



POLYTECHNIQUE
MONTREAL

UNIVERSITÉ
D'INGÉNIERIE

INF8085: Cybersécurité

Autorisation, contrôle d'accès
Nora Cuppens



Contenu du cours

- Introduction au contrôle d'accès
- Contrôle d'accès sous LINUX
- Modèles DAC et MAC
- Modèle RBAC
- Modèle ABAC
- Introduction à l'IAM



- Contrôle d'accès
 - Définition : Fonction permettant de limiter l'accès à des ressources aux individus/machines/entités qui ont le droit d'accéder à ces ressources
 - S'applique autant à des objets physiques qu'à des ressources informatiques



- 4 Aspects
 - Identification : Déterminer l'identité du demandeur
 - Authentification : Validation de l'identité du demandeur
 - Autorisation : Validation du droit d'accès au ressources
 - Audit/(« Accounting ») : Attribution d'actions à une identité
 - ➔ AAA (Authentication, Authorization and Accounting) ou IAAA
- Dans un système d'exploitation
 - Identification : nom d'utilisateur, identificateur de processus (PID)
 - Authentification : commande d'authentification
 - Autorisation: matrice d'accès, contrôleur de référence
 - Audit : journaux (« logs »)



- Contrôleur de référence (« Reference Monitor »)
 - Composant qui s'interpose entre **tous les accès** de sujets à objets
 - Vérifie chaque demande d'entrée selon une procédure stricte
 - Maintient la sécurité au niveau voulu
 - S'implémente dans la noyau du système d'exploitation (OS)
 - Sujet = Utilisateur ou processus
 - Objet = Processus ou ressource (fichier)
 - Modes d'accès = { R-Read, W-Write, X-Execute }
 - Input: requête d'accès (sujet, objet, mode d'accès)
 - Output: Réponse (oui ou non) selon que l'accès est permis ou pas



Introduction au contrôle d'accès

- Matrice d'accès

- Matrice qui liste les sujets (lignes) et objets (colonnes) dans un système, et les modes d'accès pour chaque (sujet, objet) (case de la matrice)

Sujet\Objet	File 1	File 2	Process 1	Process 2
Process 1	-	R	R,W,X	-
Process 2	-	X	R	R,W,X
User 1	R,X	W	-	R,X

- Pour un ordinateur avec A sujets et B objets, la matrice d'accès aura une taille de $A \times B$. La majorité de cellules seront vides !



- Listes de contrôle d'accès (ACL)
 - Prendre chaque colonne de la matrice d'accès pour chaque sujet non-vidé
 - Stocker la liste d'accès avec l'objet
 - Pour chaque accès à l'objet, le contrôleur de référence vérifie si le sujet a les droit requis

Sujet\Objet	File 1	File 2	Process 1	Process 2
Process 1	-	R	R,W,X	-
Process 2	-	X	R	R,W,X
User 1	R,X	W	-	R,X



- Modèle de sécurité Linux
 - Toutes les ressources sont des objets (fichier, répertoire, mémoire, IO)
 - Chaque objet a un propriétaire
 - L'administrateur peut ajouter de nouveaux utilisateurs, lire et changer tous les objets, et changer les droit d'accès de tous les objets.
 - Les utilisateurs peuvent seulement accéder aux objets pour lesquels ils ont la permission, et peuvent seulement changer les droits d'accès des objets dont ils sont propriétaires
 - Les logiciels s'exécutent avec les droits de l'utilisateur qui a lancé le programme



- Utilisateurs
 - User ID (UID) pour chaque utilisateur
 - UID 0 est réservé pour l'administrateur (root)
 - Les fichiers ont l'ID de l'utilisateur qui a créé le fichier
- Groupes
 - Group ID (GID)
 - Les utilisateurs ont un groupe principal
 - Les utilisateurs peuvent joindre d'autres groupes
 - Les fichiers ont le groupe principal de l'utilisateur qui a créé le fichier
- Utilisateurs et groupes ne sont pas des objets



Contrôle d'accès Linux

- Chaque fichier a un UID et GID assigné
- Chaque programme a un UID et GID assigné
- Avant d'exécuter un appel de fonction du système (« system call »), le contrôleur de référence vérifie :
 - Si $UID = 0$, permettre l'accès
 - Sinon, lire la liste de contrôle d'accès de l'objet et vérifier si l'accès est permis



- Permissions de fichiers
 - R (lire)
 - W (écrire/changer)
 - X (Exécuter)
- Pour
 - U (Propriétaire)
 - G (Groupe)
 - O (Autres utilisateurs)

```
-rw-r----- 1 Emilie profs 7627 Oct 1 12:50 exam  
-rw-rw-rw- 1 root root 12987 Sep 7 19:34 /etc/passwd
```



- Changement du propriétaire d'un objet
 - Commande chown
 - Exemple : chown patrick exam
 - Patrick devient le nouveau propriétaire du fichier exam
 - Seul l'administrateur root pour exécuter un chown



- Changement des droits sur un objet
 - Commande `chmod`
 - Seul le propriétaire et l'administrateur peuvent changer les droits
 - Exemple : `chmod u=rwx, g=rx, o=r myfile`
 - Commande équivalente à : `chmod 754 myfile`
 - read = 4
 - write = 2
 - execute = 1
 - pas de permission = 0
 - Ajout de droit : `chmod g+w myfile`
 - Retrait de droit : `chmod o-r myfile`



- Sticky bit
 - Pas d'effaçage du fichier
- setuid
 - Le programme s'exécute avec les permissions du propriétaire
 - `chmod +s programme`
- setgid
 - Le programme s'exécute avec les permissions d'un utilisateur dans le groupe du propriétaire



Contrôle d'accès Linux

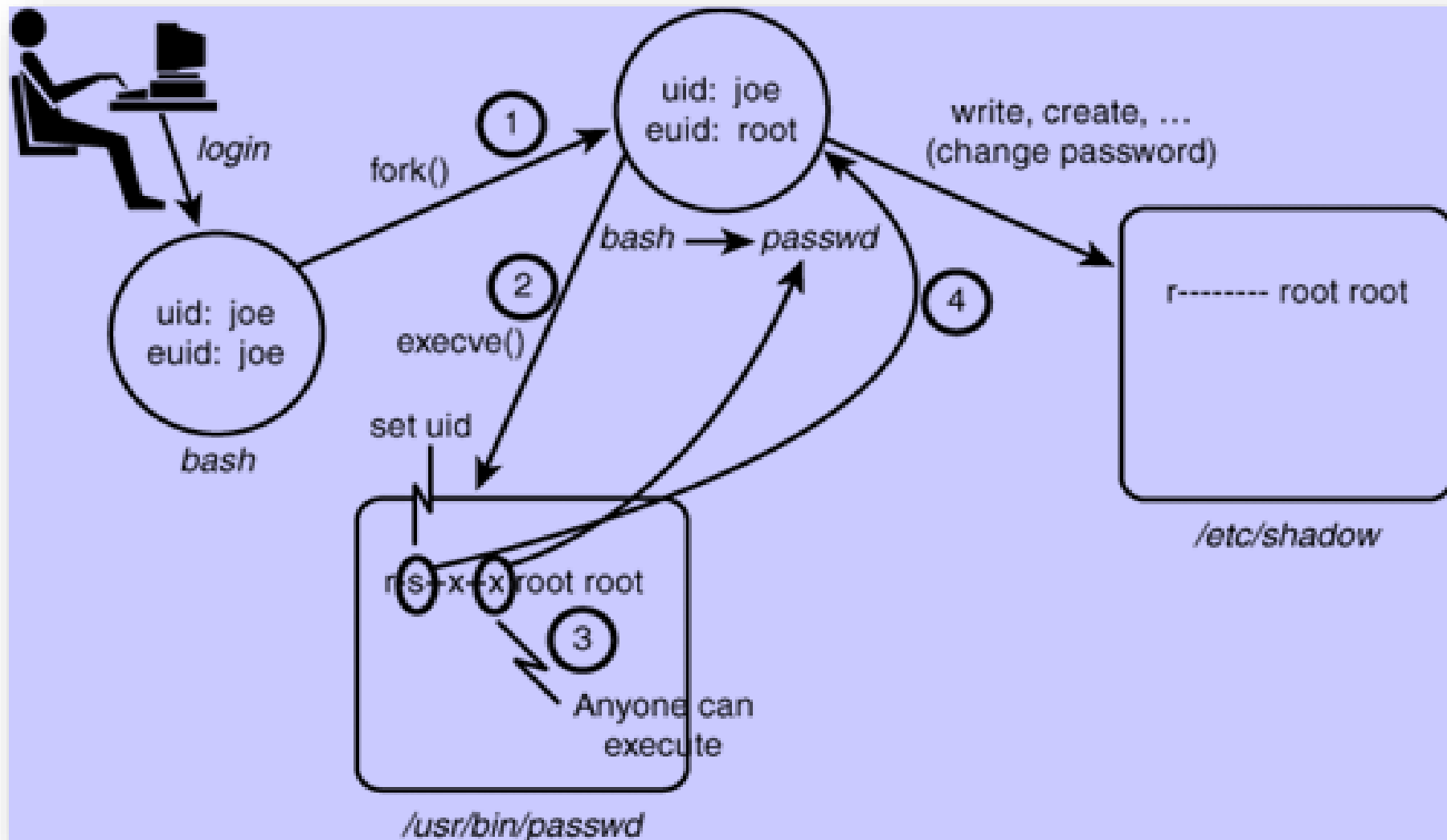
- Les mots de passe sur Linux se changent en utilisant la commande `/usr/bin/passwd`
- Les utilisateurs peuvent changer leur mot de passe
- Root peut changer le mot de passe de tous les utilisateurs

```
# passwd username
```

```
-r-sr-xr-x  1 root  wheel  9856 Sep 24 15:09  
/usr/bin/passwd  
-rw-----  1 root  wheel  2152 Sep 24 15:09 /etc/shadow
```

- Comment est-ce qu'un utilisateur peut changer son mot de passe sans permission d'écriture à `/etc/shadow` ?

Exemple setuid





Limites de DAC

- Linux utilise le Contrôle d'accès discrétionnaire (DAC)
 - Les utilisateurs peuvent changer les permissions de leur fichiers
`chmod 655 /home/david/declaration_impot_16`

- DAC représente les droits sous forme de matrice

	Jean	Paul	Marie
File1	rw	r	w
File2	r	-	rw
File3	r	w	-

- DAC fonctionne correctement sous 2 conditions
 - Si les usagers ne font pas d'erreurs
 - Si on peut faire confiance à tous les programmes
➔ Impossible !!!



Limites du modèle DAC

- Exemple de matrice

	Dossier médical	Ordonnance
Médecin	RW	RW
Patient (Attaquant)	-	R



Limites du modèle DAC

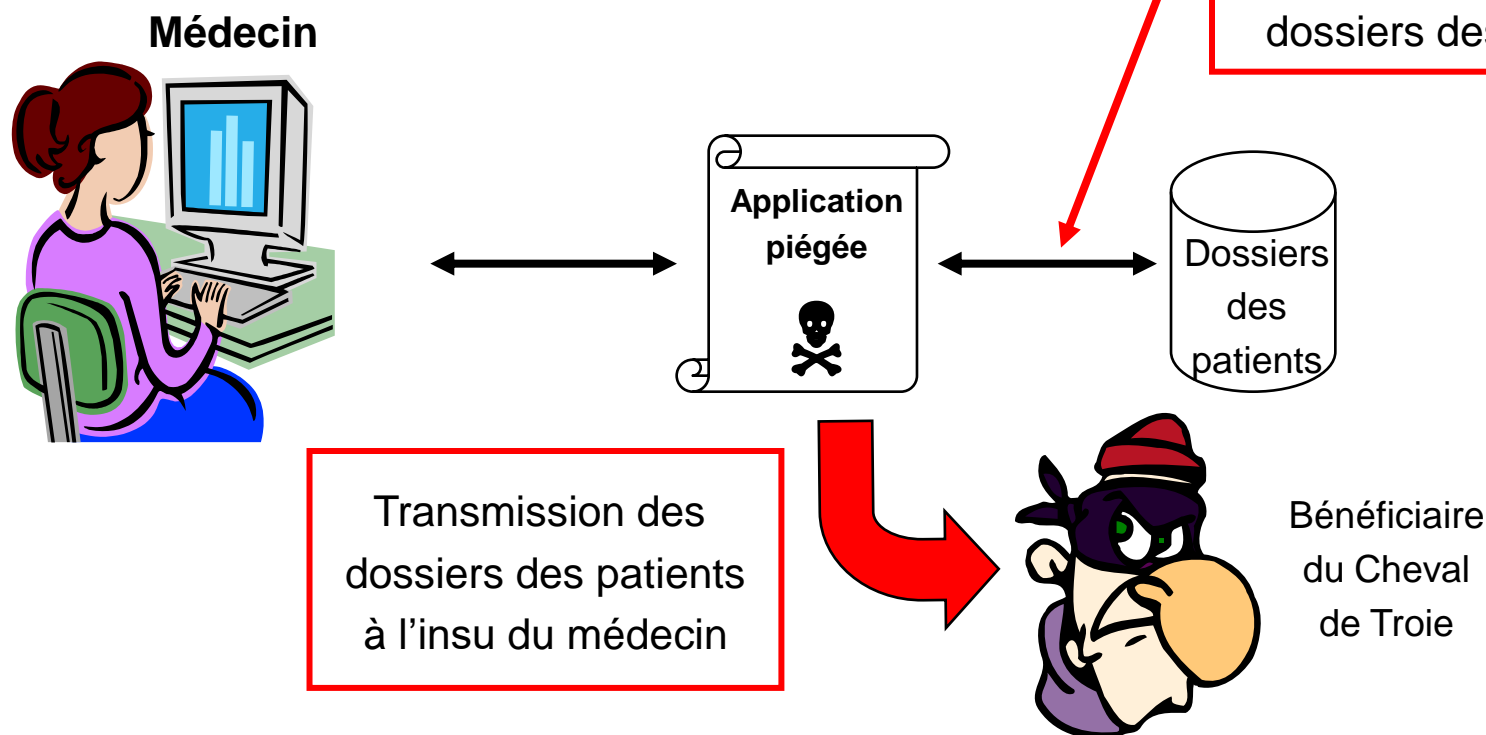
	Dossier médical	Ordonnance
Médecin	RW	RW
Patient (Attaquant)	-	R
Application piégée	RW	RW

Application
s'exécutant pour le
compte du médecin
(hérite des droits
du médecin dans
DAC)

Transfert illégal non
contrôlé par DAC

Limites du modèle DAC

- Illustration
 - Attaque par Cheval de Troie





- MAC = Mandatory Access Control
 - Contrôle d'accès obligatoire
- MAC ajoute des étiquettes à chaque sujet et objet
- Une politique d'accès contient les règles d'accès permises pour chaque sujet et objet
- Politique par défaut
 - REFUSÉ (« Deny ») : l'accès n'est pas permis, à moins qu'il y ait une règle indiquant le contraire dans la politique d'accès
- Les règles et étiquettes peuvent seulement être changées par un administrateur avec un logiciel de confiance (« trusted »)



Politique de sécurité multiniveau

- Exemple classique de contrôle d'accès MAC
- Etiquette = niveau de sécurité
- Exemple

Public \leq Confidentiel \leq Secret



Politique de sécurité multiniveau

- Les utilisateurs reçoivent un niveau d'habilitation
 - Les utilisateurs s'engagent à ne pas diffuser n'importe comment les informations qu'ils détiennent
- Les informations reçoivent un niveau de classification
 - Mesure la confidentialité de l'information



Conditions de sécurité (Modèle de Bell & LaPadula)

- No Read Up
 - Un sujet s peut lire un objet o si :
 - $\text{niveau_classification}(o) \leq \text{niveau_habilitation}(s)$
- No Write Down
 - Un sujet s peut modifier un objet o si :
 - $\text{niveau_habilitation}(s) \leq \text{niveau_classification}(o)$



Conditions de sécurité (suite)

- Objectif du « No write down »
 - Soit un programme s'exécutant au niveau « Secret »
 - Ce programme peut lire des données classées « Secret »
 - Mais le « No write down » empêche un éventuel piège contenu dans ce programme de transmettre les données lues vers un utilisateur qui ne serait pas habilité au niveau « Secret »

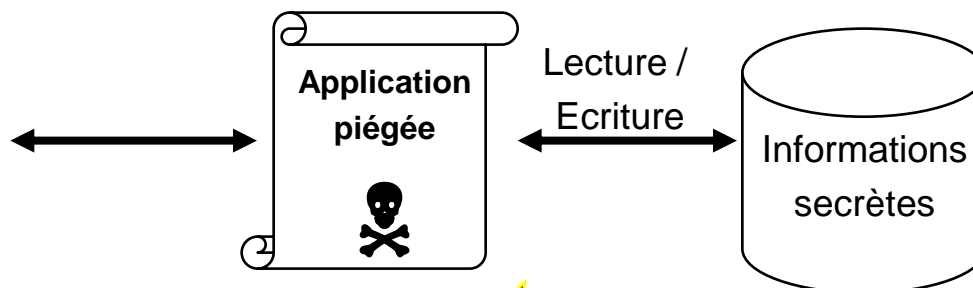


Conditions de sécurité (suite)

Utilisateur habilité

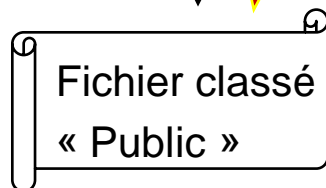
« Secret »

(Médecin)



Ecriture

No
« Write Down »



Lecture

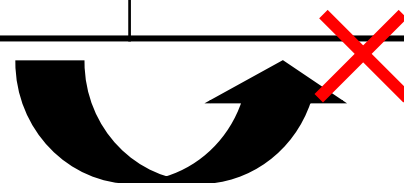


Utilisateur
habilité
« Public »



Conditions de sécurité (suite)

	Dossier médical (classé S)	Ordonnance (classé P)
Médecin (habilité S)	RW	RW
Attaquant (habilité P)	-	R
Application piégée (niveau S)	RW	R W



Transfert illégal bloqué
par Bell et LaPadula



Conditions de sécurité (suite)

	Dossier médical (classé S)	Ordonnance (classé P)
Médecin (habilité S)	RW	RW
Attaquant (habilité P)	-	R
Application piégée (niveau courant S)	RW	RW
Application piégée (niveau courant P)	RW	RW

Intérêt du niveau courant : le médecin doit travailler au niveau P pour pouvoir écrire l'ordonnance



- Conditions de Bell & LaPadula trop rigides
- Aujourd'hui utilisation d'un autre modèle MAC
 - DTE (Domain Type Enforcement)
 - Implanté dans SELinux (Security Enhanced Linux)



Pourquoi RBAC ?

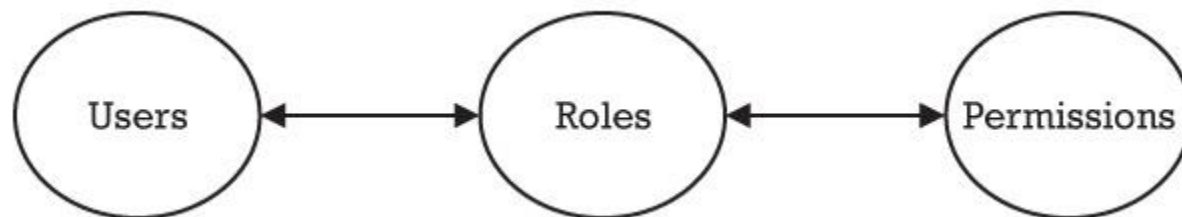
- DAC est de gestion difficile car chaque usager est un cas individuel
 - Considérez des compagnies de milliers d'employés
- DAC suppose que les usagers sont propriétaires des ressources et peuvent transférer les droits sur elles,
 - Tandis que normalement c'est l'organisation qui est propriétaire des ressources, et veut en retenir le contrôle



- RBAC = Role Based Access Control
- RBAC est basé sur deux points
 - Le fait que dans les organisations les employés sont affectés à des rôles
 - Comptable, programmeur, docteur, infirmière, technicien ...
 - Les rôles sont organisés en **hiérarchies**
 - Le fait que chaque employé, pour exécuter son rôle, a besoin de certaines permissions

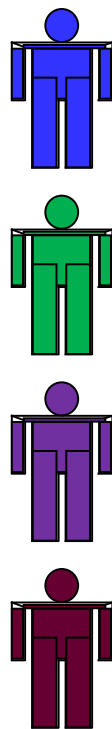


- RBAC s'appuie sur la notion organisationnelle de rôle pour associer des permissions de sécurité aux différents rôles
- Le rôle devient un mécanisme pour associer des permissions aux usagers

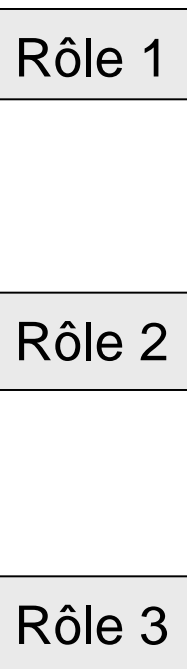




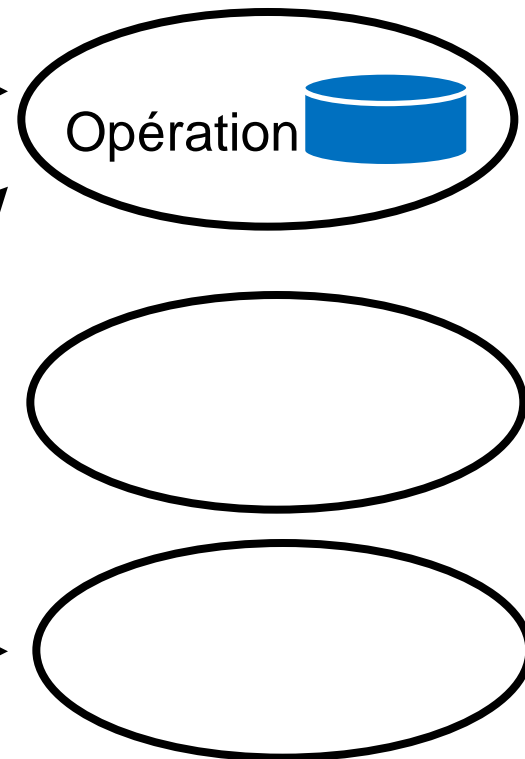
Usagers



Rôles



Permissions



Cette affectation
peut changer souvent

Cette affectation
ne change pas souvent

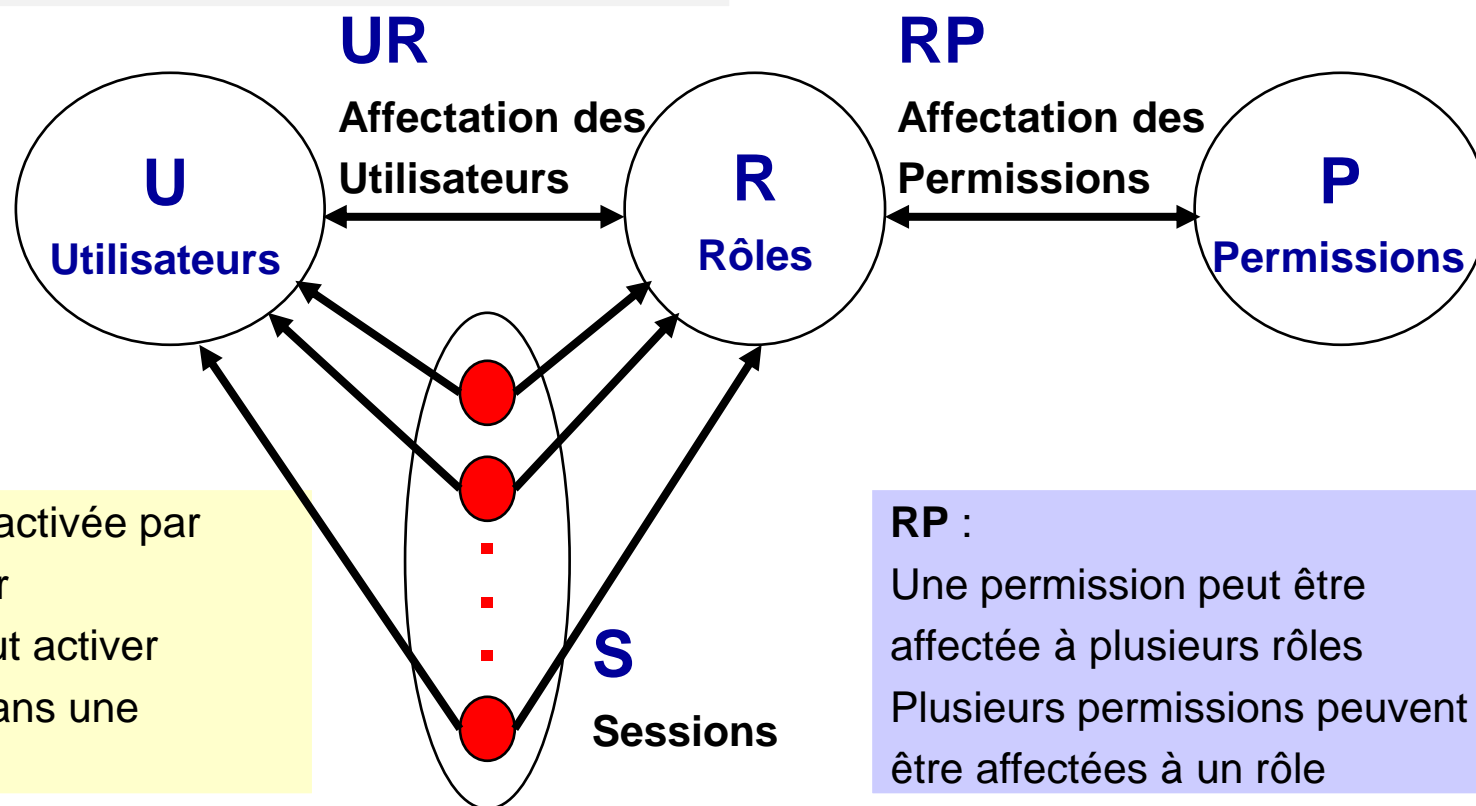


- Pour utiliser ses rôles, un usager doit activer des sessions
- Une session est un processus qui agit pour un usager
 - En changeant de session, un usager peut activer de nouveaux rôle(s)
 - P.ex. un employé de banque Paul peut activer un rôle quand il est aux prêts et un autre rôle quand il est aux investissements
- Pour accéder à une session, un sujet doit s'authentifier
- Un sujet peut se trouver dans plusieurs sessions simultanément



UR :

Un rôle peut être affecté à plusieurs utilisateurs
Plusieurs rôles peuvent être affectés à un utilisateur



Une session est activée par un seul utilisateur
Un utilisateur peut activer plusieurs rôles dans une session

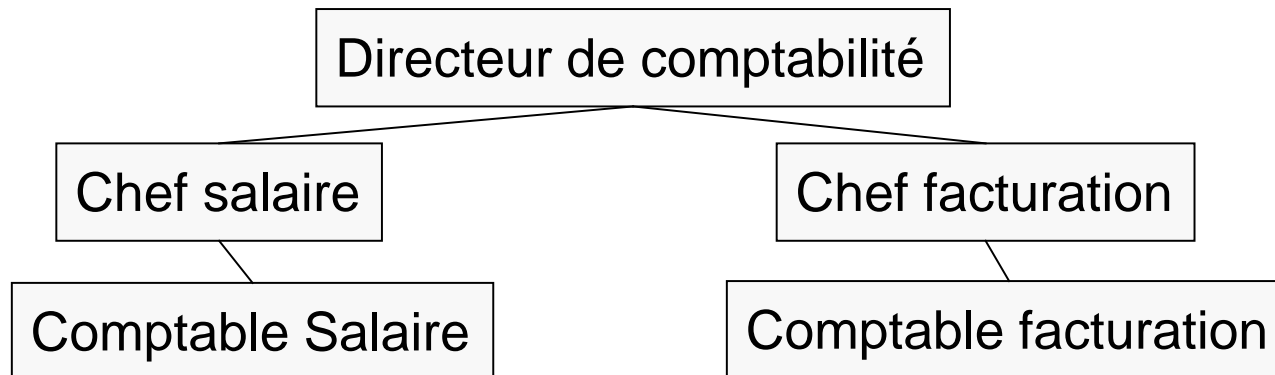
RP :

Une permission peut être affectée à plusieurs rôles
Plusieurs permissions peuvent être affectées à un rôle



RBAC Hiérarchique

- On peut introduire une hiérarchie de rôles
- Propriété de cette hiérarchie
 - Héritage des permissions



- Si Comptable Salaire a une permission, alors le Chef Salaire et le Directeur de Comptabilité l'ont aussi



- Les contraintes sont un élément extrêmement important de RBAC
 - Les contraintes servent à empêcher certaines situations indésirables
- Exemple : contrainte de cardinalité
 - Il ne doit y avoir qu'un seul utilisateur affecté au rôle de directeur



- Contraintes de « séparation des pouvoirs »
 - En anglais : « Separation of duty » (SOD)
 - Ce sont les contraintes les plus importantes
 - Exemple : Celui qui approuve un chèque (rôle R1) ne peut pas être celui qui le signe (rôle R2)
 - Dans RBAC, cela se traduit par une contrainte qui rend impossible qu'un utilisateur soit affecté à R1 et R2
- SOD statique (SSOD) ou dynamique (DSOD)
 - SSOD : la séparation s'applique en toute situation
 - DSOD : la séparation s'applique uniquement dans une même session

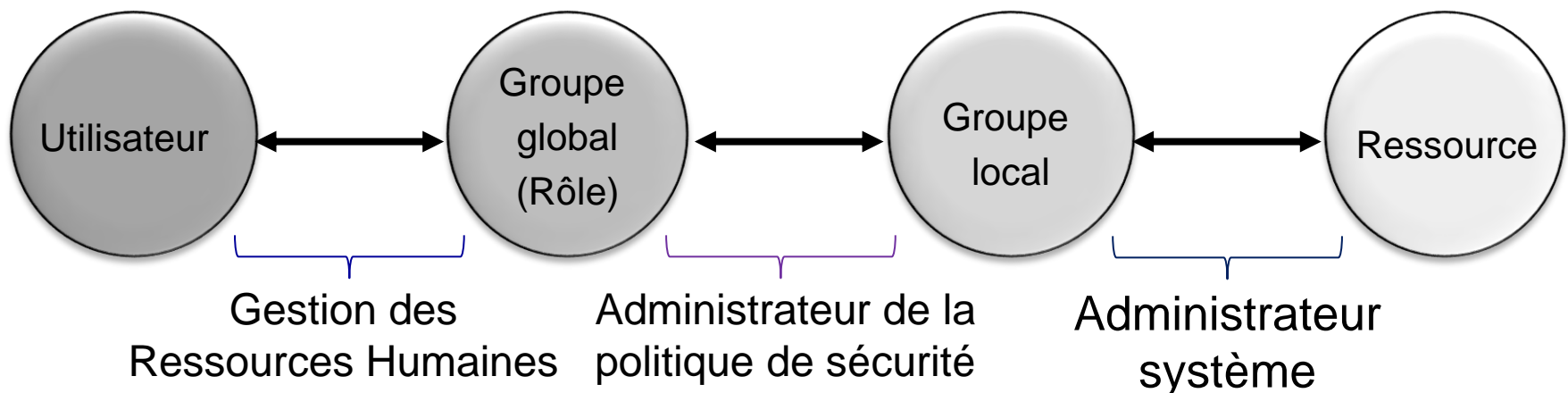


- AGLP
 - Access – Global – Local – Permissions
 - Implémentation de RBAC sous Windows
 - Repose sur les Active Directory(AD)
- Éléments
 - Groupe « Globaux »
 - regroupement des utilisateurs, typiquement selon leur rôles
 - généralement définis sur des serveurs de domaine globaux (d'où le nom)
 - Groupes « Locaux »
 - auxquels sont attribués des Permissions sur des ressources
 - généralement définis et résidants sur les serveurs où ces ressources se trouvent (d'où le nom)
 - les membres sont exclusivement des groupes globaux



- Principes de base

- On n'attribue pas de permissions aux groupes globaux ni aux utilisateurs
- On n'ajoute pas d'utilisateurs dans les groupes locaux
- Séparation des responsabilités
 - Gestion des usagers : U et G
 - Politique de sécurité : G et L
 - Administrateur de système : L et R





- ABAC = Attribute Based Access Control
- Dans ABAC, la décision d'accès dépend de politiques qui combinent entre eux des attributs
 - Attributs de l'utilisateur
 - Attributs de l'action
 - Attributs des ressources
 - Attributs liés à l'environnement
- Politique d'autorisation = ensembles de règles



- L'intention de ABAC est d'être plus général et flexible que les modèles précédents

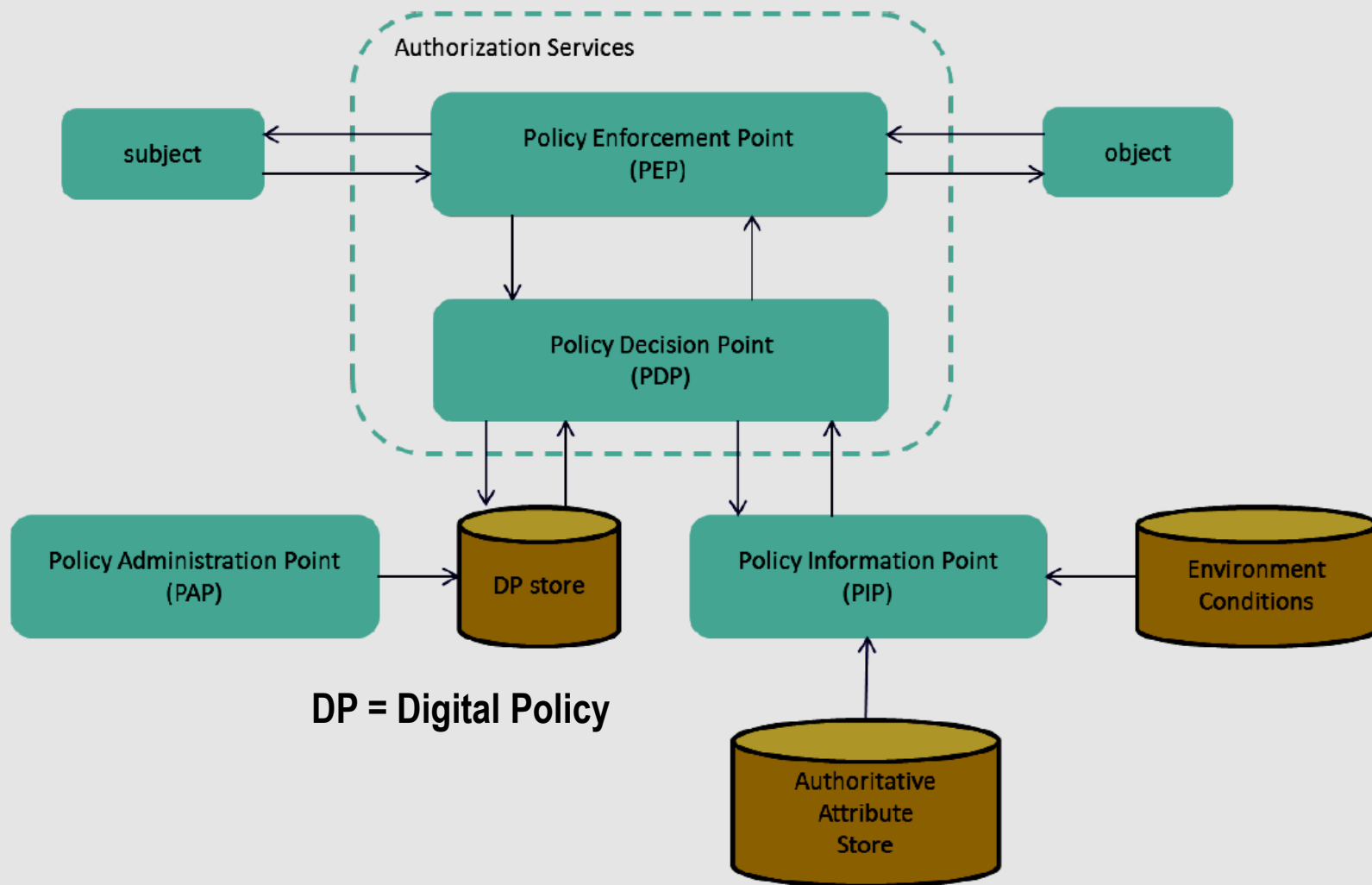


Attribute Based Access Control

- **Sujet, Ressource et Action** sont des catégories qui regroupent des attributs
 - Attributs du Sujet : Nom, Département, Rôle, etc.
 - Attributs de l'Action : Ident, Type
 - Attributs de la ressource : Type, Ident, Auteur
- **Les attributs ont des valeurs**
 - $\text{Nom}(\text{Sujet})=\text{Gervais}$, $\text{Département}(\text{Sujet})=\text{GIGL}$, $\text{Role}(\text{Sujet})=\text{Professeur}$,
 - $\text{Type}(\text{Ressource})=\text{Reserve}$, $\text{Ident}(\text{Ressource})=\text{QA.75.5.2005}$,
 - $\text{Ident}(\text{Action})=\text{EmpruntLivre}$, $\text{Type}(\text{Action})=\text{Bibliotheque}$.
- **La requête de contrôle d'accès est un ensemble d'éléments** ($\text{attribut}(\text{catégorie})=\text{valeur}$) – les paramètres de la requête
 - $\text{Nom}(\text{Sujet})=\text{Gervais}$ et $\text{Ident}(\text{Action})=\text{EmpruntLivre}$ et $\text{Ident}(\text{Ressource})=\text{QA.75.5.2005}$
- **Les règles de contrôle d'accès sont basées sur des cibles exprimées par des expressions booléennes**
 - Permettre si $(\text{Role}(\text{Sujet})=\text{Professeur}$ ou $\text{Role}(\text{Sujet})=\text{Etudiant})$ et $\text{Ident}(\text{Action})=\text{EmpruntLivre}$ et $\text{Type}(\text{Ressource})=\text{Reserve}$ et $7:00 \leq \text{Heure} \leq 20:00$



ABAC Schéma architectural



Source: NIST Special Publication 800-162



Éléments architecturaux de ABAC

- PEP: Policy Enforcement Point
 - Donne ou refuse un accès
- PDP: Policy Decision Point
 - Prend la décision si l'accès doit être donné ou refusé
 - Utilise les politiques et règles qui sont enregistrées dans une base de données appelée Policy Store
- PIP: Policy Information Point - fournit les informations dont le PDP a besoin pour prendre ses décisions
 - Les valeurs des attributs
 - L'état de l'environnement:
 - L'environnement est aussi une catégorie avec ses attributs
 - L'heure et la localisation de l'utilisateur ou de la ressource
- PAP: Policy Administration Point
 - Gère le Policy Store: ajout, suppression de règles



ABAC : Exemple

- Le PEP reçoit la requête
 - (Marc) demande (d'emprunter) (le livre QA.75.5.2005) à (18:00)
- Le PEP informe le PDP qu'il a reçu cette requête
- Le PDP détermine que la règle applicable pourrait être :
 - Permettre (au Professeur) ou (à l'Etudiant) (d'emprunter) (un livre réservé) entre (7:00) et (20:00)
- Mais il ne sait pas si Marc est un professeur, qu'il est 18:00,...
- Le PDP interroge le PIP, le PIP consulte la base des attributs et informe le PDP que :
 - *Marc est un professeur titulaire*
 - *Un professeur titulaire est un professeur*
 - *Le livre QA.75.5.2005 a été réservé*
 - *il est 18:00*
- Le PDP conclut que la demande d'accès est *Permise*
- Le PDP en informe le PEP qui informe Marc



- XACML
 - eXtensible Access Control Markup Language
 - Langage qui implémente le modèle ABAC
- Langage basé sur la syntaxe XML
- Norme OASIS
 - Organization for the Advancement of Structured Information Standards (<https://www.oasis-open.org/>)
 - Première version disponible en 2003
 - XACML v3 depuis 2013



XACML en bref

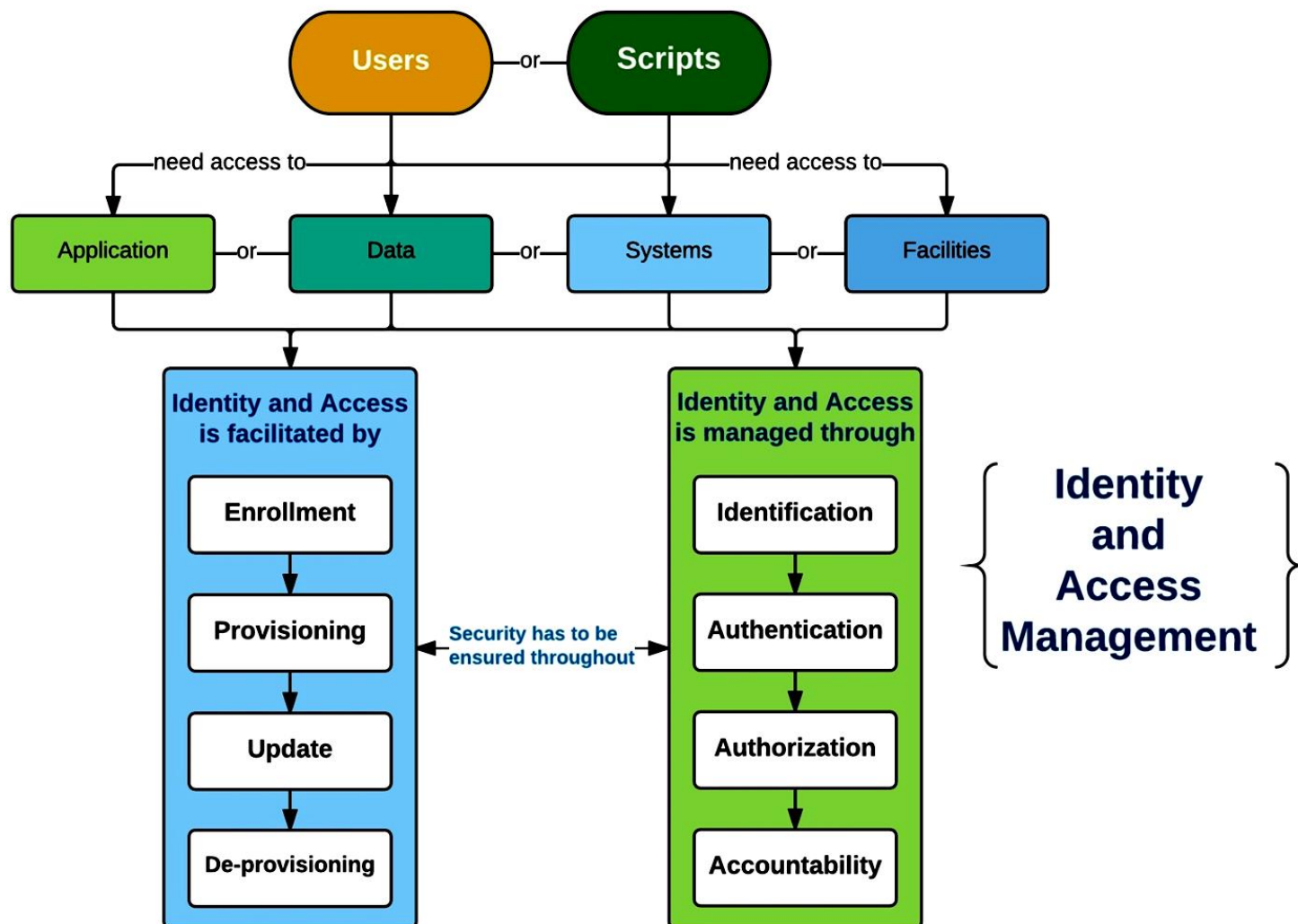
- Une architecture pour l'implémentation
- Des principes de communication entre les composants
- Un langage pour les règles et le politiques
- Un langage pour les requêtes et les réponses
- Types de données normalisés
- Fonctions et algorithmes de combinaison
- Extensibilité
- Différents profils
 - Pour RBAC, ...



- IAM = Identity and Access Management
 - En français : GIA = Gestion des Identités et des Accès
- Principe de base de l'IAM
 - Ne pas mélanger les fonctions du système et le contrôle d'accès
 - Fonctions du systèmes = Services, Applications
- Séparer l'implémentation des applications et de la sécurité
 - Ne pas coder « en dur » l'authentification dans les applications
 - Ne pas coder « en dur » les autorisations dans les applications
- Bon principe pour exprimer, déployer et mettre à jour la politique de contrôle d'accès



Fonctions principales de l'IAM





Fonctions principales de l'IAM

- Le provisionnement
 - Déploiement statique de la politique de contrôle d'accès
 - En Anglais = User Provisioning
- Objectifs
 - S'assurer automatiquement que la politique de sécurité est effectivement appliquée dans les applications
 - En général, via l'exécution de script
- Fonctions principales du provisionnement
 - Créer, mettre à jour, supprimer automatiquement les comptes dans les applications et les systèmes cibles
 - Synchroniser les mots de passe entre les comptes applicatifs



Fonctions principales de l'IAM

- La réconciliation
 - Compare l'état souhaité des comptes, décrit par la politique de sécurité, avec l'état réel existant dans les systèmes et les applications
 - De manière automatique périodique, ou suite à des modifications de la politique, ou à la demande
- Fournit des rapports de réconciliation indiquant les écarts
- Permet de traiter ces écarts manuellement ou automatiquement (avec précautions)
- Exemple
 - Le compte doit exister d'après la politique d'accès définie mais il n'existe pas dans le système cible
 - Le compte est activé dans le système cible, alors que l'autorisation est désactivée



- Gestion des identités
 - Gestion de l'annuaire des utilisateurs
 - Gestion du SSO (Sigle Sign On)
- Gestion des autorisations
 - Expression de la politique d'autorisation
 - RBAC, ABAC, ...
 - Policy Mining
 - Extraction de la politique à partir des comportements observés
 - Supervision de la politique
 - Détection des comportements anormaux et des anomalies de déploiement
 - Exemple : Compte « dormant » qui n'a pas été utilisé depuis 6 mois



**POLYTECHNIQUE
MONTREAL**

UNIVERSITÉ
D'INGÉNIERIE

A la prochaine séance