Secure Information and Resource Sharing in Cloud Infrastructure as a Service

Dissertation Defense

Yun Zhang

Department of Computer Science

Dissertation Committee:

Dr. Ravi Sandhu: Supervising Professor

Dr. Ram Krishnan

Dr. Palden Lama

Dr. Jianwei Niu

Dr. Gregory White

UTSA

I·C·S

The Institute for Cyber Security

Presentation Outline

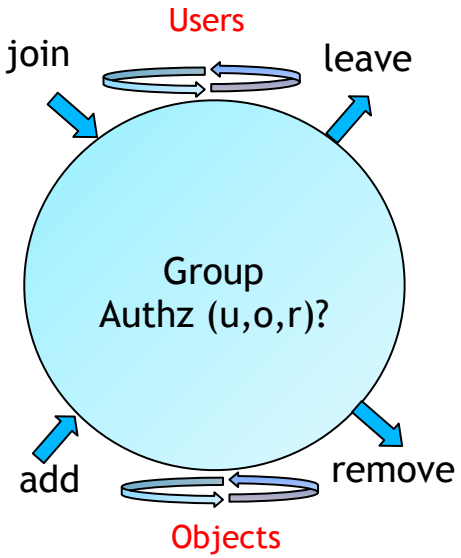
- Introduction
- Background and related work
- Secure Isolated Domain (SID) Model
- SID Model in OpenStack
- SID Model in AWS
- SID Model in Azure
- Conclusion

Introduction

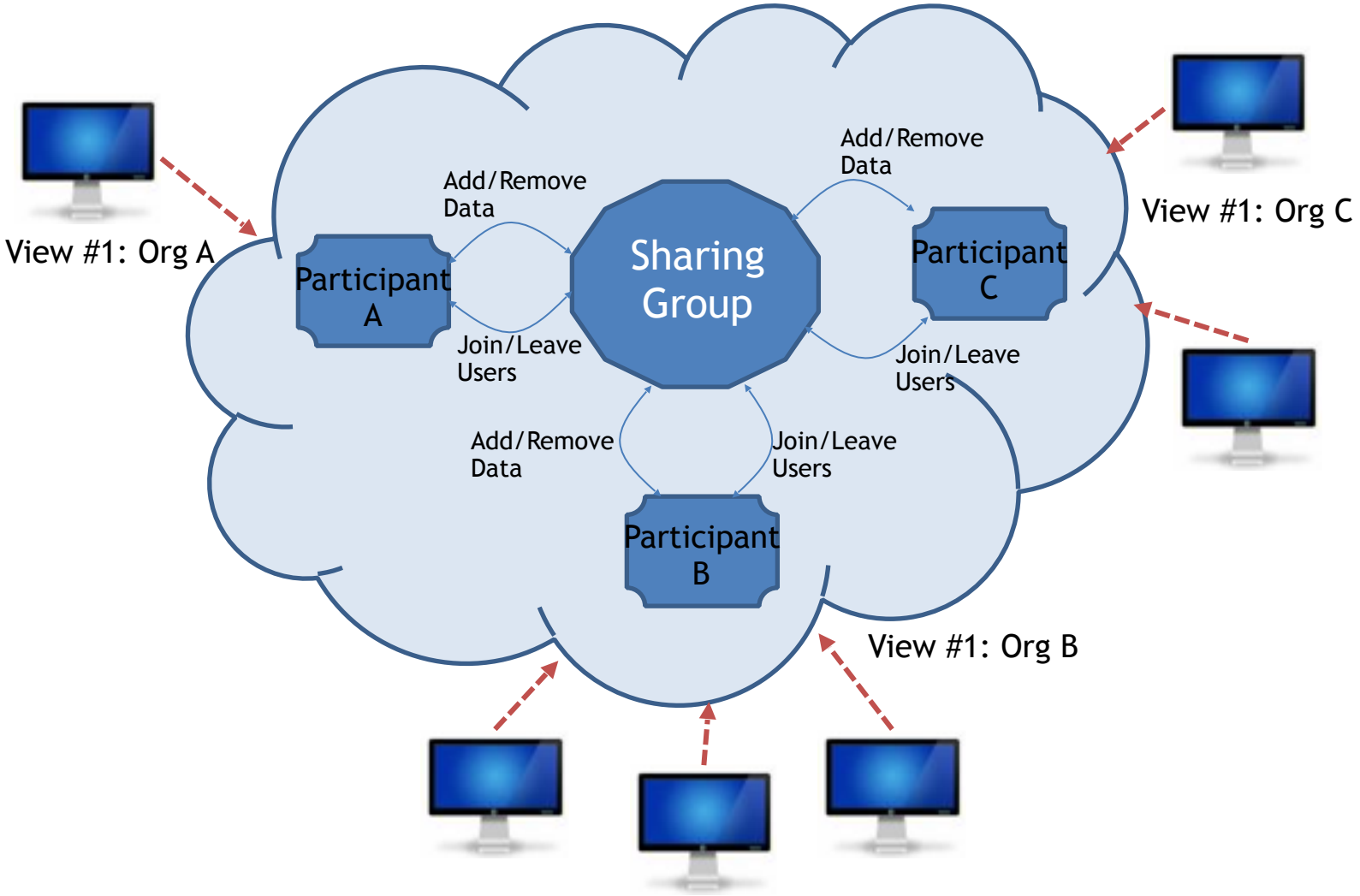
- Traditional Cyber Collaboration
  - Subscription services
  - Limitations
    - Organizations Sharing information through subscription.
    - Organizations are not actively participating in analyzing and processing the cyber information they submit.
    - Organizations don't directly interact with each other on sharing activities.
- Cloud IaaS Advantages for Cyber Incident Sharing
  - Virtualized resources
  - Operational efficiency
    - Light-weight and agile
    - Rapid deployment and configuration
    - Dynamic scaling
    - Self-service

Background/Related Work

- Group-Centric Sharing
  - Sharing for a specific purpose or mission
    - E.g. Collaboration in joint product design.
    - E.g. Inter-organizational collaboration.
  - Brings users & objects together in a group
    - Secure Meeting Room



Sharing Model in Cloud IaaS



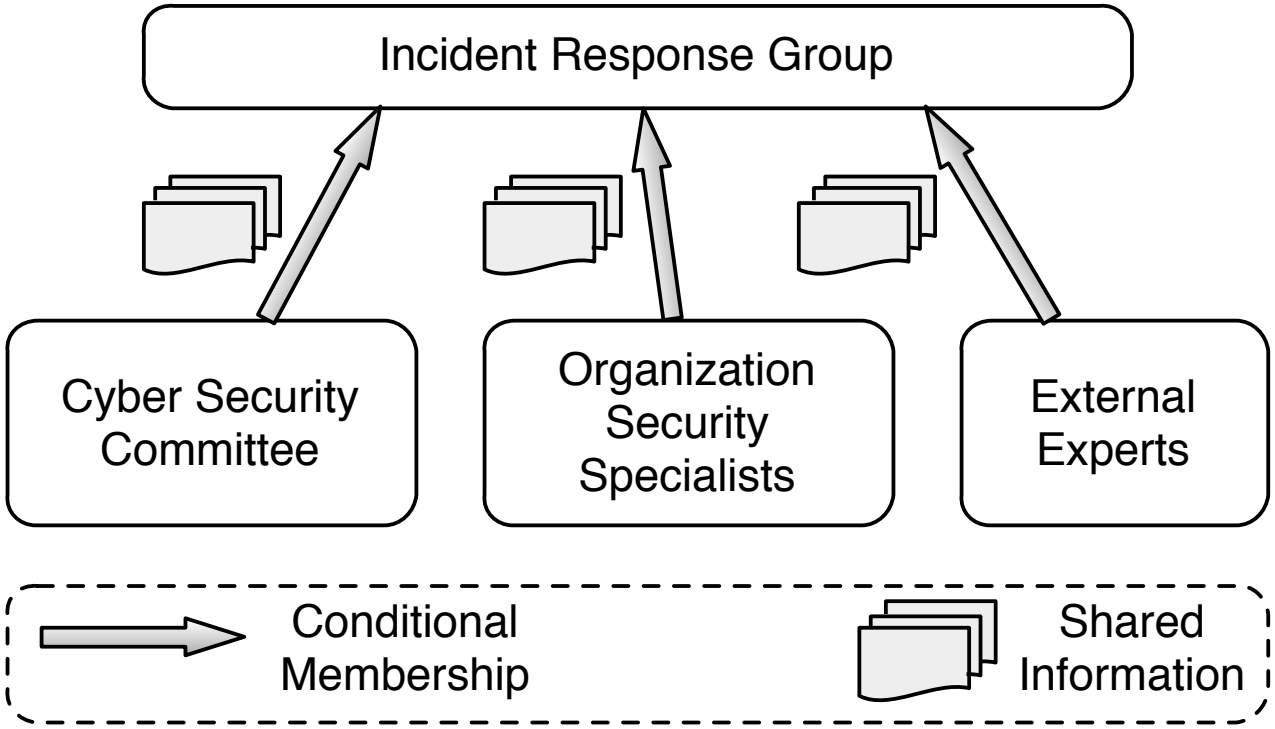
Refererence: Ram Krishnan, Ravi Sandhu, Jianwei Niu and William Winsborough, Towards a framework for group-centric secure collaboration, In Proceedings 5th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2009.

UTSA

I·C·S The Institute for Cyber Security

Background/Related Work

- Community Cyber Incident Response



Adapted from: Ravi Sandhu, Khalid Zaman Bijon, Xin Jin and Ram Krishnan, RT-based administrative models for community cyber security information sharing. In Proceedings of the 6th IEEE International Workshop on Trusted Collaboration (TrustCol), 2011.

UTSA

I·C·S The Institute for Cyber Security

Problem and Statement

- Problem Statement

  There is lack of access control models for information and resource sharing within collaborative groups in IaaS cloud platforms.

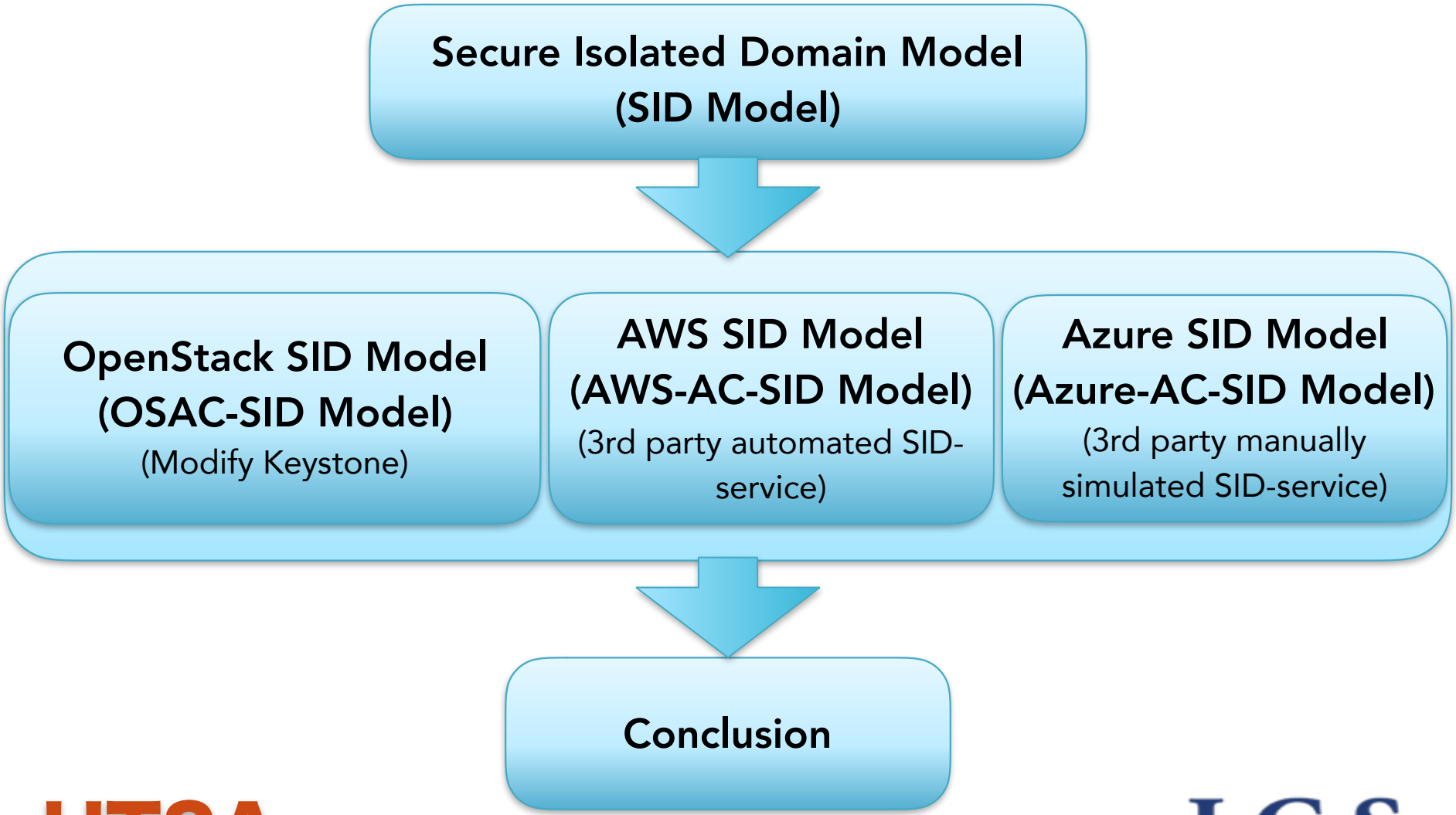
- Thesis Statement

  Secure sharing information and resources in IaaS cloud can be achieved by a common access control model that is enforceable in the currently dominant cloud IaaS platforms (viz., OpenStack, AWS and Azure).

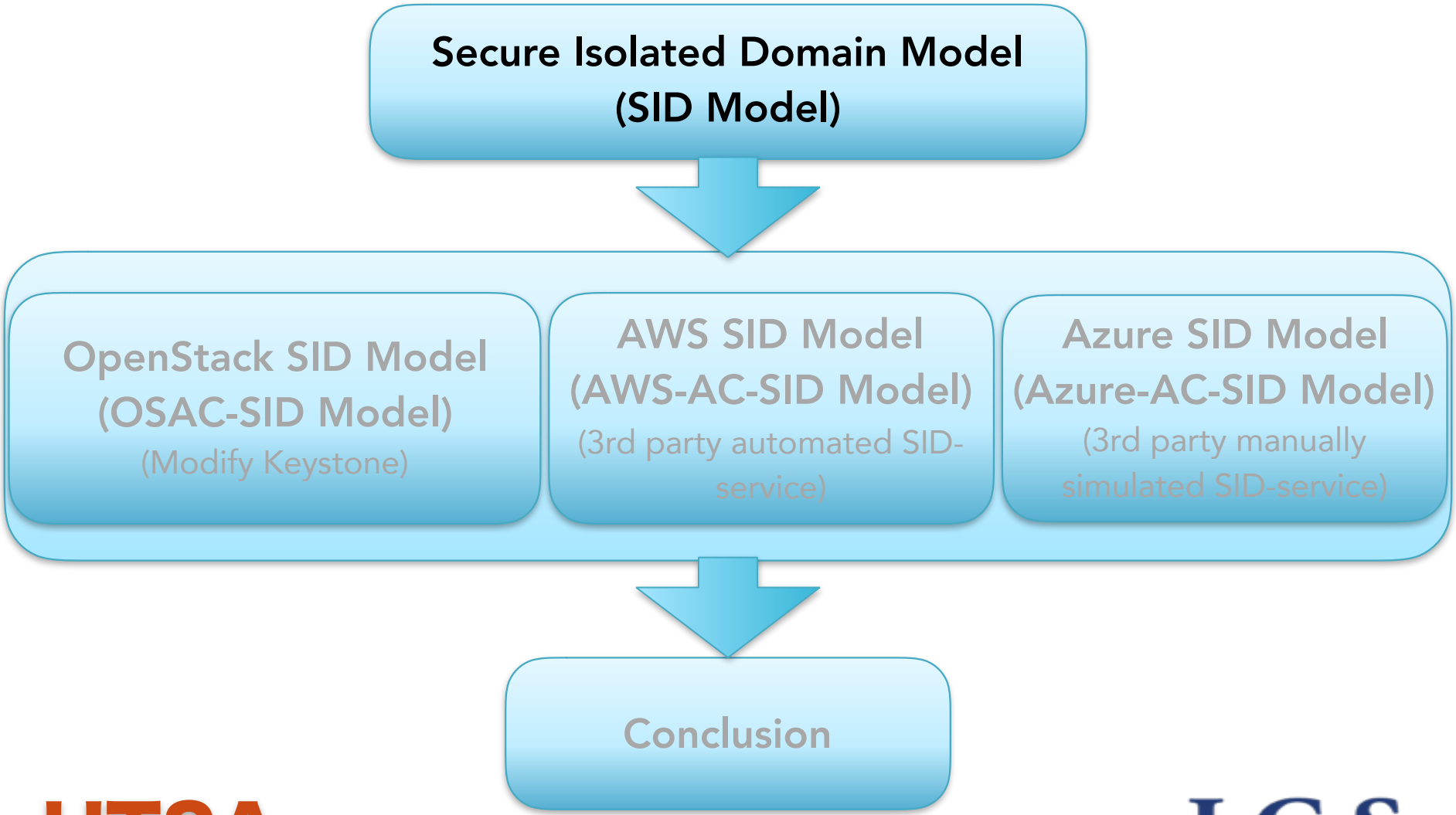
Scope

- Sharing models — sharing amongst a set of organizations
- Cloud deployment models — a single public or community cloud
- Cloud service models — focus on Infrastructure as a Service (IaaS)
- Scenario — Cyber Incident Response

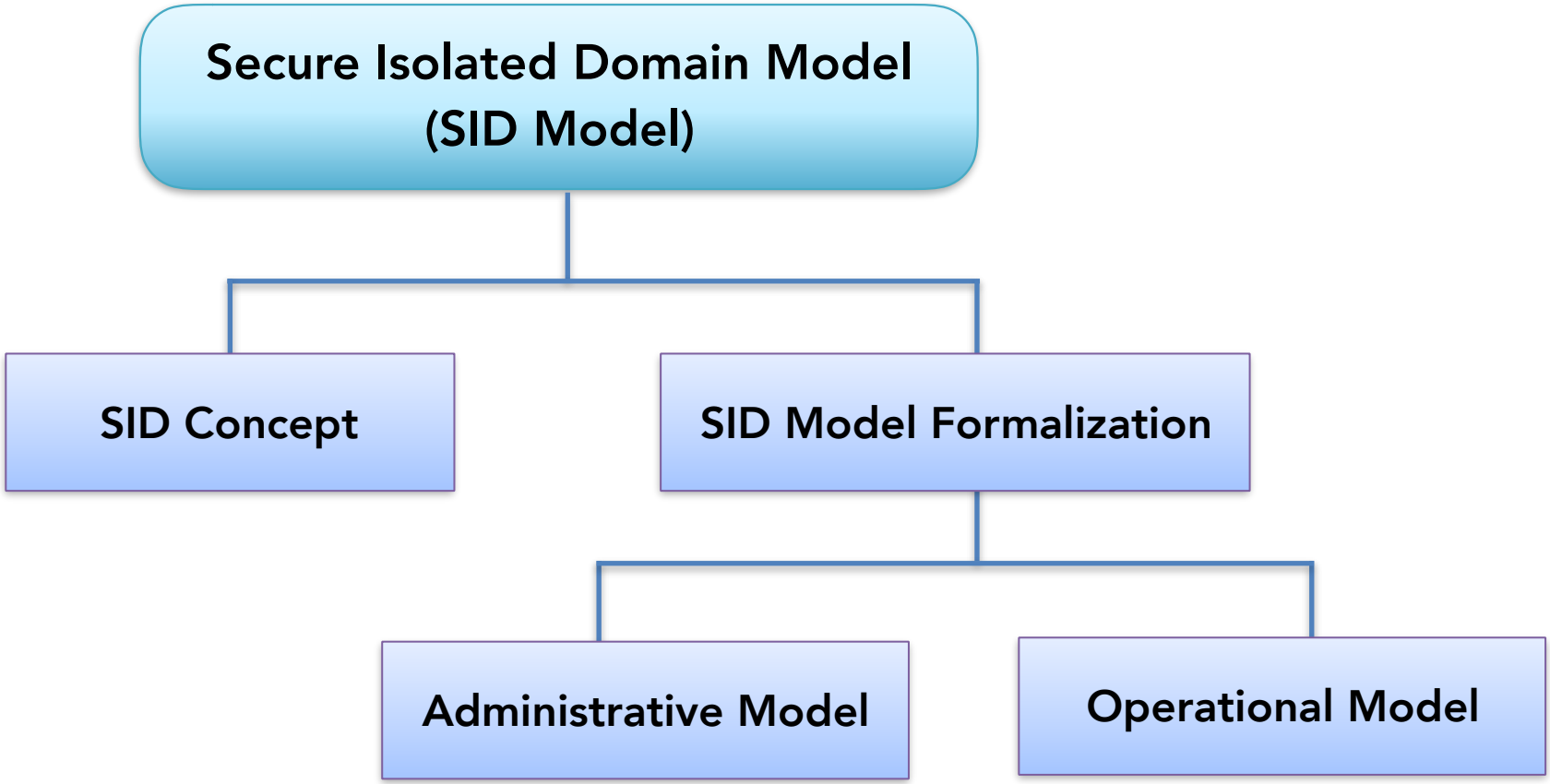
Outline


Secure Isolated Domain Model
(SID Model)
OpenStack SID Model
(OSAC-SID Model)
(Modify Keystone)
AWS SID Model
(AWS-AC-SID Model)
(3rd party automated SID-service)
Azure SID Model
(Azure-AC-SID Model)
(3rd party manually simulated SID-service)
Conclusion

Outline

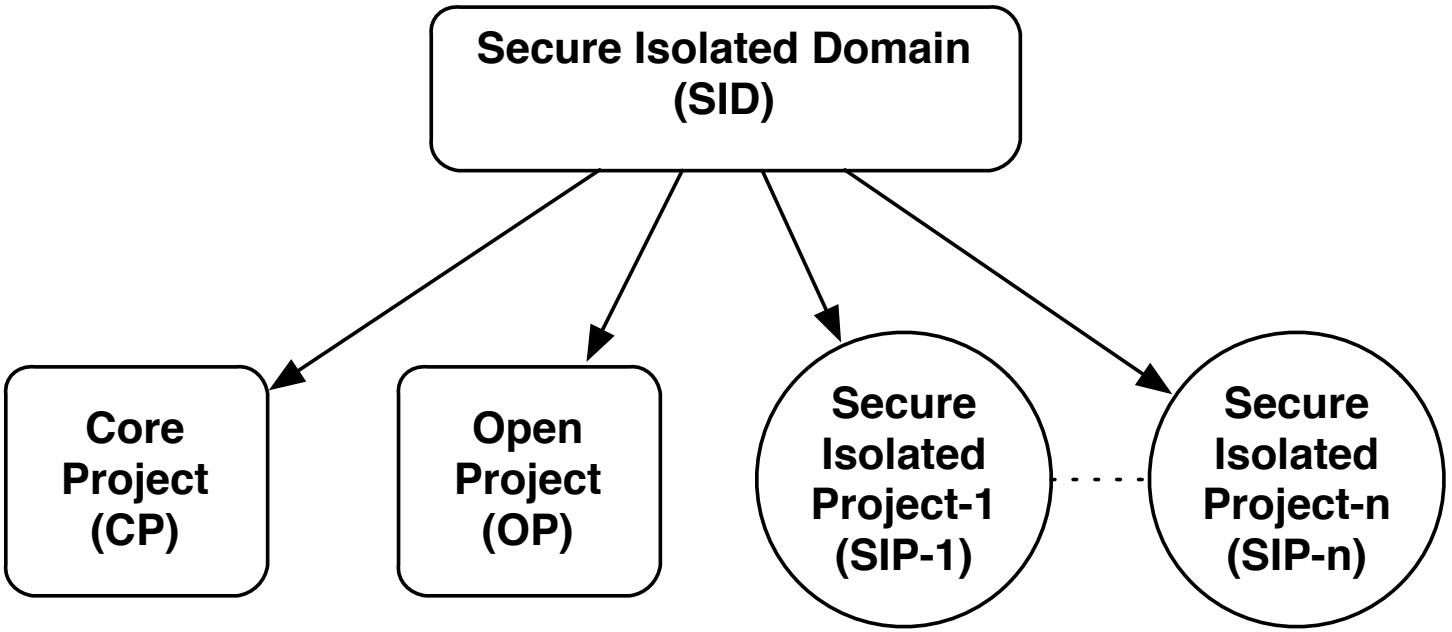

Secure Isolated Domain Model
(SID Model)
OpenStack SID Model
(OSAC-SID Model)
(Modify Keystone)
AWS SID Model
(AWS-AC-SID Model)
(3rd party automated SID-service)
Azure SID Model
(Azure-AC-SID Model)
(3rd party manually simulated SID-service)
Conclusion

SID Model


Secure Isolated Domain Model
(SID Model)
SID Concept
SID Model Formalization
Administrative Model
Operational Model

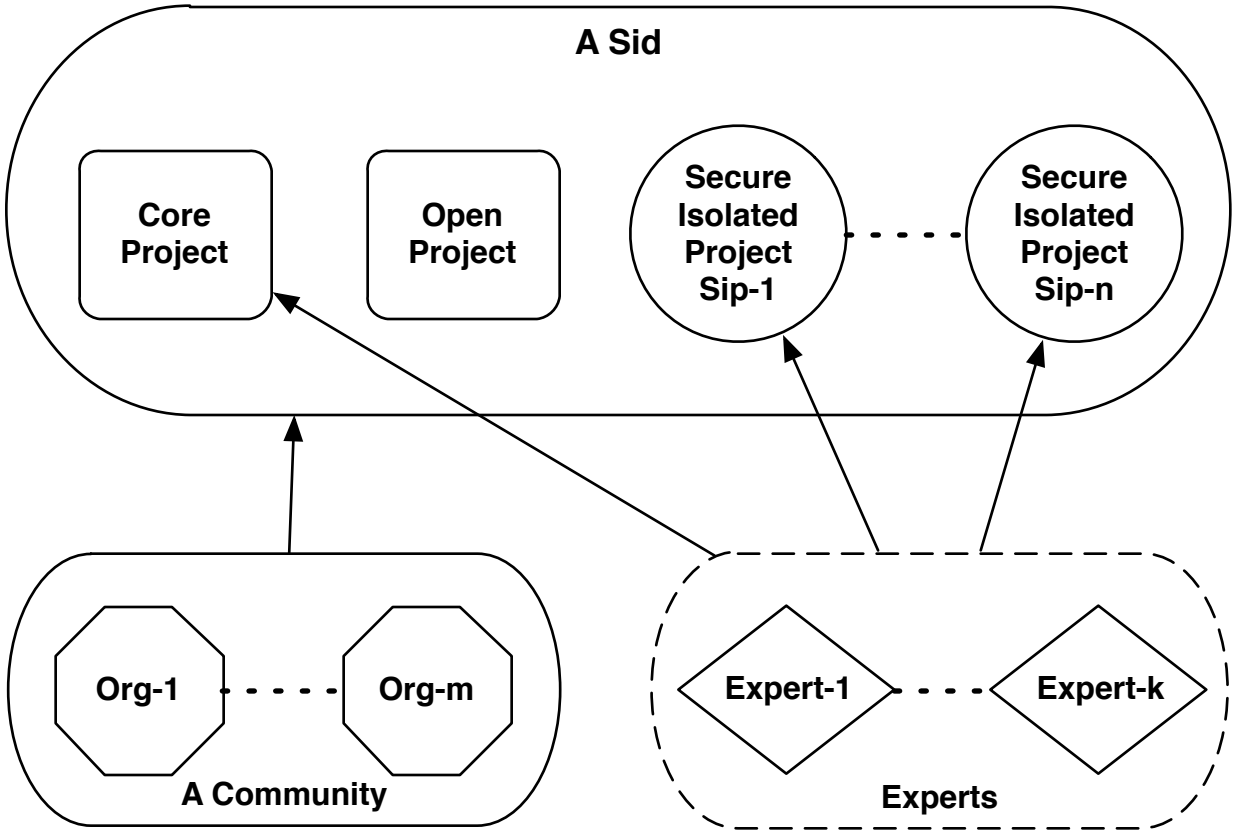
SID Concept

- Secure Isolated Domain (SID)

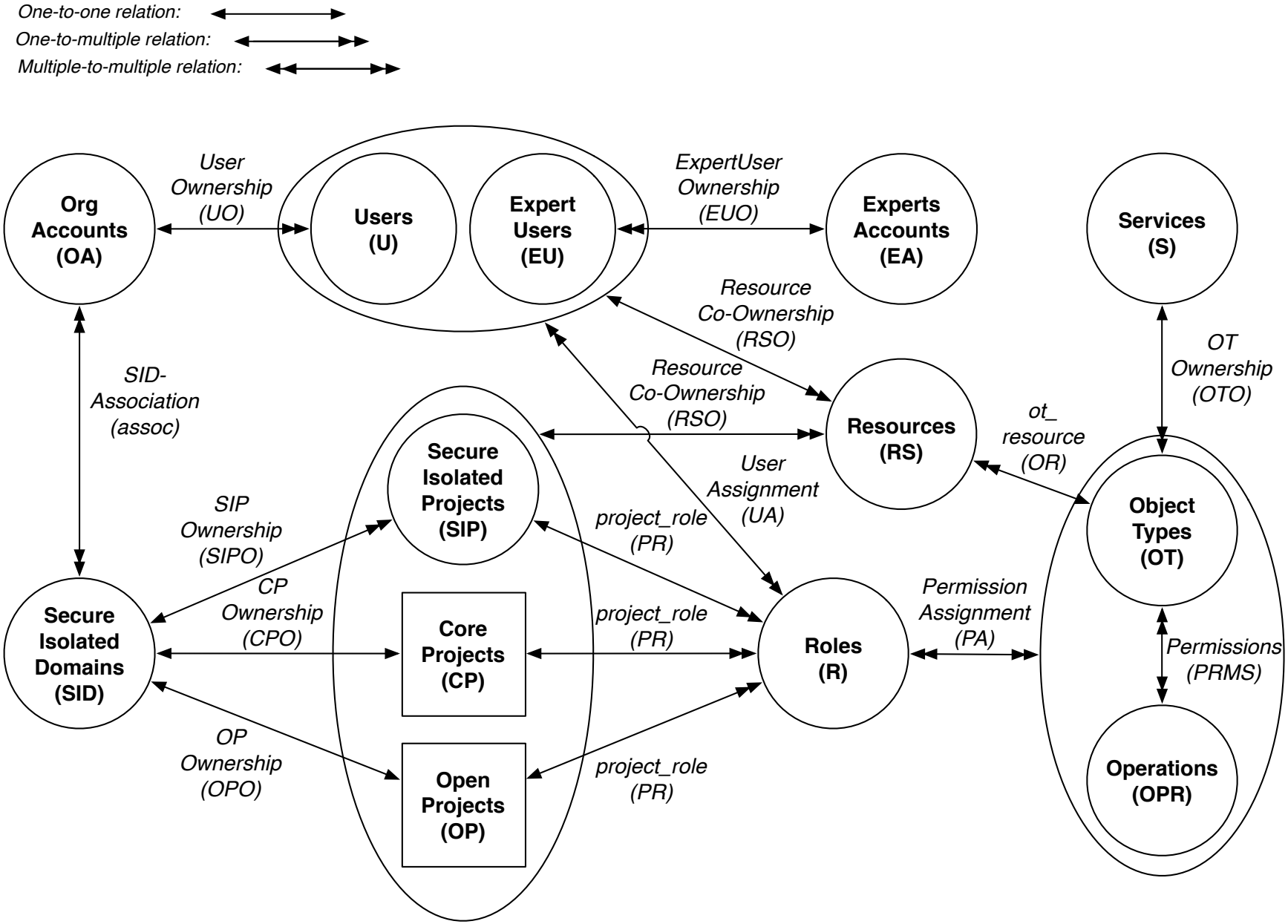


SID Concept

- A Community with a Sid



SID Model


One-to-one relation:
One-to-multiple relation:
Multiple-to-multiple relation:
Org Accounts (OA)
User Ownership (UO)
Users (U)
Expert Users (EU)
ExpertUser Ownership (EUO)
Experts Accounts (EA)
Services (S)
Resource Co-Ownership (RSO)
OT Ownership (OTO)
SID-Association (assoc)
Resource Co-Ownership (RSO)
Resources (RS)
ot_ resource (OR)
User Assignment (UA)
Secure Isolated Projects (SIP)
SIP Ownership (SIPO)
project_role (PR)
Object Types (OT)
CP Ownership (CPO)
Secure Isolated Domains (SID)
Core Projects (CP)
project_role (PR)
Roles (R)
Permission Assignment (PA)
Permissions (PRMS)
OP Ownership (OPO)
Open Projects (OP)
project_role (PR)
Operations (OPR)

SID Administrative Model

- SidCreate/SidDelete
  - An admin user representing uSet creates/deletes a sid
- SipCreate/SipDelete
  - An admin user representing uSet creates/deletes a sip
- UserAdd/UserRemove
  - Admin users add/remove a user from his home domain to a cp/op/sip
- EUserAdd/EUserRemove
  - Admin users add/remove an expert user to a cp/sip

SID Administrative Model formalization

Operation	Authorization Requirement	Update
SidCreate(adminu, uSet, sid) */* An admin user representing uSet creates a sid */*	adminu $\in$ *uSet* $\wedge$ adminu $\in$ U $\wedge$ sid $\notin$ SID	SID′ = SID $\cup$ {sid}; assoc(sid) = $\bigcup_{adminu \in uSet}$ UO(adminu); CP′ = CP $\cup$ {cp}; CPO(cp) = sid; OP′ = OP $\cup$ {op}; OPO(op) = sid; UA′ = (uSet, *SIDadmin*) $\cup$ UA; PR′ = PR $\cup$ {(cp, *SIDadmin*), (op, *SIDadmin*)}.
SidDelete(adminu, uSet, sid) */* An admin user representing uSet deletes the sid*/*	adminu $\in$ *uSet* $\wedge$ adminu $\in$ U $\wedge$ (adminu, *SIDadmin*) $\in$ UA $\wedge$ assoc(sid) = $\bigcup_{adminu \in uSet}$ UO(adminu) $\wedge$ sid $\in$ SID	SID′ = SID - {sid}; assoc(sid) = NULL; CP′ = CP - {cp}; CPO(cp) = NULL; OP′ = OP - {op}; OPO(op) = NULL; UA′ = UA - (uSet, *SIDadmin*); PR′ = PR - {(cp, *SIDadmin*), (op, *SIDadmin*)}; if $\exists$ u $\in$ (U $\cup$ EU).((u, *SIDmember*) $\in$ UA), then UA′ = UA - (u, *SIDmember*); if $\exists$ sip $\in$ SIP.(SIPO(sip) = sid), then SIP′ = SIP - sip $\wedge$ PR′ = PR - {(sip, *SIDadmin*), (sip, *SIDmember*)}.

SID Administrative Model formalization

Operation	Authorization Requirement	Update
SipCreate(adminu, sip, sid) */* An admin user representing uSet creates a sip */*	adminu ∈ U ∧ (adminu, *SIDadmin*) ∈ UA ∧ UO(adminu) ∈ assoc(sid) ∧ sip ∉ SIP	SIP′ = SIP ∪ {sip}; SIPO(sip) = sid; PR′ = PR ∪ {(sip, *SIDadmin*)}.
SipDelete(adminu, sip, sid) */* An admin user representing uSet deletes a sip*/*	adminu ∈ U ∧ (adminu, *SIDadmin*) ∈ UA ∧ UO(adminu) ∈ assoc(sid) ∧ SIPO(sip) = sid	SIP′ = SIP - {sip}; SIPO(sip) = NULL; PR′ = PR - {(sip, *SIDadmin*)}.
UserAdd(adminu, u, p, sid) */* Admin users add a user from his home domain to a cp, op or sip */*	adminu ∈ U ∧ (adminu, *SIDadmin*) ∈ UA ∧ (p, *SIDadmin*) ∈ PR ∧ u ∈ U ∧ UO(u) = UO(adminu) ∧ p ∈ (CP ∪ OP ∪ SIP) ∧ (CPO(p) = sid ∨ OPO(p) = sid ∨ SIP(p) = sid)	UA′ = UA ∪ {(u, *SIDmember*)}.
UserRemove(adminu, u, p, sid) */* Admin users remove a user from a cp, op or sip */*	adminu ∈ U ∧ (adminu, *SIDadmin*) ∈ UA ∧ (p, *SIDadmin*) ∈ PR ∧ u ∈ U ∧ UO(u) = UO(adminu) ∧ p ∈ (CP ∪ OP ∪ SIP) ∧ (CPO(p) = sid ∨ OPO(p) = sid ∨ SIP(p) = sid) ∧ (u, *SIDmember*) ∈ UA ∧ (p, *SIDmember*) ∈ PR	UA′ = UA - {(u, *SIDmember*)}.
EUserAdd(adminu, eu, p, sid) */* Admin users add an expert user to a cp or sip */*	adminu ∈ U ∧ (adminu, *SIDadmin*) ∈ UA ∧ (p, *SIDadmin*) ∈ PR ∧ eu ∈ EU ∧ p ∈ (CP ∪ SIP) ∧ (CPO(p) = sid ∨ SIPO(p) = sid)	UA′ = UA ∪ {(eu, *SIDmember*)}.
EUserRemove(adminu, eu, p, sid) */* Admin users remove an expert user from a cp or sip */*	adminu ∈ U ∧ (adminu, *SIDadmin*) ∈ UA ∧ (p, *SIDadmin*) ∈ PR ∧ eu ∈ EU ∧ p ∈ (CP ∪ SIP) ∧ (CPO(p) = sid ∨ SIPO(p) = sid) ∧ (eu, *SIDmember*) ∈ UA ∧ (p, *SIDmember*) ∈ PR	UA′ = UA - {(eu, *SIDmember*)}.

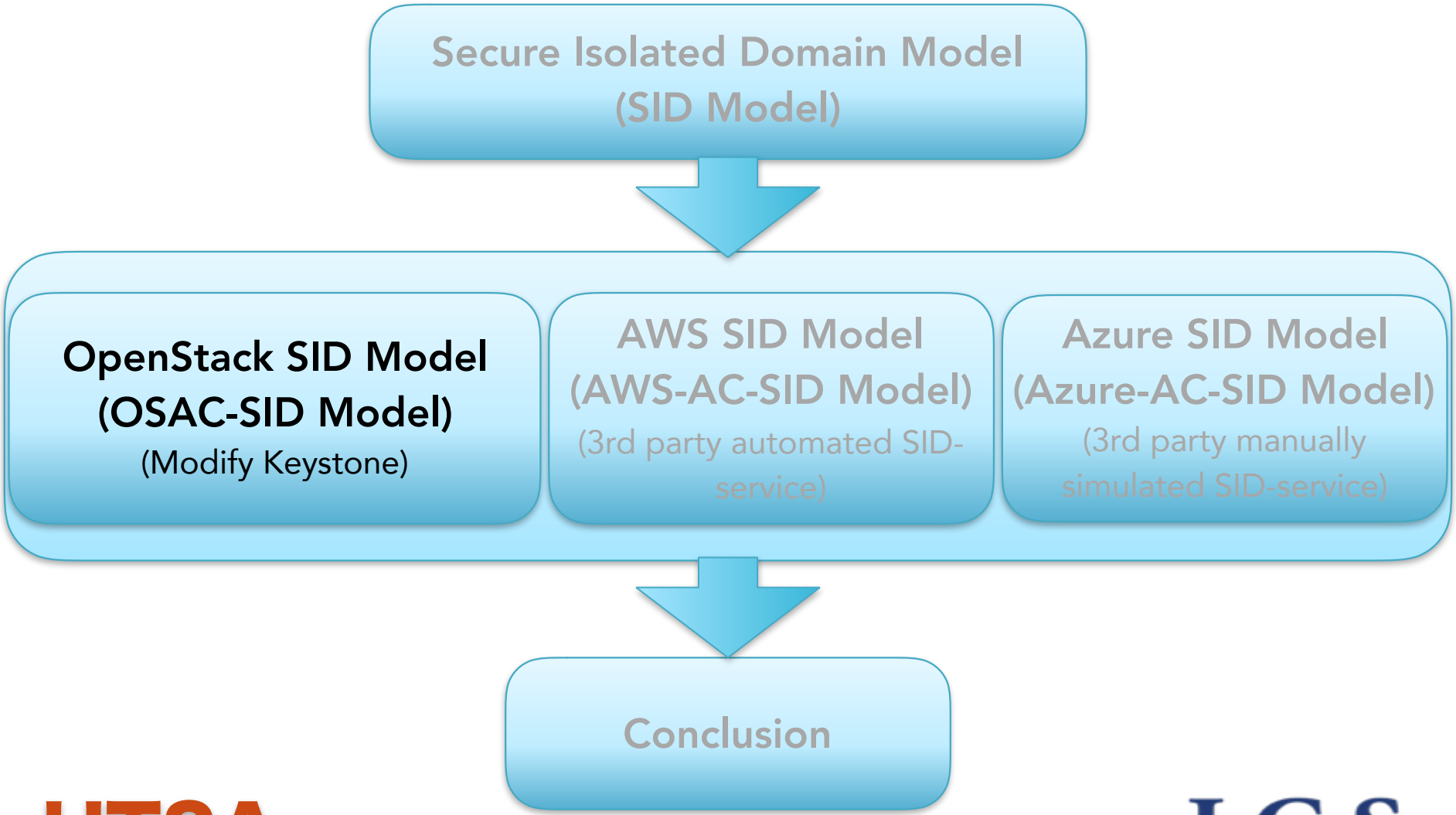
SID Operational Model formalization

- CreateVM/DeleteVM
  - A user creates/deletes a vm
- CreateSContainer/DeleteSContainer
  - A user creates/deletes a storage container
- CreateObject/DeleteObject
  - A user creates/deletes a storage container object

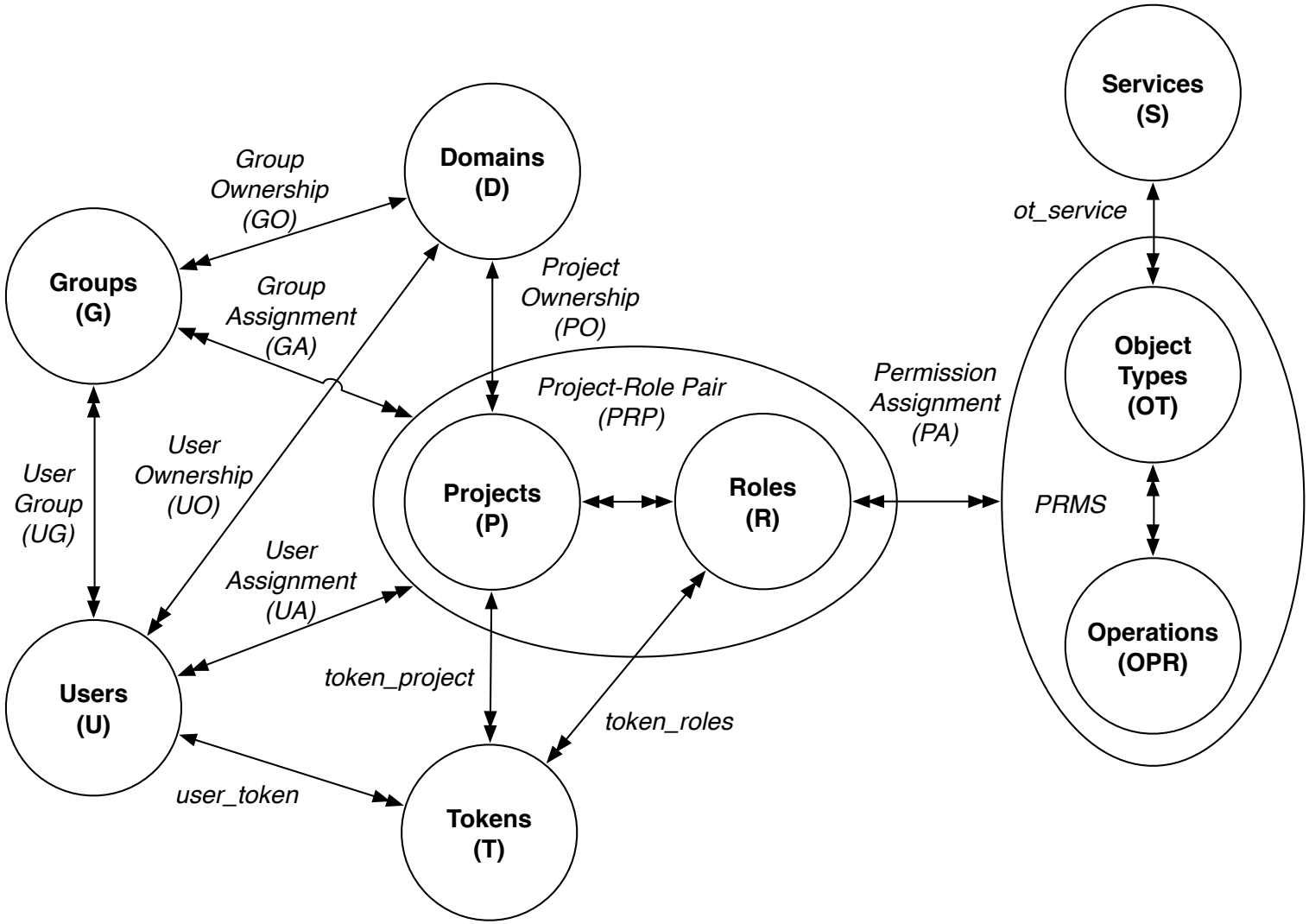
SID Operational Model

Operation	Authorization Requirement	Update
CreateVM(vm, p, u) */* A user creates a vm */*	$vm \notin RS \wedge p \in (CP \cup OP \cup SIP) \wedge u \in U \wedge \exists (perms, r) \in PA.(perms = (vm, create) \wedge (p, r) \in PR \wedge (u, (p, r)) \in UA)$	$RS' = RS \cup \{vm\}$; $RSO' = RSO \cup \{(vm, (p, u))\}$; $OR(vm) = VM$.
DeleteVM(vm, p, u) */* A user deletes a vm */*	$vm \in RS \wedge RSO(vm) = \{(p, u)\} \wedge p \in (CP \cup OP \cup SIP) \wedge u \in U \wedge \exists (perms, r) \in PA.(perms = (vm, delete) \wedge (p, r) \in PR \wedge (u, (p, r)) \in UA$	$RS' = RS - \{vm\}$; $RSO' = RSO - \{(vm, (p, u))\}$; $vm = NULL$.
CreateSContainer(sc, p, u) */* A user creates a storage container */*	$sc \notin RS \wedge p \in (CP \cup OP \cup SIP) \wedge u \in U \wedge \exists (perms, r) \in PA.(perms = (sc, create) \wedge (p, r) \in PR \wedge (u, (p, r)) \in UA)$	$RS' = RS \cup \{sc\}$; $RSO' = RSO \cup \{(sc, (p, u))\}$; $OR(sc) = SC$.
DeleteSContainer(sc, p, u) */* A user deletes a storage container */*	$sc \in RS \wedge RSO(sc) = \{(p, u)\} \wedge p \in (CP \cup OP \cup SIP) \wedge u \in U \wedge \exists (perms, r) \in PA.(perms = (sc, delete) \wedge (p, r) \in PR \wedge (u, (p, r)) \in UA$	$RS' = RS - \{sc\}$; $RSO' = RSO - \{(sc, (p, u))\}$; $sc = NULL$.
CreateObject(co, sc, p, u) */* A user creates a storage container object */*	$co \notin RS \wedge sc \in RS \wedge p \in (CP \cup OP \cup SIP) \wedge u \in U \wedge RSO(sc) = (p, u) \wedge \exists (perms, r) \in PA.(perms = (co, create) \wedge (p, r) \in PR \wedge (u, (p, r)) \in UA)$	$RS' = RS \cup \{co\}$; $RSO' = RSO \cup \{(co, (p, u))\}$; $OR(co) = CO$.
DeleteObject(co, sc, p, u) */* A user delete a storage container object */*	$co \in RS \wedge RSO(co) = \{(p, u)\} \wedge sc \in RS \wedge p \in (CP \cup OP \cup SIP) \wedge u \in U \wedge RSO(sc) = (p, u) \wedge \exists (perms, r) \in PA.(perms = (co, create) \wedge (p, r) \in PR \wedge (u, (p, r)) \in UA)$	$RS' = RS - \{co\}$; $RSO' = RSO - \{(co, (p, u))\}$; $co = NULL$.

Outline

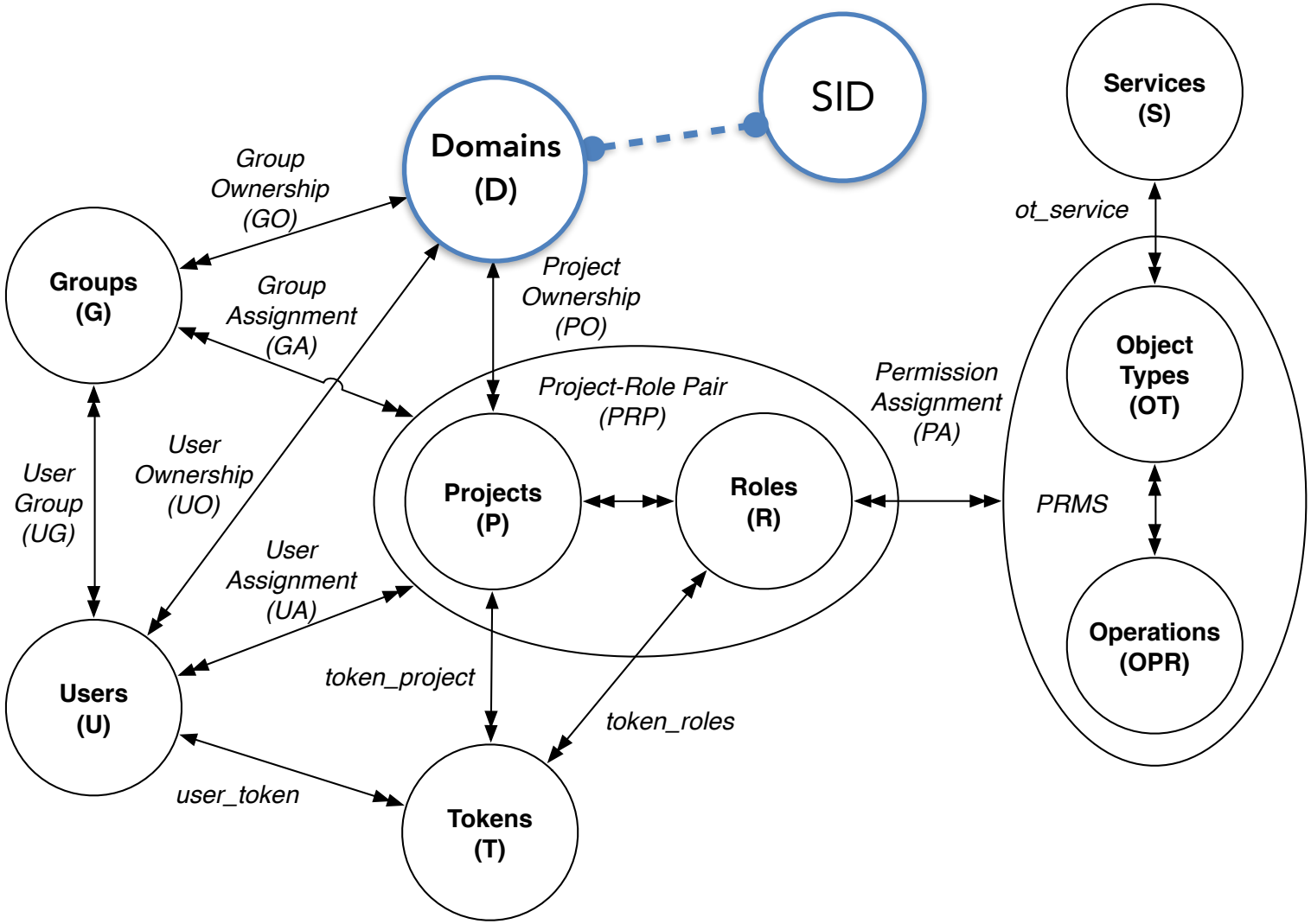

Secure Isolated Domain Model
(SID Model)
OpenStack SID Model
(OSAC-SID Model)
(Modify Keystone)
AWS SID Model
(AWS-AC-SID Model)
(3rd party automated SID-service)
Azure SID Model
(Azure-AC-SID Model)
(3rd party manually simulated SID-service)
Conclusion

OpenStack Access Control (OSAC) Model



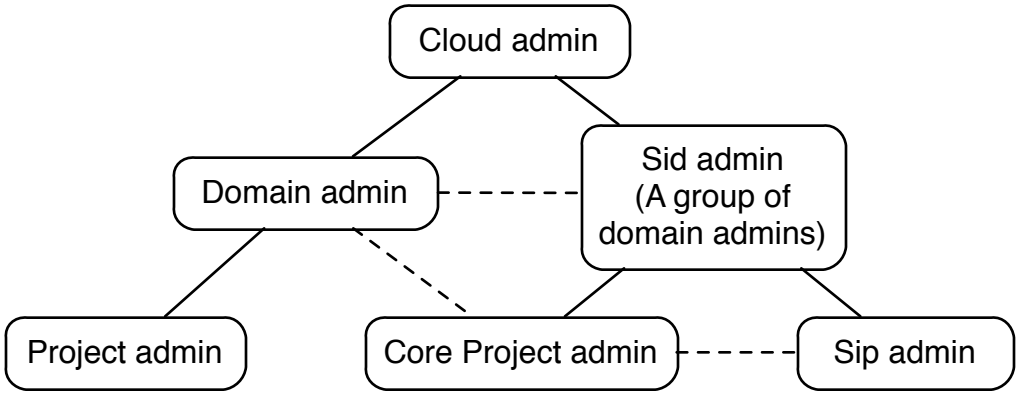
Reference: Bo Tang and Ravi Sandhu, Extending OpenStack access control with domain trust. In Proceedings 8th International Conference on Network and System Security (NSS), 2014.

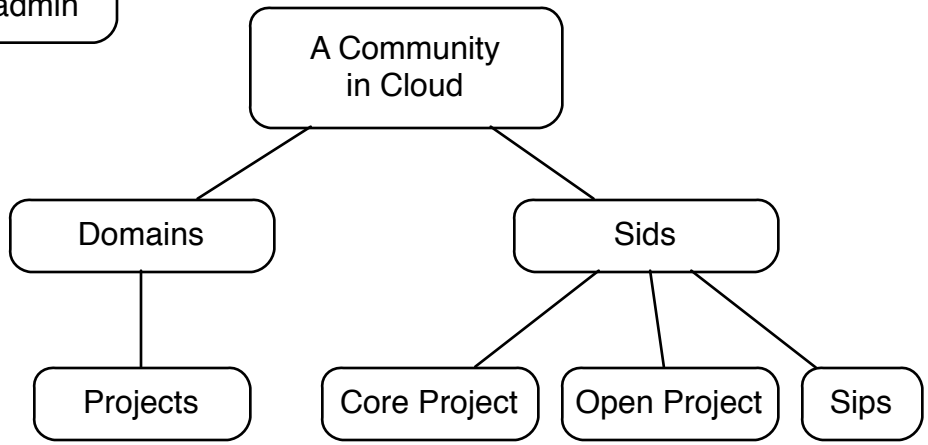
OSAC Model extends with SID



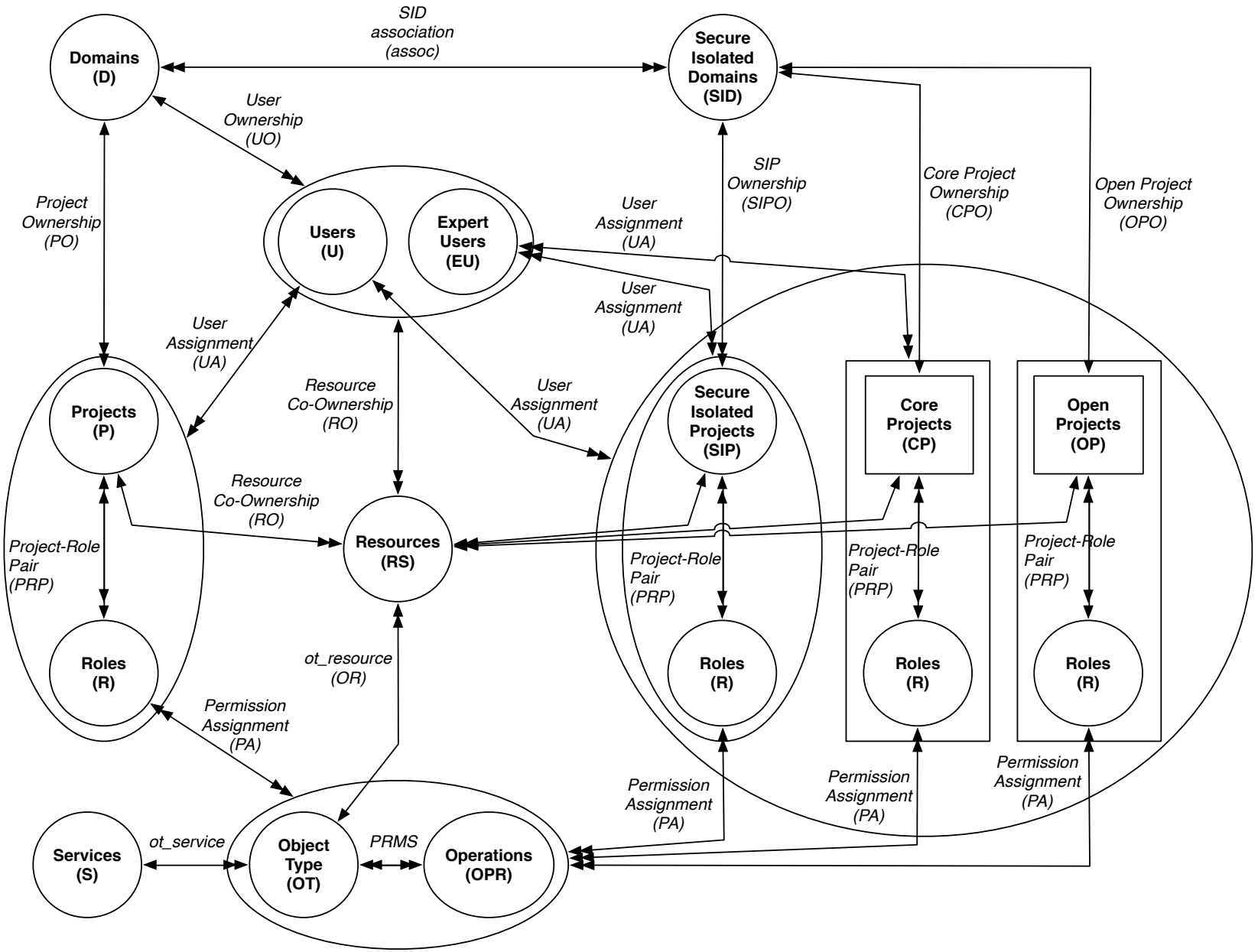
OSAC-SID Model

- Sid admins and cloud resource division

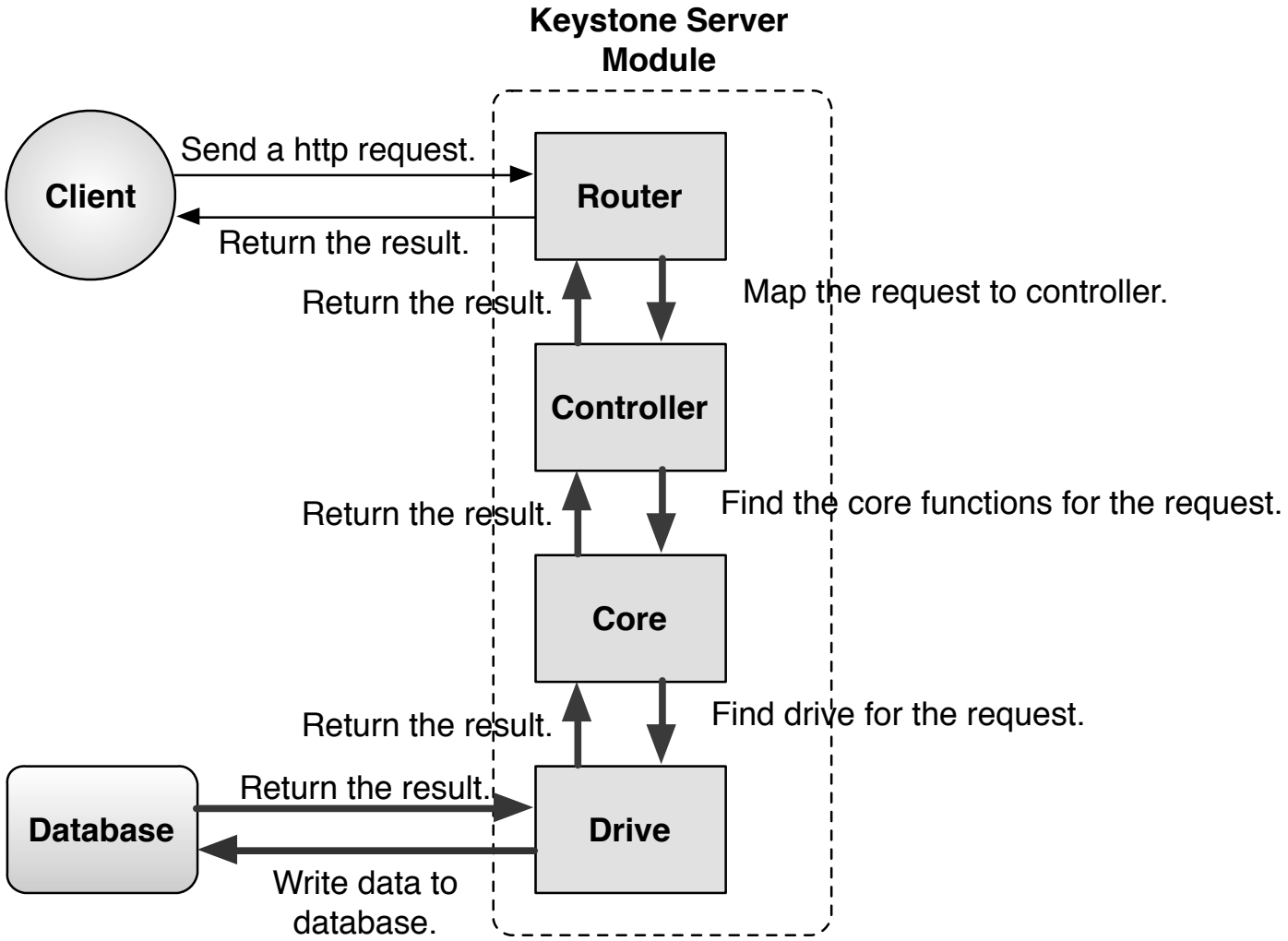




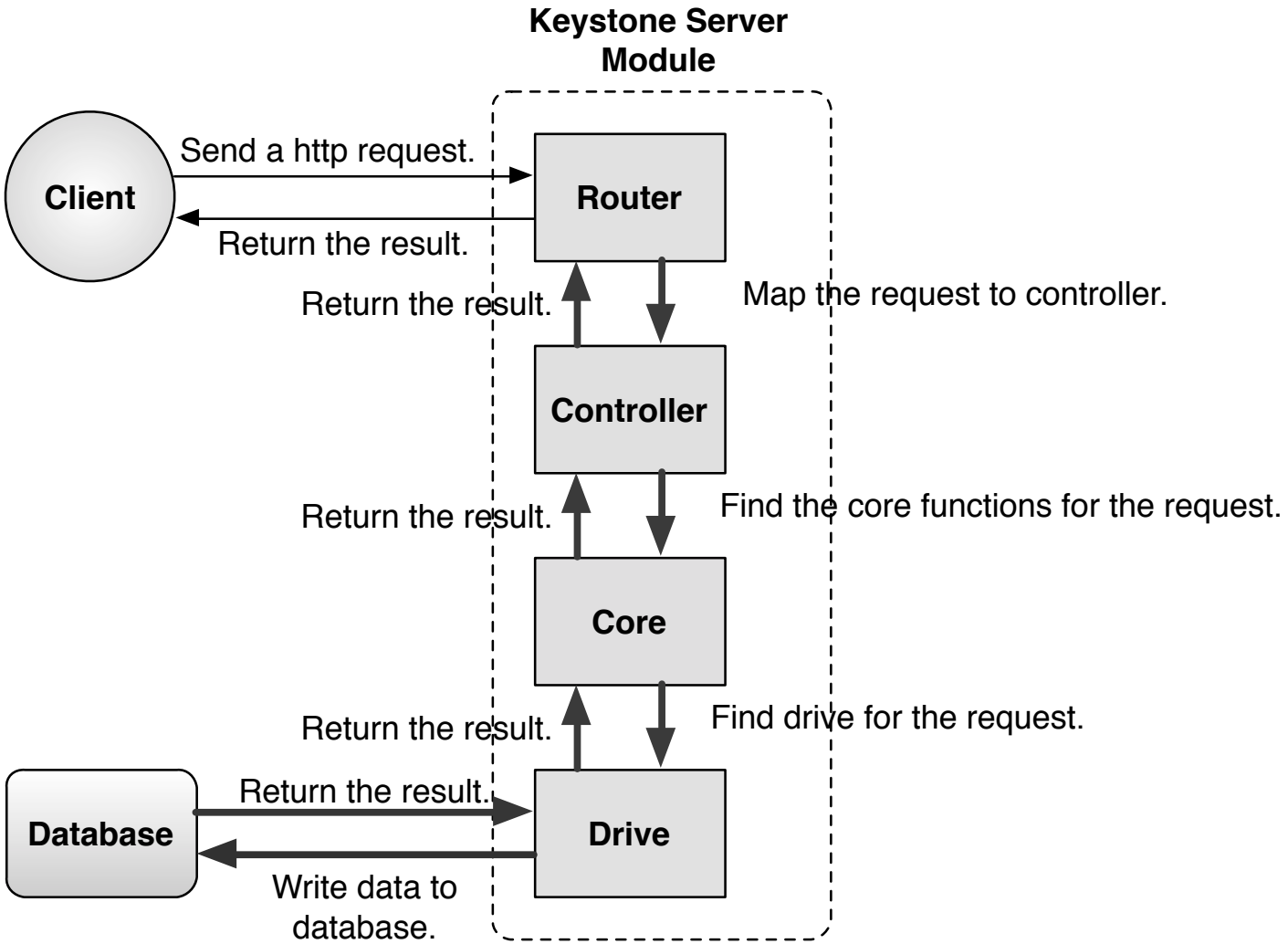
OSAC-SID Model



Enforcement - Keystone Protocol



Enforcement - Sid Request



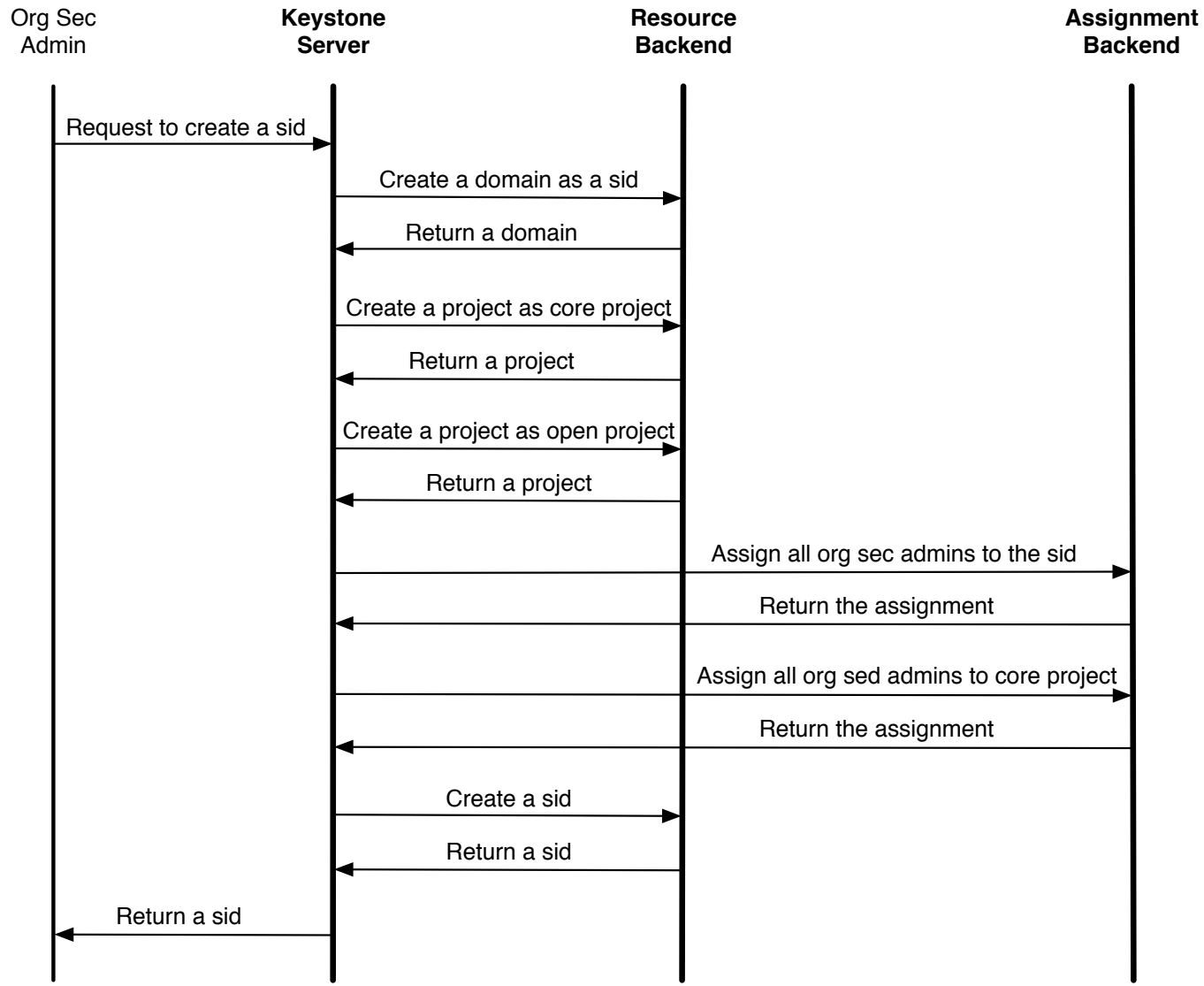
Enforcement - Backend

Sid Table in Database:

```
mysql> describe sid;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| sid_id       | varchar(64) | NO   | PRI |         |       |
| sid_name     | varchar(64) | YES  |     | NULL    |       |
| sid_members  | text        | YES  |     | NULL    |       |
| core_project | varchar(64) | YES  |     | NULL    |       |
| open_project | varchar(64) | YES  |     | NULL    |       |
| extra        | text        | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
```

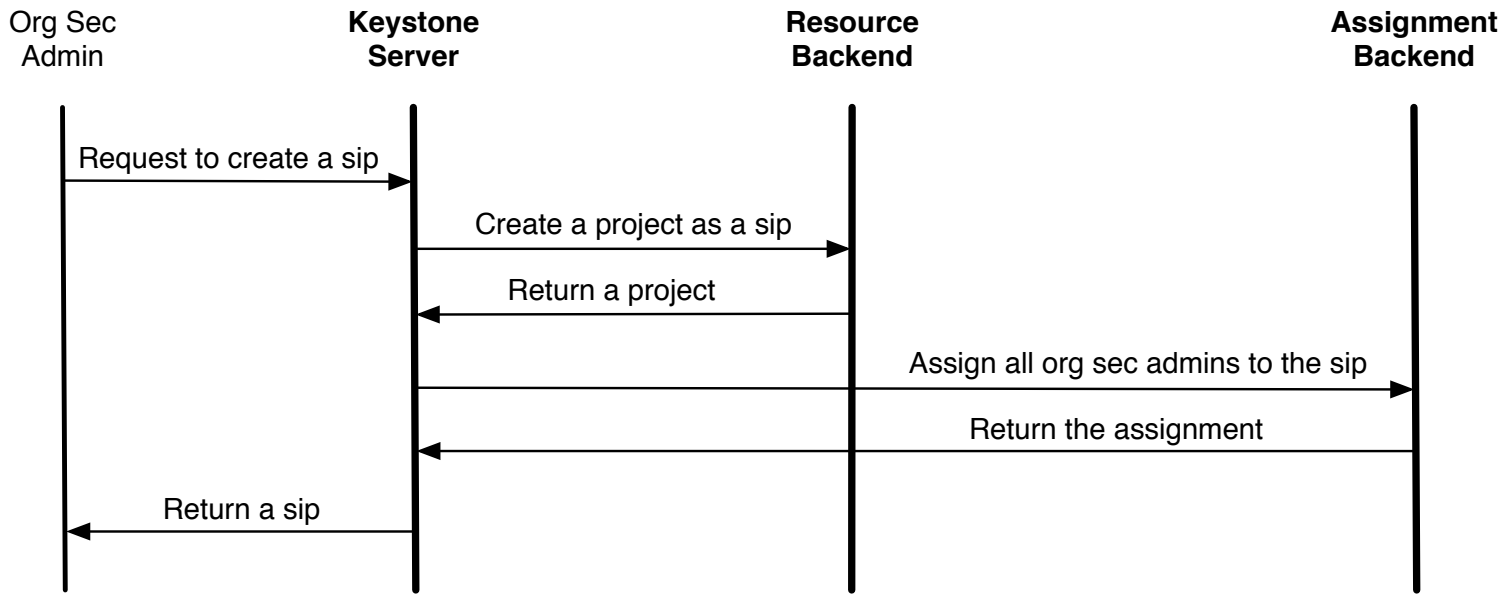
Enforcement

An Org SecAdmin Create a Sid:



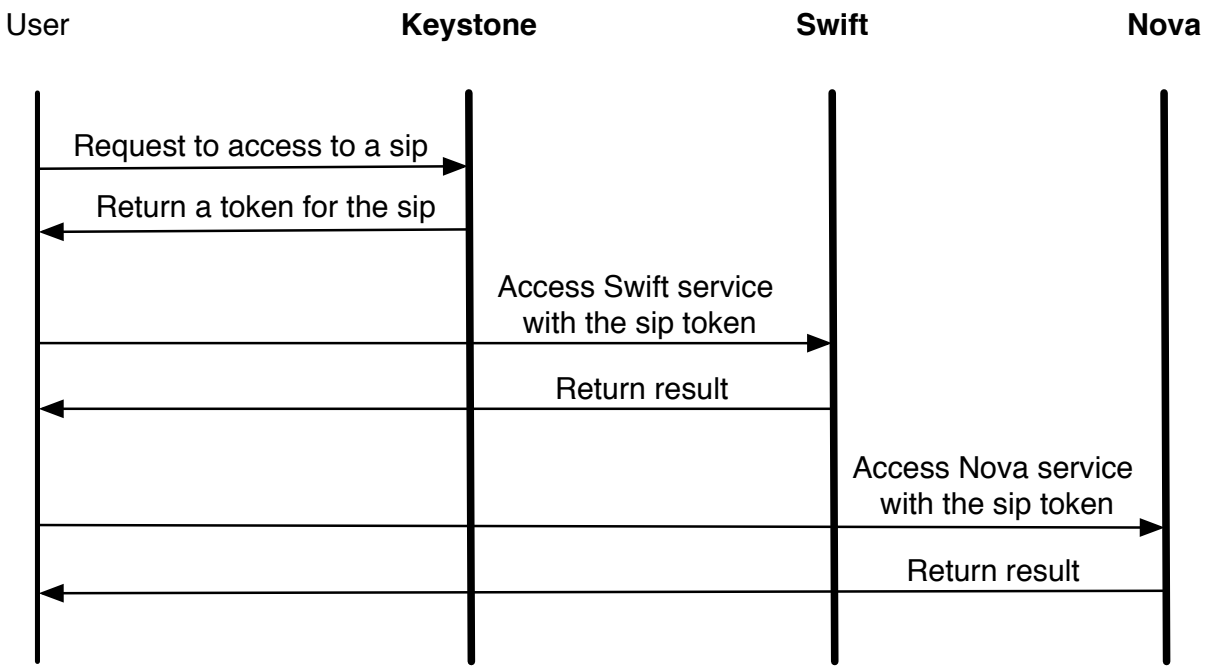
Enforcement

An Org SecAdmin Create a Sip:

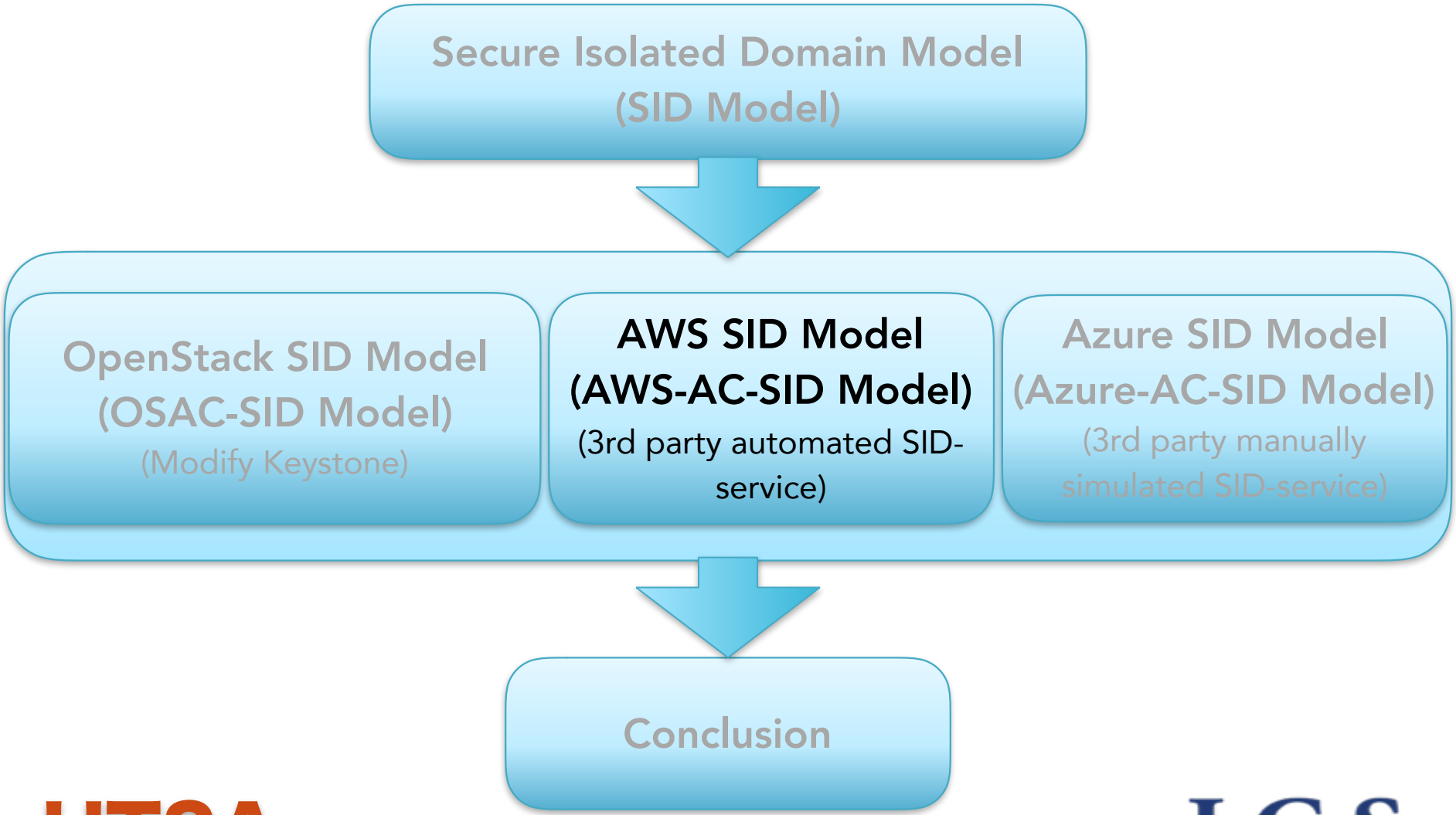


Enforcement

A User Access to a Sip:

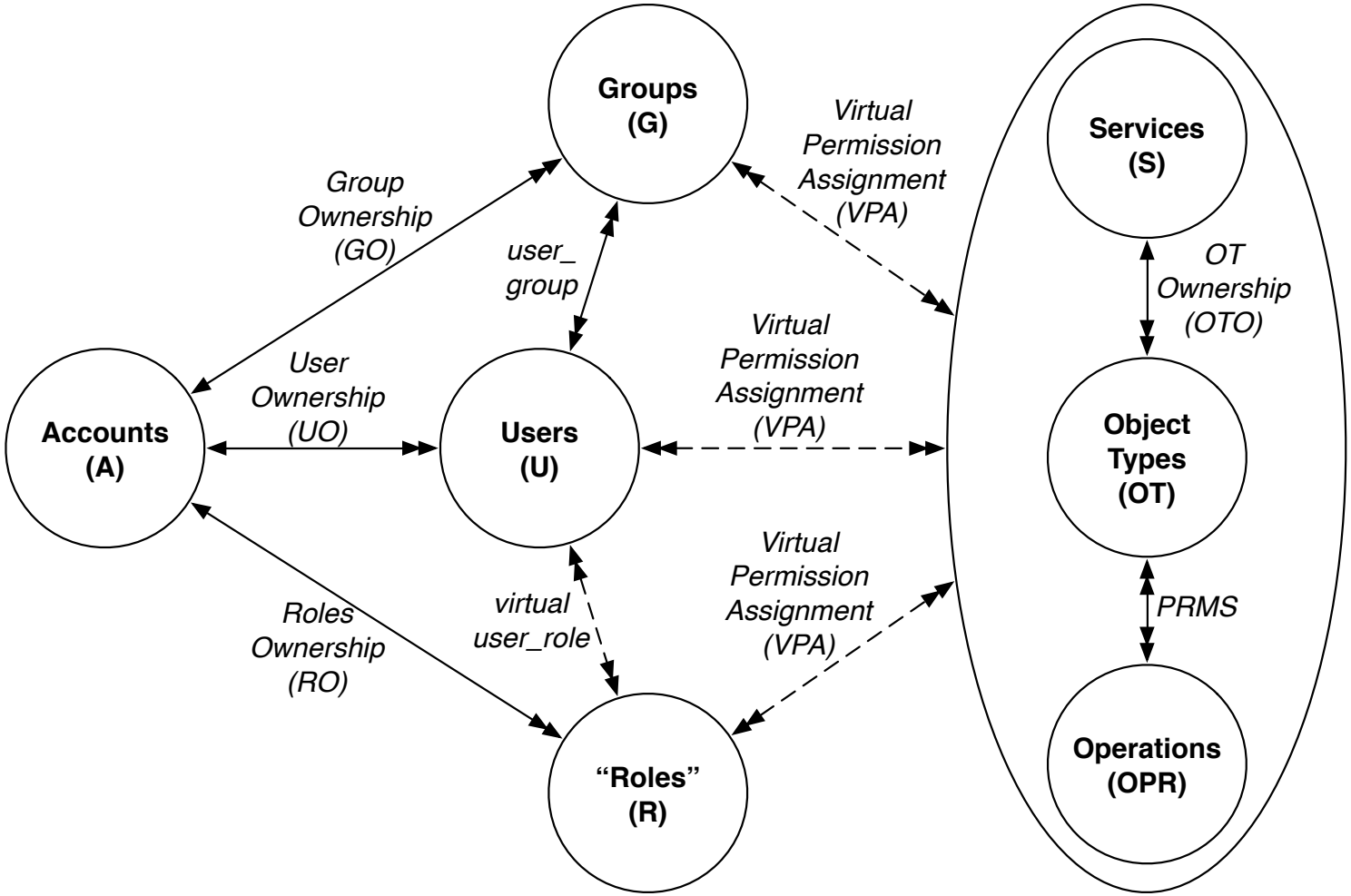


Outline


Secure Isolated Domain Model
(SID Model)
OpenStack SID Model
(OSAC-SID Model)
(Modify Keystone)
AWS SID Model
(AWS-AC-SID Model)
(3rd party automated SID-service)
Azure SID Model
(Azure-AC-SID Model)
(3rd party manually simulated SID-service)
Conclusion

AWS Access Control (AWS-AC) Model

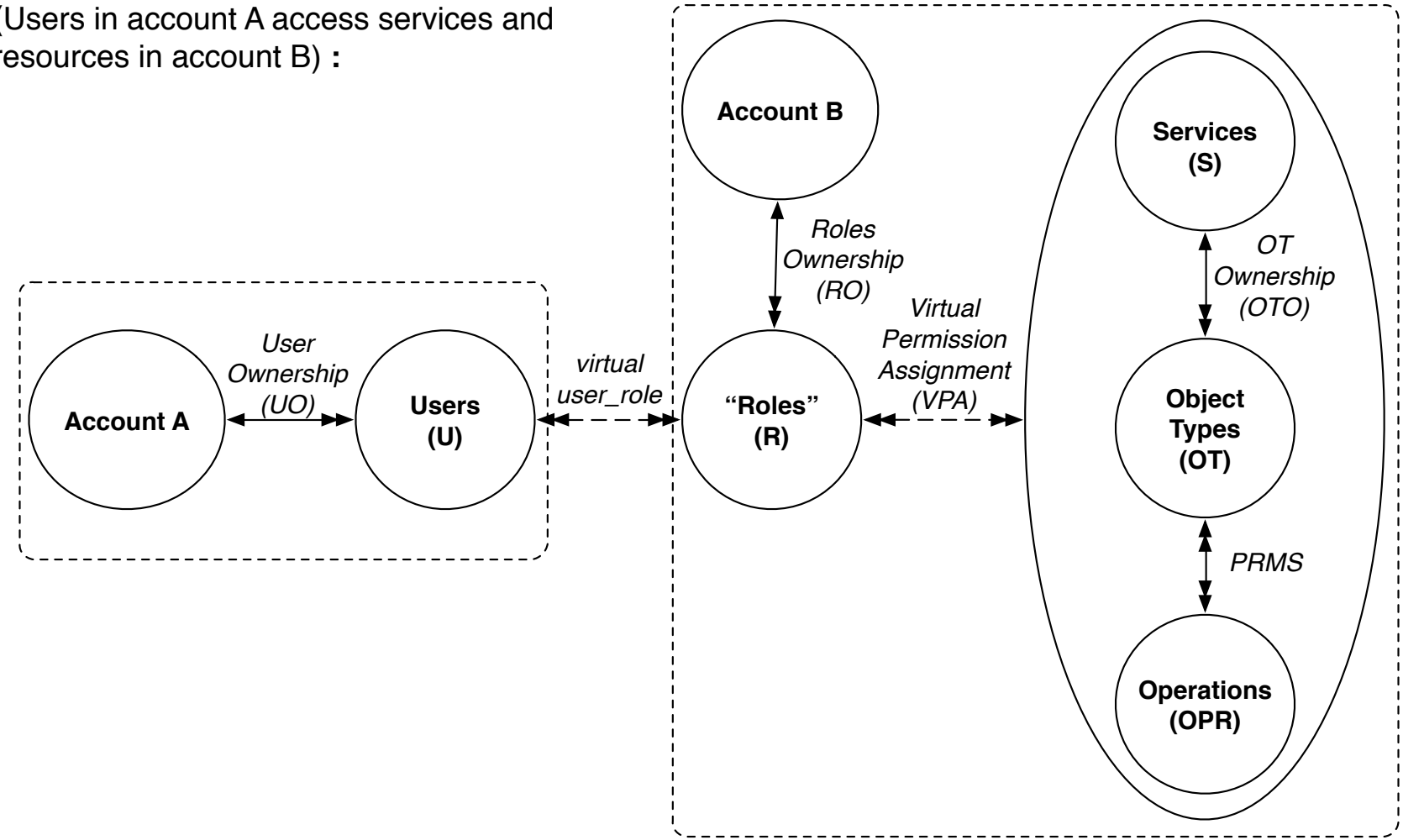
AWS Access Control within a Single Account:



Reference: Yun (Amy) Zhang, Farhan Patwa and Ravi Sandhu, Community-Based Secure Information and Resource Sharing in AWS Public Cloud. In Proceedings of the 1st IEEE International Conference on Collaboration and Internet Computing (CIC), 2015.

AWS Access Control (AWS-AC) Model

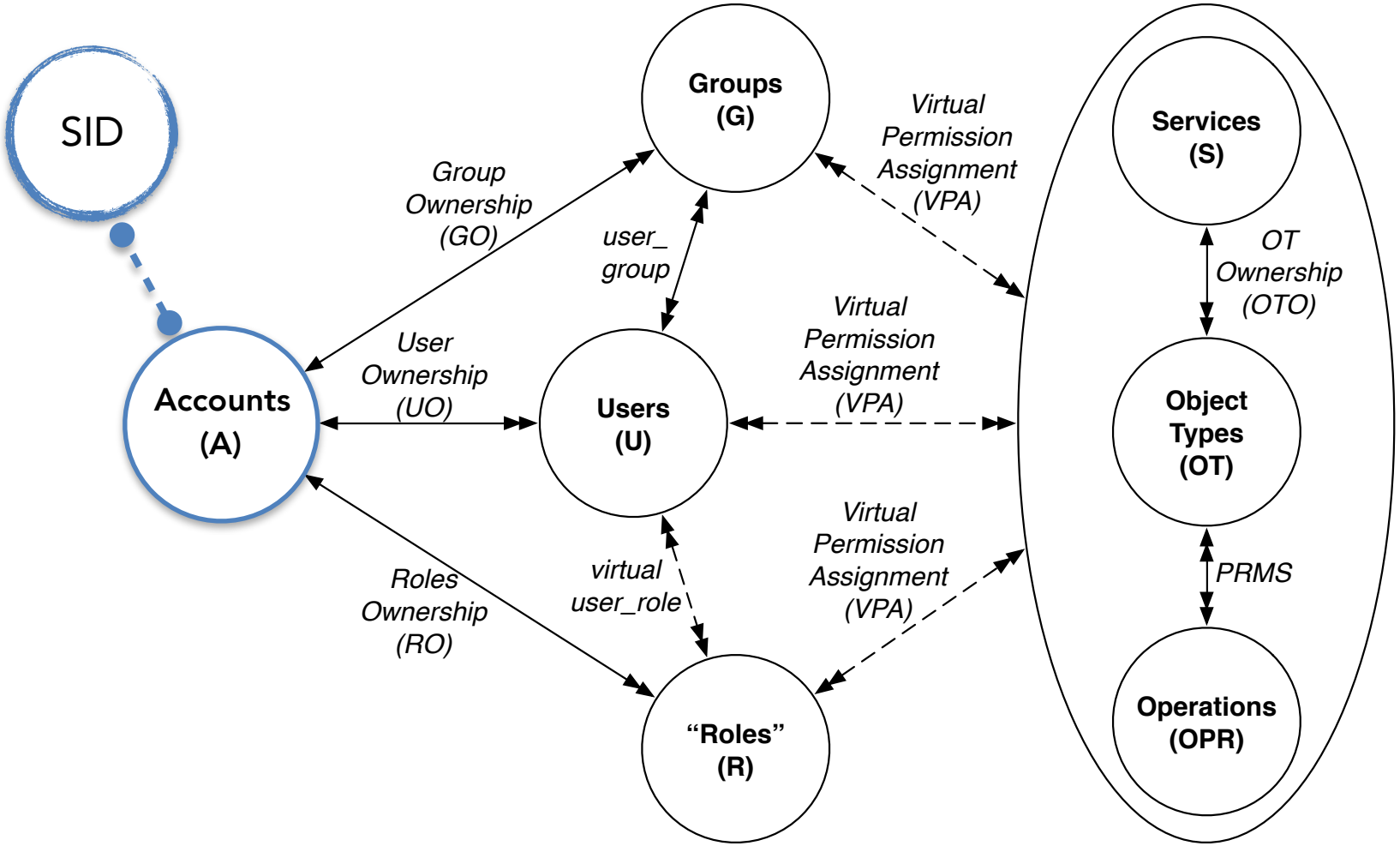
AWS Access Control across Accounts
(Users in account A access services and resources in account B) **:**



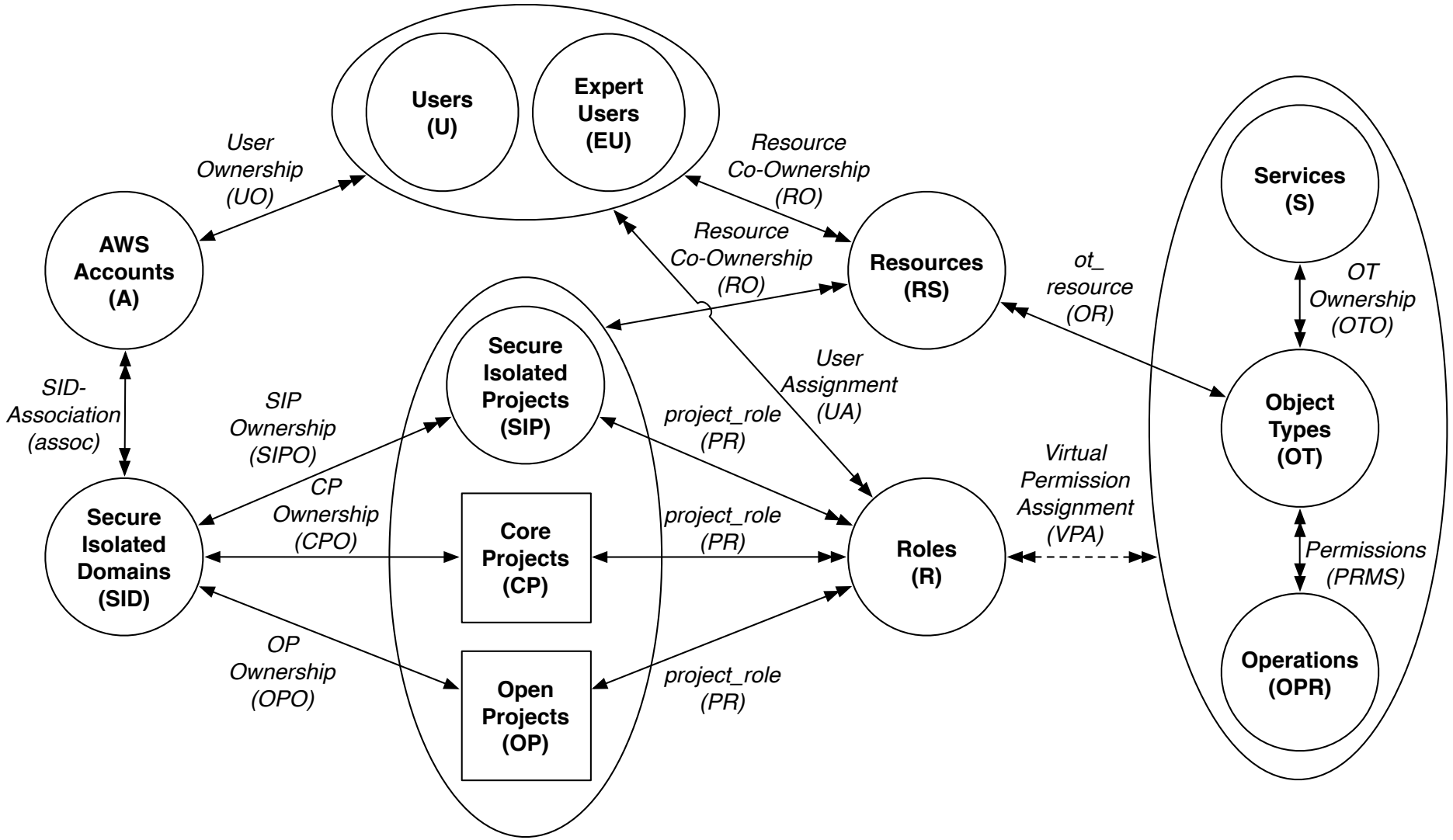
Reference: Yun (Amy) Zhang, Farhan Patwa and Ravi Sandhu, Community-Based Secure Information and Resource Sharing in AWS Public Cloud. In Proceedings of the 1st IEEE International Conference on Collaboration and Internet Computing (CIC), 2015.

AWS-AC Model extends with SID

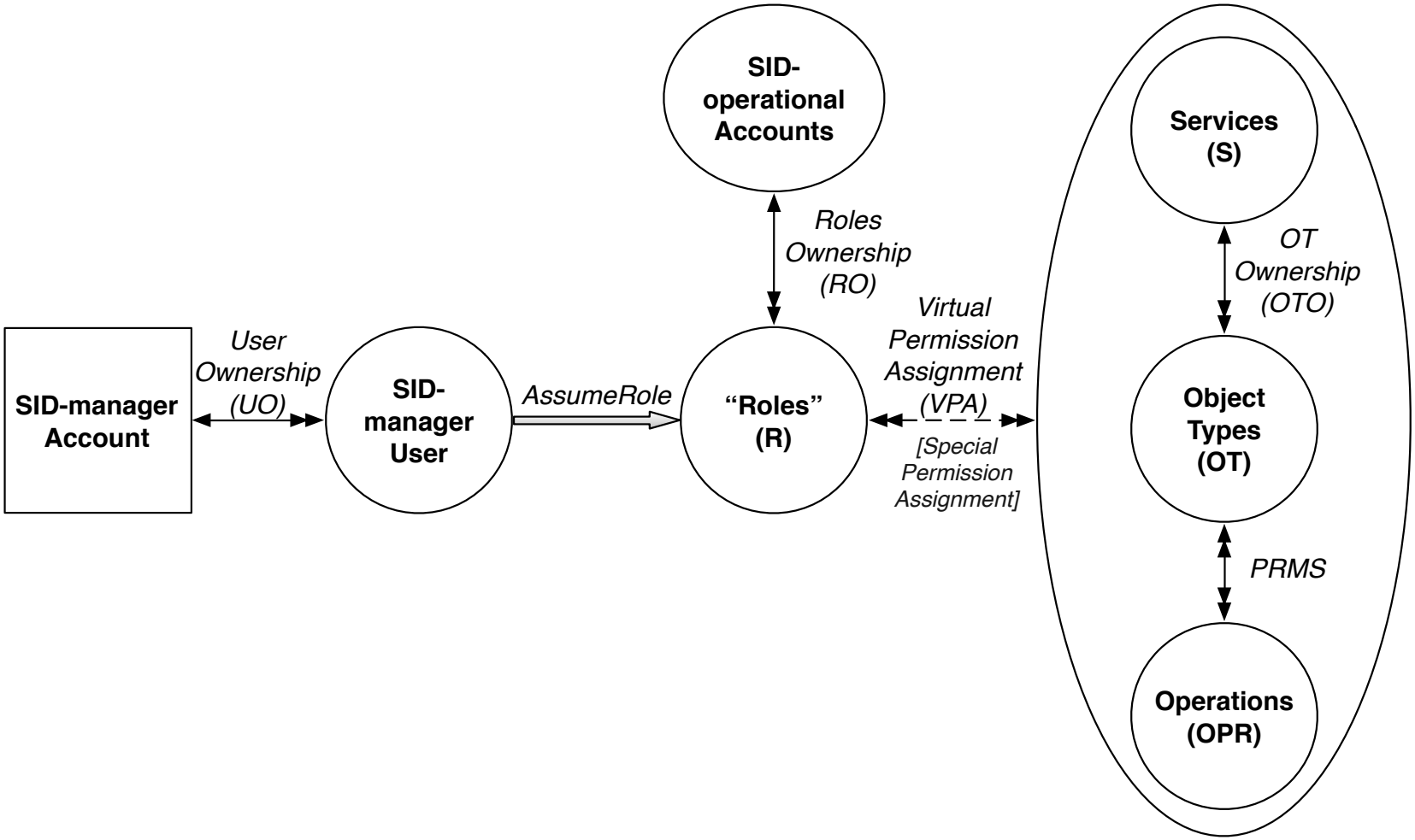
AWS Access Control within a Single Account:



AWS-AC-SID Model



Enforcement - Setup SID-service



Enforcement DataBase

sid_id	sid_name	sid_members	core_project	open_project
wbxiA97YH4c8jQARrGs1g7hkCjpHIKbu	Sid1	{"SAWS": "042298307144", "CPS": "934324332443"}	401991328752	434230153961

1 row in set (0.00 sec)

1 row in set (0.00 sec)

SIDs Table in Database

sip_account_id	account_name	sip_members	status	sid_id
401991328752	Sid1_cp	{"SAWS": "042298307144", "CPS": "934324332443"}	1	j3molQAxgAn3jCayTFZLsi5IchTf9C1w
434230153961	Sid1_op	{"SAWS": "042298307144", "CPS": "934324332443"}	1	j3molQAxgAn3jCayTFZLsi5IchTf9C1w
557554226495	Sip1	{"SAWS": "042298307144", "CPS": "934324332443"}	1	j3molQAxgAn3jCayTFZLsi5IchTf9C1w
652714115935		{}	0	

4 rows in set (0.00 sec)

SIPs Table in Database

UTSA

I·C·S The Institute for Cyber Security

Enforcement - Policy

Core Project Admin User Policy:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowSecAdminToListRolesUsers",
            "Effect": "Allow",
            "Action": [
                "iam:ListRoles",
                "iam:ListUsers",
                "iam:ListPolicies",
                "iam:GetPolicy"
            ],
            "Resource": [
                "arn:aws:iam::*"
            ]
        }
    ]
}
```

Core Project Member User Policy:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:*",
            "Resource": "*"
        }
    ]
}
```

Enforcement - Across-account Access

User in org1 accesses resources in sid1:

AssumeRole in Org1:

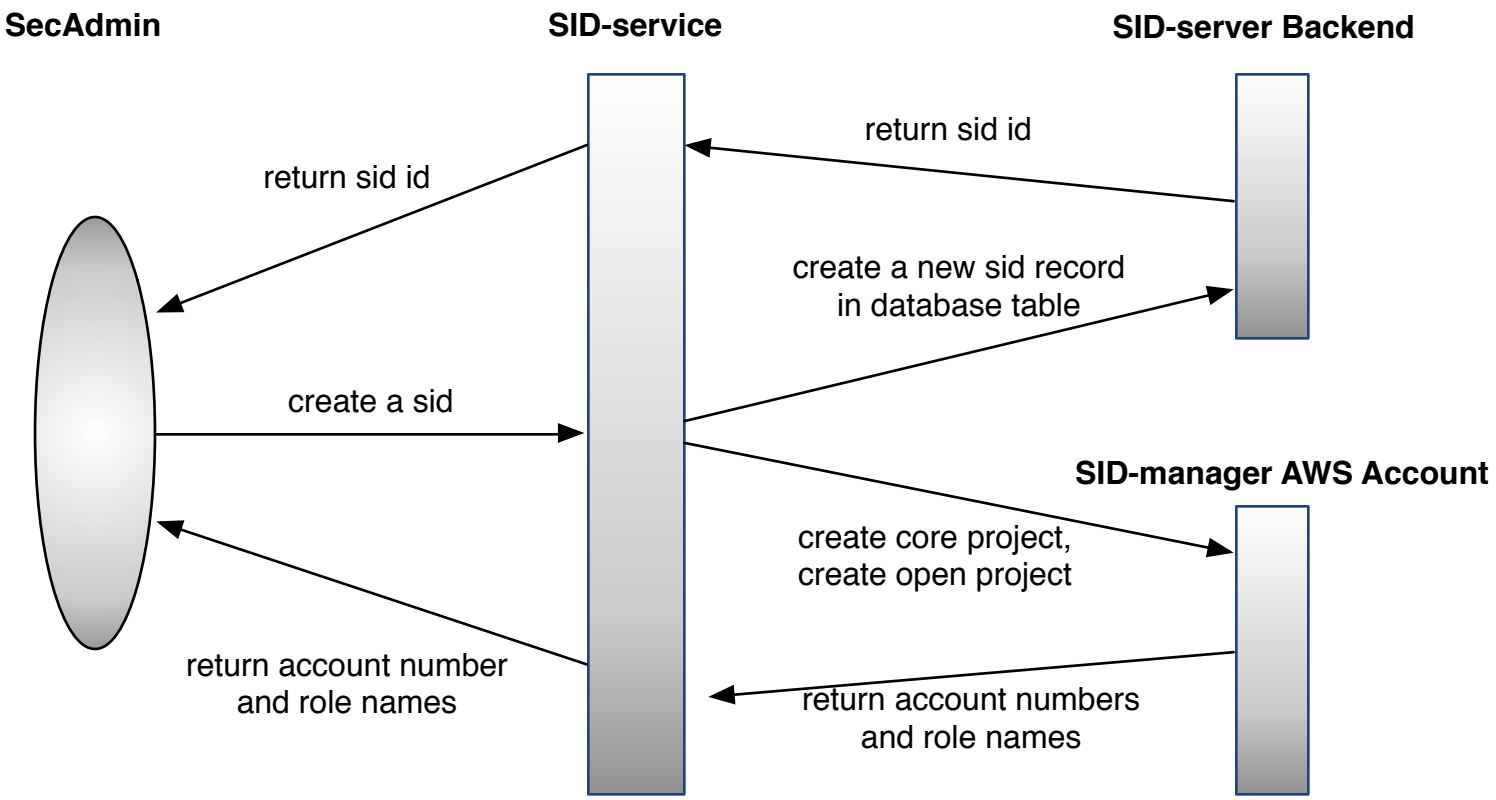
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Resource": "arn:aws:iam::*:*"
        }
    ]
}
```

Trust relationship in a Role in sid1:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123412341234:user/SipAdmin"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

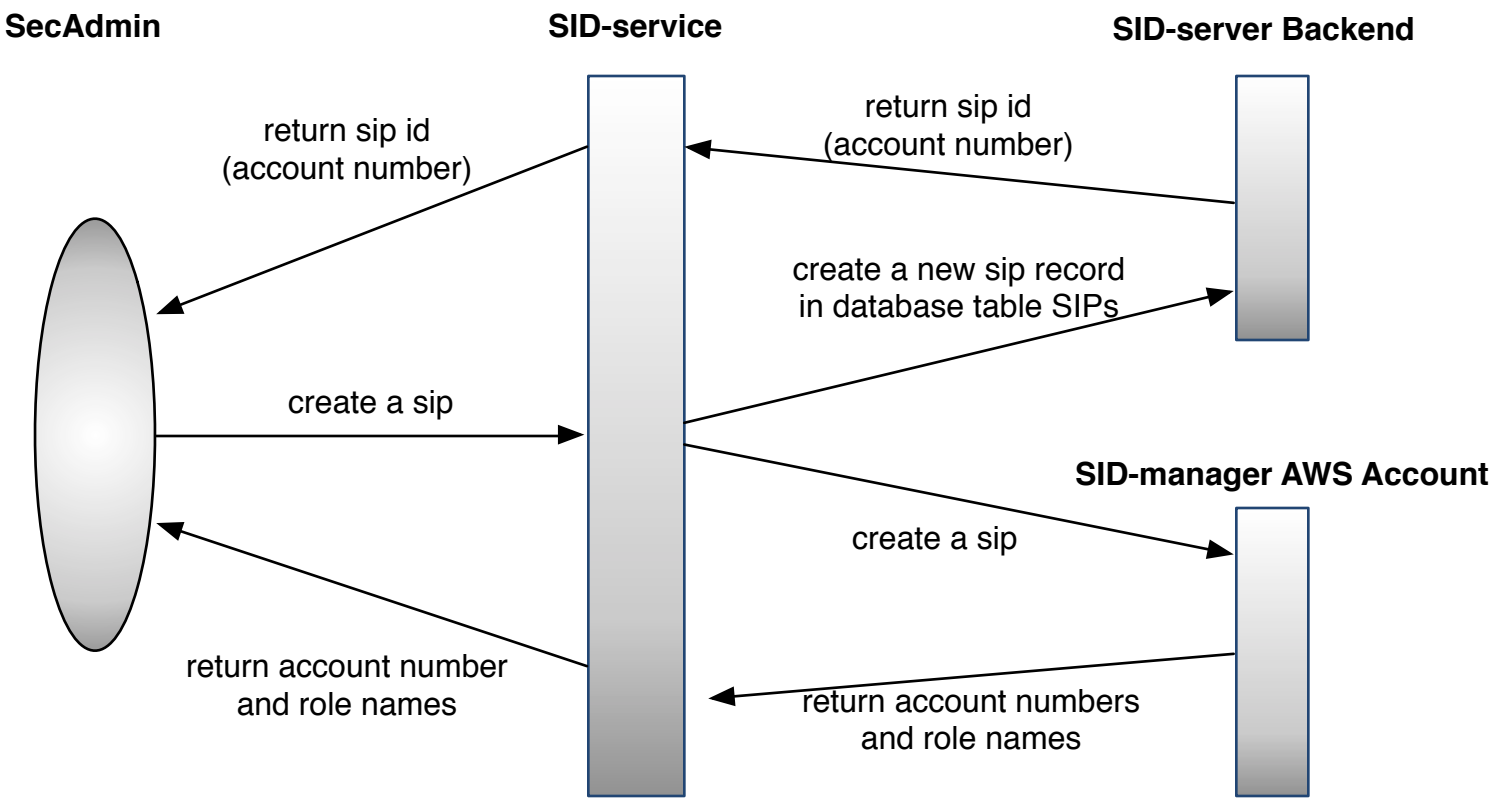
Enforcement

Process of Creating a Sid:

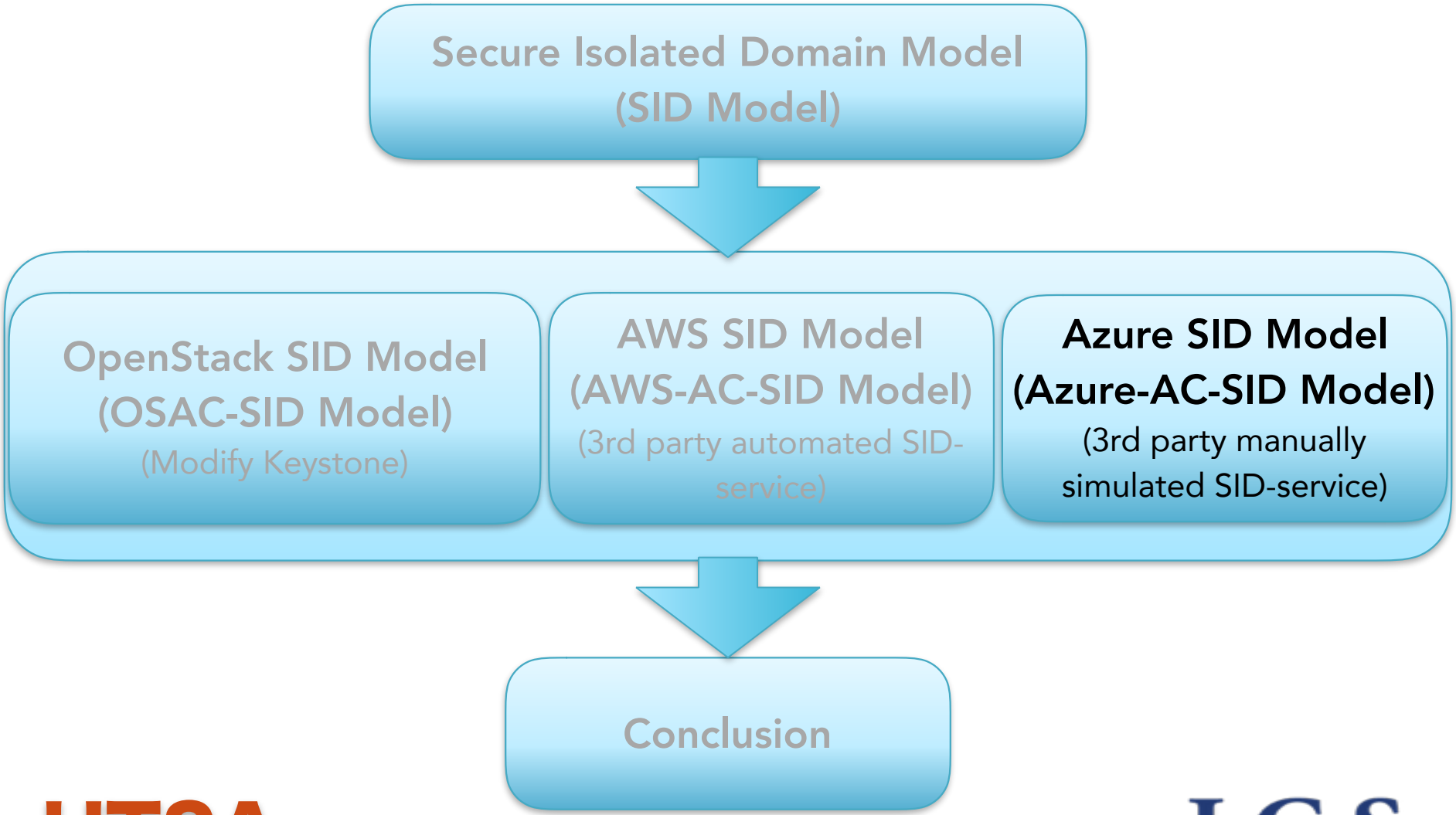


Enforcement

Process of Creating a Sip:

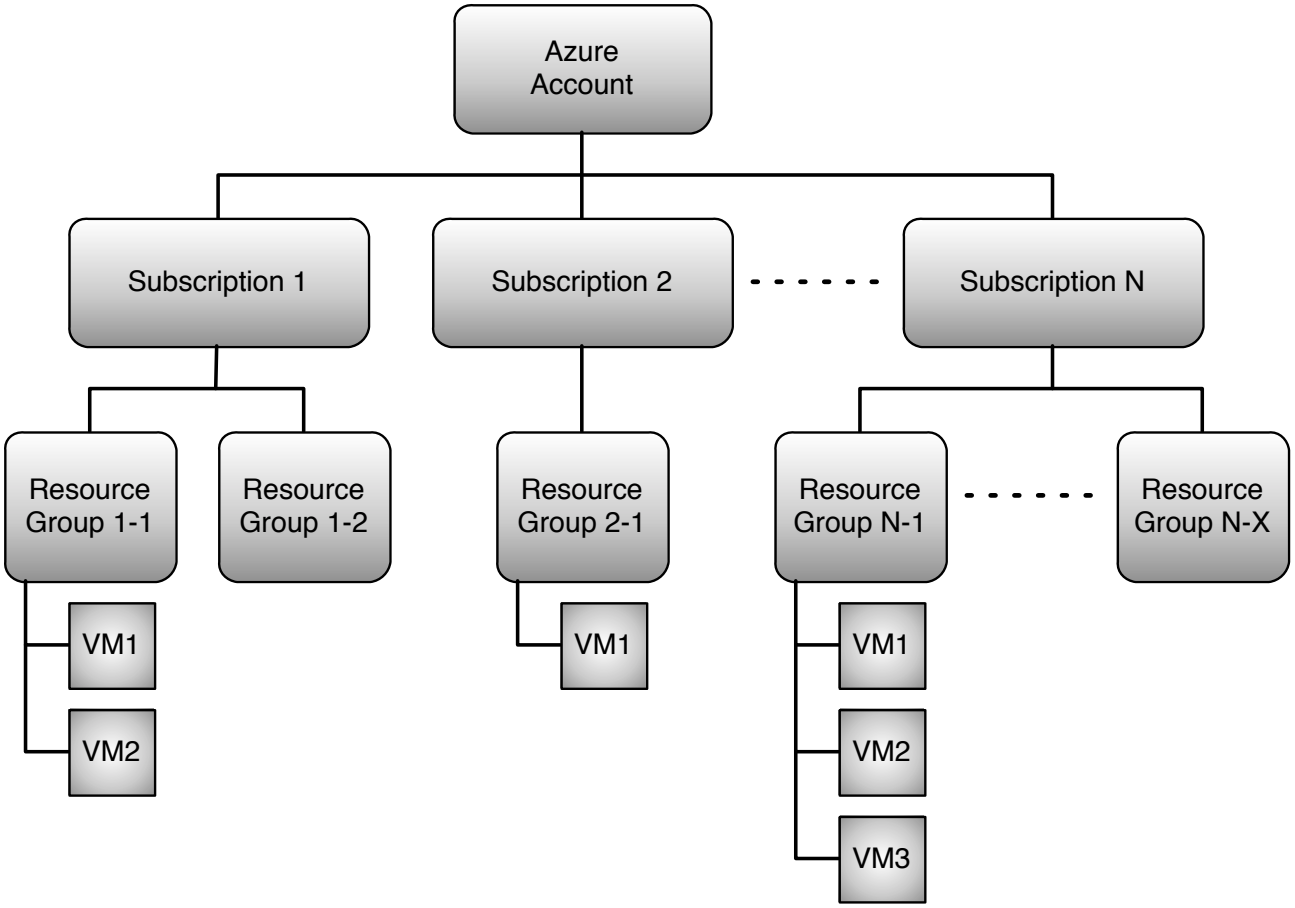


Outline

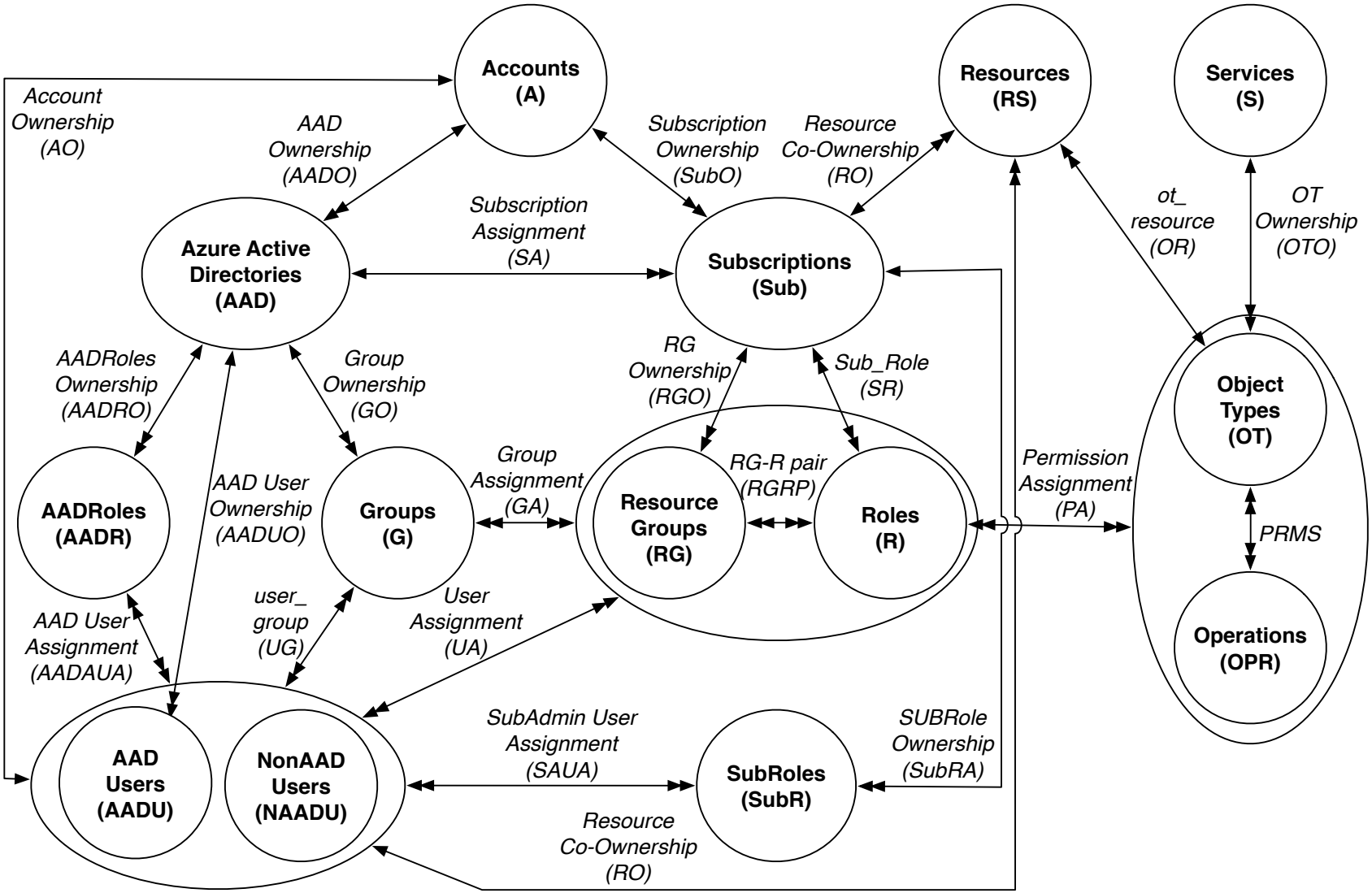

Secure Isolated Domain Model
(SID Model)
OpenStack SID Model
(OSAC-SID Model)
(Modify Keystone)
AWS SID Model
(AWS-AC-SID Model)
(3rd party automated SID-service)
Azure SID Model
(Azure-AC-SID Model)
(3rd party manually simulated SID-service)
Conclusion

Azure Introduction

- Azure Account Resource Division:

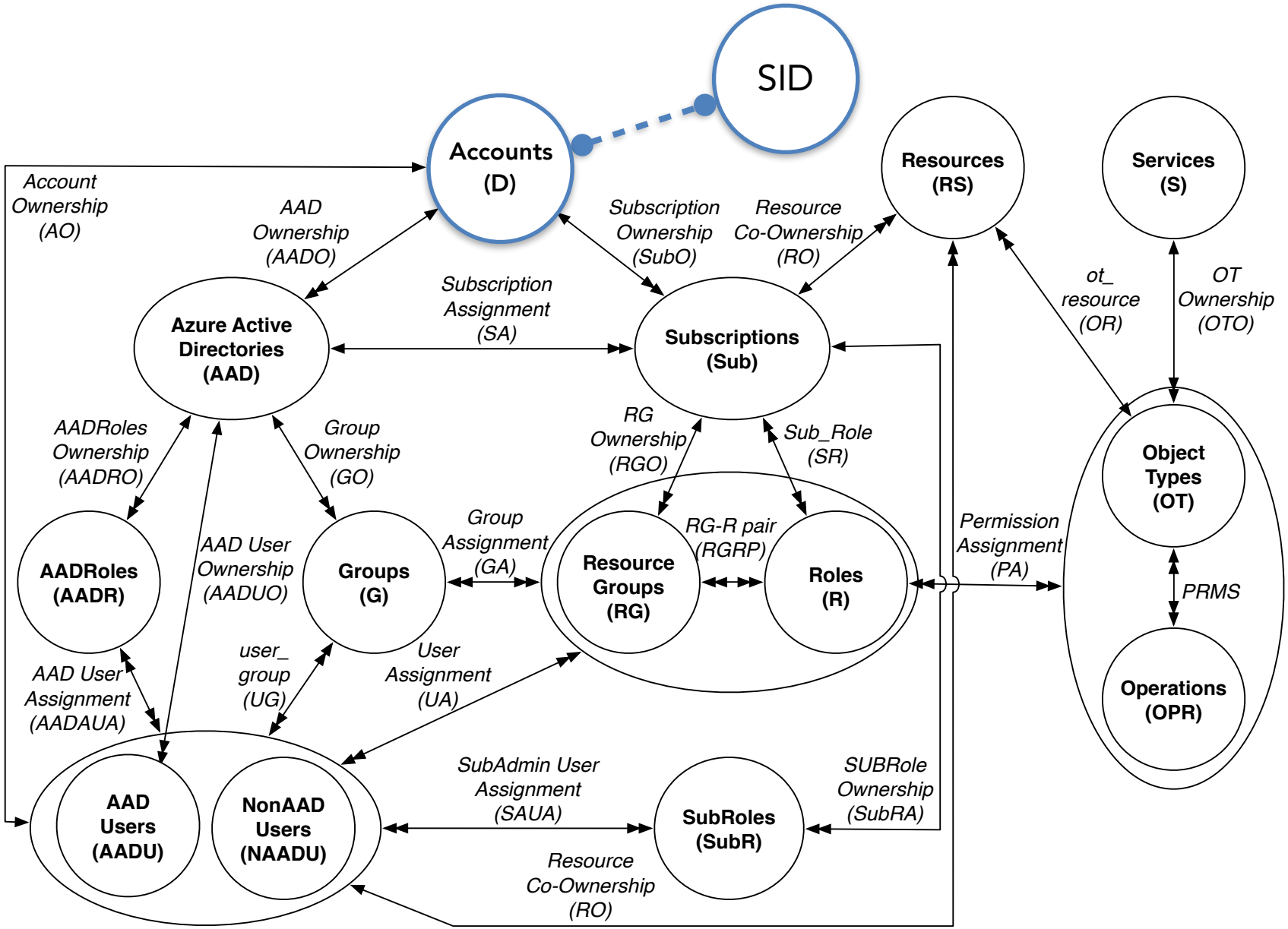


Azure Access Control (Azure-AC) Model

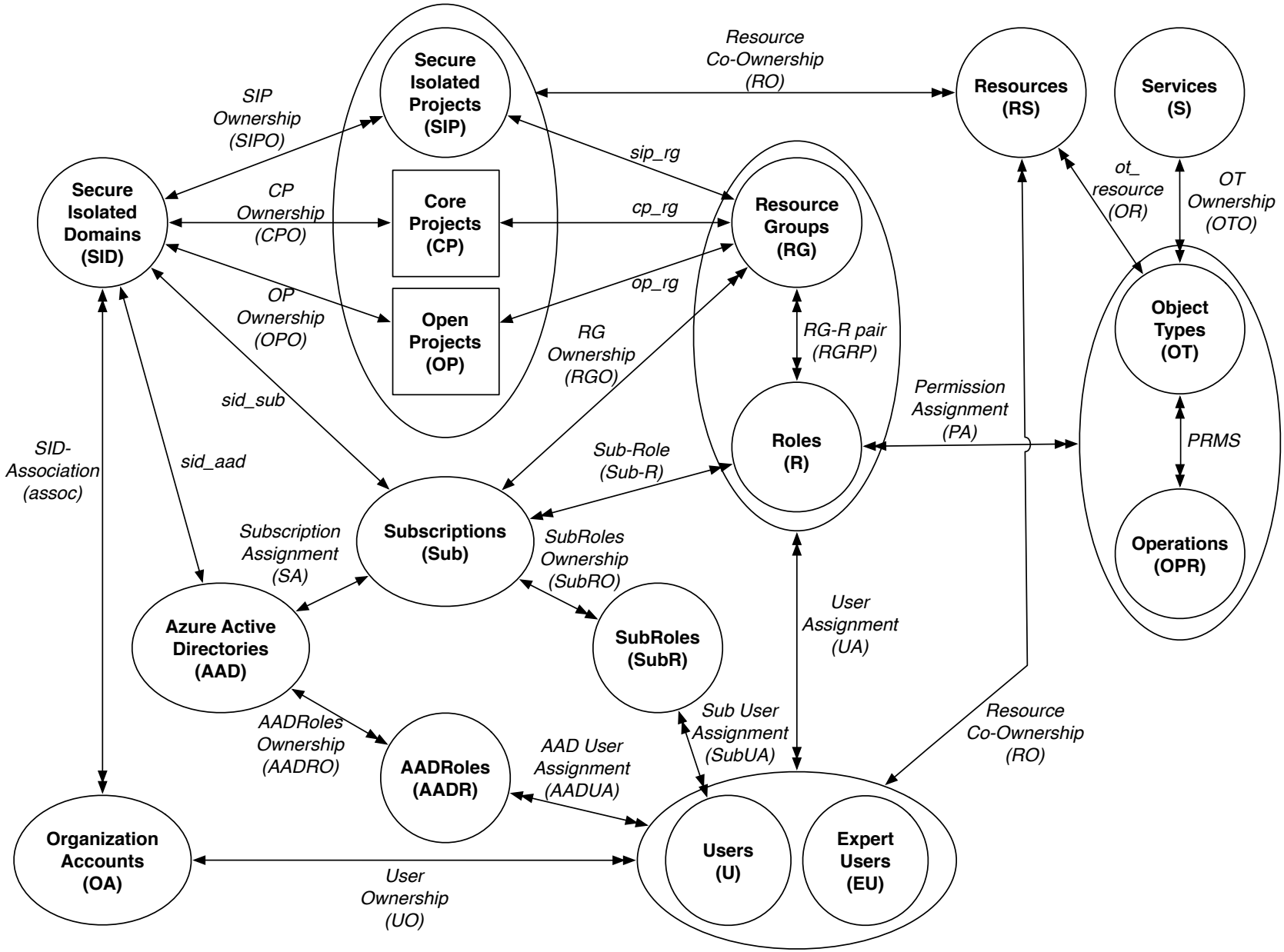


Reference: Yun (Amy) Zhang, Farhan Patwa and Ravi Sandhu, Community-Based Secure Information and Resource Sharing in Azure Cloud IaaS. In Proceedings of the 4th ACM Workshop on Security in Cloud Computing (SCC), 2016.

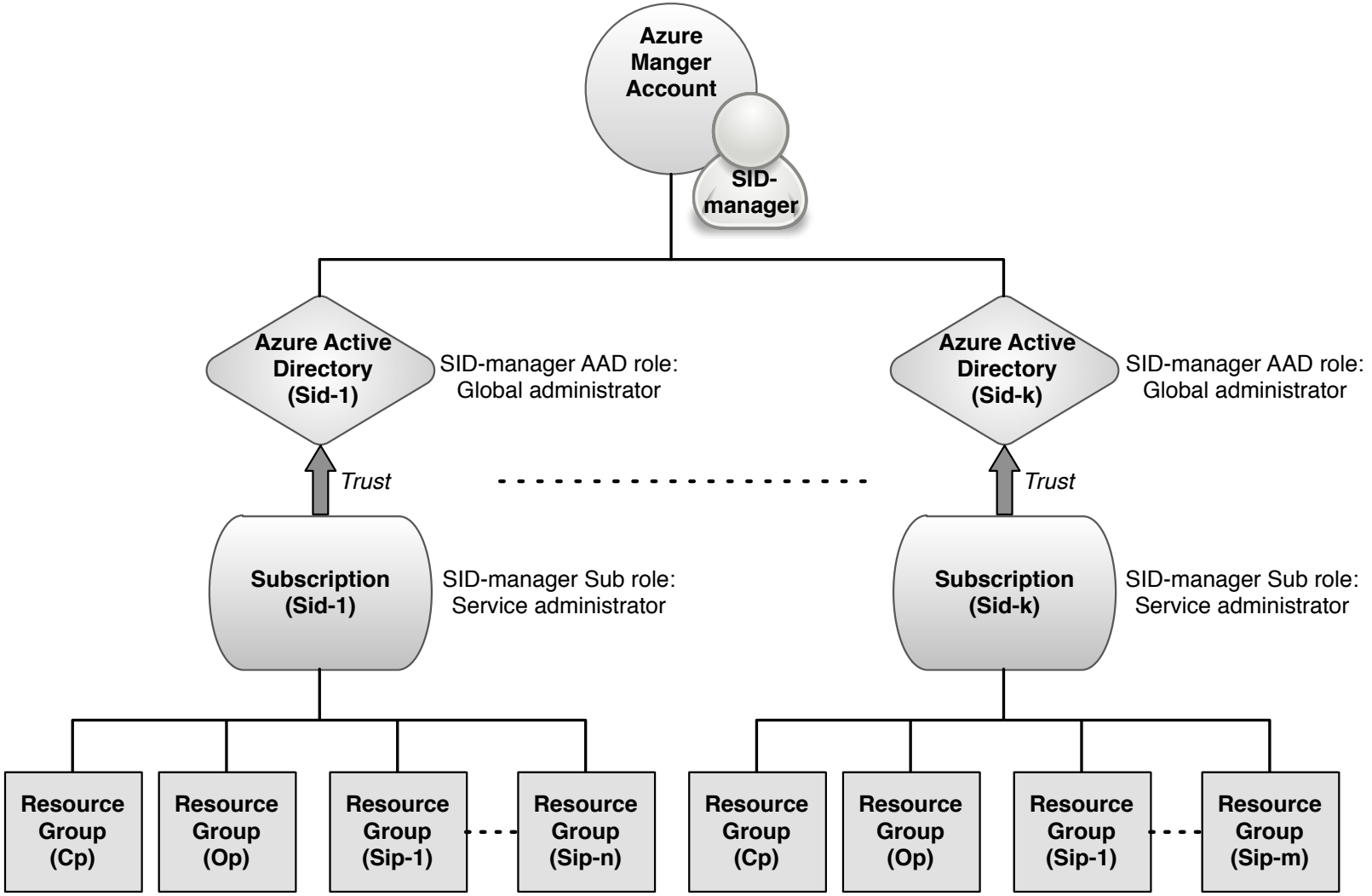
Azure-AC Model extends with SID



Azure-AC-SID Model

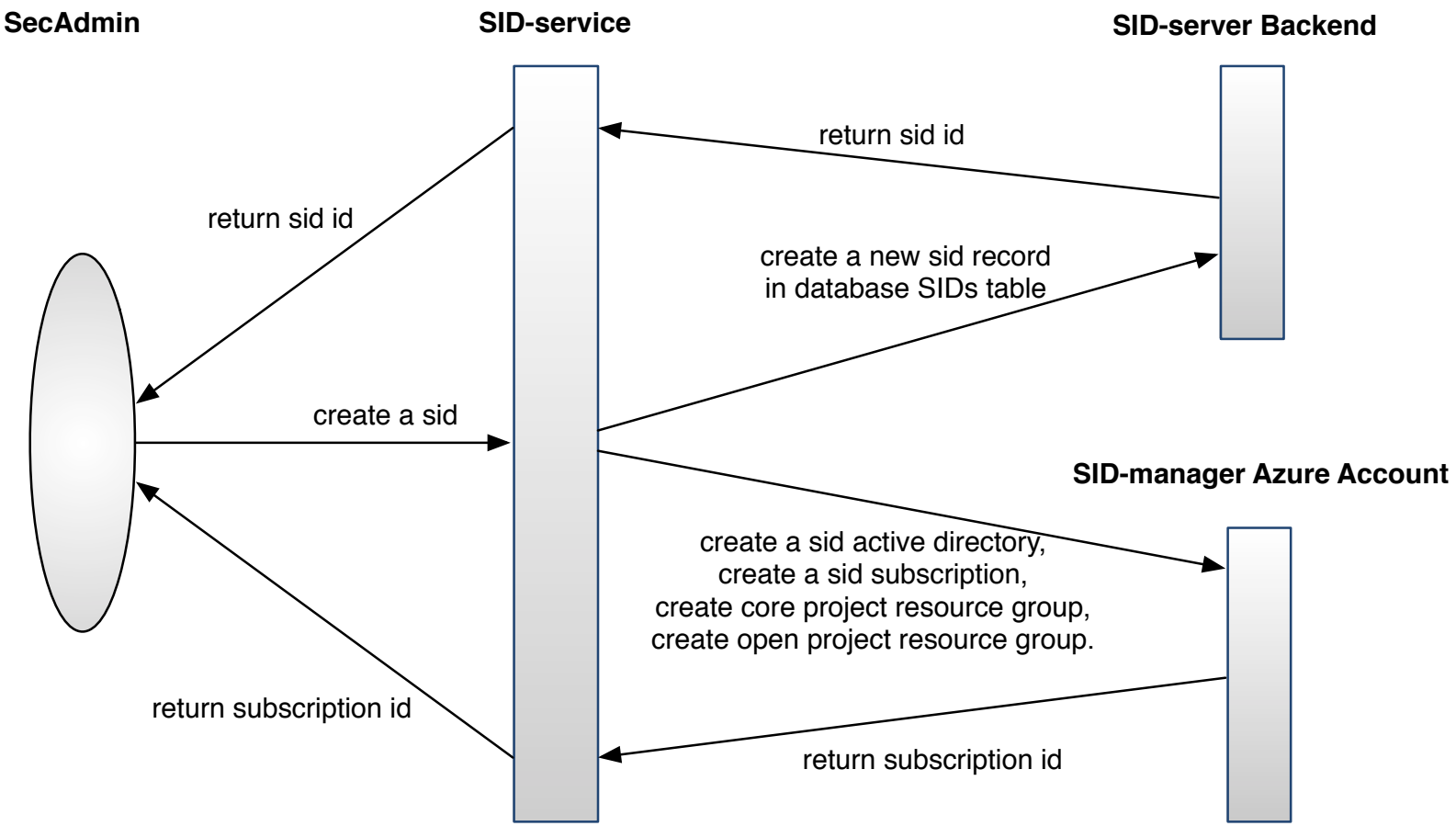


Enforcement



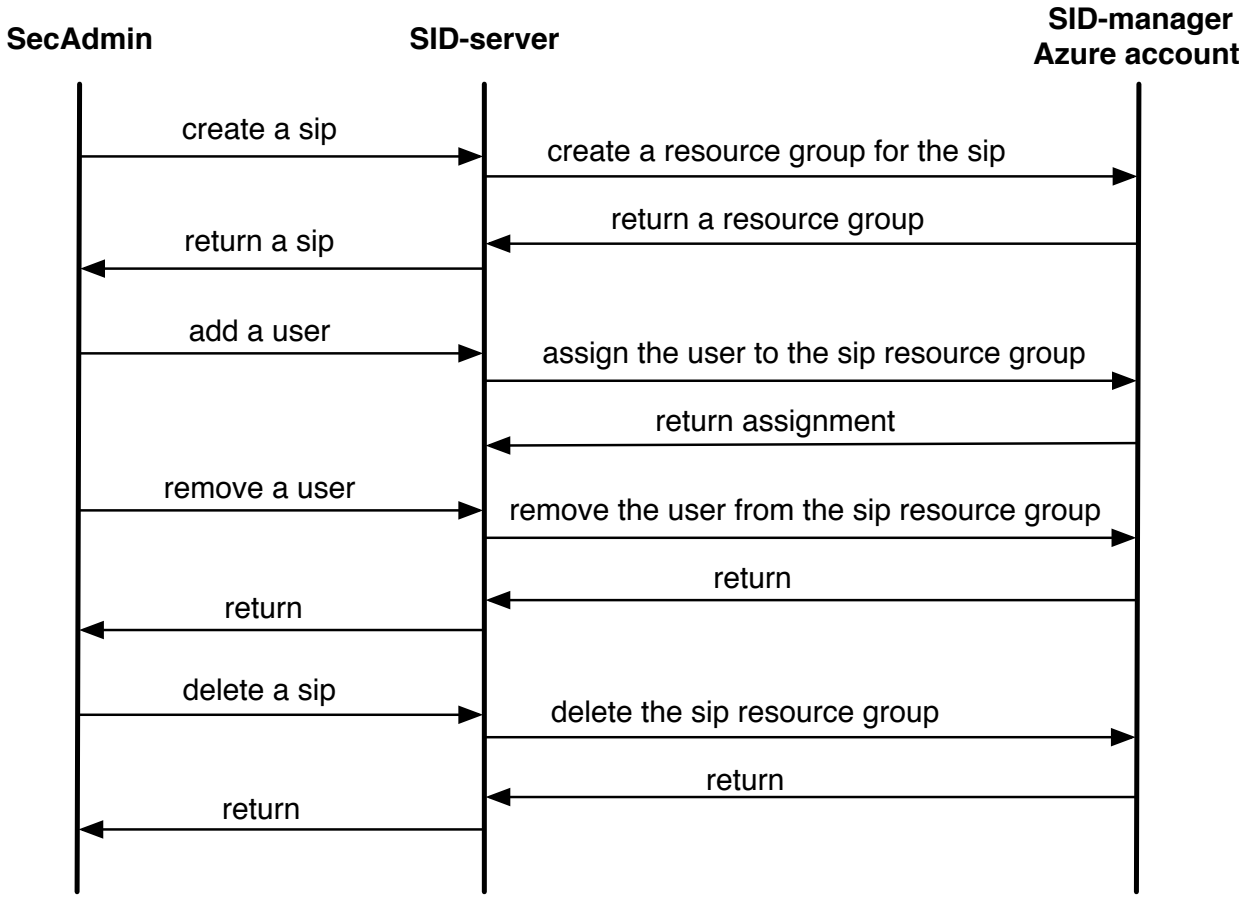
Enforcement

- Create a Sid:

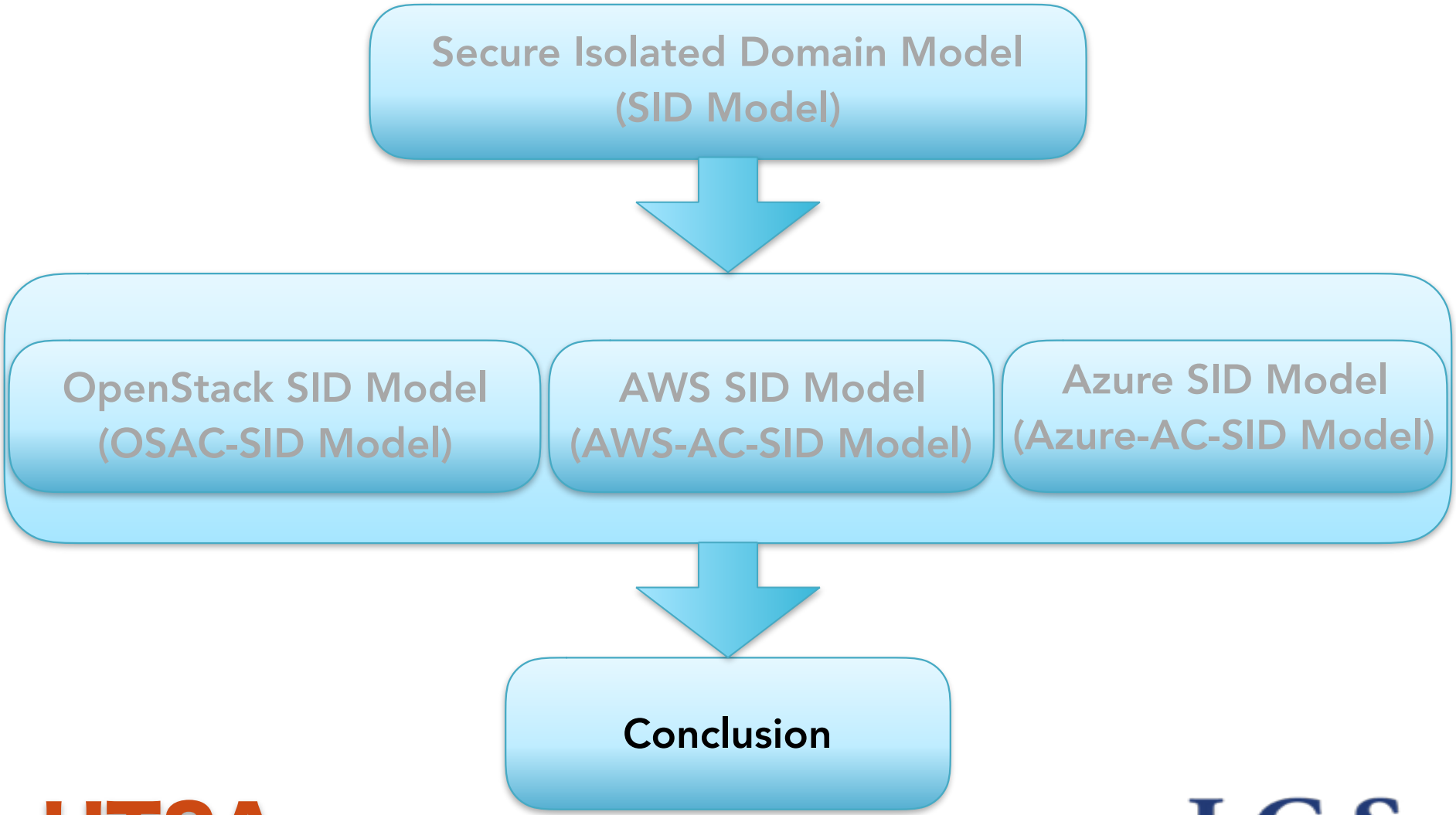


Enforcement

- SIP-requests:

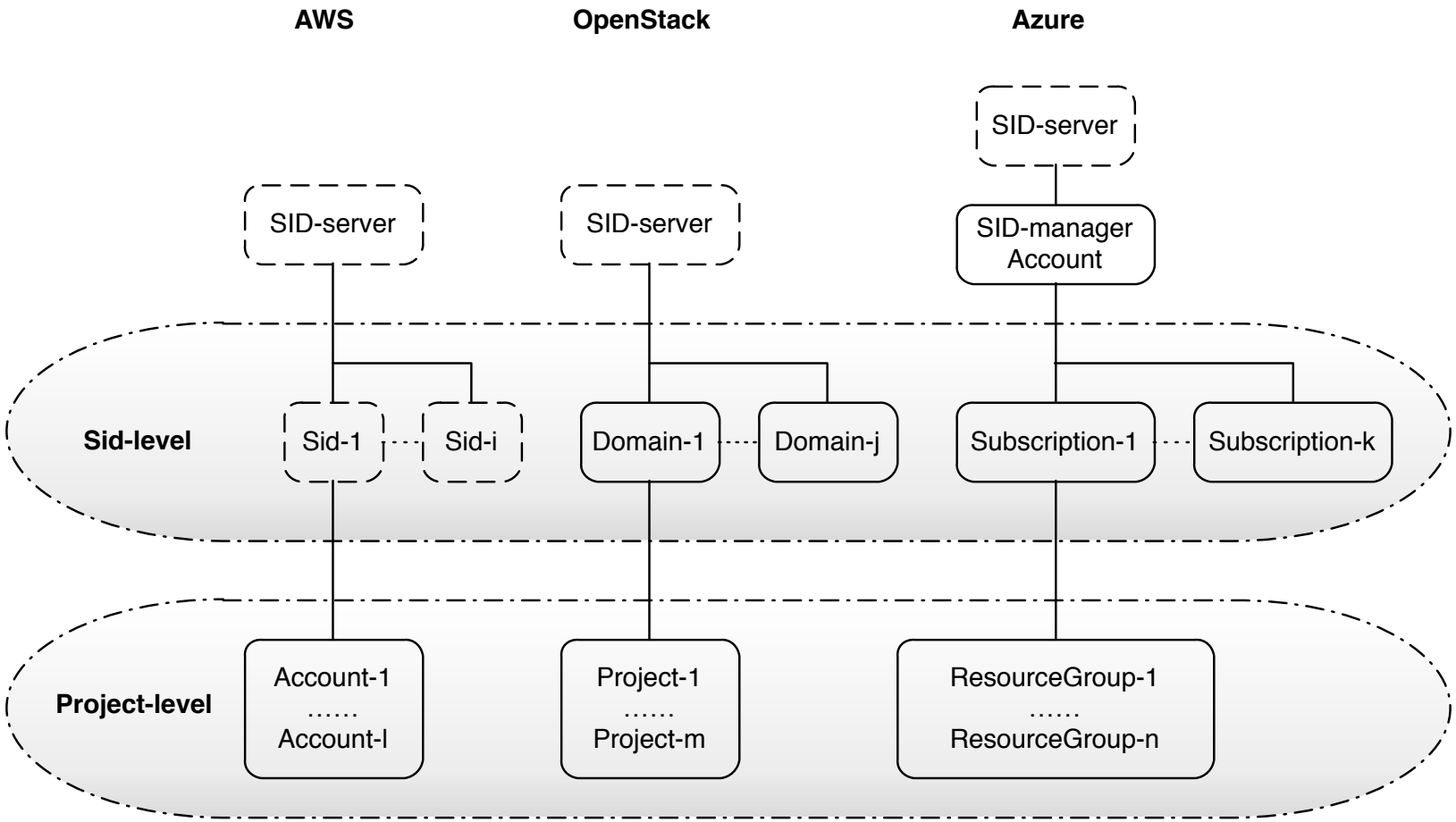


Outline


Secure Isolated Domain Model
(SID Model)
OpenStack SID Model
(OSAC-SID Model)
AWS SID Model
(AWS-AC-SID Model)
Azure SID Model
(Azure-AC-SID Model)
Conclusion

Model Comparison

- Resource Containers:



Model Comparison

- SID-services:
  - OpenStack
    - Modify cloud system itself
  - AWS & Azure
    - Build a third party SID-service server
- Roles:
  - OpenStack
    - Global roles
  - AWS
    - Local roles, with trust relations
  - Azure
    - Pre-defined roles & customized roles

Conclusion and future work

- Developed a sharing model (SID-model)
  - Formal specification
- Applied the model to three dominant IaaS cloud platforms (OpenStack, AWS and Azure)
  - Defined access control models for each cloud system
  - Extend it with SID
  - Enforcement
- Compare SID-model in different cloud system
- Future work suggestions
  - Try more control on a group of organizations creating a sid/sip;
  - Try more fine-grained roles inside a sid/sip;
  - Apply the model to multi-clouds.

Publications

- Yun (Amy) Zhang, Farhan Patwa and Ravi Sandhu, Community-Based Secure Information and Resource Sharing in Azure Cloud IaaS. *In Proceedings of the 4th ACM Workshop on Security in Cloud Computing (SCC)*, May 30, 2016, Xi'an, China, 8 pages.
- Yun (Amy) Zhang, Farhan Patwa and Ravi Sandhu, Community-Based Secure Information and Resource Sharing in AWS Public Cloud. *In Proceedings of the 1st IEEE International Conference on Collaboration and Internet Computing (CIC)*, Hangzhou, China, October 27-30, 2015, 8 pages.
- Yun (Amy) Zhang, Farhan Patwa, Ravi Sandhu and Bo Tang, Hierarchical Secure Information and Resource Sharing in OpenStack Community Cloud. *In Proceedings 16th IEEE Conference on Information Reuse and Integration (IRI)*, San Francisco, California, August 13-15, 2015, 8 pages.
- Yun (Amy) Zhang, Ram Krishnan and Ravi Sandhu, Secure Information and Resource Sharing in Cloud. *In Proceedings of the 5th ACM Conference on Data and Application Security and Privacy (CODASPY) Poster Session*, March 2-4, 2015, San Antonio, Texas, pages 131-133.
- Yun (Amy) Zhang, Ram Krishnan and Ravi Sandhu, Secure Information and Resource Sharing in Cloud Infrastructure as a Service. *In Proceedings of ACM Workshop on Information Sharing and Collaborative Security (WISCS 2014),* November 3, 2014, Scottsdale, AZ.


Thanks!