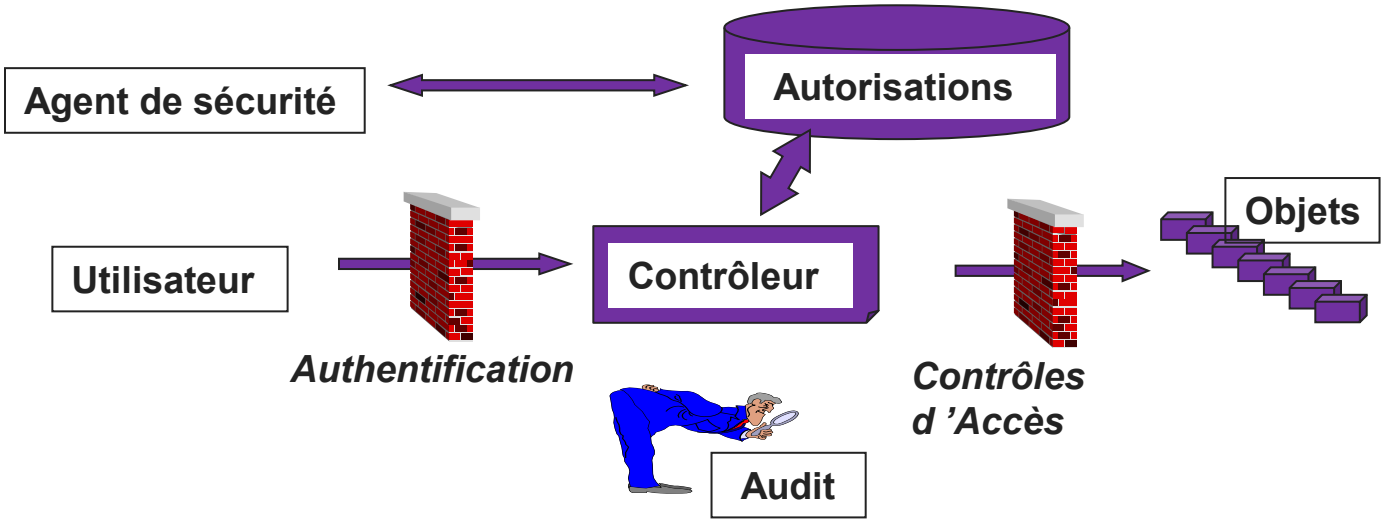
Institut pour la Maîtrise des Risques
Sûreté de Fonctionnement - Management - Cindyniques

MODELES DE SECURITE
Contrôles d'accés

Abdallah M'HAMED

Telecom SudParis- Evry
@mail: ***Abdallah.Mhamed@telecom-sudparis.eu***

Contrôles d 'accès



- **Politique de sécurité**: Ensemble des lois, règles et pratiques qui régissent la façon dont les informations sensibles sont gérées, protégées et distribuées dans le SI ".

- **Autorisation** : Ne permettre que les actions légitimes, c'est-à-dire à empêcher qu'un utilisateur puisse exécuter des opérations pour lesquelles il n'est pas habilité.

Politique de sécurité

- Politique de sécurité : **physique**, **administrative** et **logique**.

- **Physique** : Procédures et moyens de protection des locaux et les biens contre des risques majeurs (incendie, inondation, etc.) et contrôlent les accès physiques aux matériels informatiques et de communication (gardiens, badges, ...).
- **Administrative** : Procédures et moyens organisationnels au sein de l'entreprise.

- **Logique**

  Gestion du contrôle d'accès logique qui repose sur le triplet ***Identification, Authentification et Autorisation***.

  Pour construire une **politique de sécurité** il faut :

1°) définir un ensemble de **propriétés de sécurité** qui doivent être satisfaites.

2°) établir un **schéma d'autorisation**, qui présente les règles permettant de modifier l'état de protection du système.

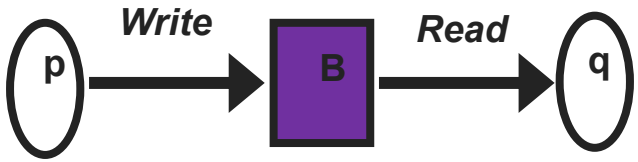
Politique de sécurité

Politique de sécurité est **Cohérente** si aucune séquence valide d'applications des règles (du schéma d'autorisation) ne puisse amener le système dans un état de violation d'un tel objectif de sécurité, en partant d'un état initial sûr.

- Partitionnement de l'ensemble des états: **Etats autorisés / non autorisés.**

Exemple: *Buffer partagé*

- État $Q = (S, O, A)$ est autorisé si et seulement si, pour chaque buffer **B** $\in O$:
  - un seul processus **p** $\in S$ tel que $W \in A$ [p, B]
  - un seul processus **q** tel que $R \in A$ [q, B]



- Transition Q vers Q' (autorisé ou non) selon la commande C : $Q \rightarrow_C Q'$

- Recours à une **méthode formelle** de construction des règles du schéma d'autorisation, à partir d'une spécification formelle des objectifs de sécurité.
- **Modèle formels**: vérifier que la politique de sécurité est complète et cohérente et que sa mise en œuvre est conforme *[ITSEC, Critères Communs*]*

Modèles de sécurité

- **Discrétionnaires** (**DAC** : Discretionary Access Control)
  - Les droits d'accès sont manipulés librement par le propriétaire, ***à sa discrétion***.

- **Obligatoires** (**MAC** : Mandatory Access Control)

  Structure ***hierarchique/multi niveaux*** des utilisateurs/ressources
  - Les privilèges des sujets sur les objets sont données par des règles générales
  - Origine: Systèmes militaires
  - Contrôle des flots d'information

- **Basées sur la notion de rôles** (**RBAC** : Role-Based Access Control)
  - + récente, adaptée aux organisations
  - Les droits sont accordés à des rôles, des rôles sont attribués aux sujets.
  - Les sujets possèdent les droits selon les rôles qu'ils détiennent.

- **Confidentialité** : ***(Lampson, HRU, BLP,…).***
- **Intégrité** : ***(Biba, Clark wilson…..)***
- **Disponibilité** : ***(Willen…)***

Modèle HRU *(1976)*

- Expression des ***permissions***: relations entre ***sujets***, ***objets*** et ***actions***

S = {***Sujets*** : entités actives} $S \subseteq O$

O = {***Objets*** : entités à protéger}

A = Matrice des droits d'accès

$D = \{A(s,o)\}$

Dans les OS :

S = {users, processus, domaines}

O = {fichiers, segments de mémoire, processus}

D = {*Read, Write, Execute, Own*}

Transitions : 6 primitives

Enter R into *A [s, o]*
Delete R from *A [s, o]*
Create Subject s
Create Object o
Destroy Objet o
Destroy Objet s

$Q \quad (S,O,A) \rightarrow Q' \quad (S',O',A')$

Objets	**M1**	**M2**	**F1**	**F2**	**P1**	**P2**
P1	*R W E*		*Own R W*			
P2		*R W E*		*Own R E*		

Sujets

Modèle HRU *(1976)*

1) Création de fichier:

Command Create file (p,f)
Create Object f
Enter Own into A [p,f]
Enter R into A [p,f]
Enter W into A [p,f]
End

2) Attribution des droits d'accès:

Command Confer.Read (p,q,f)
If Own in A [p,f]
Then enter R into A [q,f]
End

3) Retrait de droit d'accès:

Command revoke.read (p,q,f)
If Own in A [p,f]
Then delete R into A [q, f]
End

Modèle BLP *(1973)*

- **Classification**– *Informations* : **sensibilité** ou niveau de confidentialité
- **Habilitation** – *Utilisateurs:* niveau de sécurité attribué à l'utilisateur
  - Sujet *Si* ⟶ **Habilitation** *H(Si).*
  - Objet *Oj* ⟶ **Classification** *C(Oj).*
- ***Classes de sécurité****: Non Classifié, Confidentiel, Secret, Trés secret*

Règles de sécurité

Pour pouvoir accéder a l'information, le sujet doit
- avoir une habilitation égale ou supérieure à la classification de l'information
- avoir les sous classes de l'information dans sa liste des sous-classe autorisées.

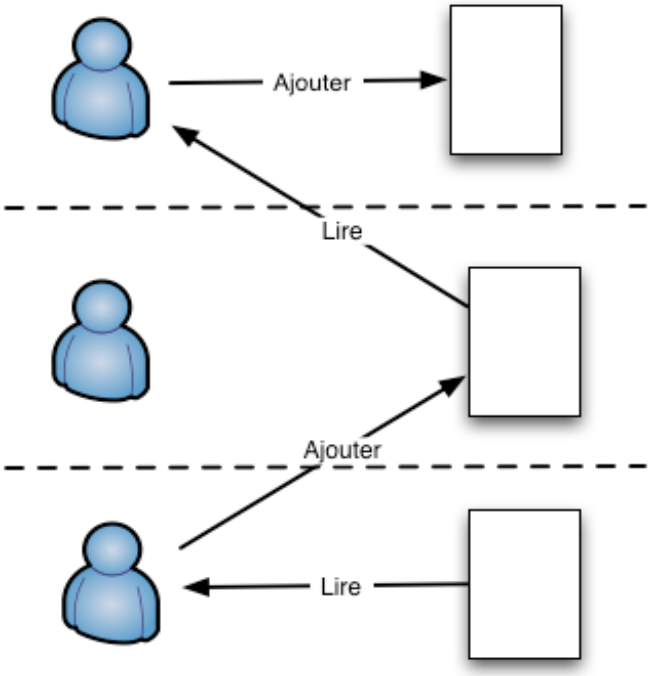
Modèle BLP *(1973)*

- Propriété **simple**:

$$(S_i, O_j, Lire) \Rightarrow H(S_i) \geq C(O_j)$$

- Propriété étoile :

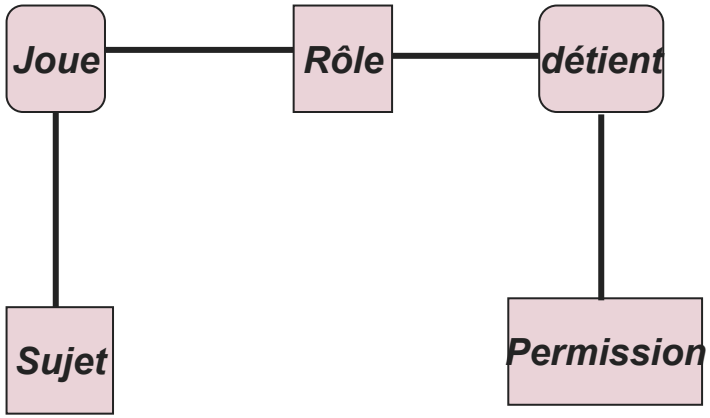
$$(S_i, O_j, lire) \wedge (S_i, O_k, écrire) \Rightarrow C(O_k) \geq C(O_j)$$



- **PS**: Si peut lire vers le bas et écrire vers le haut.
- **P***: Si ne peut pas écrire dans un objet de niveau inférieur au niveau de l'objet plus élevé auquel il peut accéder.
- Empêche Si de déclassifier l'information contenue dans les objets auxquels il a accès.

Modèle RBAC *(1992)*

- *Rôle actif* : *AR (s: sujet) = { Rôle actif du sujet s }*
- *Rôles autorisés*: *RA(s: sujet) = { Rôles que le sujet peut assumer }*
- *Transactions autorisées par rôle*:
  TA(r :rôles) ={ Transactions autorisées pour tous les rôles dans r }



- ***Attribution des rôles*** : un sujet est autorisé a exécuter une transaction uniquement s'il a choisi ou reçu un rôle :

  $\forall s : sujet, \forall t: transaction \mid exec (s , t) \Rightarrow AR (s) \neq \phi$
- ***Autorisation pour un rôle****: le rôle actif* d'un sujet doit être autorisé pour le sujet

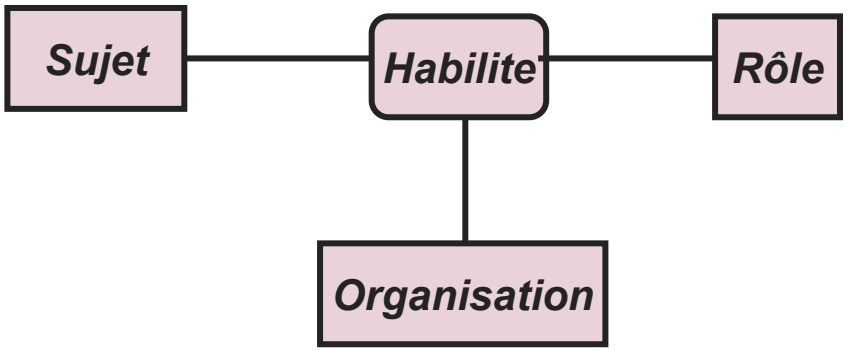
  $\forall s : sujet \mid AR (s) \subseteq RA (s)$
- ***Autorisation pour une transaction*** : un sujet peut exécuter une transaction seulement si son *rôle actif* est autorisé à l'exécuter

$\forall s :sujet, \forall t: transaction \mid exec (s , t) \Rightarrow t \in TA (RA (s))$

Modèle OrBAC

- L'entité centrale est **l'organisation**: *un groupe structuré d'entité actives*
  Sujet: entité active (*utilisateur* ou *organisation*)
- **Rôle** : utilisé pour structurer le lien entre les **sujets** et les **organisations.**

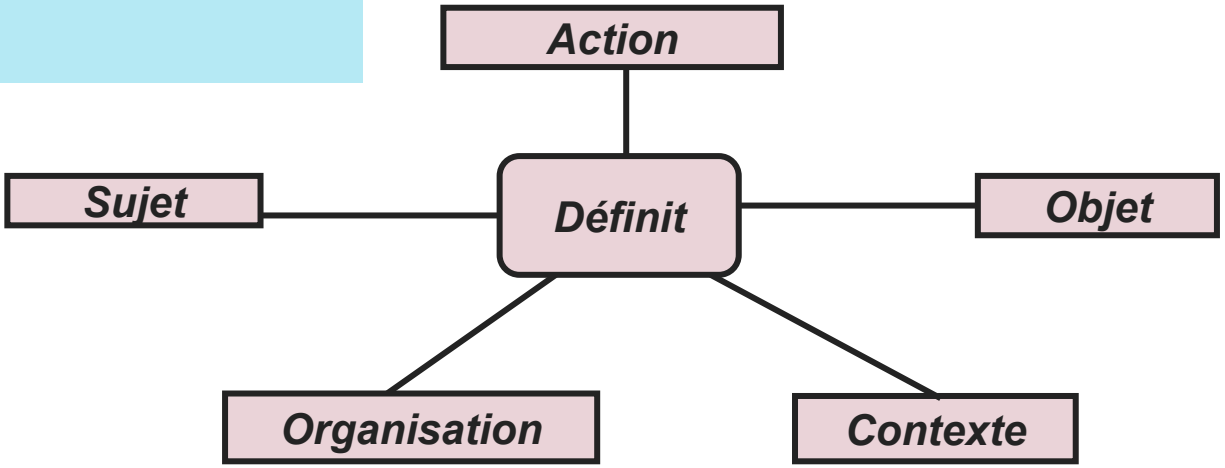
- *Habilite(Org , s , r)* :
  Org habilite le **sujet** *s* à jouer le **rôle** *r.*



Modèle OrBAC

- Actions: *Permissions*, *Interdictions*, *Obligations* et *Recommandations*.
  Permission (Org , r , a , v) : *Org* accorde au **rôle** *r* la **permission** de réaliser **l'activité** *a* sur la **vue** *v*.
- Les mêmes raisonnements s'appliquent aux *Interdictions*, *Obligations* et *Recommandations*.

- *Définit (Org , s ,α , o , c) :* Au sein d'*Org* , le **contexte** *c* est vrai entre le **sujet** *s* et **l'objet** *o* et **l'action** *α*



Evaluation CC*

Objet: garantir à un utilisateur un certain niveau de confiance dans un produit/système logiciel ou matériel.

L'évaluation est conduite par un tiers (ni client, ni fournisseur).

But: Promouvoir et faire connaître auprès de ses clients la valeur de ses produits. Se démarquer de produits concurrents ne disposant pas d'évaluation.

**Common Criteria for Information Security Evaluation (ISO 15408)*

- EAL 1 : testé fonctionnellement
- EAL 2 : testé structurellement
- EAL 3 : testé et validé méthodiquement
- EAL 4 : conçu, testé et revu méthodiquement
- EAL 5 : conçu à l'aide de **méthode semi-formelles** et testé
- EAL 6 : conception vérifiée à l'aide de **méthodes semi-formelles** et testé
- EAL 7 : conception vérifié à l'aide de **méthodes formelles** et testé