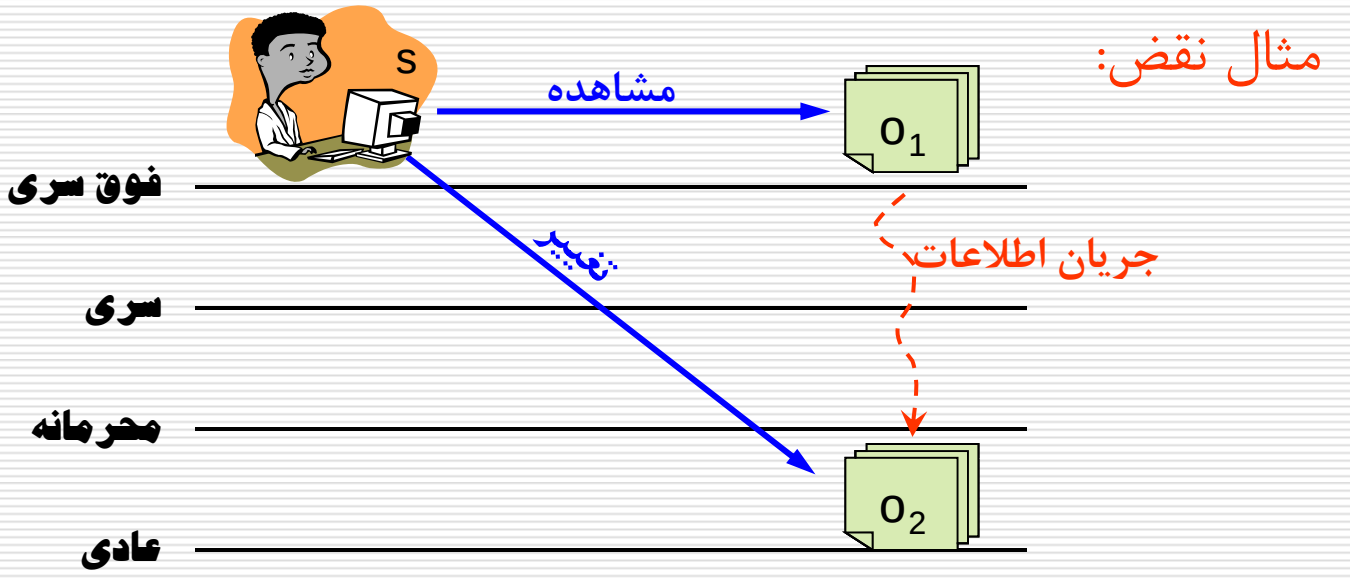
# قواعد مدل BLP – ۲

**(۲) قاعدهٔ ستاره (*):**

یک عامل، اگر امکان مشاهده شیئ $o_1$ و تغییر شیئ $o_2$ را داشته باشد، آنگاه باید:

$$f_o(o_2) \geq f_o(o_1)$$

مثال نقض:



# ۳– BLP قواعد مدل

**(۲) ادامهٔ قاعدهٔ ستاره (*):**

- یک عامل، مجوز الحاق (A) یک شئ را دارد فقط اگر سطح امنیتی فعلی عامل، کوچکتر یا مساوی سطح امنیتی شئ باشد. $f_c(s) \leq f_o(o)$
- یک عامل، مجوز نوشتن (W) یک شئ را دارد فقط اگر سطح امنیتی فعلی عامل، مساوی سطح امنیتی شئ باشد. $f_c(s) = f_o(o)$
- یک عامل، مجوز خواندن (R) یک شئ را دارد فقط اگر سطح امنیتی فعلی عامل، بزرگتر یا مساوی سطح امنیتی شئ باشد. $f_c(s) \geq f_o(o)$

# خلاصه ای از مدل BLP

*به طور خلاصه و ساده:*

- عامل از اشیای با سطوح امنیتی پایینتر یا مساوی خود، می‌خواند (مشاهده).

No Read Up

- عامل در اشیای با سطوح امنیتی بالاتر یا مساوی خود، می‌نویسد (تغییر).

No Write Down

# فهرست مطالب

- مقدمه
- مدل‌های کنترل دسترسی اختیاری
- **مدل‌های کنترل دسترسی اجباری**
  - مدل حفظ محرمانگی
  - **مدل حفظ صحت**
- مدل‌های کنترل دسترسی نقش-مبنا

# مدل حفظ صحت Biba

- مدل BLP: صرفاً حفظ محرمانگی
- نیاز به حفظ صحت داده‌ها در مقابل تغییرات غیرمجاز
- مدل ارائه شده توسط Biba در سال ۱۹۷۷
- مبانی مدل Biba مشابه مدل BLP است.
- مدل Biba بیشتر با خط‌مشی سختگیرانه اش شناخته شده است.

# مدل Biba

- عامل ها و اشیاء دارای سطح صحت
- هر سطح صحت با دو جزء مشخص می‌گردد L=(C, S):
  - C: رده (classification)
  - S: رسته (category)
  - رسته، معرف سازمان یا کاربرد است (مانند ناتو، هسته‌ای)
- سطح صحت عامل: میزان اعتماد به فرد (عامل) در عدم تغییر ناصحیح داده‌های یک شیئ.
- سطح صحت شیئ: میزان اعتماد به داده‌های یک شیئ و میزان خسارت ناشی از تغییر غیرمجاز در داده‌های آن.

# مدل Biba

- سطوح رده‌بندی صحت عاملها (یا برنامه‌ها) و اشیاء (یا منابع):
  - حیاتی یا C (مخفف Crucial)
  - خیلی مهم یا VI (مخفف Very Important)
  - مهم یا I (مخفف Important)
- **رابطه تفوق (dominance)**: مقایسه دو سطح صحت
  $L1 = (C1, S1)$ و $L2 = (C2, S2)$

$$L1 \geq L2 \Leftrightarrow C1 \geq C2 \wedge S1 \supseteq S2$$

- سطوح صحت بر اساس رابطه ترتیب جزئی تفوق ($\geq$) تشکیل یک شبکه (Lattice) می دهند.

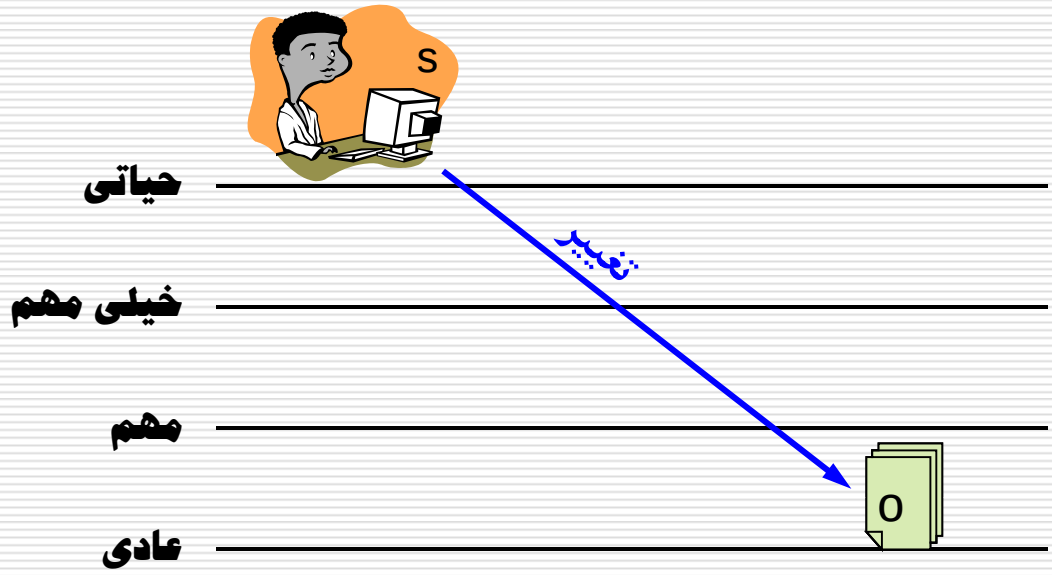
# مدل Biba

- مبتنی بر مفهوم عامل / شیئ است:
  - عاملها، اجزای فعال بوده و عملیات را اجرا می‌کنند.
  - اشیاء، اجزای غیرفعال بوده و دارای اطلاعات هستند.
- حالات دسترسی:
  - تغییر (Modify): نوشتن داده‌ها در یک شیئ
  - اجرا (Invoke): اجرای یک عامل با عامل دیگر
  - مشاهده (Observe): خواندن داده‌های یک شیئ

# قواعد اجباری در مدل Biba –۱

## (۱) خصوصیت ستاره (*) صحت

یک عامل، مجوز تغییر یک شئ را دارد فقط اگر سطح صحت عامل، بزرگتر یا مساوی سطح صحت شئ باشد. $I(s) \geq I(o)$

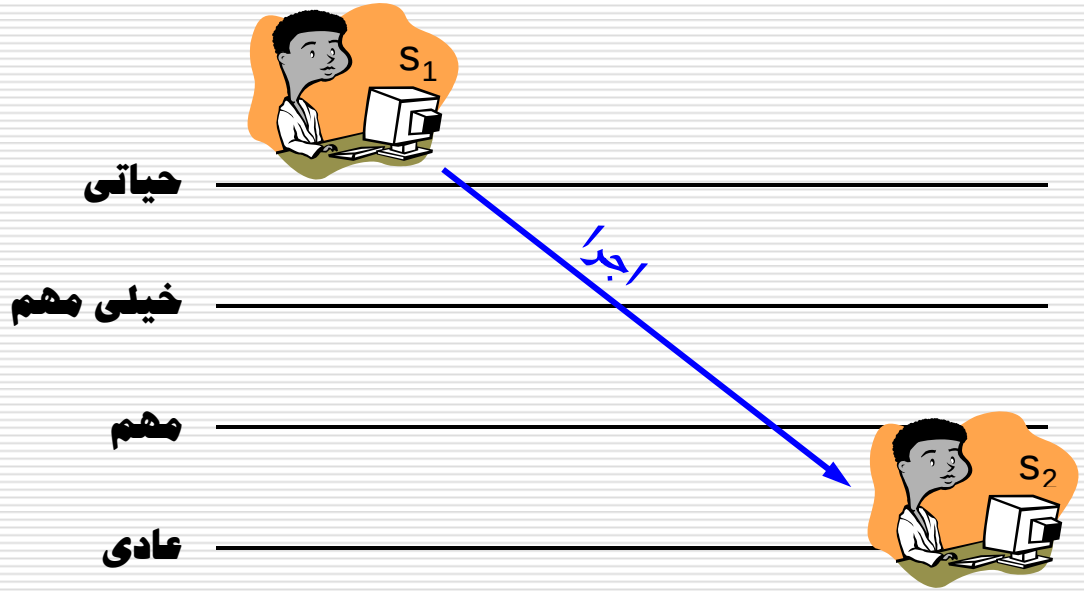


# قواعد اجباری در مدل Biba – ۲

## (۲) خصوصیت اجرا

یک عامل، مجوز اجرا یک عامل دیگر را دارد فقط اگر سطح صحت عامل اول بزرگتر یا مساوی سطح صحت عامل دوم باشد.

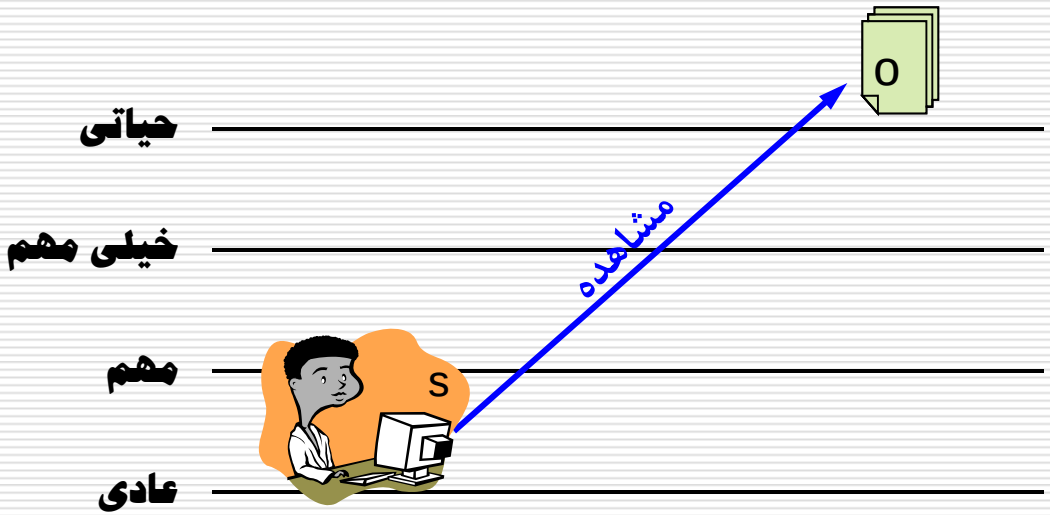
$$I(s_1) \geq I(s_2)$$



# قواعد اجباری در مدل Biba – ۳

## (۳) خصوصیت ساده صحت

یک عامل، مجوز مشاهده یک شیئ را دارد فقط اگر سطح صحت عامل، کوچکتر از سطح صحت شیئ باشد. $I(o) \geq I(s)$



# خلاصه مدل Biba

- خط مشی حفظ صحت سختگیرانه به طور خلاصه:
  - عامل از اشیای با سطوح امنیتی بالاتر یا مساوی خود، می‌خواند (مشاهده).

No Read Down

  - عامل در اشیای با سطوح امنیتی پایینتر یا مساوی خود، می‌نویسد (تغییر).

No Write Up

# فهرست مطالب

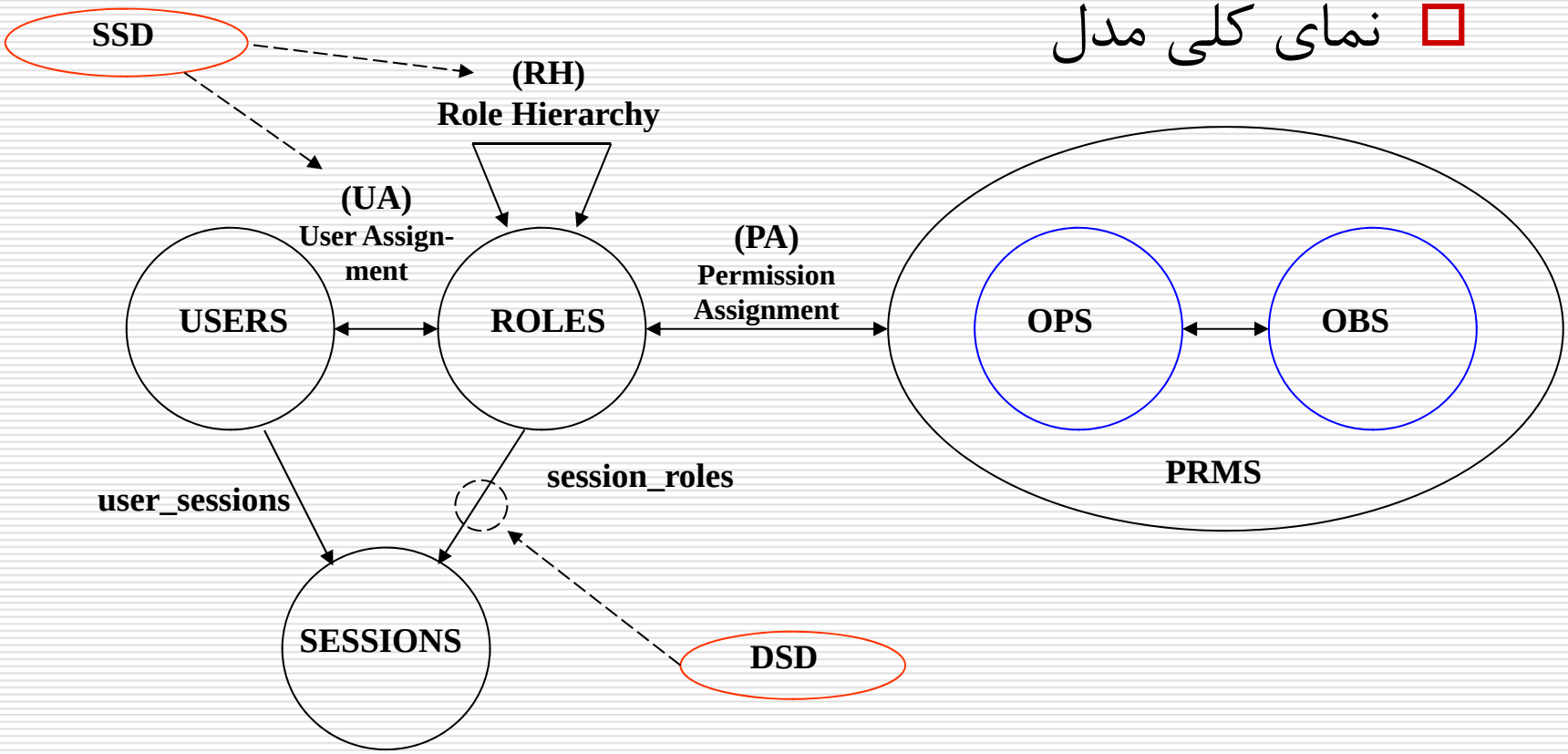
- مقدمه
- مدل‌های کنترل دسترسی اختیاری
- مدل‌های کنترل دسترسی اجباری
  - مدل حفظ محرمانگی
  - مدل حفظ صحت
- **مدل‌های کنترل دسترسی نقش-مبنا**

# مدل نقش–مبنا

- سازگاری با ساختار سازمانی
- سادگی مدیریت کنترل دسترسی
- قدرت بیان: امکان بیان خط‌مشی‌های اختیاری (DAC) و اجباری (MAC)
- اصل حداقل مجوزها (least privilege)
- تفکیک وظایف (SoD)

# مدل RBAC

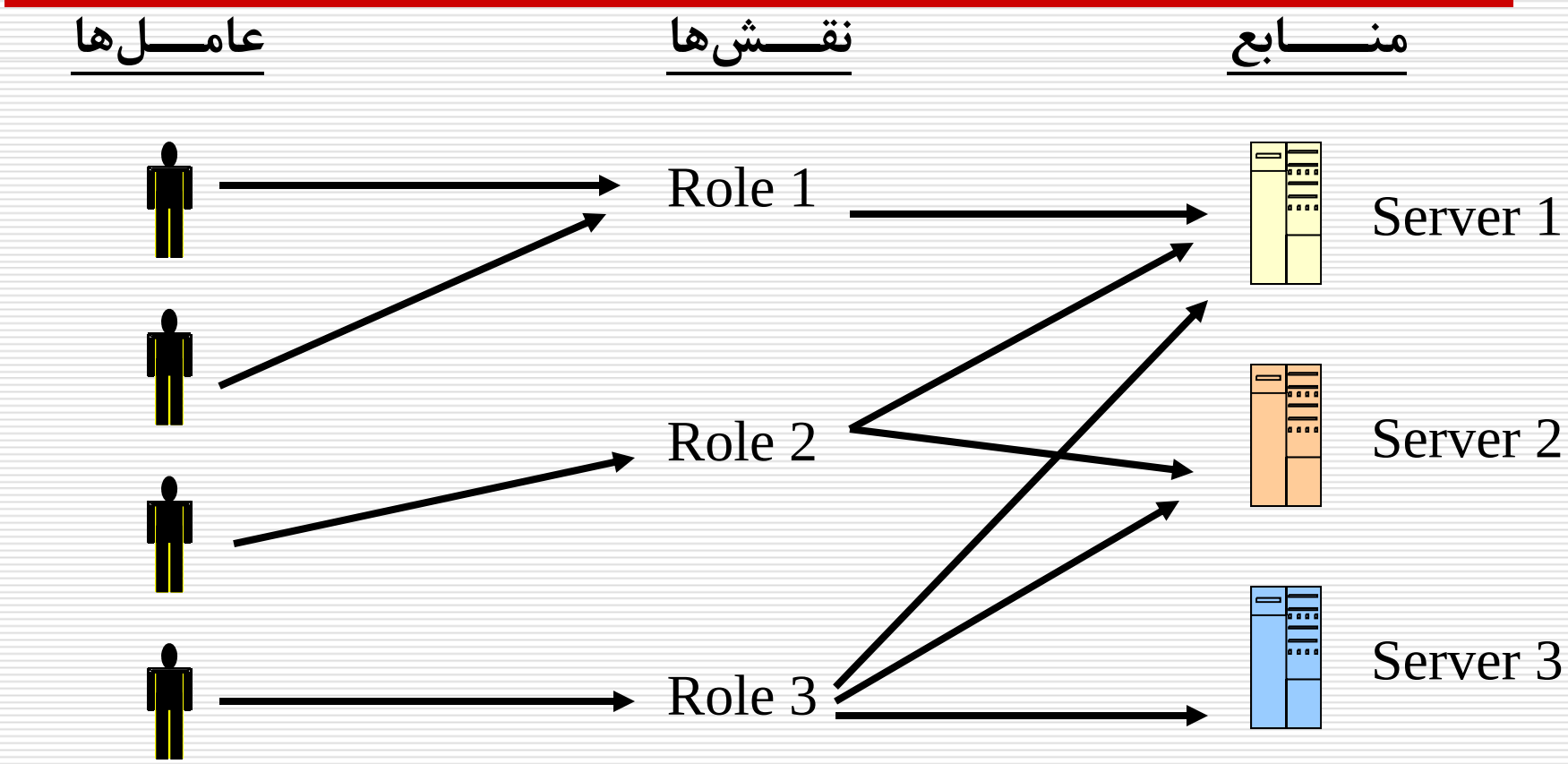
- نمای کلی مدل



# مدل نقش–مبنا

- اعطای مجوزها به نقش‌ها و نقش‌ها به کاربران (به جای اختصاص مستقیم مجوزها به کاربران)
- تعیین نقش‌ها بر اساس اصل حداقل مجوزها
- اعطای مجموعه مجوزهای موردنیاز برای اجرای وظایف مربوطه به هر نقش به آن
- امکان توصیف تفکیک وظایف (Separation of Duties) و مساله تضاد منافع (Conflict of Interests)

# مدل نقش–مبنا




کاربران دائماً تغییر می‌کنند اما نقش‌ها خیر

# انواع مدل نقش–مبنا

- مدل نقش-مبنای پایه ($RBAC_0$)
  - مولفه‌های مدل پایه
- مدل نقش-مبنای سلسله مراتبی ($RBAC_1$)
  - مدل $RBAC_0$ + سلسله مراتب نقشها
- مدل نقش-مبنا با محدودیت ($RBAC_2$)
  - مدل $RBAC_0$ + تفکیک وظایف ایستا (SSoD) و پویا (DSoD)
- مدل نقش-مبنای کامل ($RBAC_3$)
  - مدل $RBAC_1$ + مدل $RBAC_2$

# انواع مدل نقش–مبنا

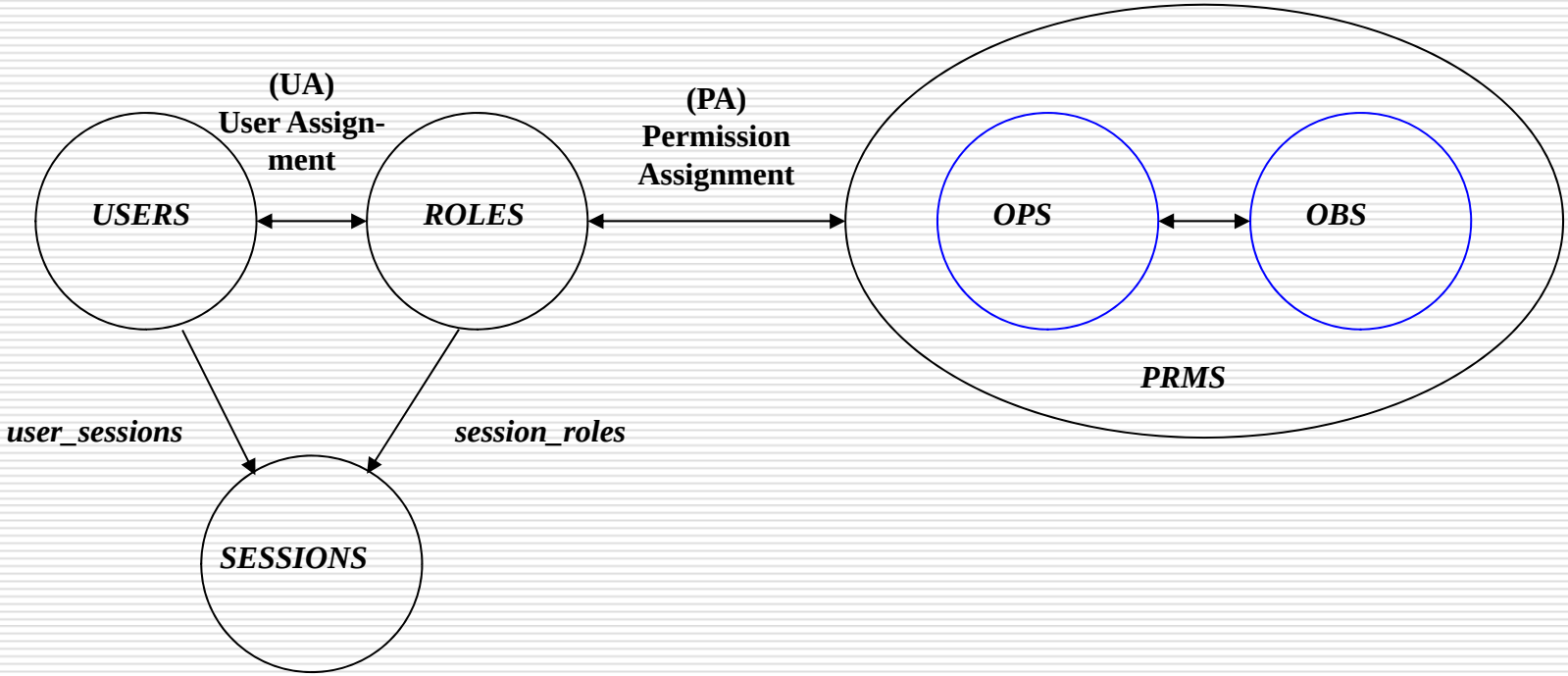
| Models | Hierarchies | Constraints |
|---|---|---|
| $RBAC_0$ | – | – |
| $RBAC_1$ | ✓ | – |
| $RBAC_2$ | – | ✓ |
| $RBAC_3$ | ✓ | ✓ |

# مدل نقش مبنای پایه $RBAC_0$

- مولفه‌های مدل پایه $RBAC_0$ :
  - عامل‌ها یا کاربران(USERS)
  - نقش‌ها (ROLES)
  - مجوزها (PRMS)
    - اعمال (OPS)
    - اشیاء (OBS)
  - رابطه اختصاص نقش به کاربر (UA)
  - رابطه اختصاص مجوز به نقش (PA)
  - نشست‌ها (SESSIONS)

# مدل $RBAC_0$

- نمای کلی مدل $RBAC_0$



# مدل $RBAC_0$

- کاربران (USERS)

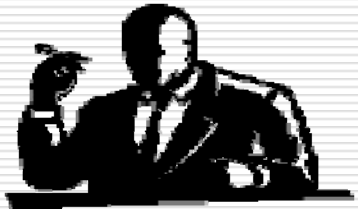
پردازه

عامل انسانی

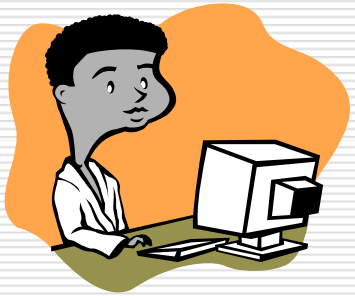
عامل ماشینی هوشمند

# مدل $RBAC_0$

- نقش‌ها (ROLES): هر نقش شامل تعدادی وظیفه‌مندی

مدیر کل



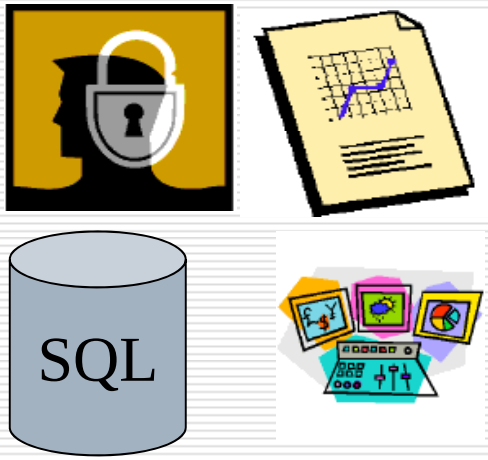
برنامه‌نویس

اپراتور راهنما

# مدل $RBAC_0$

- اعمال (OPS): اجرای عملی خاص بر روی یک شیئ یا منبع

- Database – Update Insert Append Delete
- Locks – Open Close
- Reports – Create View Print
- Applications – Read Write Execute



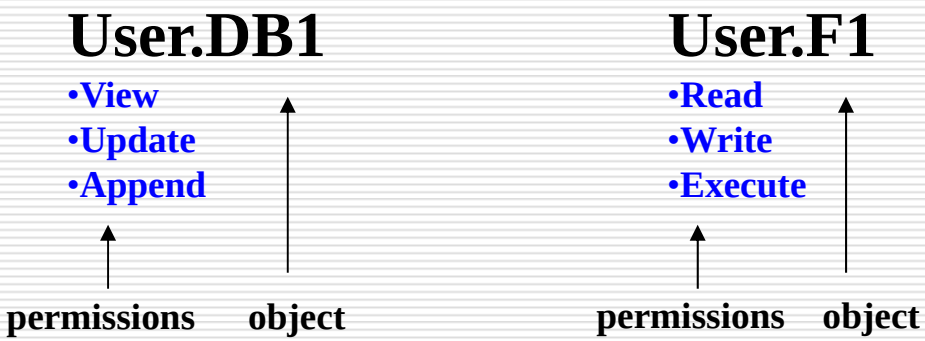
# مدل $RBAC_0$

- اشیاء یا منابع (OBS): حاوی داده‌ها

•OS Files or Directories
•DB Columns, Rows, Tables, or Views
•Printer
•Disk Space
•Lock Mechanisms

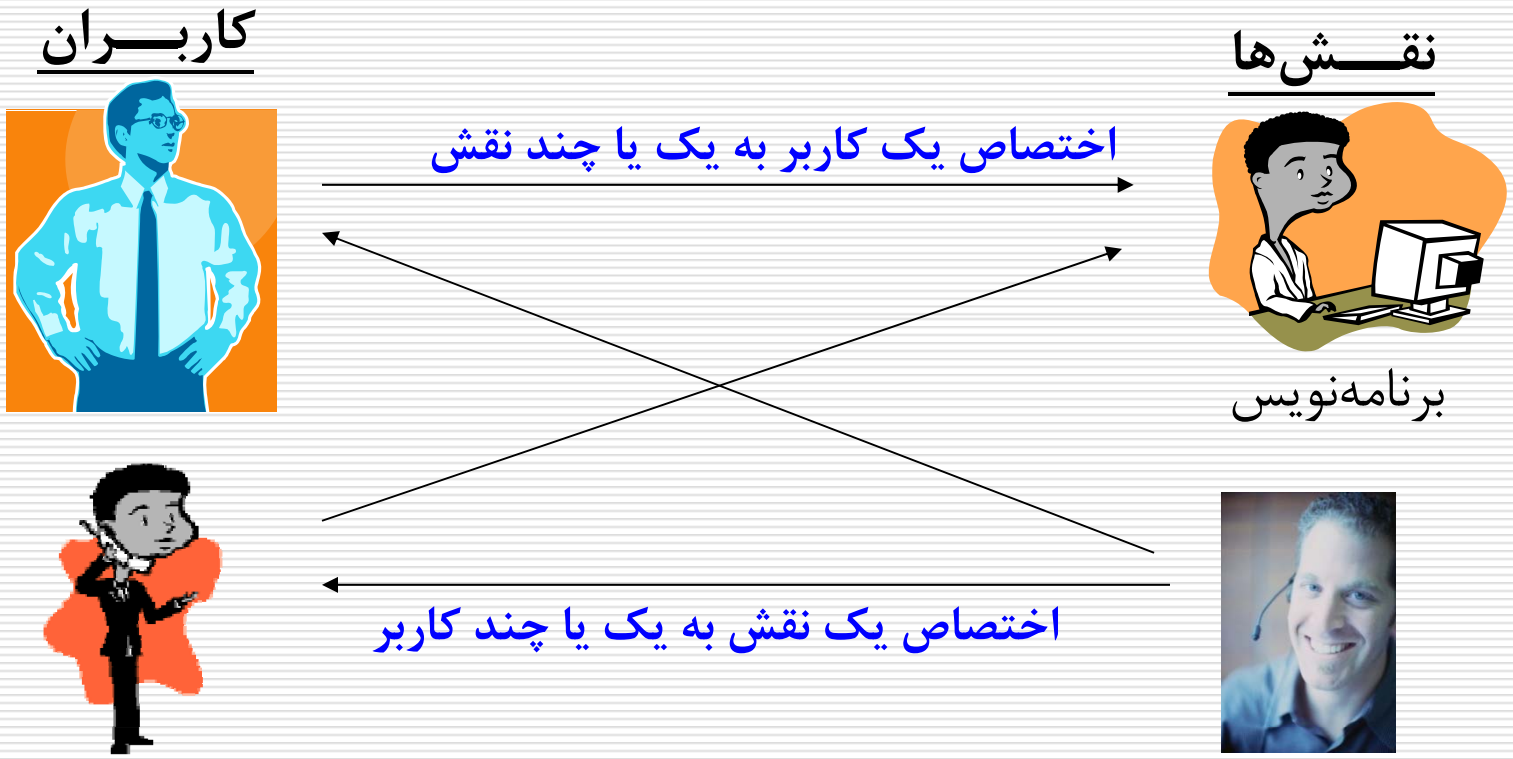
# مدل $RBAC_0$

- مجوزها (PRMS): اجرای عملی خاص بر روی یک شیئ یا منبع



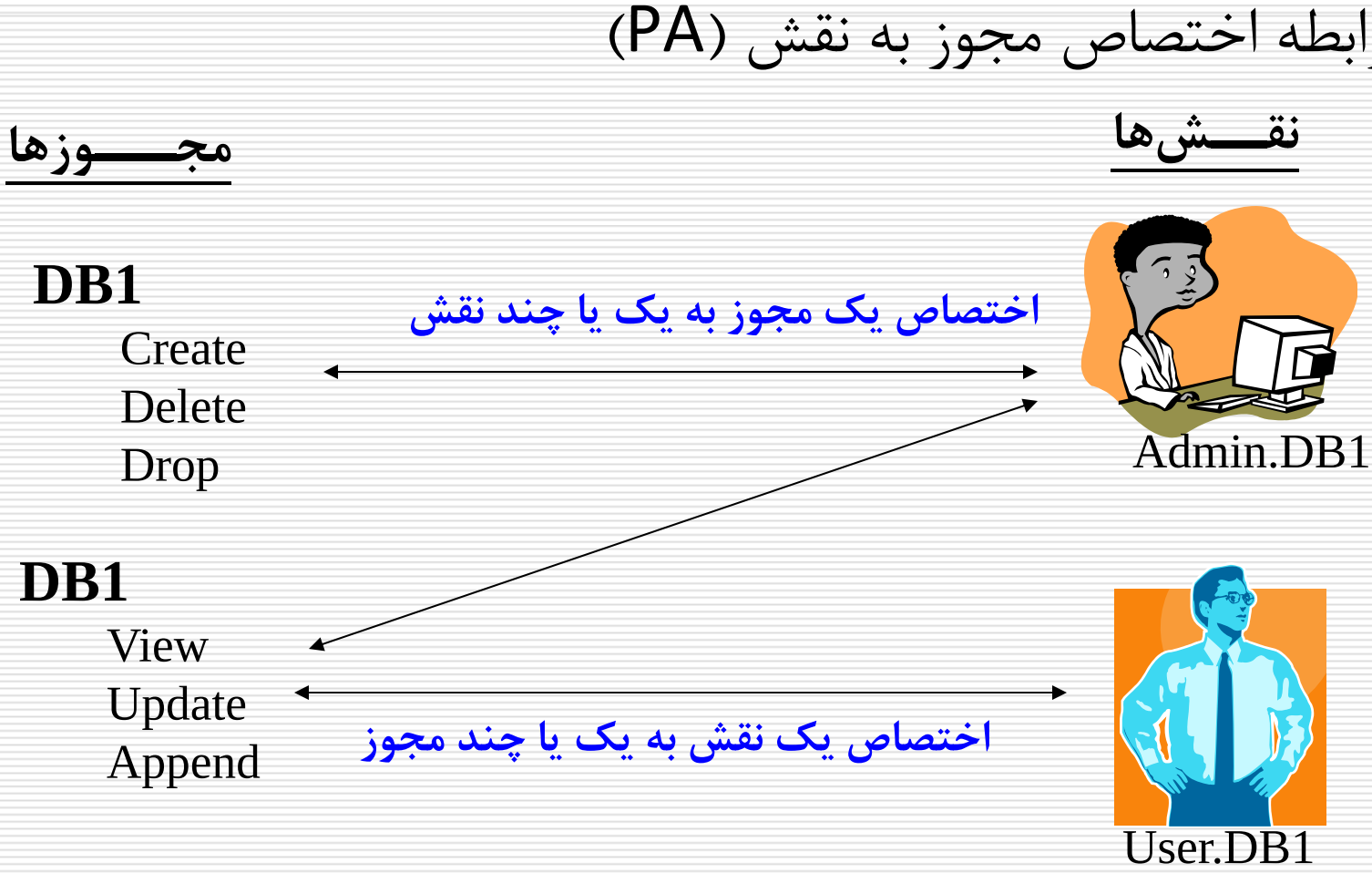
# مدل $RBAC_0$

- رابطه اختصاص نقش به کاربر (UA)



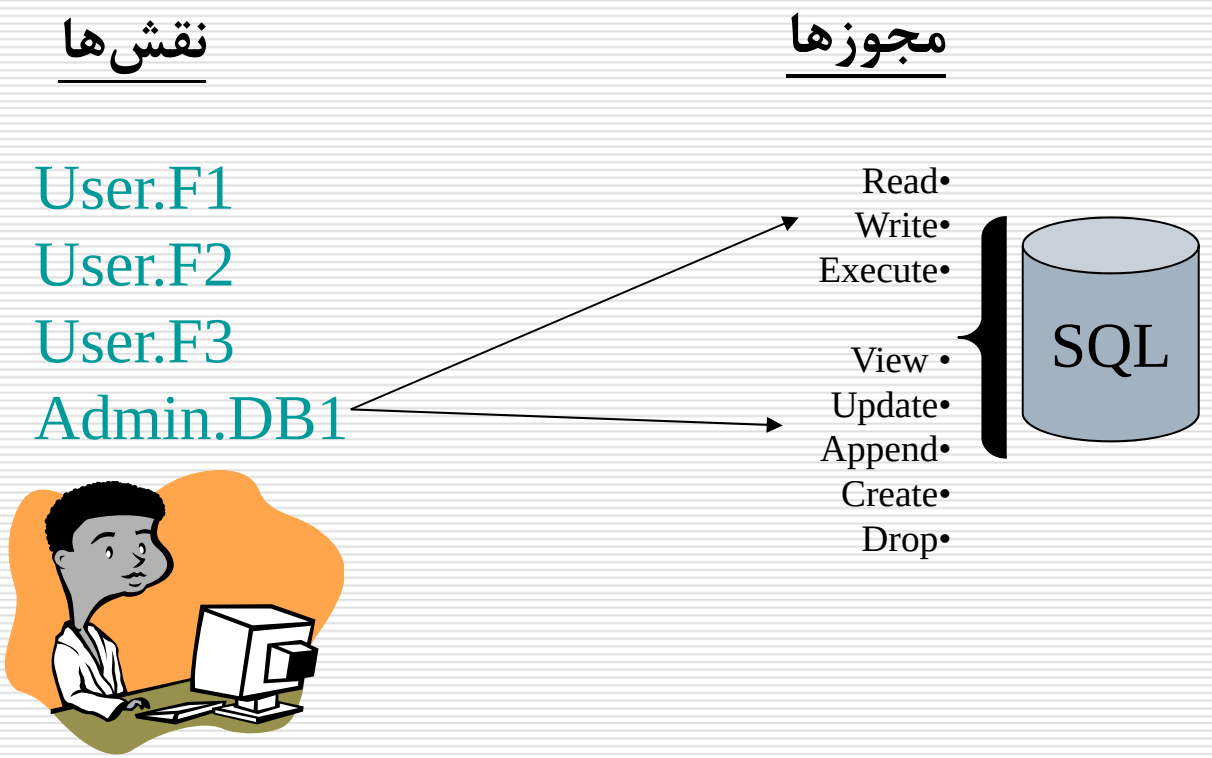
# مدل $RBAC_0$

- رابطه اختصاص مجوز به نقش (PA)



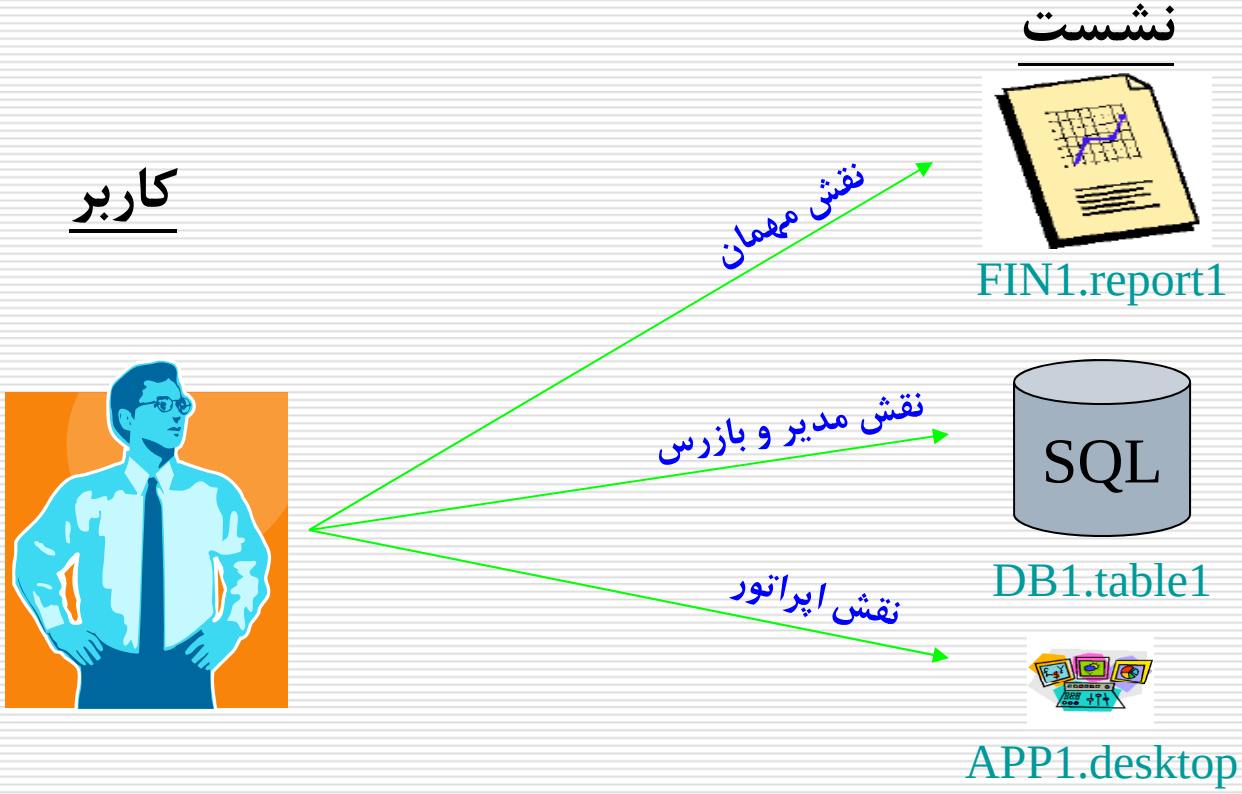
# مدل $RBAC_0$

- رابطه اختصاص مجوز به نقش (PA)



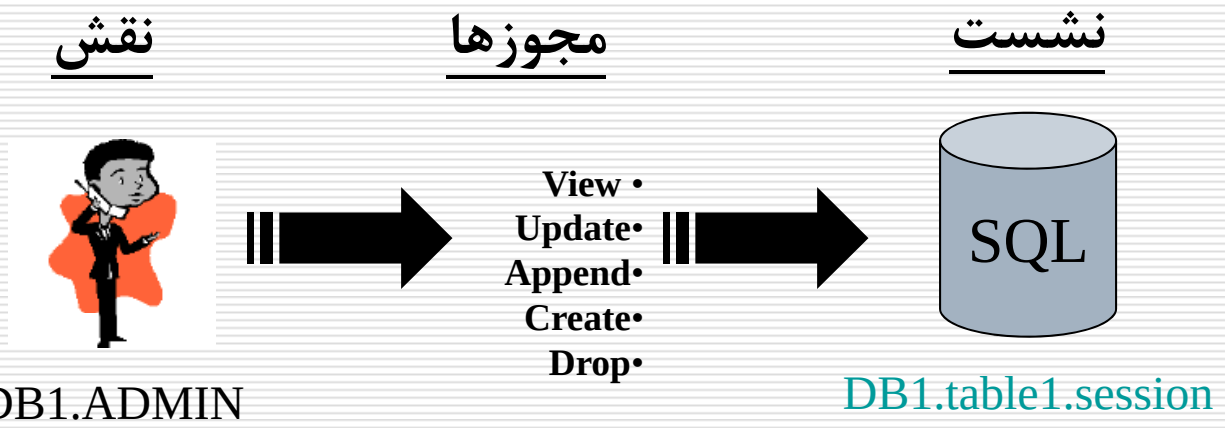
# مدل $RBAC_0$

- نشست‌ها: هر کاربر می تواند چند نشست داشته باشد و در هر نشست چند نقش اختصاص یافته (با UA) را فعال کند.



# مدل $RBAC_0$

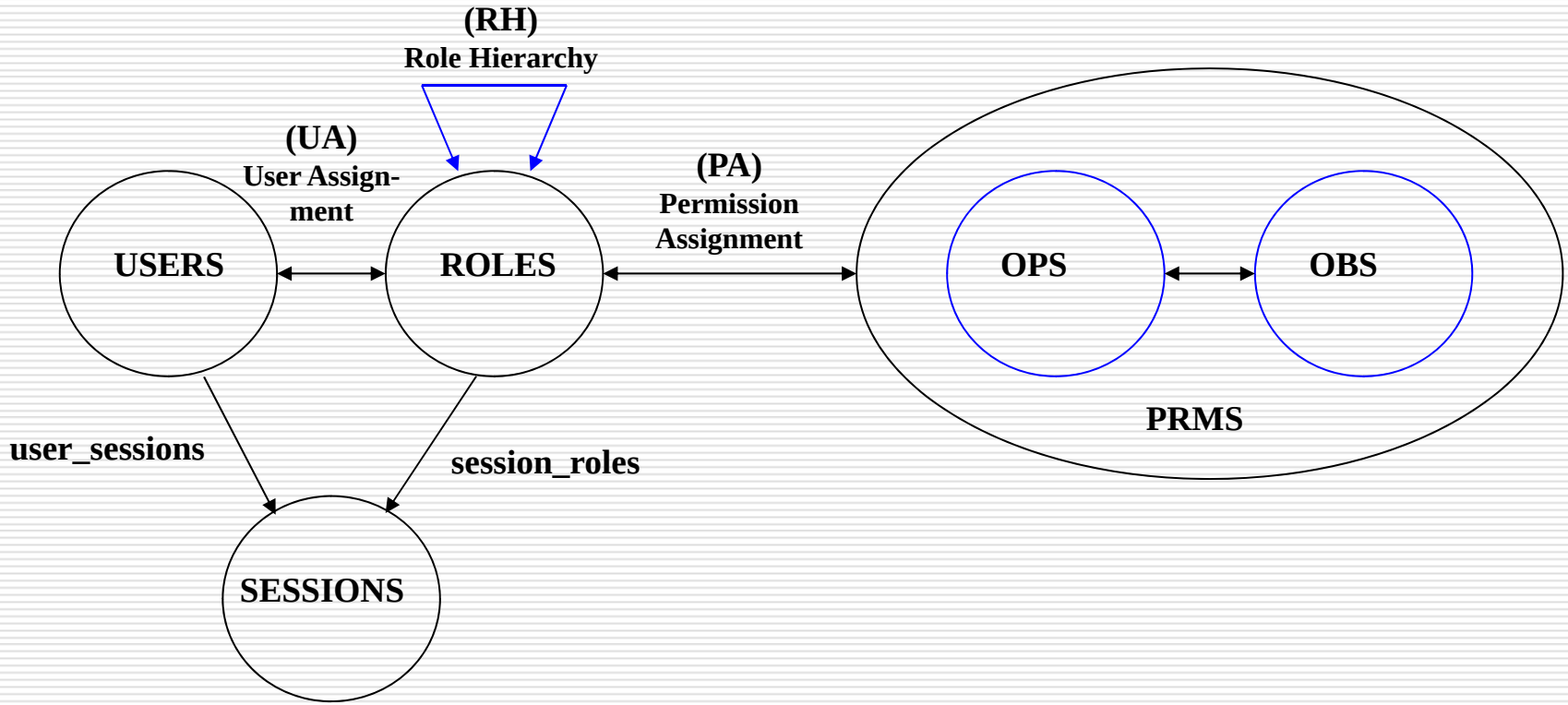
- مجوزهای یک نشست = مجموعه مجوزهای نقش‌های فعال شده در نشست



# مدل نقش–مبنای سلسله‌مراتبی $RBAC_1$

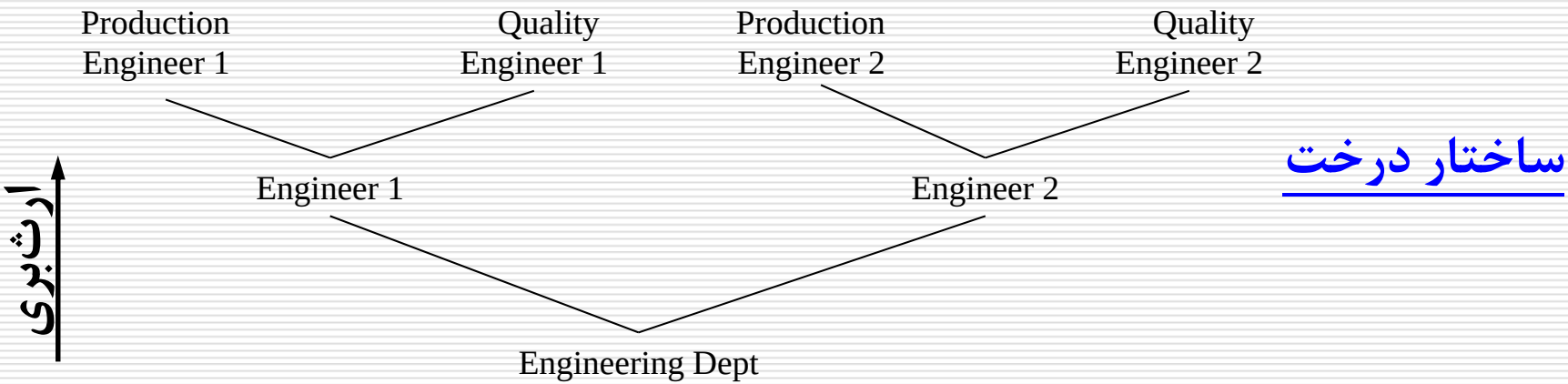
# مدل نقش–مبنای سلسله‌مراتبی $RBAC_1$

- نمای کلی مدل $RBAC_1$

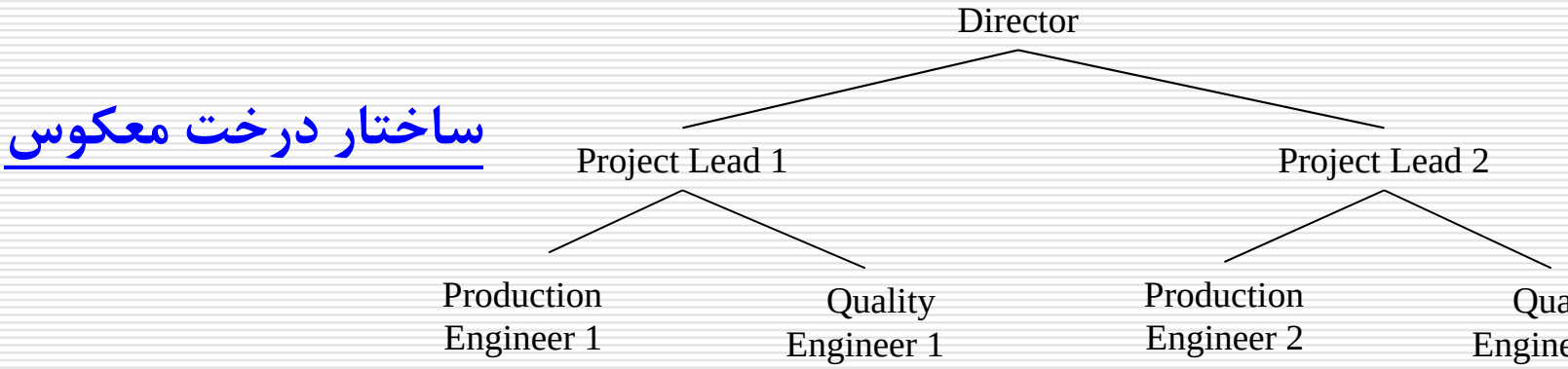


# مدل $RBAC_1$

- انواع سلسله مراتب نقش‌ها




ساختار درخت

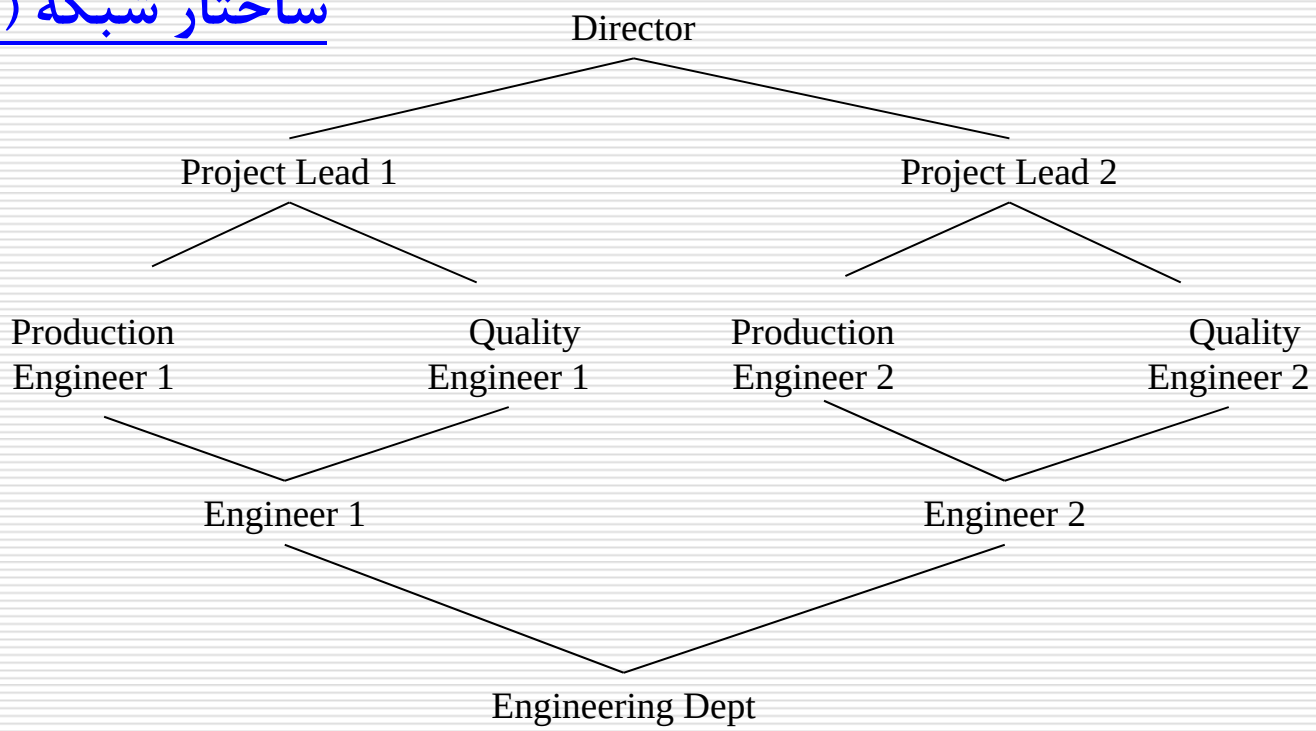



ساختار درخت معکوس

# مدل $RBAC_1$

- انواع سلسله مراتب نقش‌ها

**ساختار شبکه (Lattice)**



# مدل $RBAC_1$

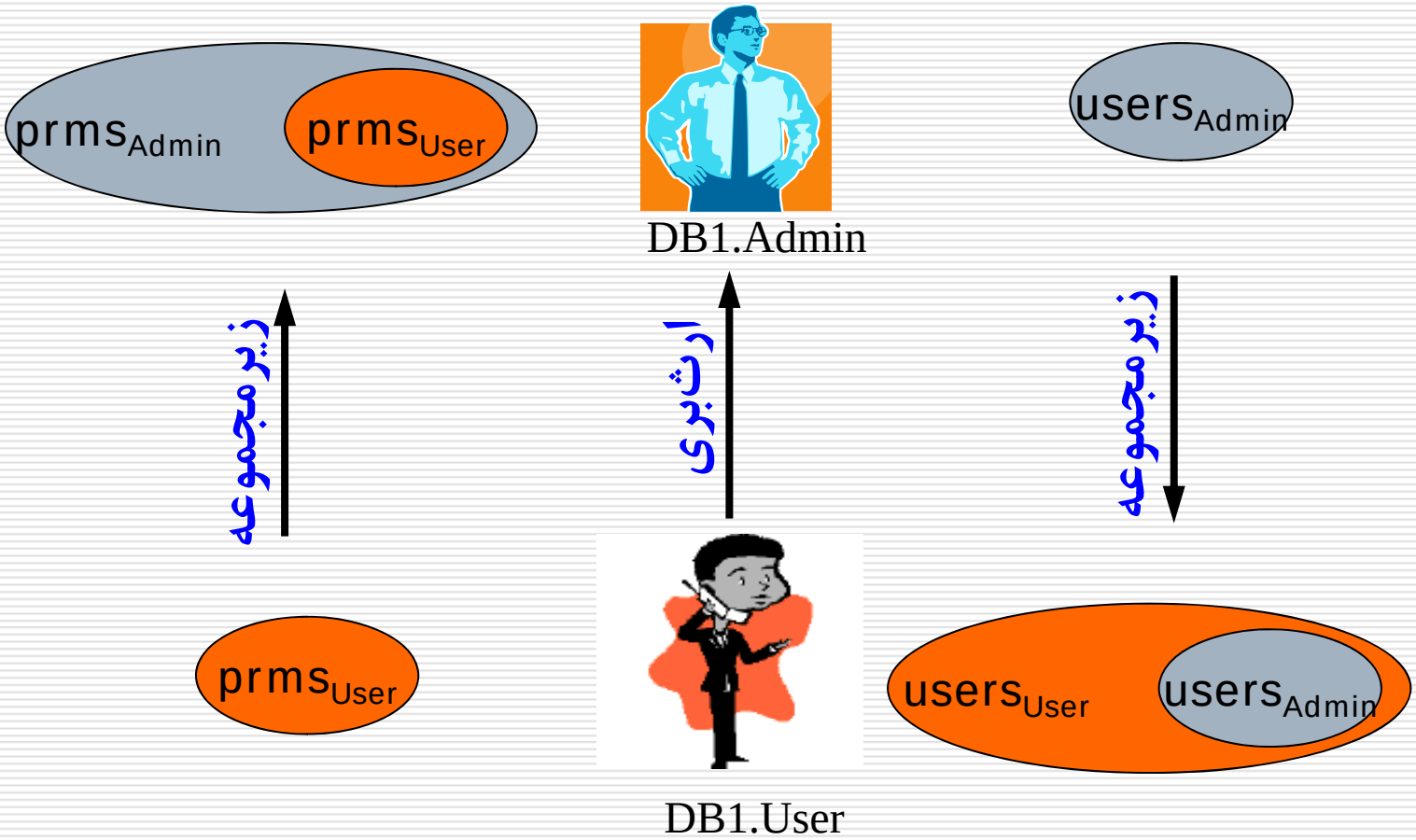
- رابطه نقش و زیرنقش (RH)
- ساختار سلسله مراتبی نقش‌ها می‌تواندبرگرفته از ساختار سازمانی باشد.
- در صورتیکه نقش $r_1$ فرزند نقش $r_2$ در سلسله مراتب باشد، همه مجوزهای آن را به ارث می‌برد.

$r_1$ به ارث می‌برد از $r_2$

| Admin | Guest |
|---|---|
| r-w-x | -r- |

# مدل $RBAC_1$

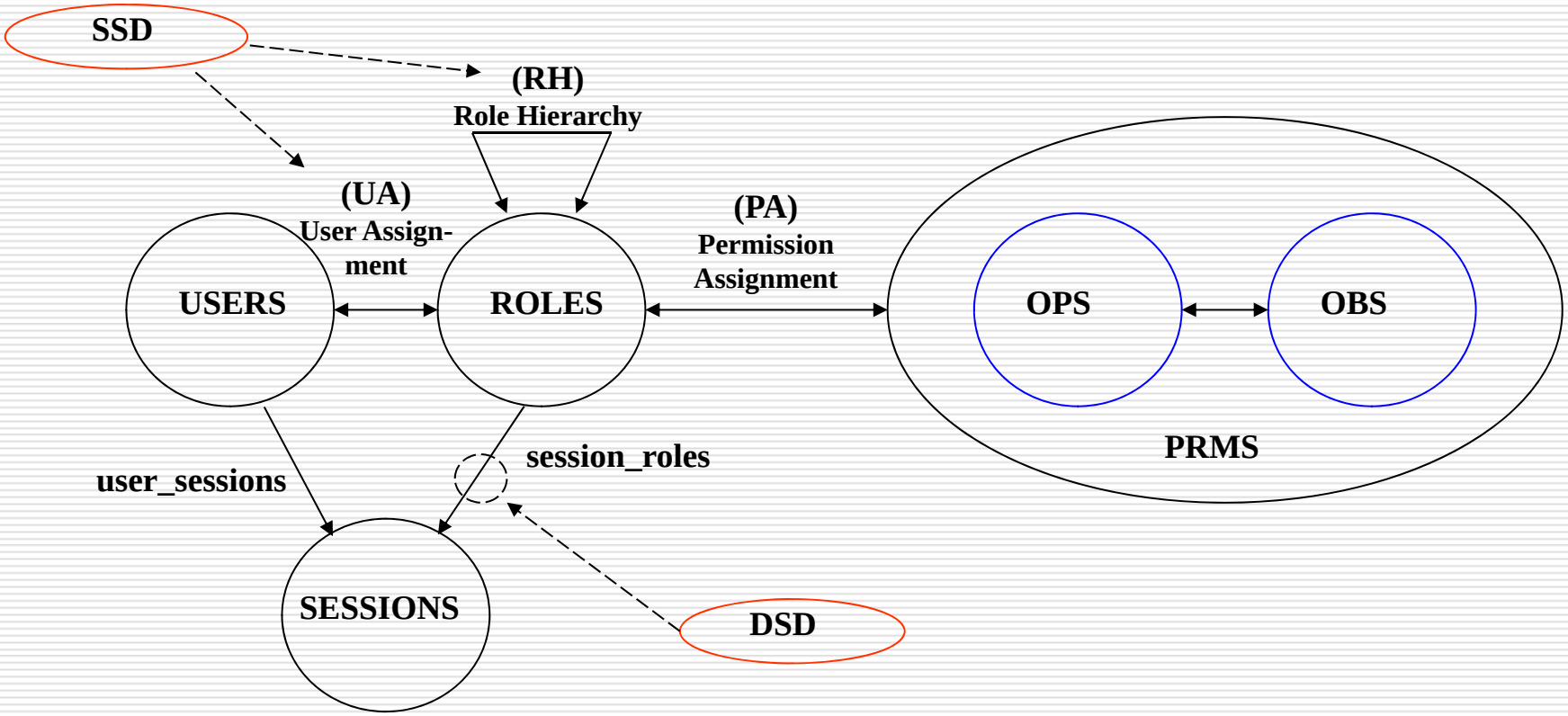
- رابطه کاربران و مجوزها در ارث‌بری نقش‌ها



# مدل نقش–مبنا با محدویت $RBAC_2$

# مدل نقش–مبنا با محدویت $RBAC_2$

- نمای کلی مدل $RBAC_2$



# مدل $RBAC_2$

## محدودیت تفکیک وظایف (SoD)

- برای جلوگیری از دستیابی کاربر به مجوزهای بیش از حد مجاز و انجام امور غیرمجاز
- دو نوع تفکیک وظایف:
  - تفکیک وظایف ایستا (SSoD)
  - تفکیک وظایف پویا (DSoD)

# مدل $RBAC_2$

SSoD: اِعمال محدودیت در اختصاص نقش به کاربر در رابطه UA

- از یک مجموعه از نقش‌های متداخل، نمی‌توان n نقش و یا بیشتر را به یک کاربر اِعطا کرد.
- مثال: در یک بانک یک فرد نمی‌تواند هر دو نقش کارمند شعبه و بازرس را داشته باشد.
- دو نقش دو بدو ناسازگار: ممکن است یک کاربر مجاز به برخورداری از دو نقش در یک زمان نباشد.

# مدل $RBAC_2$

DSoD: اِعمال محدودیت در فعال‌سازی نقش توسط کاربر در یک نشست

- از یک مجموعه از نقش‌های متداخل، نمی‌توان n نقش و یا بیشتر را در طی یک نشست فعال کرد.
- اِعمال این محدودیت نیاز به نگهداری سابقه نقش‌های فعال شده در طی یک نشست دارد.
- مثال: در یک بانک، کسی نمی‌تواند در فرآیند صدور یک چک، هم نقش صادرکننده و هم نقش تاییدکننده چک را داشته باشد.

# مدل نقش–مبنای سلسله‌مراتبی با محدودیت $RBAC_3$

# مدل نقش–مبنای سلسله‌مراتبی با محدودیت $RBAC_3$

- ترکیب دو مدل نقش-مبنای سلسله‌مراتبی $RBAC_1$ و نقش-مبنا با محدودیت $RBAC_2$

$$RBAC_3 = RBAC_1 + RBAC_2$$

- تاثیر متقابل سلسله مراتب نقش‌ها بر محدودیت‌های تفکیک وظایف

# پایان


مرکز امنیت داده و شبکه شریف

http://dnsl.ce.sharif.edu

پست الکترونیکی

m_amini@ce.sharif.edu