# Les modèles RBAC et OrBAC

# Introduction

- La technologie de contrôle d'accès a évolué grâce aux efforts de développement et de recherche supportés par le département de défense.
- Ces recherches ont données naissance à deux types fondamentaux de contrôle d'accès.
    1. Contrôle d'accès discrétionnaire.
    2. Contrôle d'accès obligatoire
- Plusieurs modèles de contrôle d'accès ont été développés par la suite.

# RBAC (Role- Based Access Control)
# - Définition-

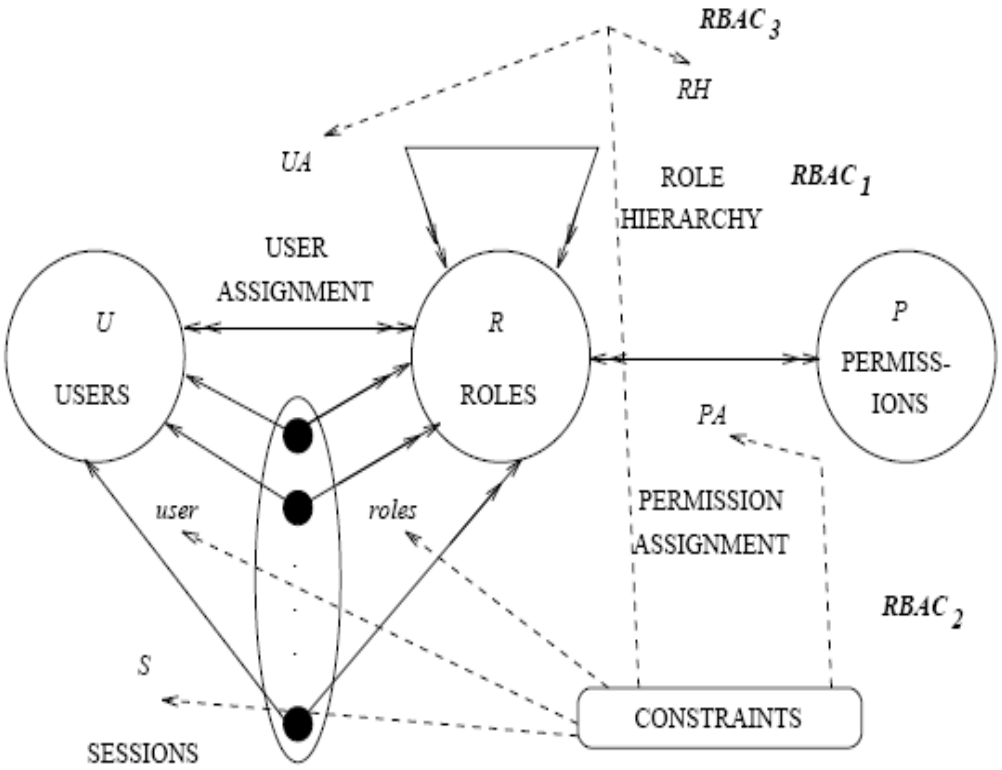
- Les décisions d'accès sont basées sur les rôles attribués aux individus dans une organisation. *Ex: chef de projet, programmeur,…*
- Les droits d'accès sont groupés par le nom du rôle, et l'utilisation des ressources est restreinte aux utilisateurs autorisés à assumer le rôle associé. *Ex: dans un système hospitalier, le rôle Médecin inclus les opérations faire diagnostique, prescrire des médicaments, …*
- L'utilisation du rôle pour le contrôle d'accès peut être une manière efficace pour développer une politique de sécurité.

# RBAC (Role- Based Access Control)
# - Utilisateurs et Rôle-

- L'attribution des rôles aux utilisateurs est faite selon leurs responsabilités dans l'organisation.
- L'opération « utilisateur est permis de faire » est basée sur le rôle de l'utilisateur.
- Le modèle $RBAC_0$ est composé de :

1. U, R, P, et S (utilisateurs, rôles, permissions et sessions respectivement),
2. $PA \subseteq P \times R$, relation d'attribution de permissions
3. $UA \subseteq U \times R$, relation d'attribution de rôles
4. user : $S \rightarrow U$, une fonction reliant chaque session $s_i$ à un seul utilisateur *user($s_i$)*
5. roles : $S \rightarrow R^2$, une fonction reliant chaque session $s_i$ à un ensemble de rôles *roles($s_i$)*

# RBAC (Role- Based Access Control) - Utilisateurs et Rôle-

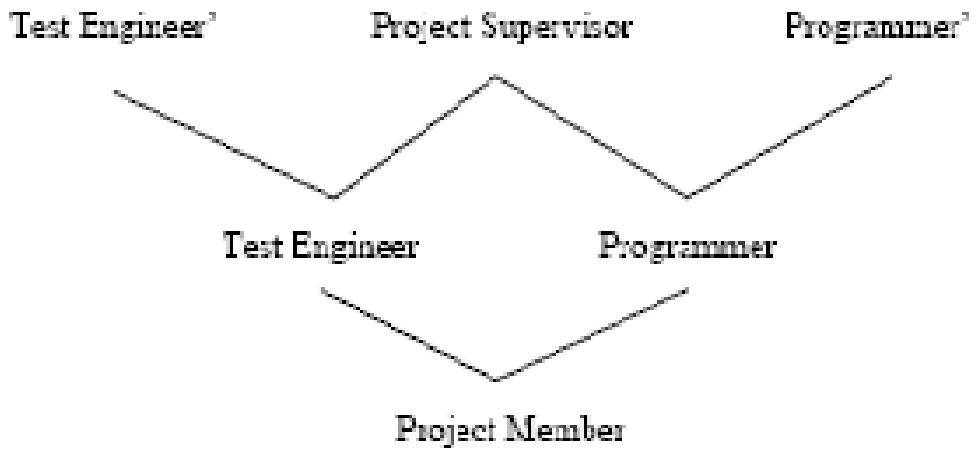



(b) RBAC models

Figure 1: A Family of RBAC Models

# RBAC (Role- Based Access Control)
# - Hiérarchie de rôle-

- Le modèle $RBAC_1$ introduit l'hiérarchie de rôle (RH)
- L'hiérarchie de rôle est la définition naturelle de la structuration des rôles pour refléter la responsabilité dans une organisation




(c)

Figure 2: Examples of Role Hierarchies

# RBAC (Role- Based Access Control)
# - Hiérarchie de rôle-

- Le modèle $RBAC_1$ se compose de:

1. U, R, P, S, PA, UA, et user restent inchangés de $RBAC_0$,
2. $RH \subseteq R \times R$ est un ordre partiel sur R nommé relation de hiérarchie de rôles, écrite aussi ≥.

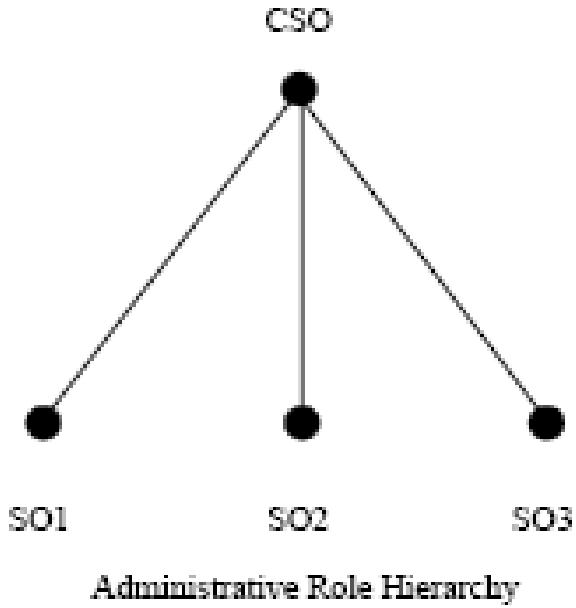



Figure 3: Role Hierarchies for a Project

# RBAC (Role- Based Access Control)
# - Contraintes-

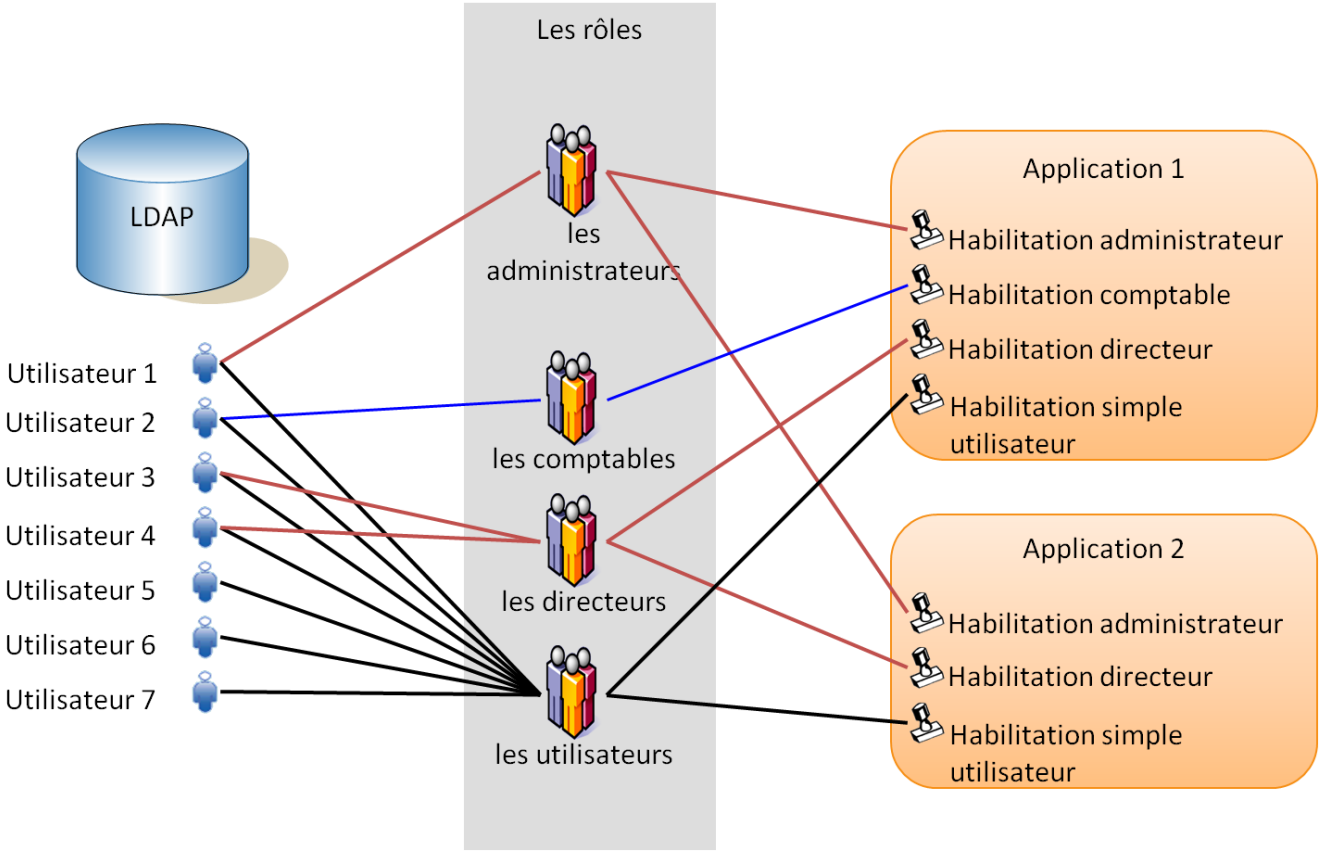
- C’est un aspects important de RBAC.
- Mécanisme fort pour mettre en œuvre une politique organisationnelle de haut niveau.
- Deux type de contraintes:

1. Dans la majorité des organisations, un individu n’est pas permis d’être membre de deux rôles, ce qui est connu par *la séparation des droits*.
2. Un rôle peut avoir un nombre maximum de membres, *par exemple, une seule personne peut être dans le rôle chef de département*, de même, un individu peut être membre d’un nombre limité de rôles, cette contrainte est nommée *contrainte de cardinalité*.

# RBAC (Role- Based Access Control)
# - Contraintes-

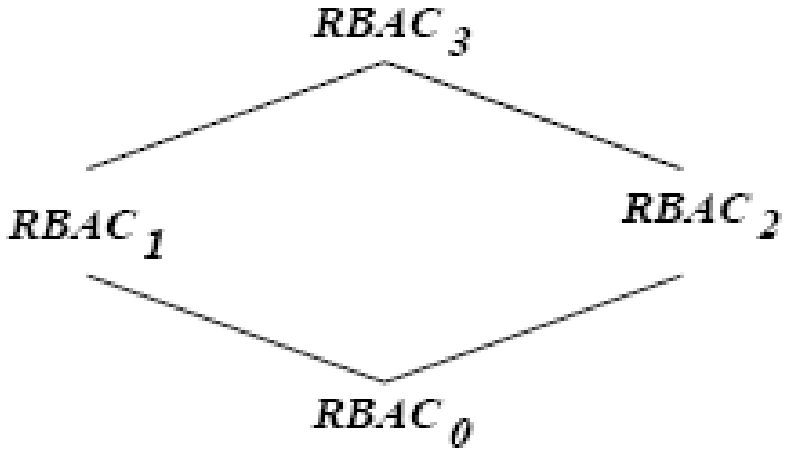
- En respectant $RBAC_0$, les contraintes sont appliquées sur les relations PA et UA, ainsi que les fonctions user, roles pour plusieurs sessions.
- Les contraintes sont des prédicats appliqués sur ses relations et fonctions, et retournent la valeur ‘accepté’ ou ‘non accepté’.
- Ils peuvent être appliqués aux sessions.
- L’hiérarchie de rôle peut aussi être considérée comme une contrainte.

# RBAC (Role- Based Access Control)
# - Exemple-



# RBAC (Role- Based Access Control)
# - Avantages-

- RBAC permet aux utilisateurs d'effectuer un large champs d'opérations et fourni une grande largeur et flexibilité d'applications.
- Permet le contrôle des actions des utilisateurs en définissant les rôles, hiérarchie de rôles, relations et contraintes




Relationship among RBAC models

# RBAC (Role- Based Access Control)
# - Inconvénients-

- Le modèle RBAC atteint rapidement ses limites dès lors que les utilisateurs sont géographiquement différentiables, ou dès lors que l'entreprise est composée de services indépendants.
- Par exemple, dans une société de services, les commerciaux sont attachés à des secteurs d'activités : un commercial chargé des affaires de l'industrie ne peut pas créer de contrat pour un client dans le secteur de la banque.

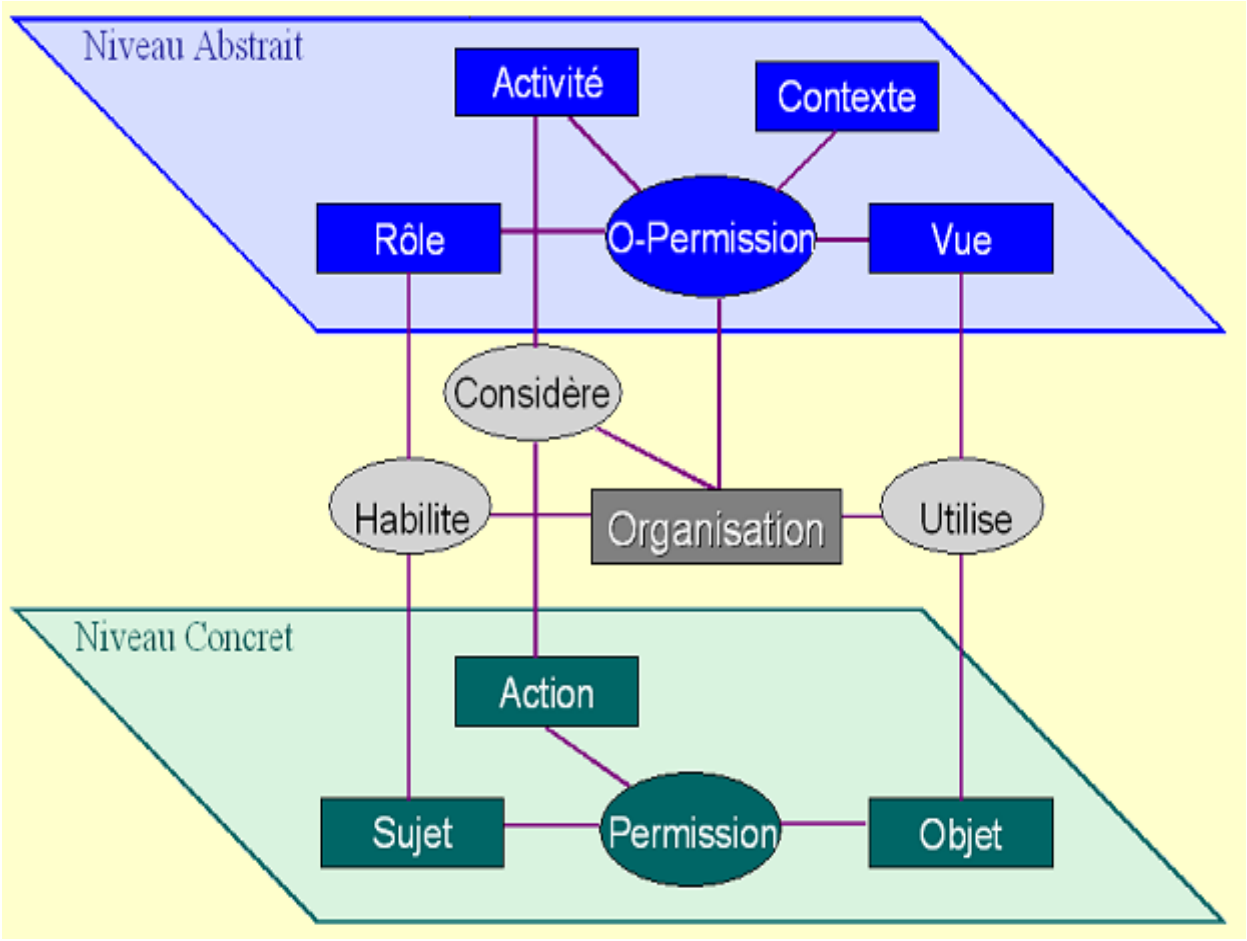
# RBAC (Role- Based Access Control)
# - Inconvénients-

- Autre exemple, un conseiller clientèle d'une banque dispose des droits de gestion de patrimoine et de gestion immobilière.
- Selon l'agence dans laquelle il travaille, il peut soit faire de la gestion de patrimoine soit de la gestion immobilière, soit les deux.
- Pour lui fournir un poste de travail adapté à ses attributions en fonction de l'agence dans laquelle il travaille, ses droits doivent être comparés aux possibilités offertes par ces agences. Ainsi, même s'il est conseiller en gestion de patrimoine et en gestion immobilière, dans une première agence il disposera d'un poste de travail restreint à la fonction de gestion immobilière – l'agent ne fera que de la gestion immobilière. Dans une autre agence il n'aura que celle de gestion de patrimoine – l'agent ne fera que de la gestion de patrimoine. Enfin, dans une troisième agence, son poste de travail lui proposera les deux fonctionnalités – l'agent pourra faire de la gestion de patrimoine ainsi que de la gestion immobilière.

# OrBAC (Organization Based Access Control) Définition

- Reprend les concepts de rôle, activité, vue et organisation introduit respectivement dans RBAC, TBAC (Task Based Authorization Controls), VBAC (View-based Access Control) et TMAC.
- L'expression d'une politique de sécurité est centrée sur le concept d'organisation.
- Il a une politique de sécurité indépendante de son implémentation, il permet aussi d'exprimer les permissions, les interdictions, et les obligations.
- Il prend en compte les contextes, et permet d'introduire des hiérarchies et la délégation.
- Afin de réduire la complexité de gestion des droits d'accès, un niveau abstrait est introduit.

# OrBAC (Organization Based Access Control) Définition




Les interactions d'OrBAC

# OrBAC (Organization Based Access Control) - Contexte -

- Les contextes permettent d’exprimer des permissions ou des interdictions dans certaines circonstances.

  *Ex: dans un contexte d’urgence, on désirera qu'un infirmier puisse accéder au dossier d'un patient x sans avoir besoin d'appeler l'administrateur afin que celui-ci lui donne les droits.*
- Différents type de contextes:

# OrBAC (Organization Based Access Control)
# - Contexte -

1. **Contexte temporel** : Ce sont des contextes régissant la durée de validité des privilèges.
2. **Contexte spatial** : Il peut être lié à l'appartenance à un réseau, ou la position géographique, ou à tout autre situation spatial.
3. **Contexte déclaré par l'utilisateur** : Ce type de contexte est activé, par exemple, par le médecin en cas d'urgence, ou pour signaler que l'on effectue un audit. Dans ces cas exceptionnels, des permissions peuvent être données alors qu'elles seraient interdites dans un cas normal.

# OrBAC (Organization Based Access Control)
## - Contexte -

L'utilisateur qui déclare le contexte est obligé en contrepartie de faire un compte-rendu des opérations effectuées et peut être des raisons qui l'on motivé à déclarer ce contexte.

4. **Contexte prérequis** : Leur utilisation permet de contraindre les sujets concernés par les permissions ou les interdictions dépendant de ces contextes et qui vient réduire ou étendre les droits d'accès hérités du rôle associé.
5. **Contexte provisionnel** : donner des privilèges en fonction de l'historique. Par exemple, le contexte "accès limités à 2 fois" regarde si le document a été accédé au moins 2 fois.

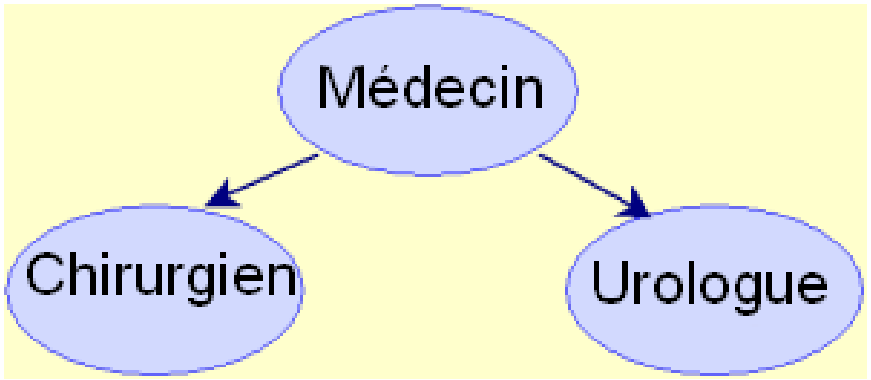
# OrBAC (Organization Based Access Control) - Hiérarchie de Rôle -

- OrBAC permet de définir des hiérarchies sur les rôles, les activités, les vues et les contextes $\Rightarrow$ héritage des permissions et des interdictions.
- Comme dans RBAC, il existe deux façons de définir la hiérarchie de l'héritage:

1. **la hiérarchie organisationnelle**. R1 est senior de R2 et R2 est junior rôle de R1, si un utilisateur jouant le rôle R1 est supérieur hiérarchique de R2.
2. la hiérarchie obtenue par la relation de **spécification/généralisation** est définie telle que R1 est un senior rôle de R2 si chaque fois qu'un utilisateur joue le rôle de R1, il joue le rôle de R2.

# OrBAC (Organization Based Access Control)
# - Hiérarchie de Rôle -

- Par exemple sur la hiérarchie présentée sur le schéma un peu plus en dessous, le chirurgien est aussi un médecin.
- Donc à chaque fois qu'un utilisateur est associé au rôle de chirurgien, il joue aussi le rôle de médecin.
- Le rôle chirurgien est un senior rôle du rôle médecin. Un rôle R1 senior de R2 hérite donc les permissions affectées à R2.




Héritage

# OrBAC (Organization Based Access Control)
# - Délégation-

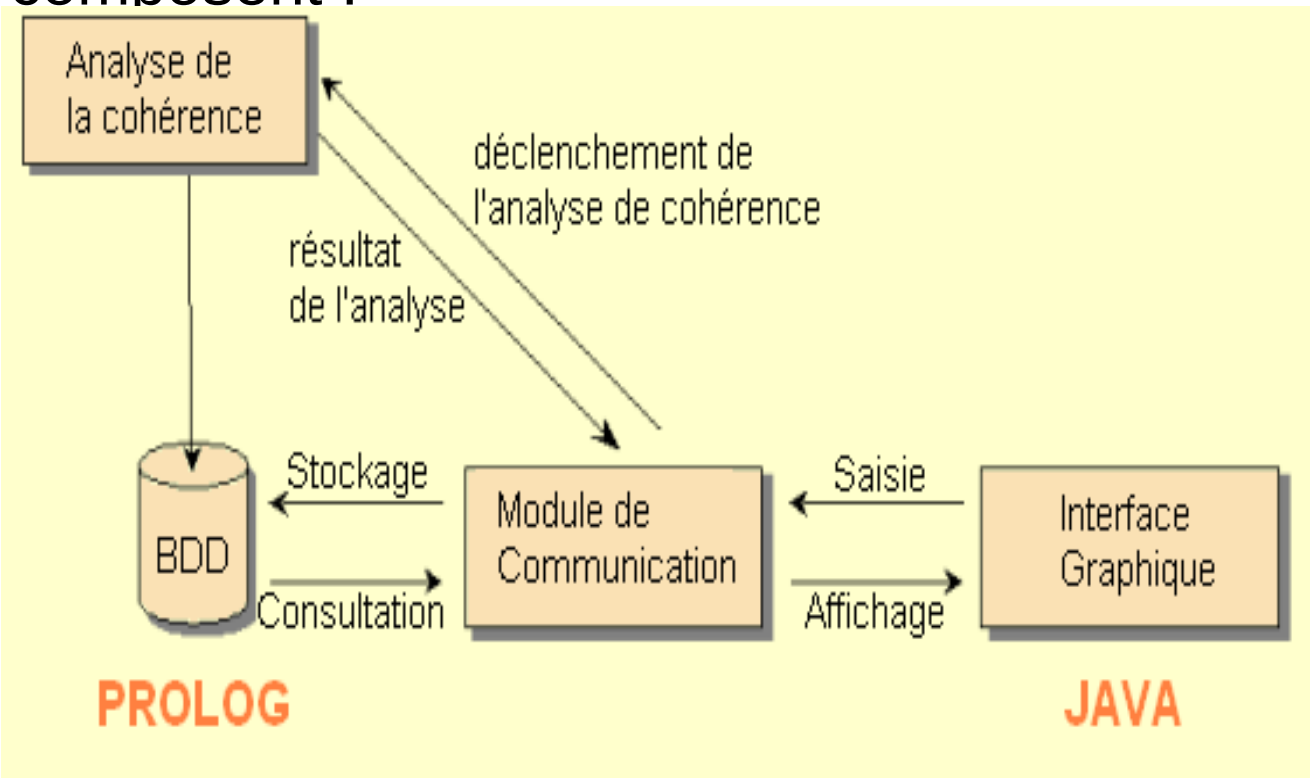
- La délégation permet de donner à un utilisateur particulier un privilège, sans donner ce privilège à toutes les personnes ayant le même rôle que lui.
- La délégation, bien que très utilisée, est très peu modélisée dans les politiques de sécurité car ce concept est très complexe.

# MotOrBAC

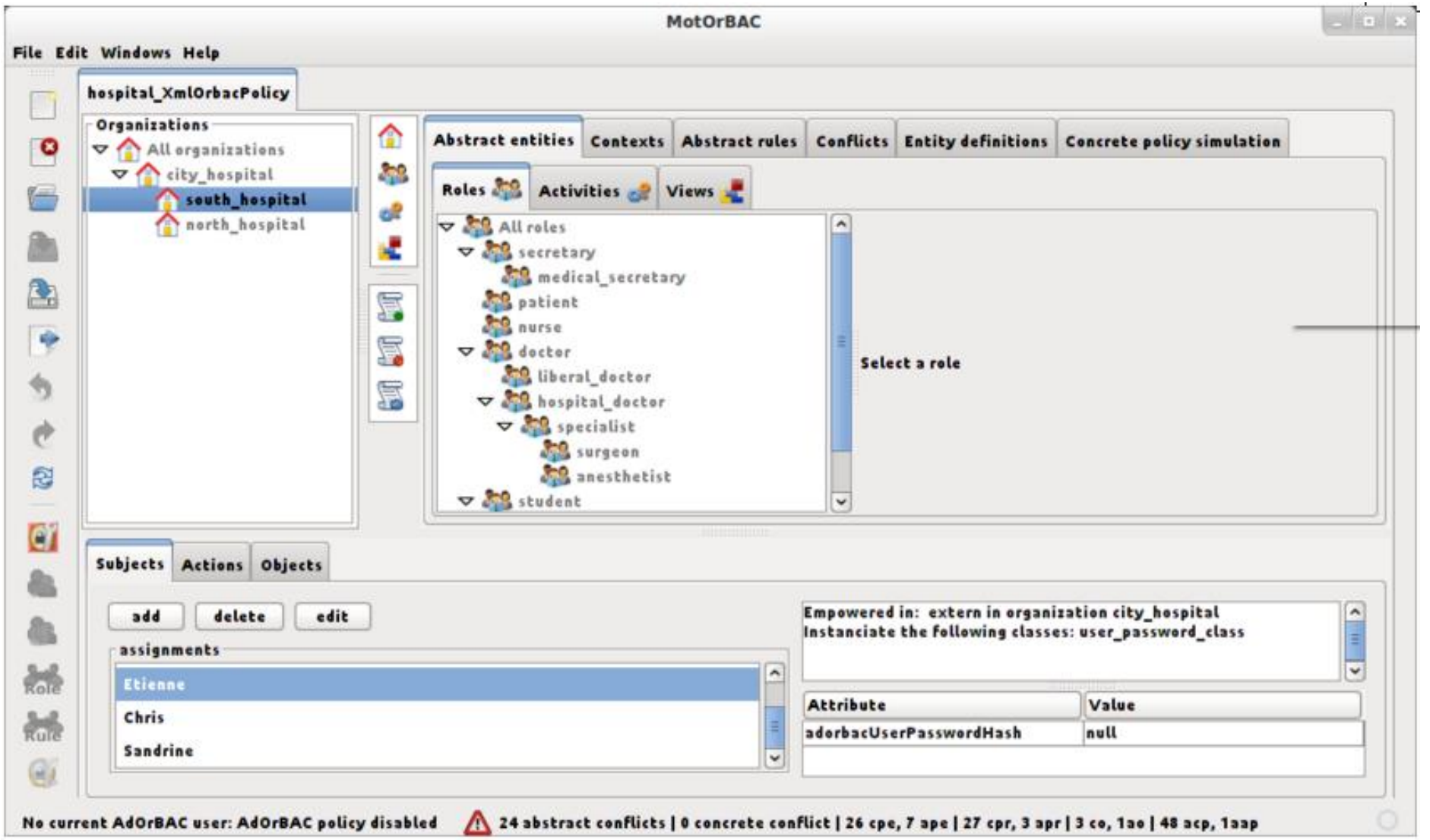
- Le prototype MotOrBAC est un outil de saisie de la politique de sécurité abstraite.
- Il permet de définir une politique de sécurité via le modéle OrBAC en créer des organisation, rôles, activités, vues, contextes et les règles qui leurs sont liées.
- MotOrBAC permet de simuler la politique en saisissant les sujets, actions et objets liés à la politique.

# MotOrBAC

On voit sur le schéma suivant les différents modules qui le composent :



# MotOrBAC


MotOrBAC
File Edit Windows Help
hospital_XmlOrbacPolicy
Organizations
All organizations
city_hospital
south_hospital
north_hospital
Abstract entities
Contexts
Abstract rules
Conflicts
Entity definitions
Concrete policy simulation
Roles
Activities
Views
All roles
secretary
medical_secretary
patient
nurse
doctor
liberal_doctor
hospital_doctor
specialist
surgeon
anesthetist
student
Select a role
Subjects
Actions
Objects
add
delete
edit
assignments
Etienne
Chris
Sandrine
Empowered in: extern in organization city_hospital
Instanciate the following classes: user_password_class
Attribute
Value
adorbacUserPasswordHash
null
No current AdOrBAC user: AdOrBAC policy disabled
24 abstract conflicts | 0 concrete conflict | 26 cpe, 7 ape | 27 cpr, 3 apr | 3 co, 1ao | 48 acp, 1aap