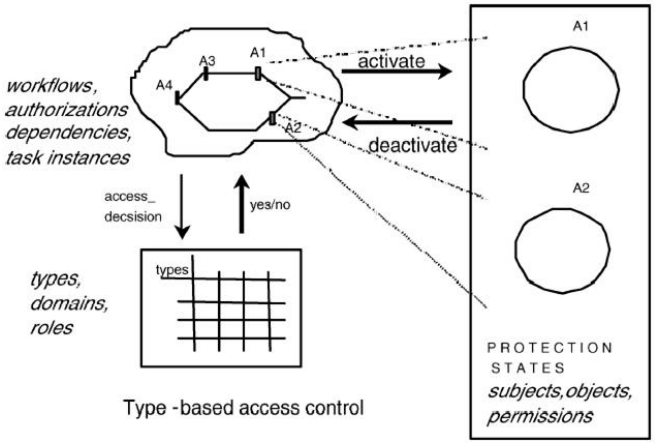
Controlul accesului in sisteme colaborative

TBAC – Task Based Access Control

- Extinde modelele traditionale (subiect-obiect-permisiuni) prin introducerea informatiei de **context** legata de taskuri
- Fiecare task (pas de prelucrare) are asociata o **Stare de protectie** (setul de permisiuni)
- Permisiunile sunt activate si dezactivate in functie de context - conform evolutiei task-urilor (model de securitate activa)

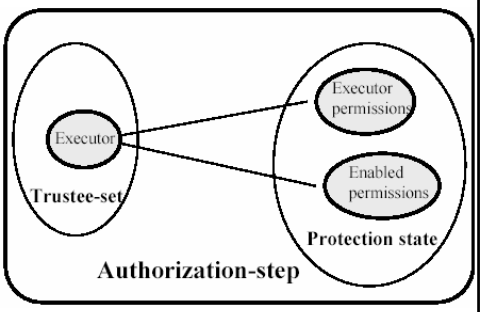


Type -based access control

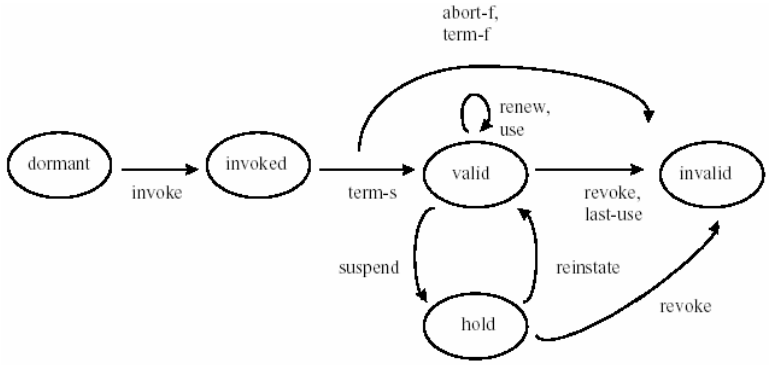
TBAC – Task Based Access Control

- Inspirat din autorizarea in "lumea" documentelor
  - Autorizarea (semnarea unui document) permite derularea unor actiuni
  - Garantul isi asuma responsabilitatea actiunilor
  - Autorizarea se da pe un timp limitat
  - Rezulta conceptul de pas de autorizare - authorization-step
- Exista un set de garanți potentiali - Trustee set
- Unul din garanți - Executor-trustee face autorizarea într-o instanță de *authorization-step*

- Executor permissions – permisiunile cerute executorului pentru a putea face garantarea
- Enabled permissions – permisiunile activate în *authorization-step*



Pasii de procesare pentru un authorization-step



Ciclu de viata cu mai multe stari:

- un pas de autorizare este invocat
- daca prelucrarea lui se termina cu succes devine valid, altfel invalid
- valid – permisiunile sunt activate si pot fi "consumate"
- permisiunile pot fi suspendate temporar (hold)
- la consumarea permisiunilor (sfarsitul ciclului de viata) pasul de autorizare devine invalid

Caracteristici TBAC

Recunoaste notiunea de ciclu de viata si pasii de procesare a autorizarilor

Introduce notiunea de consum asociata cu permisiunile (limitarea numarului de accese)

Formeaza baza auto-administrarii modelului de securitate (administrare cuplata cu activarea si terminarea taskurilor)

- Probleme
  - contextul este limitat la relatia cu pasii taskului
  - dificultati in specificarea politicilor complexe, a delegarii si revocarii privilegiilor

TMAC – Team Based Access Control

- Centrat pe notiunea de echipa – colectie de utilizatori cu roluri diferite dar care au ca obiectiv comun indeplinirea unui task.
- Permisiunile unui subiect depind de
  - rolul sau
  - grupul caruia ii apartine
- Permisiunile sunt
  - asignate unui rol (ca in RBAC) - model de securitate pasiva
  - activate / dezactivate in functie de context - model de securitate activa

Un scenariu

pacientul vine la camera de garda

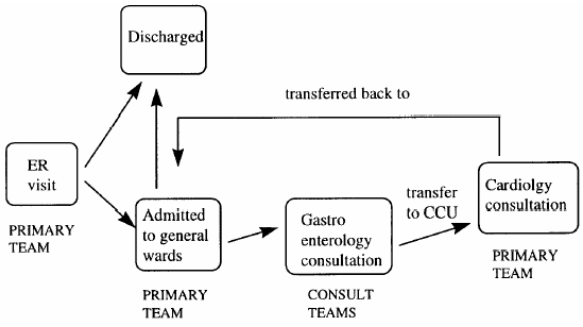
este internat la medicina generala

echipa decide sa faca o analiza la gastro-enterologie

in timpul analizei are un atac de cord si este transferat la sectia de cardiologie

dupa tratament revine la medicina generala

pacientul se recupereaza si este eliberat din spital



personalul este organizat in echipe care ingrijesc bolnavul in diverse etape

fiecare echipa este asociata unui departament

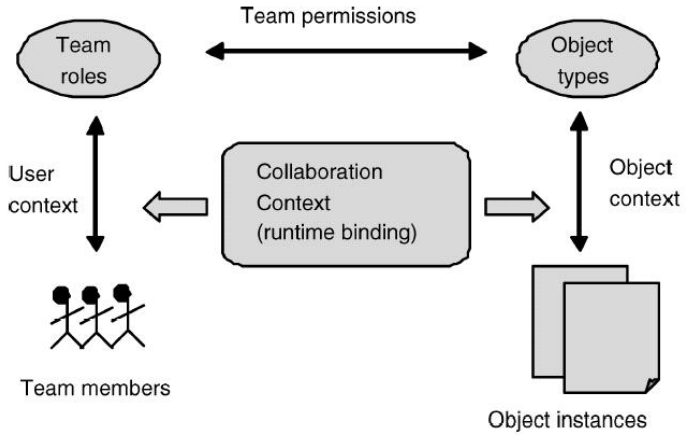
membrii echipelor au roluri diferite

echipele sunt formate dinamic (consultatie gastro)

Cerinte pentru controlul accesului

1. permisiunile asociate unui membru reflecta rolul sau
   - cardiologul poate prescrie medicatia pentru atac de cord
2. permisiunile se refera la un anumit obiect (pacient)
   - doar membrii echipelor de ingrijire a acestui pacient pot avea acces la fisele sale
   - recomandarea pentru gastro o da medicul din echipa acestui pacient
3. accesul este permis in functie de context
   - ex. traseul pacientului - echipa de generalisti are acces la fise doar cand pacientul a trecut la medicina generala
4. activare colectiva a permisiunilor la executie
   - cand pacientul trece la o alta unitate din spital, noua echipa are acces la fisa lui, nimeni altcineva
5. membrii pot delega permisiunile
   - medicul delega un rezident sa comande un medicament pentru pacient
6. permisiunile pot fi dezactivate
   - cand pacientul paraseste spitalul, permisiunile de acces la fisele sale inceteaza

TMAC – Team Based Access Control



RBAC nu surprinde instante de roluri diferite care colaboreaza intr-un grup

Echipa este o abordare mai naturala de a grupa membrii unei organizatii si de a asocia contextul de colaborare

- User context – utilizatori (specifici) care lucreaza in echipa
- Object context – obiecte (specifice) folosite in colaborare

Principalele idei TMAC

O echipa are:

un nume, t.

un set de membri / team users, TU.

un set de team roles, TR $\subseteq$ R (R = setul total de roluri)

un rol special team head (h), h € TR

un set de tipuri de obiecte, OT

un set de instante de obiecte, O

un set de team permissions TP, TP $\subseteq$ TR x OT

un context de colaborare cu doua componente

- user context (UC), UC : TR x TU
- object context (OC), OC : OT x O

Operational

Primitive

User-assign (user, team): asigneaza user la team.

User-deassign (user, team): inlatura user din team.

Team-activate (team): leaga team permissions la team members si la obiecte (TU si O).

Team-deactivate (team): dezactiveaza permisiunile pentru team.

De asemenea

User-activate (user): activare permisiuni individual

User-deactivate (user): dezactivare permisiuni individual

Primitivele invocate odata cu instantele din workflow

Contextul obiectelor se transmite de la o echipa la alta

Caracteristici TMAC

- Calitati
  - avantajele RBAC plus control orientat pe instante de subiecte si obiecte (nu doar tipuri)
  - Context-based TMAC (C-TMAC) include info de context ca timp, loc etc
- Probleme
  - extinde RBAC cu notiunea de echipa care nu este integral elaborata
  - nu are suport de auto-administrare relatii intre obiecte
  - aplicabilitate neverificata; sugestie medii hipermedia

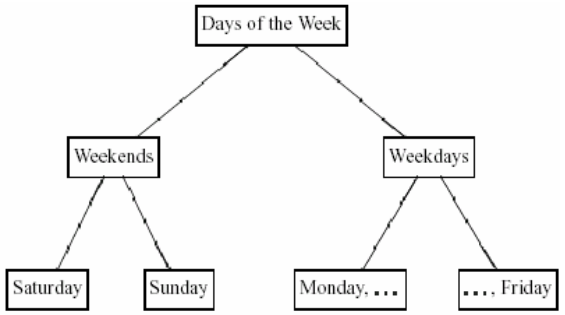
SPACE - Spatial Access Control

- Considera mediul colaborarii si ascunde fata de utilizator mecanismele explicite de securitate
- Modelul are doua componente
  - o granita
    - imparte mediul colaborativ in regiuni; se tine evidenta traversarior si prezentei in regiuni
    - se folosesc credentiale pentru control acces intr-o regiune
  - un graf de acces
    - specifica constrangerile asupra miscarilor in spatiul de colaborare
    - gestiunea cerintelor de acces
- Probleme
  - nu suporta control "fine grain"
  - siguranta nu e demonstrata (se pot crea regiuni nesigure)
  - aplicatia trebuie sa se potriveasca modelului cu regiuni

Context-Aware Access Control

- Control acces in functie de conditiile de mediu
  - timp: acces permis intr-o anumita perioada weekend, weekday
  - loc: accesul la un sistem securizat doar daca se face dintr-o incapere securizata fizic
  - incarcarea unui sistem: rol high load activat la peste 70%
- Extinde RBAC cu environment roles care surprind starea mediului
- sunt folosite pentru a lua decizii asupra accesului
  - Ex. pentru a media o cerere de acces la un sistem, acesta consulta rolurile active (high load, weekday,…)

- rolurile sunt activate pe baza conditiilor de mediu
- revocarea se face cand conditiile nu mai corespund rolului
- evitare conflicte (separarea intereselor)
  - uneori se doreste activarea unui singur rol la un moment dat
- ierarhia de roluri



Reguli de Control al accesului

- intr-o sesiune, pot fi activate mai multe roluri pentru un utilizator
- unele Environment Roles pot fi activate pe durata unei sesiuni, dar schimbarea conditiilor reclama evaluarea altor roluri
- o cerere de permisiune poate fi acordata daca:
  - permisiunea este in setul de permisiuni asociate rolului
  - rolul asociat cererii este in setul de roluri activate pentru utilizatorul care face cererea
  - nu sunt activate Environment roles care interzic acordarea permisiunii

Modelul

Din RBAC0:

se pastreaza U, R, P si S.

capteaza **users**, **roles**, **permissions** si **sessions**.

adauga ER si EC,

ER - *Environment Roles*;

EC - *Environment Conditions* folosite in definirea rolurilor.

Relatii UA, PA

– User Assignment UA $\subseteq$ U x R

– Permission Assignment PA $\subseteq$ P x R x 2^{ER}

- PA asociaza o permisiune cu rolul unui subiect si
- o conditioneaza de un set de environment roles ER active.

- Functii care definesc ce roluri utilizator sau de mediu pot fi activate:
- User: S -> 2^{R}
  – Intr-o sesiune data S, pot fi activate anumite roluri pentru utilizator.
- Request: EC -> 2^{ER}
  – Pe baza environmental conditions, un set de environment roles sunt activate la momentul cererii.
- O cerere de permisiune *p poate fi* garantata daca:
  (1) <*p, r, e-set*> ϵ PA
  (2) Rolul r este in setul de roluri active ale utilizatorului care face cererea
  (3) environment roles active in conditiile environmentale curente *EC* contin rolurile din *e-set*.