

Securitatea sistemelor distribuite

Securitatea în sisteme distribuite

Securitatea — aspecte tratate în acest curs

- **comunicației** — între utilizatori și/sau procese situate în calculatoare diferite
- **accesului** — un utilizator/proces utilizează doar serviciile/resursele la care are dreptul (autorizare)
- **gestiunea** securității (cheilor, utilizatorilor, grupurilor)

Sistemele distribuite sunt mai **vulnerabile**

- control descentralizat
- accesul se face de la distanță
- tot mai multe aplicații critice (Internet banking) sunt distribuite

Securitatea

Probleme de securitate (ce inseamna securitatea?)

- Confidentialitate
- Integritate - acopera
 - integritatea datelor
 - integritatea originii (autentificarea)
- Disponibilitate

Tipuri de amenintari

Atac	Canal	Obiect
Inspectie	Citire continut mesaj	Citire date continute de obiect
Fabricare	Inserare mesaj	Parodiare obiect
Modificare	Schimbare mesaj	Schimbare date incapsulate
Intrerupere	Prevenirea transferului	Denial of service

13.0 Information Policies

13.2 Policy on the Use of Information Technology Resources

Information technology policies ensure that everyone's use of the Institute's information technology resources supports its educational, research, public service, and administrative mission in the best possible way. Effective support of the Institute's mission requires complying with relevant legal, contractual, professional, and policy obligations whenever information technology resources are used. Effective support also means that individuals not interfere with the appropriate uses of information technology resources by others.

This policy broadly covers all of the Institute's information technology resources – hardware, software, and content; this includes but is not limited to electronic networks, systems, computers, devices, telephones, software, data, files, and all content residing in any of these (referred to as "IT resources"). This policy applies to all records of the Institute and to the information in those records, regardless of the form or the location.

13.2.1 Privacy and Confidentiality of Institute Records

All members of the MIT community are responsible for ensuring that their handling of information about individuals is consistent with the Institute's policy on privacy of personal information (see [Section 11.2 Use of Personal Information](#)). In addition, other Institute records (that is, records that do not contain personal information) must be handled with due regard for privacy and confidentiality concerns. (See [Section 13.2.2.2 Security of Information](#) and [13.2.4 Privacy of Electronic Communications, Electronic Files, and Other Files](#)).

13.2.2 Information Preservation and Security

13.2.2.1 Preservation of Information

MIT has an obligation to provide accurate, reliable information to authorized recipients and to preserve vital records (see [Section 13.3 Archival Policy](#)). MIT is increasingly dependent on the accuracy, availability, and accessibility of information stored electronically and on the computing and networking resources that store, process, and transmit this information. Records created and maintained in electronic form are included in the Institute's definition of archival materials. In addition, upon direction from the Office of the General Counsel, records must sometimes be preserved for prescribed periods of time for litigation or other legal purposes.

13.2.2.2 Security of Information

Individuals who manage or use IT resources required by the Institute to carry out its mission must take reasonable steps to protect them from unauthorized modification, disclosure, and destruction. Data and software are to be protected, regardless of the form, medium, or storage location of the information. The level of protection shall be commensurate with the risk of exposure and with the value of the information and of the IT resources.

Some information has additional legal protection, like certain medical information, education records (see [Section 11.3 Privacy of Student Records](#)), certain financial records, and specific categories of personal information covered in [MIT's Written Information Security Program](#). As described in the Written Information Security Program, departments that regularly use specified categories of personal information should have written procedures on

protecting that data, and should also implement specific procedures concerning how that data is destroyed when no longer needed.

13.2.3 Responsible Use of IT Resources

13.2.3.1 Approved Use of IT Resources

All members of the MIT community are obligated to use MIT's IT resources in accordance with applicable laws, with Institute policies (including its policy against harassment, and its standards of honesty and personal conduct), and in ways that are responsible, ethical, and professional. Users of MIT's network must also comply with the [MITnet Rules of Use](#).

The use of MIT's IT resources is restricted to Institute business and incidental personal use. Incidental personal use may not interfere with MIT work, nor may it result in additional direct cost to MIT. MIT's computers and other IT resources must be used in a manner consistent with MIT's status as a non-profit organization, and so, for example, cannot be used for the benefit of personal businesses or other organizations unless permitted by MIT policy (for example, permitted under [Section 4.5 Outside Professional Activities](#)) or otherwise authorized. Unauthorized access to and use of MIT's IT resources violates this policy.

13.2.3.2 Interference with IT Resources

Members of the Institute community should not take unauthorized actions to interfere with, disrupt, or alter the integrity of MIT's IT resources. Efforts to restrict or deny access by legitimate users of the Institute's IT resources are unacceptable. Individuals should not use MIT facilities to interfere with or alter the integrity of any IT resources, irrespective of their location.

Destruction, alteration, or disclosure without authorization of data, programs, or other content that belongs to others but that is accessed through MIT's IT resources is also prohibited. MIT may block an individual or group's access to its IT resources in order to protect its IT resources and the information contained in them.

13.2.4 Privacy of Electronic Communications, Electronic Files, and Other Files

As noted in [Section 13.2.2.2 Security of Information](#), members of the MIT community should exercise caution to protect information (and particularly personal information) from unauthorized disclosure. Particular caution should be used with electronic communications, because of the ease with which such communications can be distributed and due to concerns about unauthorized access. Unauthorized interception of email and other electronic communications is prohibited by MIT policy and may also violate state and federal law.

For legitimate business reasons, representatives of the Institute may need to access electronic or other records (including paper files) without the consent of the individuals having custody of them; examples of these business reasons include access required by law, where the individual is unavailable due to illness, in the course of an investigation, or in cases of alleged misconduct. Departments, labs, or centers may determine additional reasons for access, for example, due to sponsor requirements (as at Lincoln Laboratory). Any member of the MIT community who accesses information from records maintained by another individual without the individual's consent must seek prior approval from the applicable [Senior Officer](#) or his or her designee for such access and related disclosure; the Senior Officer or designee may consult the Office of the General Counsel. This process applies to requests for access from an outside entity or from another office within MIT.

13.2.5 Third-Party Products and Services

13.2.5.1 Restrictions on Use of Certain IT Resources from Outside Sources

Special restrictions are often placed on the use of IT resources — such as hardware, software, databases, and documentation — acquired from outside sources. Use of such IT resources may be further restricted by patent law, as a trade secret, or by contract in the form of a license or other agreement. Members of the MIT community are required to abide by the restrictions imposed by law or by contract on IT Resources acquired for use at the Institute. Any individual who arranges for authorized distribution of information technology products and services from outside sources must advise the people having access to the products and services of all the associated usage restrictions.

13.2.5.2 Copyright

Unless it has been placed in the public domain, most third-party software is protected by copyright law and may be subject to restrictions on use, copying, and distribution. More information on copyright can be found at [Section 13.5 Reproduction of Copyrighted Materials](#).

Construire sistem sigur - etape

1. Specificarea politicii de securitate (cerinte)

Politica - Ce actiuni sunt permise unei entitati (utilizator, serviciu, masina, proces etc.)

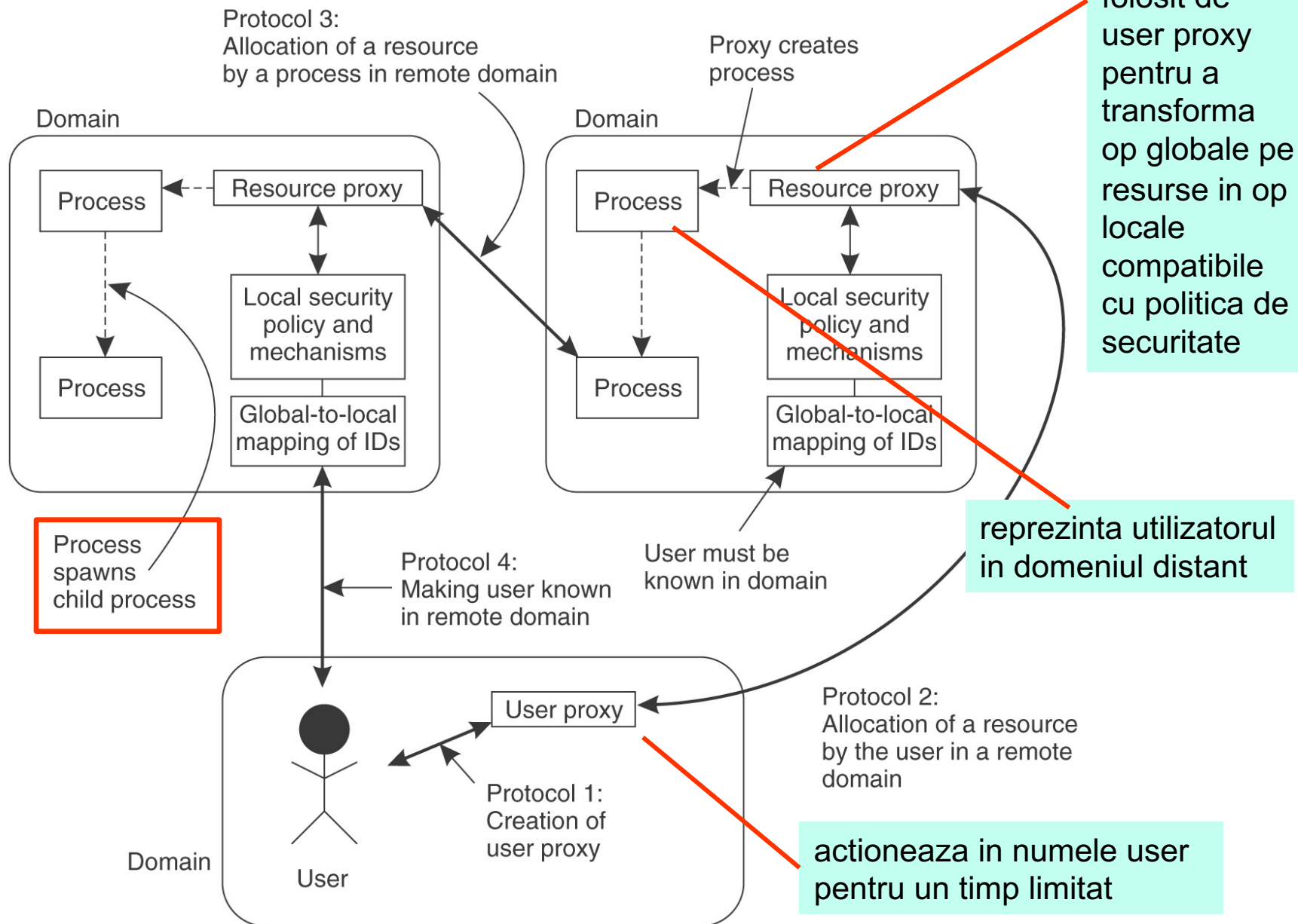
Exemplu: Globus

- Exista **mai multe domenii administrative** cu politici diferite
- **Operatii locale** – conform politicii locale
- **Operatii globale** - controlul accesului este subiectul securitatii locale
 - identificatorul initiatorului sa fie **cunoscut local**
 - **autentificarea globala** tine loc si de autentificare locala
 - cer **autentificare mutuala**
 - entitatile pot **delega privilegii** proceselor
- Pentru procese din acelasi domeniu si actionand in contul aceluiasi utilizator - **credentialele** pot fi **partajate** in grup

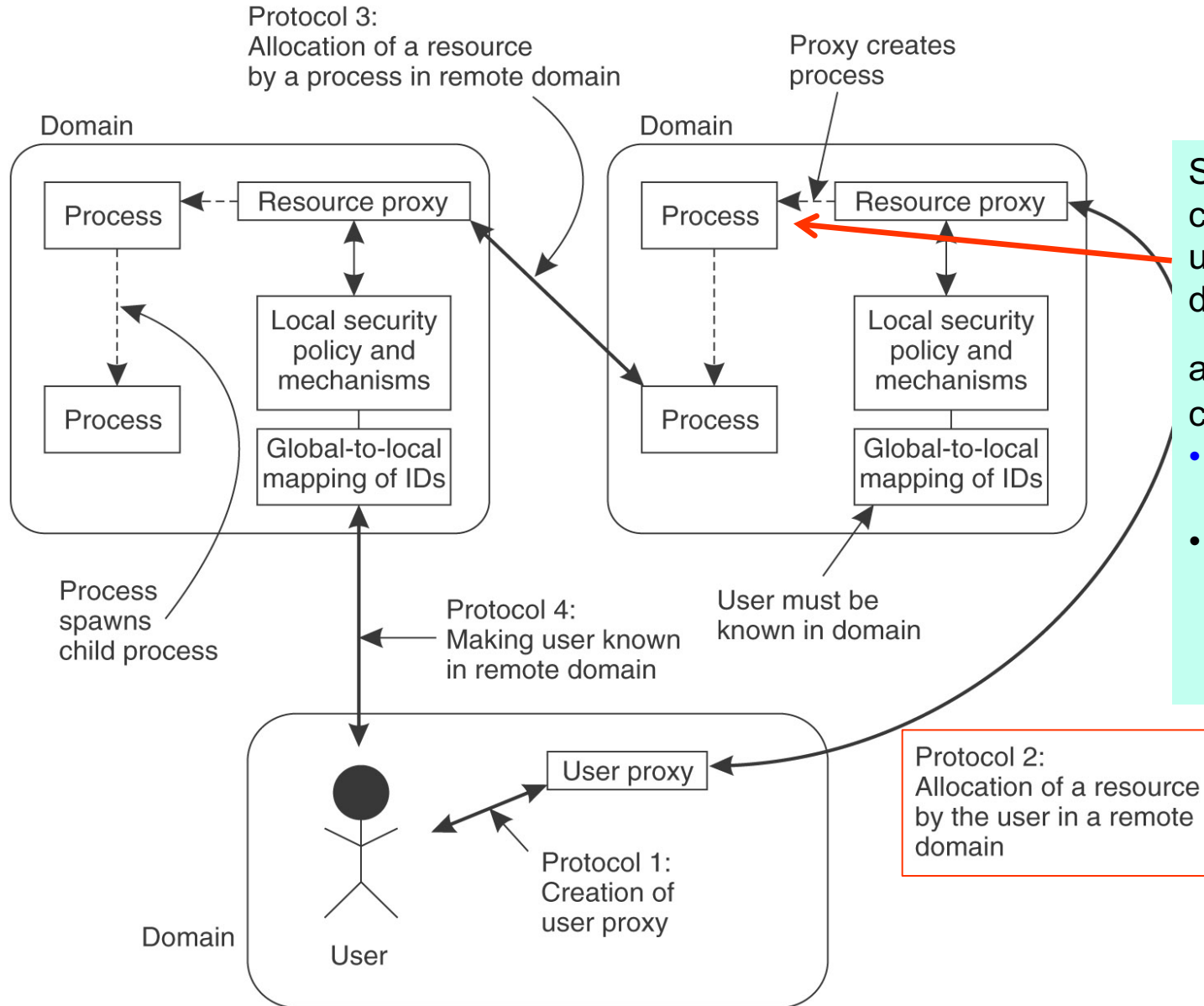
2. Alegerea mecanismelor de securitate

- Focus pe operatii globale
- **Ex.** Globus
 - Reprezentarea unui utilizator intr-un domeniu la distanta
 - Alocarea resurselor dintr-un domeniu la distanta, utiliz. sau proxy

3. Implementarea (entitati si protocoale)



Protocolul 2 – alocare resurse



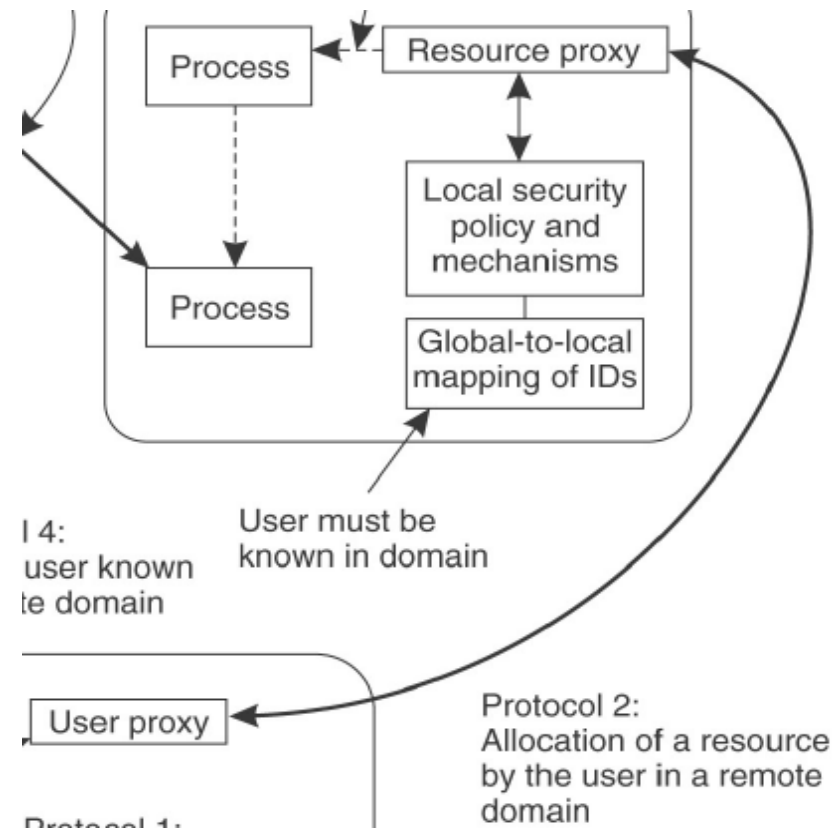
Se **creaza un proces** care reprezinta utilizatorul in domeniul distant;

acces la resursa conform

- **deciziilor locale** de acces
- si unor **credentiale (Cp)** capatate prin protocolul 2.

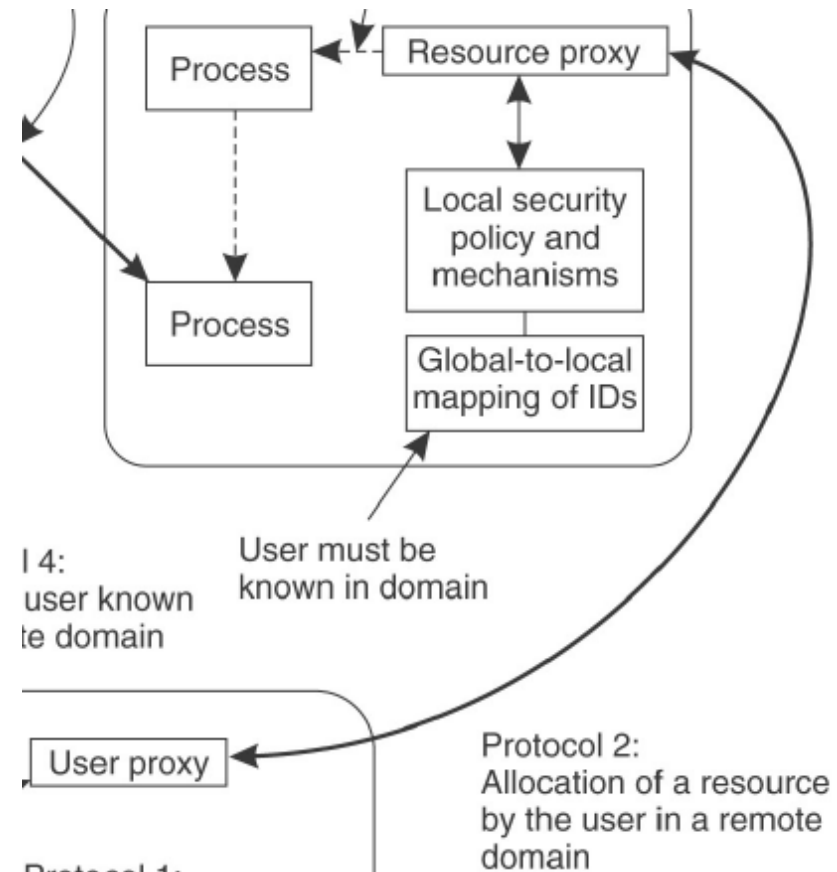
Protocolul 2 continuare

1. user proxy - **UP** si resource proxy - **RP** se **autentifica** reciproc folosind certificate **Cup** and **Crp**. **RP** verifica daca credentialele **UP** nu au expirat.
2. **UP** trimite **RP** o **cerere de alocare** semnata SigUP {specificatie alocare}.
3. **RP** verifica daca **UP** este **autorizat** local sa faca alocarea resursei.
4. Daca da, **RP** creaza un tuplu **RESOURCE-CREDENTIALS** continand numele utilizatorului, numele resursei etc.



Protocolul 2 – continuare

5. **RP** paseaza securizat **RESOURCE-CREDENTIALS** lui **UP**.
6. **UP** examineaza **RESOURCE-CREDENTIALS** si, daca o aproba, **semneaza tuplul** producand **Cp**, credentialele pentru resursa ceruta.
7. **UP** paseaza securizat **Cp** lui **RP**.
8. **RP** **aloca resursa** si paseaza **Cp** noului **proces**. (acest transfer tine cont ca **RP** si **procesul** sunt in acelasi domeniu de incredere.)



Trusted Computer Base (TCB)

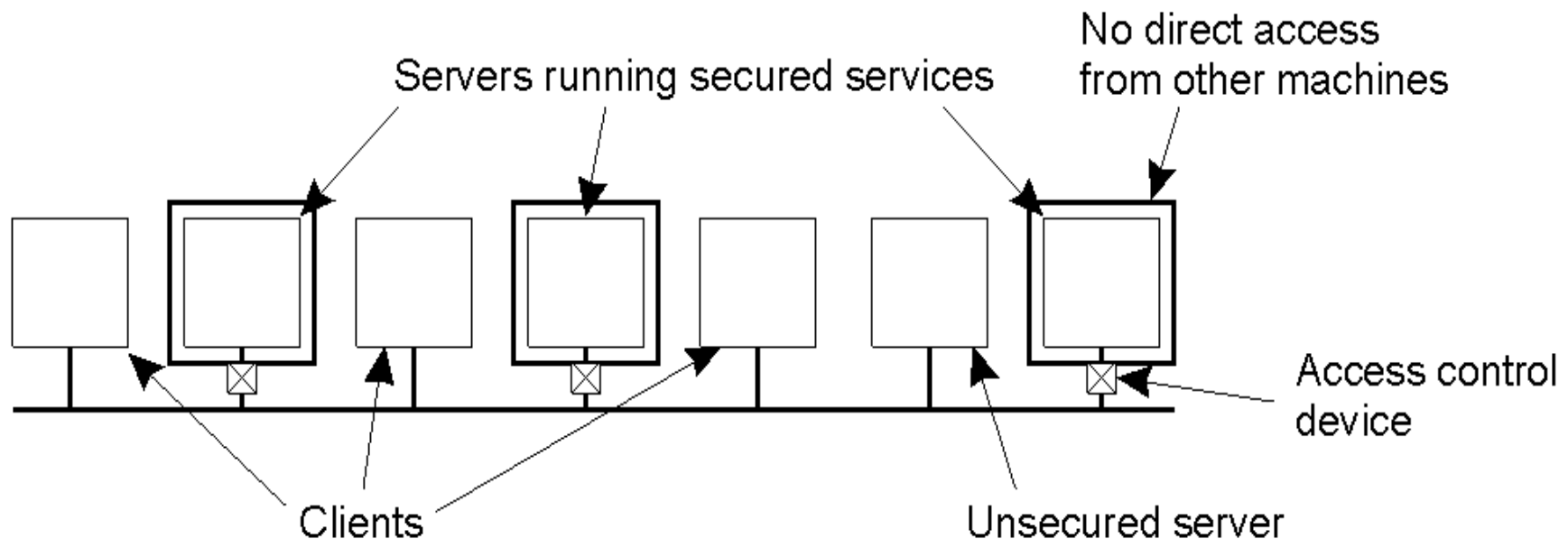
Totalitatea componentelor și mecanismelor software și hardware cerute pentru a implementa o politică de securitate

Principiul RISSC (Reduced Interfaces for Secure System Components) aplicat pentru securizarea sistemelor distribuite.

- orice server cu **cerinte critice** de securitate este plasat pe o **masina separata**, izolat de sistemele utilizatorului
- clientii și aplicațiile pot accesa serverul doar prin **interfete de rețea sigure**.

Soluție similară: **Containerele** - cresc gradul de securitate prin **izolare**

- ex. SELinux și CoreOS; Docker



4. *Asigurare* ca mecanismele functioneaza corect

Metode formale

- aplicare in **toate etapele**: specificare, proiectare, implementare

Increderea (trust) - convingerea ca entitatea realizeaza politica de securitate pentru care a fost conceputa

- caracteristica generala legata de **perceptia** subiectului: exista mai multe **grade de incredere**
- prin contrast, **securitatea** este **caracteristica binara** a obiectului, care poate fi sigur sau nu

Pentru credibilitate, evaluarea (assessment) produselor se face

- in **Conformitate cu standardele** (ex. ITSEC – Information Technology Security Evaluation Criteria)
- de catre **experti neutri**

Criterii de evaluare

TCSEC - Trusted Computer System Evaluation Criteria

- cunoscut ca Orange Book (USA), standard DoD

ITSEC - Information Technology Security Evaluation Criteria

- publicata in 1990 de Franta, Germania, Olanda si UK

Common Criteria - Common Criteria for Information Technology Security Evaluation

- standard ISO/IEC 15408 din 2005
- derivat din ITSEC, TCSEC, CTCPEC (standard Canadian)
- revizuit in 2009 (ISO/IEC 15408-1:2009) si confirmat in 2015

Common Criteria

Clasifica produsele IT dupa nivelul de incredere (**Evaluation Assurance Level**, EAL1 - EAL7)

- EAL1: Functionally Tested
- EAL2: Structurally Tested
- EAL3: Methodically Tested and Checked
- EAL4: Methodically Designed, Tested, and Reviewed
- EAL5: Semiformally Designed and Tested
- EAL6: Semiformally Verified Design and Tested
- EAL7: Formally Verified Design and Tested

vezi: http://en.wikipedia.org/wiki/Evaluation_Assurance_Level

Autentificarea

Protocoale analizate

RADIUS - Remote Authentication Dial In User Service

- Protocol de Autentificare, Autorizare și Accounting pentru accesul la rețele de pe echipamente mobile (Network Access și IP Mobility)

SECURE RPC

- Protocol pentru Autentificare și Confidentialitate în apelurile de proceduri la distanță

KERBEROS

- Protocol pentru Autentificare, Autorizare și Confidentialitate într-un sistem distribuit

RADIUS - Remote Authentication Dial In User Service

Standard "de facto" pentru autentificarea, autorizare și contabilizare (AAA – Authentication, Authorization, Accounting) pentru utilizatorii care se conectează la- și folosesc **servicii de rețea**

Ulterior, **standard IETF** (Internet Engineering Task Force)

Folosit de ISP-uri și întreprinderi pentru gestiunea accesului la Internet și rețele private, rețele wireless, servicii e-mail integrate etc.

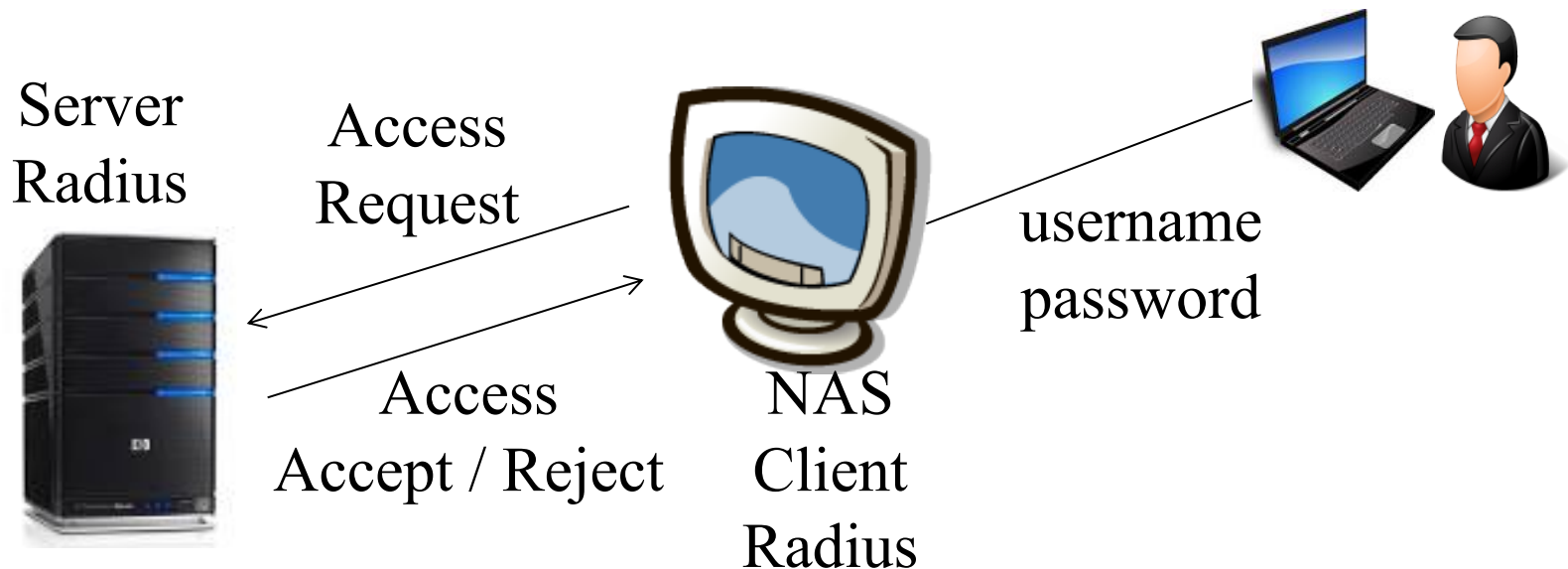
Folosit la distanță pe legături telco (de ex pentru mobilitate IP).

Contabilizează **resursele utilizate** într-o sesiune (timp, pachete ...)

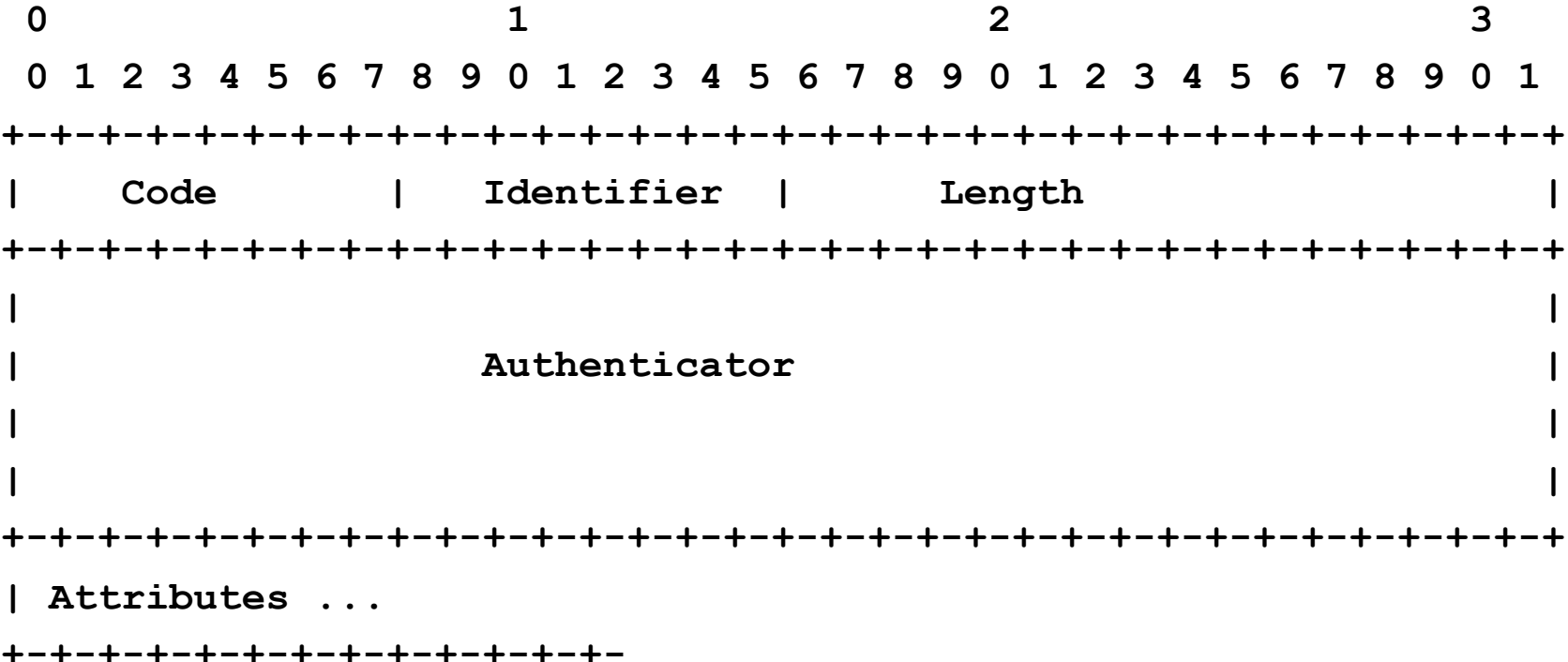
RADIUS

Protocol

1. Utilizatorul **intra** in retea prin modem.
2. **RADIUS client** (NAS – Network Access Server) cere username si password.
3. RADIUS client trimite **Access Request** (codificat MD5).
4. RADIUS server **autentifica** utilizatorul si trimite **Access Accept** sau **Access Reject** la RADIUS client.



Pachetul RADIUS



Code	Description	Code	Description
1	Access-Request	2	Access-Accept
3	Access-Reject	4	Accounting-Request
5	Accounting-Response	11	Access-Challenge
12	Status-Server (experimental)	13	Status-Client (experimental)
255	Reserved		

Actiuni client

Creaza pachet **Access-Request** care include cel putin atributele **User-Name** si **User-Password**.

Identifier—ul cererii este generat de client

Pachetul Access-Request contine **Request Authenticator**, RA (16 octeti alesi aleator).

Doar User-Password protejat:

- Client si server partajeaza un **secret S**.
- **Password** impartit in blocuri de 16-octeti p_1, p_2, \dots, p_n (cu ultimul bloc completat la 16 octeti)
- Blocurile de text cifrat sunt obtinute astfel:
$$c_1 = p_1 \text{ XOR MD5}(S + \text{RA})$$
$$c_2 = p_2 \text{ XOR MD5}(S + c_1)$$
$$\vdots$$
$$\vdots$$
$$c_n = p_n \text{ XOR MD5}(S + c_{n-1})$$

Atributul **User-Password** contine **$c_1+c_2+\dots+c_n$**
(+ denota concatenarea).

Actiuni server plus client

Server

Primește pachet RADIUS **Access-Request**.

Verifica dacă are **secret** partajat cu clientul (dacă nu, ignora).

Obține **password** neprotejat (procedura similară client).

Folosește BD autentificare pentru validare **username** și **password**.

- password valid => creează pachet **Access-Accept**
- password invalid => creează **Access-Reject**.
- Adaugă **Response Authenticator** în câmp Authenticator.

ResponseAuth =

MD5(Code+ID+Length+RequestAuth+Attributes+Secret)

(+ denota concatenarea).

Client

Gaseste cererea corespunzătoare răspunsului (cf. Identifier)

Calculează ResponseAuth (repetând operații server) și compară rezultat cu câmp din răspuns) – ignora dacă nu corespunde.

Secure RPC

Dezvoltat de Sun pentru sistemul de operare.

Bazat pe o combinatie de chei publice si chei secrete

- DES pentru criptarea datelor
- Diffie-Hellman pentru autentificare

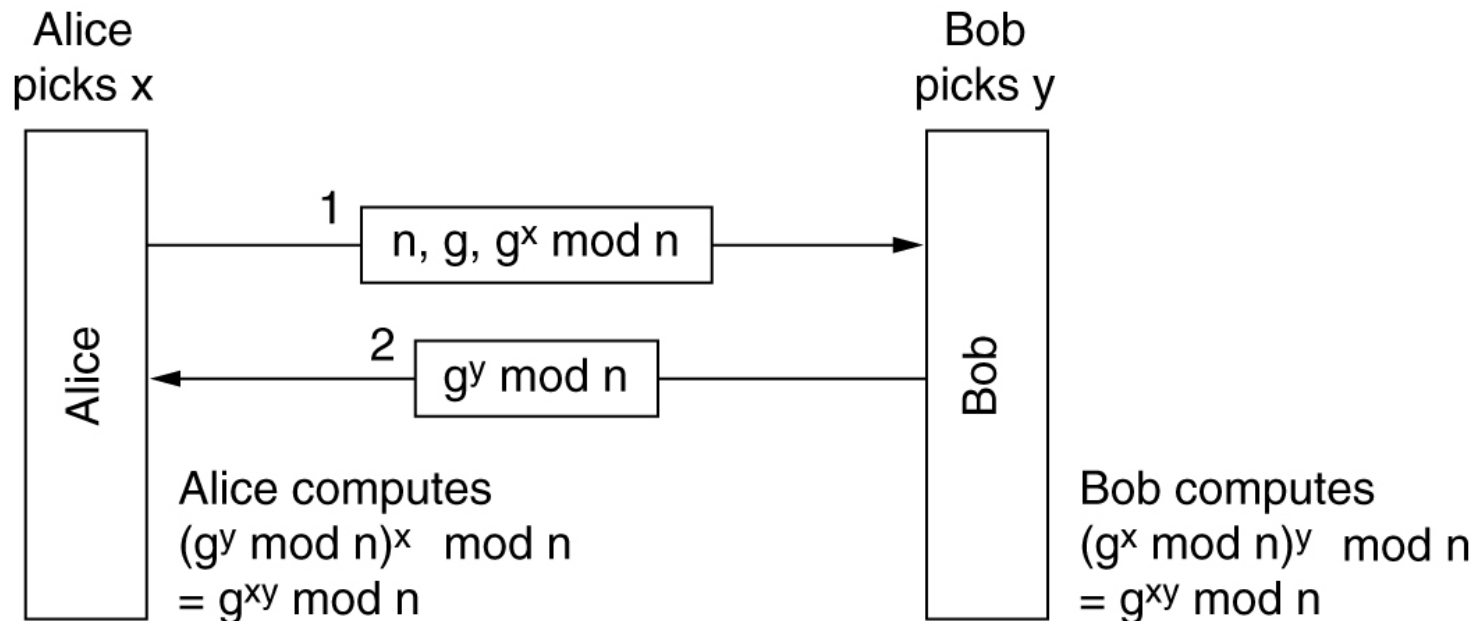
n, g – numere mari

n prim si $(n-1)/2$ prim

g intreg cu proprietatea:

orice p intre 1 si $n-1$ inclusiv poate fi scris ca $p = g^k \bmod n$.

x nu poate fi calculat din $g^x \bmod n$



Secure RPC - caracteristici

Fiecare **principal** (utilizator, calculator) are o pereche de chei (pastrate într-o BD împreună cu numele utilizatorului):

- **Publica**, pastrată în clar
- **Secreta**, pastrată în forma criptată DES cu password-ul principal-ului.

Un principal:

- Își probează identitatea arătând că poate decripta cheia secretă criptată
- Combina cheia secretă cu cheia publică a partenerului, ambii ajungând independent la o **cheie de sesiune (SK)** comună, cunoscută
- Această cheie este folosită pentru a stabili o **cheie de conversație (CK)**.

Secure RPC – Local Login

$U \rightarrow C: U, PW$

- C:
1. Regaseste din baza de date publica de chei o inregistrare **user** continand: **username**, cheia publica user, {cheia secreta user}**PW**
 2. Decripteaza cheia secreta folosind **PW** si memoreaza cheia secreta in procesul **keyserver**

Unde:

U	user
C	client
PW	password

Protocol folosit ptr autentificare client

1. C:
 1. Primește cheia publică **server** de la baza de date publică de chei
 2. Generează **session key** $(SK)_{c,s}$ pentru utilizare între client și server
2. C → S: trimite C, $\{CK\}_{SK(c,s)}, \{window\}_{CK}, \{t_1, window+1\}_{CK}$
3. S:
 1. Primește cheia publică a clientului de la BD publică de chei
 2. Generează cheia de sesiune $(SK)_{c,s}$ folosită între client și server
 3. Decriptează cheia de conversație CK folosind SK
 4. Decriptează t_1 , window și window+1
 5. Memorează într-o tabelă de credențiale, cu indexul ID: C, CK, window, t_1
4. S → C: $\{t_1 - 1\}_{CK}, ID$
5. C: Memorează ID și CK în procesul key server

Unde:

C / S	client / server
CK	cheia de conversație
window	timpul de viață al cheii CK

SK	cheia secretă generată de client și server
t_1	amprenta de timp originală

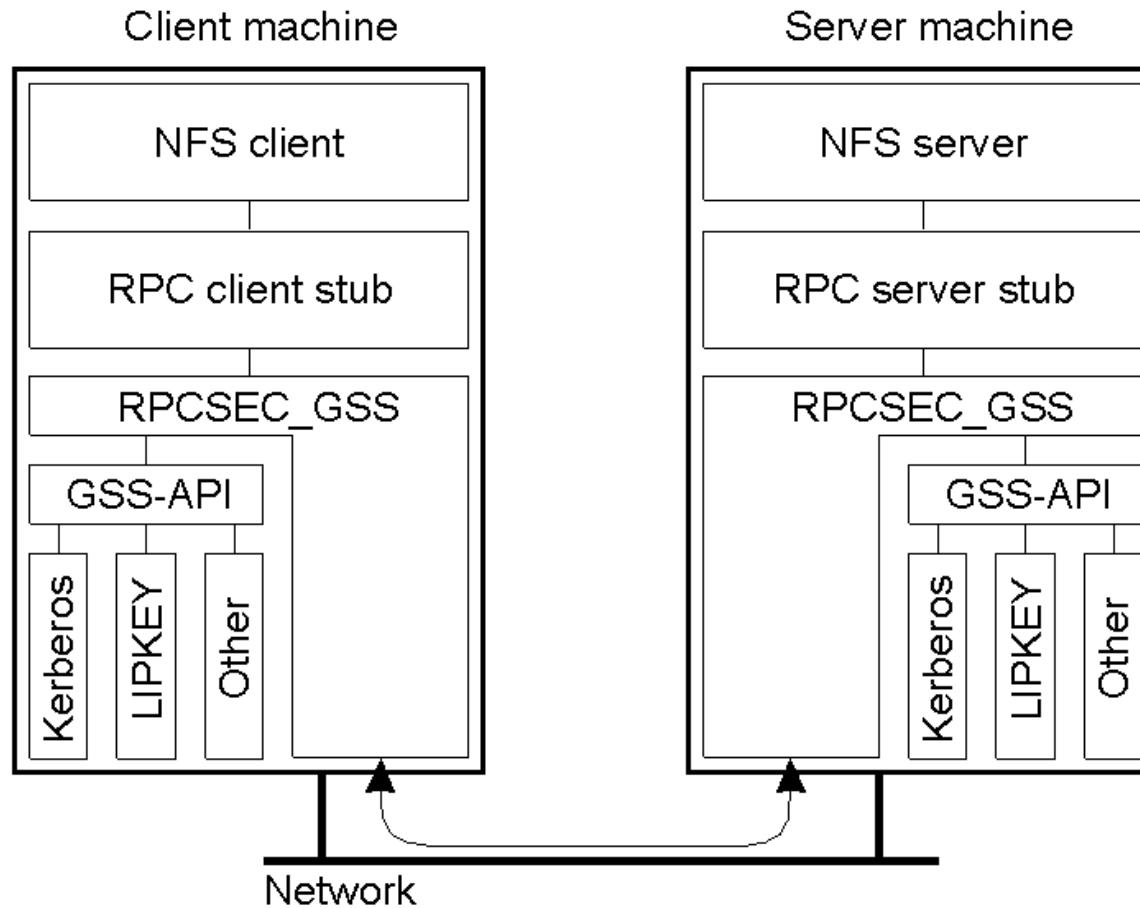
Tratare cereri dupa autentificare

6. $C \rightarrow S$: $ID, \{t_n\}_{CK}$
7. $S \rightarrow C$: $\{t_n - 1\}_{CK}, ID$

Unde:

- C client
- S server
- CK cheia de conversatie
- ID index client
- t_n amprenta de timp curenta

Secure RPC in NFS versiunea 4



GSS – Generic Security Service

Incorporeaza cai standard de securitate (Kerberos, LIPKEY - Low Infrastructure Public Key)

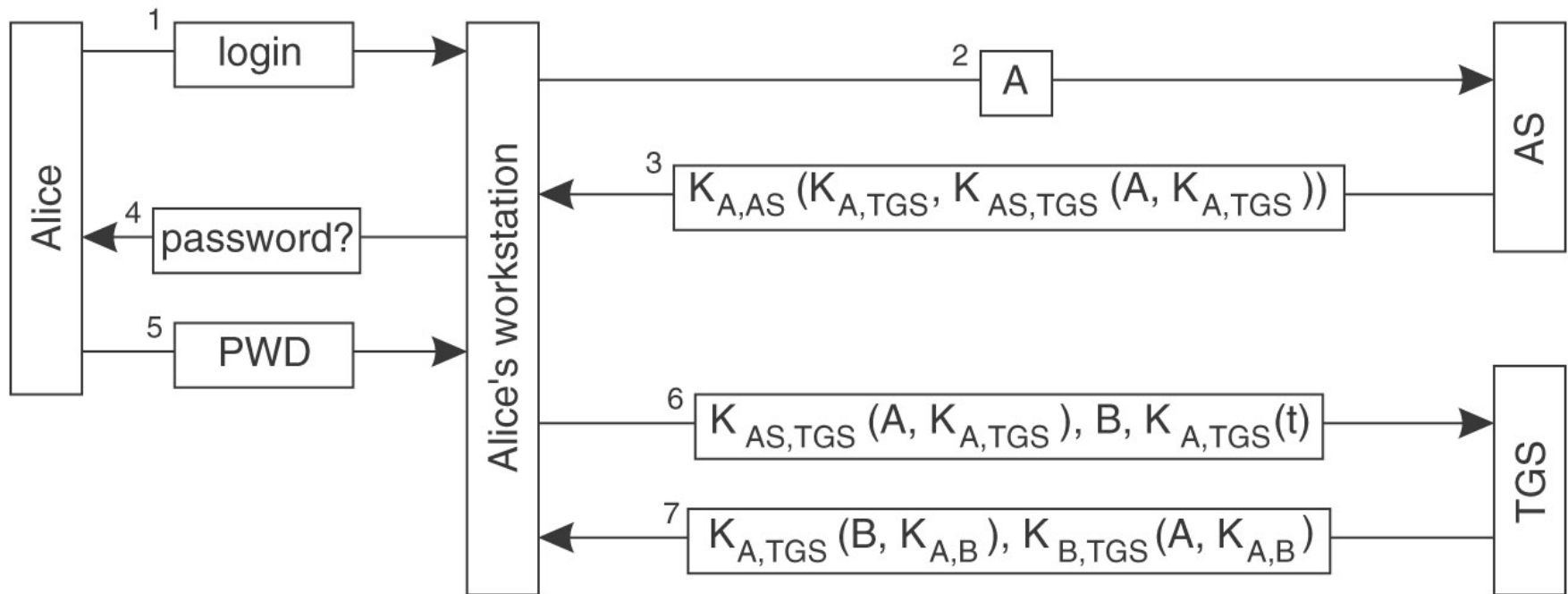
Kerberos

Protocol autentificare dezvoltat în proiectul
Athena la **MIT**

Sistem de securitate care ofera:

- Autentificare
- Autorizare
- Confidentialitate mesaje

Autentificarea in Kerberos



AS – Authentication Server

TGS – Ticket Granting Server

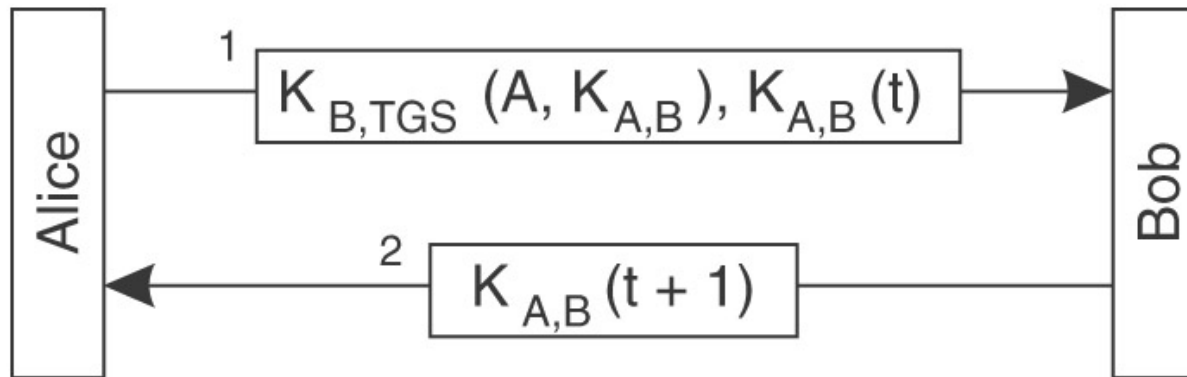
B – serviciu folosit de Alice

A – **login name** Alice

password folosit ptr decriptare mesaj 3

PWD – password

Invocarea serviciului



Securitatea sistemelor distribuite

Partea a 2-a

Controlul Accesului

Bazat pe 2 premise

- **identificarea** corecta a utilizatorului
 - facuta prin **autentificare**
 - nici un utilizator sa nu poata lua drepturile de acces ale altuia
- **protejarea** informatiei despre drepturile de acces contra modificarilor neautorizate

Controlul Accesului – Elemente de baza

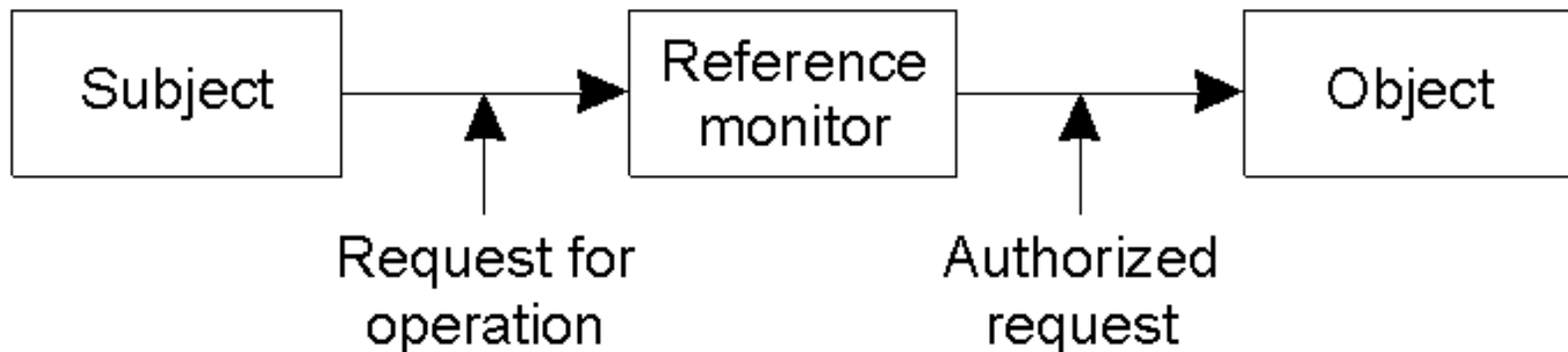
Problema: Are subiectul **s** dreptul de access **a** la obiectul **o**?

- Tuplul (s, o, a) constituie **Autorizatia**

Solutia: Controlul accesului este o functie $f(s, o, a)$ care intoarce *true* sau *false*

Reference monitor implementeaza aceasta functie

- toate cererile senzitive trec prin Reference monitor
- monitorul decide daca operatia poate continua



Clasificarea modelelor de acces

Model de securitate

- o reprezentare mai precisă și mai detaliată a unei **politici** de securitate
- folosit ca referință pentru **construirea** securității și pentru **evaluarea** ei

Clasificare după posibilitatea de **transfer** al drepturilor

- **Discretionare** - Discretionary Access Control (DAC)
 - utilizatorii pot transfera altora drepturile pe care le dețin
- **Mandatorii** - Mandatory Access Control (MAC)
 - utilizatorii nu pot transfera drepturile pe care le dețin

Clasificare după **obiectul** controlat

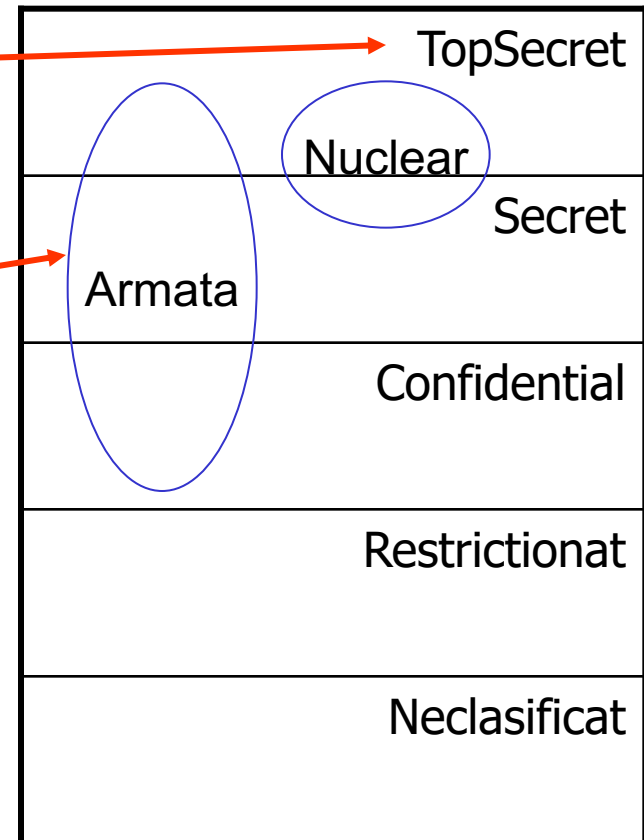
- **fluxul** informațiilor
 - modele multinivel
- **accesul** la informații
 - matrice de acces,
 - RBAC – Role Based Access Control

Model de confidentialitate multinivel

Fiecare **obiect**

- are un **nivel** de senzitivitate sau **rang**: top secret, secret, ...
- este asociat cu unul sau mai multe proiecte (**compartimente**)
 - accesul este permis celor care "trebuie sa stie" pentru ca lucreaza in proiectul respectiv

Clasa de acces a unui obiect este combinatia $\langle \text{rang}, \text{compartimente} \rangle$



TopSecret	Nuclear
Secret	Armata
Confidential	Armata
Restrictionat	
Neclasificat	

Exemple

$C1 = (\text{TopSecret}, \{\text{Nuclear}, \text{Armata}\})$

$C2 = (\text{TopSecret}, \{\text{Nuclear}\})$

$C3 = (\text{Confidential}, \{\text{Armata}\})$

Fiecare **subiect**

- are o **autorizare** (**clearance**) exprimata ca o clasa de acces $\langle \text{rang}, \text{compartimente} \rangle$
 - la ce nivel de senzitivitate are acces
 - in ce proiecte lucreaza

Accesul unui **subiect** la un **obiect** se bazeaza pe relatia de **dominanță** intre clase de acces

- C_i **domina** C_k (sau C_k este dominat de C_i), $C_i \geq C_k \Leftrightarrow$
 $\text{rang}(C_i) \geq \text{rang}(C_k)$ si
 $\text{compartimente}(C_i) \supseteq \text{compartimente}(C_k)$
- C_i **domina strict** C_k , $C_i > C_k \Leftrightarrow$
 $\text{rang}(C_i) > \text{rang}(C_k)$ si
 $\text{compartimente}(C_i) \supset \text{compartimente}(C_k)$
- C_i si C_k sunt **incomparabile** $C_i <> C_k$ daca
nici $C_i \geq C_k$ nici $C_k \geq C_i$

Exemple

Clase de acces

$C1 = (\text{TopSecret}, \{\text{Nuclear}, \text{Armata}\})$

$C2 = (\text{TopSecret}, \{\text{Nuclear}\})$

$C3 = (\text{Confidential}, \{\text{Armata}\})$

$C1 \geq C2$

$C1 > C3$

TopSecret > Confidential si
 $\{\text{Nuclear}, \text{Armata}\} \supset \{\text{Armata}\}$

$C2 < > C3$

Controlul accesului: Exemplu

Rang	Subiect	Obiect
Top secret	Ion, Rodica	Fisiere de personal
Secret	Sanda, Vasile	Fisiere e-mail
Confidential	Costi, Ioana	Fisiere log
Neclasificat	Mara, Mihai	Fisiere numere telefon

Senzitivitatea scade de sus in jos

Un singur compartiment

Cerinta 1 - Subiectii au acces doar la informatia pentru care au clasa de acces necesara

- Ioana nu are acces la Fisiere de personal
- Rodica are acces la toate fisierele

Cerinta 2 – Informatia sa nu se scurga spre clase de acces inferioare

- Rodica **poate citi** Fisiere de personal dar **nu trebuie sa le poata scrie** in Fisiere log pentru accesul Ioanei

Modelul Bell-La Padulla (BLP)

Pastreaza secretul - previne divulgarea neautorizata a informatiei; ex. secrete militare

Securitatea simpla (*no-read-up*)

- Subiectele au acces doar la informatia pentru care au clasa de acces necesara
- Formal: Un subiect s are acces *read* la un obiect o doar cand clasa subiectului *domina* clasa obiectului

$$C(s) \geq C(o)$$

Proprietatea Star (*) (*no-write-down*)

- Previne fluxul de informatie inspre obiecte cu clase de acces inferioare sau incomparabile
- Formal: Un subiect s care are acces *read* la un obiect p , poate avea acces *write* la un obiect o doar cand clasa de acces a lui o domina clasa lui p

$$C(o) \geq C(p)$$

Modelul de integritate Biba

Defineste **nivele de integritate** (I) analoage nivelelor de senzitivitate. Regulile sunt insa diferite.

Integritatea simpla

- Subiectul **s** poate modifica (acces *write*) un obiect **o** doar daca **o** are integritatea mai mica

$$I(s) \geq I(o)$$

- **Justificare**: un subiect **s** cu integritate mai mica ar scadea integritatea obiectului **o** modificat de el

Proprietatea-*

- Daca subiectul **s** are acces *read* la obiectul **p** cu integritatea **I(p)**, atunci el poate avea acces *write* la obiectul **o** doar daca

$$I(p) \geq I(o)$$

- **Justificare**: un obiect **p** cu integritatea mai mica ar scadea integritatea obiectului **o**

Caracteristici comune

Modelele Bell La-Padula si Biba

- nu specifica modul de **definire** sau de **modificare** a claselor de acces si de autorizare
- nu trateaza **delegarea** sau transferul drepturilor de acces

Topicile sunt tratate de alte modele

- ideea generala (Graham-Denning):
 - definirea unei **matrice de acces** care sa specifice drepturile de acces pentru fiecare combinatie de subiecte si obiecte

Exemplu matrice de acces

Drepturile proceselor P1 si P2 asupra
 fișierelor f1 si f2
 proceselor P1 si P2

	f1	f2	P1	P2
P1	read write own	read	read write execute own	write
P2	append	read own	read	read write execute own

Modelul Graham - Denning

Modelul este definit in termeni de stari si tranzitii

- o **stare** este reprezentata de o **matrice** de **drepturi**
- **tranzitiile** intre stari sunt descrise prin **actiuni** executate de subiecti

subiecti $S = \{s_1, \dots, s_n\}$

obiecte $O = \{o_1, \dots, o_m\}$

drepturi $R = \{r_1, \dots, r_k\}$

intrari $A[s, o] \subseteq R$

drepturile din $\{r_1, \dots, r_k\}$ pe care subiectul **s** le are asupra obiectului **o**

Graham-Denning

drepturile sunt definite ca

- **actiuni** $A[s, o]$ pe care subiectul **s** le poate executa asupra obiectului **o**
- **actiuni** $A[s_i, s_j]$ pe care subiectul **si** le poate executa asupra subiectului **sj**

	o_1	\dots	o_m	s_1	\dots	s_n
s_1						
s_2						
\dots						
s_n						

Proprietari si controlori

- fiecare **obiect** are un subiect **proprietar** (owner) cu drepturi speciale
- fiecare **subiect** are un alt subiect cu drepturi speciale (**controlor**)

Modelul propune un set fix de actiuni primitive executate de subiectul x ;

In urmatorul tabel:

- **r** reprezinta un drept
- **r^*** inseamna ca **dreptul r** transmis de **x** lui **s** este *transferabil*, adica **s** poate transfera r sau r^* altor subiecte

Actiuni primitive

Actiune	Preconditie	Efect
<i>create object o</i>	-	Adauga coloana o la A; adauga <i>Owner</i> la A[x,o]
<i>delete object o</i>	<i>Owner</i> in A[x,o]	Sterge coloana o
<i>create subject s</i>	-	Adauga linia s si col s la A; adauga <i>Control</i> la A[x,s]
<i>delete subject s</i>	<i>Control</i> in A[x,s]	Sterge linia s si col s
<i>read access rights of s on o</i>	<i>Control</i> in A[x,s] sau <i>Owner</i> in A[x,o]	Citeste A[s,o]
<i>grant access right r to s on o</i>	<i>Owner</i> in A[x,o]	Adauga r la A[s,o]
<i>delete access right r of s on o</i>	<i>Control</i> in A[x,s] sau <i>Owner</i> in A[x,o]	Sterge r din A[s,o]
<i>transfer right r or r* to s on o</i>	r* in A[x,o]	Adauga r sau r* la A[s,o]

Model Harrison-Ruzzo-Ullman – HRU

Harrison-Ruzzo-Ullman (HRU) bazat pe:

- S set de subiecte
- O set de obiecte
- R set de drepturi de acces
- O matrice de acces $M = (M_{so})_{s \in S, o \in O}$
- intrarea M_{so} este un subset din R specificand drepturile subiectului s asupra obiectului o sau asupra unui alt subiect s'

Operatii **primitive** simple din care se pot construi **comenzi**

- **enter** r into M_{so}
- **delete** r from M_{so}
- **create subject** s'
- **delete subject** s'
- **create object** o
- **delete object** o

Controlul drepturilor de acces se face la nivelul comenzilor

Comenzi

Comenzile sunt combinatii de actiuni primitive; au formatul:

```
command  $c(x_1, \dots, x_k)$ 
    if  $r_1$  in  $M_{s_1, o_1}$  and
    if  $r_2$  in  $M_{s_2, o_2}$  and
         $\vdots$ 
    if  $r_m$  in  $M_{s_m, o_m}$ 
    then  $op_1, \dots, op_n$ 
end
```

s_1, \dots, s_m și o_1, \dots, o_m sunt subiecti și obiecte care apar în lista de parametri x_1, \dots, x_k

Dacă toate **conditiile** sunt îndeplinite atunci se execută lista de operații

Sunt acceptate și comenzi mono-operatie

Example

Crearea unui fisier: s creaza o si devine proprietar

command CREATE_FILE(s,o)

create object o

enter own into $M[s,o]$

end CREATE_FILE

Transferul unui drept; de ex “read”: s1 da un drept lui s2

command CONFER_READ(s1,o,s2)

if own $\in M[s1,o]$

then enter read into $M[s2,o]$

end CONFER_READ

Revocarea unui drept; de ex “write”:

command REVOKE_WRITE(s1,o,s2)

if (own $\in M[s1,o]$) and (write $\in M[s2,o]$)

then delete write from $M[s2,o]$

end REVOKE_WRITE

Ce rezolva HRU - Sisteme de Protectie

Un **sistem de protectie**:

- Set finit de subiecti, obiecte, drepturi si comenzi

Un sistem de protectie este unul **stari-tranzitii**

- Fiecare **stare** este reprezentata de o instanta a **matricei de acces**
- **comenzile** fac **tranzitia** de la o stare la alta

Definitie **scurgere drepturi**. O **stare** M lasa sa se **scurga** dreptul r daca exista o comanda c care adauga dreptul r intr-o intrare din M care anterior nu continea r . Mai precis:

- exista s si o astfel ca $r \notin M_{so}$ si
- dupa executia comenzii c , $r \in M'_{so}$

Definitie **stare sigura**. O **stare** M intr-un sistem de protectie este **sigura** relativ la un **drept** r daca nicio **secventa de comenzi** nu poate transforma M intr-o stare in care se scurge r .

Nota: Faptul ca un drept se poate "scurge" nu este neaparat rau; multe sisteme permit subiectilor sa delege drepturi de acces altor subiecti

Exemplu de sistem “nesigur”

Fie comenzile:

```
command grant_execute ( $s, p, f$ )
  if own in  $M_{s,f}$ 
    then enter execute into  $M_{p,f}$ 
  end
```

```
command modify_own_right ( $s, f$ )
  if execute in  $M_{s,f}$ 
    then enter write into  $M_{s,f}$ 
  end
```

Exemplu

Bob a dezvoltat o aplicatie P1 si vrea ca ea sa fie **executata** de alt utilizator (Tom) dar **nu modificata** de acesta

Sistemul anterior este **nesigur**; permite urmatoarea secventa:

- Bob: grant_execute (Bob, Tom, P1)
- Tom: modify_own_right (Tom, P1)

face ca in M, intrarea $M_{Tom,P1}$ sa contina dreptul de acces write

Siguranta in modelul HRU

Problema sigurantei poate fi formulata astfel:

Este decidabil daca un *subiect* ar putea obtine vreodata un anumit *drept* relativ la un *obiect*?

Teorema. Data fiind matricea de acces M si dreptul r , verificarea **sigurantei lui M** relativa la r este o problema **nedecidabila** in cazul general.

Problema sigurantei

- **este decidabila** pentru sisteme de protectie **mono-operatie**
- **nu este intotdeauna decidabila** pentru alte tipuri de sisteme de protectie
 - **protectia in UNIX** cere mai mult de o operatie per comanda

Siguranta in modelul Take-Grant

Modelul are patru **operatii primitive**

- reprezentate prin grafuri
 - subiecti si obiecte -> **noduri**
 - drepturi -> **arce** etichetate, orientate de la subiect la obiect

Urmatoarele operatii sunt **executate de subiectul s**



Delegare drepturi

grant(o,p,r):



Subiectul **s** “deleaga” lui **o** dreptul de acces **r** asupra lui **p**.

Preconditii:

s are drept de delegare (**Grant**) a unor drepturi catre **o**

s are dreptul **r** asupra lui **p**

Preluare drepturi



Subiectul **s** preia de la **o** dreptul de acces **r** asupra lui **p**

Preconditii

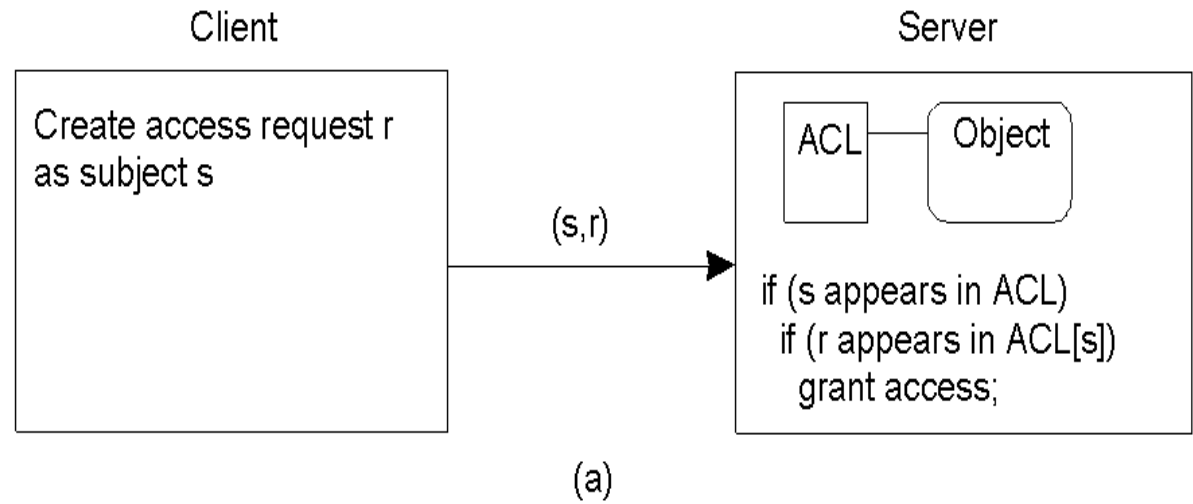
- **s** are drept de preluare (**Take**) de drepturi de la **o**
- **o** are dreptul **r** asupra lui **p**

Cu acest model **se poate decide** daca

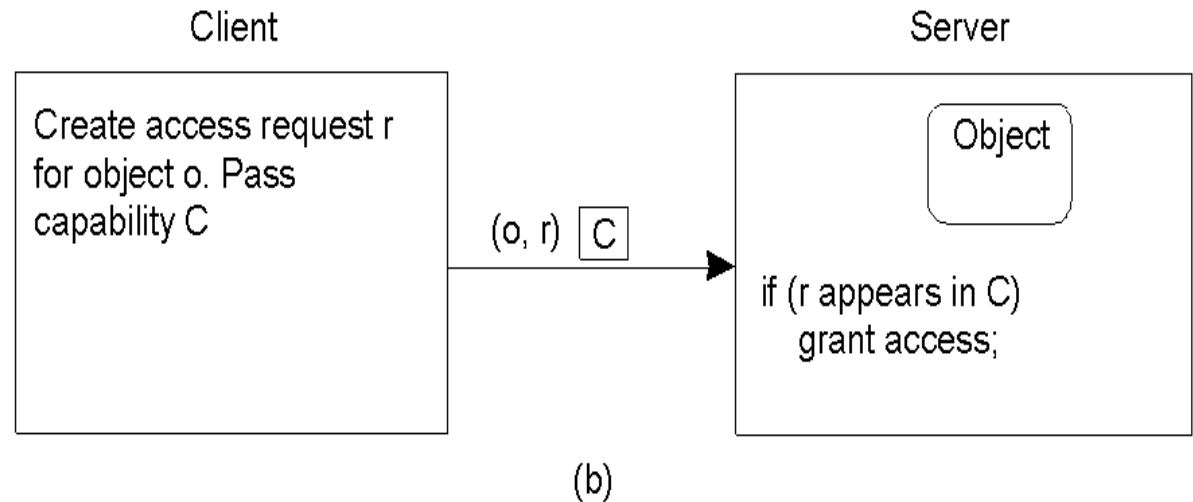
- un subiect **poate partaja** un obiect cu un alt subiect
- un subiect poate "**fura**" accesul la un obiect de la un alt subiect

Implementare Matrice Control Acces (MCA)

ACL (Access
Control List)
fiecare obiect O
pastreaza
 $MCA[*, O]$



Capabilitati
fiecare subiect S
are capabilitatile
din $MCA[S, *]$



Probleme cu modelele "clasice"

Matrice acces (respectiv ACL, C-list):

Nu pot reprezenta **modele** de acces mai **complexe**

- de ex. **bazate pe reguli** cum ar fi **competenta**,
- **cele mai reduse** privilegii sau
- **conflict** de interese

Nu suporta **schimbari dinamice**

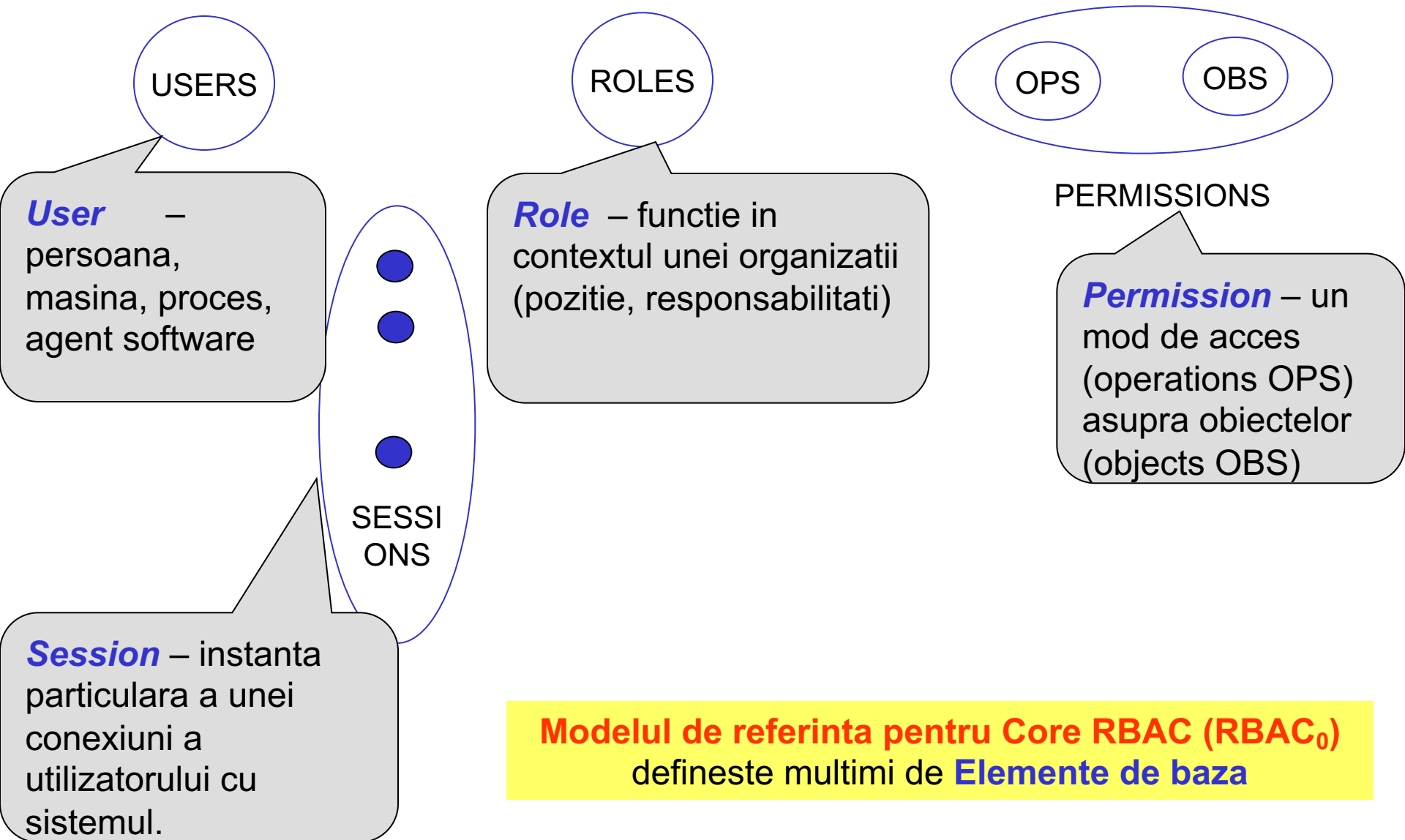
- modificarea **drepturilor unui subiect** necesita inspectarea ACL a fiecarui obiect
- **determinarea subiectilor** care au acces la un obiect necesita inspectarea tuturor listelor de Capabilitati

Nu pot gestiona **drepturi** de acces determinate de **continutul**, **atributele** obiectelor sau de **context**

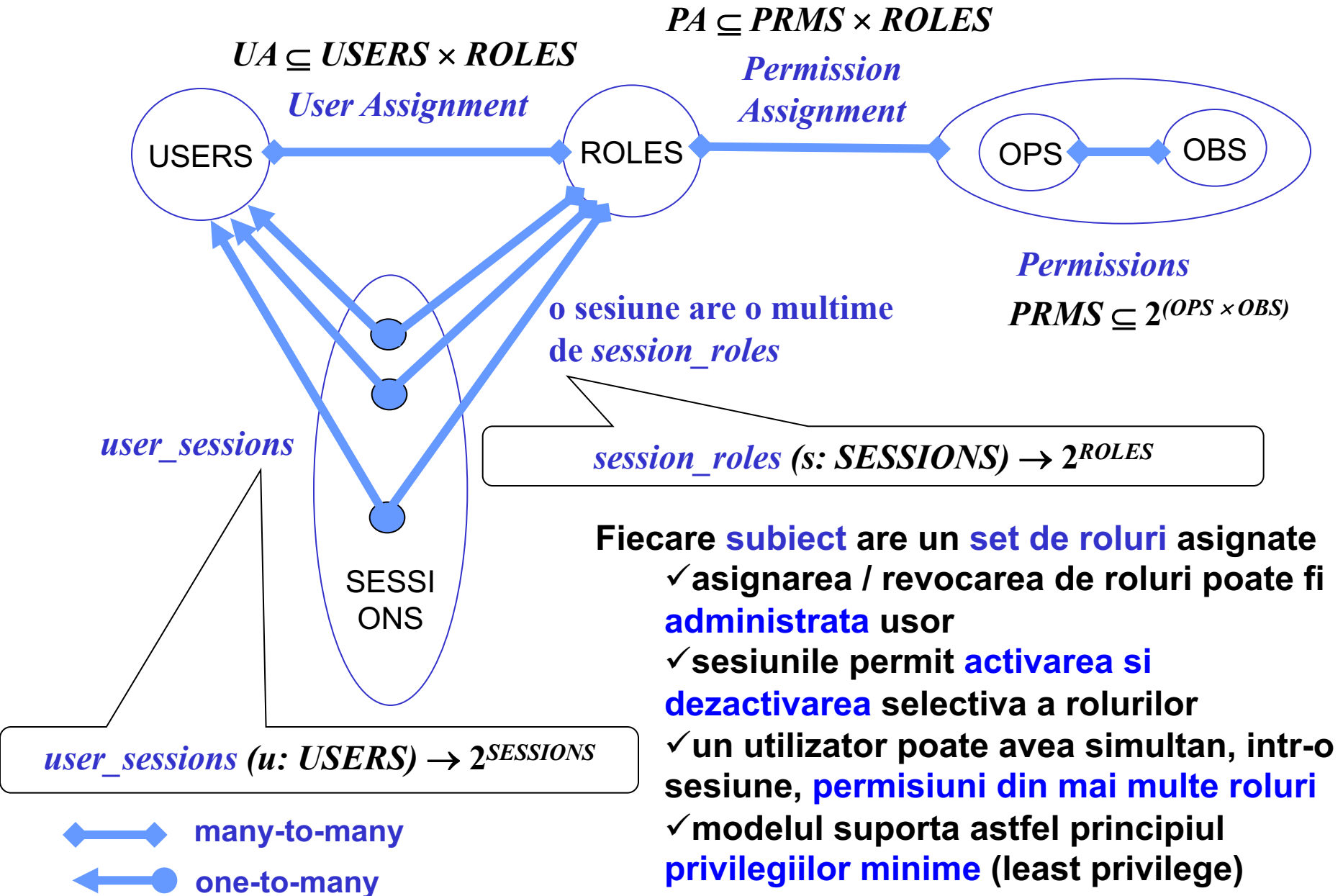
Nu **inregistreaza utilizarea** permisiunilor

- Un subiect poate utiliza permisiunile ori de cate ori

RBAC - Role Based Access Control



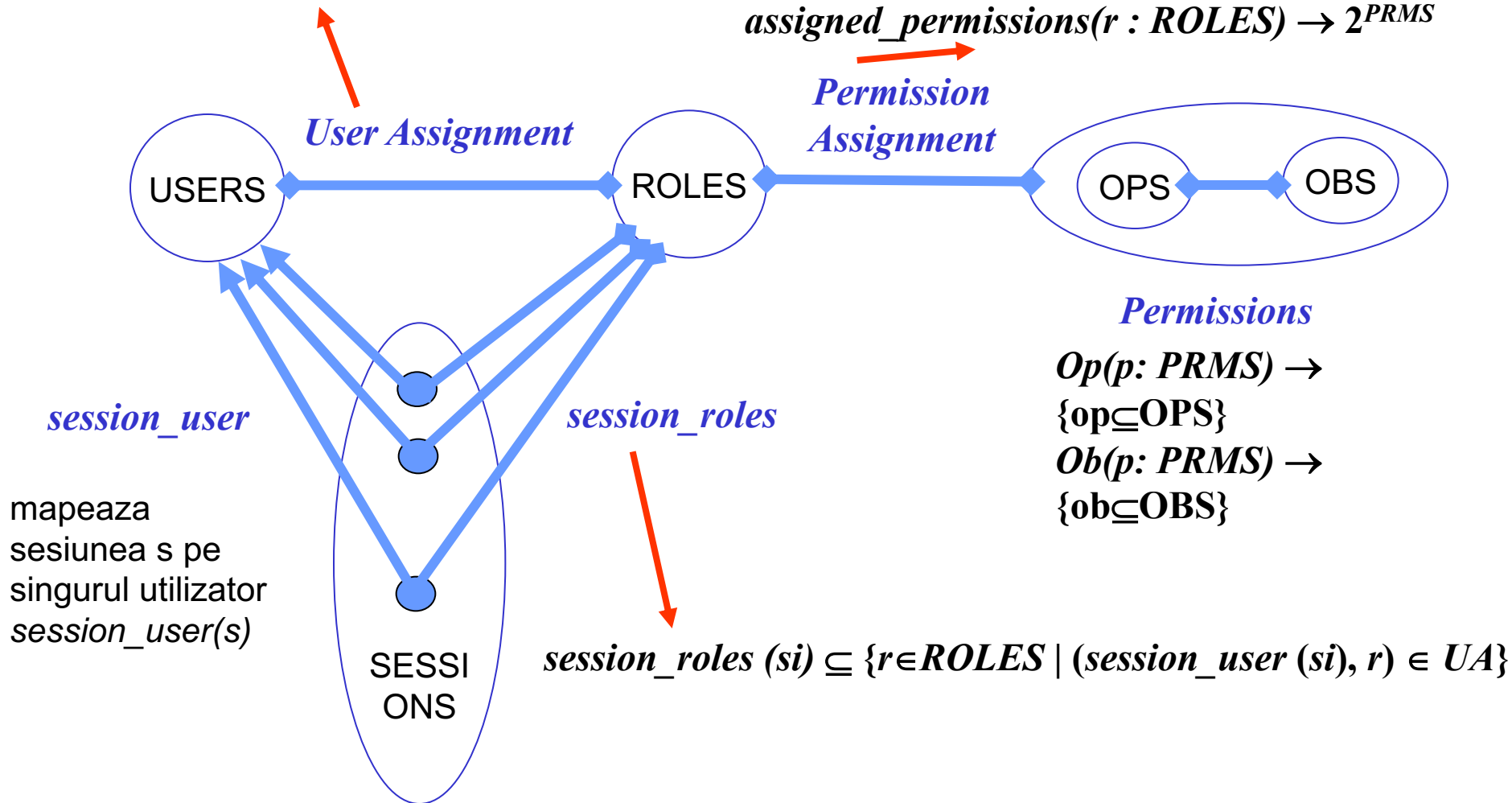
RBAC – relatii intre elementele de baza



RBAC – functii de mapare

$$\text{assigned_users}(r: \text{ROLES}) = \{u \in \text{USERS} \mid (u, r) \in \text{UA}\}$$

$$\text{assigned_permissions}(r: \text{ROLES}) \rightarrow 2^{\text{PRMS}}$$



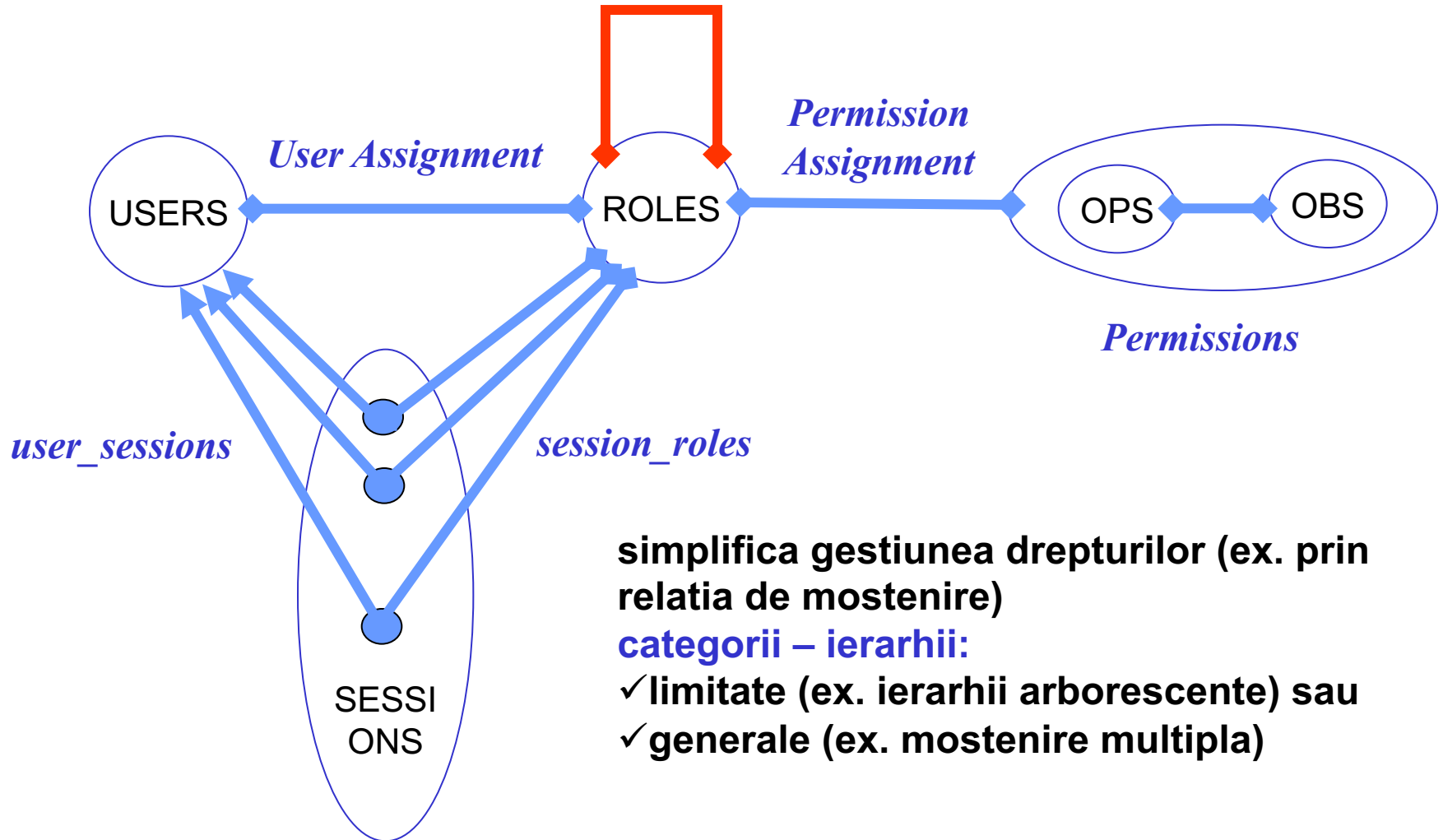
Rolurile (si nu subiectii) au asociate **permisiuni**

- ✓ permit **predefinire roluri**
- ✓ relatia **rol – permisiuni** se schimba mai rar → administrare mai usoara

RBAC ierarhic

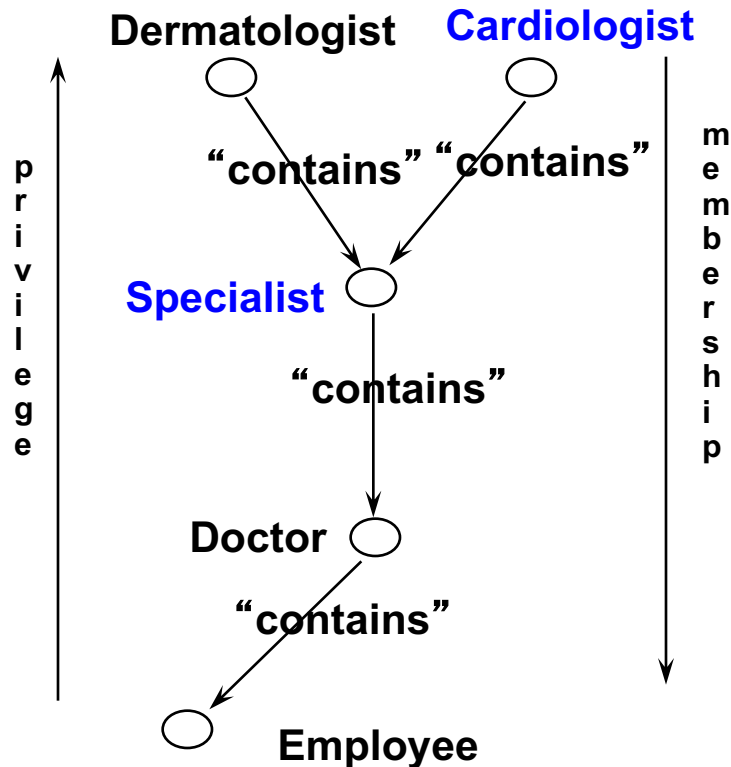
relatie intre roluri: rolul **r1** **include** rolul **r2**, notat $r1 \geq r2 \Leftrightarrow$

RolePermissions ($r2$) \supseteq **RolePermissions** ($r1$) – permisiile lui $r2$ sunt si ale lui $r1$
AssignedUsers ($r1$) \supseteq **AssignedUsers** ($r2$) – user-ii lui $r1$ sunt si ai lui $r2$

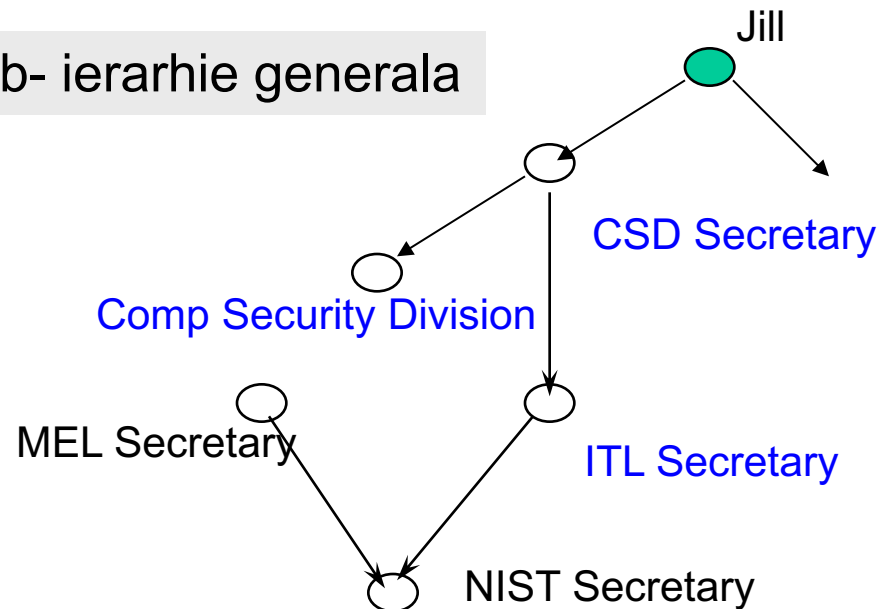


Ierarhia de roluri: exemple

a-ierarhie limitata



b- ierarhie generala



Rolul **CSD Secretary** poate fi definit din privilegiile **Comp Security Division** si **ITL Secretary** care au roluri subordonate

Un **Cardiolog** mosteneste permisiunile unui **Specialist** la care adauga altele

RBAC Separarea sarcinilor

Evita conflictul de interese

1. Forma generala

Constrangere de **cardinalitate** pe un set de roluri

$S = (\text{set roluri}, n)$ – niciun utilizator nu poate avea **n** sau mai multe roluri din set

2. Excludere mutuala a permisiunilor

roluri incompatibile: nu pot fi asumate simultan de un utilizator

daca r_i, r_k incompatibile \Rightarrow

$r_i \in \text{AssignedRoles}(u) \Rightarrow r_k \notin \text{AssignedRoles}(u)$

Doua forme

- ✓ **Static Separation of Duty Relations** – ex. **rolurile r_i si r_k sa nu fie asignate ambele** unui utilizator
- ✓ **Dynamic Separation of Duty Relations** - ex. **rolurile r_i si r_k sa nu fie active in acelasi timp** pentru un utilizator

Specificatie Functionala RBAC

Functii Administrative

Creare si Intretinere seturi de elemente

Creare si Intretinere Relatii

Functii Suport Sistem - Management sesiune

- **CreateSession** - Creaza User Session si da utilizatorului un set de roluri active implicite
- **AddActiveRole** – Adauga un rol ca rol activ in sesiunea curenta
- **DropActiveRole** – Elimina un rol activ din sesiunea curenta

Decizii de control al accesului

- **CheckAccess** – Determina daca subiectul are permisiunea sa execute o anumita operatie asupra obiectului.

Functii Administrative de informare (Review)

(M – Mandatory, O – Optional)

- **AssignedUsers** (M) - Intoarce setul de utilizatori asignat unui rol dat
- **AssignedRoles** (M) - Intoarce setul de roluri asignate unui utilizator dat
- **RolePermissions** (O) - Intoarce setul de permisiuni garantate unui rol dat
- **UserPermissions** (O) - Intoarce setul de permisiuni pe care un utilizator le are prin rolurile sale
- **SessionRoles**(O) - Intoarce setul de roluri **active** asociate cu o sesiune
- **SessionPermissions** (O) - Intoarce setul de permisiuni disponibile in sesiune (reuniunea tuturor permisiunilor asignate rolurilor active din sesiune)

Caracteristici RBAC

Calitati

- utilizatorul poate trece usor de la un rol la altul in cursul unei sesiuni, fara a schimba structura accesului
 - rezultat: RBAC **mai scalabil** ca matricile de acces
- permite **tratarea conflictelor de interese** (rolurile au reguli stricte)
- **reduce overhead la administrare** securitate la nivel de utilizator, obiect, permisiune

Relatia cu MAC si DAC

- RBAC este un **model neutru** fata de politicile de control al accesului
- **poate coexista** cu MAC sau DAC
 - accesul trebuie sa fie permis de RBAC si MAC / DAC

Utilizare

- Microsoft Active Directory, SELinux, FreeBSD, Solaris, Oracle DBMS, PostgreSQL 8.1, SAP R/3

Vă mulțumesc!