Label-Based Access Control: An ABAC Model with Enumerated Authorization Policy

Prosunjit Biswas, Ravi Sandhu and Ram Krishnan
University of Texas at San Antonio

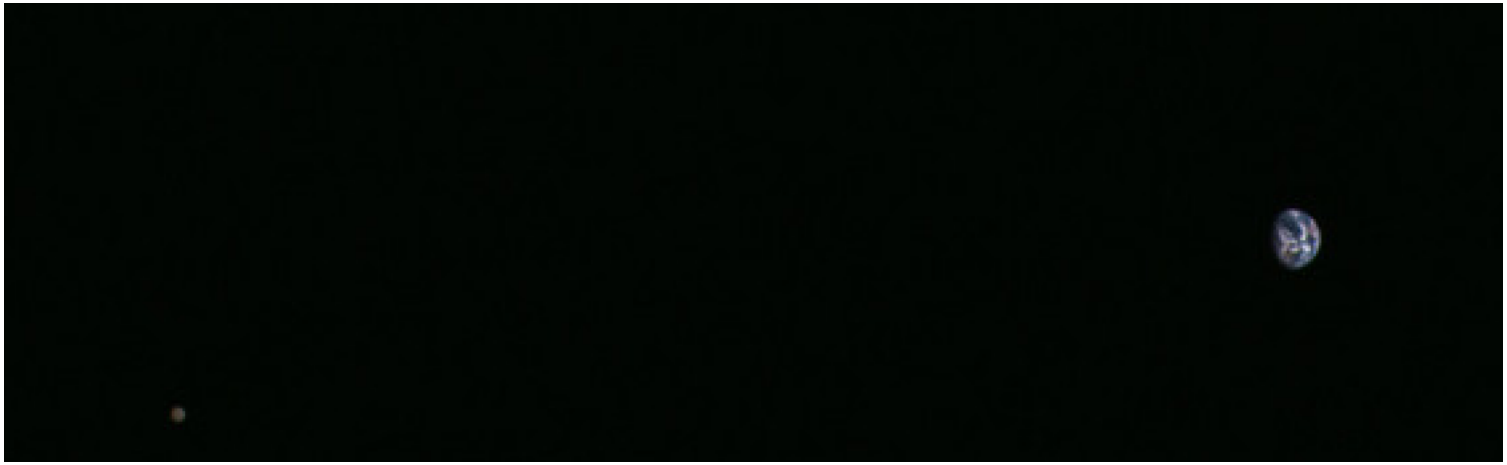
1st Workshop on Attribute Based Access Control (ABAC 2016)

Outline

- Summary
- Background & motivation
- Enumerated authorization policy ABAC model
- Relationship with existing models
- Expressive power of LaBAC
- Conclusion

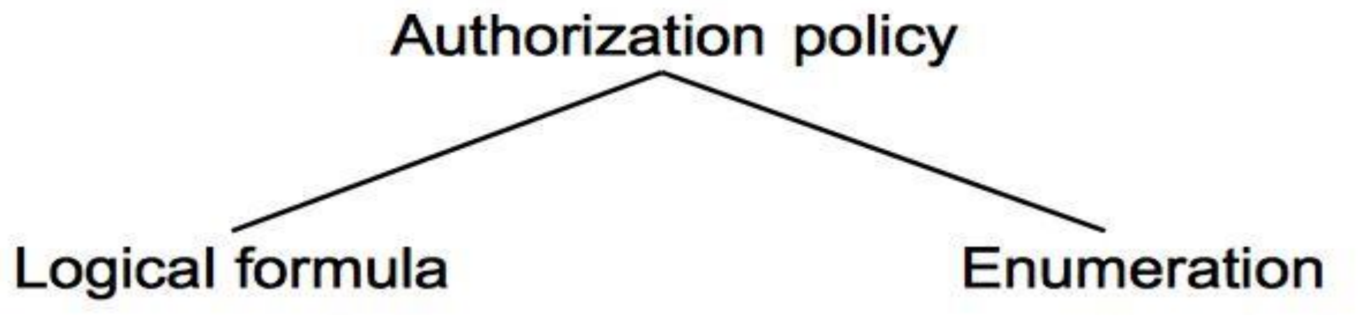
Summary

❑ We present an enumerated authorization policy ABAC model and understand its relationship with traditional access control models.



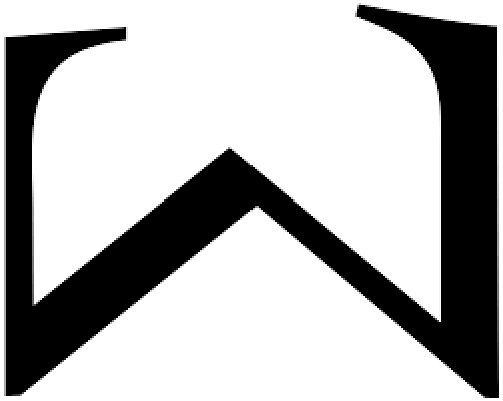
Background and Motivation

authorization policy



- Boolean expression
- E.g.: age(u)>18
- Models: ABAC_{α}, HGABAC

- Set of tuples
- {(age(u),19), (age(u),20), …. (age(u),100)} *[assuming range upper bound <=100]*
- *Models: Policy Machine, 2-sorted-RBAC*



Logical-formula Auth. Policy

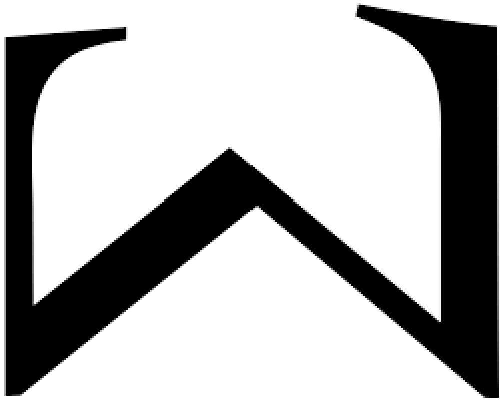
Many ways to set up a policy - ***Auth****$_{read}$*

(***Auth****$_{read}$* *allows manager* to read *TS* objects from *home* or *office)*.

(i) $mng \in role(u) \wedge (office \in location(u)\ \vee home \in location(u)\ ) \wedge TS \in sensitivity(o)$

(ii) $((mng \in role(u) \wedge office \in location(u)) \vee (mng \in role(u) \wedge home \in location(u))\ )$
$\wedge TS \in sensitivity(o)$

(iii) $((mng \in role(u) \wedge office\in \in location(u) \wedge TS \in sensitivity(o)) \vee$
$((mng \in role(u) \wedge home \in location(u) \wedge TS \in sensitivity(o))$



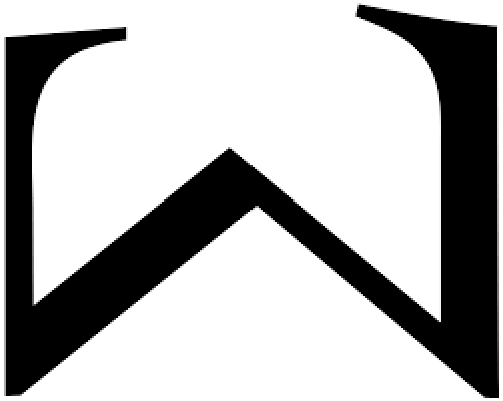
Logical-formula Auth. Policy

Update ***Auth***$_{\textbf{\textit{read}}}$ *so that*
manager can no longer read *TS* objects from *home*

(i) $mng \in role(u) \wedge (office \in location(u) \vee home \in location(u)\ ) \wedge TS \in sensitivity(o)$

(ii) $((mng \in role(u) \wedge office \in location(u)) \vee (mng \in role(u) \wedge home \in location(u))\ )$
$\wedge TS \in sensitivity(o)$

(iii) $((mng \in role(u) \wedge office{\in} \in location(u) \wedge TS \in sensitivity(o)) \vee$
$((mng \in role(u) \wedge home \in location(u) \wedge TS \in sensitivity(o))$

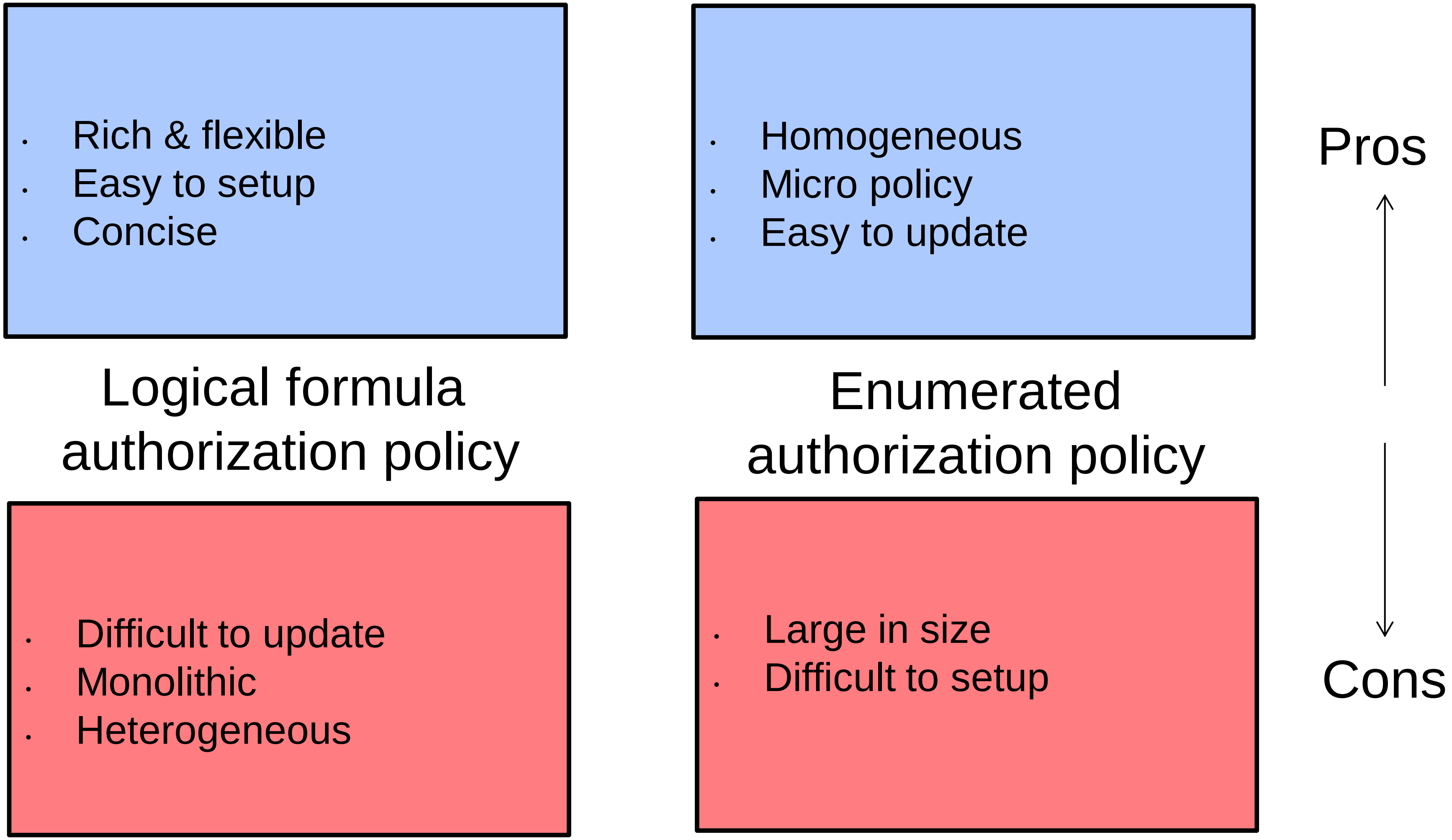


Enumerated Auth. Policy

□ Auth$_{read}$ ≡ {(mng, home, TS), (mng,office,TS)}

□ Auth\`$_{read}$ ≡ { ~~(mng, home, TS),~~ (mng,office,TS)}

Logical formula vs enumerated policy



- Rich & flexible
- Easy to setup
- Concise

- Homogeneous
- Micro policy
- Easy to update

Pros

Logical formula authorization policy

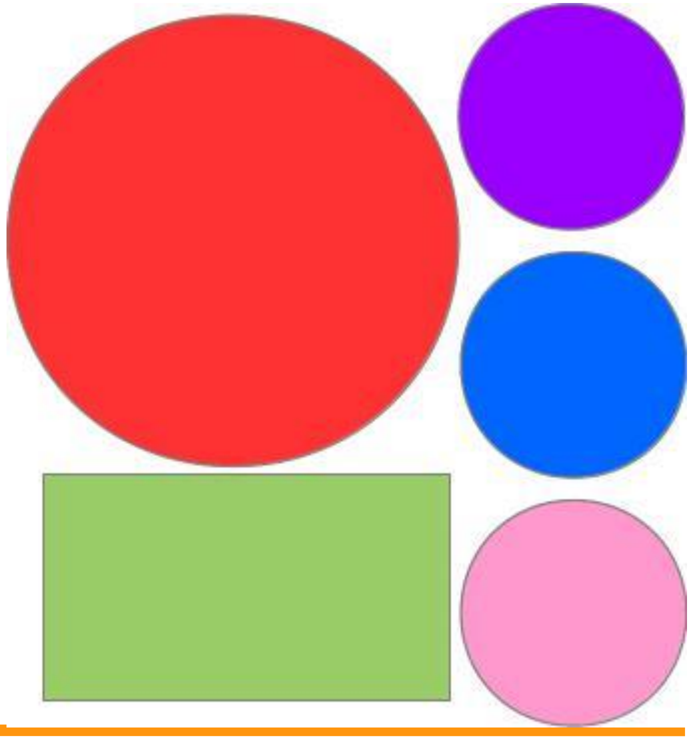
Enumerated authorization policy

- Difficult to update
- Monolithic
- Heterogeneous

- Large in size
- Difficult to setup

Cons

LaBAC: Label-Based Access Control



Characteristics

- Label vs Attribute
  - Labels are attributes with tighter semantics
- Salient features of LaBAC
  - Finite domain ABAC
  - Simple enumerated ABAC model

Family of LaBAC models

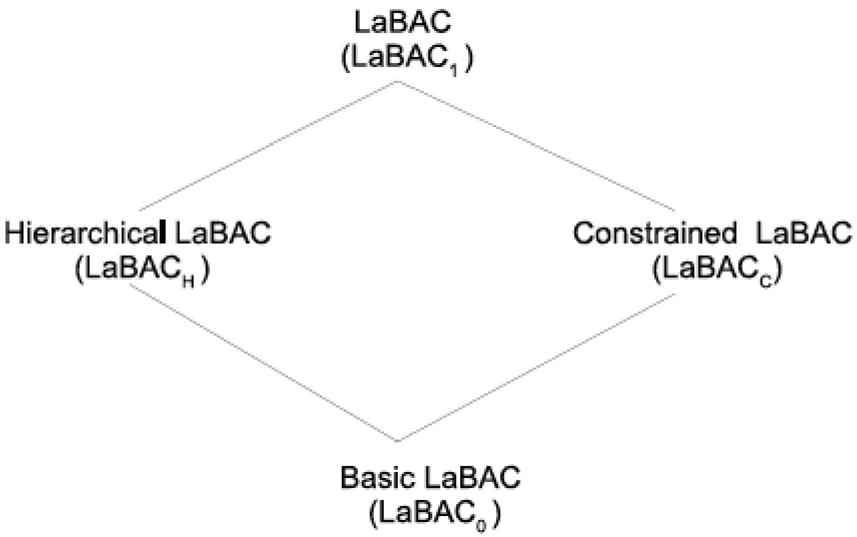


Figure 2: Family of LaBAC models

LaBAC: Core model

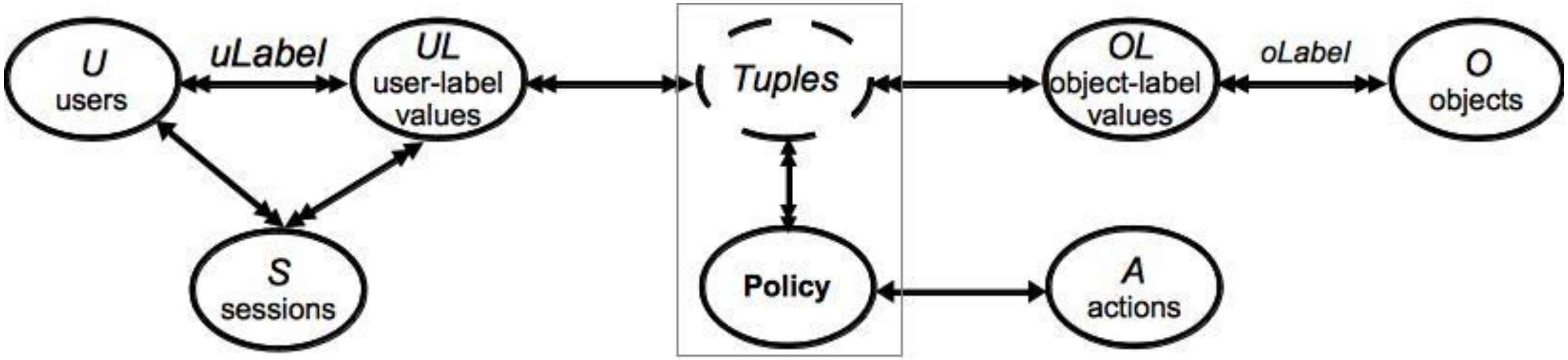


Figure 1

Salient Characteristics:

1. One user and object attribute
2. Atomic valued tuples
3. Tuples represent micro-policies

Examples

UL={manager,employee}

OL={TS,S}

Tuple1= (manager,TS)

$\text{Policy}_{\text{read}}$ = {tuple1, tuple2…}

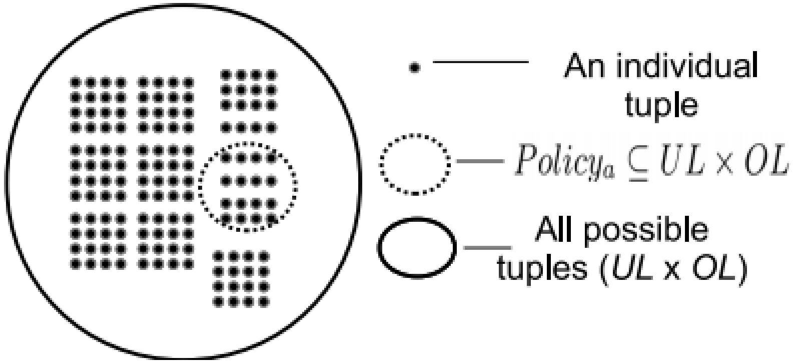

An individual tuple
$Policy_a \subseteq UL \times OL$
All possible tuples (UL x OL)

Figure 2

LaBAC: Hierarchical model

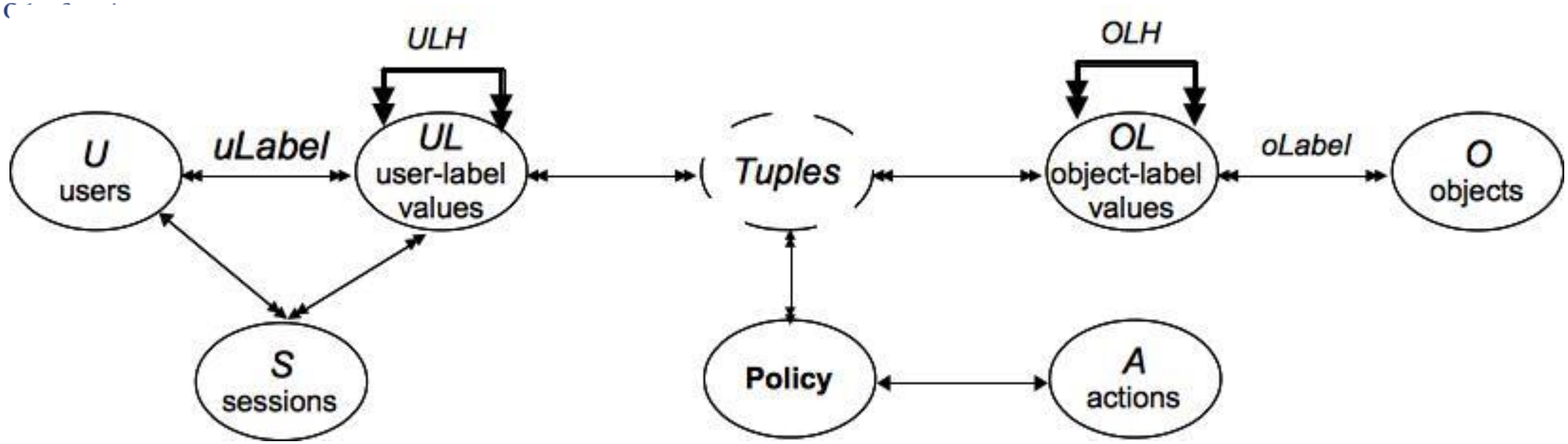


Figure 1

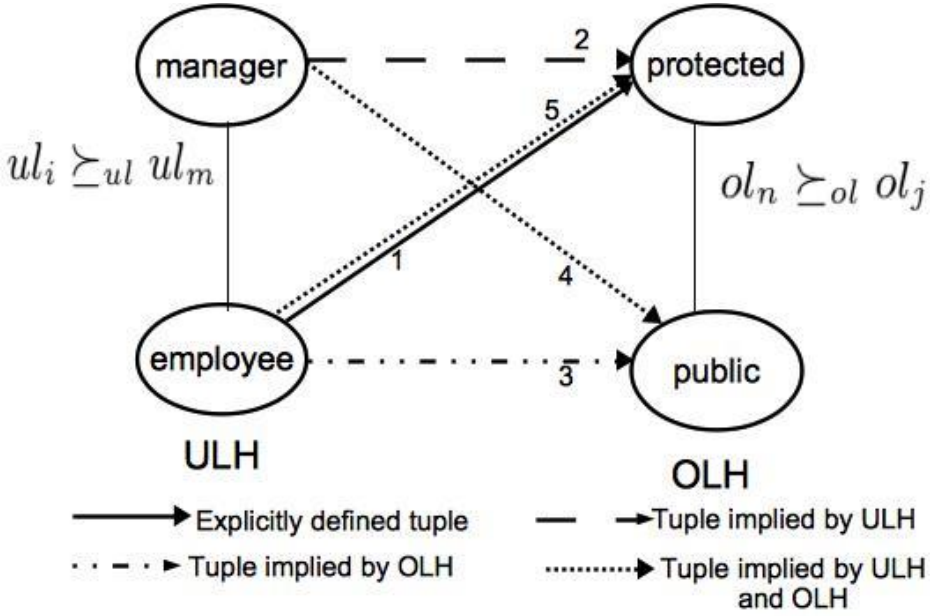


Figure 2

Examples

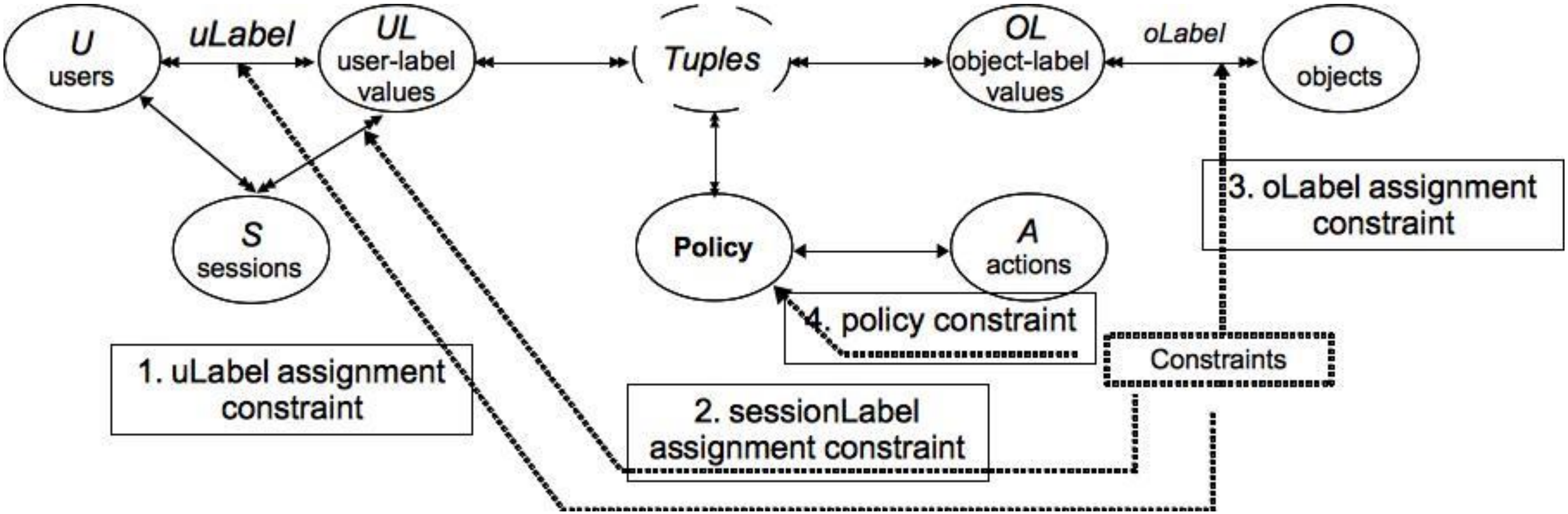
ULH={(manager,employee)}

OLH={(protected, public)}

Policy$_a$ = {(employee,protected)}

ImpliedPolicy$_a$ = { (employee, protected), (manager, proteced), (employee,public), (manager, public)}

LaBAC: Constrained model



Examples

uLabel assgn. cons: a user cannot be both manager & director.

Session assgn. cons: at most one value can be activated in a session.

oLabel assgn. cons: A object cannot be both private & public

Policy cons: (employee, TS) can never be used.

Relationship of LaBAC with other enumerated policy models

LaBAC equivalent to 2-sorted-RBAC

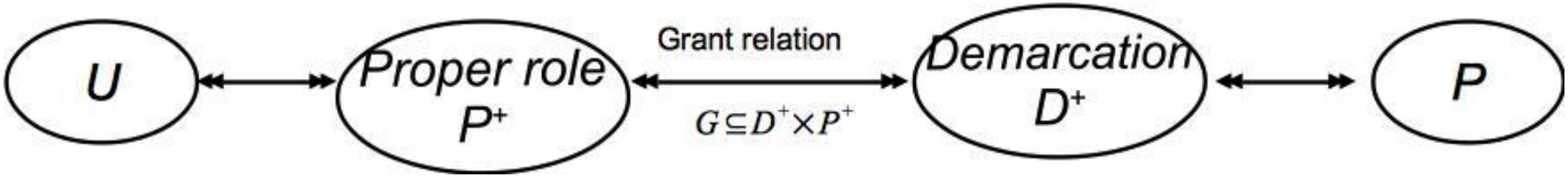


Figure 1: 2-sorted-RBAC

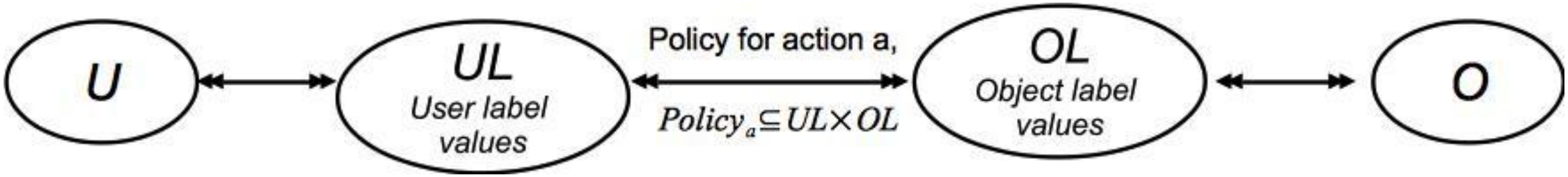


Figure 2: LaBAC

2-sorted-RBAC vs LaBAC:

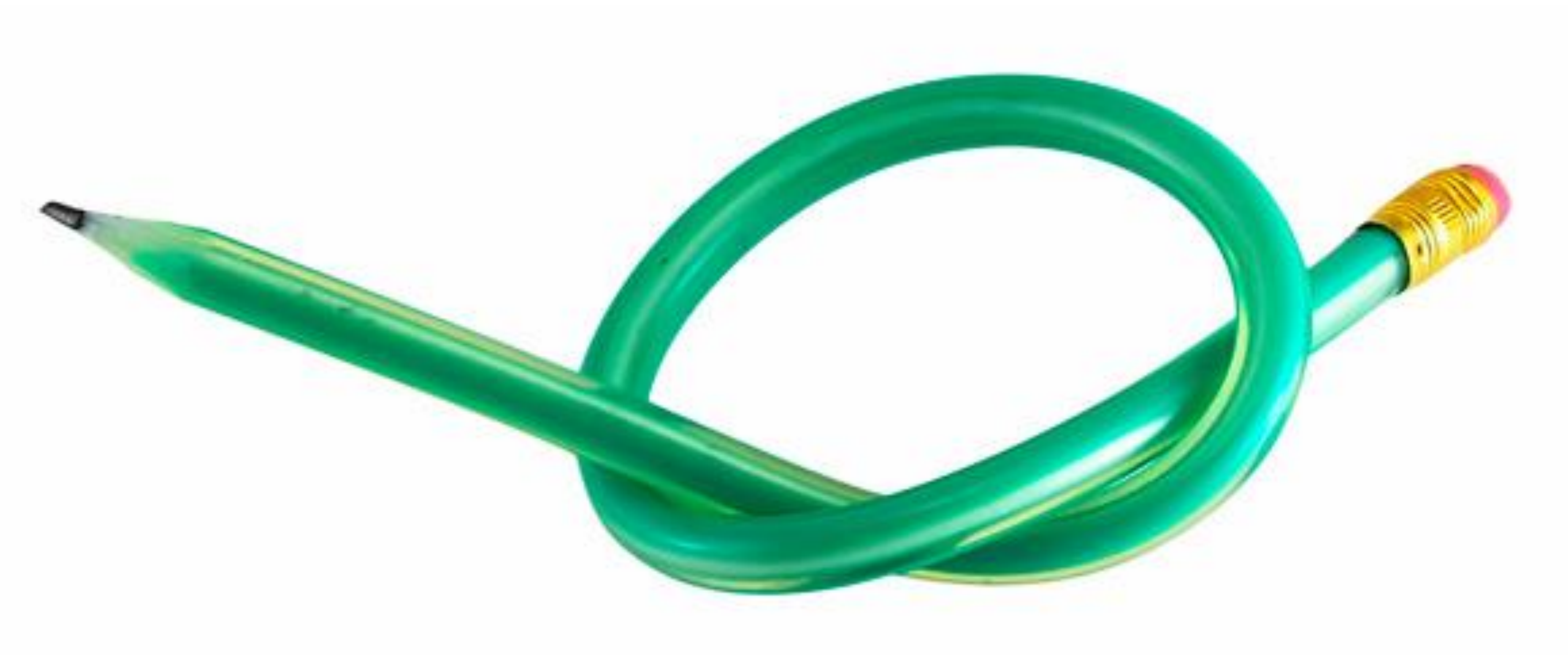
1. Use of attributes
2. Separation of object and action from permission

LaBAC as an instance of Policy Machine

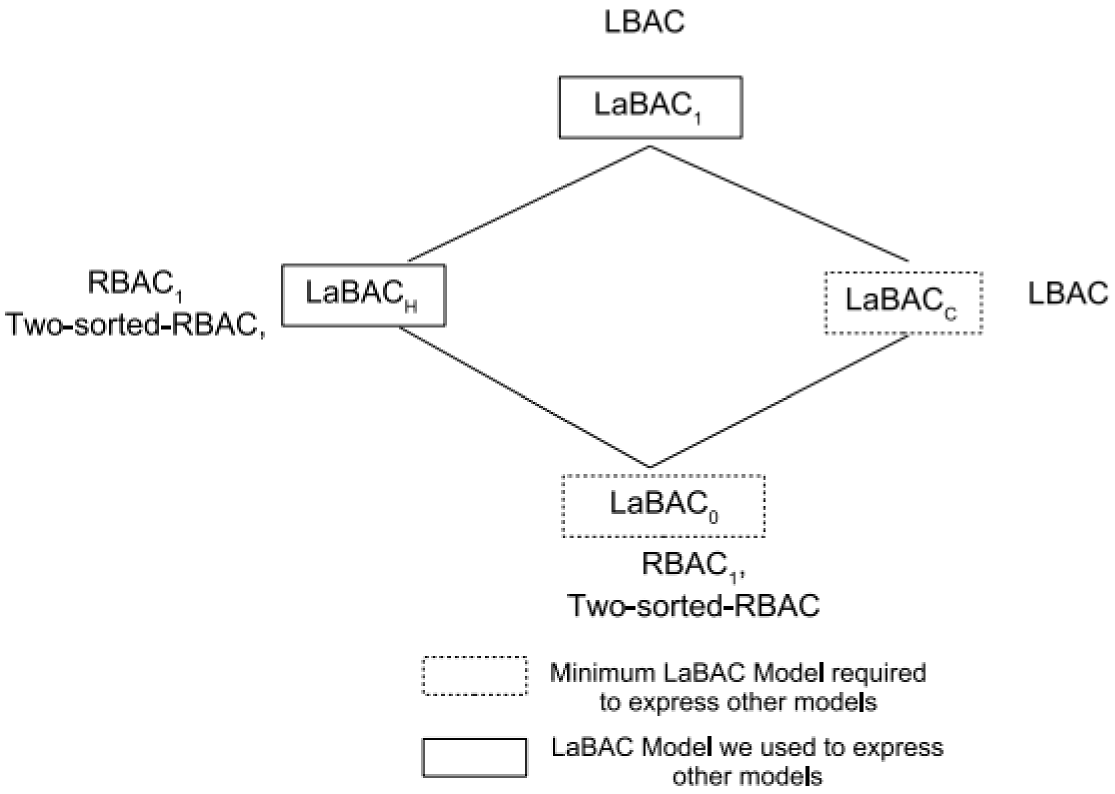
- Policy Machine $_{\text{mini}}$
  - Only ASSIGN and ASSOCIATION relation
  - Default policy class

- Configuration of LaBAC in Policy Machine $_{\text{mini}}$

Flexibility in expressing traditional models



Expressiveness of LaBAC models



LBAC

LaBAC$_1$

RBAC$_1$
Two-sorted-RBAC,

LaBAC$_H$

LaBAC$_C$

LBAC

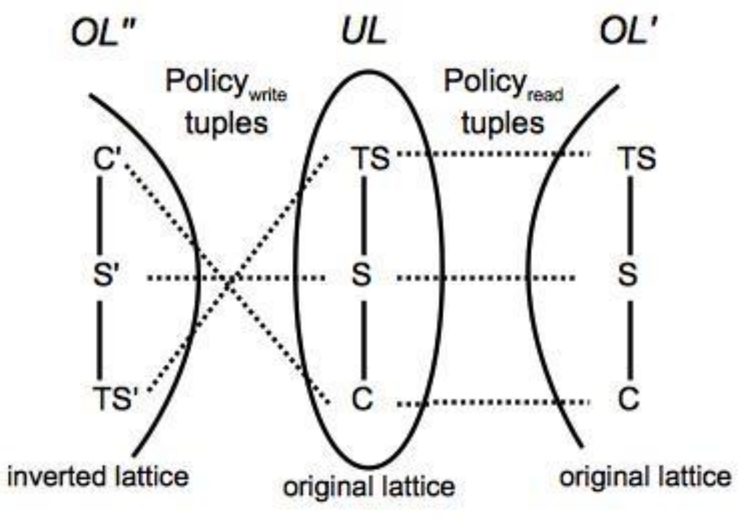
LaBAC$_0$

RBAC$_1$,
Two-sorted-RBAC

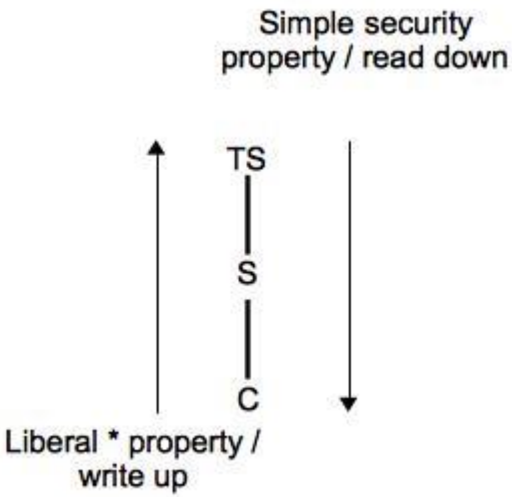
Minimum LaBAC Model required to express other models

LaBAC Model we used to express other models

LBAC in LaBAC



$$OL = OL' \cup OL''$$

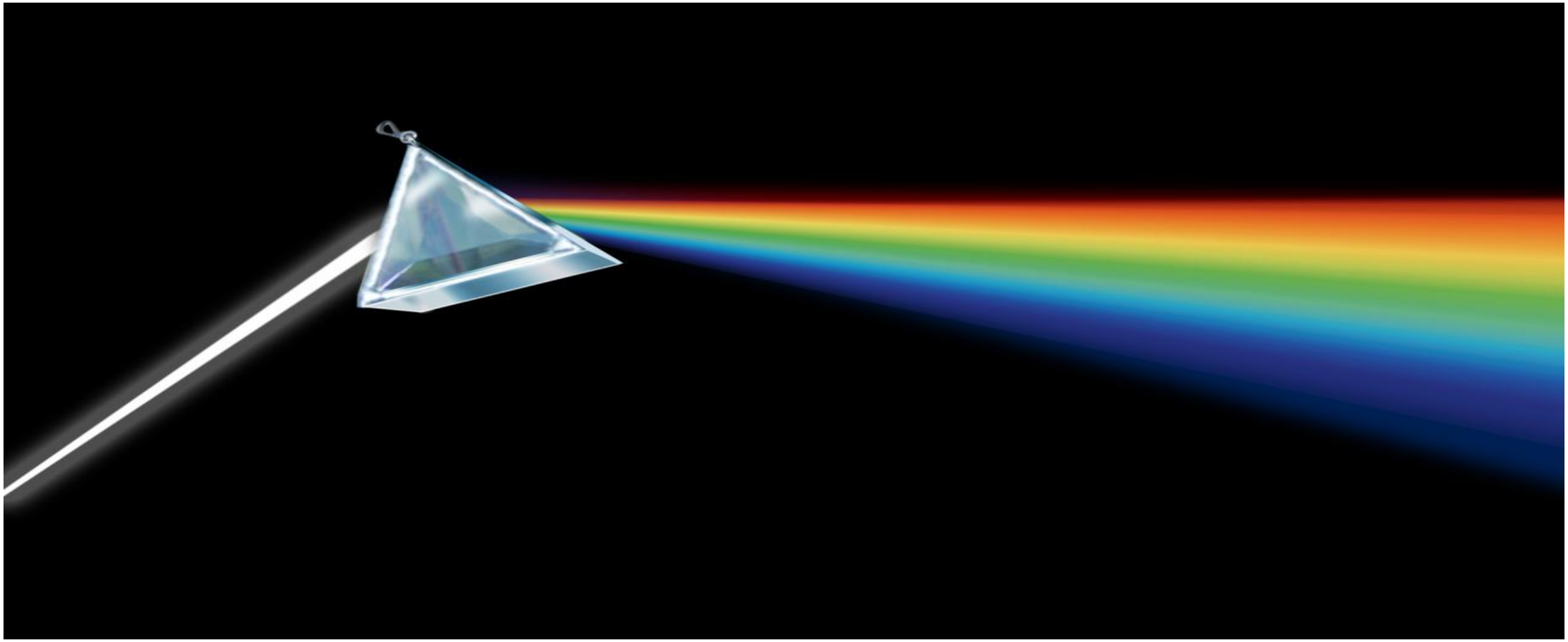


LBAC assumptions:

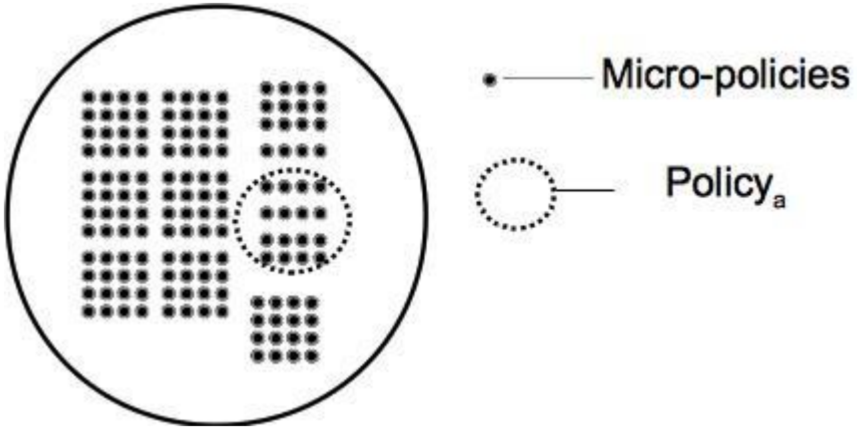
1. Tranquility
2. Object operation: creation only

$$|UL| = |SC| \text{ and } |OL| = 2 * |SC|$$
$$|Policy| = 2\ (Policy_{read} \text{ and } Policy_{write})$$

Micro-policy in LaBAC



Micro-policy in LaBAC



- ❑ micro-policy as the smallest unit of administration
- ❑ Example of a micro-policy: (manager, TS)

What is next

❑ Any other form of representation for authorization policy?

❑ How expressive power of enumerated authorization policy is compared with that of logical-formula auth. policy?

❑ What would be the cost of storing large number of enumerated tuples?

Thank
you!