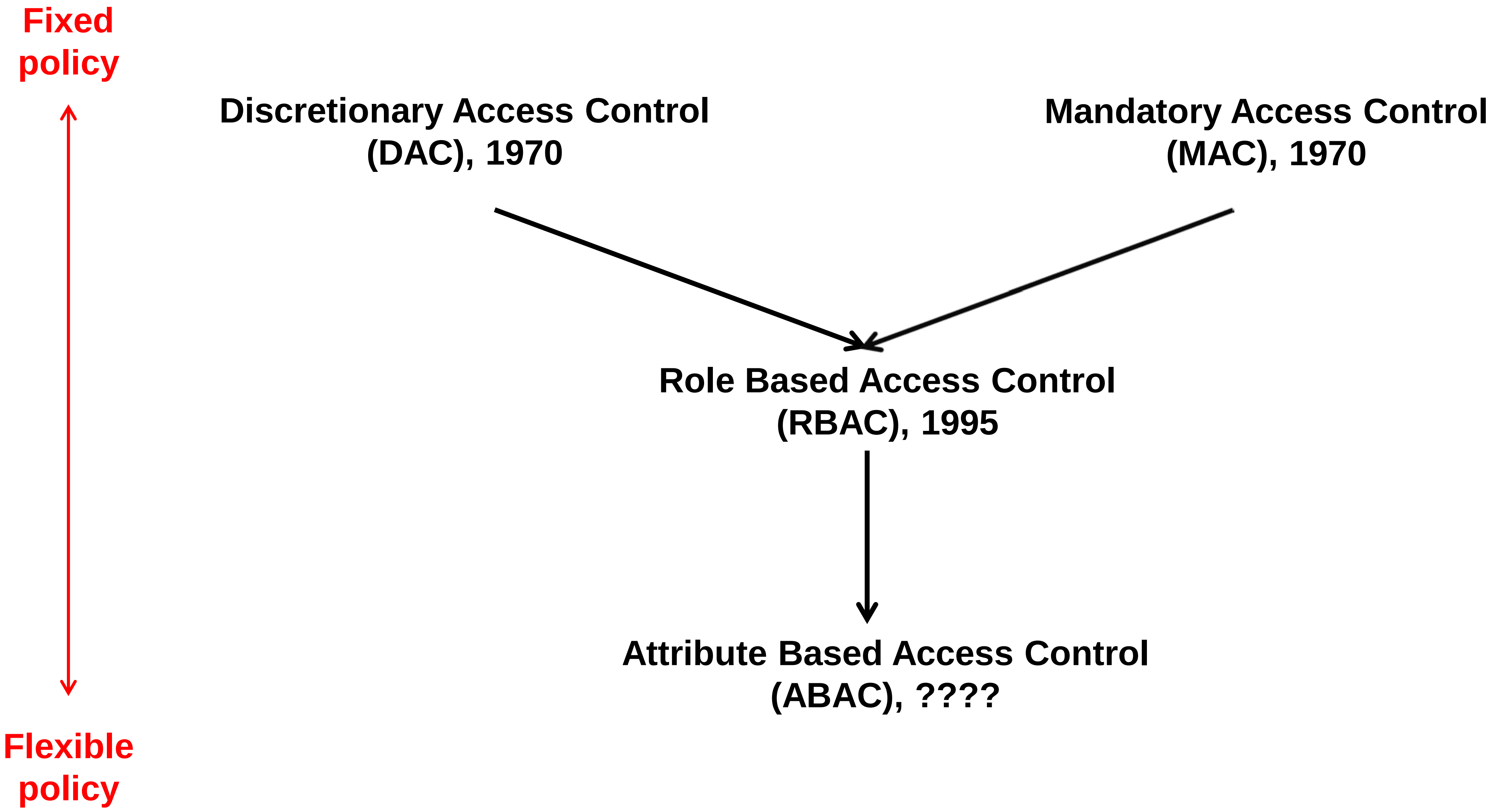
Attribute-Based Access Control (ABAC)

Prof. Ravi Sandhu
Executive Director and Endowed Chair

Lecture 5

ravi.utsa@gmail.com
www.profsandhu.com

Access Control



Fixed policy

Discretionary Access Control (DAC), 1970

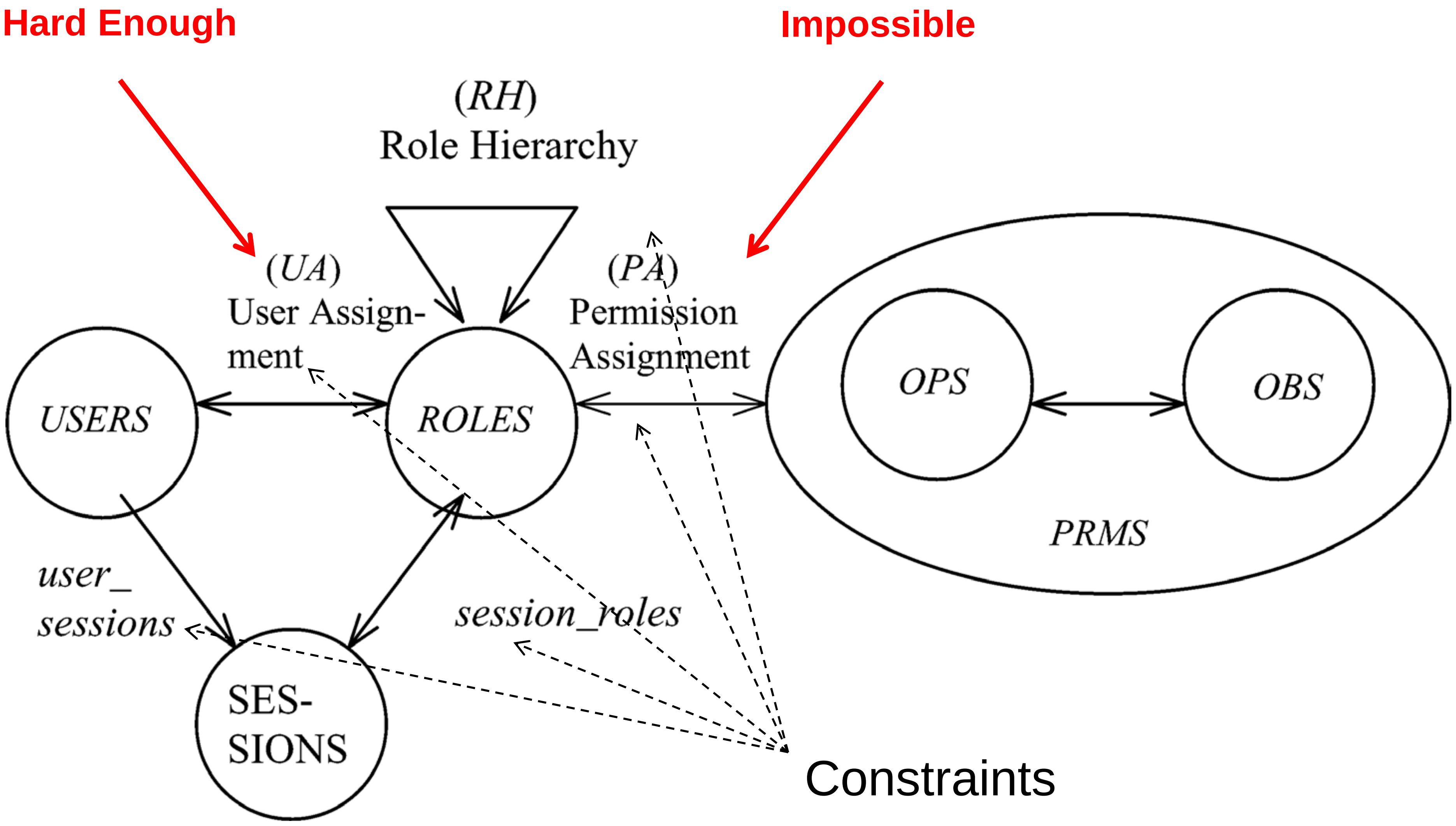
Mandatory Access Control (MAC), 1970

Role Based Access Control (RBAC), 1995

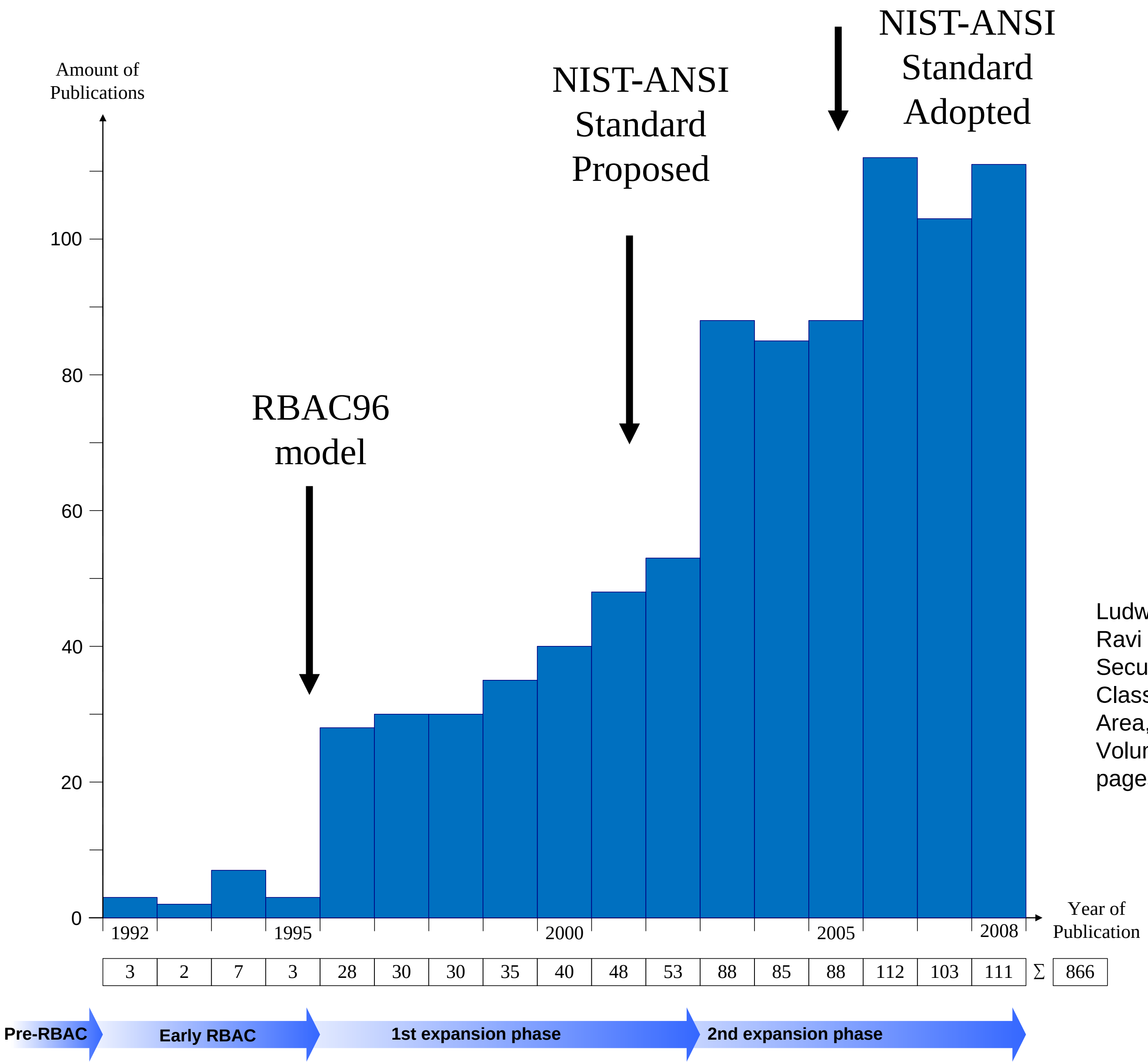
Attribute Based Access Control (ABAC), ????

Flexible policy

RBAC Shortcomings



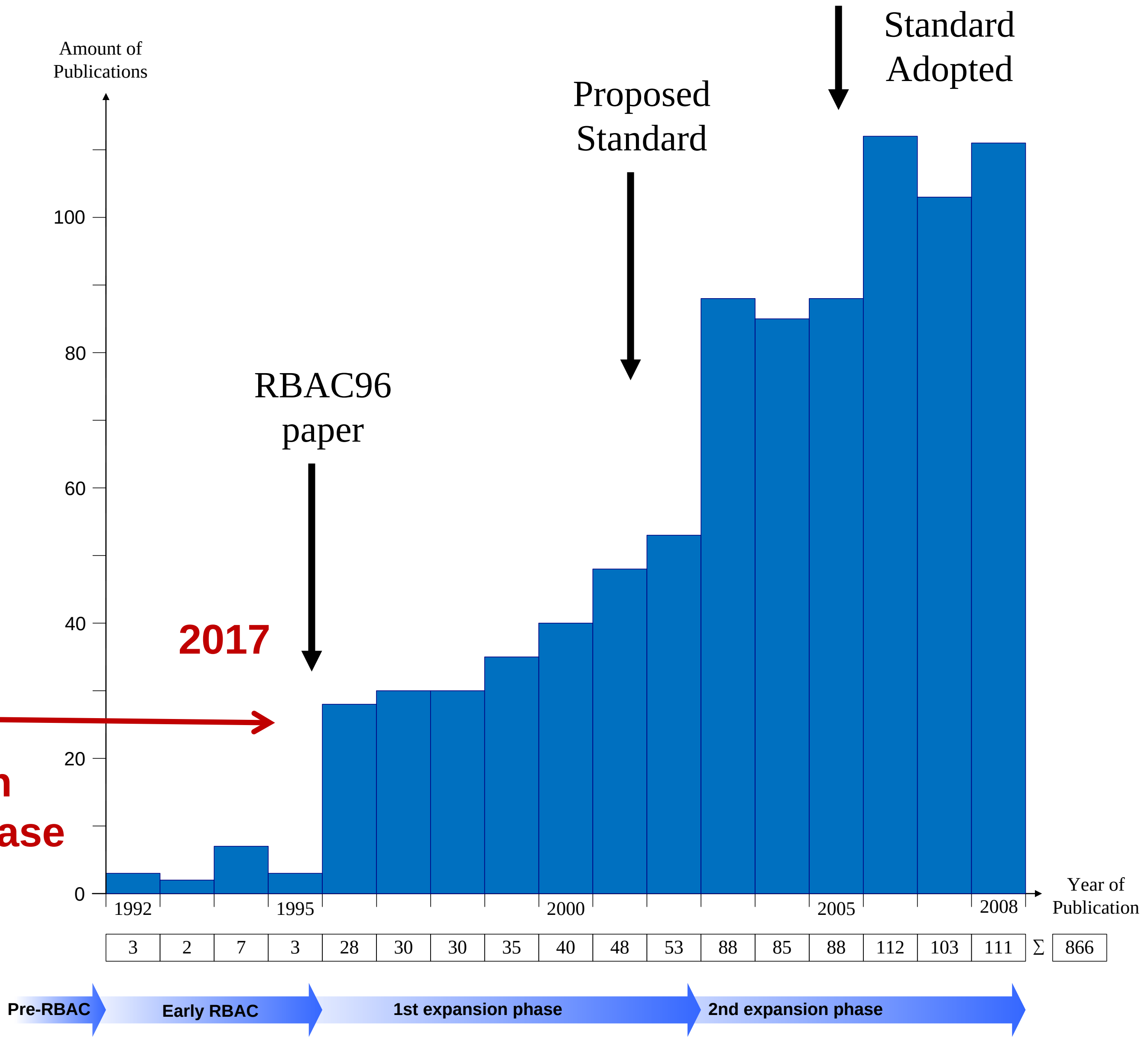
The RBAC Story



3	2	7	3	28	30	30	35	40	48	53	88	85	88	112	103	111	Σ	866

Ludwig Fuchs, Gunther Pernul and Ravi Sandhu, Roles in Information Security-A Survey and Classification of the Research Area, Computers & Security, Volume 30, Number 8, Nov. 2011, pages 748-76

ABAC Status



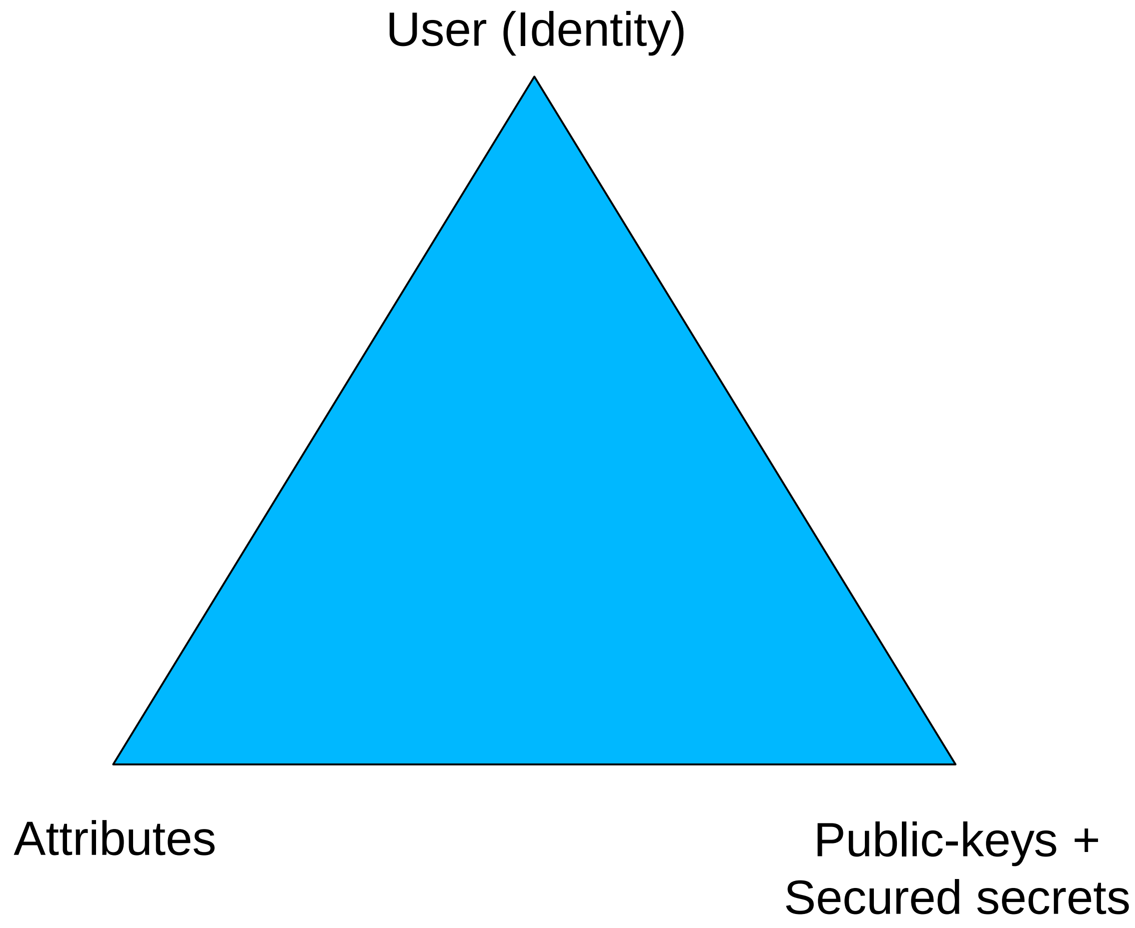
3	2	7	3	28	30	30	35	40	48	53	88	85	88	112	103	111	Σ	866

1990?

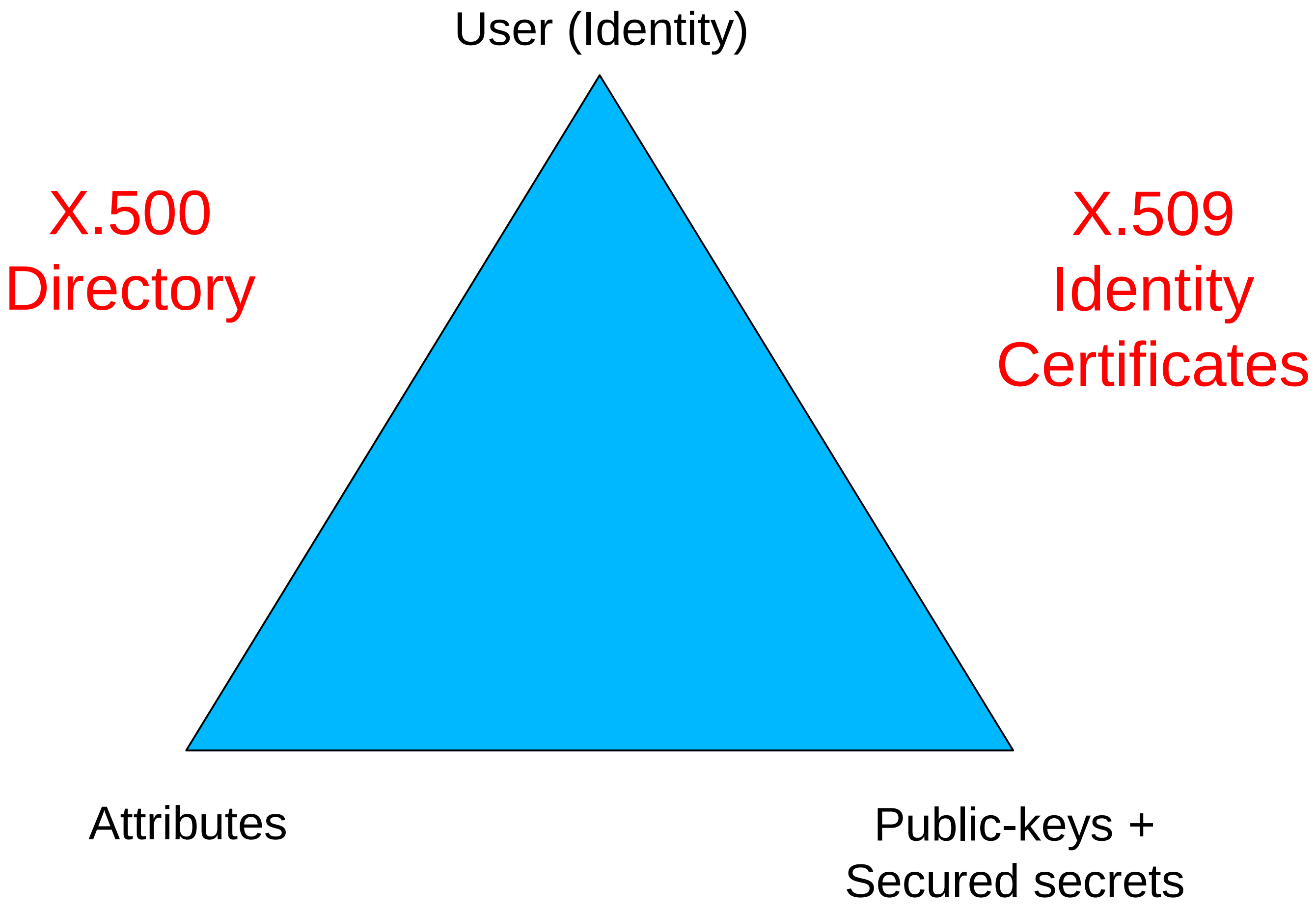
2017

ABAC still in pre/early phase

ABAC is not New



ABAC is not New



User (Identity)

X.500
Directory

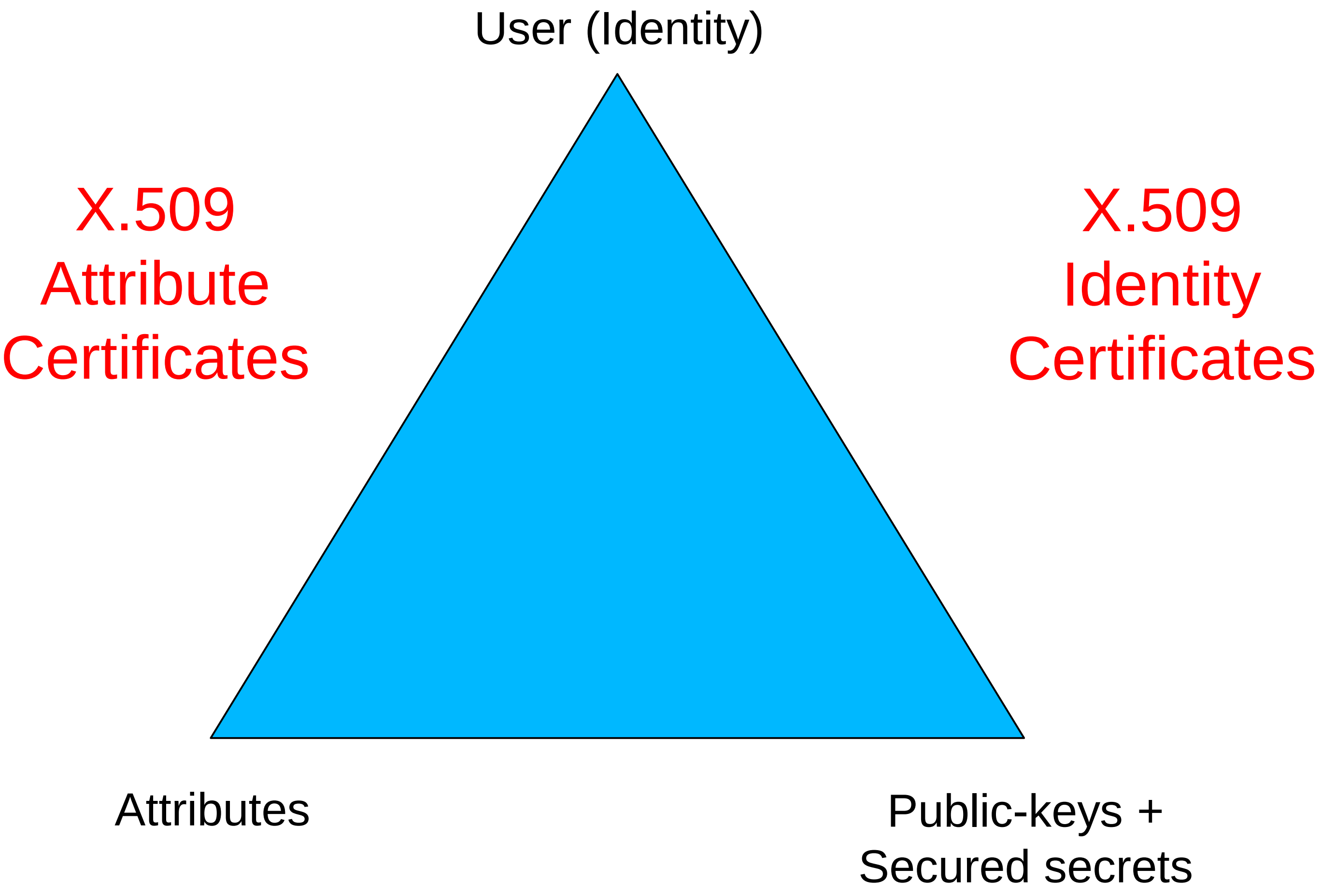
X.509
Identity
Certificates

Attributes

Public-keys +
Secured secrets

Pre Internet, early 1990s

ABAC is not New



User (Identity)

X.509 Attribute Certificates

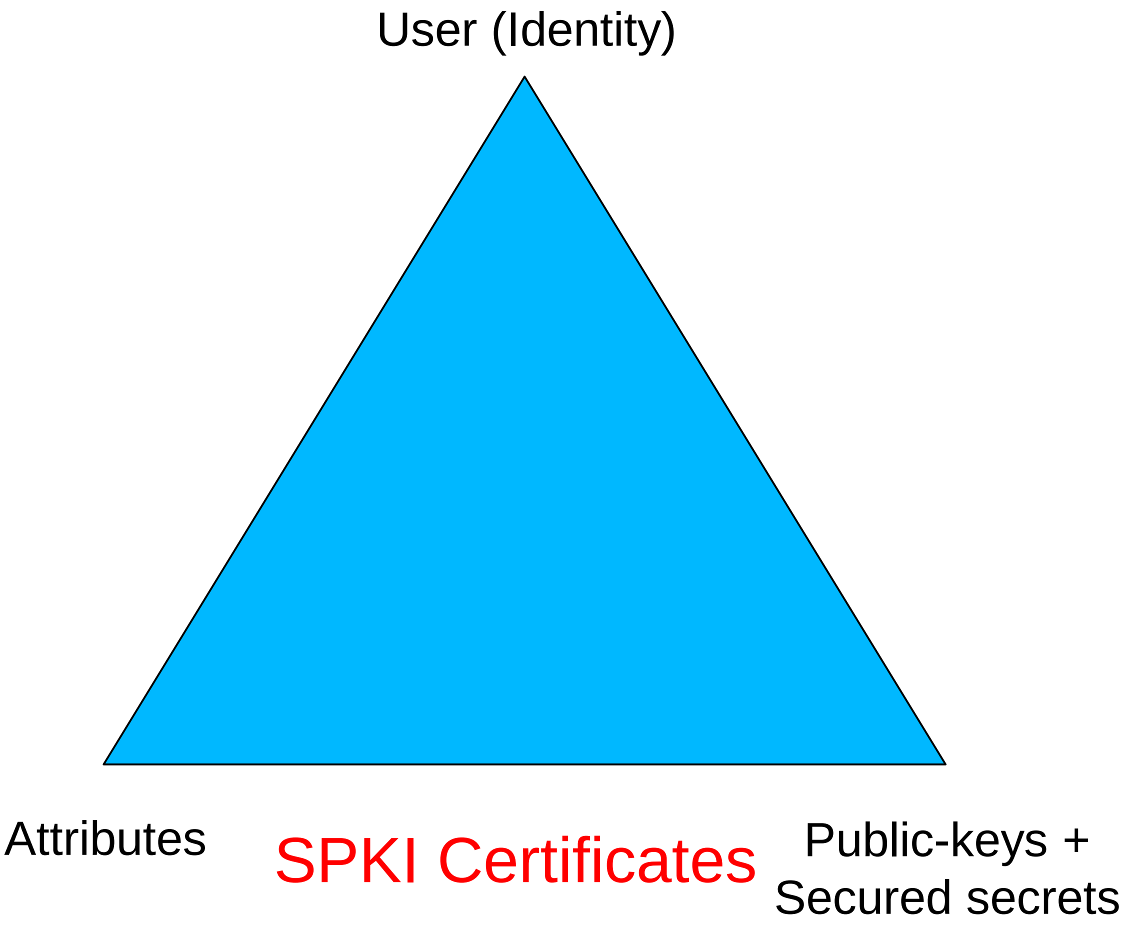
X.509 Identity Certificates

Attributes

Public-keys + Secured secrets

Post Internet, late 1990s

ABAC is not New


User (Identity)

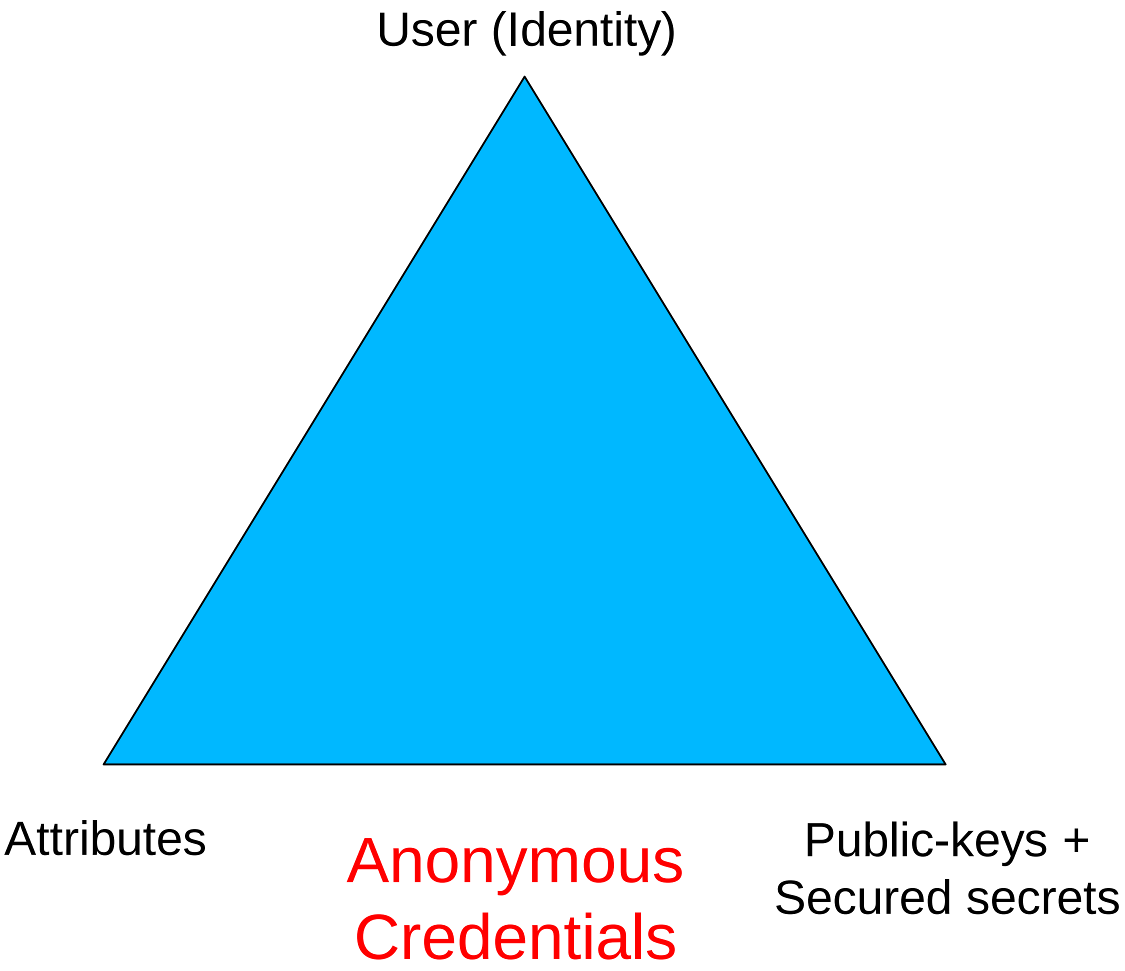
Attributes

SPKI Certificates

Public-keys +
Secured secrets

Post Internet, late 1990s

ABAC is not New



User (Identity)

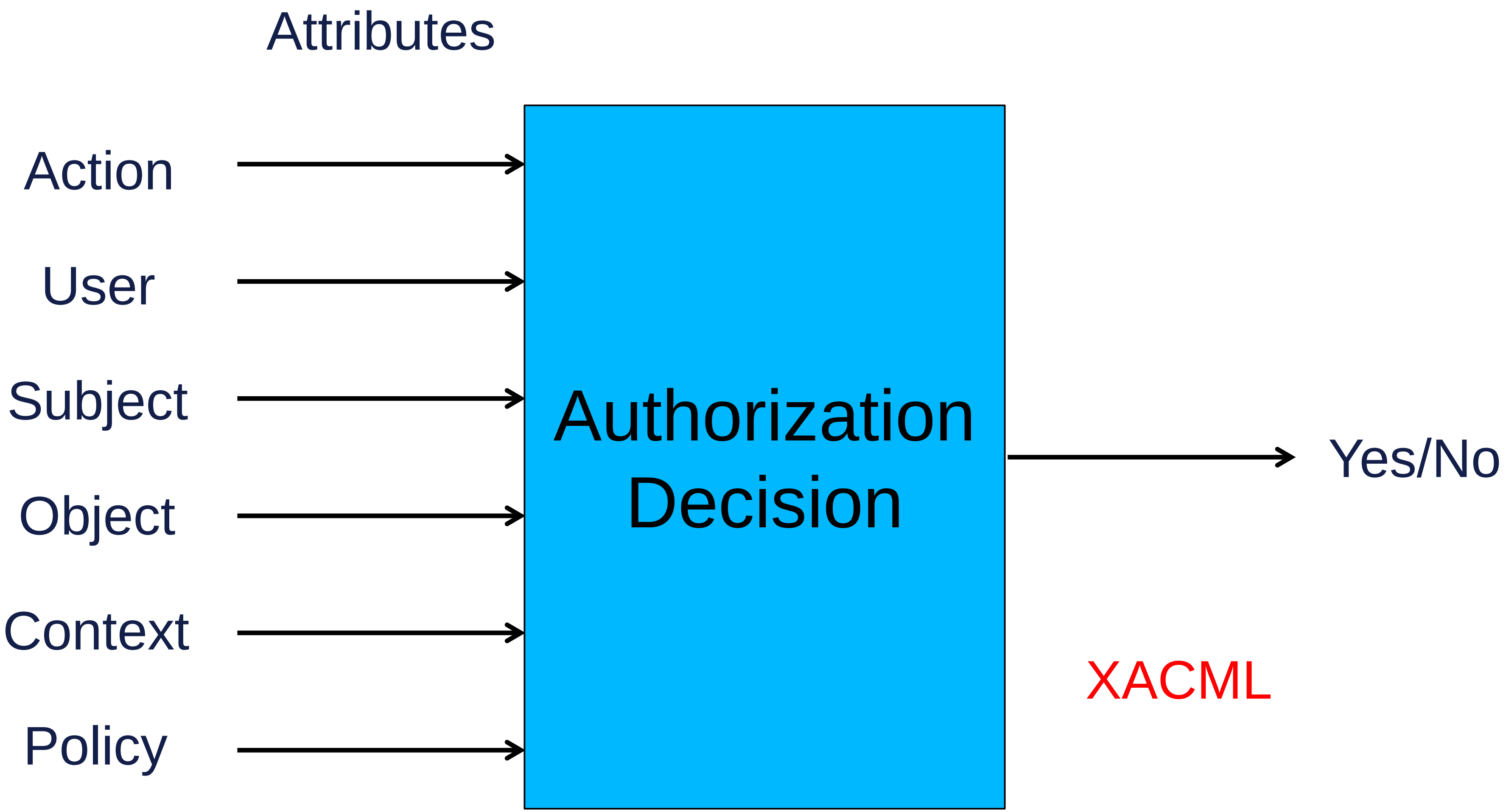
Attributes

Anonymous Credentials

Public-keys +
Secured secrets

Mature Internet, 2000s

ABAC is not New



Attributes

Action

User

Subject

Object

Context

Policy

Authorization Decision

Yes/No

XACML

Mature Internet, 2000s

ABACα and ABACβ Models

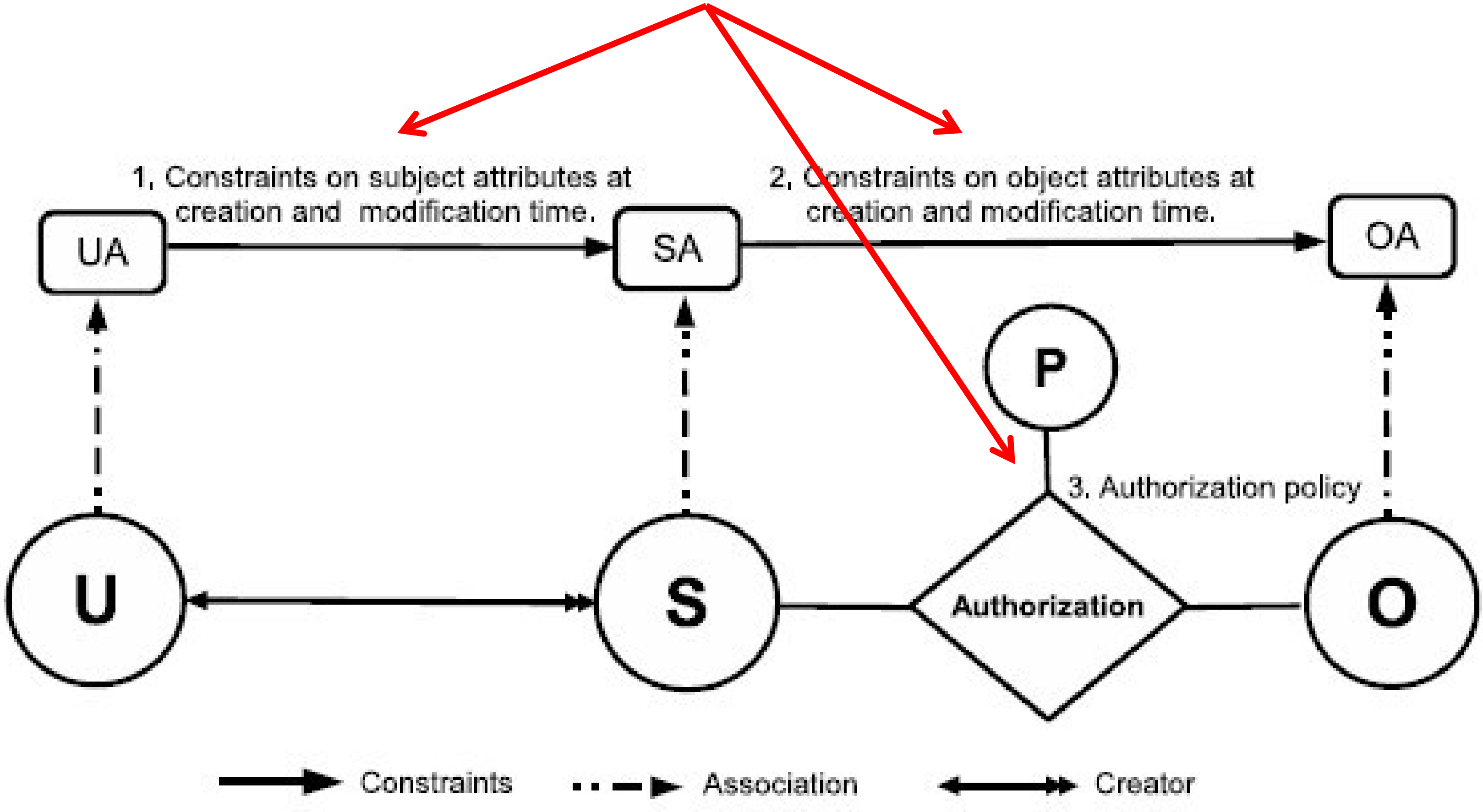
ABACα Features

- ABAC-alpha covers
  - DAC: strict DAC
  - MAC: LBAC with tranquility
  - RBAC: RBAC$_0$ and RBAC$_1$

	Subject attribute Value constrained by creating user ?	Object attribute value constrained by creating subject ?	Attribute range ordered?	Attribute function returns set value?	Object attribute modification?	Subject attribute modification by creating user?
DAC	YES	YES	NO	YES	YES	NO
MAC	YES	YES	YES	NO	NO	NO
RBAC0	YES	NA	NO	YES	NA	YES
RBAC1	YES	NA	YES	YES	NA	YES
ABAC-alpha	YES	YES	YES	YES	YES	YES

ABACα Model Structure

Policy Configuration Points


1. Constraints on subject attributes at creation and modification time.
2. Constraints on object attributes at creation and modification time.
UA
SA
OA
P
3. Authorization policy
U
S
Authorization
O
Constraints
Association
Creator

Just sufficient mechanism to do simple forms of DAC, MAC, RBAC

ABACα Authorization Policy

- ❖DAC

$$Authorization_{read}(s,o) \equiv SubCreator(s) \in reader(o)$$

$$Authorization_{write}(s,o) \equiv SubCreator(s) \in writer(o)$$

- ❖MAC

$$Authorization_{read}(s,o) \equiv sensitivity(o) \leq sclearance(s)$$

$$\text{Liberal star}: Aauthorization_{write}(s,o) \equiv sclearance(s) \leq sensitivity(o)$$

$$\text{Strict star}: Aauthorization_{write}(s,o) \equiv sensitivity(o) = sclearance(s)$$

- ❖RBAC0

$$Authorization_{read}(s,o) \equiv \exists r \in srole(s).r \in rrole(o)$$

- ❖RBAC1

$$Authorization_{read}(s,o) \equiv \exists r1 \in srole(s).\exists r2 \in rrole(o).r2 \leq r1$$

ABACα Subject Attribute Constraints

- ❖MAC creation $ConstrSub(u, s, \{(sclearance, value)\}) \equiv value \leq uclearance(u)$

  modification FALSE

- ❖RBAC0 $ConstrSub(u, s, \{srole, value\}) \equiv value \subseteq urole(u)$

- ❖RBAC1 $ConstrSub(u, s, \{srole, value\}) \equiv \forall r1 \in value.\exists r2 \in urole(u).r1 \leq r2$

ABACα Object Attribute Constraints

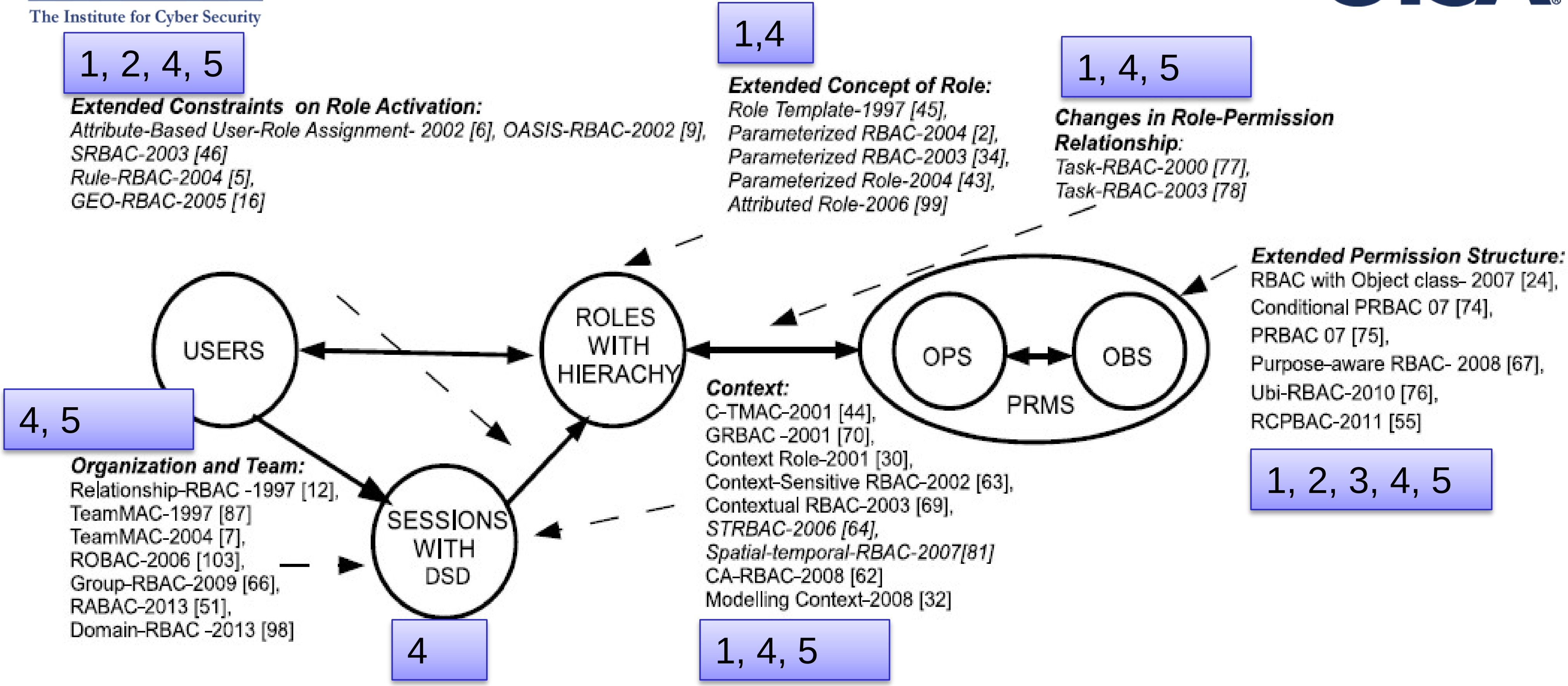
- DAC Creation $ConstrObj(s,o,\{(reader,val1),(writer,val2),(createdby,val3)\}) \equiv$ $val3 = SubCreator(s)$

  Modification $ConstrObj(s,o,\{(reader,val1),(writer,val2),(createdby,val3)\}) \equiv$ $createdby(o) = SubCreator(s)$

- MAC Creation $ConstrObj(s,o,\{sensitivity,value\}) \equiv sclearance(s) \leq value$

  Modification FALSE

RBAC Extensions



Extended Constraints on Role Activation:
Attribute-Based User-Role Assignment- 2002 [6], OASIS-RBAC-2002 [9],
SRBAC-2003 [46]
Rule-RBAC-2004 [5],
GEO-RBAC-2005 [16]

Extended Concept of Role:
Role Template-1997 [45],
Parameterized RBAC-2004 [2],
Parameterized RBAC-2003 [34],
Parameterized Role-2004 [43],
Attributed Role-2006 [99]

Changes in Role-Permission Relationship:
Task-RBAC-2000 [77],
Task-RBAC-2003 [78]

Extended Permission Structure:
RBAC with Object class- 2007 [24],
Conditional PRBAC 07 [74],
PRBAC 07 [75],
Purpose-aware RBAC- 2008 [67],
Ubi-RBAC-2010 [76],
RCPBAC-2011 [55]

Context:
C-TMAC-2001 [44],
GRBAC -2001 [70],
Context Role-2001 [30],
Context-Sensitive RBAC-2002 [63],
Contextual RBAC-2003 [69],
STRBAC-2006 [64],
Spatial-temporal-RBAC-2007[81]
CA-RBAC-2008 [62]
Modelling Context-2008 [32]

Organization and Team:
Relationship-RBAC -1997 [12],
TeamMAC-1997 [87]
TeamMAC-2004 [7],
ROBAC-2006 [103],
Group-RBAC-2009 [66],
RABAC-2013 [51],
Domain-RBAC -2013 [98]

1. Context Attributes

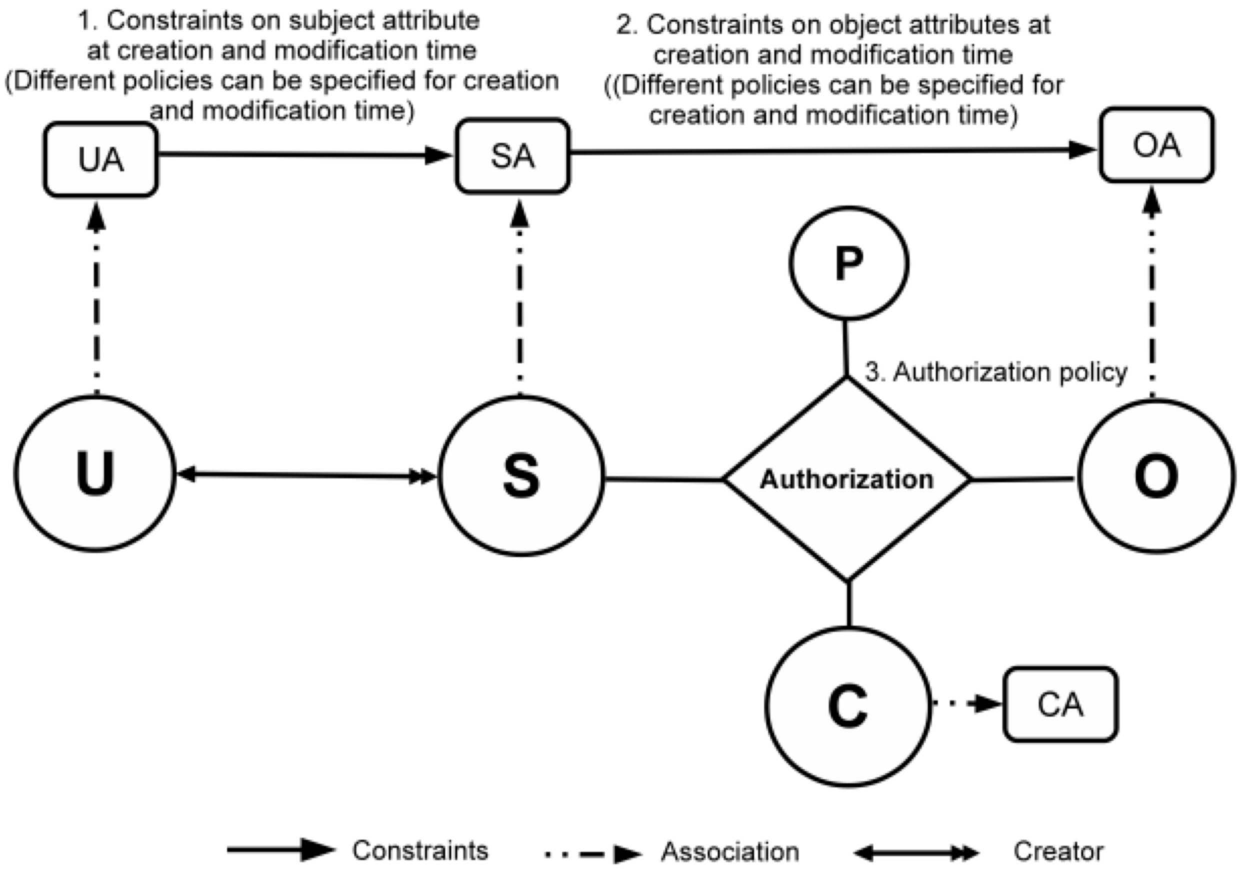
2. Subject attribute constraints policy are different at creation and modification time.

3. Subject attributes constrained by attributes of subjects created by the same user.

4. Policy Language

5. Meta-Attributes

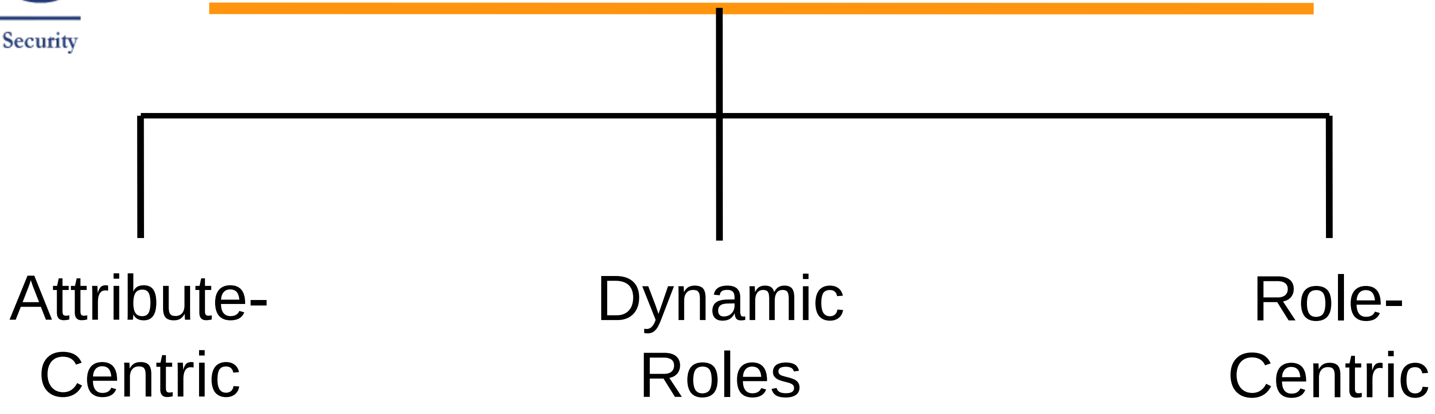
ABAC$_\beta$ Model


1. Constraints on subject attribute at creation and modification time (Different policies can be specified for creation and modification time)
2. Constraints on object attributes at creation and modification time ((Different policies can be specified for creation and modification time)
UA
SA
OA
P
3. Authorization policy
U
S
Authorization
O
C
CA
Constraints
Association
Creator

Can be configured to do many but not all RBAC extensions

Roles and Attributes

Roles and Attributes



Attribute-Centric

Dynamic Roles

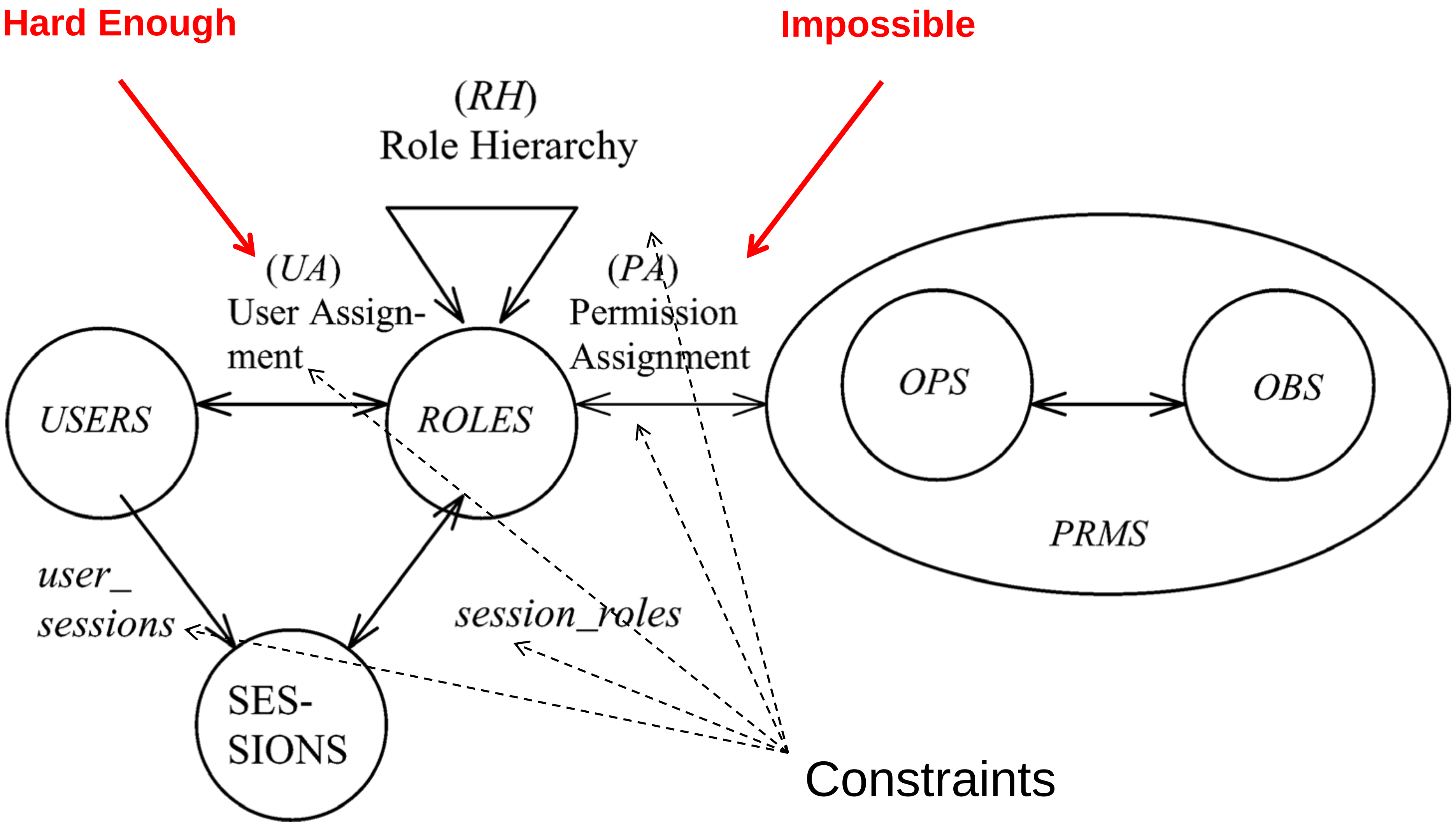
Role-Centric

Role is just another attribute. Nothing special about it.

Compute user roles from user attributes

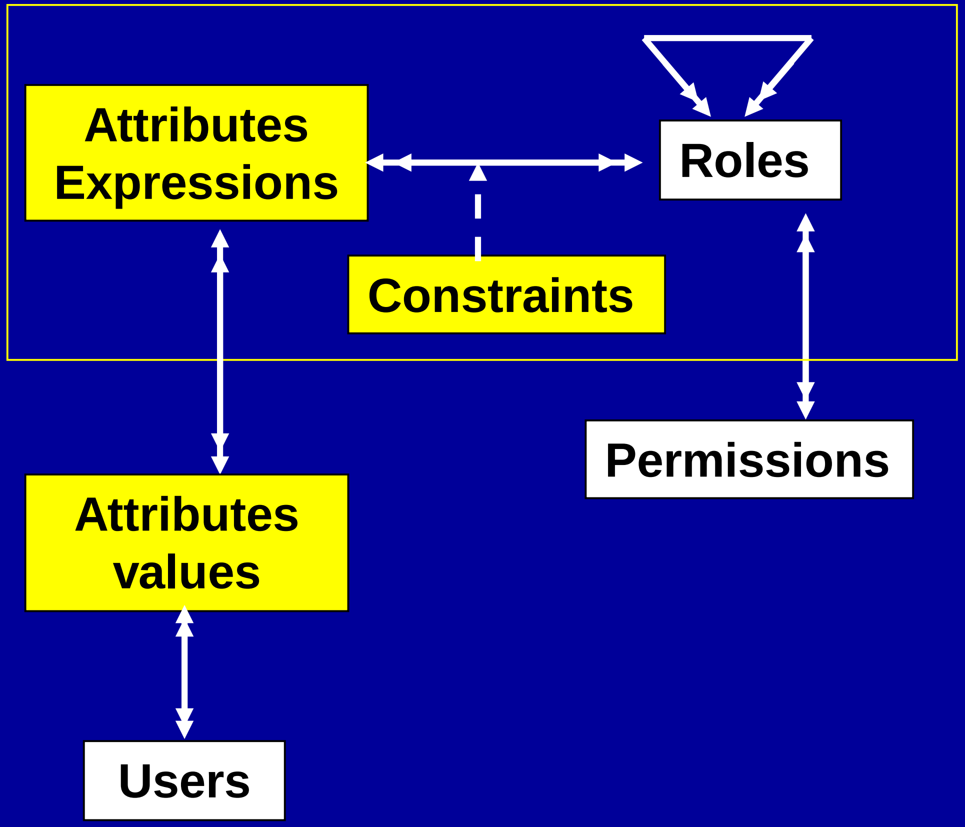
Attributes constrain permissions of roles for each user

User-Role Assignment Problem

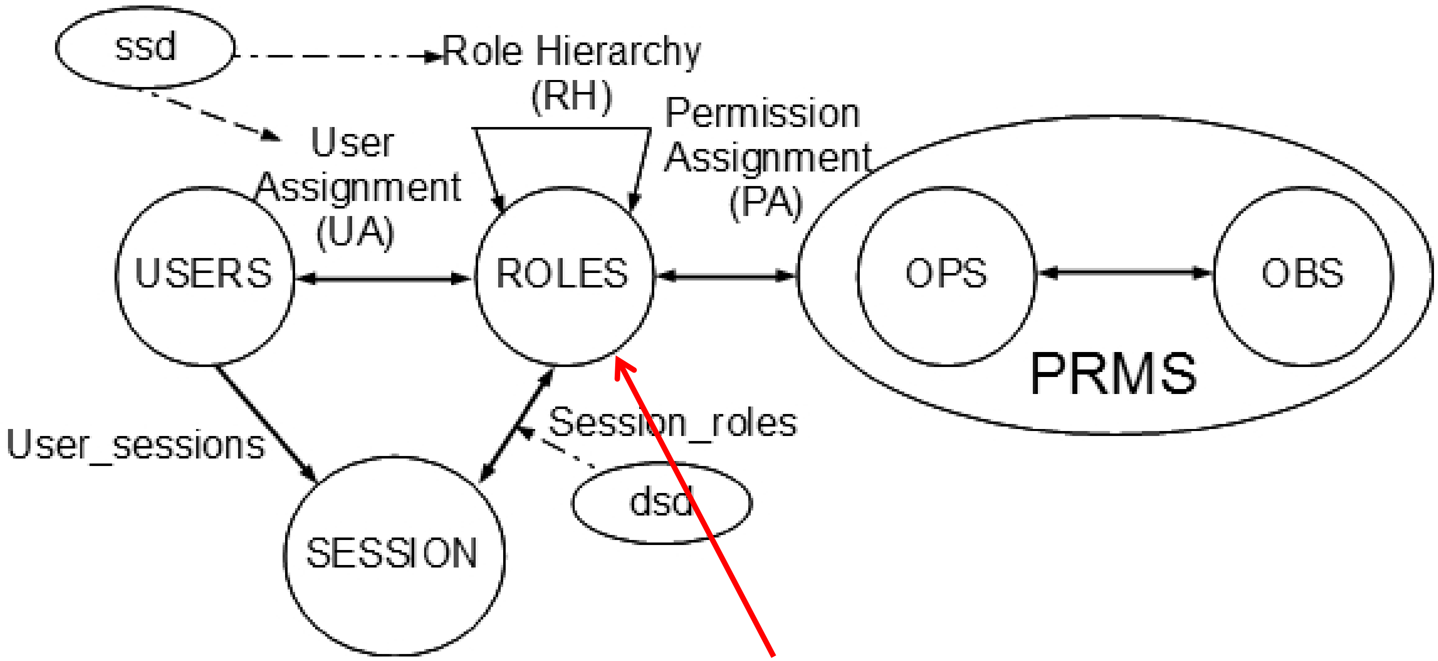


Dynamic Roles

- **Rule-Based RBAC (RB-RBAC)**



Role Explosion Problem

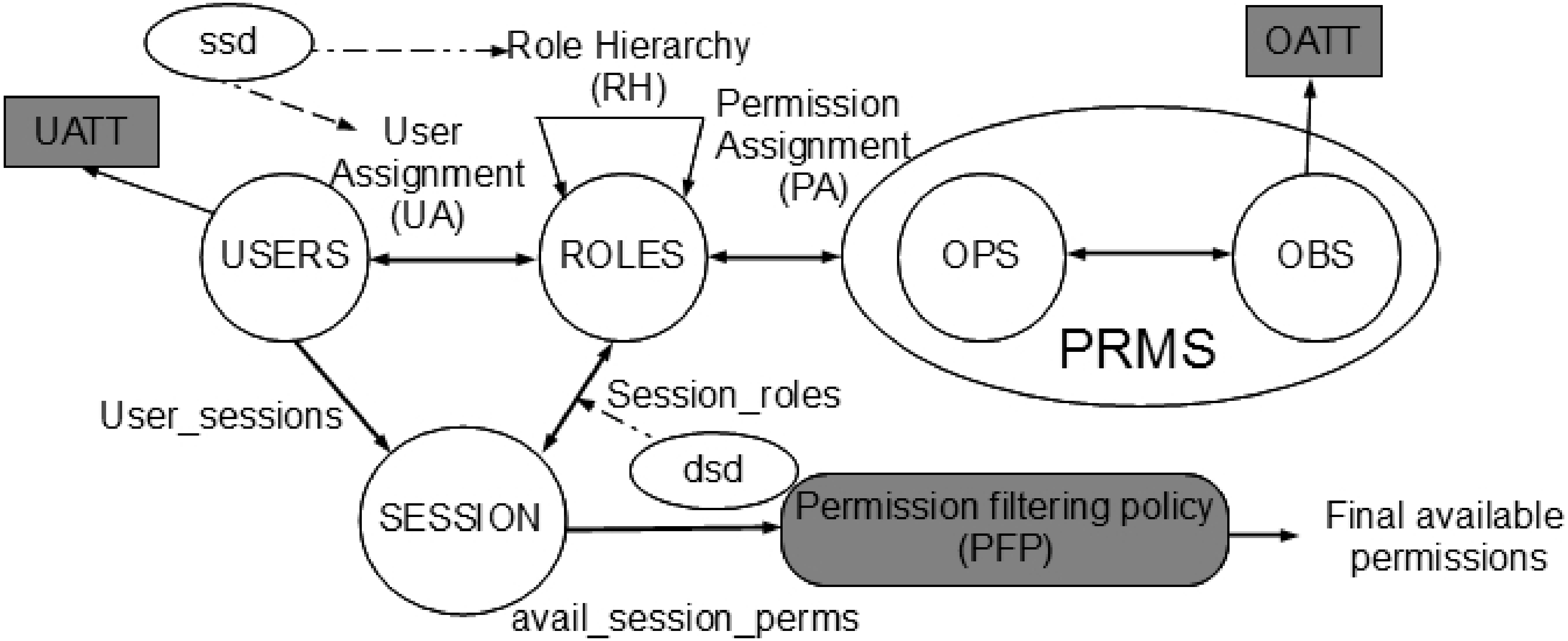


Role Explosion

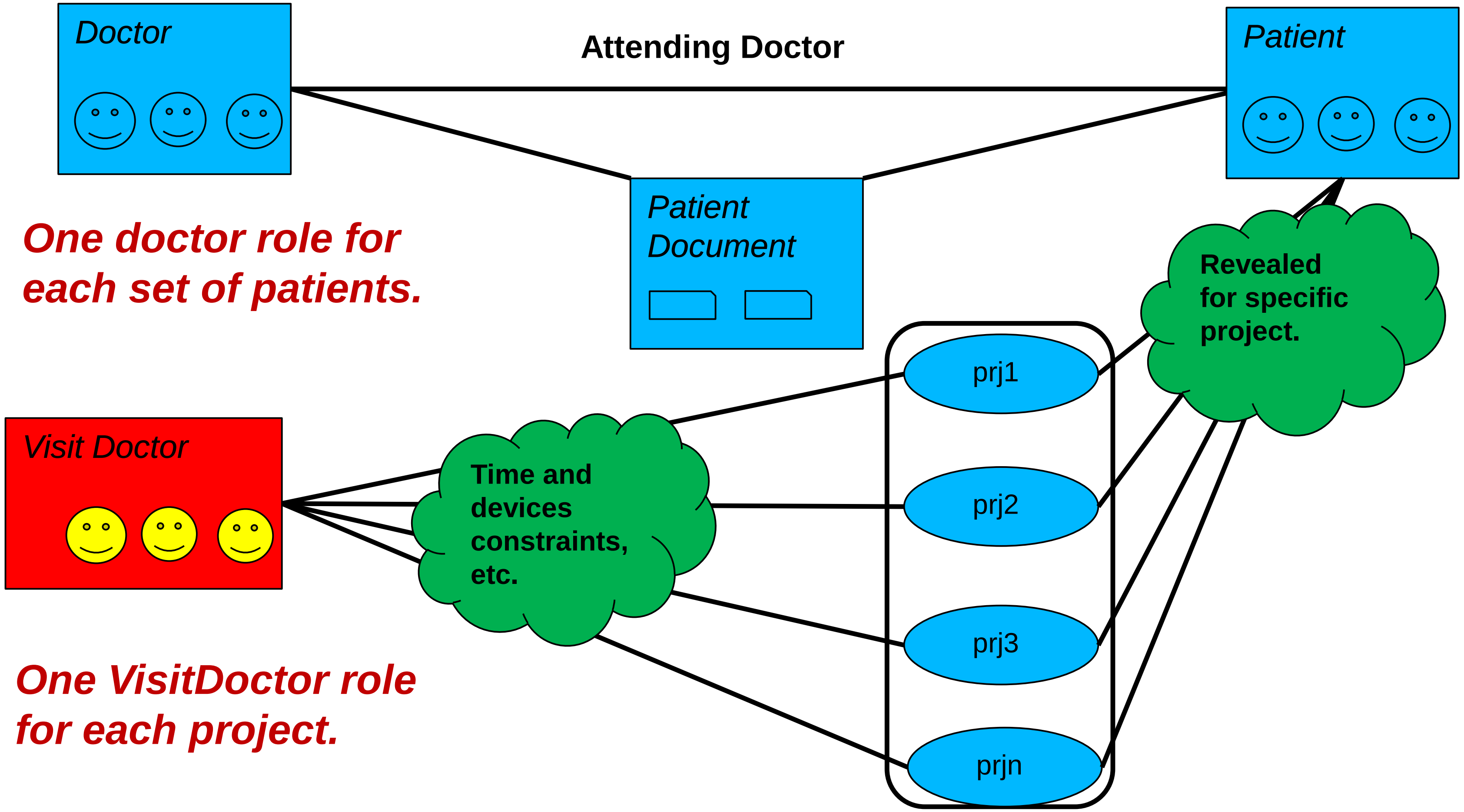
Number of roles is supposed to be much smaller than number of users.

Role Explosion : Different roles have to be defined for slightly different sets of permissions.

RABAC (Role-Centric ABAC)



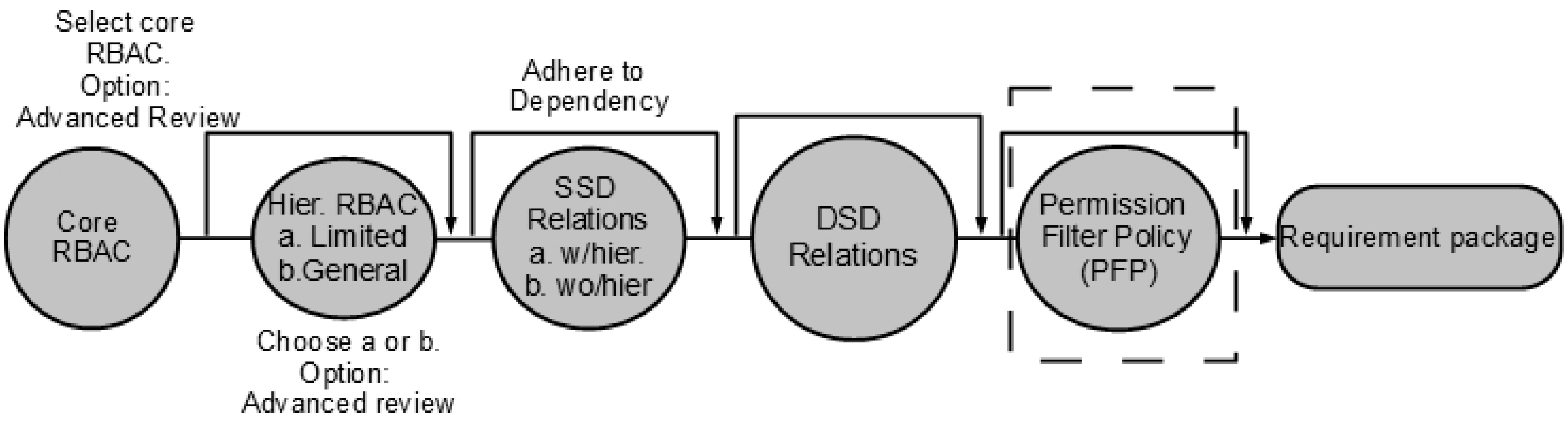
Example



One doctor role for each set of patients.

One VisitDoctor role for each project.

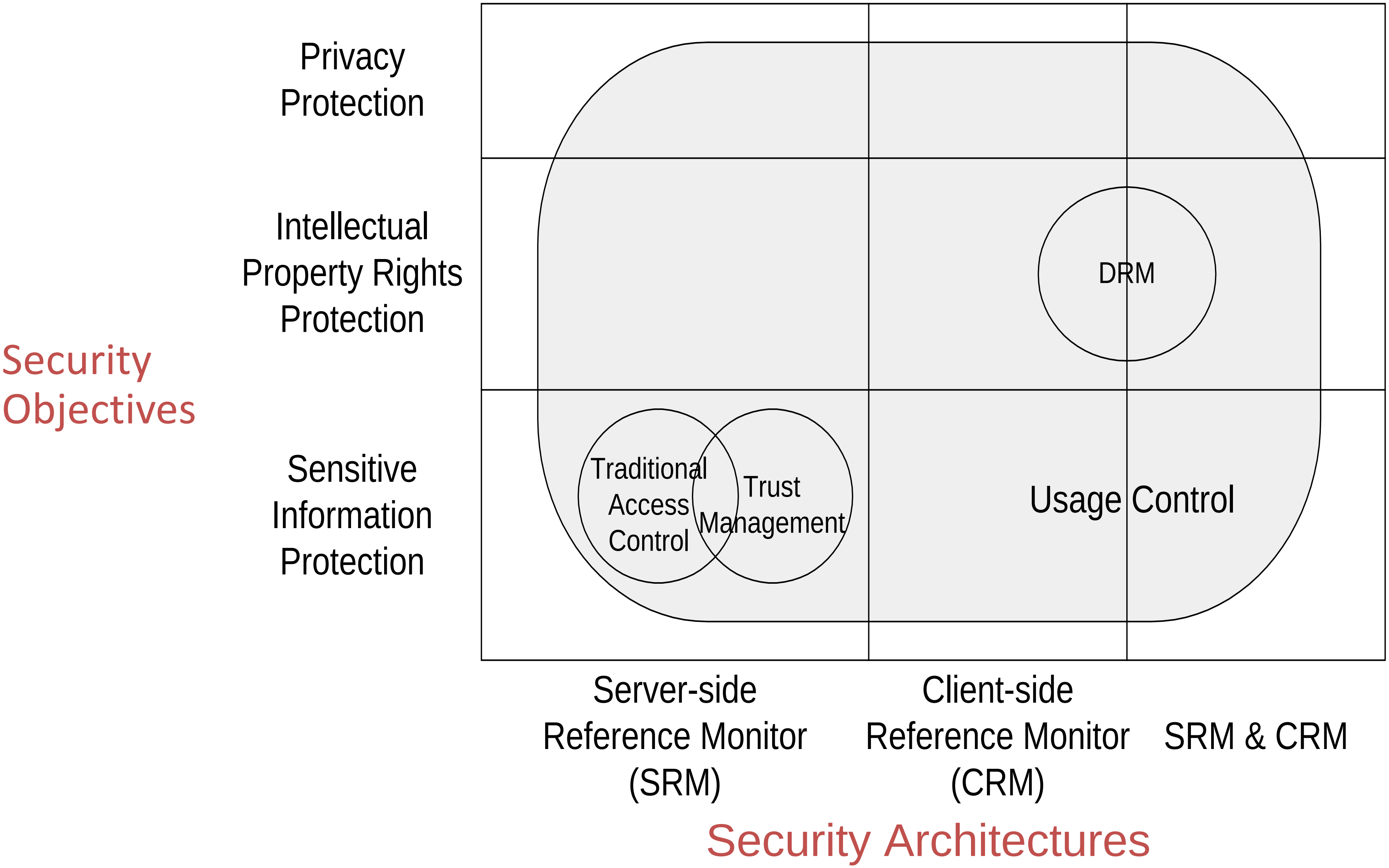
RABAC wrt NIST RBAC Model



Select core RBAC. Option: Advanced Review
Adhere to Dependency
Core RBAC
Hier. RBAC a. Limited b.General
SSD Relations a. w/hier. b. wo/hier
DSD Relations
Permission Filter Policy (PFP)
Requirement package
Choose a or b. Option: Advanced review

UCON Model

Usage Control (UCON)



Privacy Protection

Intellectual Property Rights Protection

Sensitive Information Protection

Security Objectives

DRM

Traditional Access Control

Trust Management

Usage Control

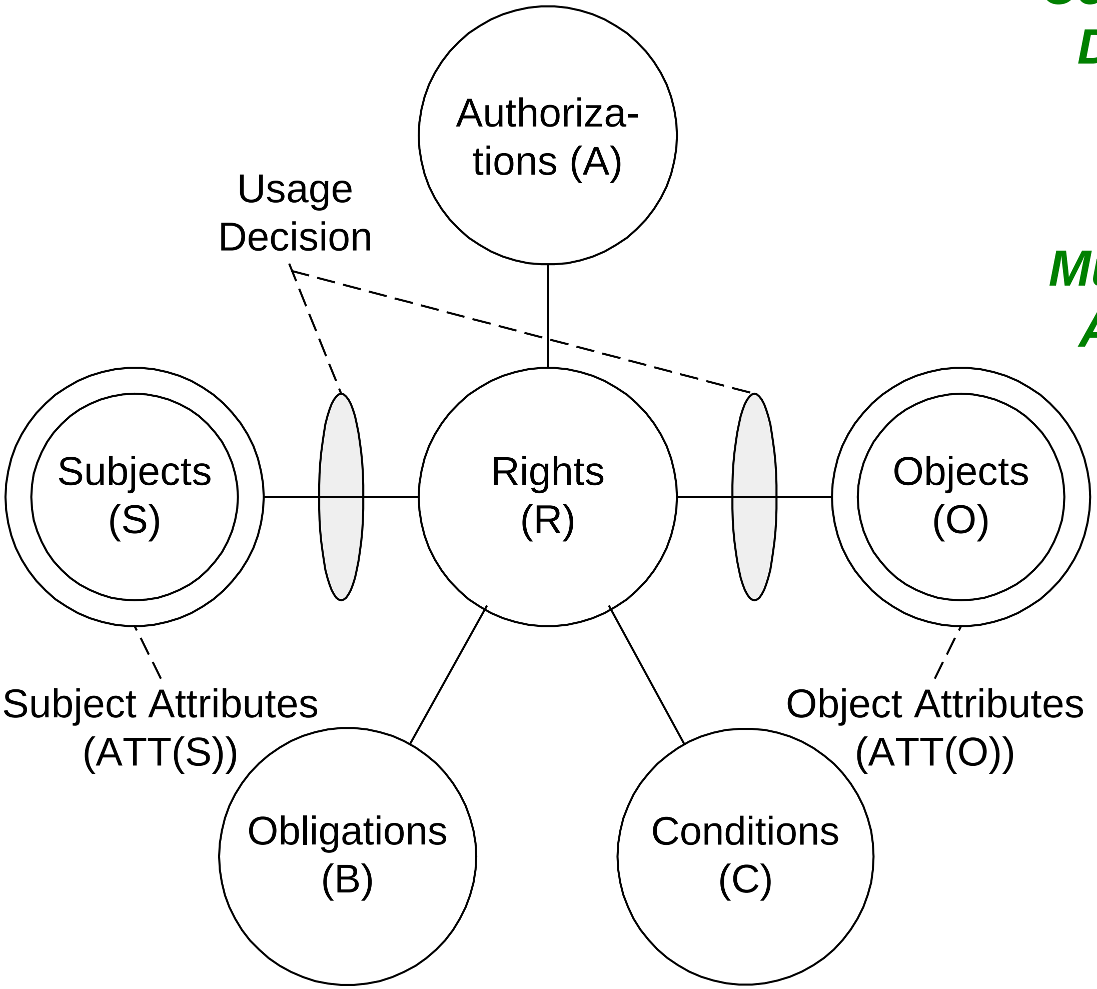
Server-side Reference Monitor (SRM)

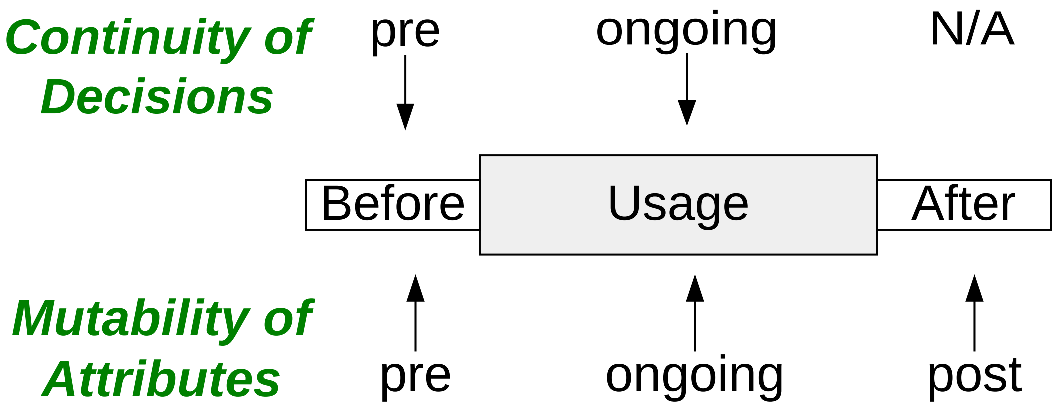
Client-side Reference Monitor (CRM)

SRM & CRM

Security Architectures

Building UCON$_{ABC}$ Models





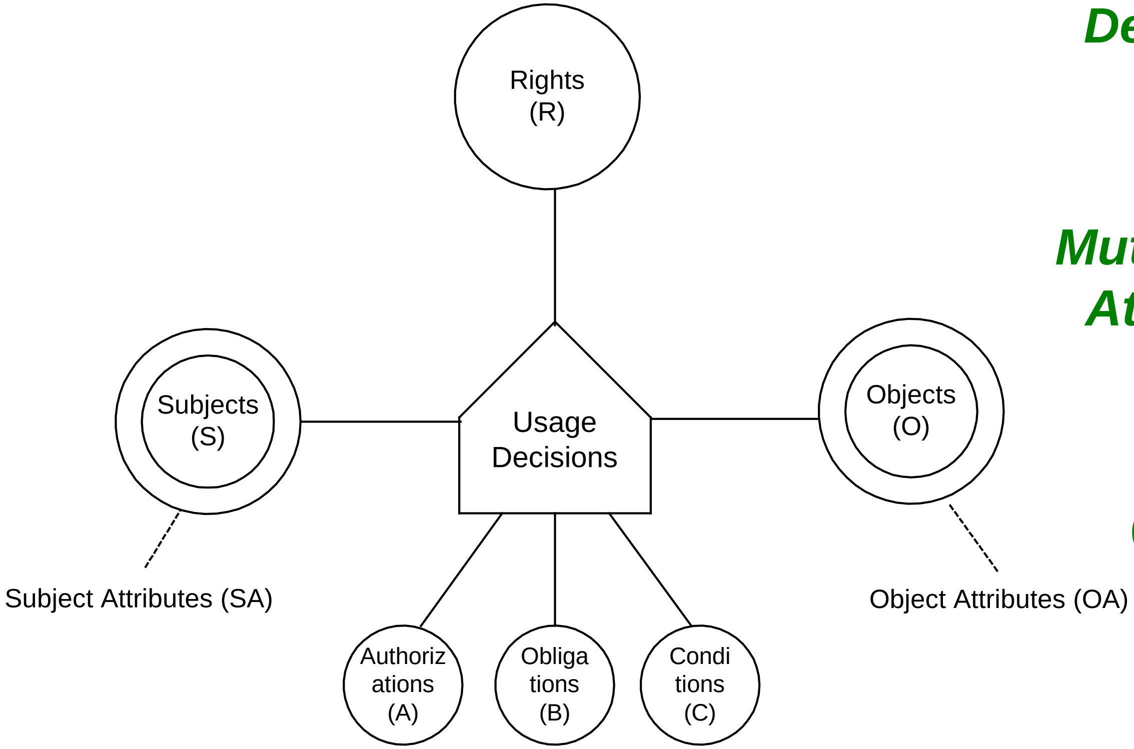
Continuity

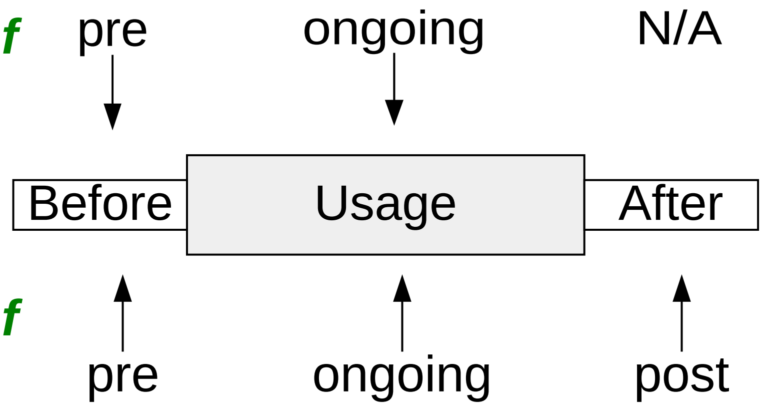
Decision can be made during usage for continuous enforcement

Mutability

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

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

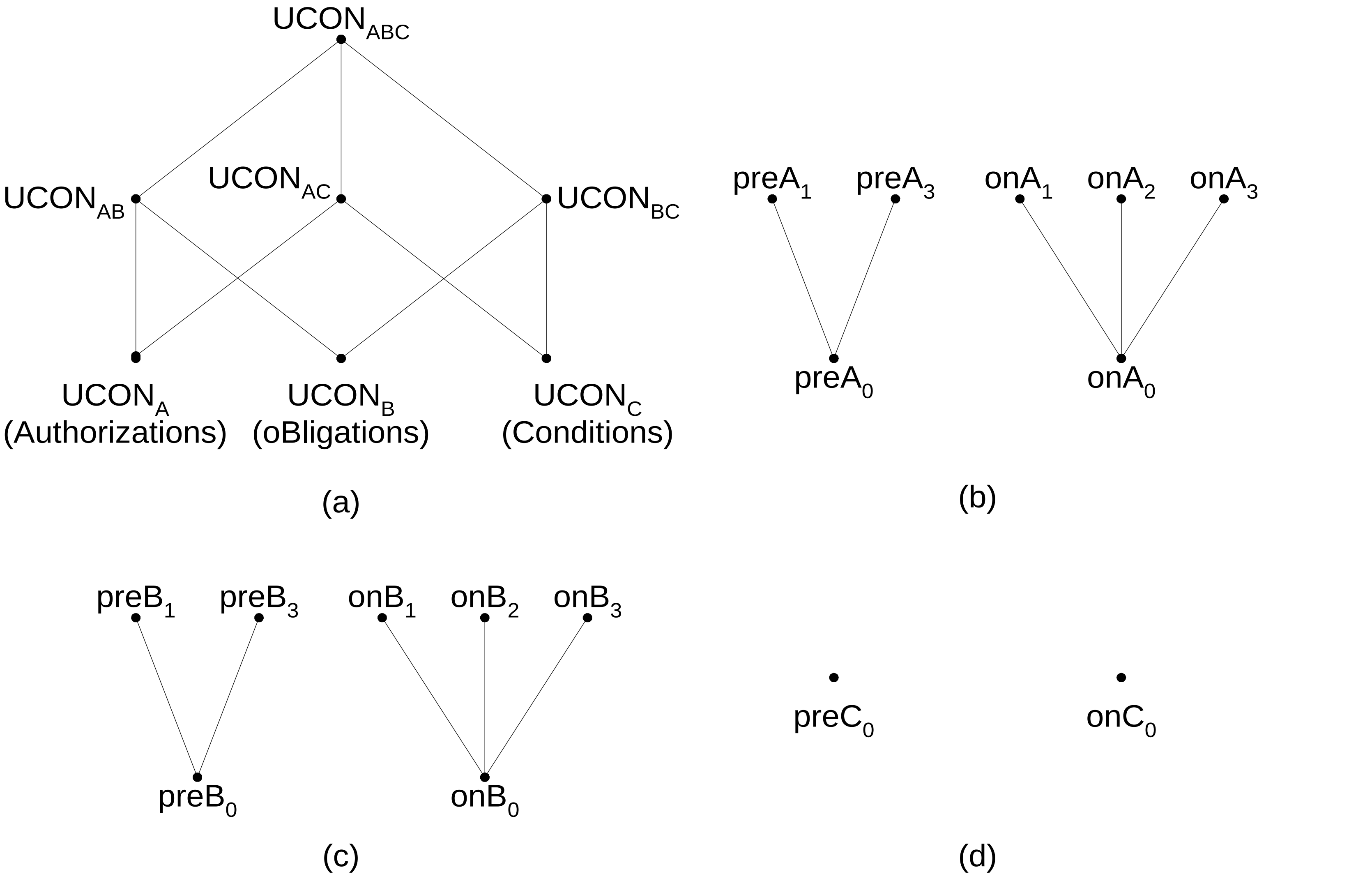
- Long-distance phone (pre-authorization with post-update)
- Pre-paid phone card (ongoing-authorization with ongoing-update)
- Pay-per-view (pre-authorization with pre-updates)
- Click Ad every 30 minutes (ongoing-obligation with ongoing-updates)
- Business Hours (pre-/ongoing-condition)

UCON_{ABC} Model Space

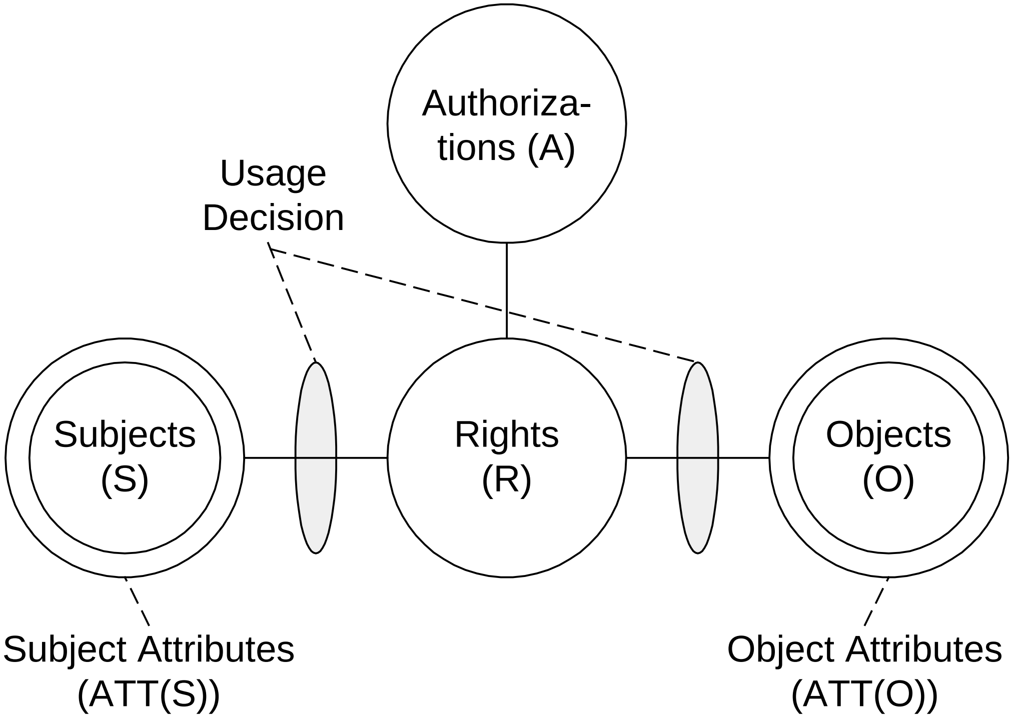
	0(Immutable)	1(pre)	2(ongoing)	3(post)
preA	Y	Y	N	Y
onA	Y	Y	Y	Y
preB	Y	Y	N	Y
onB	Y	Y	Y	Y
preC	Y	N	N	N
onC	Y	N	N	N

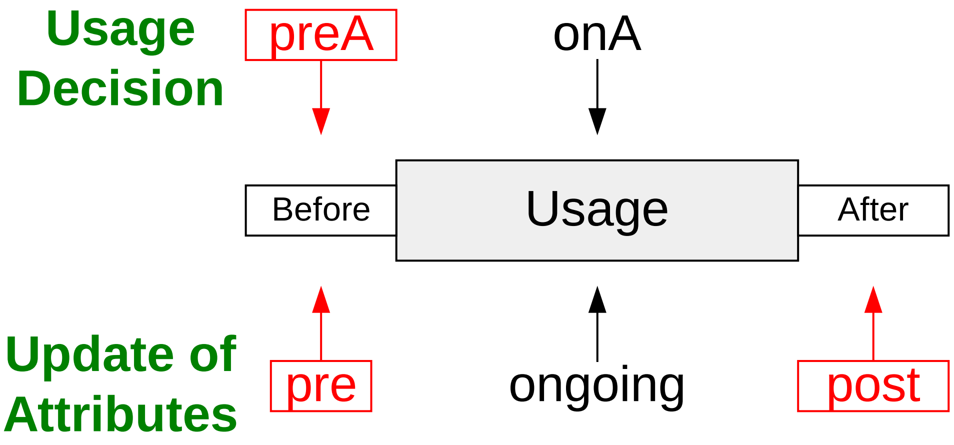
N : Not applicable

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



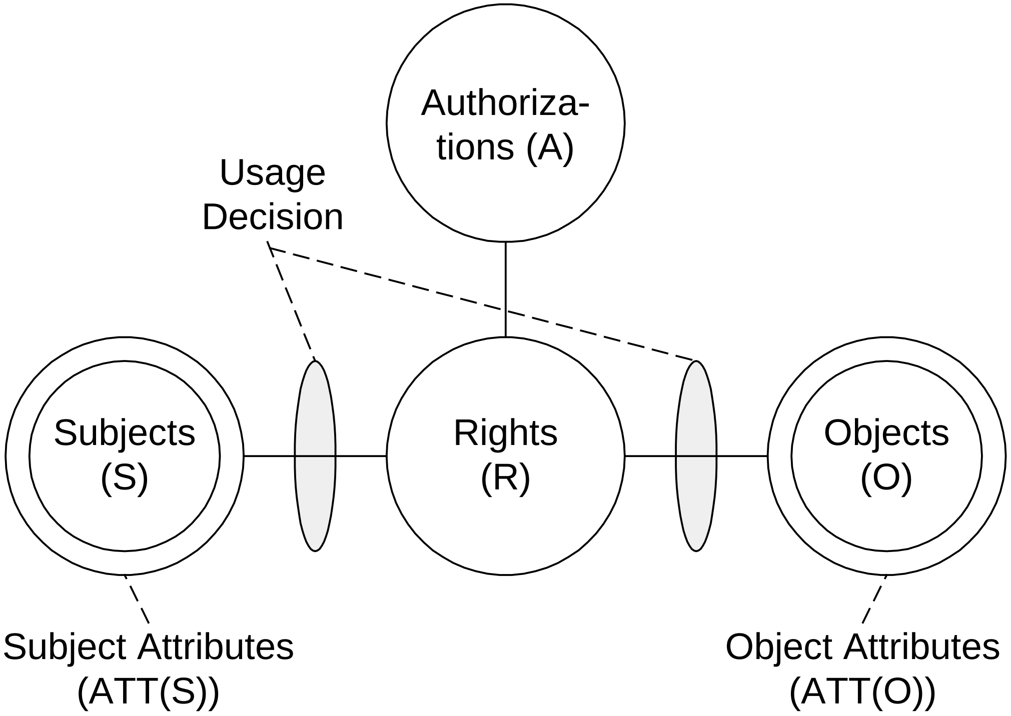
$\text{UCON}_{\text{preA}}$

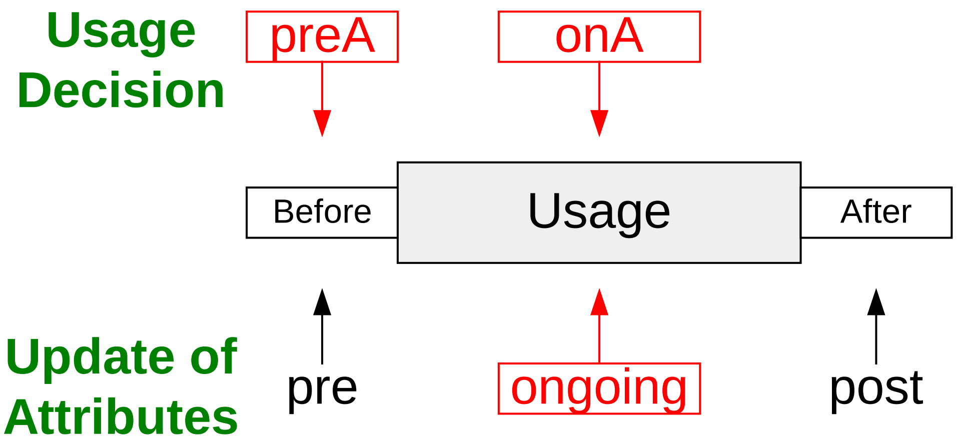




- Online content distribution service
  - Pay-per-view (pre-update)
  - Metered payment (post-update)

UCON$_{onA}$





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

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

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

UCON$_{preA0}$: MAC Example

- *L* is a lattice of security labels with dominance relation ≥
- *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: $RBAC_1$ $(UCON_{preA0})$

- $P = \{(o,r)\}$
- *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$_{preA3}$: DRM Example

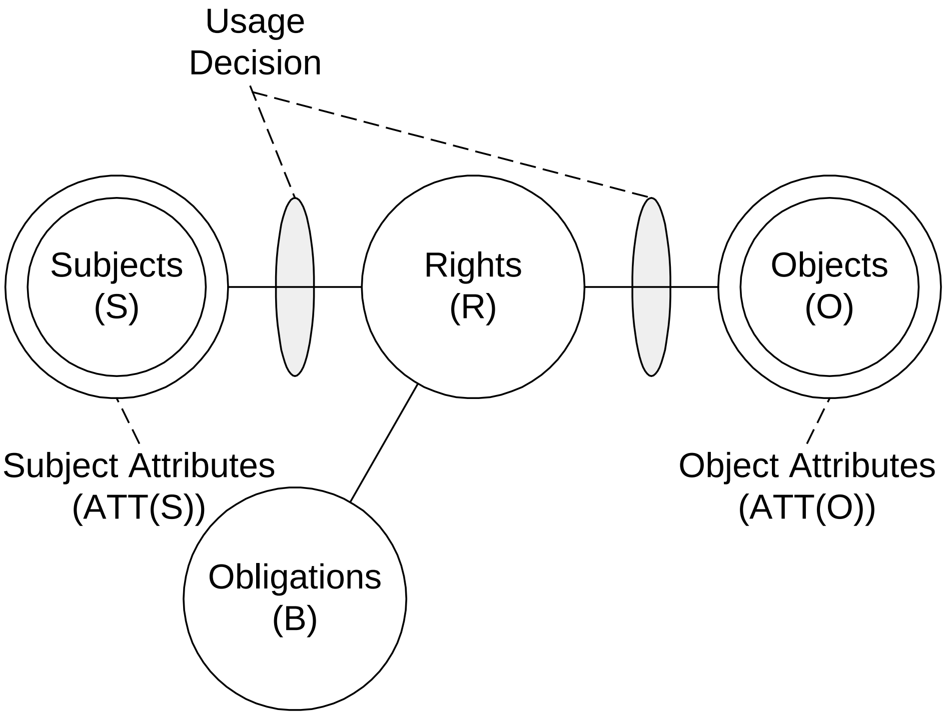
- Membership-based metered payment
  - *M* is a set of money amount
  - *ID* is a set of membership identification numbers
  - *TIME* is a current usage minute
  - *member: S* → *ID*
  - *expense: S* → *M*
  - *usageT: S* → *TIME*
  - *value: O x R* → *M* (a cost per minute of r on o)
  - *ATT(s): {member,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

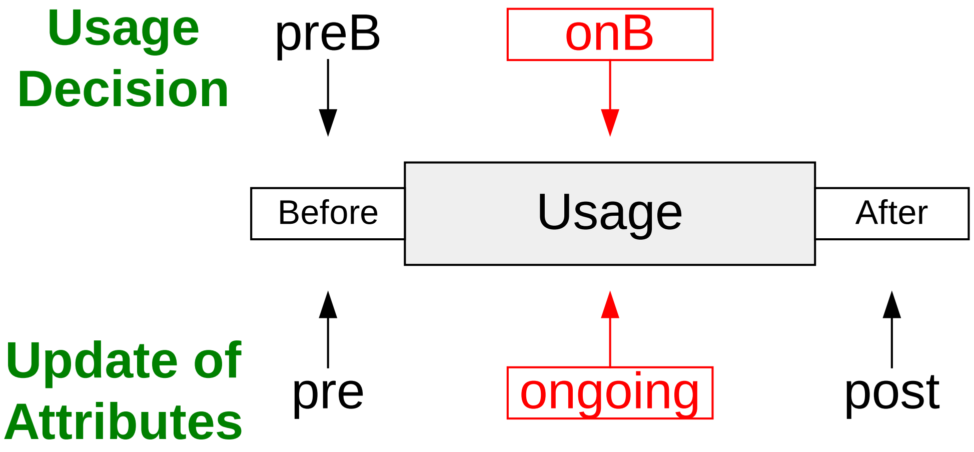
$\mathrm{UCON}_{\mathrm{onA}}$: ongoing-Authorizations Model

- $\mathrm{UCON}_{\mathrm{onA0}}$
  - $S, O, R, ATT(S), ATT(O)$ and onA;
  - $allowed(s,o,r) \Rightarrow true;$
  - $Stopped(s,o,r) \Leftarrow \neg onA(ATT(s),ATT(o),r)$
- $\mathrm{UCON}_{\mathrm{onA1}}$, $\mathrm{UCON}_{\mathrm{onA2}}$, $\mathrm{UCON}_{\mathrm{onA3}}$
  - *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$_B$


Usage
Decision
Subjects
(S)
Rights
(R)
Objects
(O)
Subject Attributes
(ATT(S))
Object Attributes
(ATT(O))
Obligations
(B)


Usage
Decision
preB
onB
Before
Usage
After
Update of
Attributes
pre
ongoing
post

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

$\text{UCON}_{\text{preB0}}$: 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 $getPreOBL(s,o,r)=\varnothing$;
- $allowed(s,o,r) \Rightarrow preB(s,o,r)$.

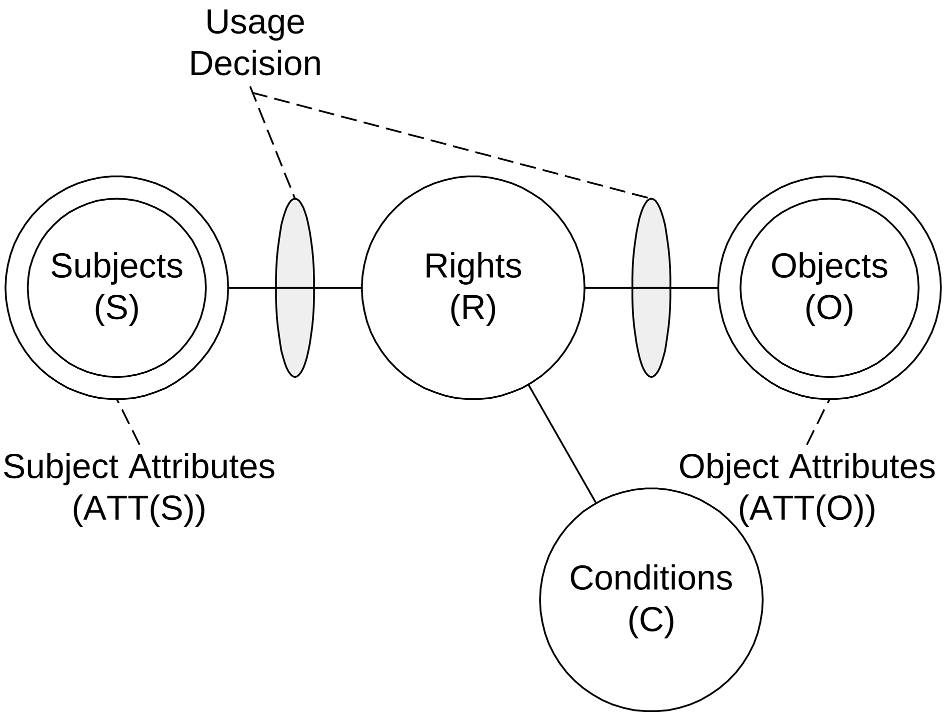
- Example: License agreement for a whitepaper download

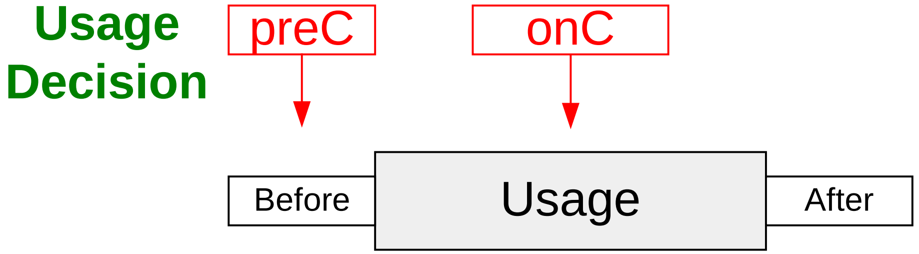
UCON$_{onB0}$: ongoing-oBligations w/ no update

- *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 x OBO x OB x T*;
- *onFulfilled: OBS x OBO x OB x T* $\rightarrow$ *{true,false}*;
- *getOnOBL: S x O x 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);
- $preConChecked: preCON \rightarrow \{true,false\}$;
- $getPreCON: S \text{ x } O \text{ x } R \rightarrow 2^{preCON}$;
- $preC(s,o,r) = \Lambda_{preCon\_i \in getPreCON(s,o,r)} preConChecked(preCon_i)$;
- $allowed(s,o,r) \Rightarrow preC(s,o,r)$.

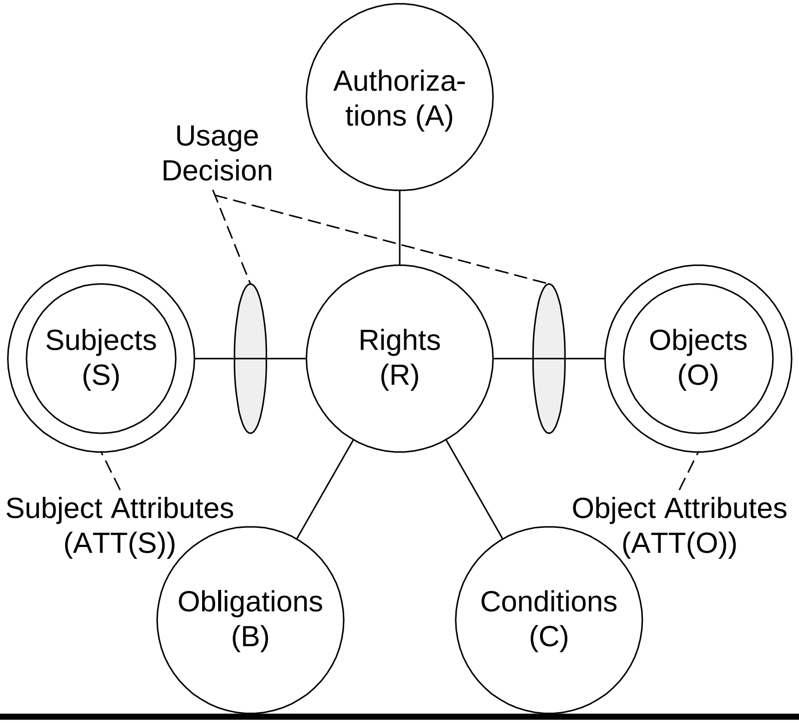
- Example: location checks at the time of access requests

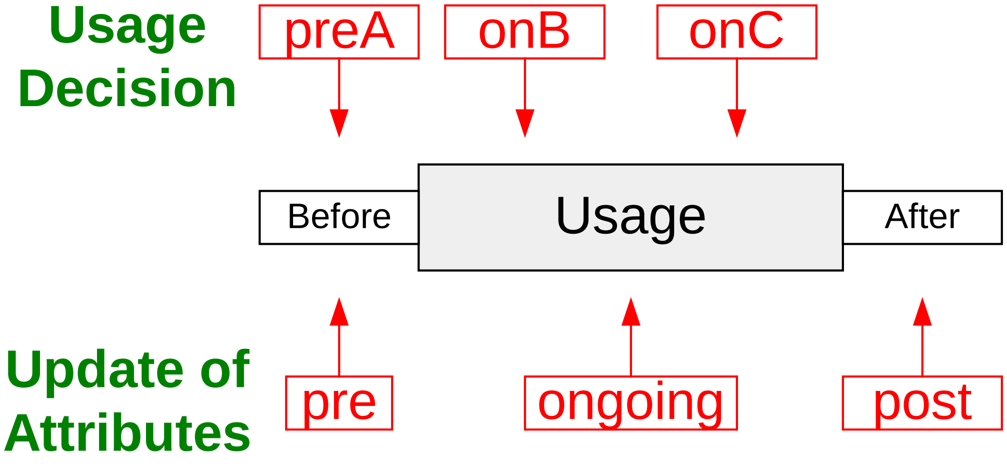
UCON$_{onC0}$: ongoing-Condition model

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

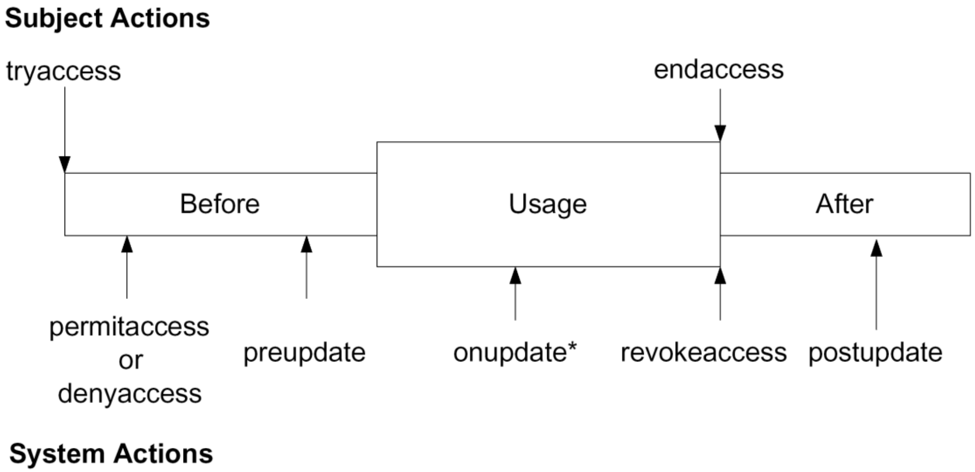


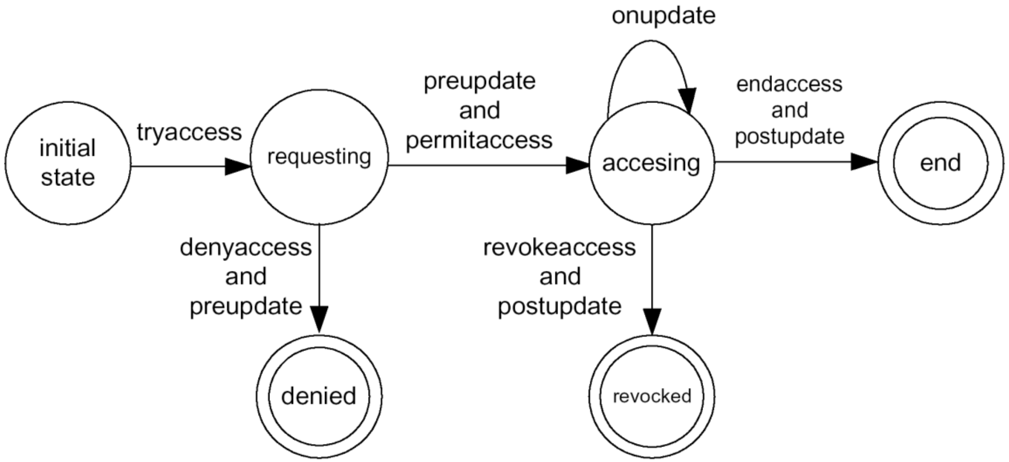


- Free ISP
  - Membership is required (pre-authorization)
  - Have to click Ad periodically while connected (on-obligation, on-update)
  - Free member: no evening connection (on-condition), no more than 50 connections (pre-update) or 100 hours usage per month (post-updates)

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.


Subject Actions
tryaccess
endaccess
Before
Usage
After
permitaccess
or
denyaccess
preupdate
onupdate*
revokeaccess
postupdate
System Actions


onupdate
initial state
tryaccess
requesting
preupdate and permitaccess
accesing
endaccess and postupdate
end
denyaccess and preupdate
denied
revokeaccess and postupdate
revocked