Usage Control (UCON)
or
ABAC on Steroids

Prof. Ravi Sandhu
Executive Director and Endowed Chair

February 26, 2016

ravi.sandhu@utsa.edu
www.profsandhu.com

Motivation

- Traditional access control models are not adequate for today’s distributed, network-connected digital environment.
  - Authorization only – No obligation or condition based control
  - Decision is made before access – No ongoing control
  - No consumable rights - No mutable attributes
  - Rights are pre-defined and granted to subjects

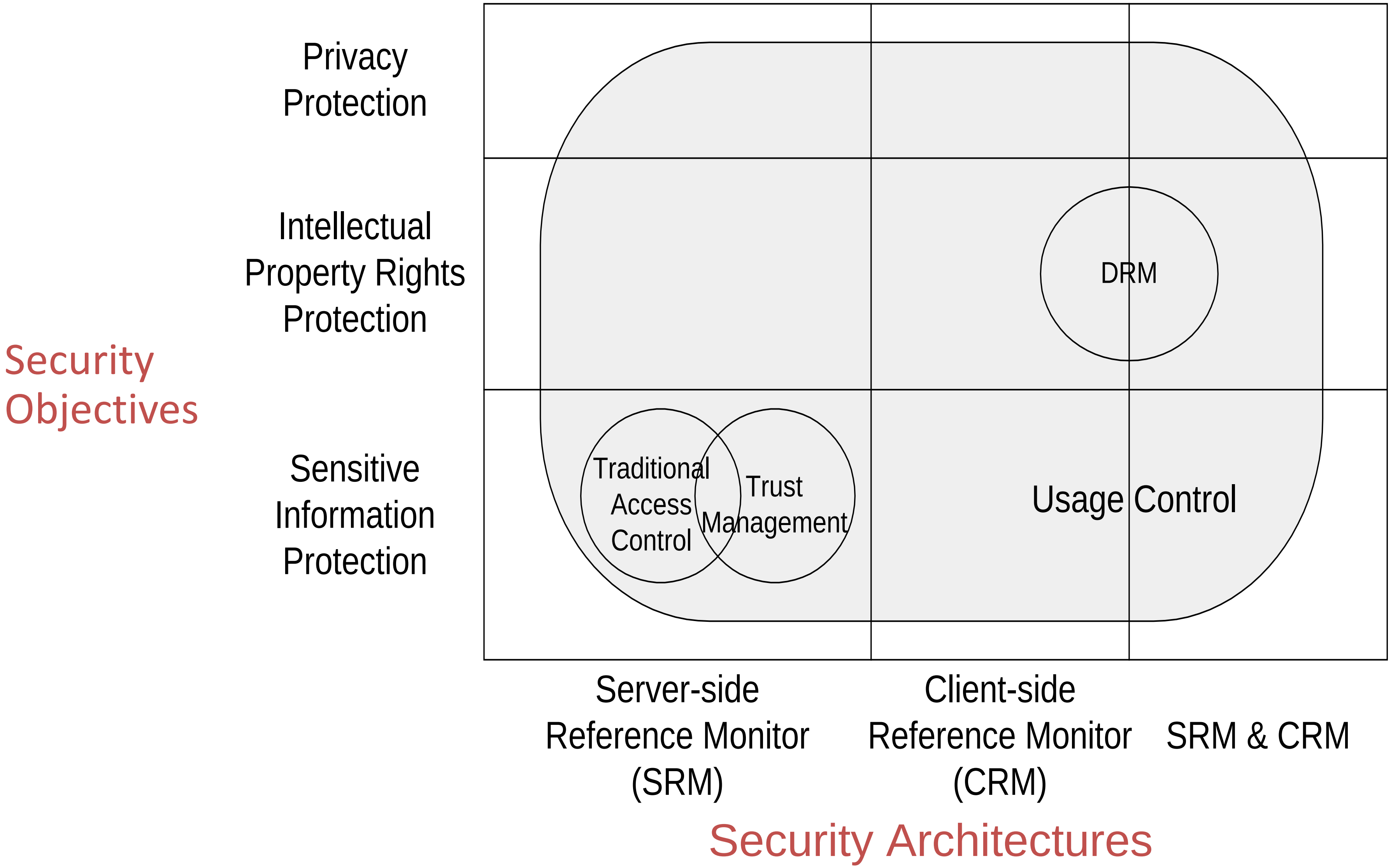
Motivation

- No access control model available to capture Digital Rights Management (DRM)
  - Control after dissemination
  - IPR protection
- Need for a unified model that can encompass traditional access control models, DRM and other enhanced access control models from recent literature

Usage Control (UCON)

- Scope
  - Encompass traditional access controls, trust management, digital rights management and more
  - For sensitive information protection, IPR protection, and privacy protection
- Model
  - General purpose, policy neutral models
  - Policy is assumed to be given to the system
  - Transaction based control
  - Existence of right is determined when access is attempted by a subject (no predefined access matrix)
  - Attribute-based access control

Usage Control (UCON)



Security Objectives

Privacy Protection

Intellectual Property Rights Protection

Sensitive Information Protection

DRM

Traditional Access Control

Trust Management

Usage Control

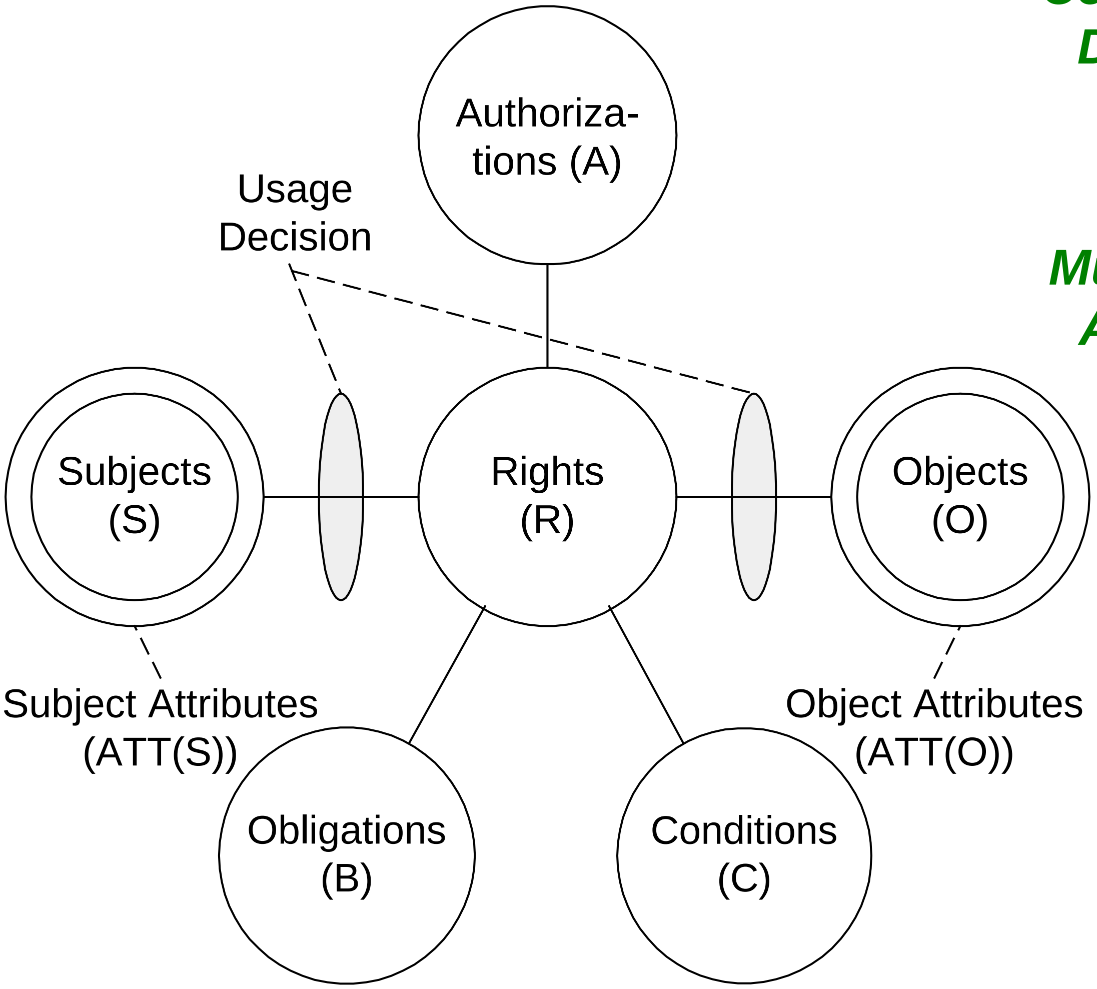
Server-side Reference Monitor (SRM)

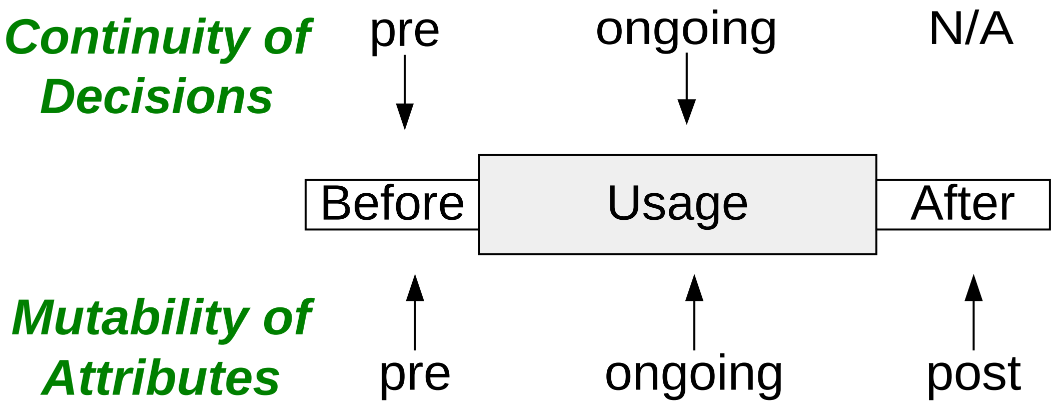
Client-side Reference Monitor (CRM)

SRM & CRM

Security Architectures

Building UCON$_{ABC}$ Models





Continuity

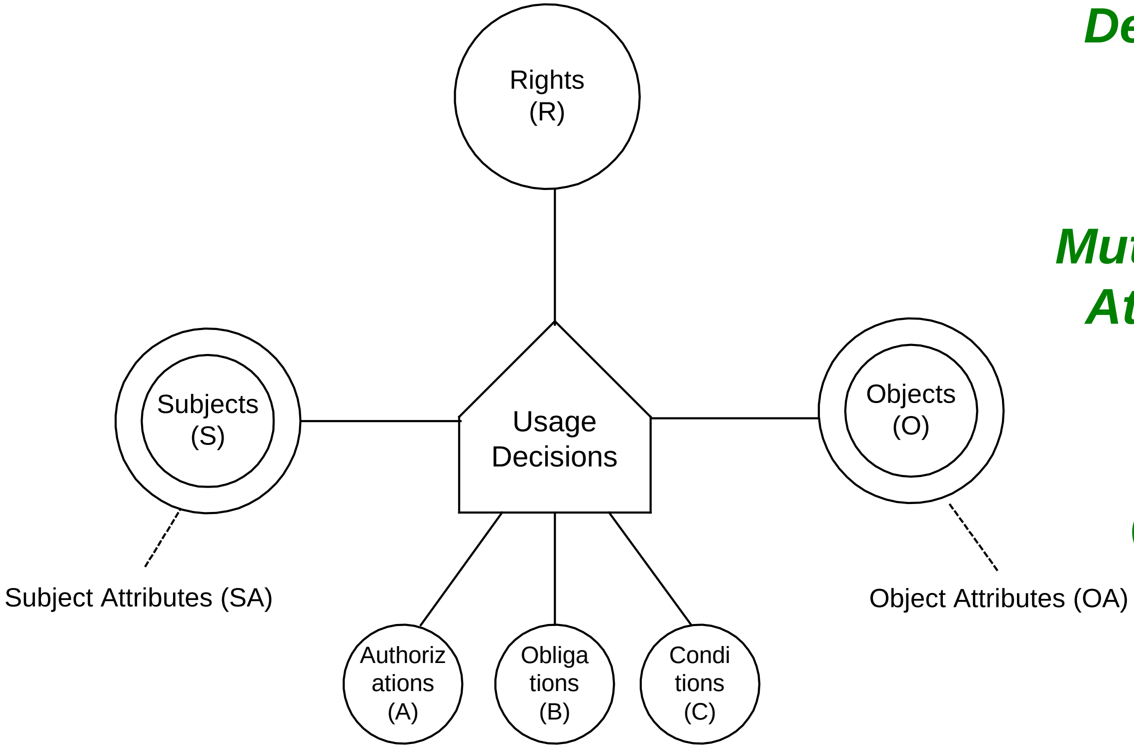
Decision can be made during usage for continuous enforcement

Mutability

Attributes can be updated as side-effects of subjects' actions

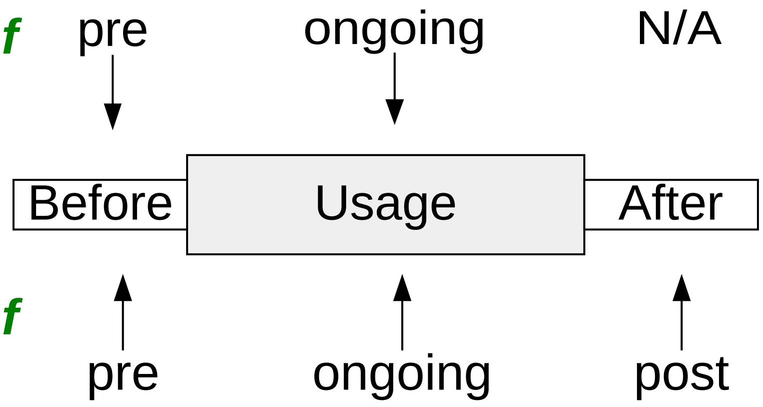
World-Leading Research with Real-World Impact!

Building UCON_{ABC} Models



Continuity of Decisions

Mutability of Attributes



Continuity

Decision can be made during usage for continuous enforcement

Mutability

Attributes can be updated as side-effects of subjects’ actions

Examples

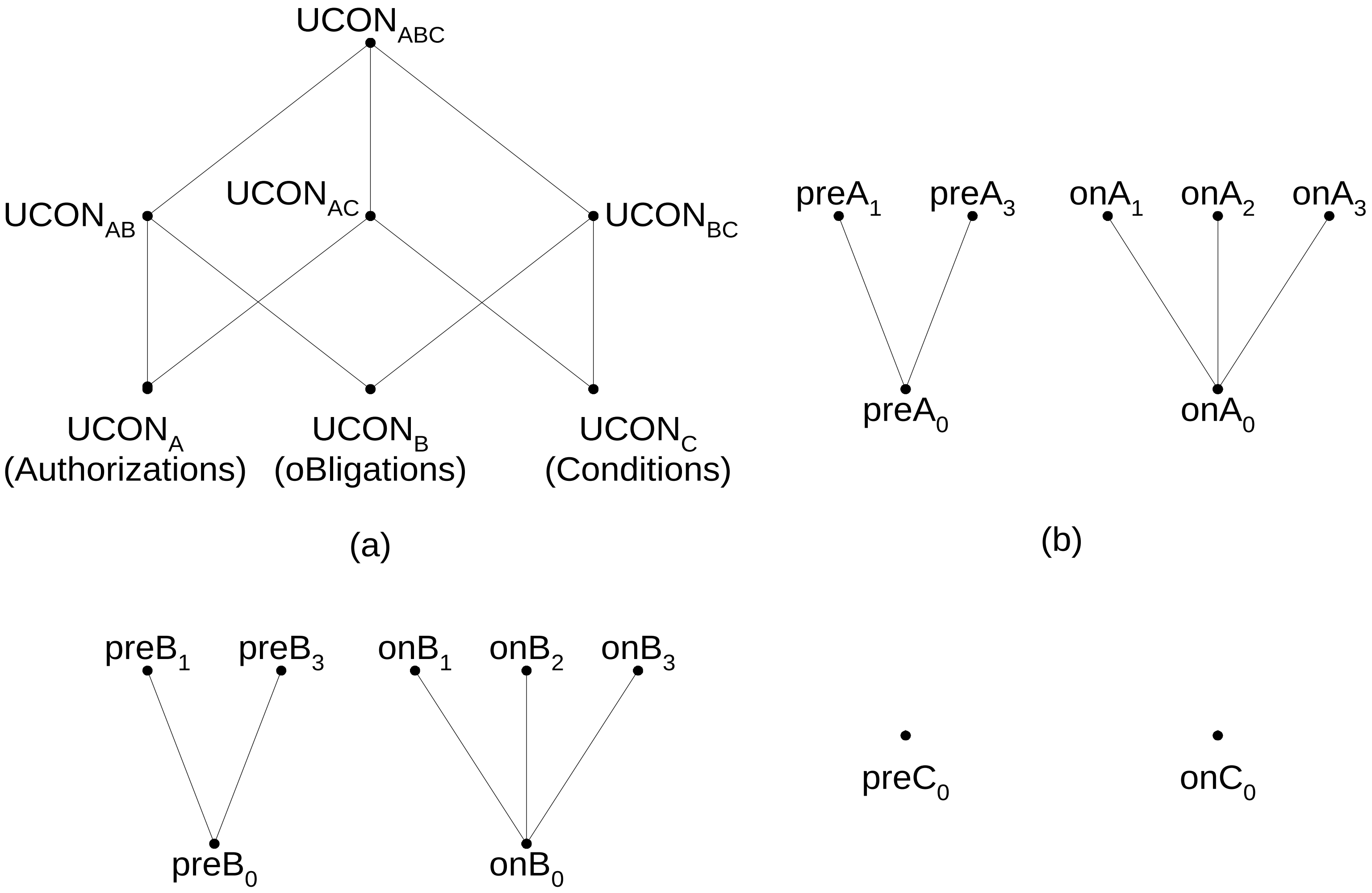
- Long-distance phone (pre-authorization with post-update)
- Pre-paid phone card (ongoing-authorization with ongoing-update)
- Pay-per-view (pre-authorization with pre-updates)
- Click Ad every 30 minutes (ongoing-obligation with ongoing-updates)
- Business Hours (pre-/ongoing-condition)

UCON_{ABC} Model Space

	0(Immutable)	1(pre)	2(ongoing)	3(post)
preA	Y	Y	N	Y
onA	Y	Y	Y	Y
preB	Y	Y	N	Y
onB	Y	Y	Y	Y
preC	Y	N	N	N
onC	Y	N	N	N

N : Not applicable

A Family of UCON$_{ABC}$ Core Models



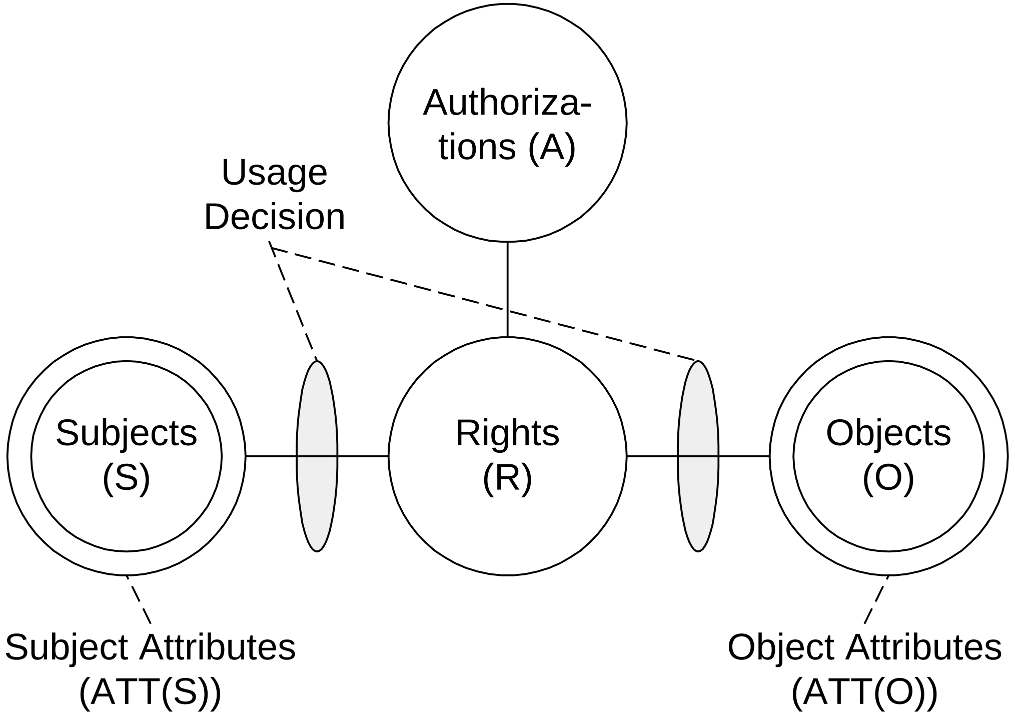
(a)

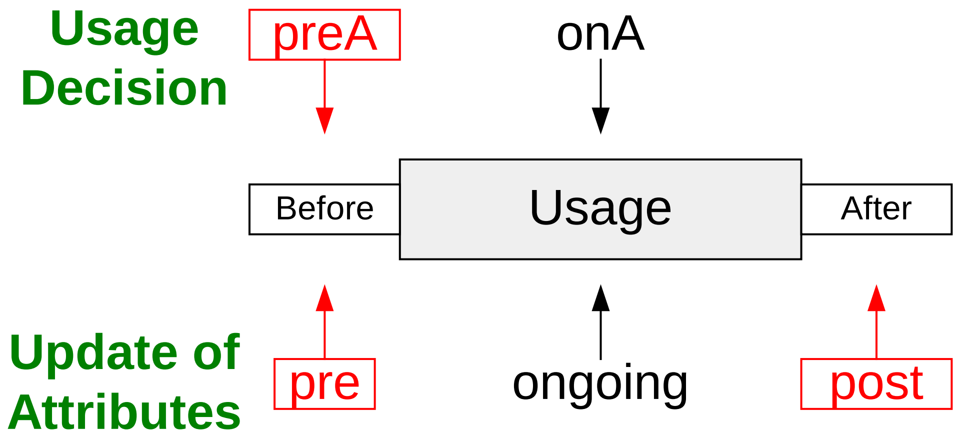
(b)

(c)

(d)

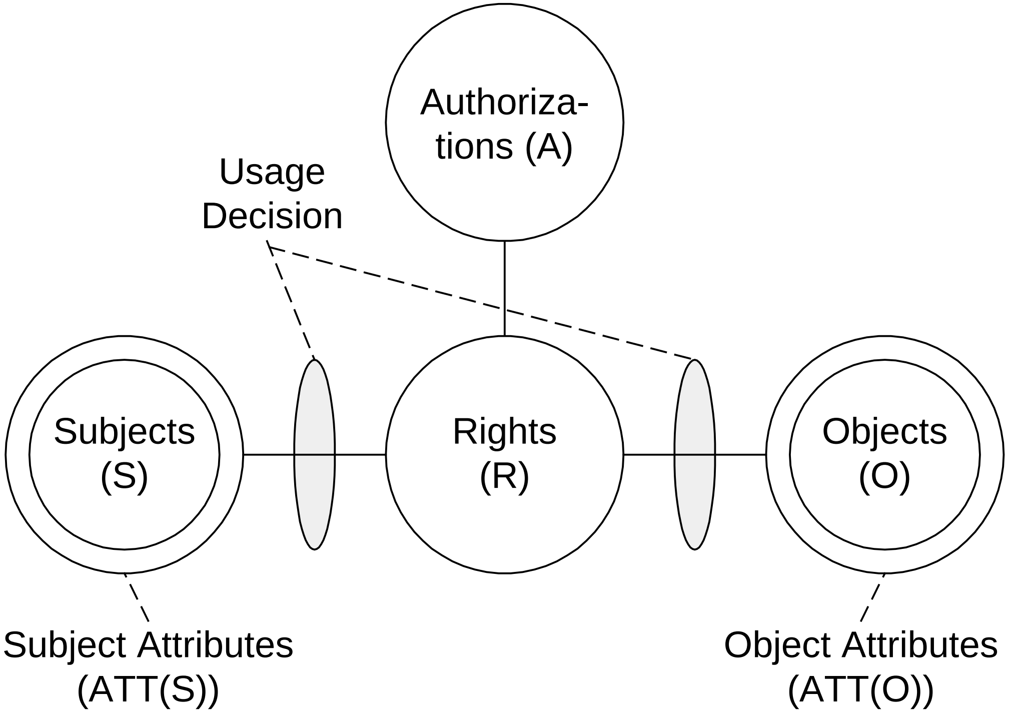
UCON$_{preA}$

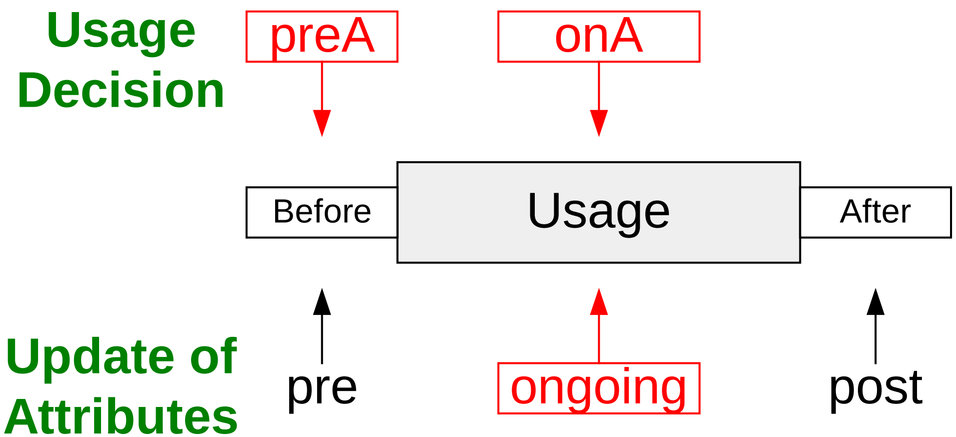




- Online content distribution service
  - Pay-per-view (pre-update)
  - Metered payment (post-update)

UCON_{onA}





- Pay-per-minutes (pre-paid Phone Card)

UCON$_{preA}$: pre-Authorizations Model

- UCON$_{preA0}$
  - *S, O, R, ATT(S), ATT(O)* and *preA* (subjects, objects, rights, subject attributes, object attributes, and pre-authorizations respectively);
  - *allowed(s,o,r)* $\Rightarrow$ *preA(ATT(s),ATT(o),r)*
- UCON$_{preA1}$
  - *preUpdate(ATT(s)),preUpdate(ATT(o))*
- UCON$_{preA3}$
  - *postUpdate(ATT(s)),postUpdate(ATT(o))*

UCON$_{\text{preA0}}$: MAC Example

- *L* is a lattice of security labels with dominance relation $\geq$
- *clearance:* $S \rightarrow L$
- *classification:* $O \rightarrow L$
- $ATT(S) = \{clearance\}$
- $ATT(O) = \{classification\}$
- $allowed(s,o,read) \Rightarrow clearance(s) \geq classification(o)$
- $allowed(s,o,write) \Rightarrow clearance(s) \leq classification(o)$

DAC in UCON: *with ACL ($UCON_{preA0}$)*

- *N* is a set of identity names
- $id : S \rightarrow N$, one to one mapping
- $ACL : O \rightarrow 2^{N \times R}$, *n* is authorized to do *r* to *o*
- $ATT(S) = \{id\}$
- $ATT(O) = \{ACL\}$
- $allowed(s,o,r) \Rightarrow (id(s),r) \in ACL(o)$

RBAC in UCON: *RBAC₁ (UCON$_{preA0}$)*

- $P = \{(o,r)\}$
- *ROLE* is a partially ordered set of roles with dominance relation $\geq$
- $actRole: S \rightarrow 2^{ROLE}$
- $Prole: P \rightarrow 2^{ROLE}$
- $ATT(S) = \{actRole\}$
- $ATT(O) = \{Prole\}$
- $allowed(s,o,r) \Rightarrow \exists role \in actRole(s), \exists role' \in Prole(o,r), role \geq role'$

DRM in UCON: *Pay-per-use with pre-paid credit* (*UCON*$_{preA1}$)

- *M* is a set of money amounts
- *credit*: $S \rightarrow M$
- *value*: $O \times R \rightarrow M$
- *ATT(s): {credit}*
- *ATT(o,r): {value}*
- *allowed(s,o,r)* $\Rightarrow$ *credit(s)* $\geq$ *value(o,r)*
- *preUpdate(credit(s)): credit(s) = credit(s) - value(o,r)*

UCON$_{\text{preA3}}$: DRM Example

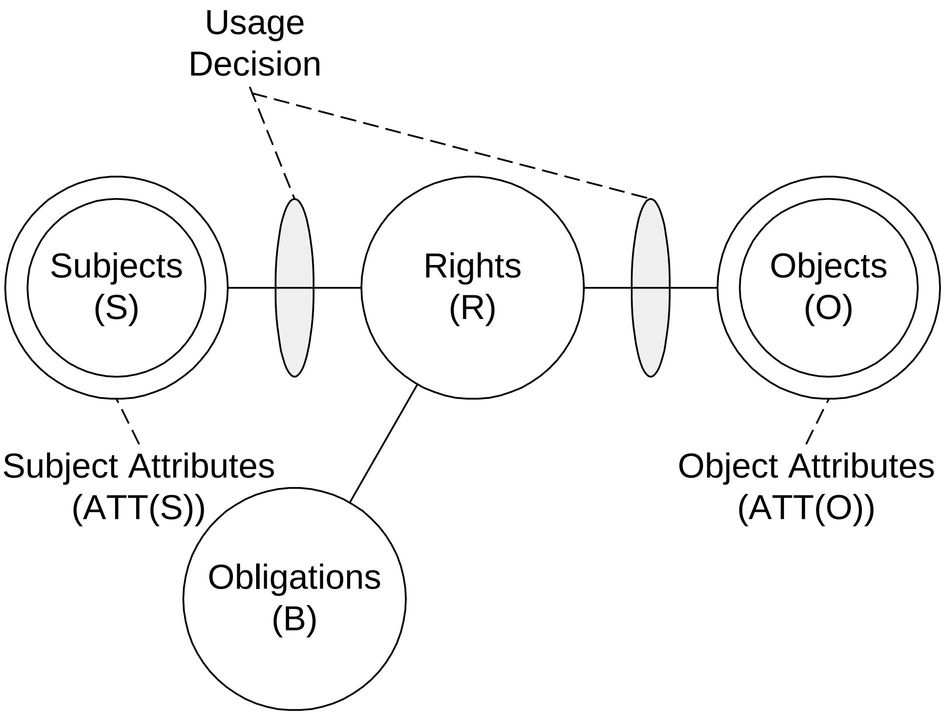
- Membership-based metered payment
  - M is a set of money amount
  - ID is a set of membership identification numbers
  - $TIME$ is a current usage minute
  - $member: S \rightarrow ID$
  - $expense: S \rightarrow M$
  - $usageT: S \rightarrow TIME$
  - $value: O \times R \rightarrow M$ (a cost per minute of r on o)
  - $ATT(s): \{member, expense, usageT\}$
  - $ATT(o,r): \{valuePerMinute\}$
  - $allowed(s,o,r) \Rightarrow member(s) \neq \varnothing$
  - $postUpdate(expense(s)): expense(s) = expense(s) + (value(o,r) \times usageT(s))$

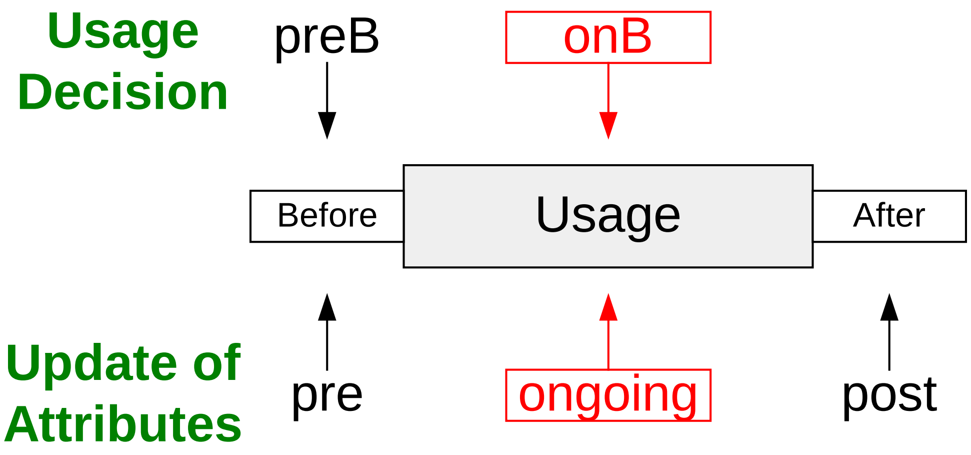
UCON$_{onA}$: ongoing-Authorizations Model

- UCON$_{onA0}$
  - *S, O, R, ATT(S), ATT(O)* and *onA*;
  - *allowed(s,o,r)* ⇒ *true;*
  - *Stopped(s,o,r)* ⇐ ¬*onA(ATT(s),ATT(o),r)*
- UCON$_{onA1}$, UCON$_{onA2}$, UCON$_{onA3}$
  - *preUpdate(ATT(s)),preUpdate(ATT(o))*
  - *onUpdate(ATT(s)),onUpdate(ATT(o))*
  - *postUpdate(ATT(s)),postUpdate(ATT(o))*

- Examples
  - Certificate Revocation Lists
  - revocation based on starting time, longest idle time, and total usage time

UCON_B





- Free Internet Service Provider
  - Watch Ad window (no update)
  - Click ad within every 30 minutes (ongoing update)

UCON$_{preB0}$: pre-oBligations w/ no update

- $S, O, R, ATT(S)$, and $ATT(O)$;
- *OBS, OBO* and *OB* (obligation subjects, obligation objects, and obligation actions, respectively);
- *preB* and *preOBL* (pre-obligations predicates and pre-obligation elements, respectively);
- $preOBL \subseteq OBS \times OBO \times OB$;
- $preFulfilled: OBS \times OBO \times OB \rightarrow \{true, false\}$;
- $getPreOBL: S \times O \times R \rightarrow 2^{preOBL}$, a function to select pre-obligations for a requested usage;
- $preB(s,o,r) = \Lambda_{(obs\_i,obo\_i,ob\_i) \in getPreOBL(s,o,r)} preFulfilled(obs_i,obo_i,ob_i);$
- $preB(s,o,r) = true$ by definition if $getPreOBL(s,o,r)=\varnothing$;
- $allowed(s,o,r) \Rightarrow preB(s,o,r)$.

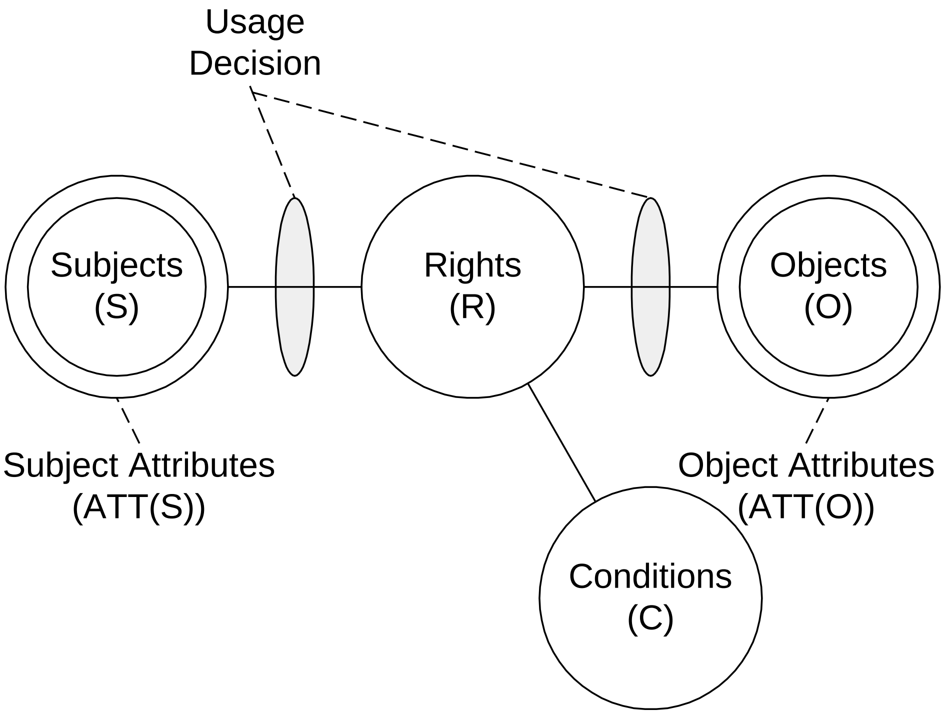
- Example: License agreement for a whitepaper download

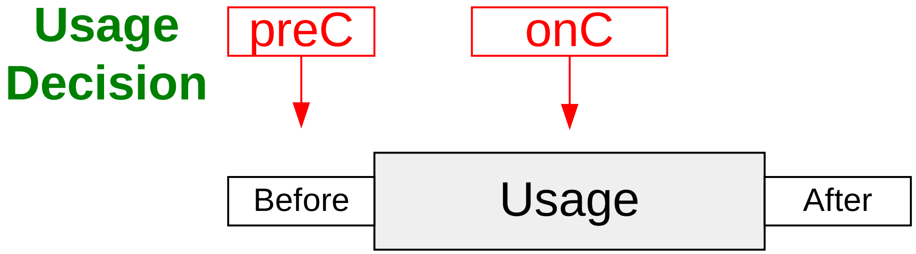
$UCON_{onB0}$: ongoing-oBligations w/ no update

- *S, O, R, ATT(S), ATT(O), OBS, OBO* and *OB*;
- *T*, a set of time or event elements;
- *onB* and on*OBL* (on-obligations predicates and ongoing-obligation elements, respectively);
- $onOBL \subseteq OBS \times OBO \times OB \times T$;
- $onFulfilled: OBS \times OBO \times OB \times T \rightarrow \{true, false\}$;
- $getOnOBL: S \times O \times R \rightarrow 2^{onOBL}$, a function to select ongoing-obligations for a requested usage;
- $onB(s,o,r) = \Lambda_{(obs\_i, obo\_i, ob\_i,\ t\_i)\ \in\ getOnOBL(s,o,r)}\ onFulfilled(obs_i, obo_i, ob_i, t_i)$;
- $onB(s,o,r) = true$ by definition if $getOnOBL(s,o,r) = \varnothing$;
- $allowed(s,o,r) \Rightarrow true$;
- $Stopped(s,o,r) \Leftarrow \neg\, onB(s,o,r)$.

- Example: Free ISP with mandatory ad window

UCON$_C$





Update of Attributes: No-Update is possible

- Location check at the time of access request
- Accessible only during business hours

$UCON_{preC0}$: pre-Condition model

- *S, O, R, ATT(S)*, and *ATT(O)*;
- *preCON* (a set of pre-condition elements);
- *preConChecked: preCON* $\rightarrow$ *{true,false}*;
- *getPreCON: S x O x R* $\rightarrow 2^{preCON}$;
- $preC(s,o,r) = \Lambda_{preCon\_i \in getPreCON(s,o,r)} preConChecked(preCon_i)$;
- *allowed(s,o,r)* $\Rightarrow$ *preC(s,o,r)*.

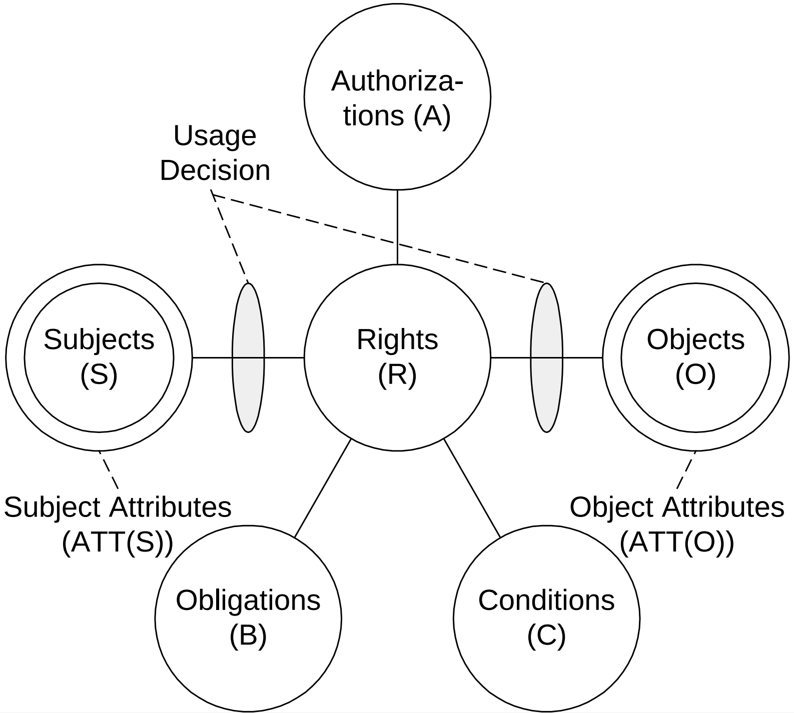
- Example: location checks at the time of access requests

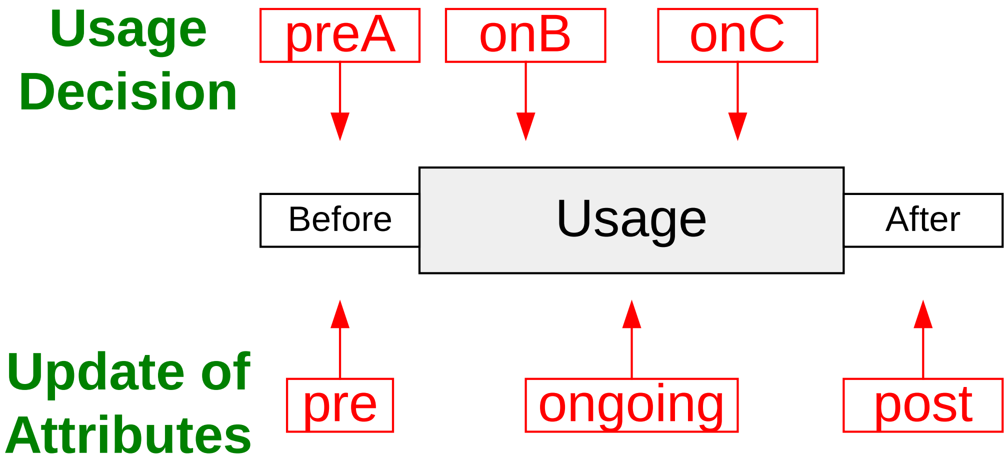
UCON$_{onC0}$: ongoing-Condition model

- *S, O, R, ATT(S)*, and *ATT(O)*;
- *onCON* (a set of on-condition elements);
- $onConChecked: onCON \rightarrow \{true, false\}$;
- $getOnCON: S \times O \times R \rightarrow 2^{onCON}$;
- $onC(s,o,r) = \Lambda_{onCon\_i \in getOnCON(s,o,r)}\, onConChecked(onCon_i)$;
- $allowed(s,o,r) \Rightarrow true$;
- $Stopped(s,o,r) \Leftarrow \neg onC(s,o,r)$

- Example: accessible during office hour

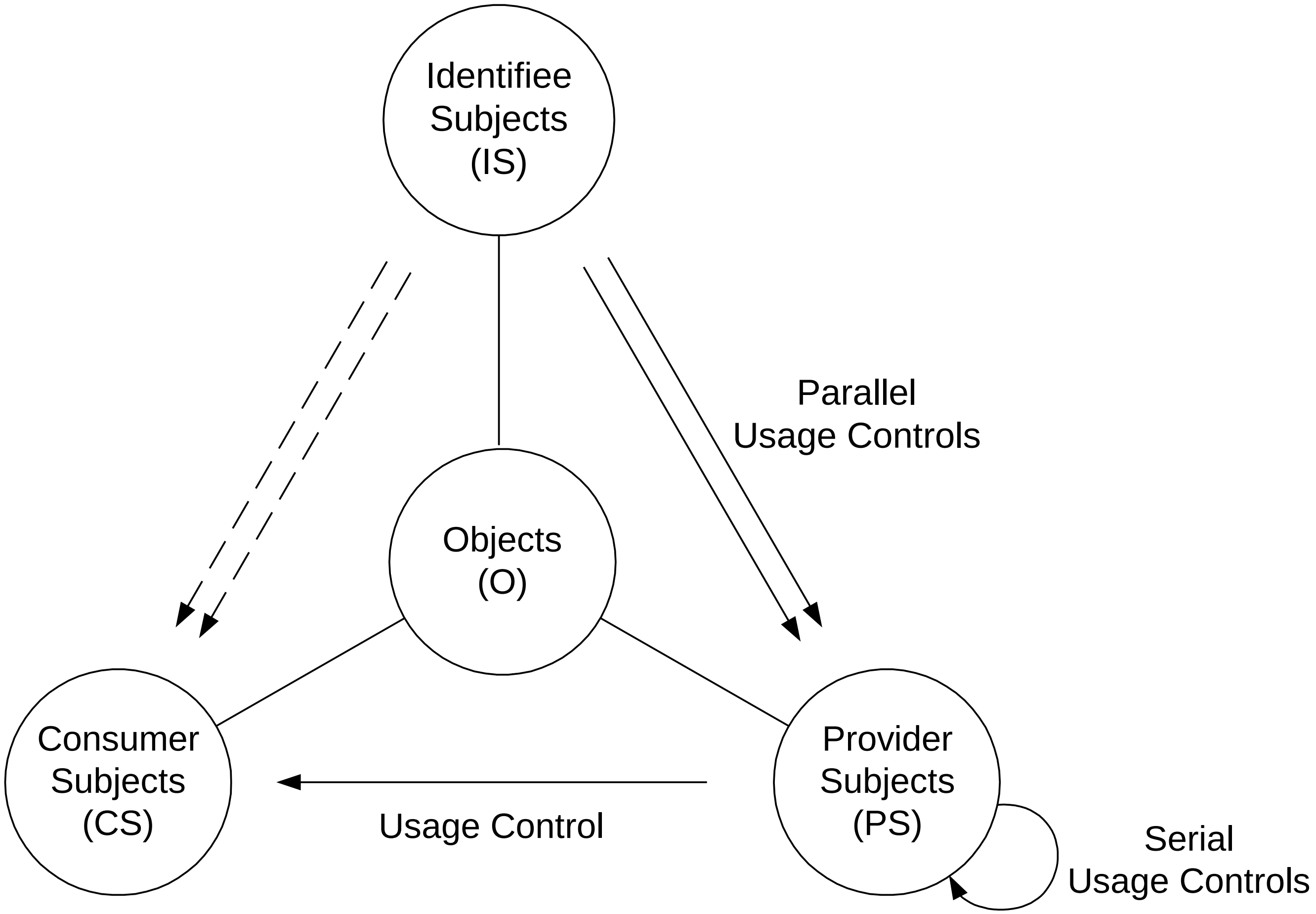
UCON$_{ABC}$





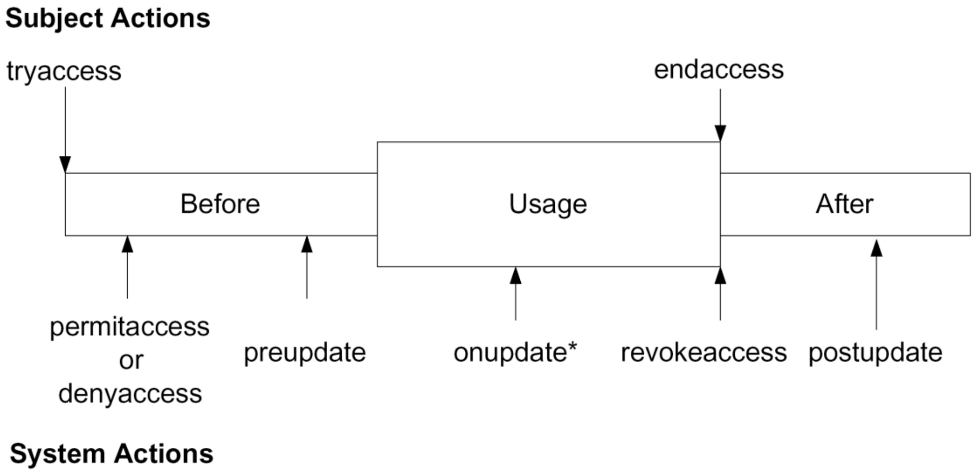
- Free ISP
  - Membership is required (pre-authorization)
  - Have to click Ad periodically while connected (on-obligation, on-update)
  - Free member: no evening connection (on-condition), no more than 50 connections (pre-update) or 100 hours usage per month (post-updates)

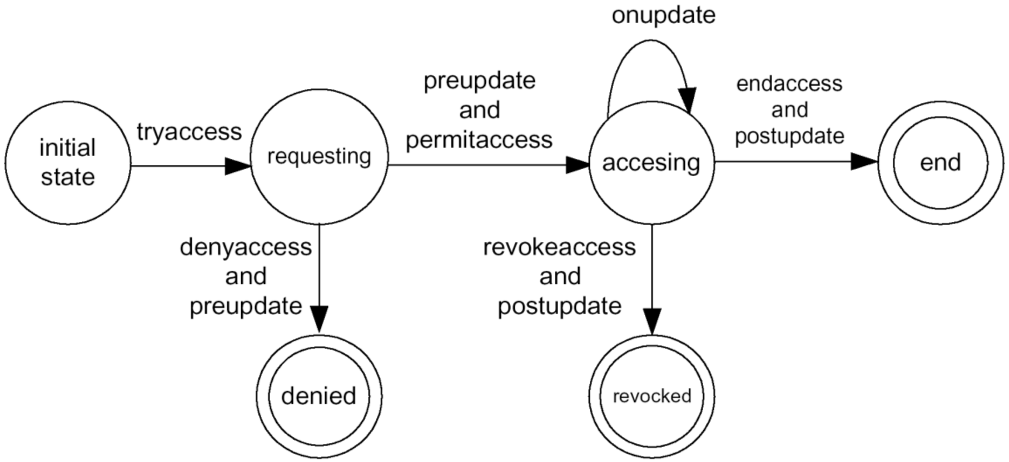
Beyond the UCON_{ABC} Core Models



Logic Model of UCON

- Actions: boolean expressions built from attributes in two states.
  - Alice.credit'=Alice.credit - $50.0
- Two types of actions:
  - Control actions: change the state of single usage process
    - Actions performed by the subject
    - Actions performed by the system
  - Obligation actions:
    - Actions that have to be performed before or during an access.
    - May or may not be performed by the requesting subject and on the target object.





Summary of UCON

- Coined the concept of Usage Control for modern computing systems
- Developed A family of UCON_{ABC} core models for Usage Control (UCON) to unify *traditional access control models, DRM*, and other modern enhanced models.
- UCON_{ABC} model integrates *authorizations, obligations, conditions*, as well as *continuity* and *mutability* properties.

Discuss Pretschner 2006 paper