Expressive Power, Safety and Cloud Implementation of Attribute and Relationship Based Access Control Models

Dissertation Defense: Tahmina Ahmed

Dissertation Committee:
Dr. Ravi Sandhu, Supervising Professor
Dr. Jianwei Niu
Dr. Gregory White
Dr. Weining Zhang
Dr. Ram Krishnan

Outline

- Introduction
- Comparison of ReBAC and ABAC
- Object-to-Object Relationship Based Access Control: Model and Multicloud demonstration
- Safety and Expressive Power Comparison of $\boldsymbol{ABAC_{\alpha}}$ and its Enhancements
- Conclusion

Outline

- Introduction
- Comparison of ReBAC and ABAC
- Object-to-Object Relationship Based Access Control: Model and Multicloud demonstration
- Safety and Expressive Power Comparison of $ABAC_{\alpha}$ and its Enhancements
- Conclusion

Access Control Evolution

Discretionary Access Control (DAC), 1970

Mandatory Access Control (MAC), 1970

Role Based Access Control (RBAC), 1995

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

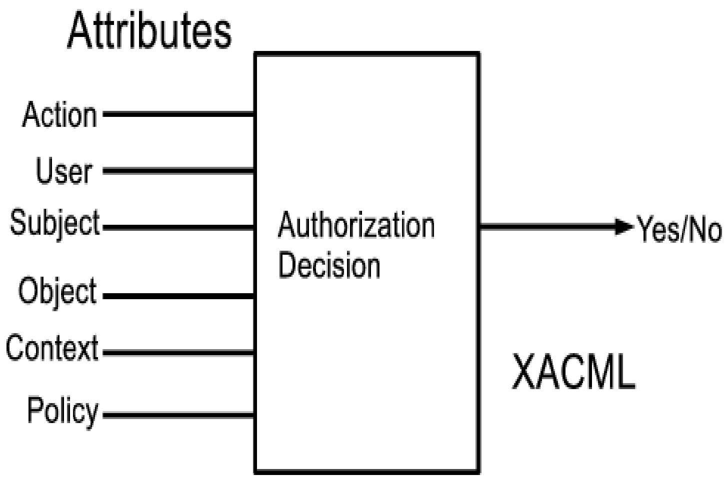
Relationship Based Access Control (ReBAC) ????

Born 1990s

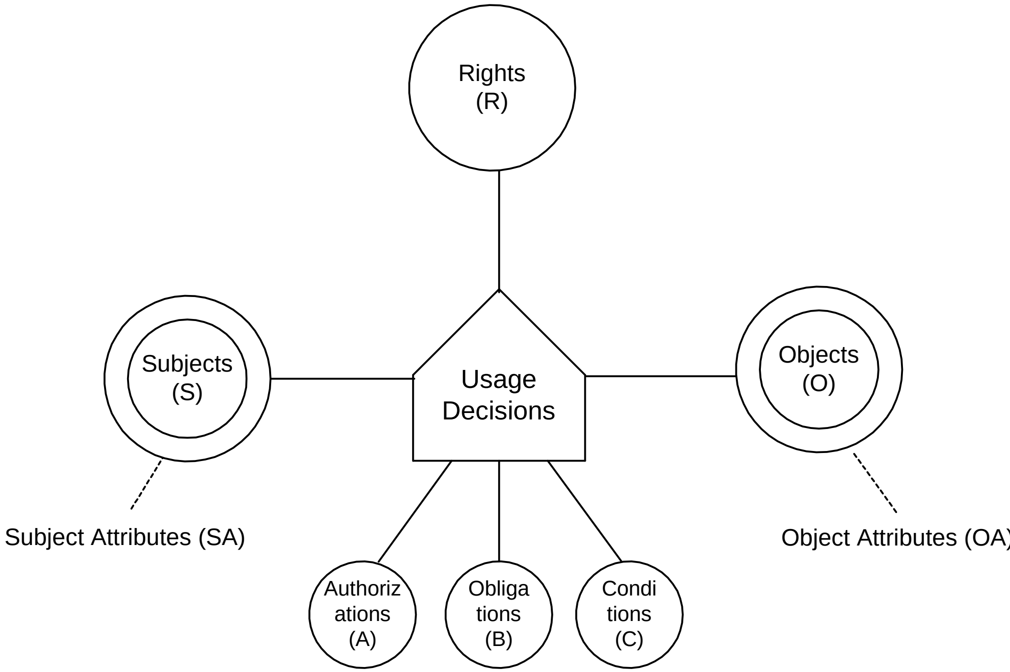
Born mid 2000s

Figure 1: Evolution of Access Control

ABAC: Using Attributes for controlling access



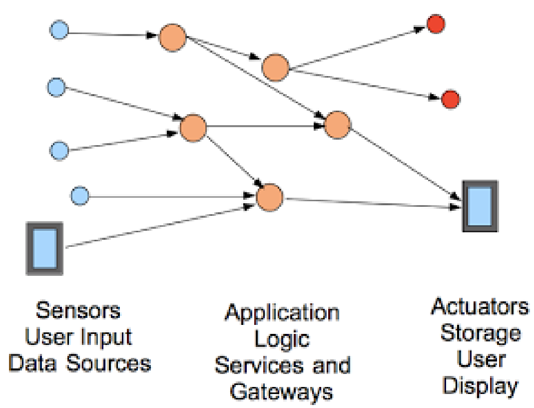
X.500 standard(1994): Manages object information through attributes

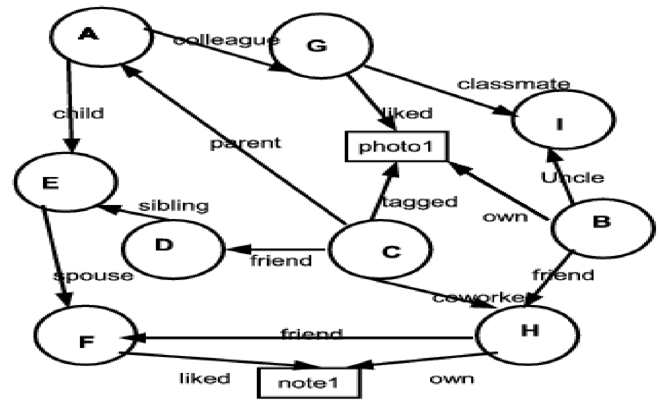


Using attributes for controlling usage of digital resources (Park and Sandhu 2004)

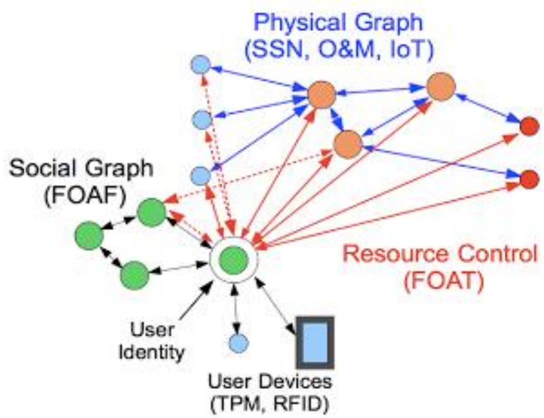
ReBAC:Using Relations for Controlling Access

IoT Application is a Graph

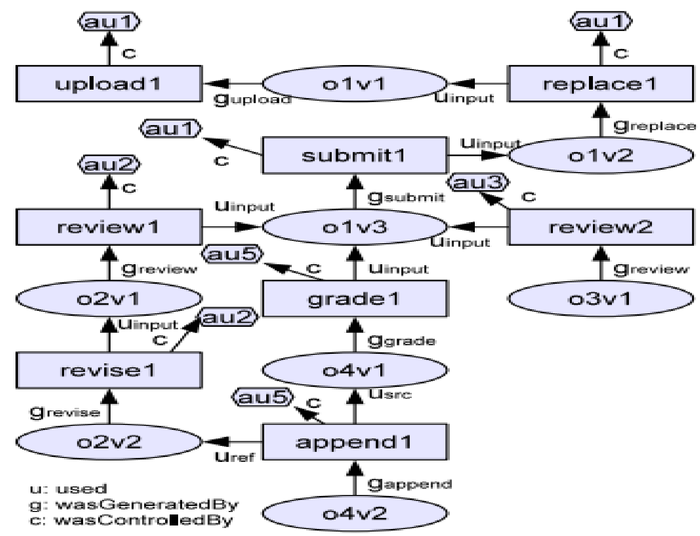




A sample social graph

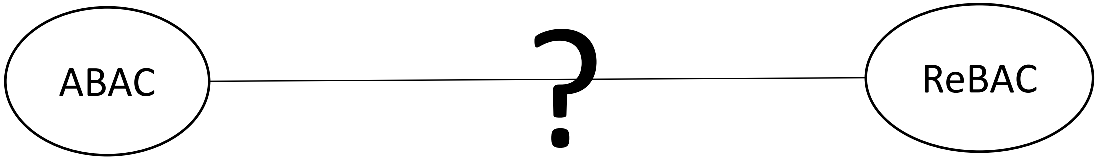


Access control for IOT



A sample Provenance Graph (Park et al. 2012)

Problem Statement



- Are they Comparable ? Can Attributes Express Relationships?
- Can ReBAC Configure ABAC? Vice versa?
- Do they have equal expressive power? If not which one is more expressive?

ABAC vs. ReBAC : There is a fundamental lack of understanding regarding the relationship between ABAC and ReBAC.

What are the novel ways other than OSN ReBAC can be seen, extended and applied?

ReBAC Potential: The potential of   ReBAC has recently been recognized and there remain many directions in which ReBAC models can be developed.

Problem Statement(Cont..)

- Which one is a standard ABAC model:
  UCON? $ABAC_{\alpha}$? $ABAC_{\beta}$? NIST ABAC?
- What are the core characteristics of an ABAC model
- What is the safety property and expressive power variance among the existing ABAC models

ABAC vs. ABAC: There is a proliferation of ABAC models without a formal understanding of their safety properties and relative expressive power.

Summary of Contribution

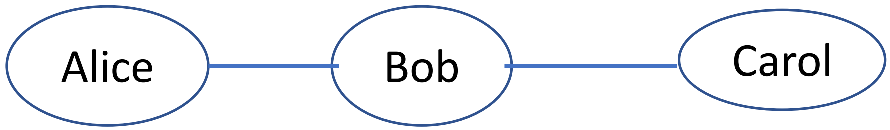
- A Comparison of ReBAC and ABAC.
- A novel ReBAC model definition and its application in the cloud.
- Safety and Expressive Power analysis of $ABAC_{\alpha}$ and its extensions.

- Introduction
- **Comparison of ReBAC and ABAC**
- Object-to-Object Relationship Based Access Control: Model and Multicloud demonstration
- Safety and Expressive Power Comparison of $ABAC_{\alpha}$ and its Enhancements
- Conclusion

Attribute Types

1. Attribute Value Structure
   - ❑ Atomic-valued or Single-valued Attribute (e.g. gender)
   - ❑ Set-valued or Multi-valued Attribute (e.g. phoneNumber)
   - ❑ Structured Attribute (e.g person-Info (name, age, phoneNumber))
2. Attribute Value Scope
   - ❑ Entity Attribute (e.g. friend)
   - ❑ Non-entity Attribute (e.g. age)
3. Boundedness of attribute range
   - ❑ Finite Domain Attribute (e.g. gender)
   - ❑ Infinite Domain Attribute (e.g. time)
4. Attribute association
   - ❑ Contextual or Environmental Attribute (e.g. currentTime)
   - ❑ Meta Attribute (e.g. role(user) = manager , task(manager) = supervise)
5. Attribute mutability
   - ❑ Mutable Attribute
   - ❑ Immutable Attribute

Expressing Multilevel Relationship With Attributes



Attribute Composition

- ❑ Needs one attribute: friend
- ❑ Policy Expression uses Attribute composition

friend(Alice)={Bob}
friend(friend(Alice))={Carol}

Composite Attribute

- ❑ Needs two attribute
  1. friend
  2. friendOfFriend
- ❑ Policy Expression uses direct attributes

friend(Alice) ={Bob}
friendOfFriend(Alice)={Carol}

ReBAC Classification

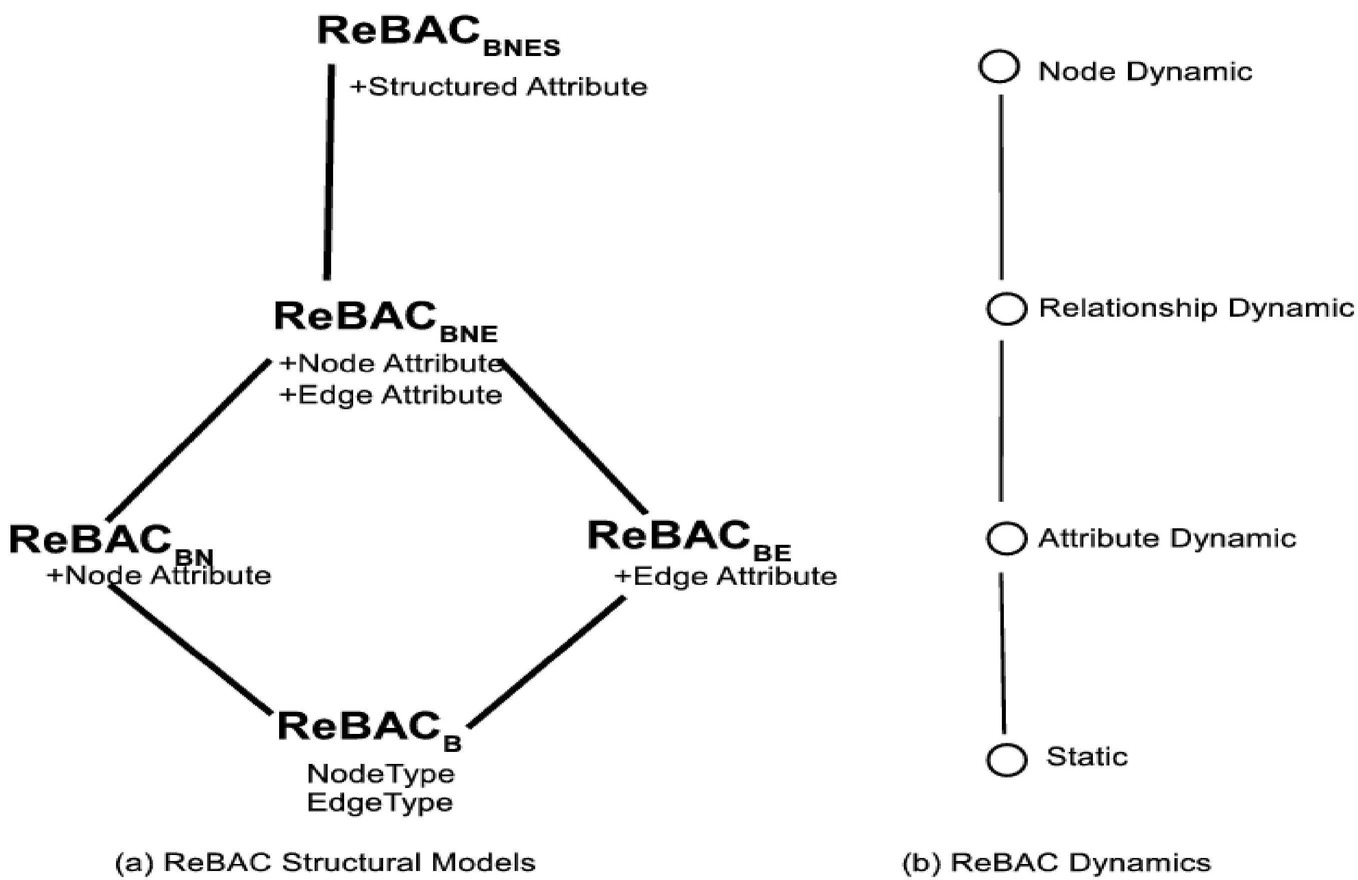


Figure 2: ReBAC Classification

ABAC Classification

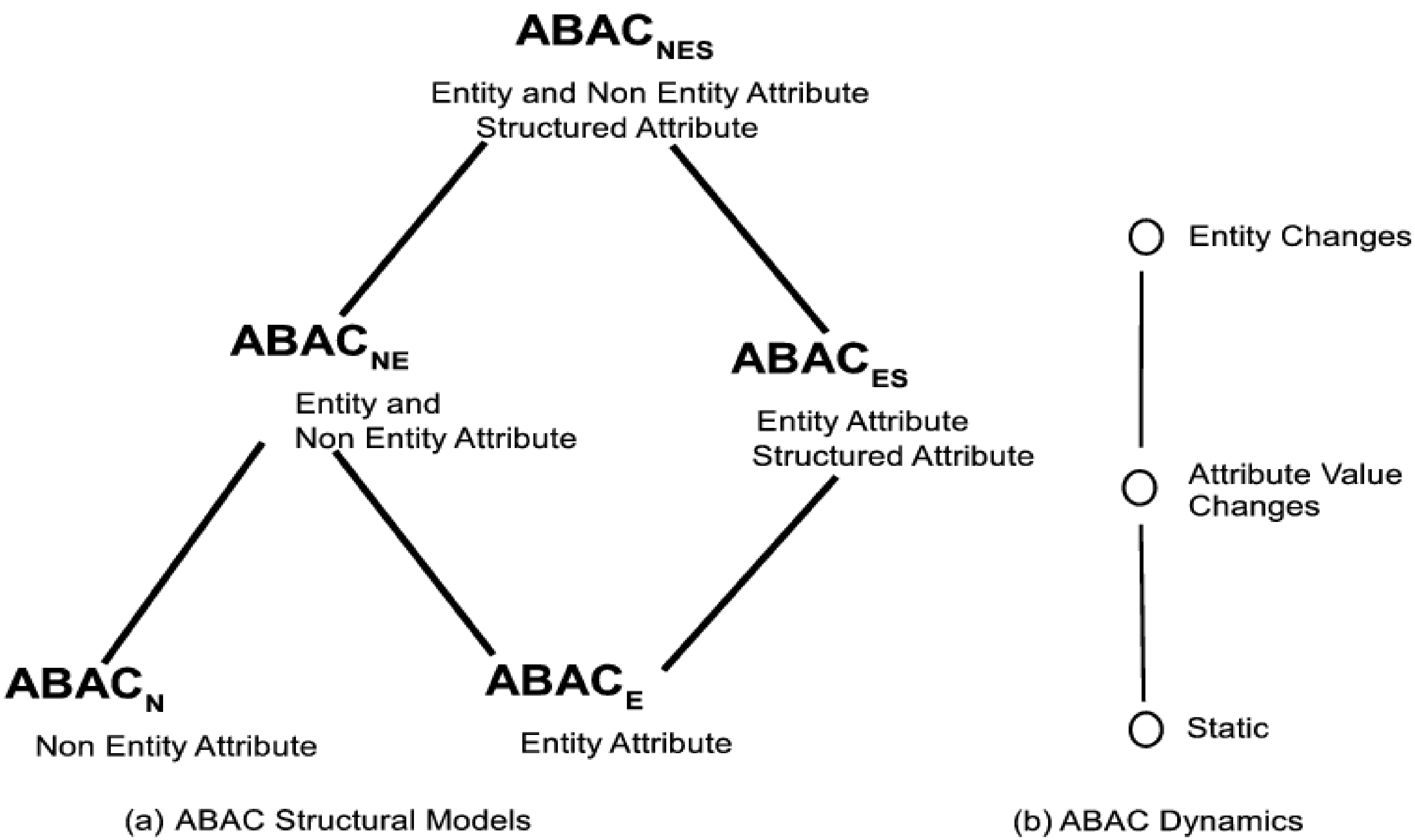


Figure 3: ABAC Framework

Expressing Relationship Graph with Attributes

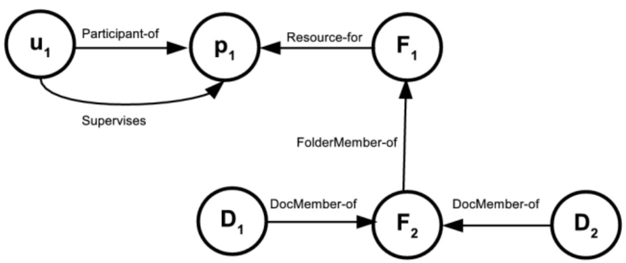


Figure 4: Relationship Graph [Crampton et al 2014] Expressible with $ReBAC_B$ and $ABAC_E$

- Entity types = {user, project, folder , document}
- Attributes:
  - User attributes ={Participant-of, Supervises}
  - Folder attributes = {Resource-for, FolderMember-of}
  - Project attributes = {}
  - Document attributes ={DocMember-of}

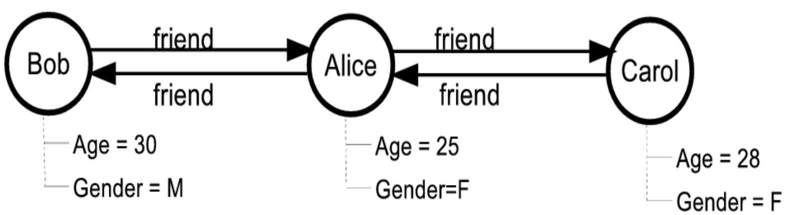


Figure 5:Relationship Graph Expressible with $ReBAC_{BN}$ and $ABAC_E$

- entityType = {user}
- Attribute:
  - User's entity attribute ={friend}
  - User's Non Entity Attribute ={Name, Age, Gender}

Expressing Relationship Graph with Attributes (Continued…)

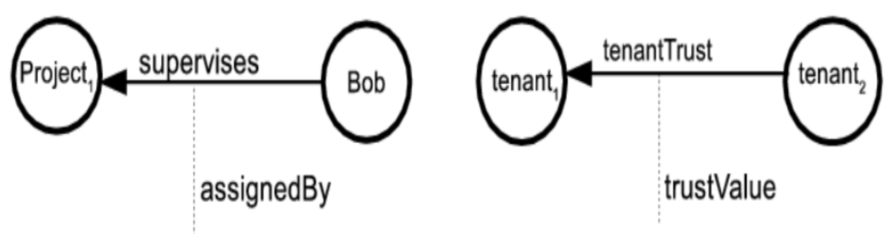


Figure 6:Relationship Graph Expressible with ReBAC$_{BE}$ and ABAC$_{ES}$

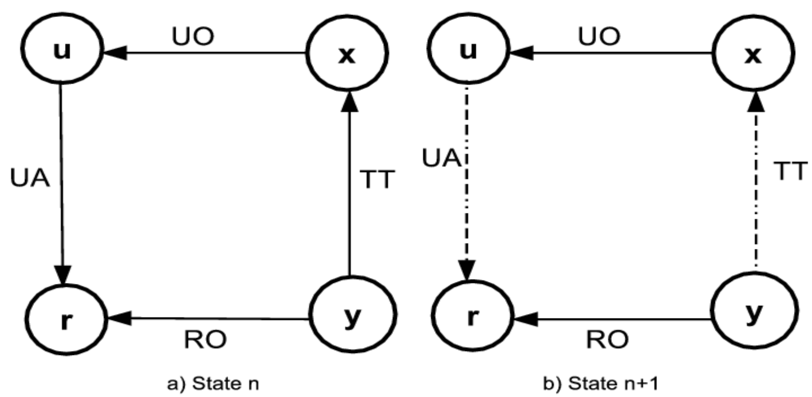


Figure 7: Relationship Graph [cheng et al 2016] Expressible with ReBAC$_{BNES}$ and ABAC$_{ES}$

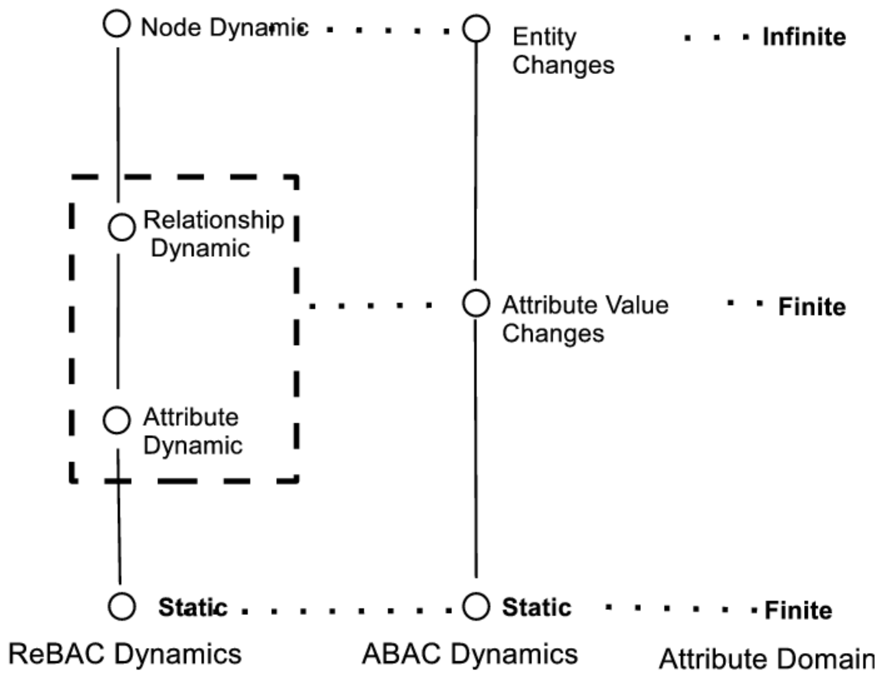
- entityType = {user, project, tenant}
- Attribute:
  - ❑ user's atomic entity attribute ={supervises}
  - ❑ User's structured entity Attribute ={assignedBy}

  e.g. assignedBy(Bob) =("Project1", "supervises","Alice")

- Entity types: {user, tenant, role}
- Attribute:
  - ❑ User's atomic entity attribute: {UO,UA}
  - ❑ Users Structured Entity Attribute: {dependentEdge}

  dependentEdge(u) = ("r","UA", {(y,x,TT)})

Comparison: On Dynamics



$$ABAC_X \equiv ReBAC_Y \; Means$$

- Static and finite attribute domain $ABAC_X \equiv Static\ ReBAC_Y$
- $ABAC_X$ *Attribute value changes with finite domain* $\equiv Relationship\ Dynamic\ ReBAC_Y$
- $ABAC_X$ *with entity changes and infinite domin entity attribute* $\equiv node\ dynamic\ ReBAC_Y$

Figure 8: ReBAC Dynamics, ABAC Dynamics and Attribute Domain wise Comparison between ReBAC and ABAC

Comparison: Equivalent Structural Models for ReBAC and ABAC

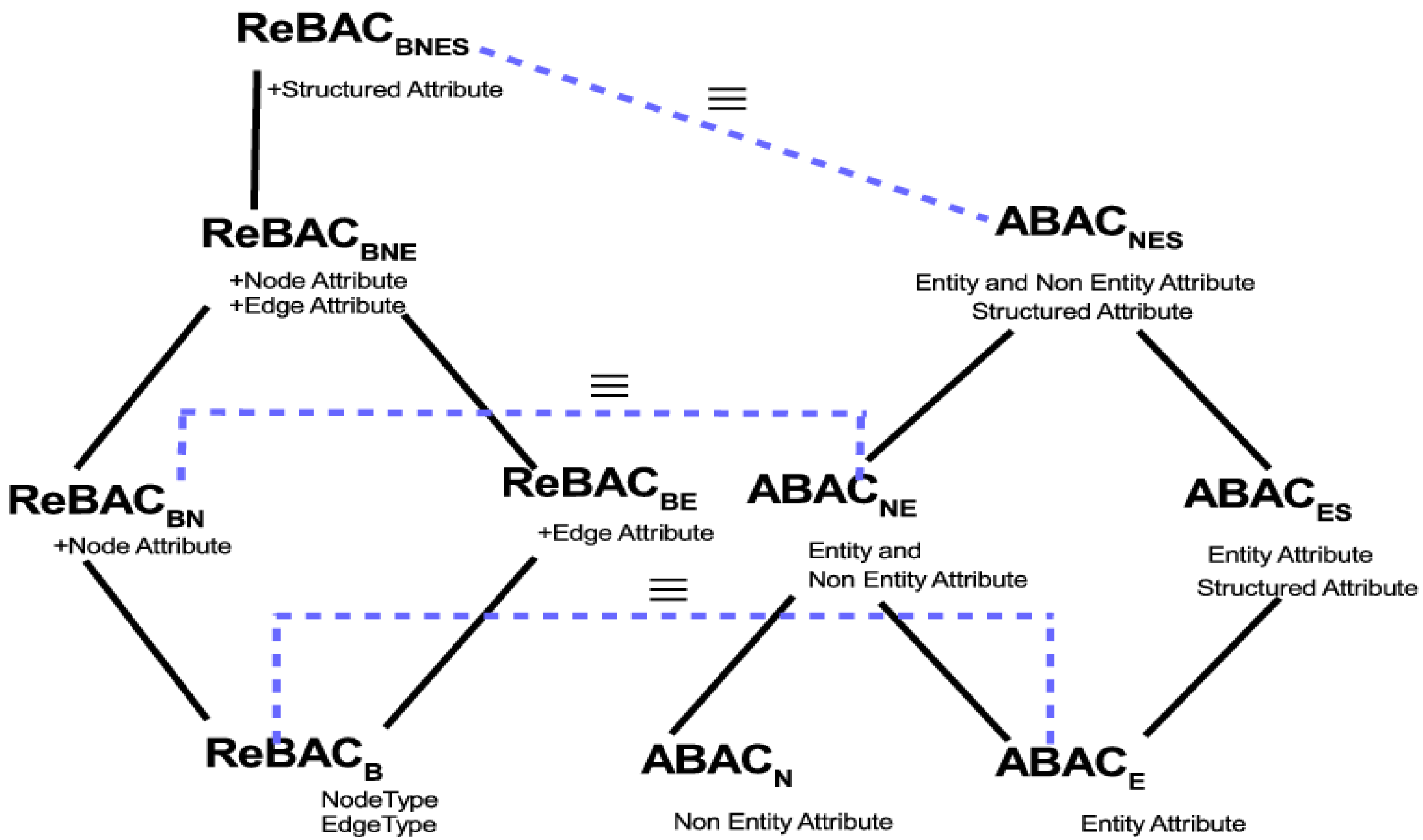


Figure 9: Equivalence of ReBAC and ABAC Structural Classification

Comparison: Non-Equivalent Structural models for ReBAC and ABAC

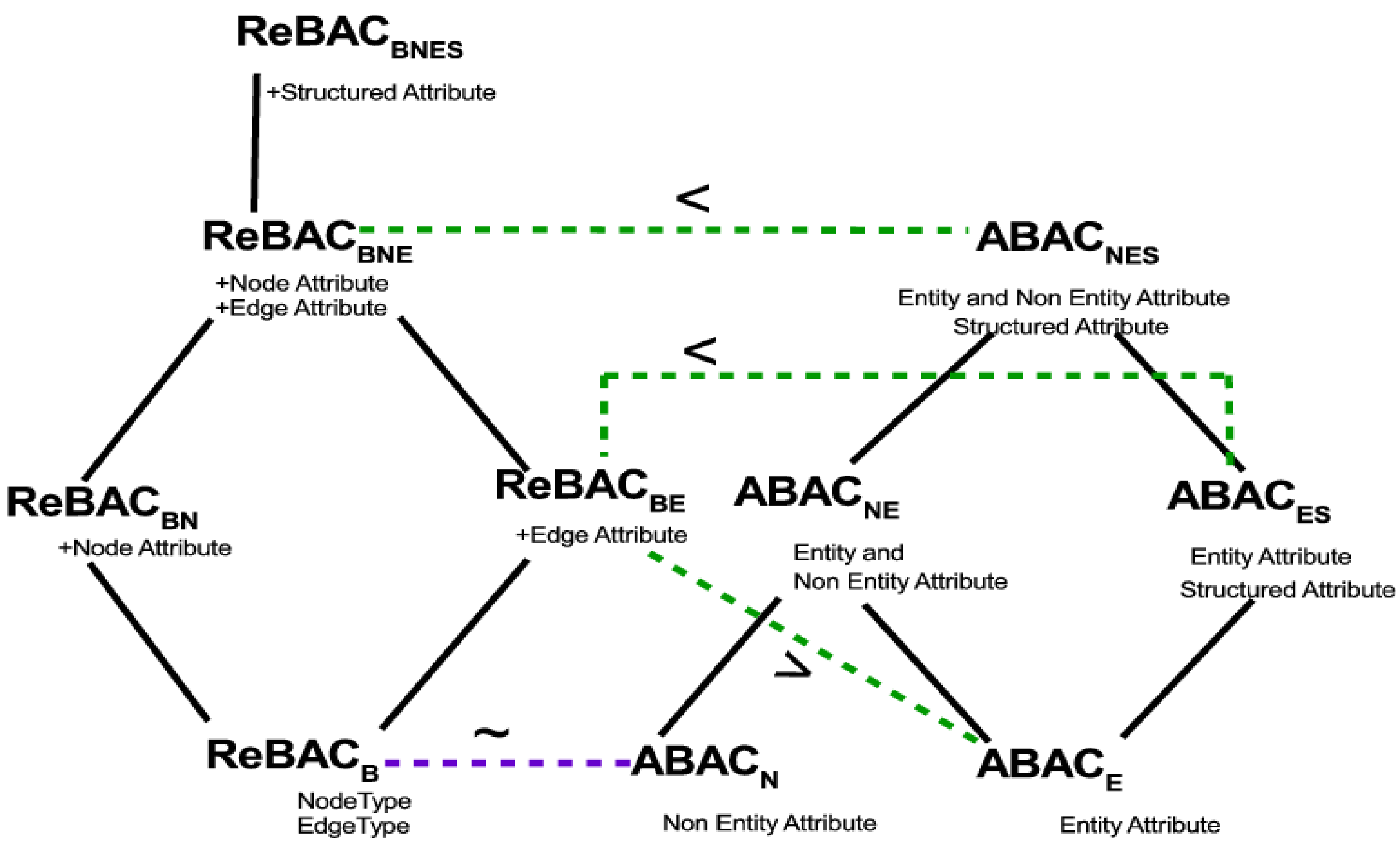


Figure 10: Non-Equivalence of ReBAC and ABAC Structural Classification

Comparison

- Attribute Composition: Polynomial complexity for authorization policy and constant complexity on update
- Composite attribute: Constant complexity on authorization policy and polynomial complexity on update to maintain relationship changes.
- Performance Depends on :
  - ❑ Node Dynamics
  - ❑ Relationship Dynamics
  - ❑ Density of the Relationship Graph

Choice of Models:

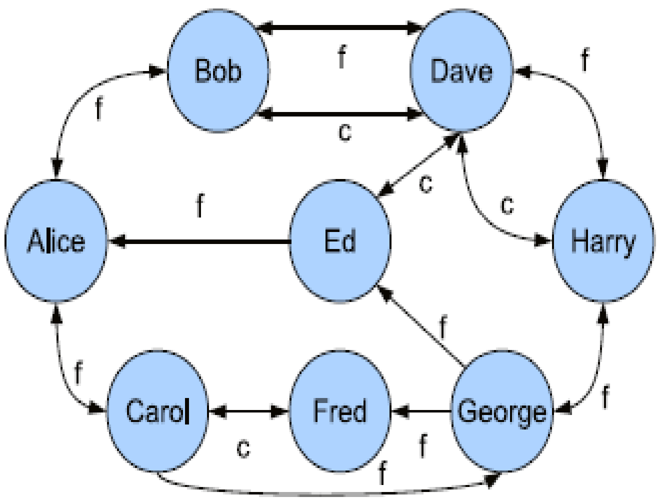
- For static system or only non entity attribute change------Composite attribute is the best approach
- System with huge node dynamics, relationship dynamics and high relationship density----- Attribute composition is the best option
- If the system is in the middle between two extremes ---- A hybrid approach where both composite attribute and attribute composition is used.
- Hybrid Approach:

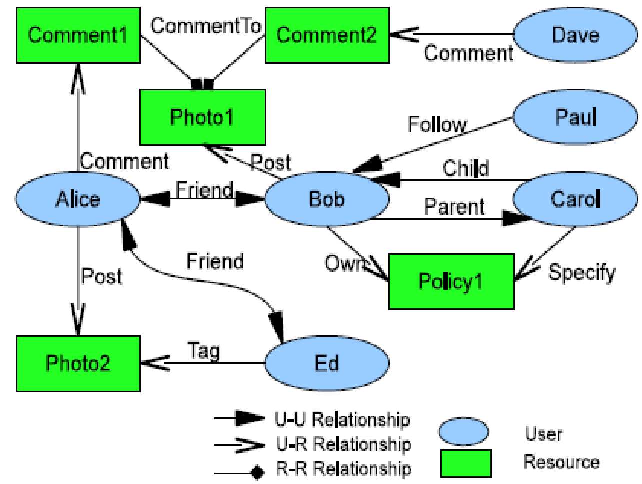
To achieve p level relationship composition it uses m level composite attribute and n level attribute composition where p = n X m.

Outline

- Introduction
- Comparison of ReBAC and ABAC
- Object-to-Object Relationship Based Access Control: Model and Multicloud demonstration
- Safety and Expressive Power Comparison of $\boldsymbol{ABAC_{\alpha}}$ and its Enhancements
- Conclusion

Relationships in OSN





User to user relationships in a sample social graph [UURAC, Cheng et al. 2012]

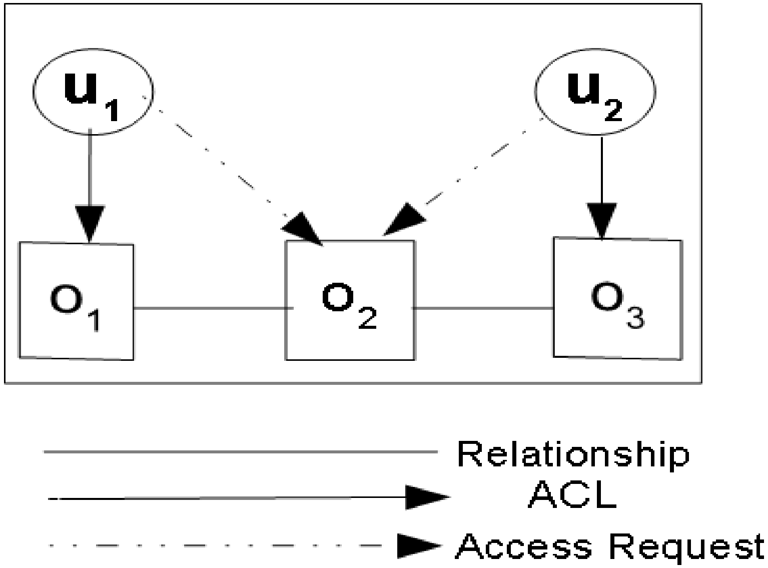
User to user, user to resource and resource to resource relationships in a sample social graph [URRAC, Cheng et al. 2012]

Limitations:

Cannot configure relationship between objects independent of user.
Cannot express authorization policy solely considering object relationship.

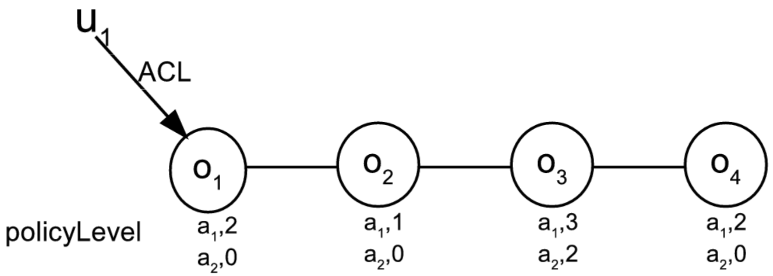
How the model would look like?

An Object to Object Relationship Based Access Control



$ACL(o_1) = \{u_1\}$
$ACL(o_2) = \{\}$
$ACL(o_3) = \{u_2\}$

Policy Level Example



$policyLevel(a_1, o_1) = 2$
$policyLevel(a_2, o_1) = 0$
$policyLevel(a_1, o_2) = 1$
$policyLevel(a_2, o_2) = 0$
$policyLevel(a_1, o_3) = 3$
$policyLevel(a_2, o_3) = 2$
$policyLevel(a_1, o_4) = 2$
$policyLevel(a_2, o_4) = 0$

OOReBAC: Model Components and Definition

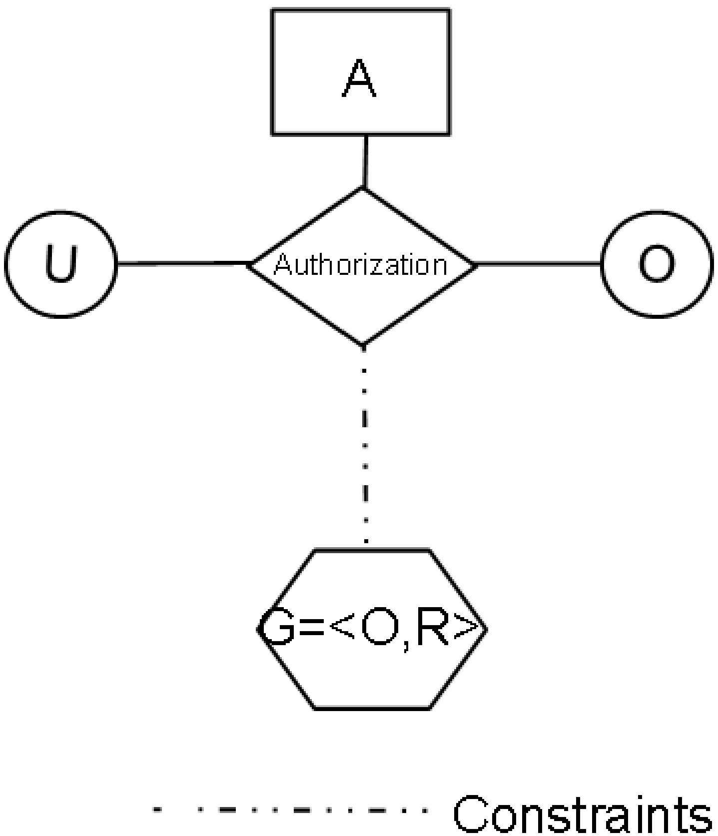
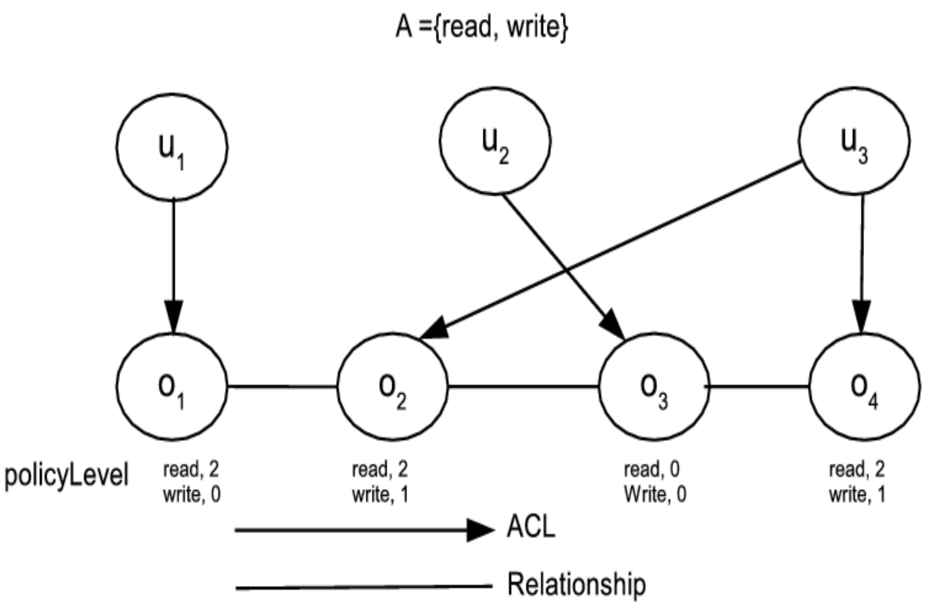


Figure 10: OOReBAC Model Components

- U is a set of users
- O is a set of objects
- $R \subseteq \{z \mid z \subset O \wedge \mid z \mid = 2\}$
- $G = \langle O, R \rangle$ is an undirected relationship graph with vertices O and edges R
- A is a set of actions
- $P^i(o_1) = \{ o_2 \mid$ there exists a simple path of length p in graph G from o_1 to $o_2\}$
- policyLevel: $O \times A \rightarrow \mathbb{N}$
- ACL: $O \rightarrow 2^U$ which returns the Access control List of a particular object.
- There is a single policy configuration point. Authorization Policy. for each action $a \in A$, $Authz_a(u{:}U,o{:}O)$ is a boolean function which returns true or false and u and o are formal parameters.
- Authorization Policy Language:
  Each action "a" has a single authorization policy $Authz_a(u{:}U,o{:}O)$ specified using the following language.
  $\phi := u \in PATH_i$
  $PATH_i := ACL(P^0(o)) \cup \ldots \cup ACL(P^i(o))$ where $i = \min(\mid O \mid - 1, policyLevel(a,o))$
  where for any set X, $ACL(X) = \bigcup_{x \in X} ACL(x)$

OOReBAC: An Example

Sequence of operations and its outcome:



- $U = \{u_1, u_2, u_3\}$
- $O = \{ o_1, o_2, o_3, o_4\}$
- $R = \{\{o_1, o_2\}, \{o_2, o_3\}, \{o_3, o_4\}\}$
- $ACL(o_1) = \{u_1\}$
  $ACL(o_2) = \{u_3\}$
  $ACL(o_3) = \{u_2\}$
  $ACL(o_4) = \{u_3\}$
- $policyLevel(read, o_1) = 2$
  $policyLevel(write, o_1) = 0$
  $policyLevel(read, o_2) = 2$
  $policyLevel(write, o_2) = 1$
  $policyLevel(read, o_3) = 0$
  $policyLevel(write, o_3) = 0$
  $policyLevel(read, o_4) = 2$
  $policyLevel(write, o_4) = 1$

Configuration:

- $A = \{read, write\}$
- $Authz_{read}(u:U, o:O) \equiv u \in P^{policyLevel(read,o)}$
- $Authz_{write}(u:U, o:O) \equiv u \in P^{policyLevel(write,o)}$

Sequence of operations and its outcome:

- $read(u_1, o_3)$, $write(u_1, o_3)$ are denied
- $read(u_2, o_1)$ is allowed, $write(u_2, o_1)$ is denied
- $read(u_1, o_4$, $write(u_1, o_4)$ are denied

OOReBAC:Application

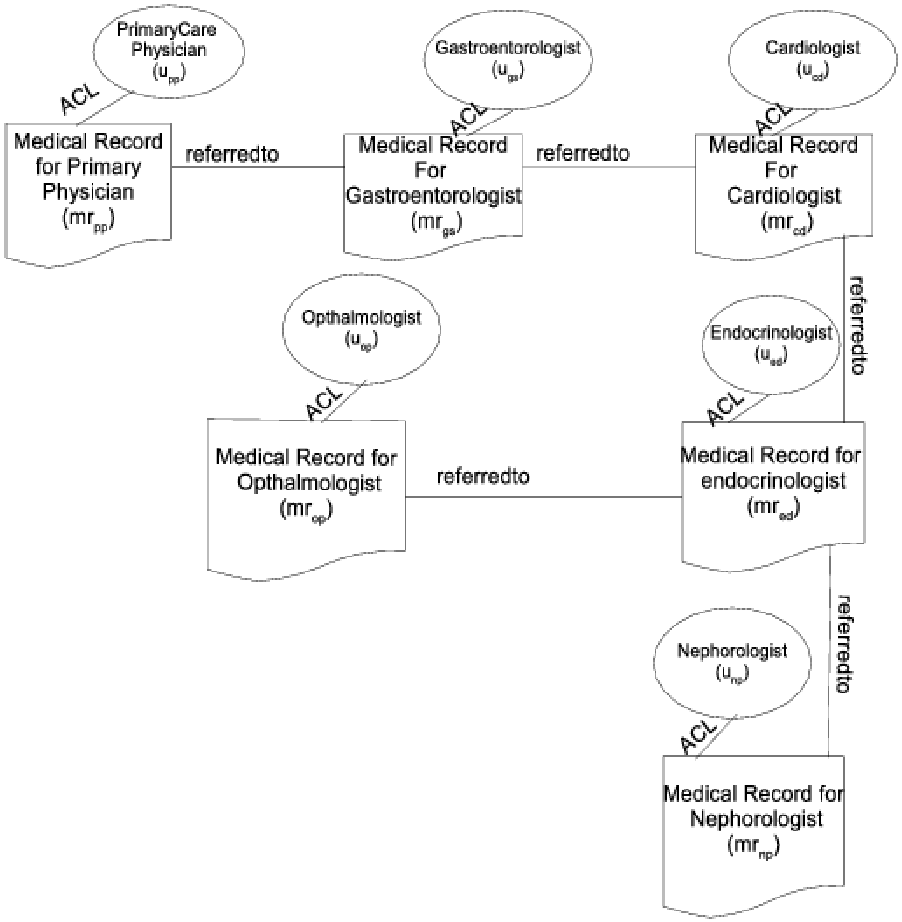


Figure 11: An Example of OOReBAC Application in Medical

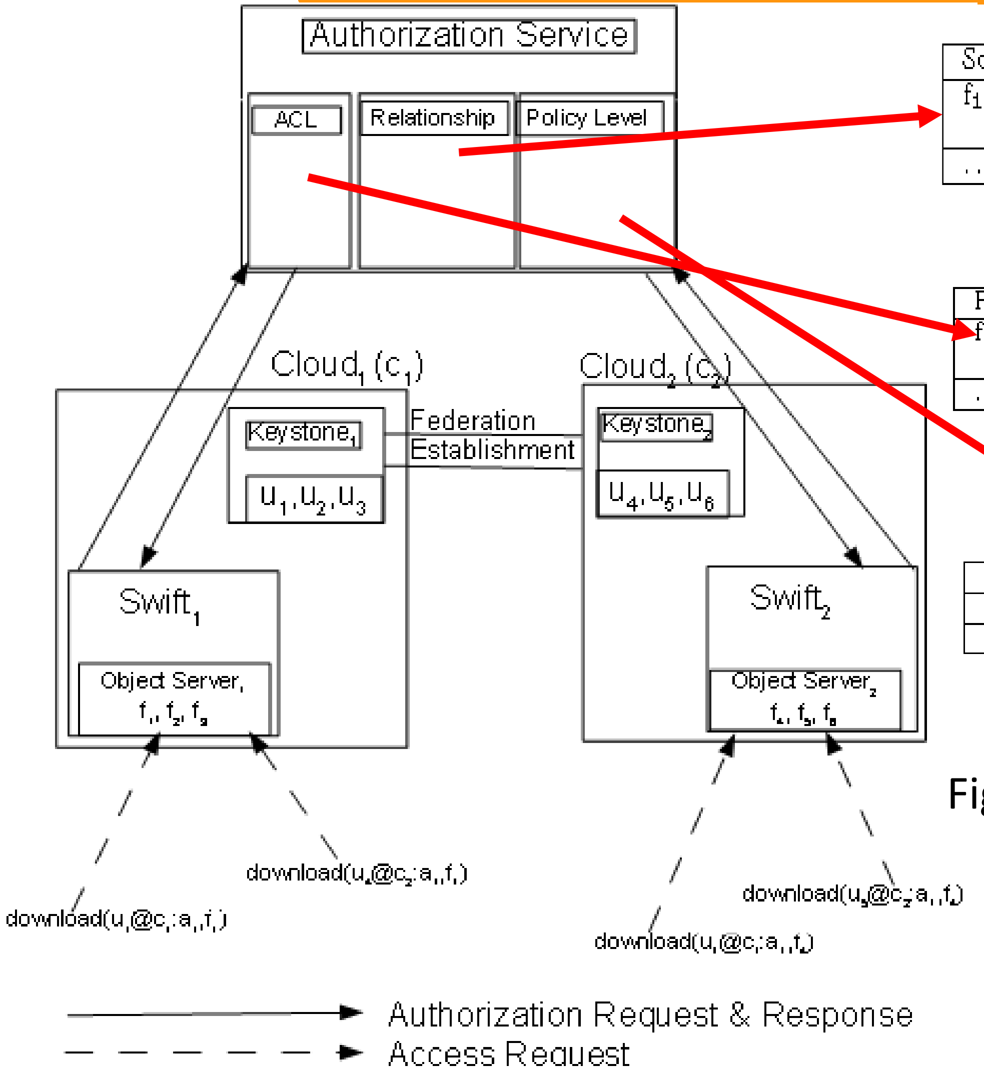
An OOReBAC Instantiation

- $U = \{ u_{pp}, u_{gs}, u_{cd}, u_{op}, u_{ed}, u_{np} \}$
- $O = \{ mr_{pp}, mr_{gs}, mr_{cd}, mr_{op}, mr_{ed}, mr_{np} \}$
- $R = \{\{mr_{pp}, mr_{gs}\}, \{mr_{gs}, mr_{cd}\}, \{mr_{cd}, mr_{ed}\}, \{mr_{op}$ $mr_{ed}\},\{mr_{np}, mr_{ed}\}\}$
- $ACL(mr_{pp}) = \{u_{pp}\}$,
  $ACL(mr_{gs}) = \{u_{gs}\}$,
  $ACL(mr_{cd}) = \{u_{cd}\}$,
  $ACL(mr_{op}) = \{u_{op}\}$,
  $ACL(mr_{ed}) = \{u_{ed}\}$,
  $ACL(mr_{np}) = \{u_{np}\}$
- Action $=\{$read, write$\}$
- policyLevel(read,mr_{pp})=∞, policyLevel(write,mr_{pp})=0,
  policyLevel(read,mr_{gs})=∞, policyLevel(write,mr_{gs})=0,
  policyLevel(read,mr_{cd})=∞, policyLevel(write,mr_{cd})=0,
  policyLevel(read,mr_{op})=∞, policyLevel(write,mr_{op})=0,
  policyLevel(read,mr_{ed})=∞, policyLevel(write,mr_{ed})=0,
  policyLevel(read,mr_{np})=∞, policyLevel(write,mr_{np})=0
- Authorization policy:
  $Authz_{read}(u,o) \equiv u \in P^{policyLevel(read,o)}$
  $Authz_{write}(u,o) \equiv u \in P^{policyLevel(write,o)}$

Sequence of Operations and Outcomes

1) $read(u_{np}, mr_{pp})$: authorized
2) $read(u_{cd}, mr_{np})$: authorized
3) $write(u_{np}, mr_{np})$: authorized
4) $write(u_{np}, mr_{pp})$: denied
5) $write(u_{np}, mr_{pp})$: denied

Implementation: Openstack Object Storage (Swift)



Relationship

SourceFileName	TargetFileList
f_1@$cloud_1$:$account_1$:$container_1$	{f_2@$cloud_1$:$account_1$:$container_1$, f_3@$cloud_2$:$account_1$:$container_1$}
...	...

ACL

Filename	UserList
f_1@$cloud_1$:$account_1$:$container_1$	{u_1@$cloud_1$:$account_1$, u_2@$cloud_1$:$account_1$}
...	...

PolicyLevel

FileName	Policy Level
f_3@$cloud_1$:$account_2$:$container_3$	(download,2)
...	...

Figure 12: OOReBAC Implementation

Outline

- Introduction
- Comparison of ReBAC and ABAC
- Object-to-Object Relationship Based Access Control: Model and Multicloud demonstration
- Safety and Expressive Power Comparison of $\boldsymbol{ABAC_{\alpha}}$ and its Enhancements
- Conclusion

ABAC$_{\alpha}$

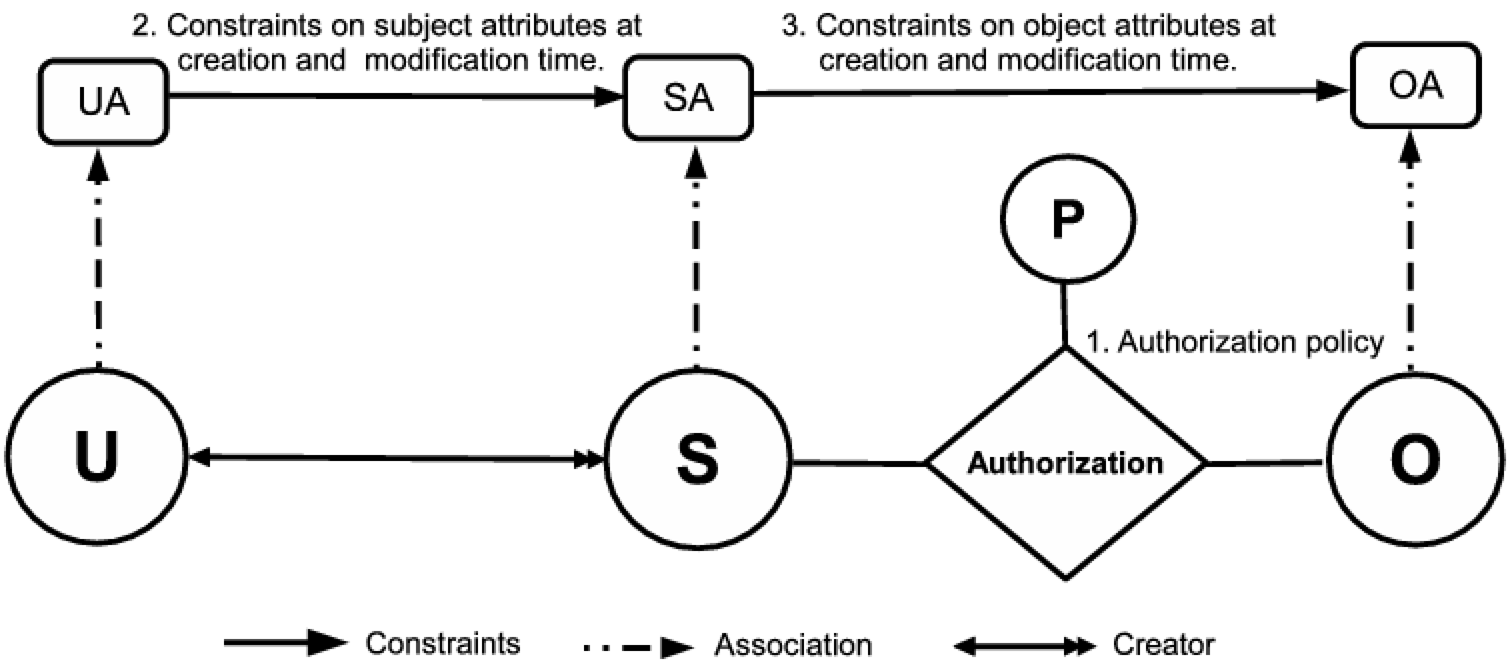


Figure 13: $\boldsymbol{ABAC_{\alpha}}$ Model [Jin et al. 2012]

$UCON_{preA}^{finite}$ Model

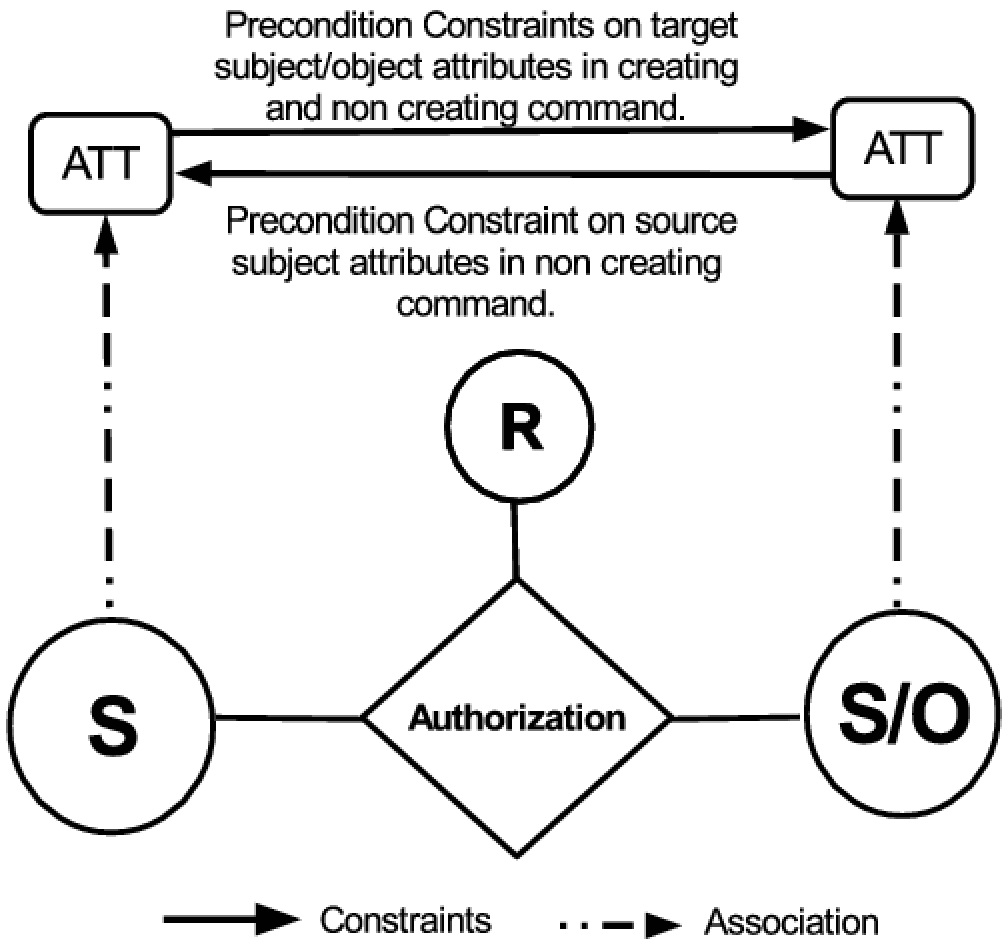


Figure 14: $UCON_{preA}^{finite}$ Model

ABAC_α vs. $\mathrm{UCON}_{\mathrm{preA}}^{\mathrm{finite}}$

	$ABAC_\alpha$	$UCON_{preA}^{finite}$
Attribute Value Structure	Atomic and set valued	Atomic valued
Attribute Value Scope	finite entity + Non-entity	Non-entity
Boundedness of Attr. Range	finite	finite
Attribute Association	No context / meta attribute	No context/meta attribute
Attribute Mutability	Immutable	Mutable
Entities	User, subject , object	object
Operations	Configurable Condition + Mandatory update	Command specific precondition + tightly coupled optional update
Precondition	Configurable Boolean Expression	Command specific Boolean function
Update value	Direct value from range	Command specific computed value

Central Result

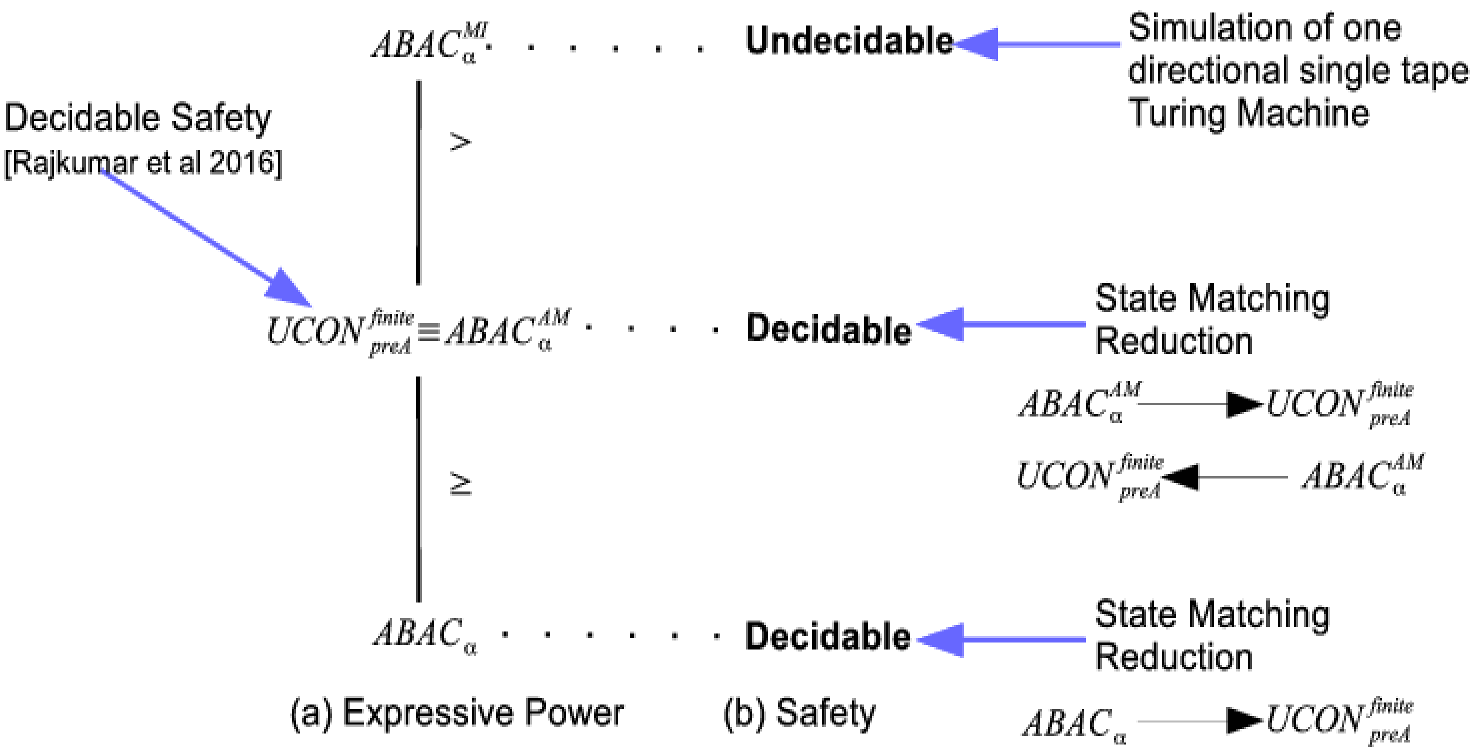


Figure 15: Central Result

$ABAC_{\alpha}^{AM}$ and $ABAC_{\alpha}^{MI}$

In addition to all the features of ABAC_{α} , $\mathrm{ABAC}_{\alpha}^{\mathrm{AM}}$ has the following properties:

1. Subject can create, delete or modify another subject and at the same time can modify its own attribute value
2. Subject can modify itself.
3. Subject modification by user can modify user's own attribute value

In addition to all the features of $\mathrm{ABAC}_{\alpha}^{\mathrm{AM}}$, $\mathrm{ABAC}_{\alpha}^{\mathrm{MI}}$ has the following properties:

- Infinite domain entity attribute.

Outline

- Introduction
- Comparison of ReBAC and ABAC
- Object-to-Object Relationship Based Access Control: Model and Multicloud demonstration
- Safety and Expressive Power Comparison of $\boldsymbol{ABAC_{\alpha}}$ and its Enhancements
- Conclusion

Conclusion: Summary of Contribution

- The most general form ABAC and ReBAC are equivalent. The relationship between less general ABAC and ReBAC is subtle and variable depending on the precise flavor of these two access control approaches in any given model.

- OOReBAC is the first attempt towards using object relationship independent of user in authorization policy specification. Its application is possible for multicloud resource sharing in Openstack object storage Swift.

- Safety and Expressive power of an ABAC model depend onto the detail of that model.

Conclusion: Future Work

This work can be expanded in many directions:

- Formal definition of specific ReBAC and its structural equivalent ABAC model would bring more realistic result for theoretical equivalence.
- To better understand the relative advantages and disadvantages of ReBAC and ABAC we can consider metrics beyond theoretical equivalence such as performance, maintainability, robustness, and agility.
- OOReBAC model can be extended to accommodate multiple type asymmetric relationships to configure version control and object oriented system.
- Application of relationship based authorization policy in various fields such as IoT.

Dissertation Publications

Conference Papers(Published):

1. Tahmina Ahmed, Farhan Patwa and Ravi Sandhu, "Object-to-Object Relationship-Based Access Control: Model and Multi-Cloud Demonstration". In Proceedings of the 17th IEEE Conference on Information Reuse and Integration (IRI), Pittsburgh, Pennsylvania, July 28-30, 2016, 8 pages.
2. Tahmina Ahmed, Ravi Sandhu and Jaehong Park, "Classifying and Comparing Attribute – Based and Relationship-Based Access Control".In Proceedings of the 7th ACM Conference on Data and Application Security and Privacy (CODASPY), March 22-24, 2017, Scottsdale, Arizona, 12 pages..
3. Tahmina Ahmed and Ravi Sandhu, " Safety of $ABAC_{\alpha}$ is Decidable". In Proceedings of the 11th International Conference on Network and System Security (NSS), Helsinki, Finland, August 21-23, 2017, 15 pages.

Journal Papers (Work in Progress):

1. Tahmina Ahmed and Ravi Sandhu, "The $\mathrm{ABAC}_{\alpha}^{\mathrm{AM}}$ Model: An Enhancement of $ABAC_{\alpha}$ Equivalent to $\boldsymbol{UCON_{preA}^{finite}}$"
2. Tahmina Ahmed, Ravi Sandhu and Jaehong Park, "On the Formal Relationship Between ReBAC and ABAC"

Questions/Comments