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
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

Adapted from Dr. Ram Krishnan's dissertation defense:
Group-Centric Secure Information Sharing Models, Fall 2009
Sharing Model in Cloud IaaS

Background/Related Work

• Community Cyber Incident Response

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
Secure Isolated Domain Model (SID Model)

- SID Concept
- SID Model Formalization
  - Administrative Model
  - Operational Model
SID Concept

- Secure Isolated Domain (SID)

- Core Project (CP)
- Open Project (OP)
- Secure Isolated Project-1 (SIP-1)
- Secure Isolated Project-n (SIP-n)
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)

SIP Ownership (SIPO)

CP Ownership (CPO)

OP Ownership (OPO)

Secure Isolated Domains (SID)

Users (U)

Expert Users (EU)

Secure Isolated Projects (SIP)

Core Projects (CP)

Open Projects (OP)

Secure Accounts (SA)

Resources (RS)

Roles (R)

Object Types (OT)

Operations (OPR)

Permissions (PRMS)

One-to-one relation:
One-to-multiple relation:
Multiple-to-multiple relation:
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

<table>
<thead>
<tr>
<th>Operation</th>
<th>Authorization Requirement</th>
<th>Update</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SidCreate</strong></td>
<td>adminu ∈ uSet ∧ adminu ∈ U ∧ sid ∉ SID</td>
<td>SID' = SID ∪ {sid}; assoc(sid) = ∪_{adminu∈uSet} UO(adminu); CP' = CP ∪ {cp}; CPO(cp) = sid; OP' = OP ∪ {op}; OPO(op) = sid; UA' = (uSet, SIDadmin) ∪ UA; PR' = PR ∪ {(cp, SIDadmin), (op, SIDadmin)}.</td>
</tr>
<tr>
<td><strong>SidDelete</strong></td>
<td>adminu ∈ uSet ∧ adminu ∈ U ∧ (adminu, SIDadmin) ∈ UA ∧ assoc(sid) = ∪_{adminu∈uSet} UO(adminu) ∧ sid ∈ SID</td>
<td>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 ∃ u ∈ (U ∪ EU).((u, SIDmember) ∈ UA), then UA' = UA - (u, SIDmember); if ∃ sip ∈ SIP.(SIPO(sip) = sid), then SIP' = SIP - sip ∧ PR' = PR - {(sip, SIDadmin), (sip, SIDmember)}.</td>
</tr>
</tbody>
</table>
## SID Administrative Model formalization

<table>
<thead>
<tr>
<th>Operation</th>
<th>Authorization Requirement</th>
<th>Update</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SipCreate</strong>&lt;br&gt;adminu, sip, sid&lt;br&gt;/* An admin user representing uSet creates a sip */</td>
<td>adminu ∈ U ∧ (adminu, SIDadmin) ∈ UA ∧ UO(adminu) ∈ assoc(sid) ∧ sip ∉ SIP</td>
<td>SIP' = SIP ∪ {sip}; PR' = PR ∪ {(sip, SIDadmin)}.</td>
</tr>
<tr>
<td><strong>SipDelete</strong>&lt;br&gt;adminu, sip, sid&lt;br&gt;/* An admin user representing uSet deletes a sip */</td>
<td>adminu ∈ U ∧ (adminu, SIDadmin) ∈ UA ∧ UO(adminu) ∈ assoc(sid) ∧ SIPO(sip) = sid</td>
<td>SIP' = SIP - {sip}; PR' = PR - {(sip, SIDadmin)}.</td>
</tr>
<tr>
<td><strong>UserAdd</strong>&lt;br&gt;adminu, u, p, sid&lt;br&gt;/* Admin users add a user from his home domain to a cp, op or sip */</td>
<td>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)</td>
<td>UA' = UA ∪ {(u, SIDmember)}.</td>
</tr>
<tr>
<td><strong>UserRemove</strong>&lt;br&gt;adminu, u, p, sid&lt;br&gt;/* Admin users remove a user from a cp, op or sip */</td>
<td>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</td>
<td>UA' = UA - {(u, SIDmember)}.</td>
</tr>
<tr>
<td><strong>EUUserAdd</strong>&lt;br&gt;adminu, eu, p, sid&lt;br&gt;/* Admin users add an expert user to a cp or sip */</td>
<td>adminu ∈ U ∧ (adminu, SIDadmin) ∈ UA ∧ (p, SIDadmin) ∈ PR ∧ eu ∈ EU ∧ p ∈ (CP ∪ SIP) ∧ (CPO(p) = sid ∨ SIPO(p) = sid)</td>
<td>UA' = UA ∪ {(eu, SIDmember)}.</td>
</tr>
<tr>
<td><strong>EUUserRemove</strong>&lt;br&gt;adminu, eu, p, sid&lt;br&gt;/* Admin users remove an expert user from a cp or sip */</td>
<td>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</td>
<td>UA' = UA - {(eu, SIDmember)}.</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Operation</th>
<th>Authorization Requirement</th>
<th>Update</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CreateVM(vm, p, u)</strong> /* A user creates a vm */</td>
<td>$vm \not\in RS \land p \in (CP \cup OP \cup SIP) \land u \in U \land \exists (perms, r) \in PA. (perms = (vm, create) \land (p, r) \in PR \land (u, (p, r)) \in UA)$</td>
<td>$RS' = RS \cup {vm}$; $RSO' = RSO \cup {(vm, (p, u))}$; $OR(vm) = VM$.</td>
</tr>
<tr>
<td><strong>DeleteVM(vm, p, u)</strong> /* A user deletes a vm */</td>
<td>$vm \in RS \land RSO(vm) = {(p, u)} \land p \in (CP \cup OP \cup SIP) \land u \in U \land \exists (perms, r) \in PA. (perms = (vm, delete) \land (p, r) \in PR \land (u, (p, r)) \in UA)$</td>
<td>$RS' = RS - {vm}$; $RSO' = RSO - {(vm, (p, u))}$; $vm = NULL$.</td>
</tr>
<tr>
<td><strong>CreateSContainer(sc, p, u)</strong> /* A user creates a storage container */</td>
<td>$sc \not\in RS \land p \in (CP \cup OP \cup SIP) \land u \in U \land \exists (perms, r) \in PA. (perms = (sc, create) \land (p, r) \in PR \land (u, (p, r)) \in UA)$</td>
<td>$RS' = RS \cup {sc}$; $RSO' = RSO \cup {(sc, (p, u))}$; $OR(sc) = SC$.</td>
</tr>
<tr>
<td><strong>DeleteSContainer(sc, p, u)</strong> /* A user deletes a storage container */</td>
<td>$sc \in RS \land RSO(sc) = {(p, u)} \land p \in (CP \cup OP \cup SIP) \land u \in U \land \exists (perms, r) \in PA. (perms = (sc, delete) \land (p, r) \in PR \land (u, (p, r)) \in UA)$</td>
<td>$RS' = RS - {sc}$; $RSO' = RSO - {(sc, (p, u))}$; $sc = NULL$.</td>
</tr>
<tr>
<td><strong>CreateObject(co, sc, p, u)</strong> /* A user creates a storage container object */</td>
<td>$co \not\in RS \land sc \in RS \land p \in (CP \cup OP \cup SIP) \land u \in U \land RSO(sc) = {(p, u)} \land \exists (perms, r) \in PA. (perms = (co, create) \land (p, r) \in PR \land (u, (p, r)) \in UA)$</td>
<td>$RS' = RS \cup {co}$; $RSO' = RSO \cup {(co, (p, u))}$; $OR(co) = CO$.</td>
</tr>
<tr>
<td><strong>DeleteObject(co, sc, p, u)</strong> /* A user delete a storage container object */</td>
<td>$co \in RS \land RSO(co) = {(p, u)} \land sc \in RS \land p \in (CP \cup OP \cup SIP) \land u \in U \land RSO(sc) = {(p, u)} \land \exists (perms, r) \in PA. (perms = (co, create) \land (p, r) \in PR \land (u, (p, r)) \in UA)$</td>
<td>$RS' = RS - {co}$; $RSO' = RSO - {(co, (p, u))}$; $co = NULL$.</td>
</tr>
</tbody>
</table>
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
OSAC Model extends with SID

Users (U) → Tokens (T)
- user_token

Groups (G) → Projects (P)
- Group Ownership (GO)
- Group Assignment (GA)
- User Group Ownership (UG)
- User Ownership (UO)
- User Assignment (UA)

Domains (D) → Project Ownership (PO)

Roles (R) → Project-Role Pair (PRP)

Projects (P) → Permissions Assignment (PA)

Tokens (T) →
- token_project
- token_roles

Services (S) → Object Types (OT)

Object Types (OT) → Operations (OPR)

PRMS

utsa
OSAC-SID Model

- Sid admins and cloud resource division
OSAC-SID Model
Enforcement - Keystone Protocol

Client
- Send a http request.
- Return the result.

Router
- Map the request to controller.
- Return the result.

Controller
- Find the core functions for the request.
- Return the result.

Core
- Find drive for the request.
- Return the result.

Drive
- Write data to database.
- Return the result.

Keystone Server Module

Database
Enforcement - Sid Request

1. **Client**
   - Send a http request.
   - Return the result.

2. **Router**
   - Return the result.

3. **Controller**
   - Map the request to controller.
   - Find the core functions for the request.
   - Find drive for the request.
   - Return the result.

4. **Core**
   - Return the result.

5. **Drive**
   - Return the result.
   - Write data to database.

6. **Database**
   - Return the result.

7. **Keystone Server Module**
   - Write data to database.
## Enforcement - Backend

### Sid Table in Database:

```sql
mysql> describe sid;
```

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Null</th>
<th>Key</th>
<th>Default</th>
<th>Extra</th>
</tr>
</thead>
<tbody>
<tr>
<td>sid_id</td>
<td>varchar(64)</td>
<td>NO</td>
<td>PRI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sid_name</td>
<td>varchar(64)</td>
<td>YES</td>
<td></td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>sid_members</td>
<td>text</td>
<td>YES</td>
<td></td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>core_project</td>
<td>varchar(64)</td>
<td>YES</td>
<td></td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>open_project</td>
<td>varchar(64)</td>
<td>YES</td>
<td></td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>extra</td>
<td>text</td>
<td>YES</td>
<td></td>
<td>NULL</td>
<td></td>
</tr>
</tbody>
</table>
Enforcement

An Org SecAdmin Create a Sid:

Org Sec Admin

Keystone Server

Resource Backend

Assignment Backend

Request to create a sid

Create a domain as a sid

Return a domain

Create a project as core project

Return a project

Create a project as open project

Return a project

Assign all org sec admins to the sid

Return the assignment

Assign all org sed admins to core project

Return the assignment

Create a sid

Return a sid

Return a sid
An Org SecAdmin Create a Sip:

1. Request to create a sip
2. Create a project as a sip
3. Return a project
4. Assign all org sec admins to the sip
5. Return the assignment
6. Return a sip
Enforcement

A User Access to a Sip:

User → Keystone:
- Request to access to a sip
- Return a token for the sip
- Access Swift service with the sip token
- Return result
- Access Nova service with the sip token
- Return result
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

UTSA
AWS Access Control (AWS-AC) Model

AWS Access Control within a Single Account:

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

AWS-AC Model extends with SID

AWS Access Control within a Single Account:

- **Accounts (A)**
  - User Ownership (UO)
  - Group Ownership (GO)
  - Roles Ownership (RO)

- **Users (U)**
  - user_group
  - virtual user_role

- **Groups (G)**
  - Virtual Permission Assignment (VPA)

- **Roles (R)**
  - Virtual Permission Assignment (VPA)

- **Services (S)**
  - OT Ownership (OTO)

- **Object Types (OT)**
  - PRMS

- **Operations (OPR)**
Enforcement - Setup SID-service
## Enforcement DataBase

### SIDS Table in Database

<table>
<thead>
<tr>
<th>sid_id</th>
<th>sid_name</th>
<th>sid_members</th>
<th>core_project</th>
<th>open_project</th>
</tr>
</thead>
<tbody>
<tr>
<td>wbxiA97YH4c8jQARRGslg7hkCjpHIKbu</td>
<td>Sid1</td>
<td>{&quot;SAWS&quot;: &quot;042298307144&quot;, &quot;CPS&quot;: &quot;934324332443&quot;}</td>
<td>401991328752</td>
<td>434230153961</td>
</tr>
</tbody>
</table>

1 row in set (0.00 sec)

### SIPs Table in Database

<table>
<thead>
<tr>
<th>sip_account_id</th>
<th>account_name</th>
<th>sip_members</th>
<th>status</th>
<th>sid_id</th>
</tr>
</thead>
<tbody>
<tr>
<td>401991328752</td>
<td>Sid1_cp</td>
<td>{&quot;SAWS&quot;: &quot;042298307144&quot;, &quot;CPS&quot;: &quot;934324332443&quot;}</td>
<td>1</td>
<td>j3molQAxgAn3jCayTFZLsi5IchTf9C1w</td>
</tr>
<tr>
<td>434230153961</td>
<td>Sid1_op</td>
<td>{&quot;SAWS&quot;: &quot;042298307144&quot;, &quot;CPS&quot;: &quot;934324332443&quot;}</td>
<td>1</td>
<td>j3molQAxgAn3jCayTFZLsi5IchTf9C1w</td>
</tr>
<tr>
<td>557554226495</td>
<td>Sip1</td>
<td>{&quot;SAWS&quot;: &quot;042298307144&quot;, &quot;CPS&quot;: &quot;934324332443&quot;}</td>
<td>1</td>
<td>j3molQAxgAn3jCayTFZLsi5IchTf9C1w</td>
</tr>
<tr>
<td>652714115935</td>
<td></td>
<td>{}</td>
<td>0</td>
<td>j3molQAxgAn3jCayTFZLsi5IchTf9C1w</td>
</tr>
</tbody>
</table>

4 rows in set (0.00 sec)
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:

- SecAdmin
  - return sid id
  - create a sid
  - return account number and role names

- SID-service
  - return sid id
  - create a new sid record in database table
  - create core project, create open project
  - return account numbers and role names

- SID-server Backend
  - SID-manager AWS Account
Enforcement

Process of Creating a Sip:

SecAdmin
- return sip id (account number)
- create a sip
- return account number and role names

SID-service
- return sip id (account number)
- create a new sip record in database table SIPs
- return account numbers and role names

SID-manager AWS Account
- create a sip
- return account numbers and role names

SID-server Backend
- create a new sip record in database table SIPs
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-AC Model extends with SID
Enforcement

- Azure Manger Account
  - SID-manager
    - Azure Active Directory (Sid-1)
      - SID-manager AAD role: Global administrator
    - Subscription (Sid-1)
    - Resource Group (Cp)
    - Resource Group (Op)
    - Resource Group (Sip-1)
    - Resource Group (Sip-n)
  - Azure Active Directory (Sid-k)
    - SID-manager AAD role: Global administrator
    - Subscription (Sid-k)
    - Resource Group (Cp)
    - Resource Group (Op)
    - Resource Group (Sip-1)
    - Resource Group (Sip-m)

- SID-manager Sub role: Service administrator
Enforcement

• Create a Sid:

- Create a Sid:
  - create a sid
  - return sid id
  - create a sid active directory, create a sid subscription, create core project resource group, create open project resource group.
  - return subscription id

- SID-service
  - create a new sid record in database SIDs table
  - return sid id

- SID-server Backend
  - return subscription id

- SID-manager Azure Account
  - return sid id

SecAdmin
Enforcement

- SIP-requests:

<table>
<thead>
<tr>
<th>SecAdmin</th>
<th>SID-server</th>
<th>SID-manager Azure account</th>
</tr>
</thead>
<tbody>
<tr>
<td>create a sip</td>
<td>create a resource group for the sip</td>
<td></td>
</tr>
<tr>
<td>return a sip</td>
<td>return a resource group</td>
<td></td>
</tr>
<tr>
<td>add a user</td>
<td>assign the user to the sip resource group</td>
<td></td>
</tr>
<tr>
<td>remove a user</td>
<td>remove the user from the sip resource group</td>
<td></td>
</tr>
<tr>
<td>return</td>
<td>return assignment</td>
<td></td>
</tr>
<tr>
<td>delete a sip</td>
<td>delete the sip resource group</td>
<td></td>
</tr>
<tr>
<td>return</td>
<td>return</td>
<td></td>
</tr>
</tbody>
</table>
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:

  - **AWS**
    - Sid-level:
      - Sid-1
      - Sid-i
    - Project-level:
      - Account-1
      - Account-i

  - **OpenStack**
    - Sid-level:
      - Domain-1
      - Domain-j
    - Project-level:
      - Project-1
      - Project-m

  - **Azure**
    - Sid-level:
      - Subscription-1
      - Subscription-k
    - Project-level:
      - ResourceGroup-1
      - ResourceGroup-n

  - SID-server

  - SID-manager
  - Account

UTSA
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

Thanks!