I·C·S
The Institute for Cyber Security

Institute for Cyber Security

UTSA

Access, Usage and Activity Controls

Mar. 30, 2012
UTSA CS6393

Jaehong Park
Institute for Cyber Security
University of Texas at San Antonio
jae.park@utsa.edu

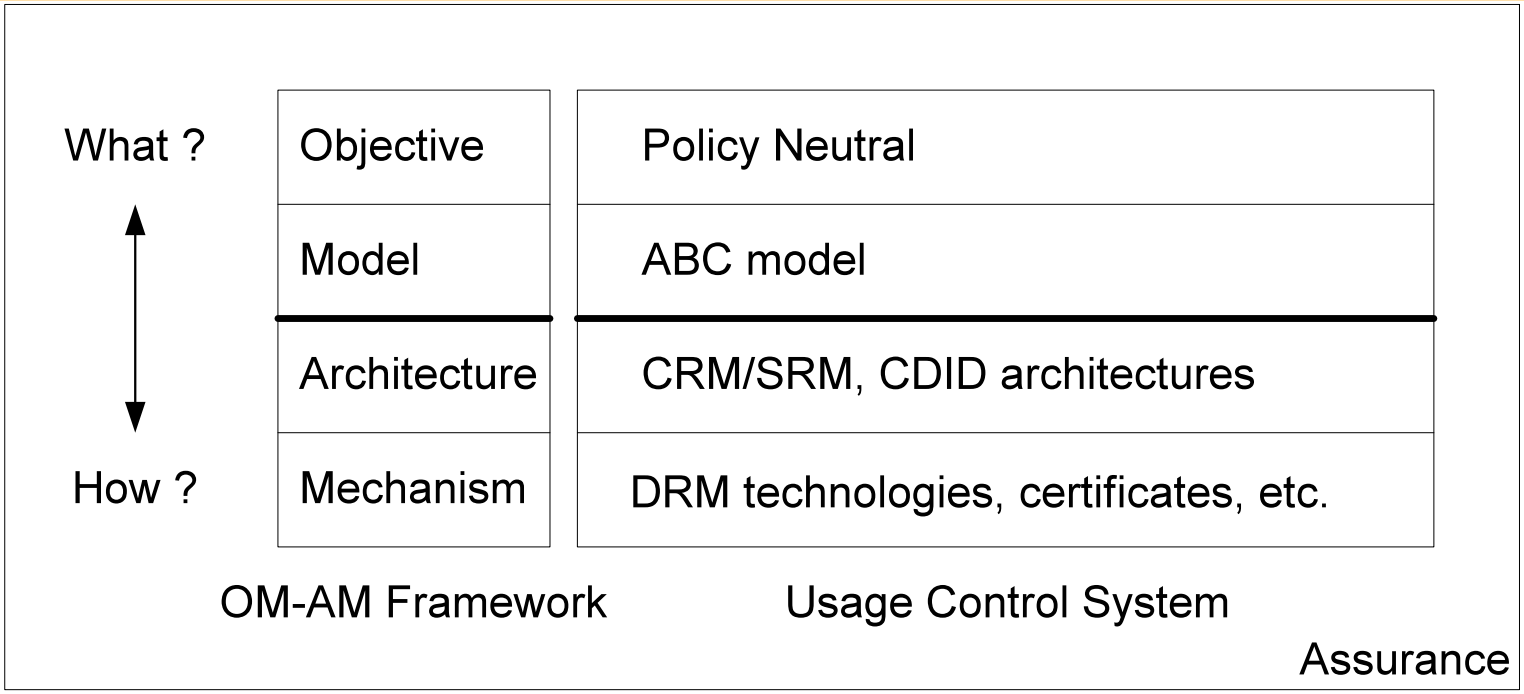
"You spelled 'confidential' wrong."

- Who's the victim?
- Problems/Concerns?
- Our assumptions?
- Necessary access controls?

Access Control Considerations

- Objectives
  - Security, Privacy, Intellectual Property Rights Protection and trustworthiness
- Target domain
  - System-level, application-level
- Access Target
  - System resource, data, user, policies and attributes
- Access Types
  - Access, usage, activity

OM-AM layered Approach



Model examples: Access Matrix, Lattice-based model, Role-base access control model

ABC core model for UCON

I·C·S
The Institute for Cyber Security

UTSA

Usage Control (UCON)

Motivation (1)

- 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

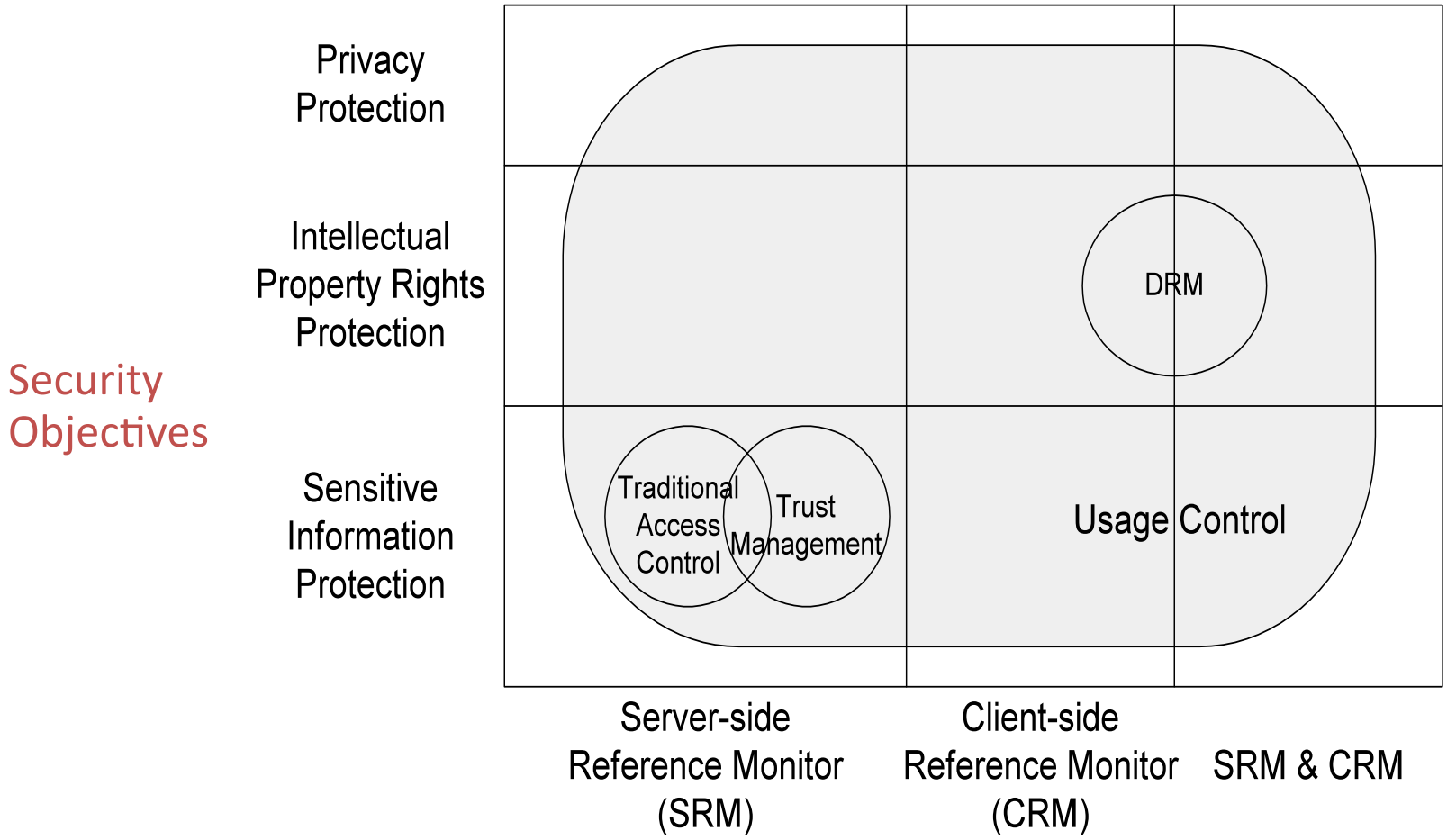
Motivation (2)

- 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

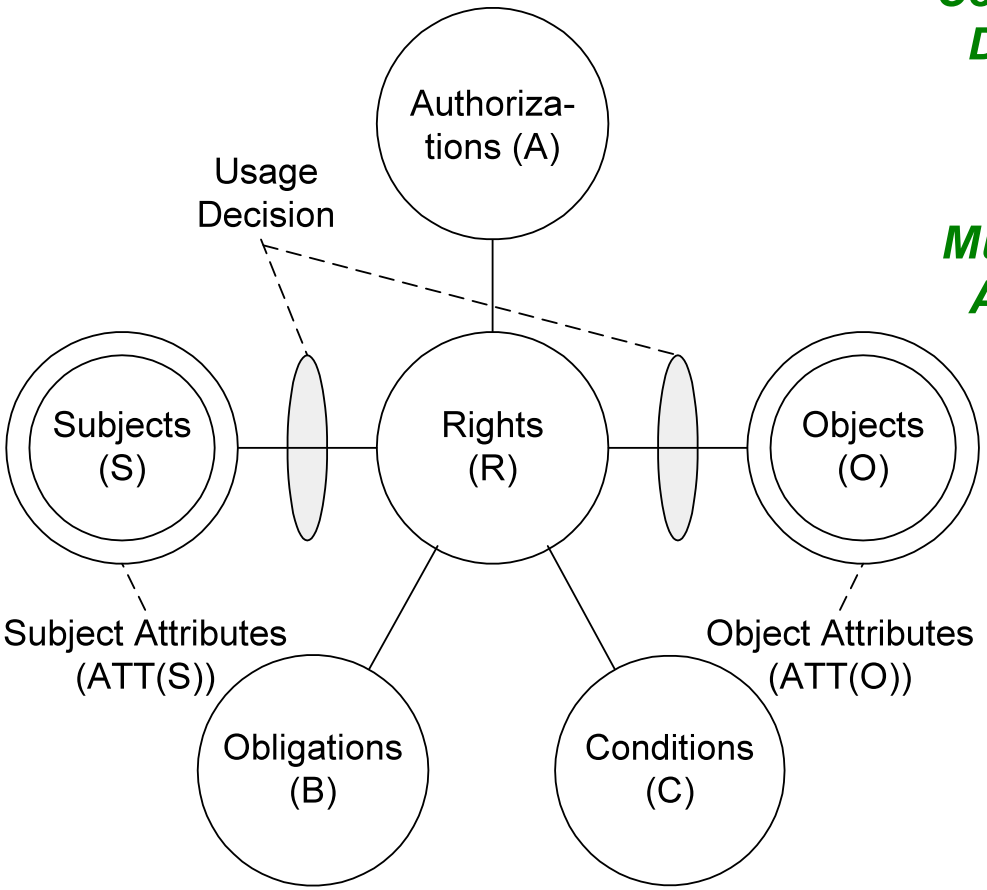
Usage Control (UCON)

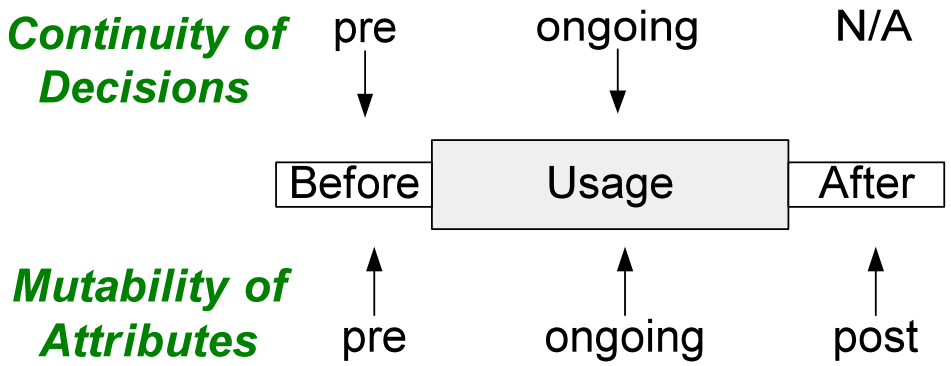
- Scope
  - Encompass traditional access controls, trust management, digital rights management and more
  - For sensitive information protection, IPR protection, and privacy protection
- Model
  - General purpose, policy neutral models
  - Policy is assumed to be given to the system
  - Transaction based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

Usage Control (UCON) Coverage



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

Examples

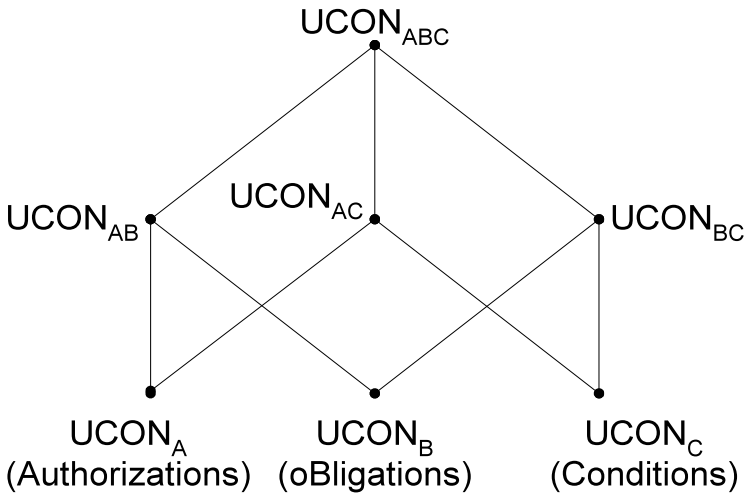
- Long-distance phone (pre-authorization with post-update)
- Pre-paid phone card (ongoing-authorization with ongoing-update)
- Pay-per-view (pre-authorization with pre-updates)
- Click Ad within every 30 minutes (ongoing-obligation with ongoing-updates)
- Business Hour (pre-/ongoing-condition)

$UCON_{ABC}$ Model Space

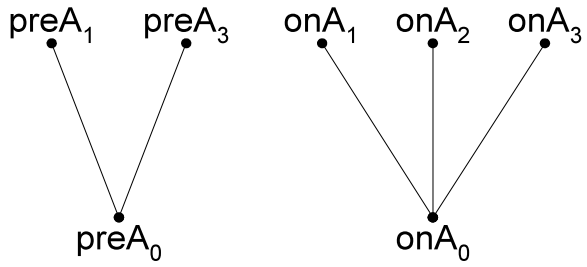
	0(Immutable)	1(pre)	2(ongoing)	3(post)
preA	Y	Y	N	Y
onA	Y	Y	Y	Y
preB	Y	Y	N	Y
onB	Y	Y	Y	Y
preC	Y	N	N	N
onC	Y	N	N	N

N : Not applicable

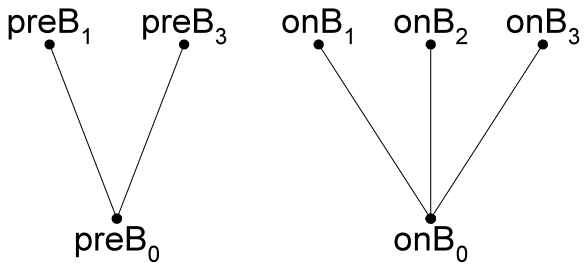
A Family of $UCON_{ABC}$ Core Models



(a)



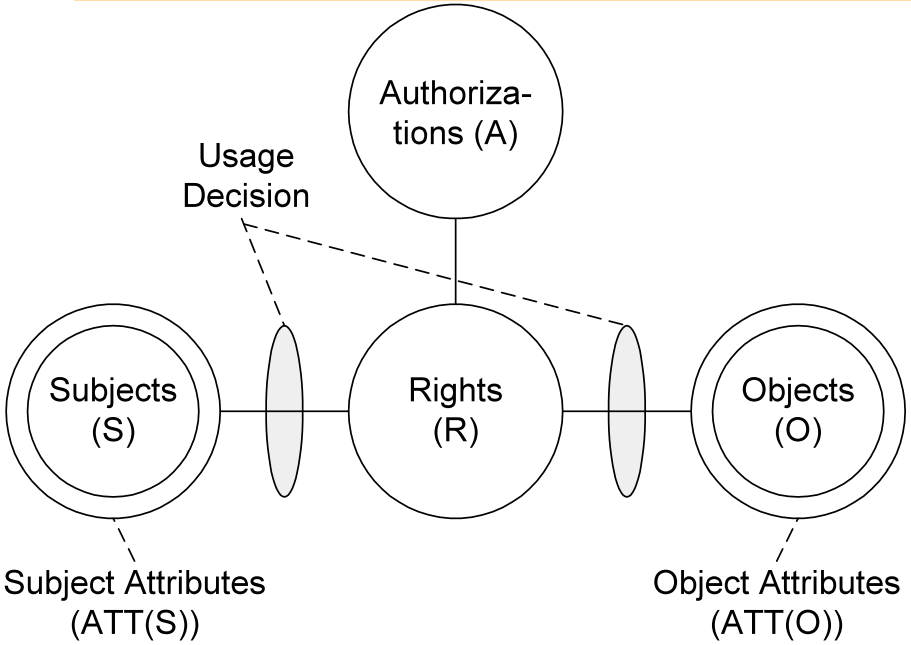
(b)

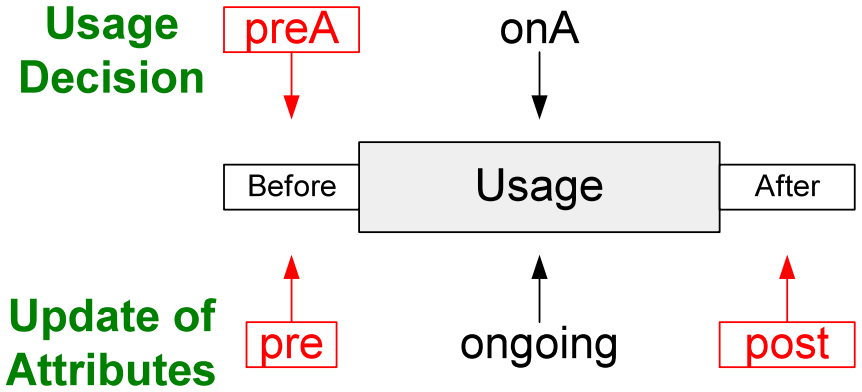


(c)

(d)

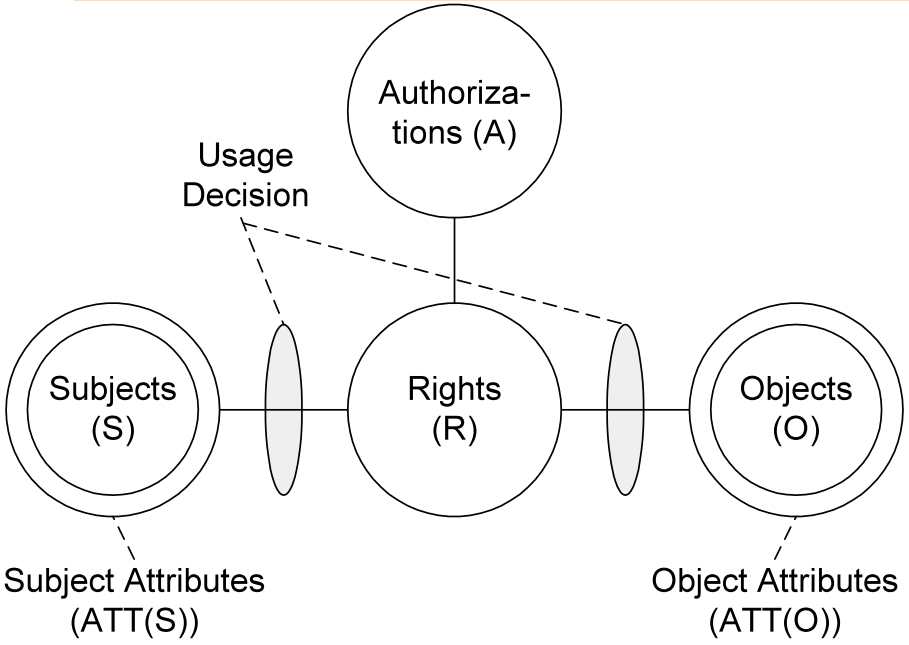
$UCON_{preA}$

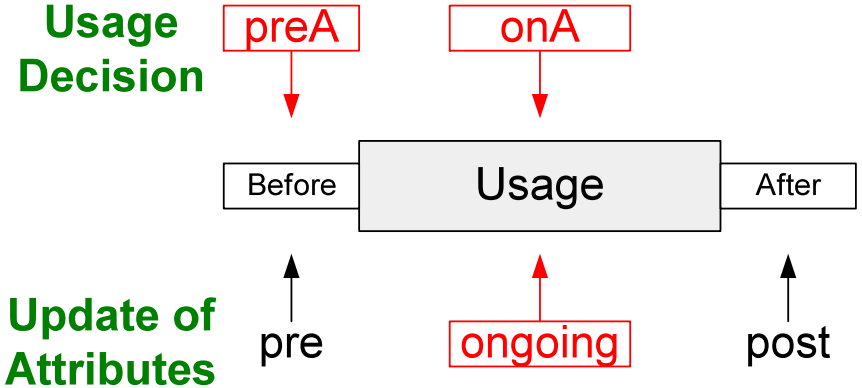




- Online content distribution service
  - Pay-per-view (pre-update)
  - Metered payment (post-update)

$UCON_{onA}$





- 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))*

$UCON_{preA0}$: MAC Example

- *L* is a lattice of security labels with dominance relation ≥
- *clearance: S → L*
- *classification: O → L*
- *ATT(S) = {clearance}*
- *ATT(O) = {classification}*
- *allowed(s,o,read) ⇒ clearance(s) ≥ classification(o)*
- *allowed(s,o,write) ⇒ clearance(s) ≤ 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: $RBAC_1$ ($UCON_{preA0}$)

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

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

- *M* is a set of money amount
- *credit: S → M*
- *value: O x R → M*
- *ATT(s): {credit}*
- *ATT(o,r): {value}*
- *allowed(s,o,r) ⇒ credit(s) ≥ value(o,r)*
- *preUpdate(credit(s)): credit(s) = credit(s) - value(o,r)*

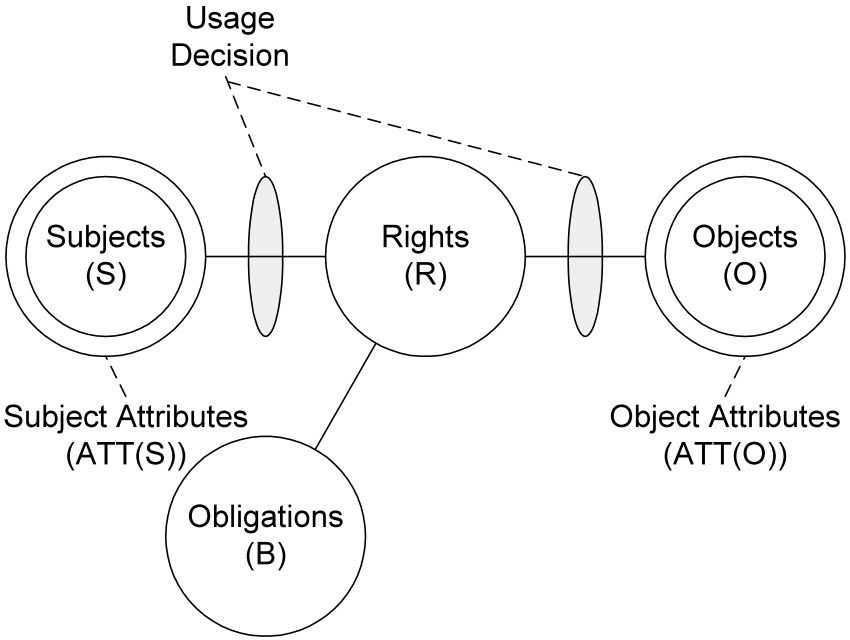
$UCON_{preA3}$: DRM Example

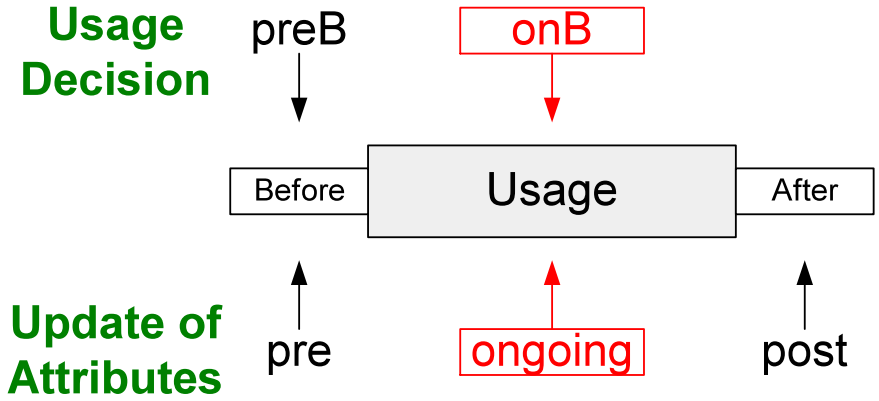
- Membership-based metered payment
  - *M* is a set of money amount
  - *ID* is a set of membership identification numbers
  - *TIME* is a current usage minute
  - *member: S → ID*
  - *expense: S → M*
  - *usageT: S → TIME*
  - *value: O x R → M* (a cost per minute of r on o)
  - *ATT(s): {member, expense, usageT}*
  - *ATT(o,r): {valuePerMinute}*
  - *allowed(s,o,r) ⇒ member(s) ≠ ∅*
  - *postUpdate(expense(s)): expense(s) = expense(s) + (value(o,r) x usageT (s))*

$UCON_{onA}$: ongoing-Authorizations Model

- $UCON_{onA0}$
  - *S, O, R, ATT(S), ATT(O)* and *onA*;
  - *allowed(s,o,r)* $\Rightarrow$ *true;*
  - *Stopped(s,o,r)* $\Leftarrow \neg$*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;
- $preB(s,o,r) = \bigwedge_{(obs\_i,obo\_i,ob\_i) \in getPreOBL(s,o,r)} preFulfilled(obs_i,obo_i,ob_i)$;
- $preB(s,o,r) = true$ by definition if $getPreOBL(s,o,r)=\varnothing$;
- $allowed(s,o,r) \Longrightarrow preB(s,o,r)$.

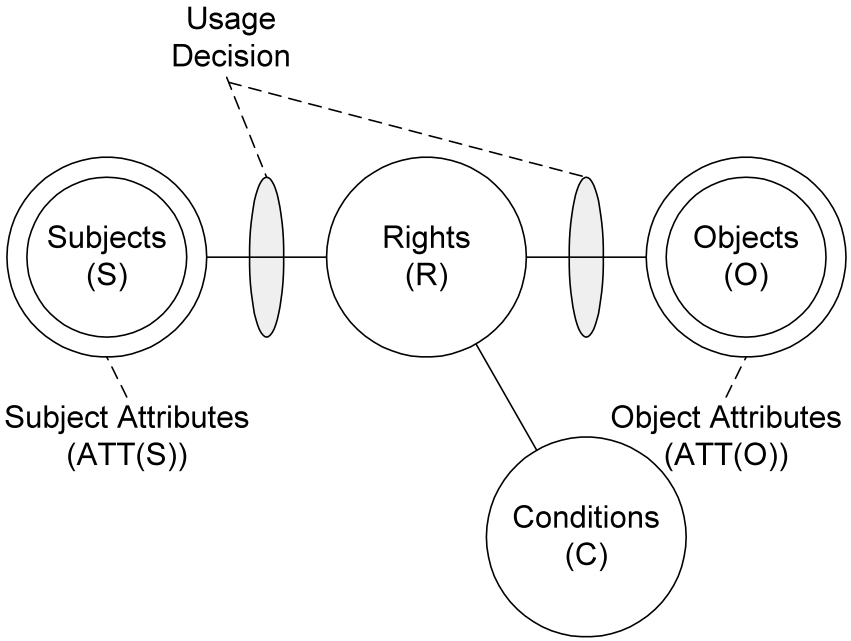
- Example: License agreement for a whitepaper download

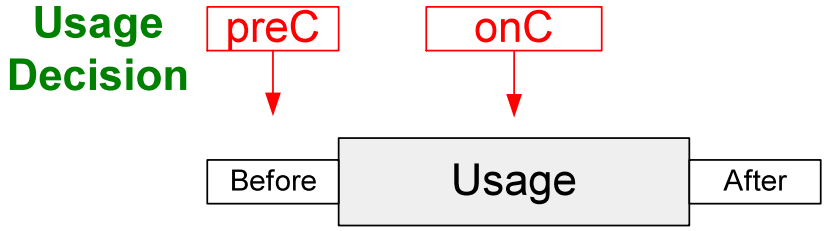
$UCON_{onB0}$: ongoing-oBligations w/ no update

- *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* $\rightarrow 2^{preCON}$;
- $preC(s,o,r) = \Lambda_{preCon\_i \in getPreCON(s,o,r)}\ preConChecked(preCon_i)$;
- $allowed(s,o,r) \Longrightarrow preC(s,o,r)$.

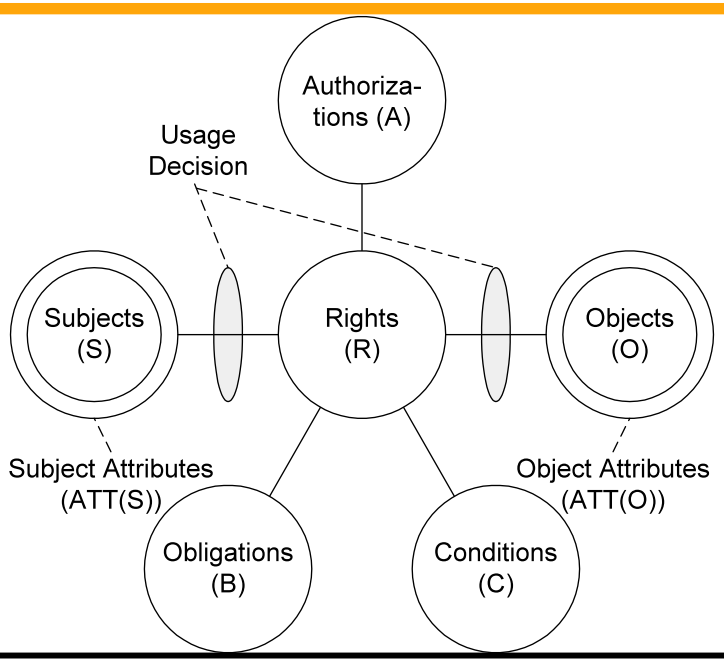
- Example: location checks at the time of access requests

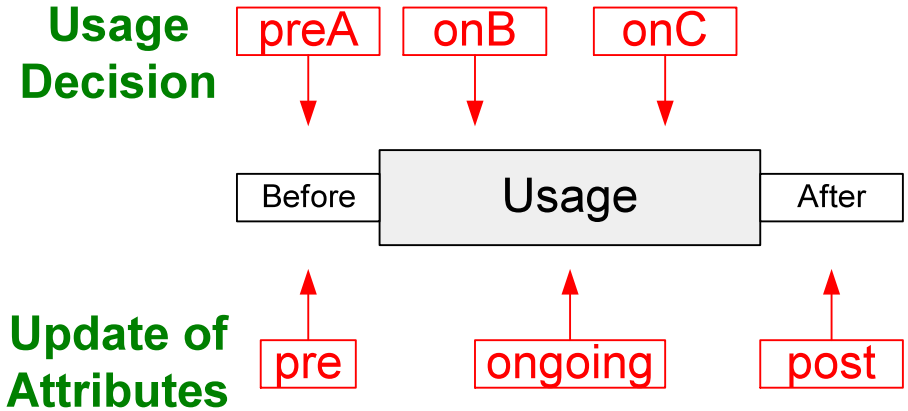
$UCON_{onC0}$: ongoing-Condition model

- *S, O, R, ATT(S),* and *ATT(O)*;
- *onCON* (a set of on-condition elements);
- *onConChecked: onCON* $\rightarrow$ *{true,false}*;
- *getOnCON: S x O x 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

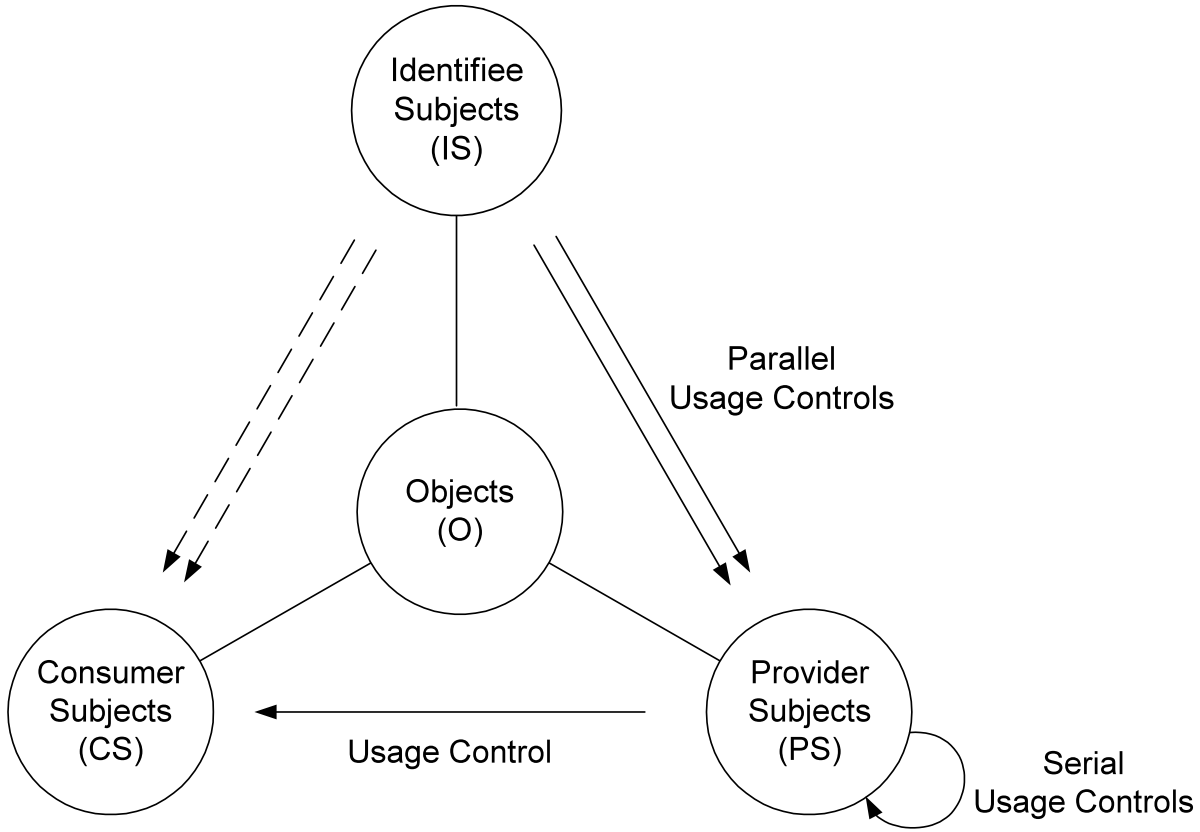
$UCON_{ABC}$





- 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



Summary of UCON

- Coined the concept of Usage Control for modern computing system Environment
- 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.

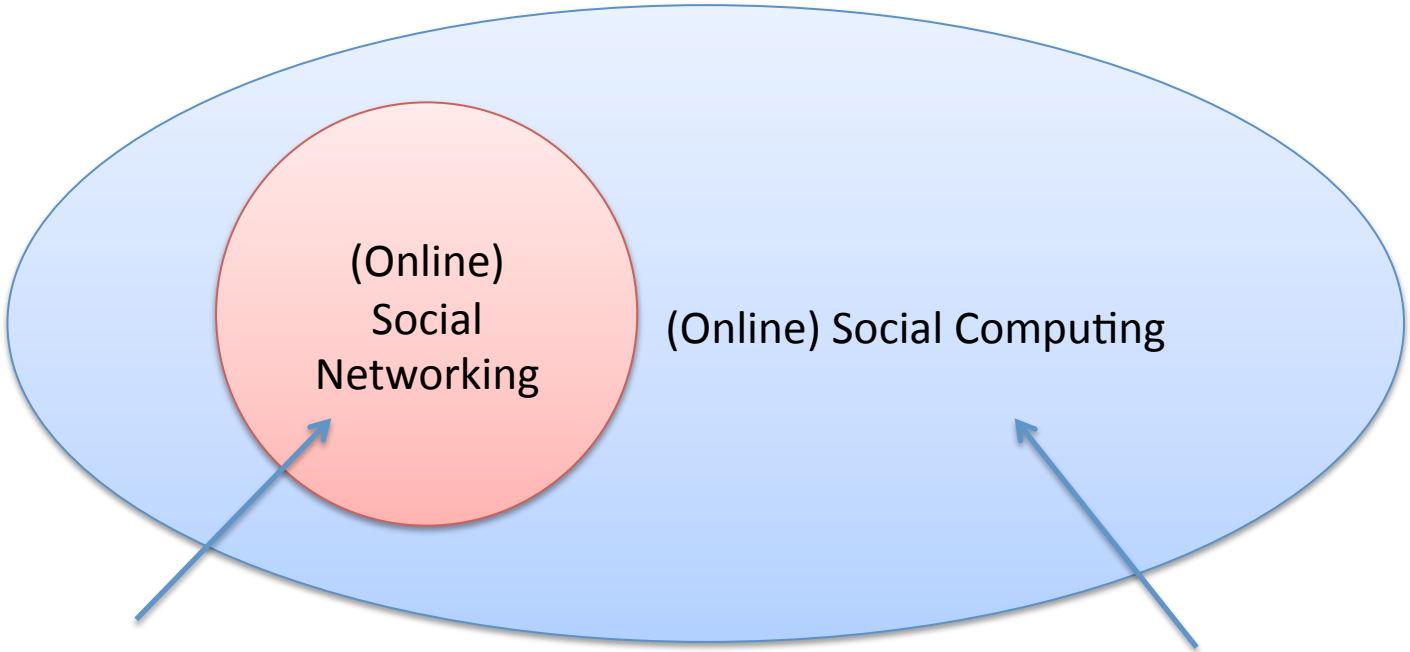
ACON: Activity-Centric Access Control for Social Computing

Social Networking vs. Social Computing

- Social Networking Systems

- Social computing Systems

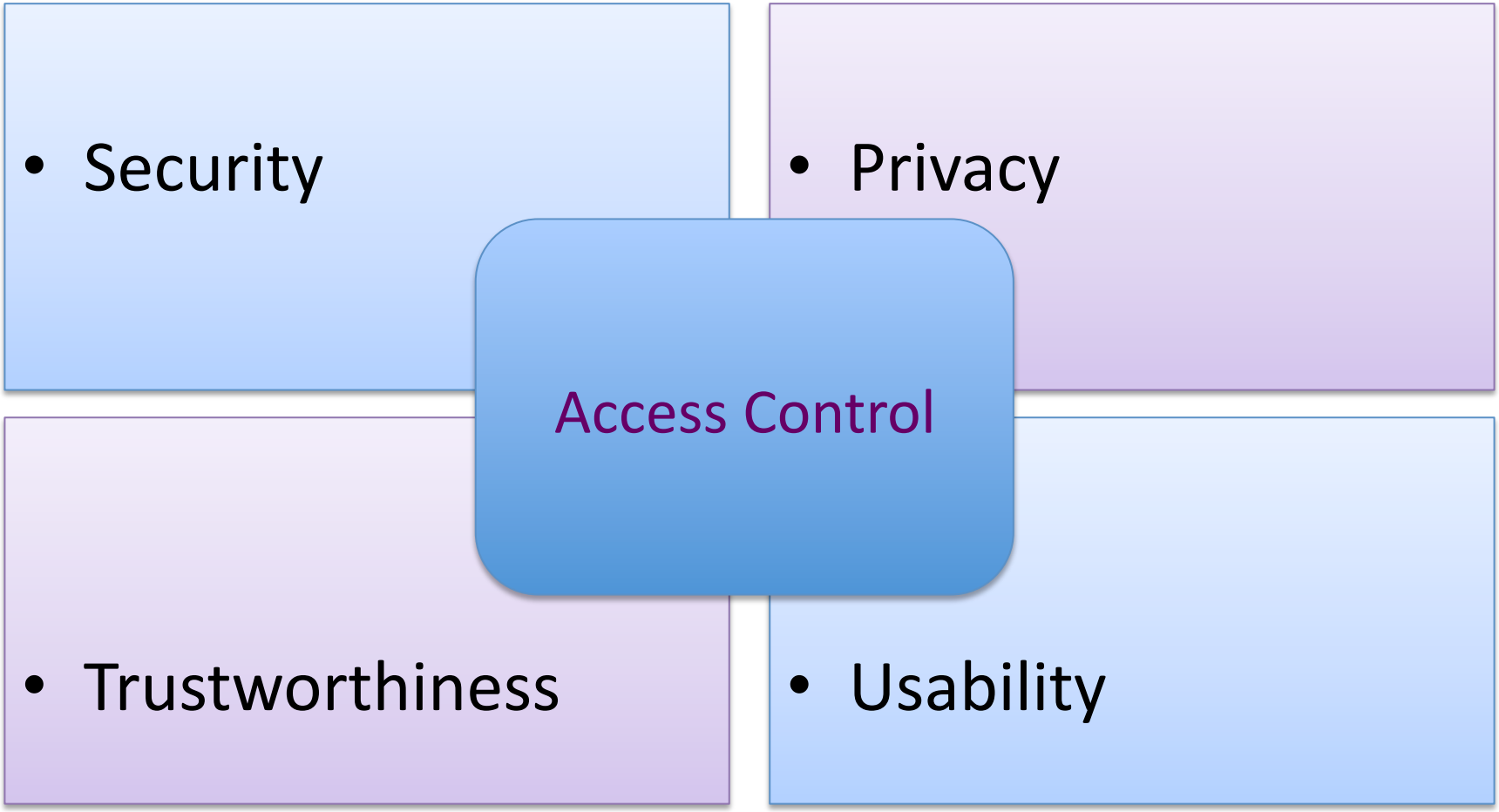
Social Networking vs. Social Computing



•To share 'social' interests by utilizing online social network connections

• To share 'social' interests that may or may not require social network connections

Some Issues in Social Computing



Access Control Considerations

- Objectives
  - Security, Privacy, Intellectual Property Rights Protection, trustworthiness
- Target domain
  - System-level, application-level
- Access Target
  - System resource, data, user
- Access Types
  - Access, usage, activity

Sharing in Social Computing

- Users share with others:
  - knowledge, opinion, interests, information of their (sharing) activities, etc.
- Social computing systems (SCS) provide services to promote information sharing by utilizing user activity information and shared contents
  - Best seller, friends recommendation, friend activity notification, location-based service, etc.

- Both users and SCS provide/access information to be shared
- Both resource and user as a target of activity
  - Alice pokes bob, a buyer rates sellers

Controls in Social Computing

- A user wants to control other user's or SCS's activities against shared information (or users) related to the user
  - My children cannot be a friend of my co-worker
  - System should not notify my activity to my friends
- User wants to protect their privacy
  - Only friends can read my comments
  - Do not notify friends activity to me
  - Do not recommend a friend
- A user's activity influences access control decisions
  - Rating-based popularity

Activities in SCS

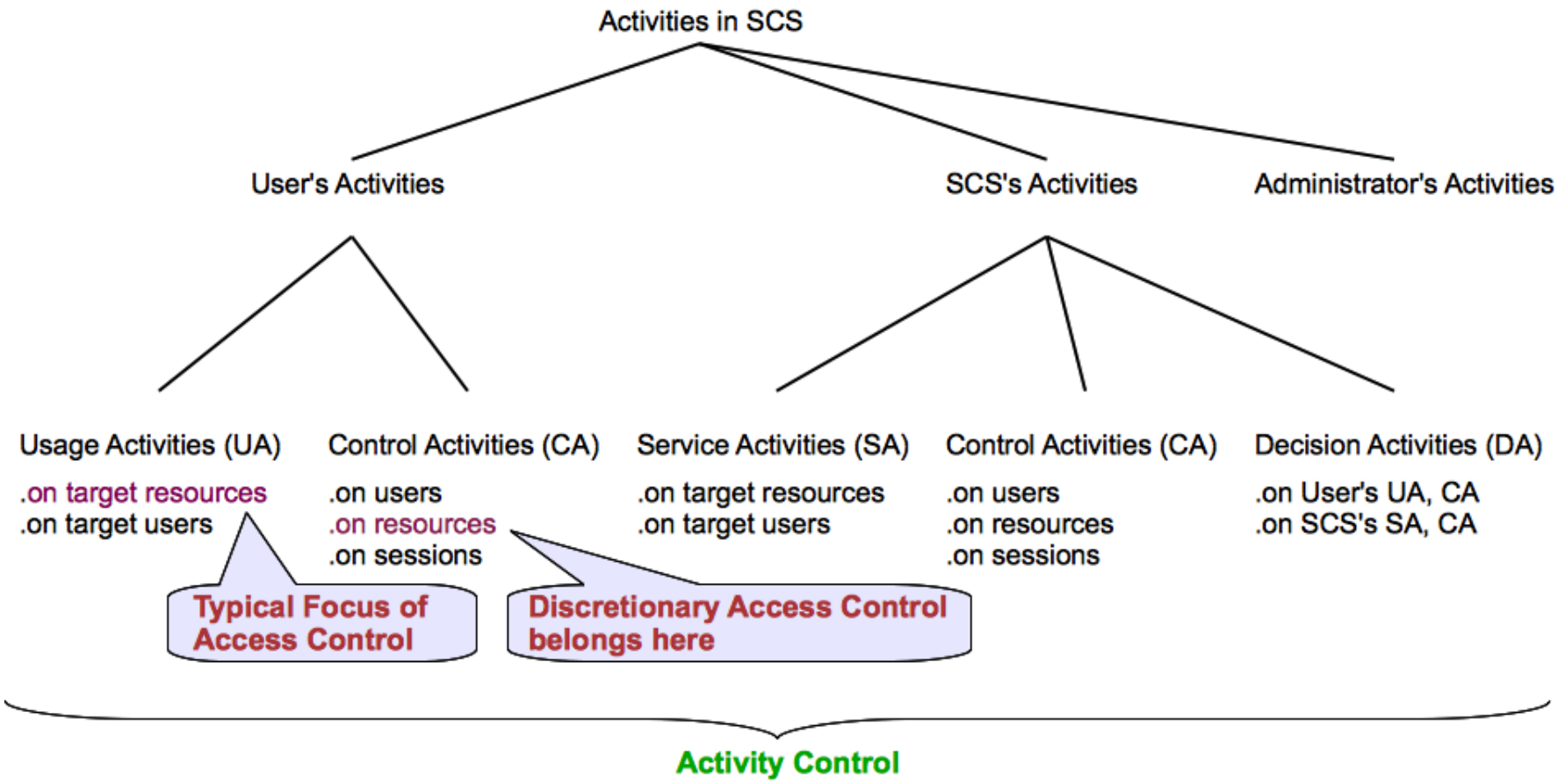
- No traditional access control can provide necessary controls on all kinds of activities found in SCS
- Activity as a key concept for access control

- Why Activity-centric?
  - Multiple kinds of activities (in addition to user's general usage activity against resource) that have to be controlled.
    - User's usage/control activity on user/resource, SCS's service/control activities
  - A user's activities influence SCS's control decisions on own and other users' activities as well as SCS activities.
    - Once Alice invites Bob as a friend, Bob is allowed to see Alice's information
    - If Alice is a friend of Bob and Bob becomes a friend of Chris, 1) if Chris allows friends of friends to his contents, Alice can access Chris's contents; 2) SCS may recommend Chris and Alice as a friend
    - Buyers' ratings on a seller may collectively used to control the seller's sales activity.

Activities in SCS

	User	System
Control	User's Control Activity (UCA)	System's Control Activity (SCA)
Service (create)	User's Service Activity (USA)	System's Service Activity (SSA)
Usage (consume)	User's Usage Activity (UUA)	System's Usage Activity (SUA)

Activity Taxonomy in SCS



User's Usage Activities

- Users' consuming activity on Target Resources
  - Read/view shared comments/photos
  - Typical Focus of Access Control

User's Service Activities

- Users' activity that shares certain information with other users or SCS
  - Add comments/review
  - Rate sellers/products
  - Tag photos w/ users
  - Like a comment
- Service Activity on Target Users
  - Poke a friend

User's Control Activities

- Control Activity on Resources
  - By changing attributes and policies of resources
  - set a resource as a violent content (attribute), accessible only by direct friends (policy)
  - Parents can set attributes and policies of children's resources
  - Discretionary Access Control belongs here
- Control Activity on Users
  - By changing user attributes and policies
  - To control activity performed by/against a particular user (self or other related users) without knowing a particular resource
- Control Activity on Sessions
  - By controlling session attributes and policies that are inherited from a user

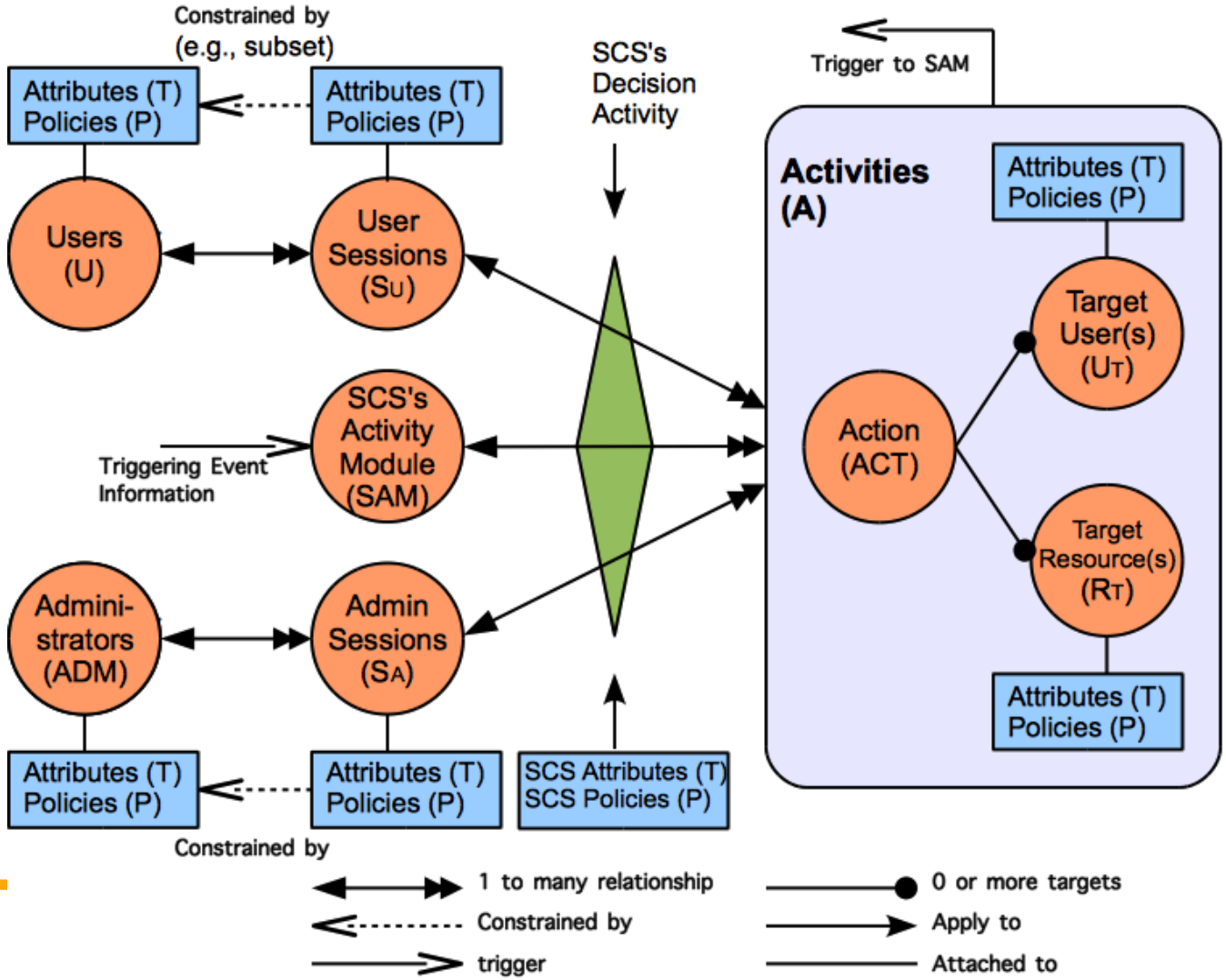
SCS's (Automated) Activities

- Usage Activities
  - Typically accessing users' shared information to use it as an input for service activities
- Service Activities
  - To promote users' social interactions and information sharing
  - Friends recommendation, friend activity notification, location-based coupons, most-viewed videos
- Control Activities
  - Through managing policies and attributes of users, resources and sessions
  - User rating-based seller trustworthiness or product popularity
- Decision Activities
  - SCS evaluates requests for user's usage and control activities as well as SCS's service and control activities

Activity(-centric Access) Control Framework

- To capture various users and SCS activities and their influences on control decisions
- To support controls on various access/usage, service and control activities in SCS
- To support personalized user privacy control
- To support automated management of SCS services and controls

ACON Framework



I·C·S
The Institute for Cyber Security

UTSA

ACON Framework Components

- Users
  - represent a human being who performs activities in an SCS
  - Carry attributes and policies
- Sessions
  - Represent an active user who has logged into the SCS
  - A user can have multiple sessions, but not vise versa
  - Carry attributes and policies that could be different from user attributes and policies

ACON Framework Components (cont)

- Activities
  - User, SCS, SCS administrator's activities
  - Comprise action, target users, target resources
    - Action
      - An abstract function available in SCS
      - E.g., read, rate, poke, friend-invite, activity notification
    - Target users(' sessions)
      - Recipients of an action
    - Target Resources
      - Include users'/SCS's shared contents, user/resource/session policies and attributes

ACON Framework Components (cont)

- SCS's Decision Activity
  - based on the consolidated individual user/resource policies and attributes together w/ SCS policies and attributes
- SCS's Activity Module (SAM)
  - A conceptual abstraction of functions that performs SCS's automated usage, service and control activities
- SCS Administrators
  - Human being w/ a management role

ACON Framework Characteristics

- Policy Individualization
  - A user's individual policy includes privacy preferences and activity limits
  - Collectively used by SCS for control decision on activities
  - Can be configured by related users
- User and Resource as a Target
  - User as a target requires policies for actions against target users.
  - Poking, friends recommendation
- Separation of user policies for incoming and outgoing actions
  - Incoming action policy: Homer doesn't want to receive notification of friends' activities
  - Outgoing action policy: Homer says Bart cannot be a friend of Homer's coworkers
- User-session distinction
- User relationship independent access control
- SCS's automated usage, service and control activities

Examples

- Request (Alice, read, video1) is allowed if Alice is a friend of the owner of video1 and if Alice is over 18
  - Request: AU: Alice, UsageAct: read, TR:video1
  - Attributes: AU.ATT: friend={bob}; DOB=1980, TR.ATT: owner=Bob, contentType=violent
  - Policies: TR.P=can be read by owner's friends, System.P=only >18 can play/read violent contents
- Request (Alice, poke, Bob) is allowed if Alice is a friend of Bob

$ACON_{user}$ Model – User Activity Control

- *U, S, ACT, R, T, P, SCS* and *D* (users, sessions, actions, resources, attributes, policies, social computing system and decision predicate, respectively);
- $U_T \subseteq U$ and $R_T \subseteq R$ (target users and target resources, respectively);
- dot notation: we understand *e.T* and *e.P* to respectively denote the set of attributes and set of policies associated with entity *e*;
- *A*, the set of activities is defined as $A \subseteq ACT \times (2^{R_T} \times 2^{U_T} - \varnothing)$;
- Let $A = \{a_1, a_2, ..., a_n\}$, we denote the components of each individual element as $a_i = (a_i.ACT, a_i.R_T, a_i.U_T)$;

$ACON_{user}$ Model – User Activity Control

- $AP\_R_T : A \rightarrow 2^{R_T \times P}$, $AP\_U_T : A \rightarrow 2^{U_T \times P}$, $AT\_R_T : A \rightarrow 2^{R_T \times T}$, $AT\_U_T : A \rightarrow 2^{U_T \times T}$, mappings of activity to a set of target resources and policies, a set of target users and policies, a set of target resources and attributes, and a set of target users and attributes respectively defined as:
  - $AP\_R_T(\{a_1,..,a_n\}) = AP\_R_T(\{a_1\}) \cup ... \cup AP\_R_T(\{a_n\})$, $AP\_R_T(\{a_i\}) = \{(r_t,p) | r_t \in a_i.R_T, p \in r_t.P\}$
  - $AP\_U_T(\{a_1,..,a_n\}) = AP\_U_T(\{a_1\}) \cup ... \cup AP\_U_T(\{a_n\})$, $AP\_U_T(\{a_i\}) = \{(u_t,p) | u_t \in a_i.U_T, p \in u_t.P\}$
  - $AT\_R_T(\{a_1,..,a_n\}) = AT\_R_T(\{a_1\}) \cup ... \cup AT\_R_T(\{a_n\})$, $AT\_R_T(\{a_i\}) = \{(r_t,t) | r_t \in a_i.R_T, t \in r_t.T\}$
  - $AT\_U_T(\{a_1,..,a_n\}) = AT\_U_T(\{a_1\}) \cup ... \cup AT\_U_T(\{a_n\})$, $AT\_U_T(\{a_i\}) = \{(u_t,t) | u_t \in a_i.U_T, t \in u_t.T\}$;

$ACON_{user}$ Model – User Activity Control

- $AP(a)=AP\_R_T(a) \cup AP\_U_T(a)$,
- $AT(a)=AT\_R_T(a) \cup AT\_U_T(a)$;

- $allowed(s,a) \Rightarrow D(s.P,s.T,a,AP(a), AT(a),scs.P,scs.T)$, where $s \in S$ and $a \in A$.

$ACON_{user}$ Model – Session Management

- $user\_sessions : U \rightarrow 2^S$, $session\_users : S \rightarrow U$;
- $user\_added\_sessionT : S \rightarrow 2^T$, $user\_removed\_sessionT : S \rightarrow 2^T$;
- $scs\_added\_sessionT : S \rightarrow 2^T$, $scs\_removed\_sessionT : S \rightarrow 2^T$, $scs\_required\_sessionT : S \rightarrow 2^T$;
- $user\_added\_sessionP : S \rightarrow 2^P$, $user\_removed\_sessionP : S \rightarrow 2^P$;
- $scs\_added\_sessionP : S \rightarrow 2^P$, $scs\_removed\_sessionP : S \rightarrow 2^P$, $scs\_required\_sessionT : S \rightarrow 2^T$;

- $user\_removed\_sessionT(s) \subseteq \{t \in T \mid t \in session\ users(s).T \wedge t \notin scs\_required\_sessionT(s)\}$;
- $user\_removed\_sessionP(s) \subseteq \{p \in P \mid p \in session\ users(s).P \wedge p \notin scs\ required\_sessionP(s)\}$;

$ACON_{user}$ Model – Session Management

- $assignS\_T : S \rightarrow 2^T$, $assignS\_P : S \rightarrow 2^P$, assignment of attributes and policies to sessions respectively;
- $assignS\_T(s) \subseteq \{t \in T | (t \in session\_users(s).T) \vee (t \in user\_added\_sessionT(s)) \vee (t \in scs\_added\_sessionT(s)) \wedge \neg((t \in user\_removed\_sessionT(s)) \vee (t \in scs\_removed\_sessionT(s)))\};$
- $assignS\_P(s) \subseteq \{p \in P | (p \in session\_users(s).P) \vee (p \in user\_added\_sessionP(s)) \vee (p \in scs\_added\_sessionP(s)) \wedge \neg((p \in user\_removed\_sessionP(s)) \vee (p \in scs\_removed\_sessionP(s)))\}.$

Examples

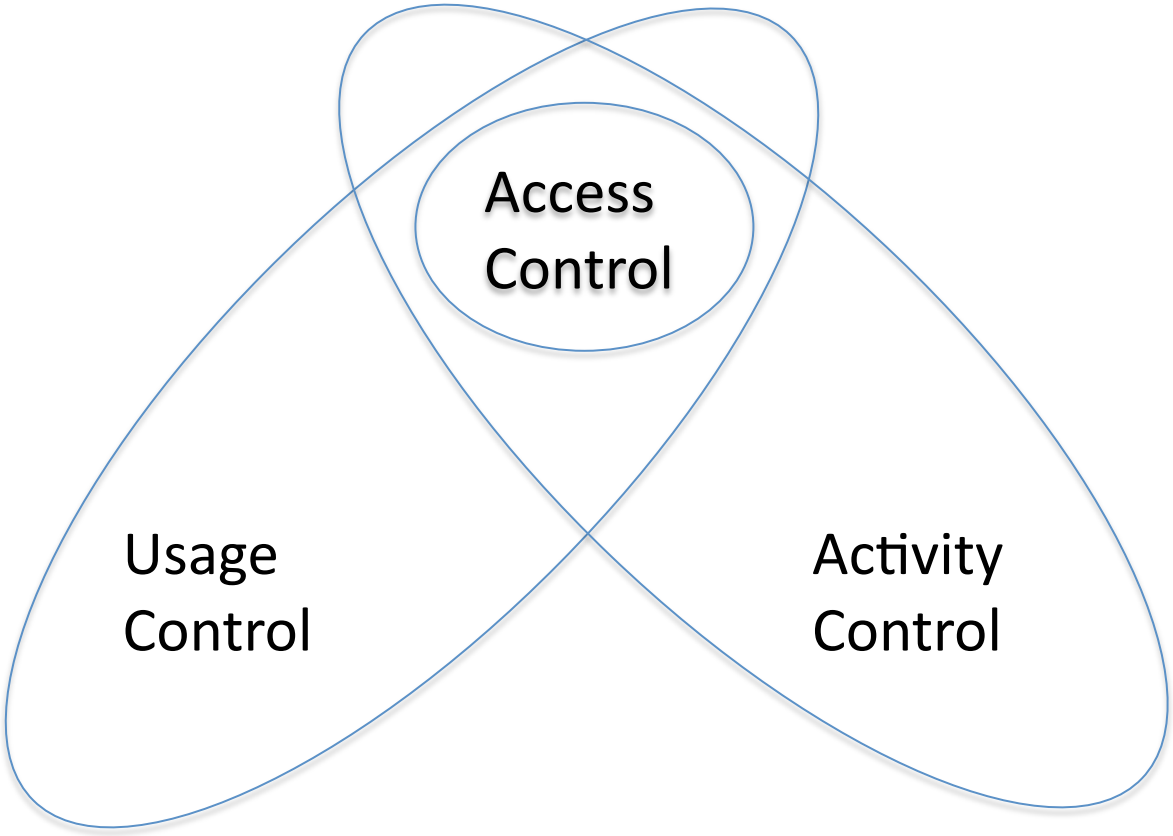
- A buyer can rate a seller only if the buyer bought a product from the seller (SCS.P).
  - *N: a list of users, sellerList : S → 2^N*
  - *allowed(s, rate, u_t) ⇒ u_t ∈ sellerList(s)*
- A user can recommend a friendship between two friends if they are not a friend to each other(SCS.P).
  - *N: a list of users, friends : S → 2^N*
  - *allowed(s, f-recommend, u_t1, u_t2)⇒*
    ({u_t1, u_t2}∈ friends(s)) ∧ (u_t2 ∉ friends(u_t1)) ∧
    (u_t1 ∉ friends(u_t2))

Summary

- Developed activity-centric access control framework for security and privacy in social computing systems.

- Developed initial models for user activity controls and session management.

Controlling Access, Usage and Activity



- Comments and Questions?

- Contact info
  - jae.park@utsa.edu
  - www.drjae.com
  - http://ics.utsa.edu