Multi-Tenancy Authorization Models for Collaborative Cloud Services

Bo Tang, Ravi Sandhu, and Qi Li

Presented by Bo Tang

OUTLINE

- Introduction
- Background & Motivation
- Formalized Models
  - MTAS
  - AMTAS
  - Enhanced Trust Models
- Policy Specification
- Conclusion and Future Work

OUTLINE

- Introduction
- Background & Motivation
- Formalized Models
  - MTAS
  - AMTAS
  - Enhanced Trust Models
- Policy Specification
- Conclusion and Future Work

Cloud Computing

- Shared infrastructure
  - [$$$] -----> [$|$|$]
- Multi-Tenancy
  - Virtually dedicated resources
- Drawbacks:
  - Data Locked-in
    - Collaborations can only be achieved through desktop.
    - E.g.: open Dropbox files with GoogleDoc.
  - How to collaborate?

Source: http://blog.box.com/2011/06/box-and-google-docs-accelerating-the-cloud-workforce/

Collaborative Access Control

- Centralized Facility
  - Chance for centralized models in distributed systems
- Agility
  - Collaboration and collaborators are temporary
- Homogeneity
  - Handful of popular brands
- Out-Sourcing Trust
  - Built-in collaboration spirit

Industry Solutions

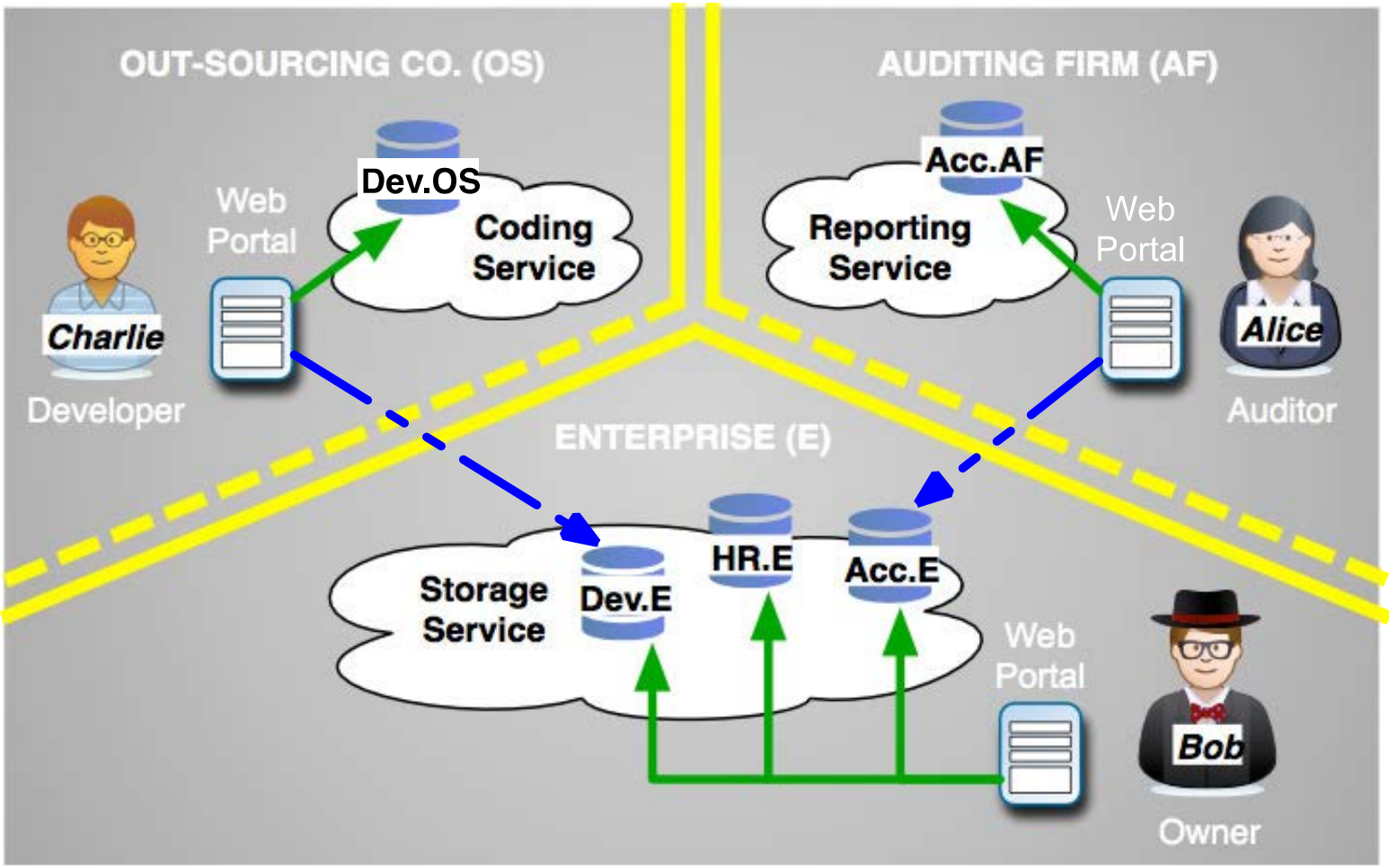
- Microsoft and IBM: Fine-grained data sharing in SaaS using DB schema
  - Only feasible in DB
- NASA: RBAC + OpenStack
  - Lacks ability to support collaborations
- Salesforce (Force.com): SSO + SAML
  - Focus on authentication
  - Heavy management of certificates

Source: http://msdn.microsoft.com/en-us/library/aa479086.aspx
http://nebula.nasa.gov/blog/2010/06/03/nebulas-implementation-role-based-access-control-rbac/
http://wiki.developerforce.com/page/Single_Sign-On_with_SAML_on_Force.com

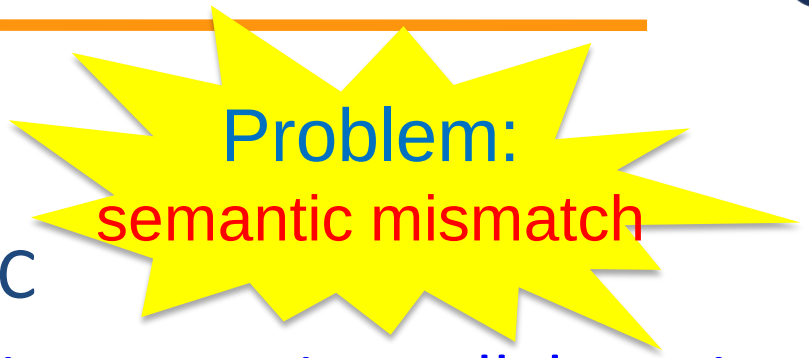
OUTLINE

- Introduction
- Background & Motivation
- Formalized Models
  - MTAS
  - AMTAS
  - Enhanced Trust Models
- Policy Specification
- Conclusion and Future Work

Example


OUT-SOURCING CO. (OS)
AUDITING FIRM (AF)
Dev.OS
Coding Service
Acc.AF
Reporting Service
Web Portal
Web Portal
Charlie
Developer
Alice
Auditor
ENTERPRISE (E)
Storage Service
Dev.E
HR.E
Acc.E
Web Portal
Bob
Owner
Intra-Issuer Access
Cross-Issuer Access

Literature



- RBAC
  - CBAC, GB-RBAC, ROBAC
  - Require central authority managing collaborations
- Delegation Models
  - dRBAC and PBDM
  - Lacks agility (which the cloud requires)
- Grids
  - CAS, VOMS, PERMIS
  - Absence of centralized facility and homogeneous architecture (which the cloud has)

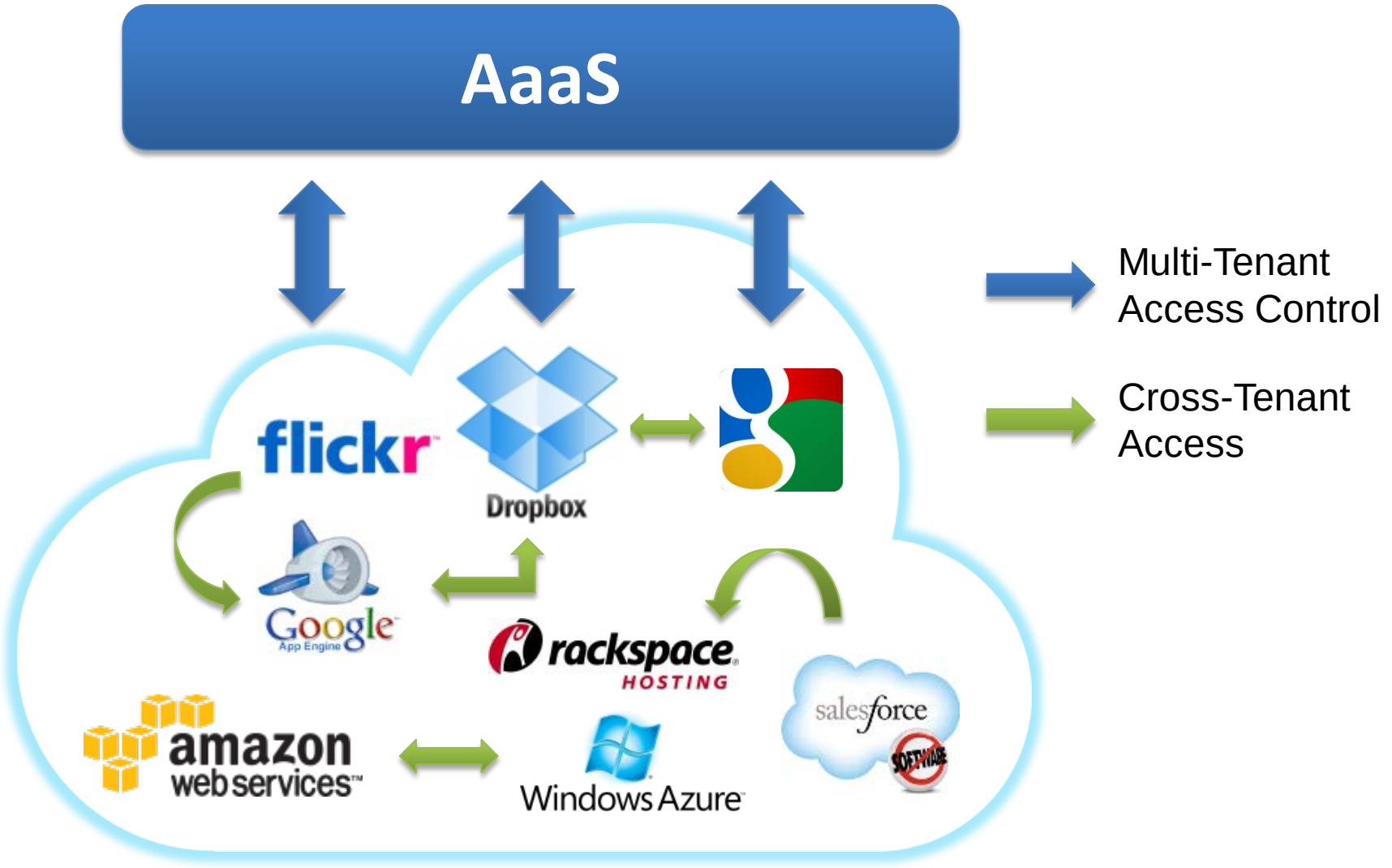
Literature (Contd.)

- Role-based Trust
  - RT, Traust, RMTN AND RAMARS_TM
  - Calero et al: towards a multi-tenant authorization system for cloud services
    - Implementation level PoC
    - Open for extensions in trust models
  - Suits the cloud (out-sourcing trust)

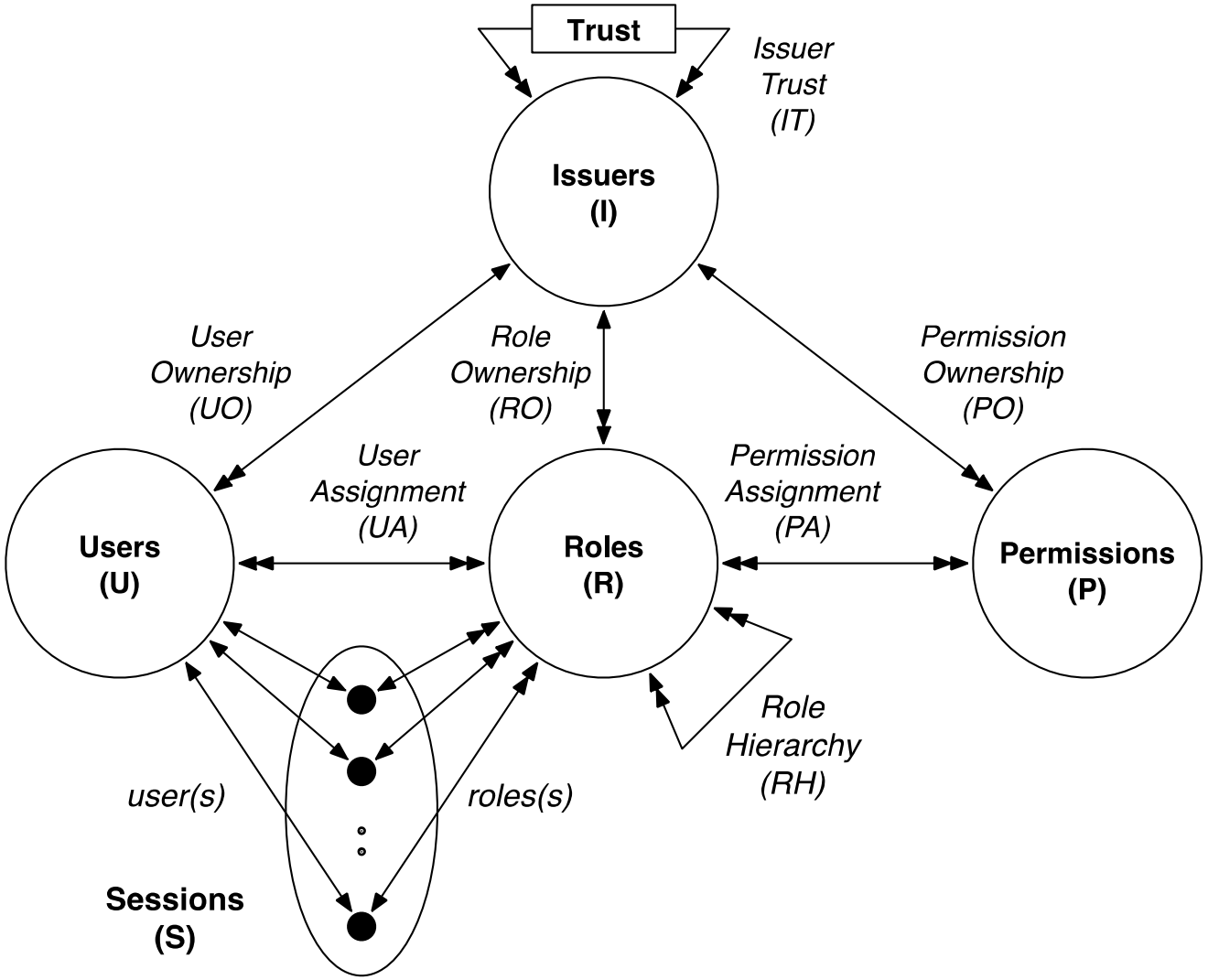
OUTLINE

- Introduction
- Background & Motivation
- Formalized Models
  - MTAS
  - AMTAS
  - Enhanced Trust Models
- Policy Specification
- Conclusion and Future Work

Authorization as a Service (AaaS)

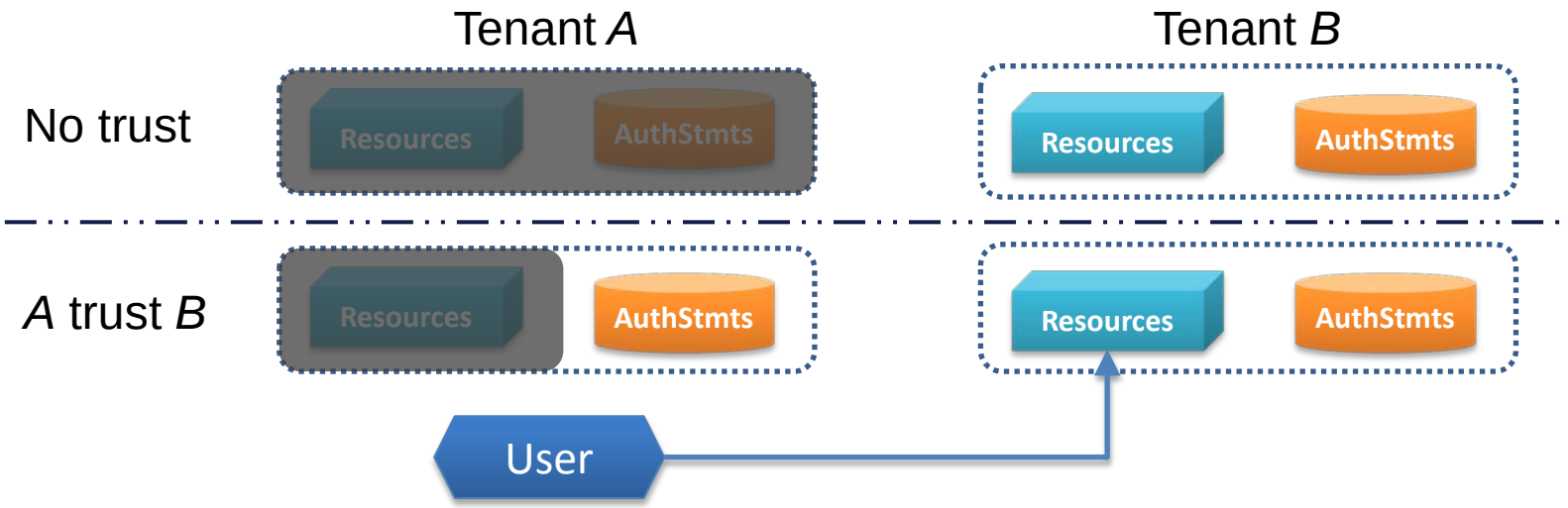


MTAS


Trust
Issuer
Trust
(IT)
Issuers
(I)
User
Ownership
(UO)
Role
Ownership
(RO)
Permission
Ownership
(PO)
User
Assignment
(UA)
Permission
Assignment
(PA)
Users
(U)
Roles
(R)
Permissions
(P)
Role
Hierarchy
(RH)
user(s)
roles(s)
Sessions
(S)

MTAS Trust Model

- If A trusts B then B (resource owner) can assign
  - B's permissions to A's roles; and
  - B's roles as junior roles to A's roles.



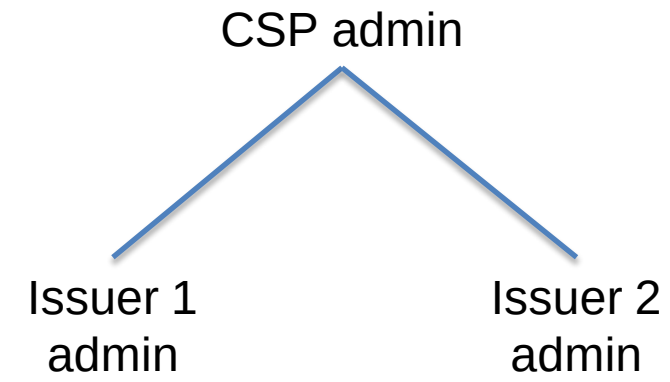


TABLE I
ADMINISTRATION FUNCTIONS OF AMTAS FOR ISSUER i

Function	Condition	Update
$assignUser(i,r,u)$	$i = roleOwner(r) \wedge u \in U$	$UA' = UA \cup \{u \rightarrow r\}$
$revokeUser(i,r,u)$	$i = roleOwner(r) \wedge u \in U \wedge u \rightarrow r \in UA$	$UA' = UA \setminus \{u \rightarrow r\}$
$assignPerm(i,r,p)$	$i = permOwner(p) \wedge i \in canUse(r)$	$PA' = PA \cup \{p \rightarrow r\}$
$revokePerm(i,r,p)$	$i = permOwner(p) \wedge i \in canUse(r) \wedge p \rightarrow r \in PA$	$PA' = PA \setminus \{p \rightarrow r\}$
$assignRH(i,r_1,r)$	$i = roleOwner(r) \wedge i \in canUse(r_1) \wedge \neg(r_1 \gg r) \wedge \neg(r \geq r_1)$ [a]	$\geq' = \geq \cup \{r_2, r_3 : R \mid r_2 \geq r_1 \wedge r \geq r_3 \wedge roleOwner(r_3) \in canUse(r_2) \bullet r_2 \rightarrow r_3\}$
$revokeRH(i,r_1,r)$	$i = roleOwner(r) \wedge i \in canUse(r_1) \wedge r_1 \gg r$ [b]	$\geq' = (\gg \setminus \{r_1 \rightarrow r\})^*$ [c]
$assignTrust(i,i_1)$	$i_1 \in I$	$\lesssim' = \lesssim \cup \{i \rightarrow i_1\}$
$revokeTrust(i,i_1)$	$i_1 \in I \wedge i \lesssim i_1 \wedge i \neq i_1$	$\lesssim' = \lesssim \setminus \{i \rightarrow i_1\}$ [d]

a. This condition avoids cycle creation in the role hierarchy.
b. It requires r_1 to be an immediate ascendant of r.
c. Implied relations are preserved after revocation.
d. By revoking the trust relation, the canUse() function of i's roles automatically updates accordingly, same as PA and RH.

Enhanced Trust Models

- Problem of MTAS
  - Over exposure of truster's authorization information
- Truster-Centric Public Role (TCPR)
  - Expose only the truster's public roles
- Relation-Centric Public Role (RCPR)
  - Expose public roles in terms of each trust relation

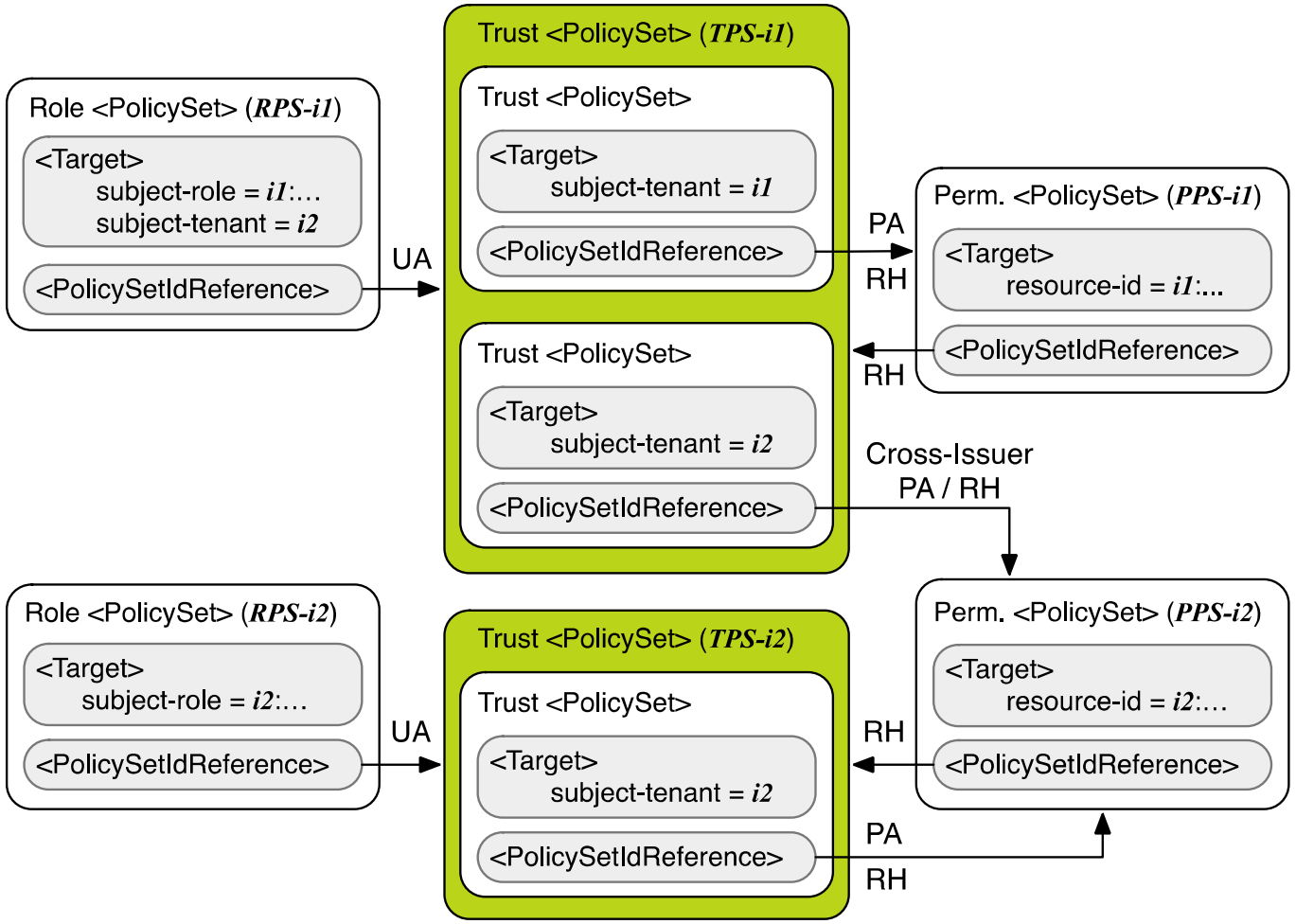
Constraints

- Cyclic Role Hierarchy: lead to implicit role upgrades in the role hierarchy
- SoD: conflict of duties
  - Tenant-level
    - E.g.: SOX compliance companies may not hire same the same company for both consulting and auditing.
  - Role-level
    - across tenants
- Chinese Wall: conflict of interests among tenants

OUTLINE

- Introduction
- Background & Motivation
- Formalized Models
  - MTAS
  - AMTAS
  - Enhanced Trust Models
- Policy Specification
- Conclusion and Future Work

Example


Role <PolicySet> (RPS-i1)
<Target>
subject-role = i1:…
subject-tenant = i2
<PolicySetIdReference>
UA
Trust <PolicySet> (TPS-i1)
Trust <PolicySet>
<Target>
subject-tenant = i1
<PolicySetIdReference>
Trust <PolicySet>
<Target>
subject-tenant = i2
<PolicySetIdReference>
PA
RH
RH
Perm. <PolicySet> (PPS-i1)
<Target>
resource-id = i1:...
<PolicySetIdReference>
Cross-Issuer
PA / RH
Role <PolicySet> (RPS-i2)
<Target>
subject-role = i2:…
<PolicySetIdReference>
UA
Trust <PolicySet> (TPS-i2)
Trust <PolicySet>
<Target>
subject-tenant = i2
<PolicySetIdReference>
RH
PA
RH
Perm. <PolicySet> (PPS-i2)
<Target>
resource-id = i2:…
<PolicySetIdReference>

OUTLINE

- Introduction
- Background & Motivation
- Formalized Models
  - MTAS
  - AMTAS
  - Enhanced Trust Models
- Policy Specification
- Conclusion and Future Work

Conclusion

- Collaboration needs in the cloud eco-system
- Novel service model: AaaS
- Proposed formal models
  - MTAS, AMTAS, Enhanced Trust Models
  - Constraints
- Policy Specification

Future Work

- Accomplished
  - Prototype and evaluation
    - Performance overhead ≈ 0.016 seconds
    - Scalable in the cloud
  - MT-RBAC (delegation-centric trust model)
- On-going Projects
  - OpenStack Keystone extensions
  - Integrate trust into ABAC: MT-ABAC
  - Unified trust framework


Q & A


Thank You!