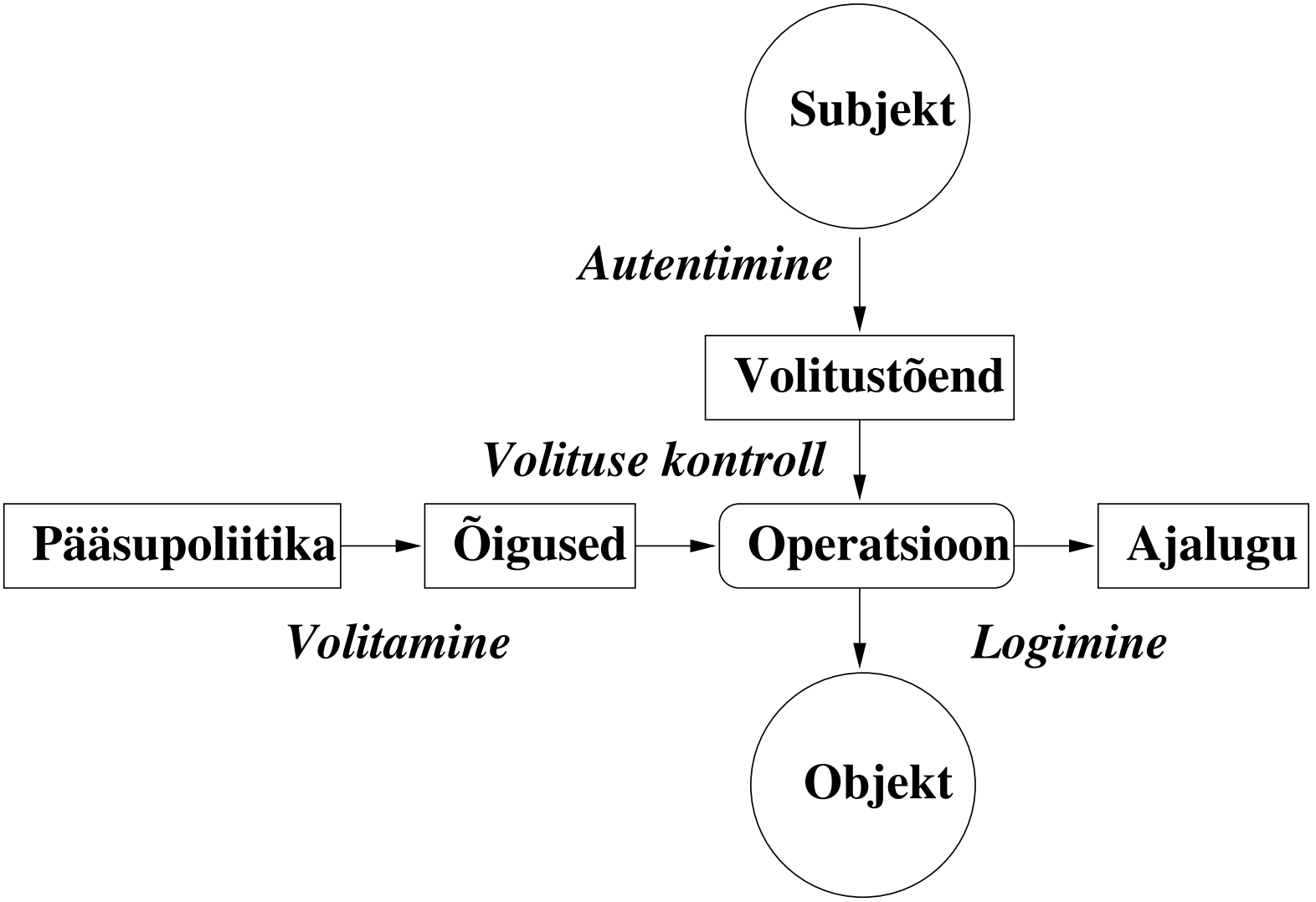
Mitmekasutajasüsteemid

- Probleemi püstitus
- Diskretsionaarne pääsu reguleerimine
- Mandatoorne pääsu reguleerimine
- Rollipõhine pääsu reguleerimine

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
- $\Rightarrow$ Vaja õigusi autenditud kasutajatele volitada

Juurdepääsu kontroll



Diskretsionaarne pääsu reguleerimine

- 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 reaalsetest volitustest Linuxis:
  - CAP_DAC_OVERRIDE
  - CAP_KILL
  - CAP_SYS_RAWIO
  - CAP_NET_BIND_SERVICE

DAC realiseerimine

- Pääsureeglid 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

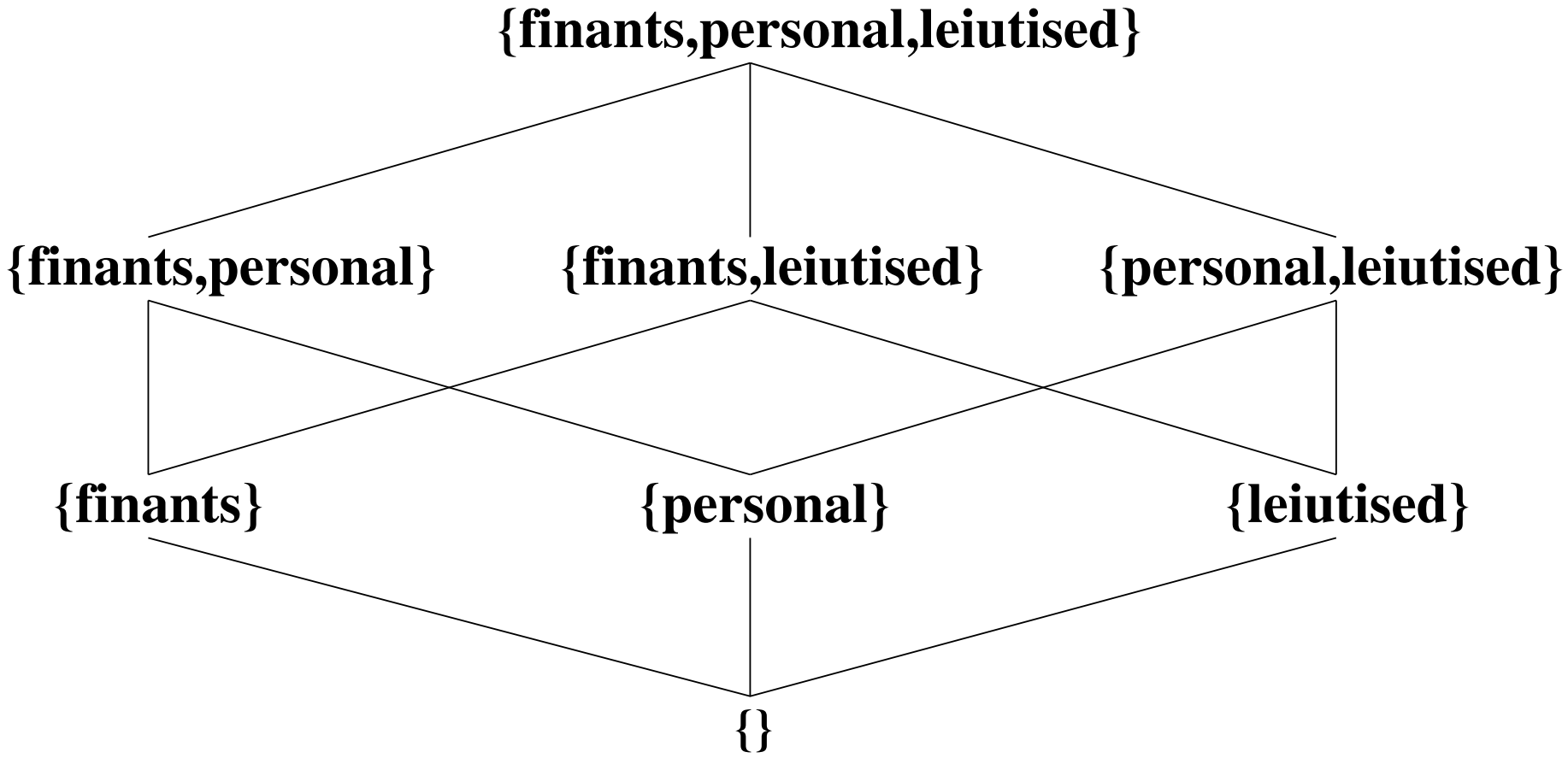
Mandatoorne pääsu reguleerimine

- Lühend MAC (*Mandatory Access Control*)
- Siia kuuluvad reeglistikud, mis kõrvaldavad erinevaid DAC turvaprobleeme (eeskätt soovitakse infovoogu kontrolli all hoida)
- Reeglid on süsteemsed, mitte objekti omaniku hallatavad
- Objektidele ja subjektidele omistatakse turvaklassid

Tundlikkusmärgendid

- Turvaklasside tähistamiseks kasutatakse tundlikkusmärgendeid (*sensitivity label*)
- Iga märgend koosneb kahest osast:
  - Tüüpi kirjeldav kategooria (järjestamata)
  - Turvataset 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.

Turvamärgendide võre



Juurdepääsu võre kategooriate sisaldumise järgi

Mandatoorse pääsu reguleerimise mudelid

- Formaalseid meetodeid on erinevaid
- Levinuimad on Bell-LaPadula mudel (konfidentsiaalsuse kaitseks) ja Biba mudel (duaalne, tervikluse kaitseks)
- 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, ...

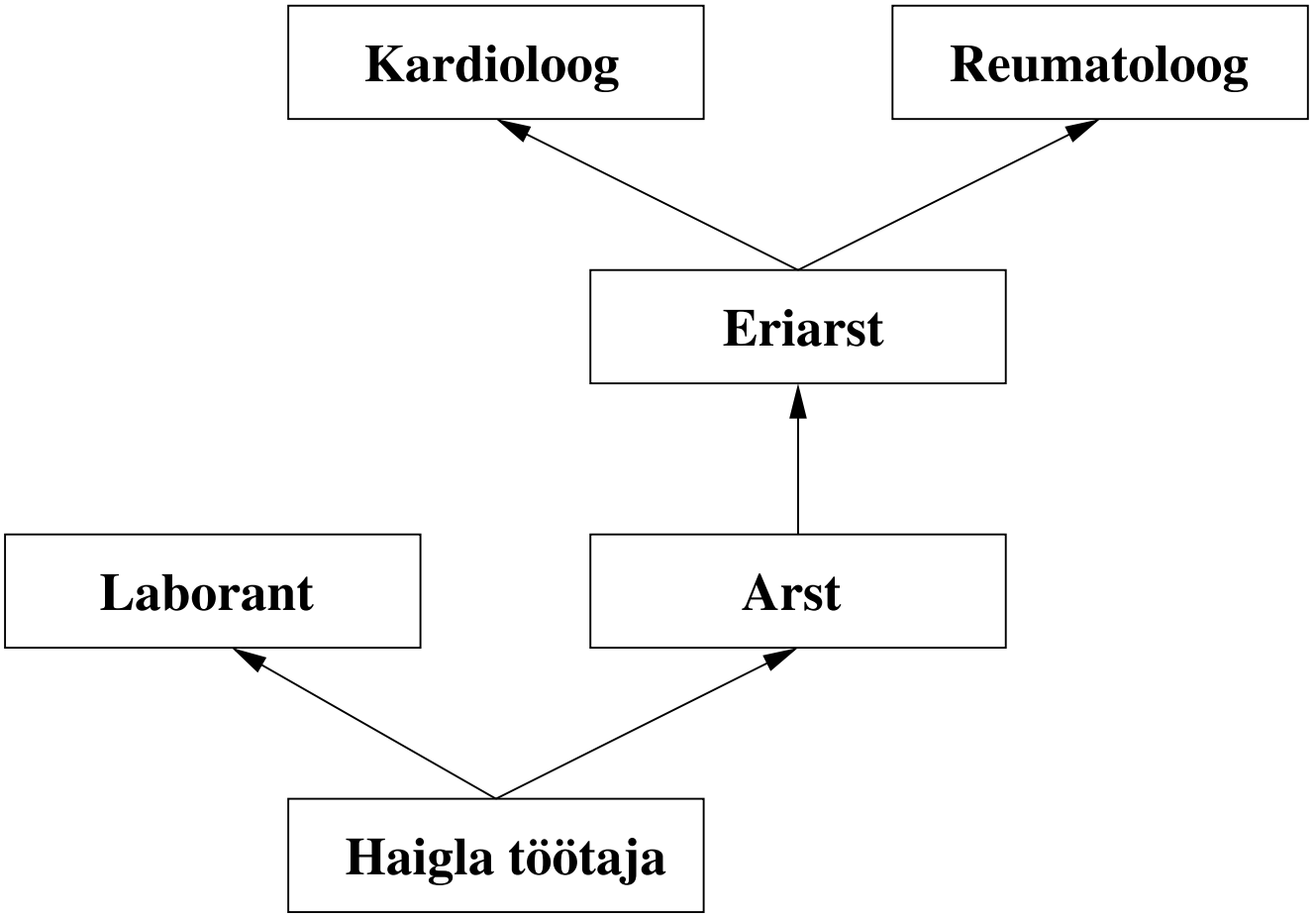
MAC head ja vead

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

Rollipõhine pääsu reguleerimine

- 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 $\rightarrow$ 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