

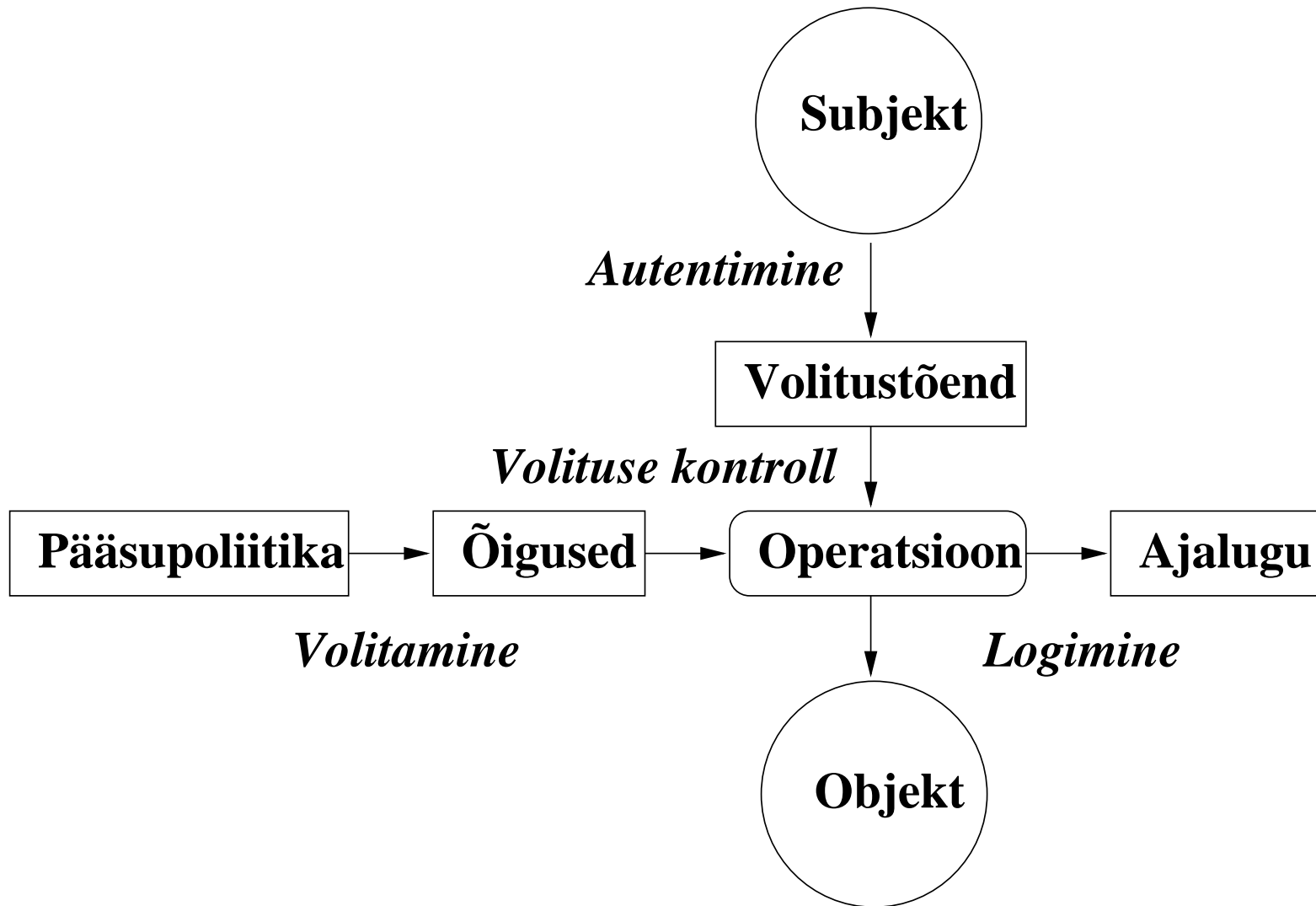
## Mitmekasutajasüsteemid

- Probleemi püstitus
- Diskretsionaarne pääsupoliitika
- Mandatoorne pääsupoliitika
- Rollipõhine pääsupoliitika

## Mitmekasutajasüsteemid

- Probleem:
  - Hulk kaitstavaid objekte (failid, kirjed, kataloogid, seadmed, ...)
  - Hulk juurdepääsu omada võivaid subjekte (inimesed, protsessid, ...)
  - Erinevatel kasutajatel on objektidele erinevad õigused
  - Autentimise abil saime teada, kes on kes
- ⇒ Vaja õigusi autenditud kasutajatele volitada

# Juurdepäasu kontroll



## Diskretsionaarsed pääsupoliitikad

- Lühend DAC (*Discretionary Access Control*)
- Põhineb informatsiooni omamisel ja õiguste delegeerimisel
- Levinuim laiatarbesüsteemides (peaaegu ainuvaldav)
- Kinnine DAC-poliitika: vaikimisi keelatud, pääsuluba antakse subjekt-objekt paaride määratlemisega
- Lahtine DAC-poliitika: vaikimisi lubatud, paaridega määratakse keelatud tegevused
- Igal objektil on omanik (mingi subjekt)
- Omanikul on objektile maksimaalsed õigused
- Omanik saab määrata teiste subjektide juurdepääsuõigusi oma objektidele
- Subjektide grupeerimine lihtsuse huvides

## DAC realiseerimine

- Iga objektiga võib siduda pääsuloendi (ACL - *Access Control List*). ACL sisaldab subjektikirjelduste ja lubatud/keelatud tegevuste paare.

```
fail.txt: (mati: LK, kati: L)
```

- Iga subjektiga võib siduda voliloendi (*capability list*). Voliloend sisaldab objekti ja lubatava pöörduse paare.

```
mati: (printer: K, CD: L)
```

- *capability* tõlgitakse volituseks
- Näited reaalistest volitustest Linuxis:
  - CAP\_DAC\_OVERRIDE
  - CAP\_KILL
  - CAP\_SYS\_RAWIO
  - CAP\_NET\_BIND\_SERVICE

## DAC realiseerimine

- Pääsupoliitika võib esitada maatriksina (nn. maatriksmudel):

	$O_1$	$O_2$	$O_3$	...	$O_n$
$S_1$	L	LK	L	...	-
$S_2$	LK	-	L	...	L
...	...	...	...	...	...
$S_m$	-	-	L	...	-

$O_i$  — objektid,  $S_i$  — subjektid

L — lugemine, K — kirjutamine

## DAC probleemid

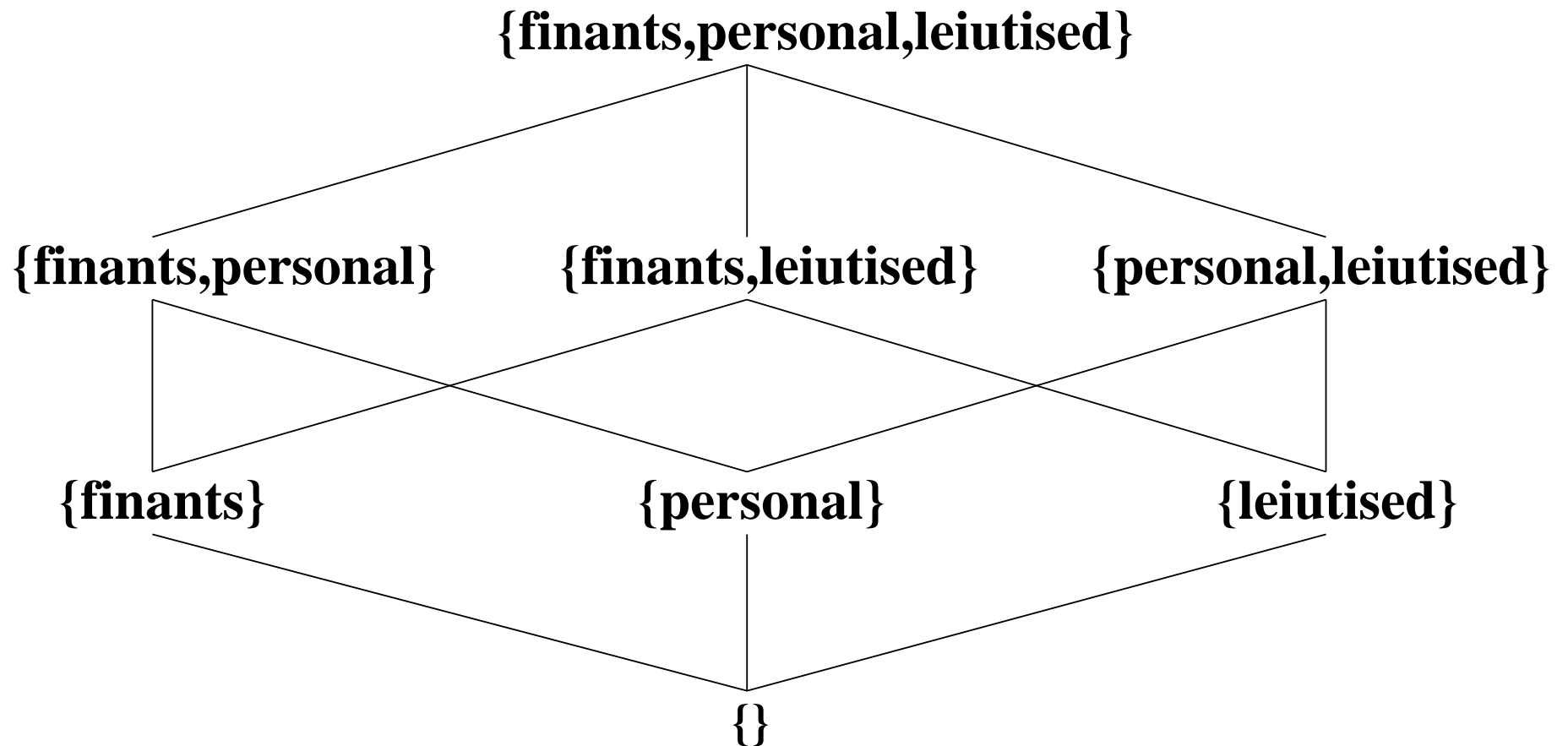
- Subjekt saab oma õigusi meelevaldselt teistele edasi delegeerida
- Puuduvad vahendid ülesüsteemsete pääsureeglite kehtestamiseks (iga kasutaja teeb omamoodi)
- Enamasti ei kehti reeglid kõrgemal tasemel (süsteemiülem omab kõiki õigusi)
- Trooja hobused, viirused jms programmid lasevad infol lekkida
- Konfidentsiaalsust kaitsvaid kitsendusi ei saa selgelt määratleda

# Mandatoorsed pääsupoliitika 1

- Lühend MAC (*Mandatory Access Control*)
- Siia kuuluvad pääsupoliitika, mis kõrvaldavad erinevaid DAC turvaprobleeme (eeskätt soovitakse infovoogu kontrolli all hoida)
- Objektidele ja subjektidele omistatakse turvaklassid
- Iga märgend koosneb kahest osast:
  - Tüüpi kirjeldav kategooria (järjestamata)
  - Turvatase määrav järjestatud tundlikkusaste (objektidel) või lojaalsusaste (subjektidel)
- Märgendid on ainult osaliselt järjestatud (matemaatilises mõttes on tegemist võrega)
- Klass  $c_1$  on kõrgem ( $\geq$ ) klassist  $c_2$ , kui  $c_1$  turvatase on suurem  $c_2$  turvatasemest ning  $c_2$  kategooriad sisalduvad  $c_1$  omades.



## Mandatoorsed pääsupoliitika 2



Juurdepääsu võre kategooriate sisaldumise järgi

## Mandatoorsed pääsupoliitika 3

- Formaalseid meetodeid on erinevaid
- Levinuimad on Bell-LaPadula mudel ja Biba mudel (duaalsed)
- Bell-LaPadula mudel rakendatuna andmeüksustele:
  - Subjekt  $S$  tohib lugeda andmeüksust  $A$ , kui  $c_S \geq c_A$
  - Subjekt  $S$  tohib kirjutada andmeüksust  $A$ , kui  $c_S \leq c_A$
- Andmebaaside jaoks on MAC meetodeid edasi arendatud, eri turvatasemed tähendavad erinevaid vaateid. Leidub MAC poliitikat kasutada suutvaid andmebaasisüsteeme, igaüks natuke erinev.
- MAC kasutavaid operatsioonisüsteeme kutsutakse usaldatavateks (*trusted*) — Trusted IRIX, Trusted Solaris, ...

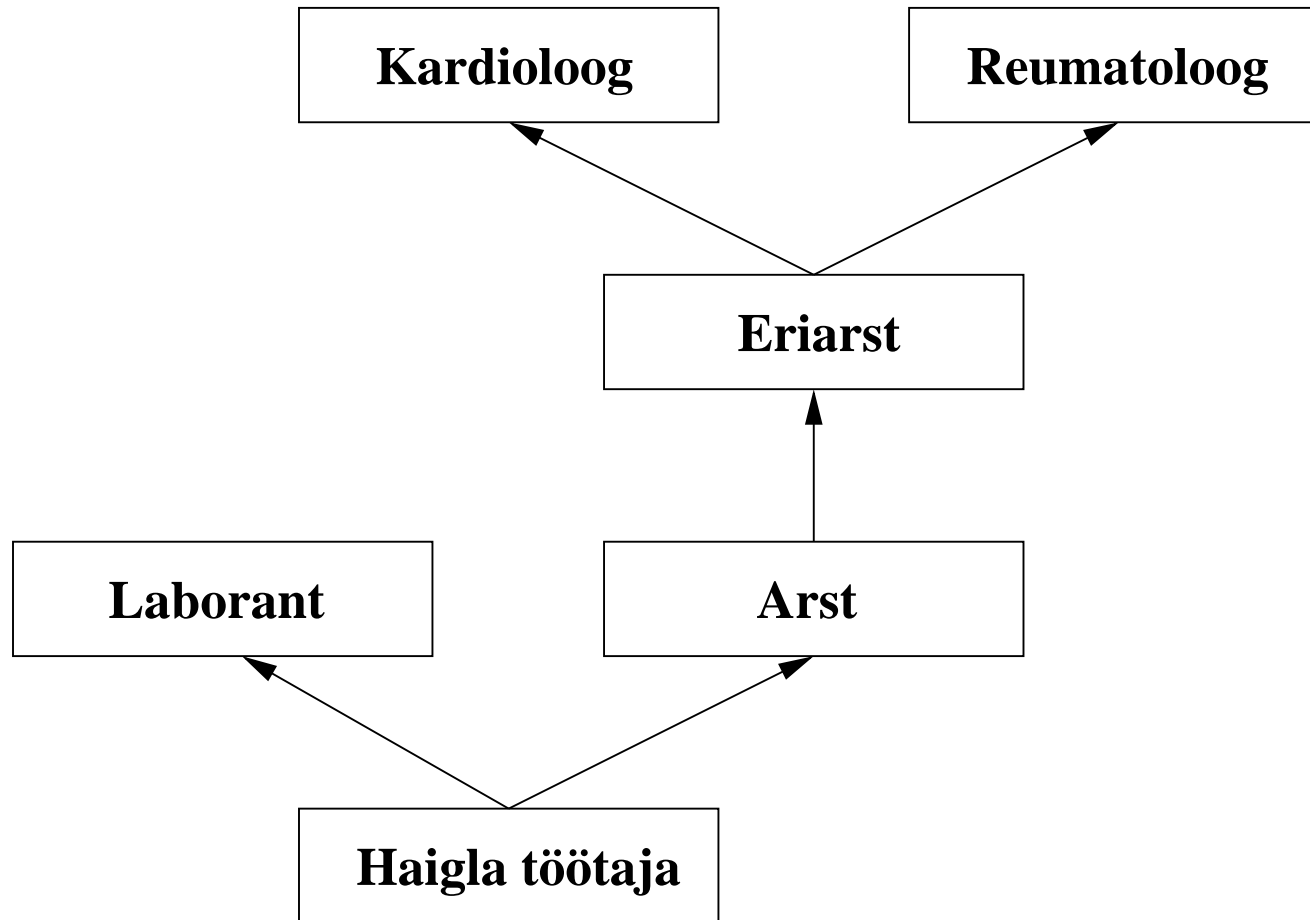
## MAC head ja vead

- Trooja hobuste probleem on likvideeritud
- Kasutajate kokkumängust tingitud salakanalid nende vahel on alles
- Konfidentsiaalsuse tagamine olemas, "pime kirjutamine" võib terviklusele probleemiks saada
- Väga jäik ja staatiline — kasutajad ja protsessid ise ei saa objektide turvatasemeid muuta
- Jäikuse tõttu ilmselt sobimatu kommertsmaailma
- Ei saa automaatselt rakendada võimalikku kollektiivpääsu

## Rollipõhised pääsupoliitikad

- Lühend RBAC (*Role-Based Access Control*)
- Reaalsesse maailma ja ärimudelitesse sobivamad kui DAC ja MAC
- Kasutajate pääsu reguleeritakse vastavalt nende tööalasele funktsioonile
- Roll väljendab tööülesandeid ("raamatupidaja")
- Ühes rollis võib olla mitu kasutajat, üks kasutaja võib olla erinevatel aegadel erinevates rollides
- Rollide ülesanded ja privileegid kattuvad osaliselt → tekib rollihierarhia koos päritavate õigustega
- Võib kombineerida nii DAC kui MAC poliitikatega, RBAC ise on "poliitiliselt neutraalne" — vahend etteantud poliitika realiseerimiseks

## Rollihierarhia näide



Nooled iseloomustavad pärimise suunda üldisemalt üksikule

## Rollipõhise poliitika põhiprintsiibid

- Staatiline kohustuste lahusus — määratakse üksteist välistavad rollid
- Rollikandjate arvu piiramine — iga rolli saab kanda teatud etteantud arv kasutajaid
- Dünaamiline kohustuste lahusus — vahel peab inimene olema mitmes üksteist välistavas rollis, aga igal ajahetkel siiski ühes korraga
- Protseduuriline kohustuste lahusus — piiratakse ühe ja sama isiku juurdepääsu mingi kriitilise funktsiooni järjestikustele faasidele