

modelli

utenti e sistemi

- un sistema informatico viene tipicamente usato da molti utenti
- conflitti e convergenza di interessi
 - alcune risorse devono essere protette
 - altre risorse devono essere condivise

modelli per il confinamento

- le politiche di sicurezza impongono un confinamento
 - gli utenti (o i soggetti) possono operare solo su certe risorse e non su altre
- molti modelli sono stati sviluppati per esprimere politiche di confinamento
- scopi
 - applicativi o teorici
 - espressione di politiche o configurazione di meccanismi

requisiti

per confinare bisogna...

- distinguere gli utenti (soggetti)
- identificare l'operazione richiesta
- identificare l'oggetto su cui l'operazione opera
- prendere una decisione
 - operazione ammessa o negata
- opzionalmente si può loggare

modelli

tali requisiti sono alla base di due modelli:

- **AAA**
 - nato in ambito telecomunicazioni con scopi non di sicurezza
- **reference monitor**
 - nato in ambito militare per semplificare la certificazione dei sistemi

sono entrambi molto usati

- gli altri modelli riguardano come esprimere politiche di accesso

utenti non “distinti”

- non necessariamente ciascun utente ha una “utenza” (login name)
 - es. web server viene acceduto da utenti senza utenza specifica
- ma in tal caso tutti gli utenti “indistinti” hanno gli stessi diritti

AAA

- Authentication, Authorization, Accounting
 - IETF RFC 2903, anno 2000
 - orientato alle reti
 - radius è un “AAA protocol”
 - accounting: contabilizzazione del consumo delle risorse
 - **ACCOUNTING** ≠ gestione degli account
- il focus era sull’accesso a servizi di connettività, non su sicurezza
 - servizi dial-up, ADSL, telefonia, ecc.

AAA e la sicurezza

- AAA è ora associato soprattutto alla sicurezza
- non solo reti, ma anche sistemi sw e sistemi operativi
- chi ne parla in ambito di sicurezza spesso confonde
 - Authentication, Authorization e ...
 - AUDITING: intendendo spesso logging
- infatti l'accounting non è molto rilevante in ambito sicurezza

autenticazione

- identifica l'utente
 - es. con username e password
 - l'utente è rappresentato nel sistema in qualche modo
 - nei sistemi operativi l'utente è rappresentato da uno o più processi con adeguate informazioni “attaccate” al processo
 - nelle reti può essere una firma elettronica o un identificatore di utente in una sessione
- tale informazione è input alla fase di autorizzazione

id utente $\leftrightarrow$ utente “umano”

- ciascun utente umano dovrebbe essere rappresentato da **uno e un solo** identificatore
 - per identificatore si intende login name
 - altrimenti alcune politiche configurate sui sistemi possono venir invalidate
 - **richiede una policy esterna al sistema**
 - cioè non è forzabile tecnologicamente
 - un matematico la chiamerebbe “corrispondenza biunivoca” un databasista “relazione 1 a 1”

più user id per un utente

- la politica dice: se un utente viola la regola R allora il suo accesso ai sistemi deve essere revocato
 - il meccanismo usato per implementare la politica è la chiusura dell'account che ha violato R
 - se l'utente U ha due account U1 e U2 e viola R con U1 la chiusura di U1 non comporta la revoca dell'accesso ai sistemi poiché U ha ancora U2
- la politica dice: ciascun utente o può avere diritti per l'operazione op1 o l'operazione op2 ma non per entrambe.
 - l'implementazione di questa politica verifica che ciascun account non abbia la possibilità di fare entrambe le operazioni
 - se un utente U ha due account U1 e U2 dove U1 è abilitato a op1 e U2 è abilitato a op2, U ha in sostanza diritti per entrambe le operazioni in violazione della politica

più utenti per un user id

- “strict accountability”
 - si può sempre ricondurre una linea di log ad un utente
 - non possibile in caso di più utenti per una singolo user id
- spesso i sistemi non posseggono adeguate funzionalità di condivisione
- esempio tipico: più utenti conoscono la password di amministratore

autorizzazione

è in realtà composta di tre fasi

- richiesta di accesso
 - stimolata da parte dell'utente o dell'entità che lo rappresenta nel sistema
 - prevede un **oggetto** e una **operazione**
- **controllo di accesso (access control)**
 - in base a delle regole
- autorizzazione
 - concessa (allow): operazione eseguita (cioè accesso)
 - negata (deny): operazione non eseguita

auditing

tipici eventi oggetto di auditing nel modello AAA

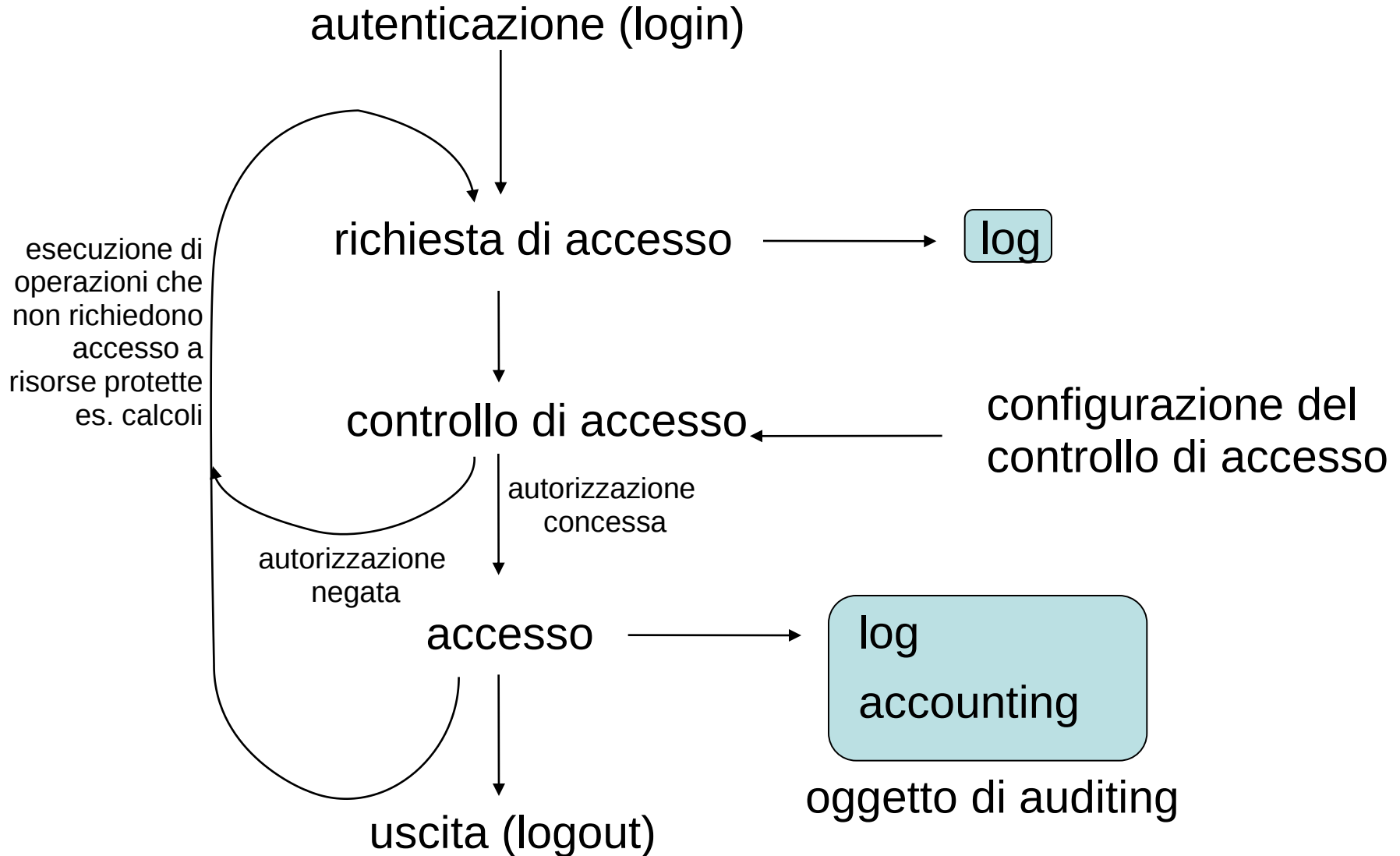
- autenticazione (riuscita o negata)
- richiesta di accesso
- autorizzazione per una richiesta di accesso (concessa o negata)
- risultato di una operazione

fuori dal modello AAA molti altri eventi possono essere oggetto di auditing

auditing

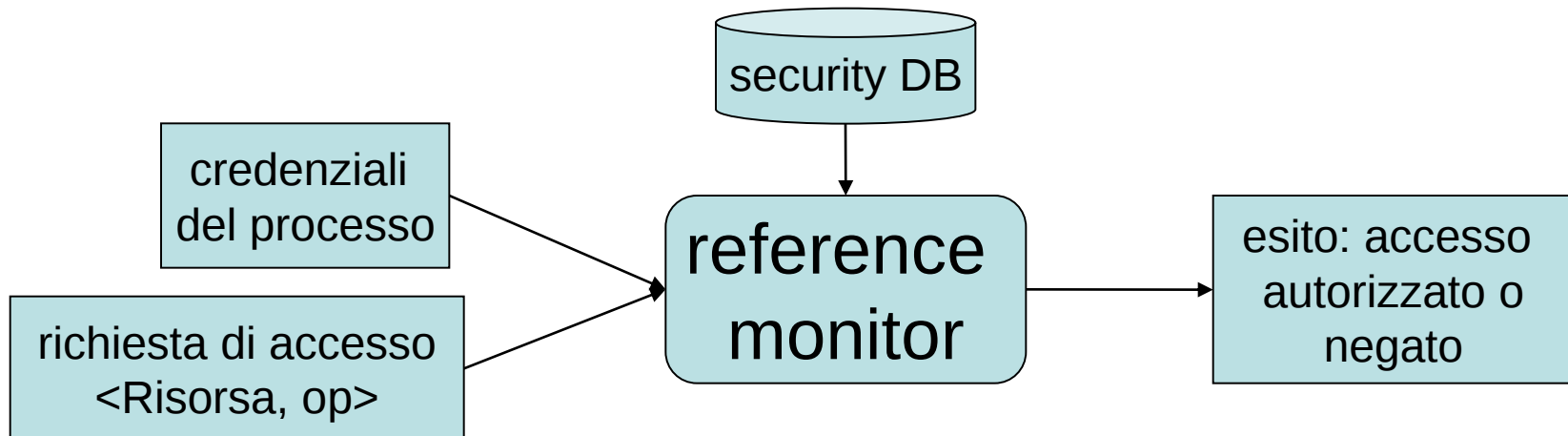
- auditing = controllo o verifica di “adeguatezza”
 - tipicamente manuale o semi-automatizzato
 - solo l'uomo ha l'adeguata flessibilità per adattarsi a variazioni di esigenze, strumenti, ambiente di lavoro, ecc.
- può essere fatto a vari livelli (varie accezioni diverse)
 - access auditing
 - log auditing
 - system security auditing
 - network security auditing
 - auditing di procedure (iso 27001)
 - auditing di competenze (di persone) (CISSP – SSCP e CISA – CISM)
 - ecc.

AAA: ciclo operativo



reference monitor (security kernel)

- parte del s.o. che effettua il controllo di accesso
- caratteristiche:
 - invocato ad ogni richiesta di accesso
 - a prova di intrusione (nessuna vulnerabilità)
 - abbastanza piccolo da essere verificabile
- introdotto nel 1972 in James Anderson et al. - Computer Security Technology Planning Study
- richiesto in certe certificazioni: es. in TCSEC $\geq B2$



reference monitor: realizzazioni

- sistemi senza reference monitor
 - Windows 3.x, 95,98, Me (controllo di accesso limitato)
 - Linux <2.6 (controllo di accesso “efficace” ma architettura senza r.m.)
- sistemi con reference monitor
 - Windows NT, 2000, XP, 2003, Vista, 7, 10
 - Linux >2.6 (Linux Security Modules)
 - vari modi di realizzare un security kernel
 - SELinux (Android), AppArmor, Smack, ecc.

modelli per le politiche di controllo di accesso

- Discretionary Access Control (DAC)
- Mandatory Access Control (MAC)

Discretionary Access Control

- gli utenti hanno il diritto per cambiare i diritti sulle risorse
 - es. il proprietario U1 di un file può dare, a sua discrezione, il diritto all'utente U2 di accedere a tale file
- implementazioni
 - Linux, Windows NT/2000/XP/2003/Vista/7

Mandatory Access Control

- gli utenti **NON hanno il diritto di** cambiare i diritti sulle risorse
 - potrebbe non esistere il concetto di proprietario
 - il proprietario di una risorsa è l'organizzazione (ad es. rappresentata dall'amministratore di sistema) e non un certo utente
- i diritti sono configurati dall'amministratore
- non è detto che l'utenza di amministratore esista
- implementazioni
 - Linux: es. SELinux, AppArmor
 - Windows:(Mandatory Integrity Control, MIC)

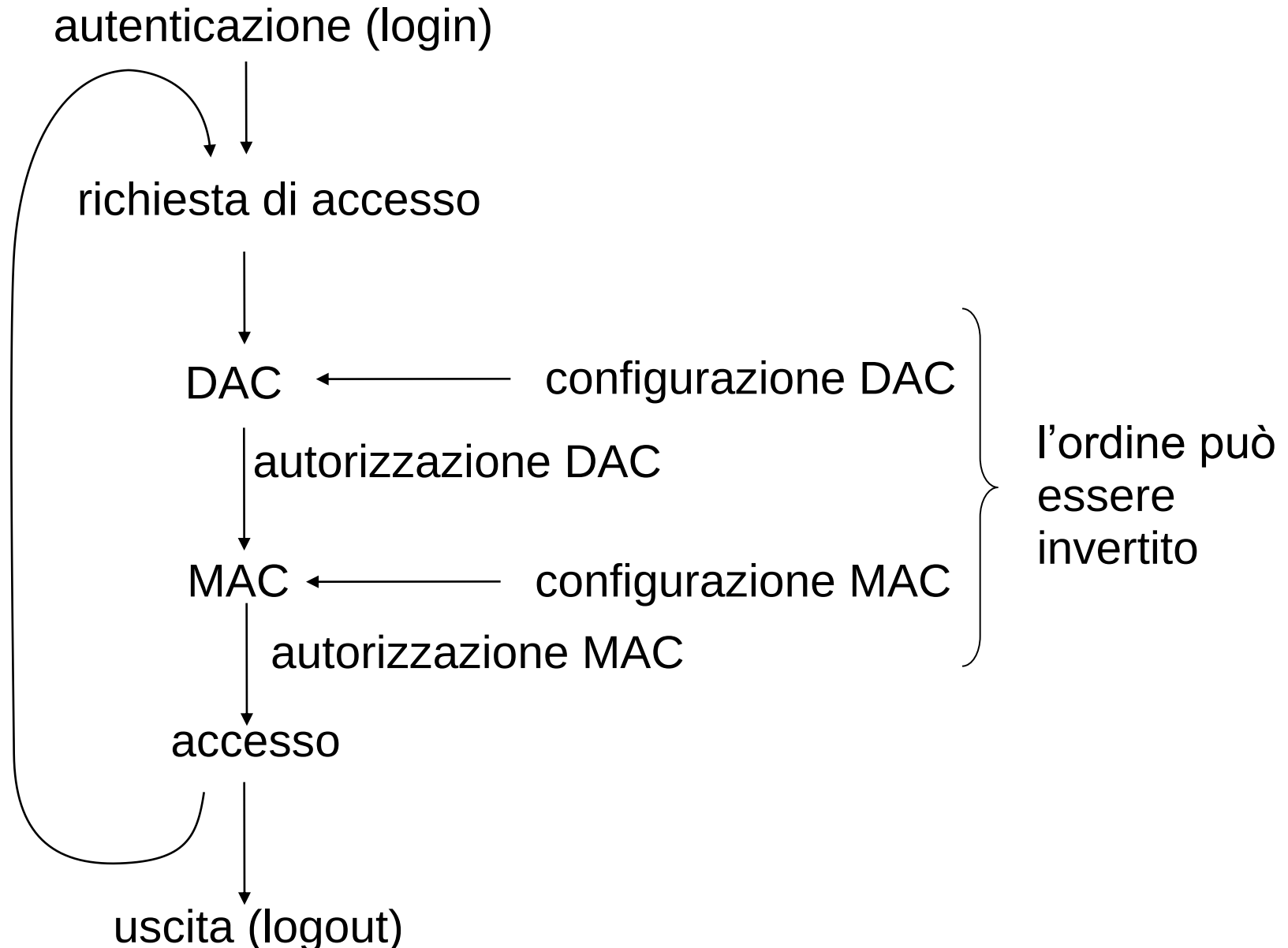
DAC vs. MAC

- DAC il più diffuso
 - flessibile
 - sicurezza “delegata” agli utenti (tipicamente al proprietario della risorsa)
- MAC considerato il più sicuro
 - molto scomodo per gli utenti
 - se due utenti devono scambiare un file devono chiamare l'amministratore
 - usato molto in ambito militare e sui server

MAC+DAC

- i sistemi con MAC tipicamente supportano anche DAC
- accesso consentito se sia i controlli mandatory che quelli discretionary danno autorizzazione
- permette di avere isole di “discrezionarietà” confinati da muri “mandatori”
 - es. il web server è separato dagli utenti da una configurazione MAC
 - ma MAC non isola gli utenti tra di loro, essi sono eventualmente isolati mediante DAC

AAA: MAC+DAC



access matrix

- matrice che descrive i diritti di ciascun soggetto sugli oggetti o soggetti

objects (entities)

	o_1	...	o_m	s_1	...	s_n
s_1						
s_2						
...						
s_n						

subjects

- Subjects $S = \{ s_1, \dots, s_n \}$
- Objects $O = \{ o_1, \dots, o_m \}$
- Rights $R = \{ r_1, \dots, r_k \}$
- Entries $A[s_i, o_j] \subseteq R$
- $A[s_i, o_j] = \{ r_x, \dots, r_y \}$ means subject s_i has rights $r_x, \dots, r_y$ over object o_j

i soggetti sono anche oggetti

- nella maggioranza dei casi i soggetti sono anche oggetti
 - cioè un soggetto può operare su altri soggetti (che svolgono il ruolo di oggetti)
 - es. un processo può essere killato, debuggato, stoppato, ecc.
 - es. un utenza può essere creata, bloccata, cancellata

usi della access matrix

- per esprimere politiche
- per esprimere il “protection state”

protection state

- è la parte dello stato di un sistema che è rilevante per la sicurezza
- una transizione di stato è un cambiamento della access matrix
 - creazione e distruzione di oggetti
 - aggiunge o cancella colonne
 - creazione e distruzione di soggetti
 - aggiunge o cancella righe e colonne
 - inserimento di un diritto in una cella
 - cancellazione di un diritto in una cella
- **le transizioni di stato possono essere vincolate da particolari diritti nella matrice stessa**
 - eccezioni: il diritto di creazione è eventualmente associato a un soggetto e non ad una cella della matrice

rappresentazione del protection state

- access control list
 - ciascun oggetto ha associato una struttura dati che esprime quali soggetti possono agire sull'oggetto stesso e con che operazioni
- capabilities
 - ciascun soggetto ha associato una struttura dati che esprime su quali oggetti può agire e con che operazioni
- sono modi per rappresentare una “matrice sparsa”

ownership

- s_i own o_j , cioè $\text{own} \in A[s_i, o_j]$
- si ha il diritto di modificare a piacimento l'elemento $A[s_i, o_j]$
 - cioè i propri diritti su o_j
 - più che un diritto su o_j è un diritto su $A[s_i, o_j]$, cioè un meta-diritto
- vincoli tipici
 - un solo owner
 - come in unix e windows
 - owner non modificabile
 - ma alle volte l'ownership è trasferibile (unix: `chown`, win: “take ownership”)
- spesso implica il diritto grant

grant e attenuazione del privilegio

- s_i grant o_j
- si ha il diritto di modificare gli elementi della colonna $A[s, o_j]$ con $s \neq s_i$
 - cioè dare ad altri soggetti diritti su o_j
- più che un diritto su o_j è un diritto su $A[. , o_j]$
- principio di attenuazione del privilegio: si non può dare ad altri diritti che lui stesso non possiede
 - è un vincolo tipico nei DBMS

access matrix, DAC e MAC

- una matrice di accesso può rappresentare politiche DAC o MAC
- la presenza di diritti grant rendono la politica DAC

esempio

- Processes p, q soggetti
- Files f, g oggetti
- Rights *Read, Write, eXecute, Append, Own+grant*

	f	g	p	q
p	rwo	r	rwXO	w
q	a	ro	r	rwXO

- politica DAC

esempio

- Procedures *inc_ctr*, *dec_ctr*, *manage* soggetti
- Variable *counter* oggetti
- Rights +, −, *call*

	<i>counter</i>	<i>inc_ctr</i>	<i>dec_ctr</i>	<i>manage</i>
<i>inc_ctr</i>	+			
<i>dec_ctr</i>	-			
<i>manage</i>		<i>call</i>	<i>call</i>	<i>call</i>

- politica MAC