Multi-Tenant Access Control for Cloud Services

PhD Dissertation Defense

Bo Tang

Committee Members:

Dr. Ravi Sandhu, Chair

Dr. Kay Robbins

Dr. Gregory White

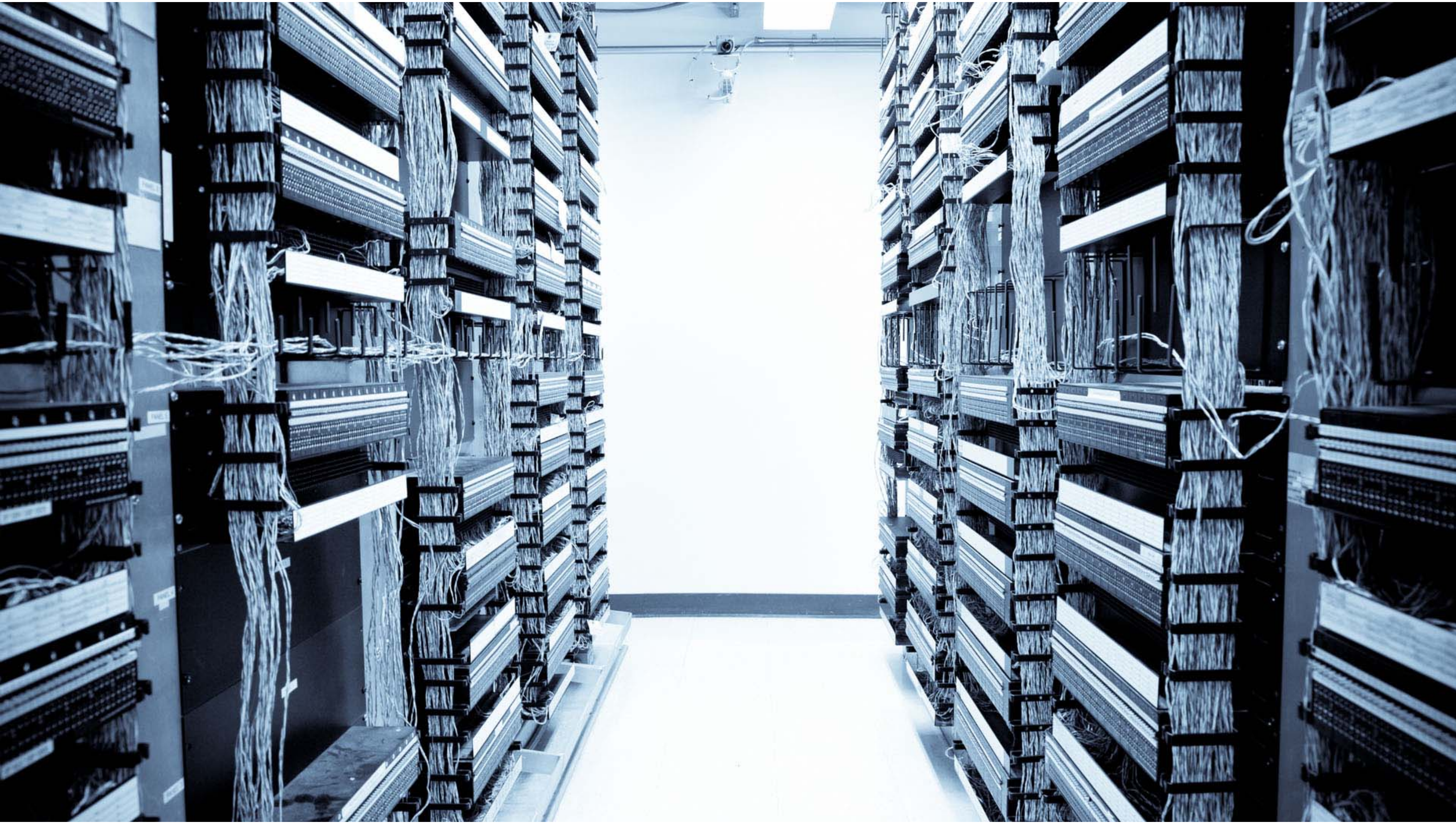
Dr. Weining Zhang

Dr. Jaehong Park

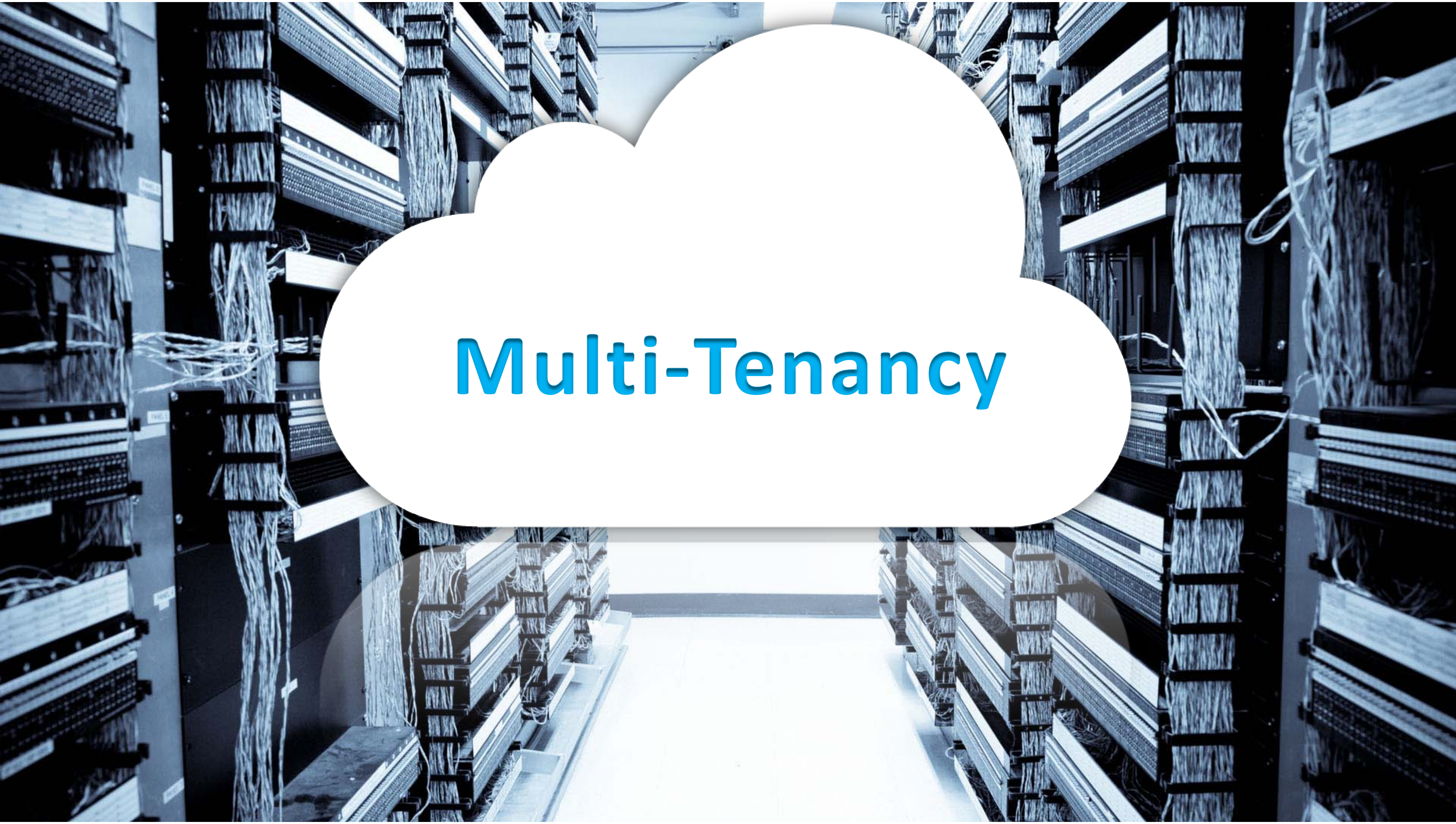
07/31/2014

The Cloud

Really? But where is my data?



Really? But where is my data?



Cloud & Multi-Tenancy

- Shared infrastructure
  - [$$$] -----> [$|$|$]
- Multi-Tenancy
  - Isolated workspace for customers
  - Virtually temporarily dedicated resources
- Problem:
  - How to collaborate across tenants?
    - Even if across my own tenants?

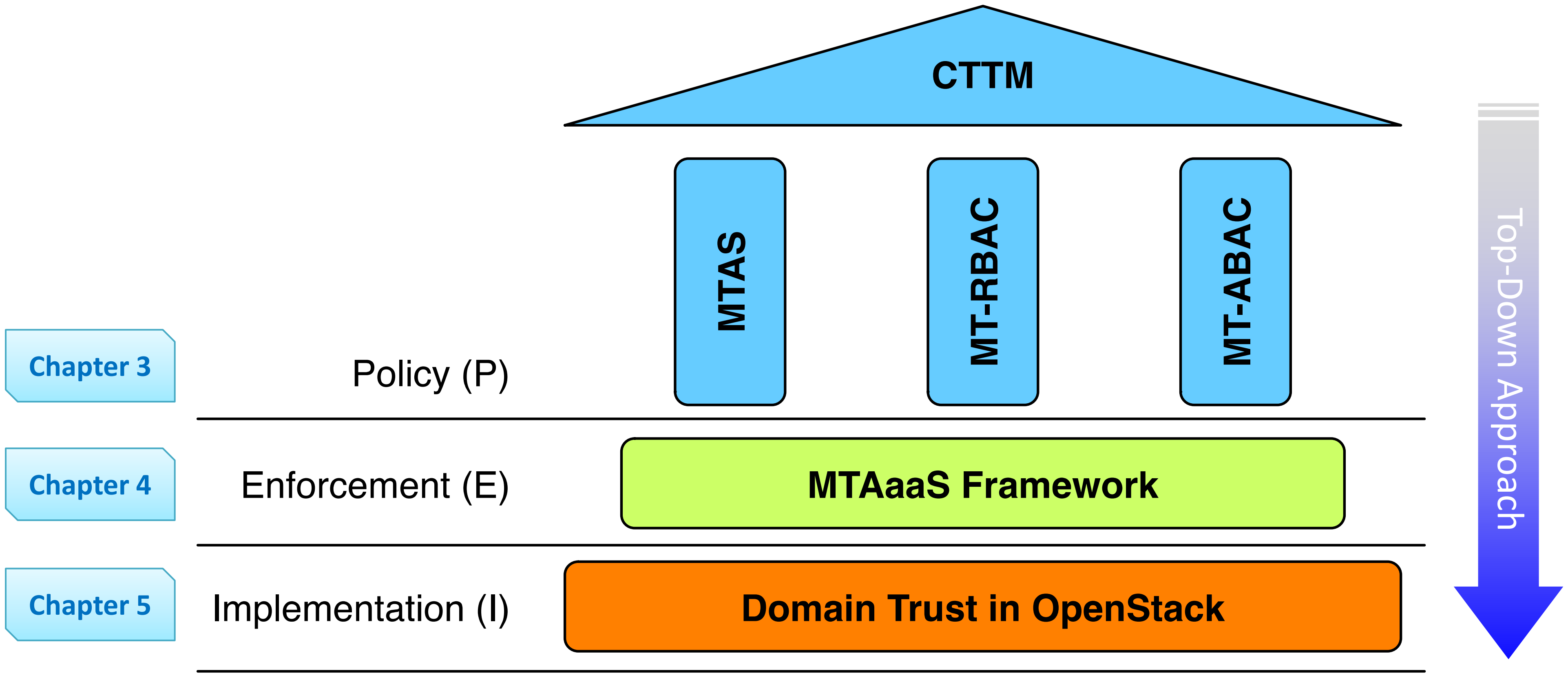
Define Tenant

- All deployment models are multi-tenant
  - E.g.: public cloud, private cloud and community cloud.
- From Cloud Service Provider (CSP) perspective
  - A billing customer
  - Manages its own users and cloud resources
- The owner of a tenant can be
  - An individual, an organization or a department in an organization, etc.

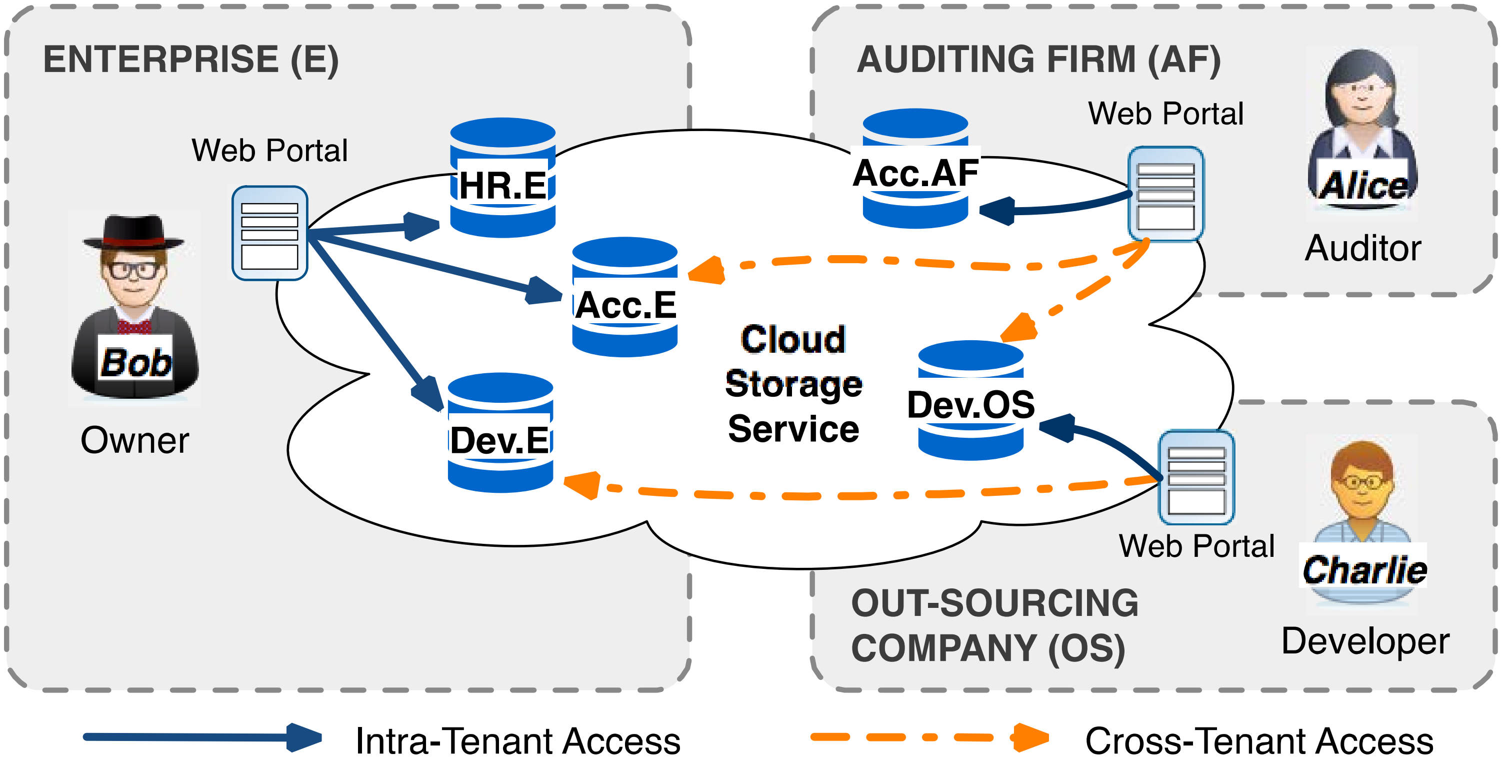
Characteristics of Cloud

- Centralized Facility
  - Resource pooling
- Self-Service Agility
  - Each tenant manages its own authorization
  - Tenants, users and resources are temporary
- Homogeneity
  - Identical or similar architecture and system settings
- Out-Sourcing Trust
  - Built-in collaboration spirit

Multi-Tenant Access Control (MTAC)



Motivation


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

Problem & Thesis

➢ Problem Statement

The fact that contemporary cloud services are intrinsically not designed to cultivate collaboration between tenants limits the development of the cloud. Fine-grained access control models in traditional distributed environments are not directly applicable.

➢ Thesis Statement

The problem of multi-tenant access control in the cloud can be partially solved by integrating various types of unidirectional and unilateral trust relations between tenants into role-based and attribute-based access control models.

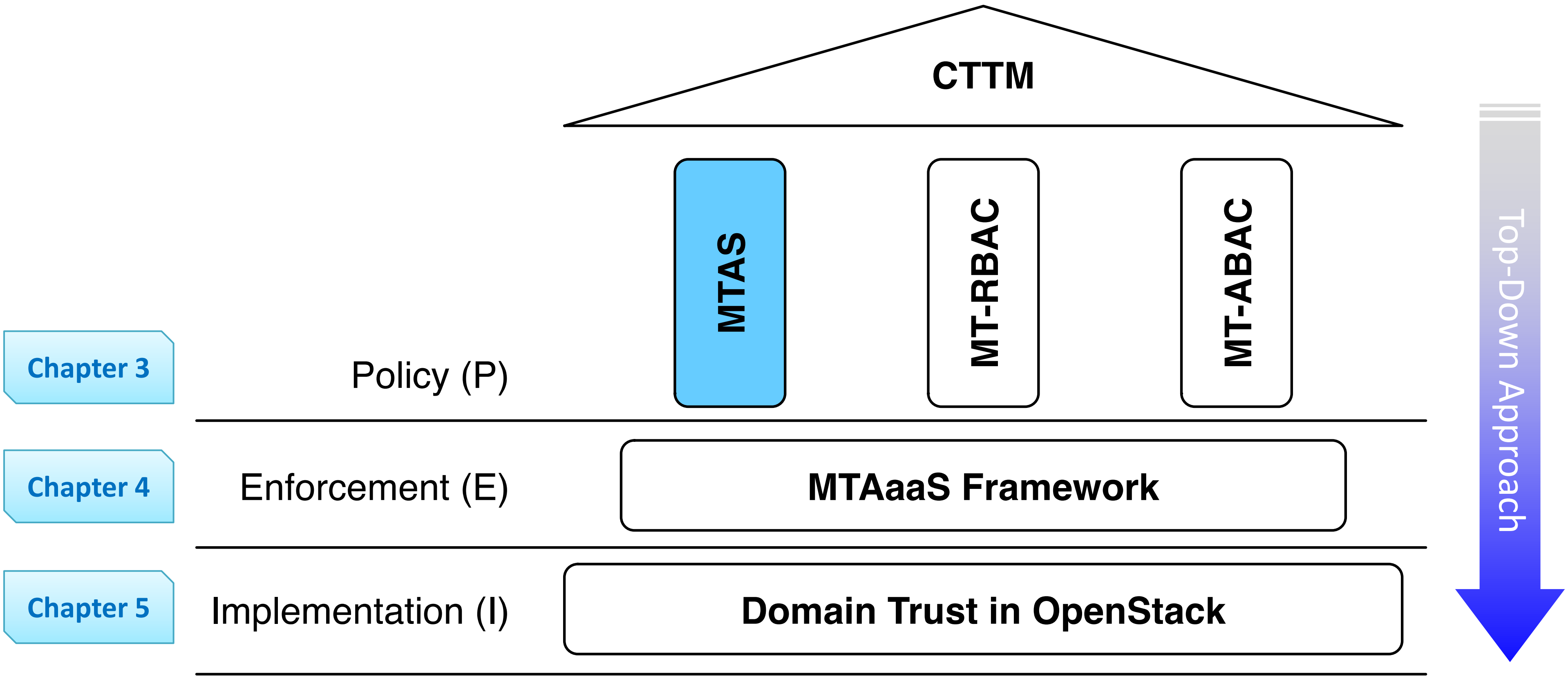
Chapter 2: Related Work

- Centralized Approaches
  - RBAC extensions: ROBAC, GB-RBAC
  - Multi-domain role mapping
- Decentralized Approaches
  - RT, dRBAC: credential-based delegation
  - Delegation models: PBDM, RBDM
- Attribute-Based Approaches
  - NIST ABAC: application framework for collaboration
  - ABAC models: ABURA, RBAC-A, $ABAC_{\alpha}$, $ABAC_{\beta}$
- Enforcement and Implementation
  - Grid: PERMIS, VOMS, CAS
  - Web: ABAC for SOA systems
  - Cloud: centralized authorization service with trust models

Scope and Assumptions

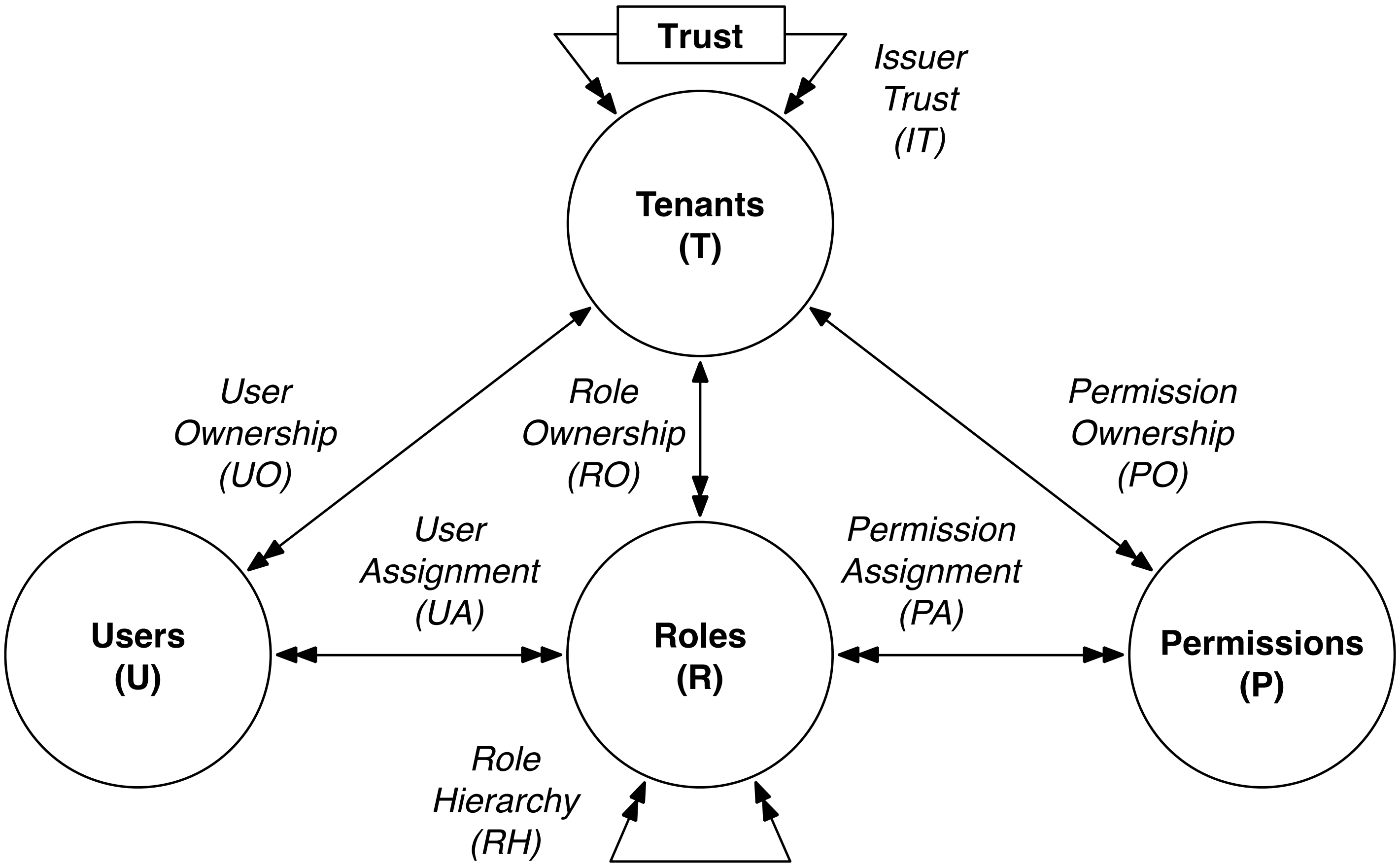
- Standardized APIs
  - Cross-tenant accesses are functionally available
- Properly authenticated users
- One Cloud Service
  - Of a kind: IaaS, PaaS or SaaS.
- Two-Tenant Trust (rather than community trust)
- Unidirectional Trust Relations
  - "I trust you" does not mean "you trust me"
- Unilateral Trust Relations (trustor trusts trustee)
  - Trustee cannot control the trust relation

Multi-Tenant Access Control (MTAC)



MTAS

Formalizing Calero et al work

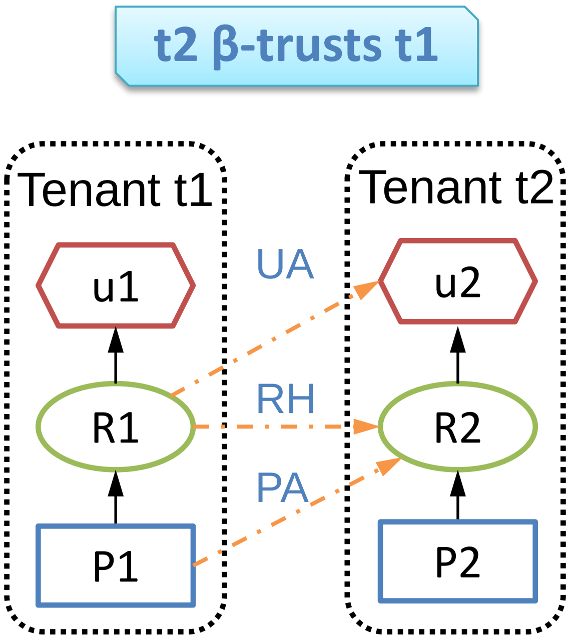


Tenant Trust

- Tenant Trust (*TT*) relation is not partial order
  - Reflexive: $A \trianglelefteq A$
  - But not transitive: $A \trianglelefteq B \land B \trianglelefteq C \nRightarrow A \trianglelefteq C$
  - Neither symmetric: $A \trianglelefteq B \nRightarrow B \trianglelefteq A$
  - Nor anti-symmetric: $A \trianglelefteq B \land B \trianglelefteq A \nRightarrow A \equiv B$

Administrative MTAS

- Tenants are managed by CSP
  - on self-service basis
- Each tenant administer:
  - Trust relations with other tenants
  - Entity components:
    - users, roles and permissions
  - UA, PA and RH assignments
    - Cross-tenant assignments are issued by the trustee (t1)
      - UA: trustor (t2) users to trustee (t1) roles
      - PA: trustee (t1) permissions to trustor (t2) roles
      - RH: trustee (t1) roles junior to trustor (t2) roles



Fine-grained Trust Extensions

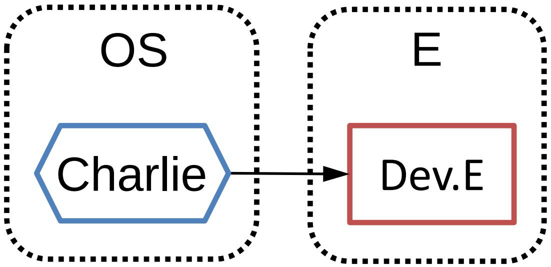
- Problem of MTAS trust model
  - Over exposure of trustor's authorization information
- Trustor-Centric Public Role (TCPR)
  - Expose only the trustor's public roles
    - E.g.: OS expose only the dev.OS role to all the trustees
- Relation-Centric Public Role (RCPR)
  - Expose public roles specific for each trust relation
    - E.g.: OS expose only the dev.OS role to E when OS trusts E

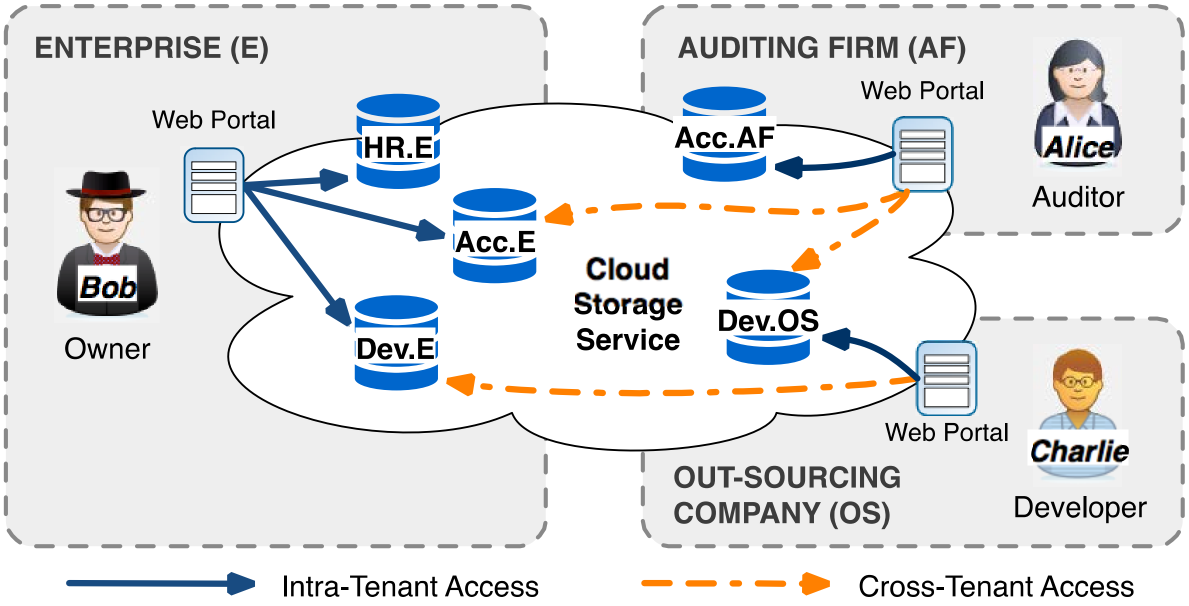
Trust Types Between Tenants

- Intuitive Trust (Type-α)
  - Delegations: RT, PBDM, etc.
  - Trustor gives access to trustee
    - Trustor has full control
- MTAS trust (Type-β)
  - Trustee gives access to trustor
- Other Types?
  - Trustee takes access from trustor (Type-γ)
  - Trustor takes access from trustee (Type-δ)
  - And more?

Example of Cross-Tenant Trust

[$]: grant the access

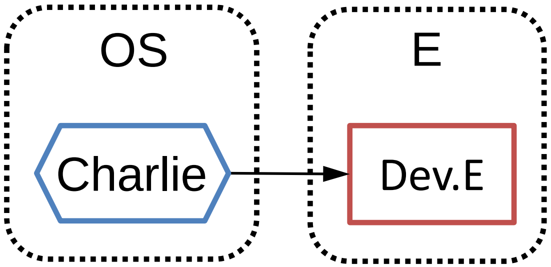


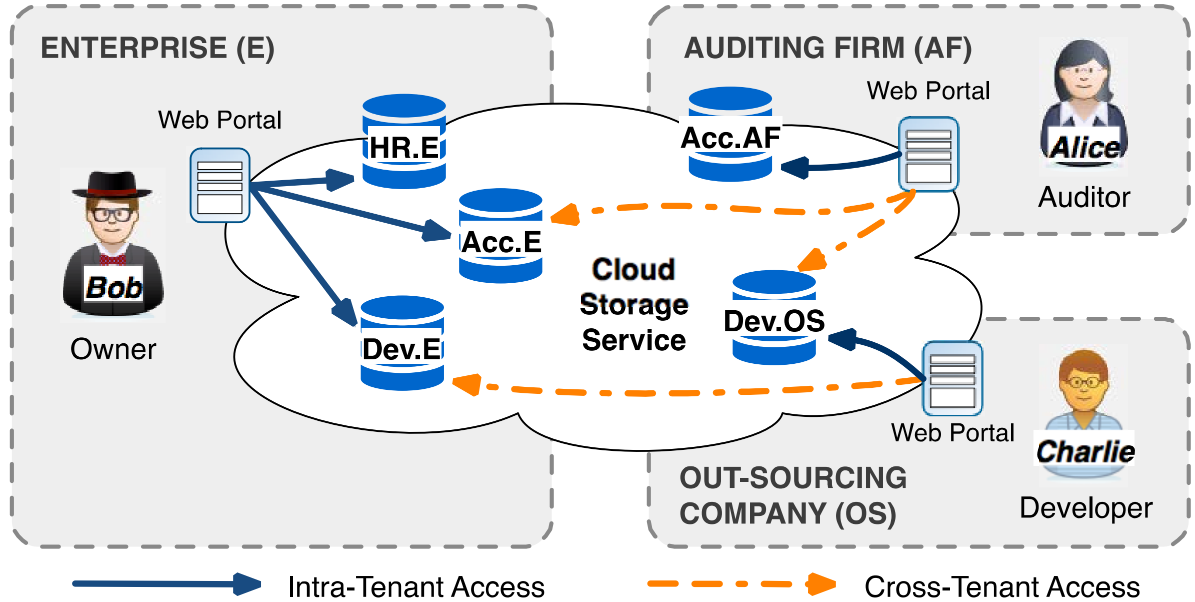


- Example:
  - Type-α: E trusts OS so that E can say [$].
  - Type-β: OS trusts E so that E can say [$].
  - Type-γ: E trusts OS so that OS can say [$].
  - Type-δ: OS trusts E so that OS can say [$].

Example of Cross-Tenant Trust

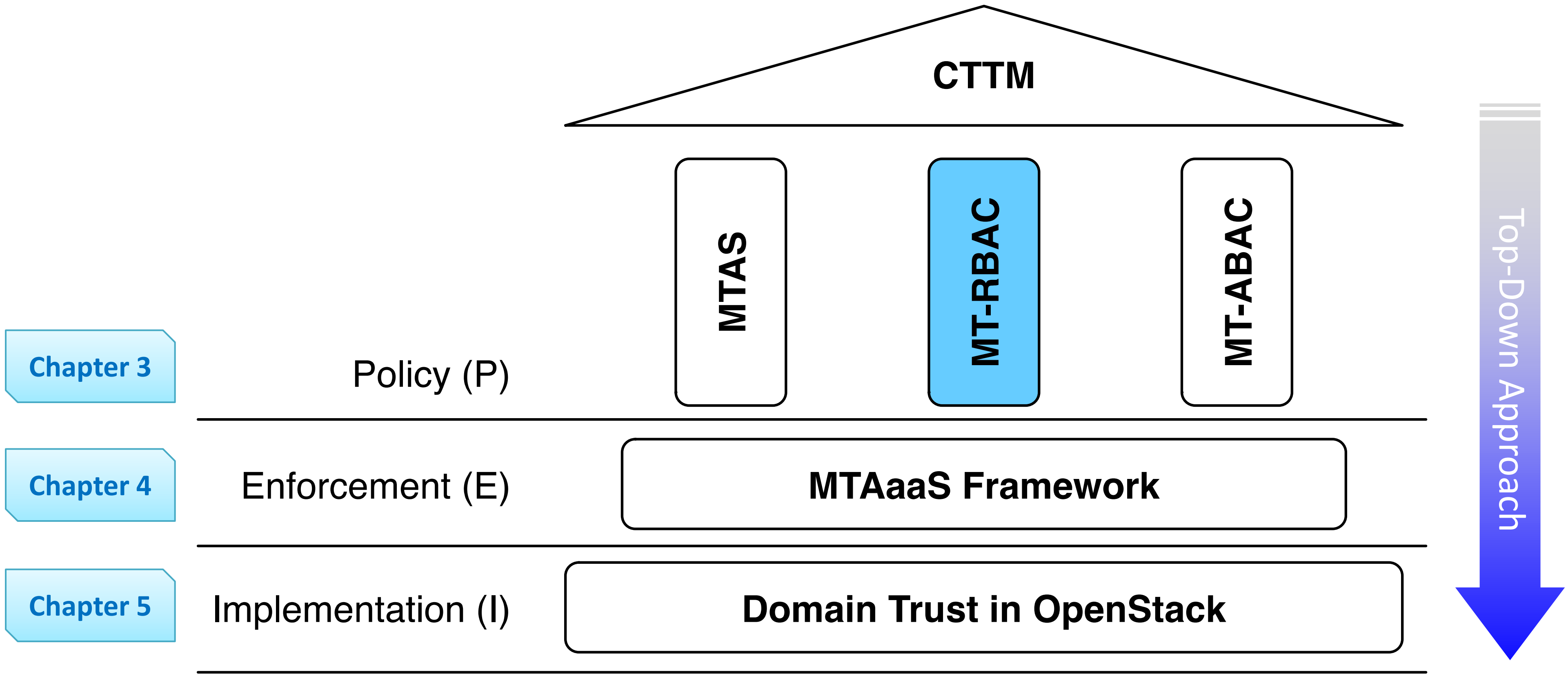
[$]: grant the access



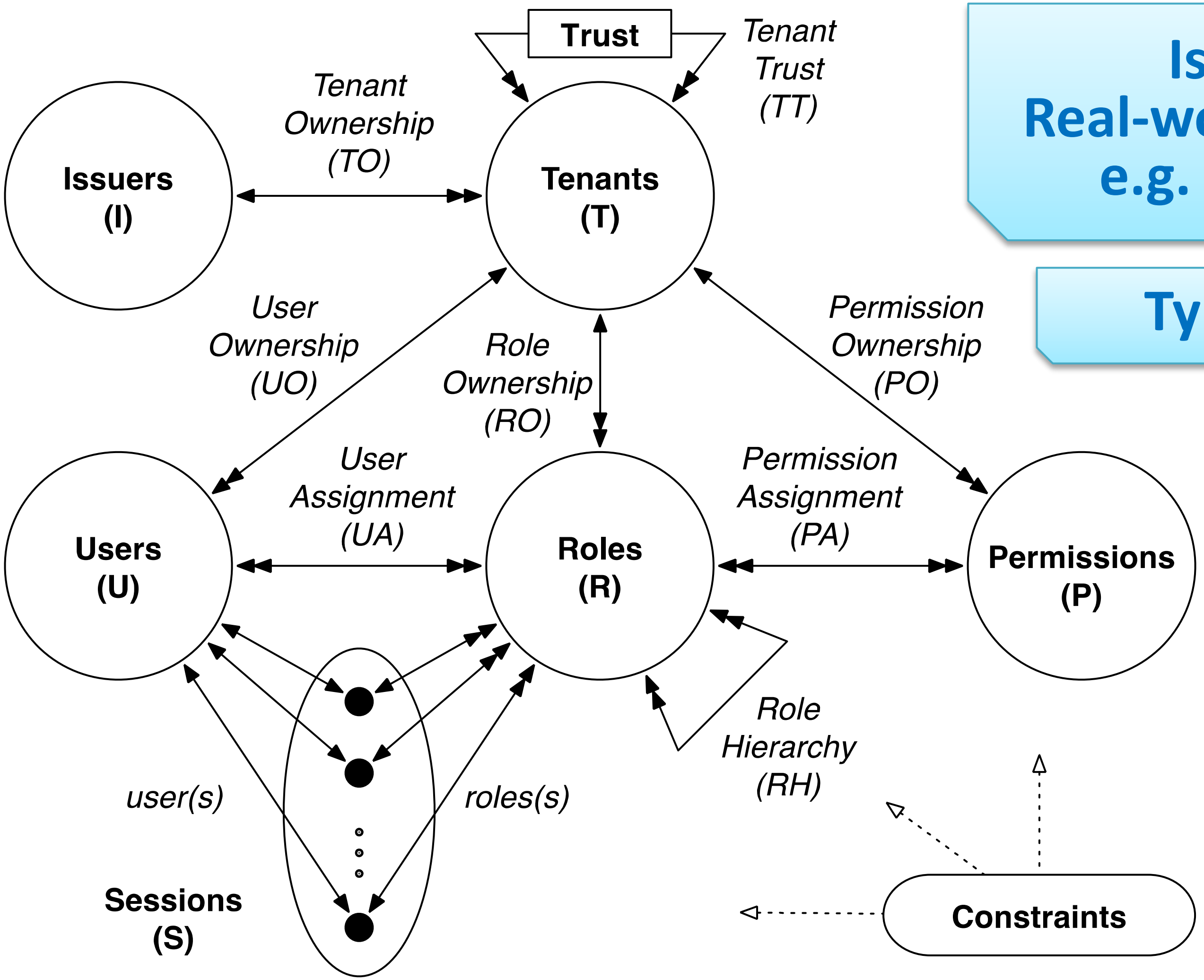


- Example:
  - Type-α: E trusts OS so that E can say [$].
  - Type-β: OS trusts E so that E can say [$].
  - Type-γ: E trusts OS so that OS can say [$].
  - ~~Type-δ: OS trusts E so that OS can say [$].~~

Multi-Tenant Access Control (MTAC)



MT-RBAC

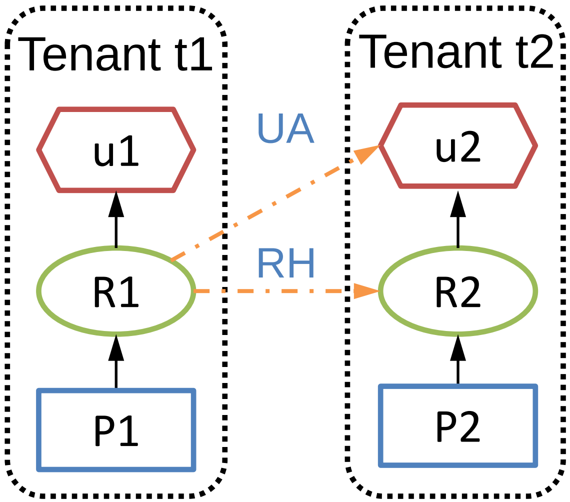


Issuers: Real-world Owners e.g. E and OS

Type-γ Trust

Administrative MT-RBAC

t1 γ-trusts t2



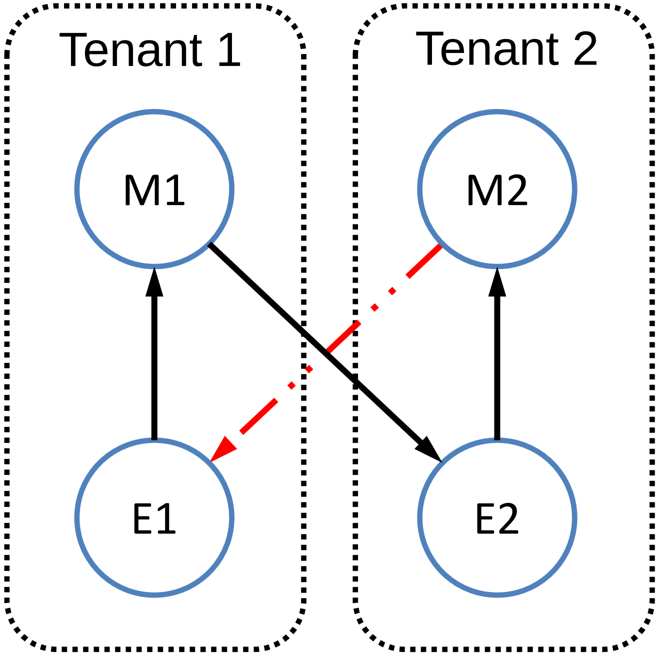
- Issuers administer tenants
- Each issuer administer:
  - Trust relations from owned tenants
  - Entity components:
    - tenants, users, roles and permissions
  - UA, PA and RH assignments
    - Cross-tenant assignments are issued by the trustee's (t2's) issuer
      - UA: trustee (t2) users to trustor (t1) roles
      - RH: trustor (t1) roles junior to trustee (t2) roles
    - Cross-tenant PA assignments are intentionally banned
      - PA: trustee (t2) assign trustor (t1) permissions to trustee (t2) roles
      - Problem:
        - » Trustor cannot revoke PA other than remove the trust

Finer-grained Trust Models

- MT-RBAC0: Base Model
  - Trustor exposes all the roles to trustees
- MT-RBAC1: Trustee-Independent Public Role (TIPR)
  - Expose only the trustor's public roles
    - E.g.: E expose only the dev.E role to all the trustees
- MT-RBAC2: Trustee-Dependent Public Role (TDPR)
  - Expose public roles specific for each trustee
    - E.g.: E expose only the dev.E role to OS when E trusts OS

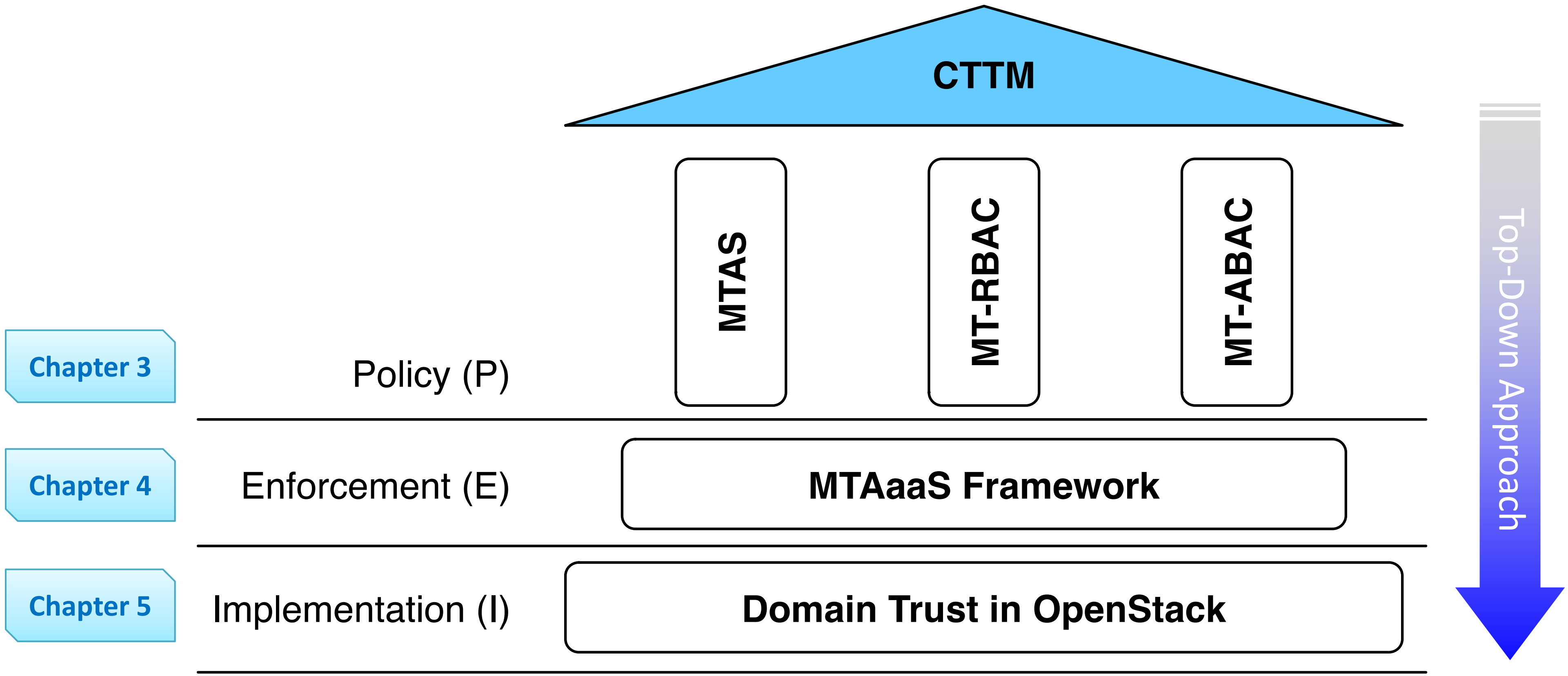
Constraints

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



- Chinese Wall: conflict of interests among tenants
    - E.g.: never share resources with competitors.

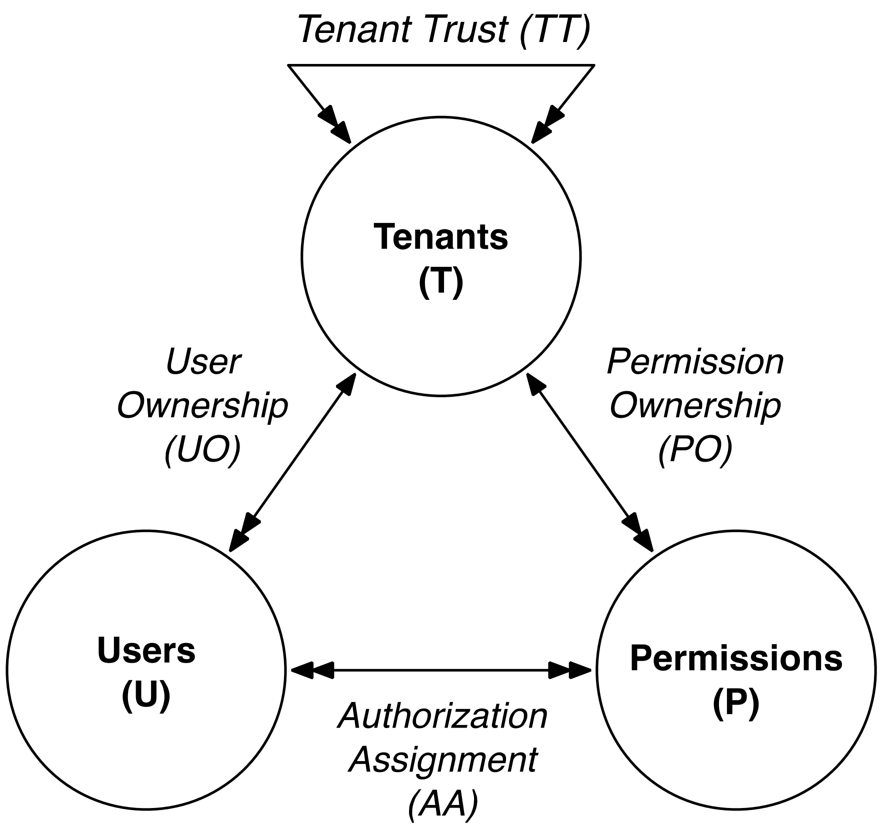
Multi-Tenant Access Control (MTAC)



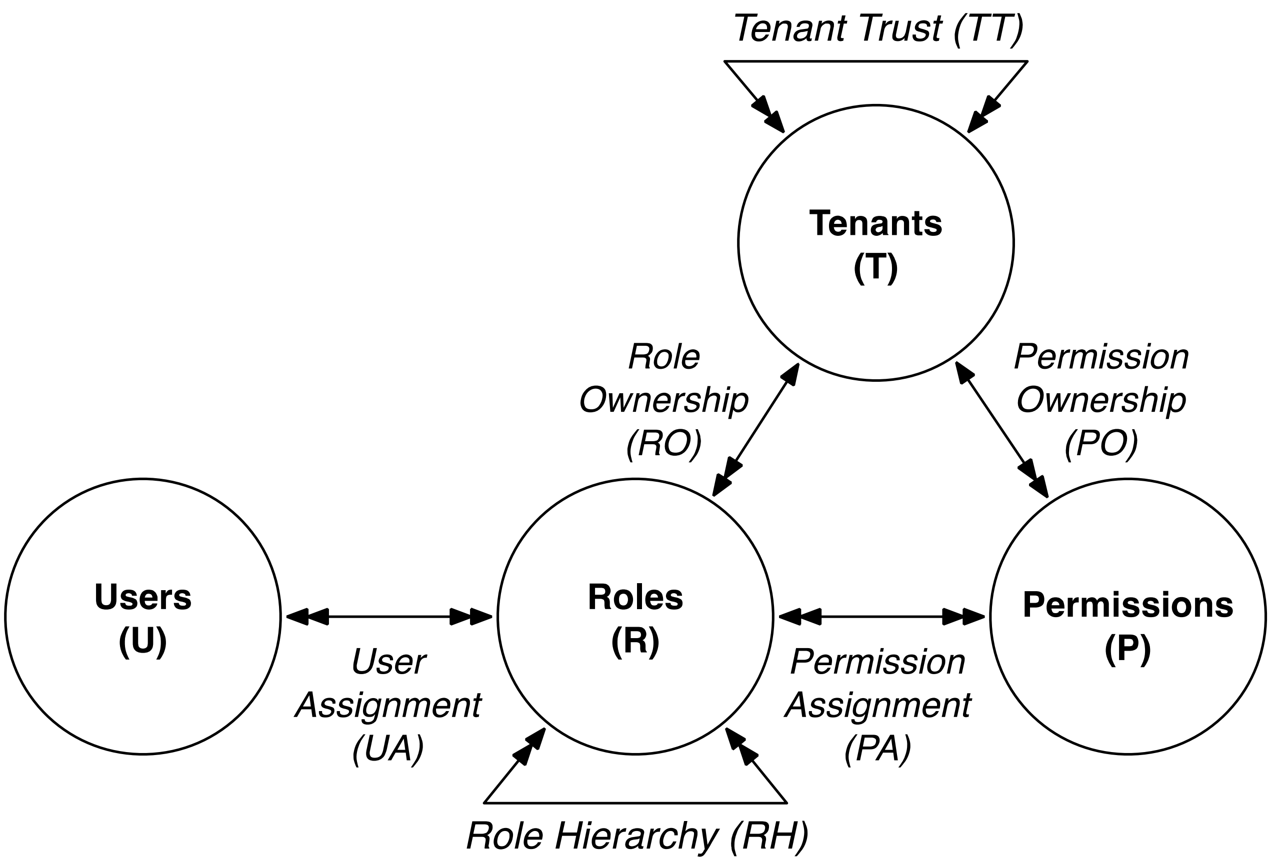
CTTM Trust Types

- Four potential trust types:
  - Type-α: trustor can <u>give</u> access to trustee. (e.g. RT)
  - Type-β: trustee can <u>give</u> access to trustor. (e.g. MTAS)
  - Type-γ: trustee can <u>take</u> access from trustor. (e.g. MT-RBAC)
  - ~~Type-δ: trustor can <u>take</u> access from trustee.~~
    - No meaningful use case, since the trustor holds all the control of the cross-tenant assignments of the trustee's permissions.

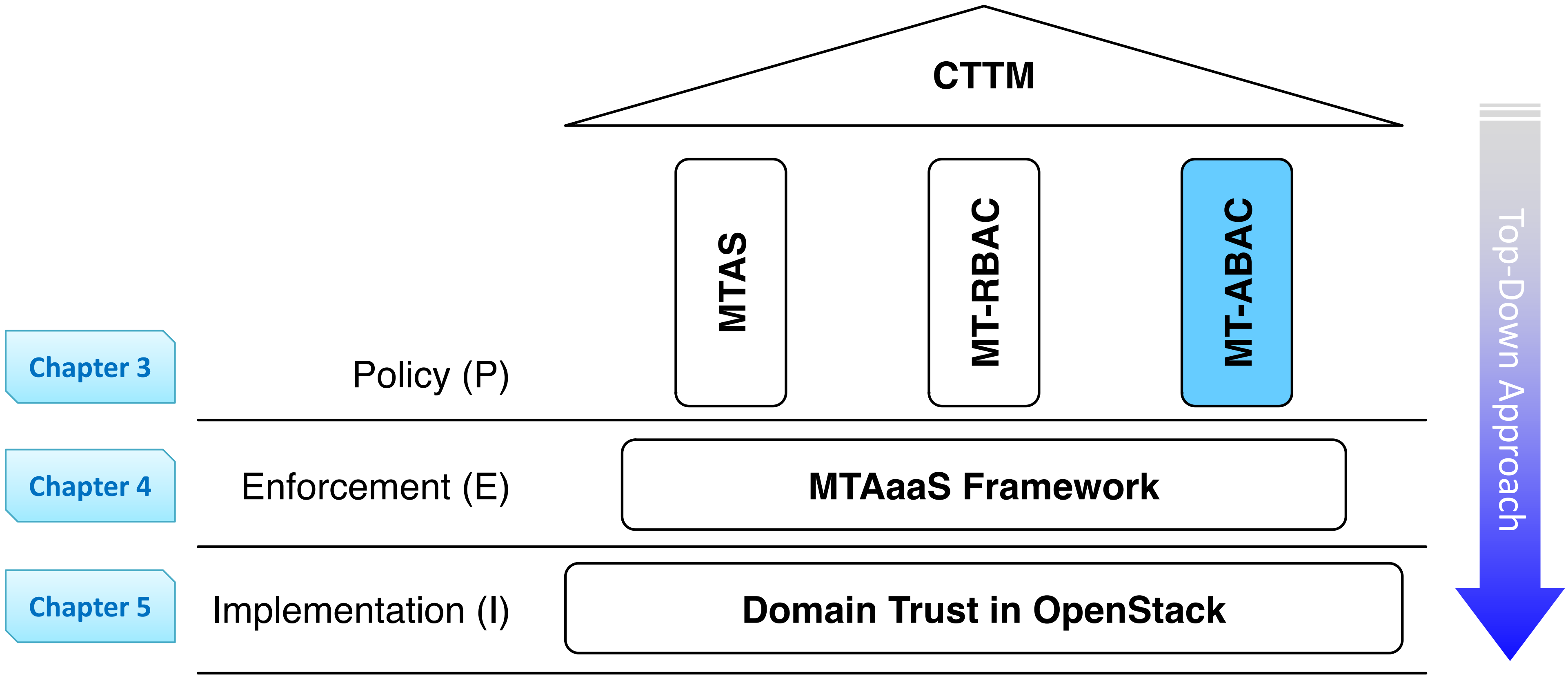
Formalized CTTM Model



Role-Based CTTM



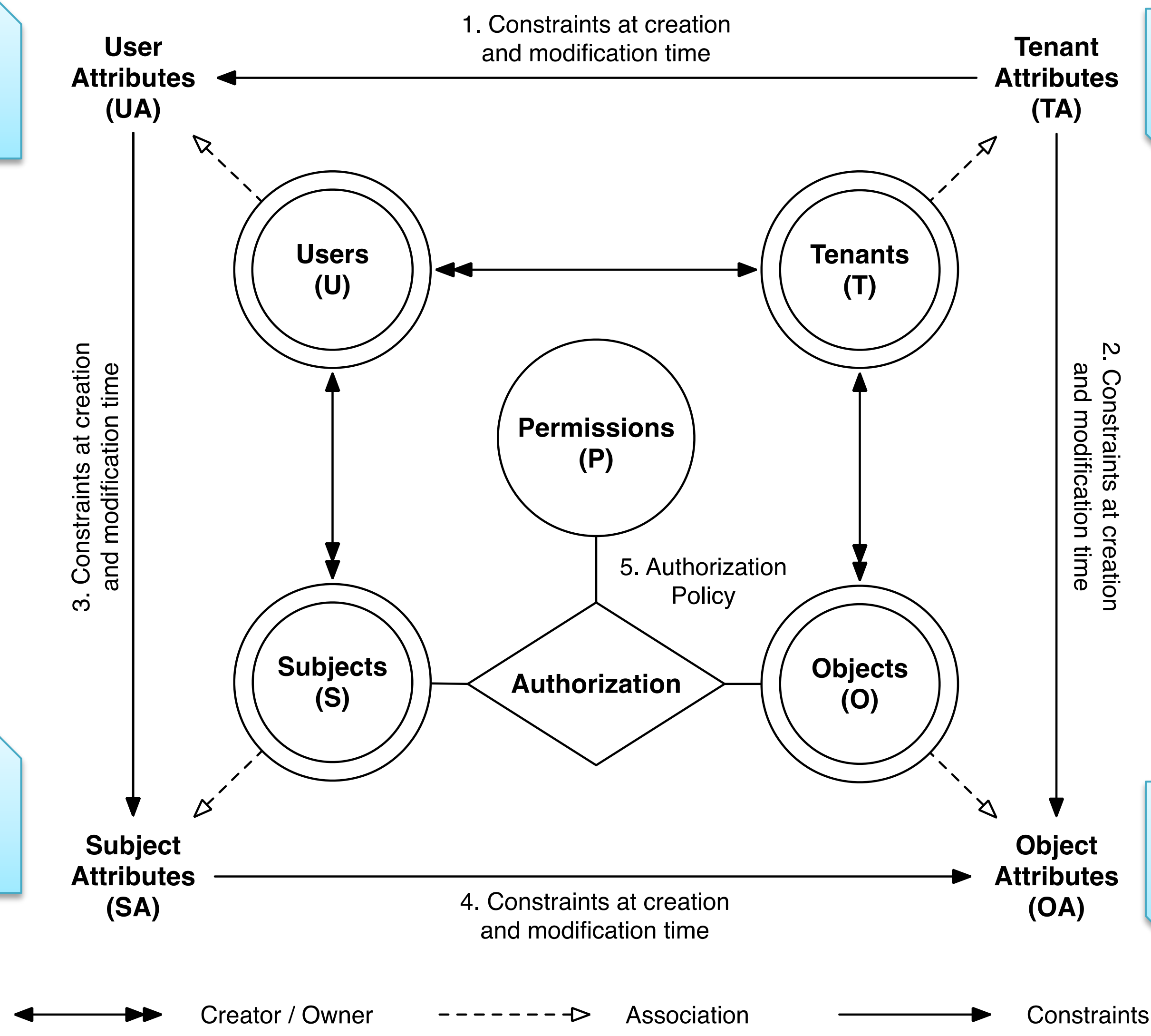
Multi-Tenant Access Control (MTAC)



MT-ABAC

uid: u2
utid: t2

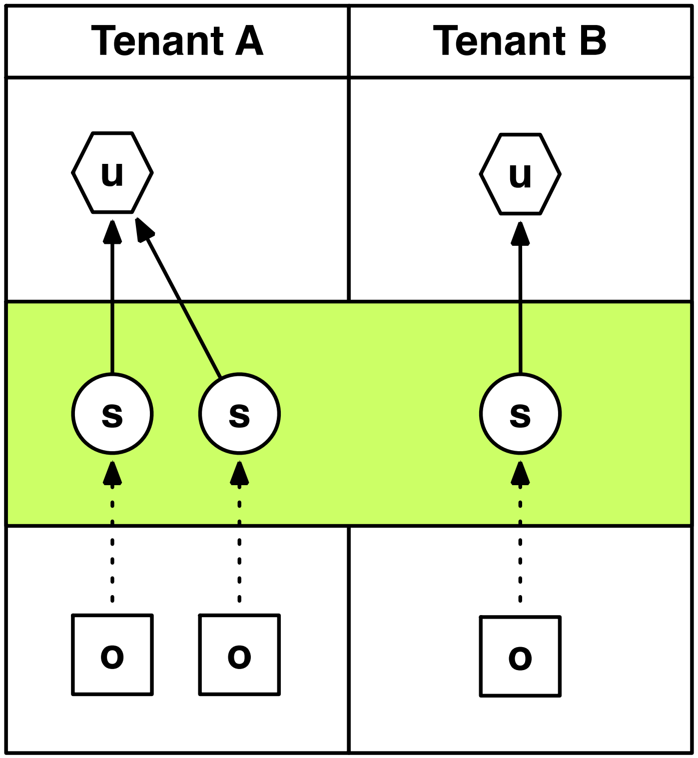
γ-trustee: {t2}
tid: t1



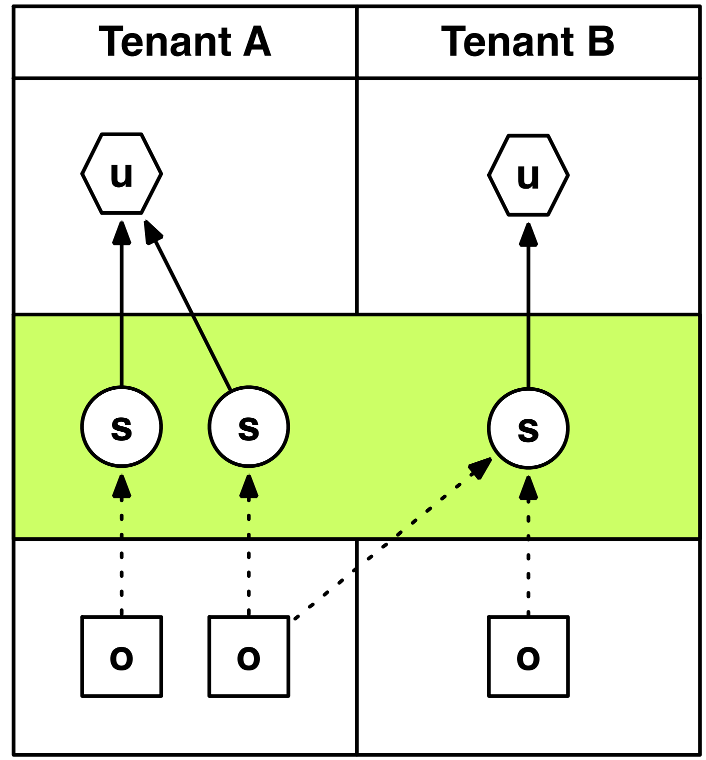
sowner: u2
sid: s2

oid: o1
otid: t1

Multi-Tenant Access Example



(a) no trust required

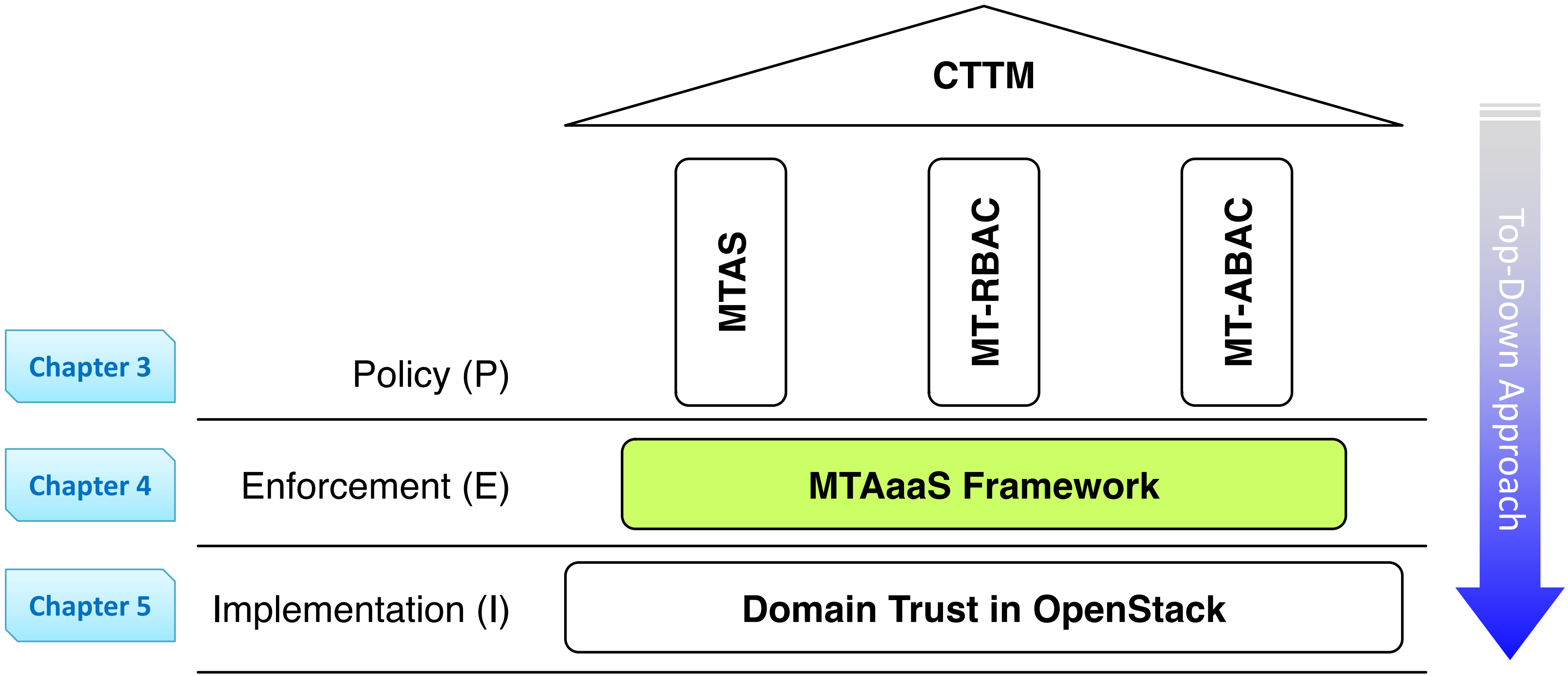


(b) require A trust B

Real-World Clouds

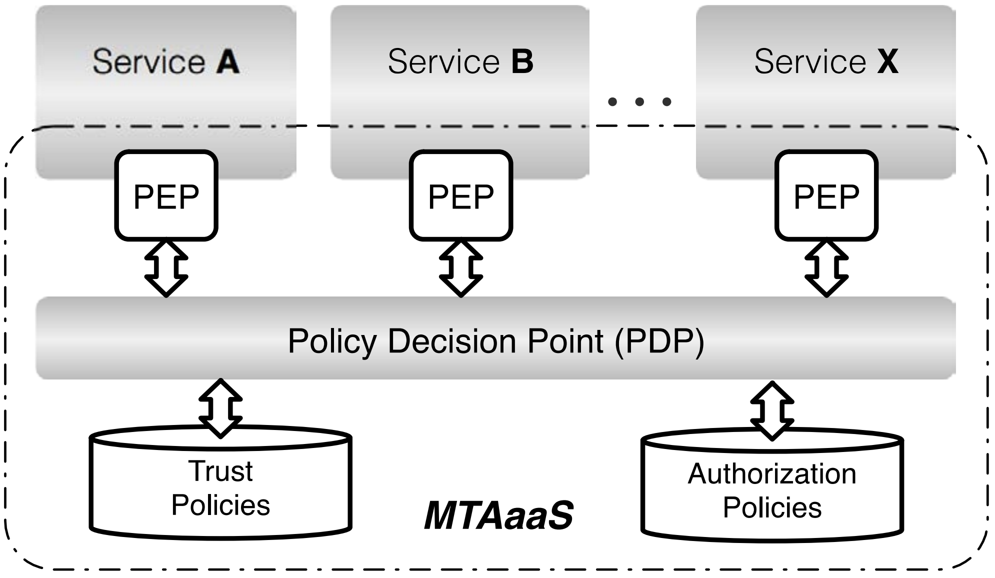
- AWS
  - Collaboration between accounts
    - E.g.: E trusts OS
  - Unilateral trust relation (Type-α)
    - The trustor needs to map the roles
- OpenStack
  - User-level delegation (trust) can be established
  - Cross-domain assignments bear no control

Multi-Tenant Access Control (MTAC)



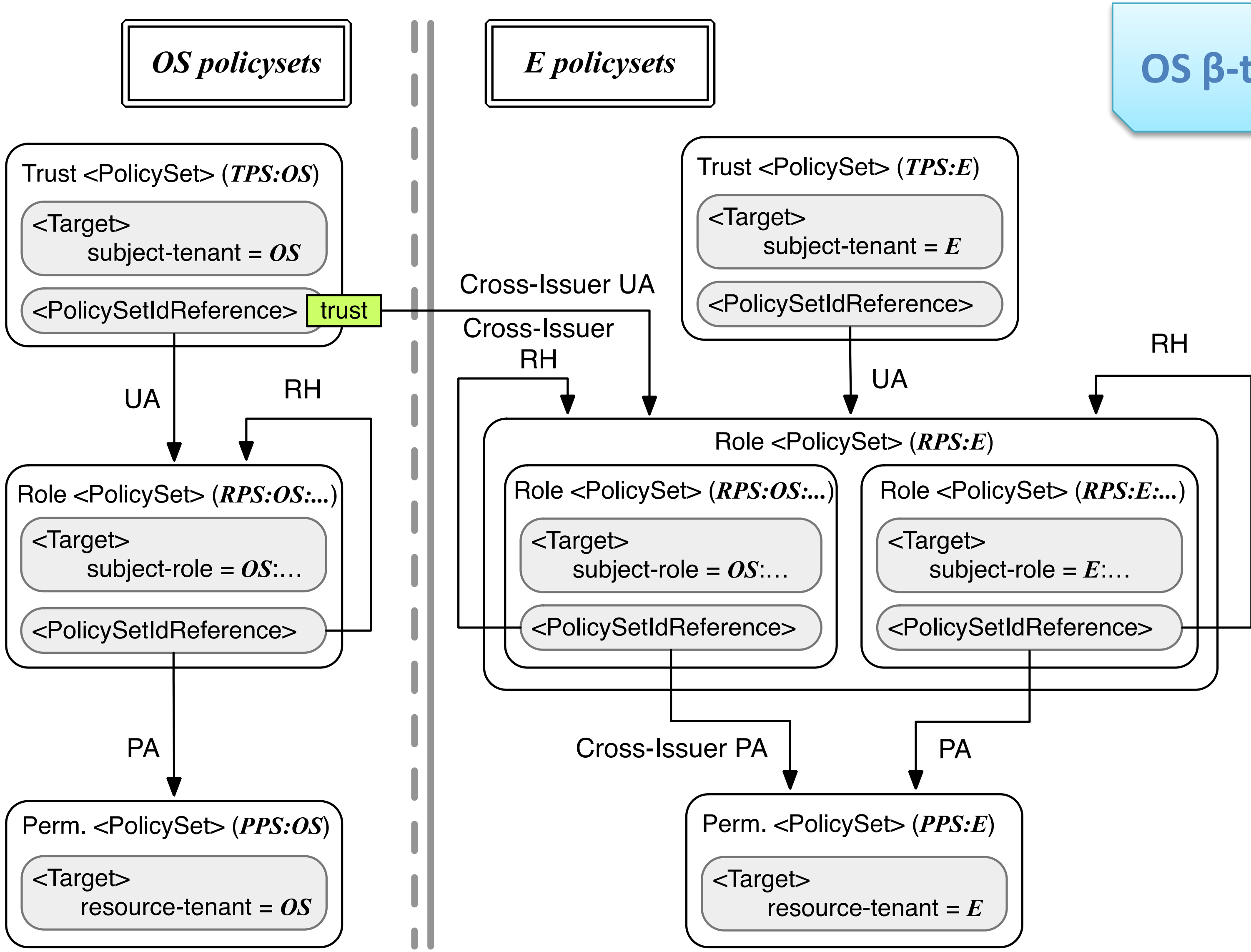
MTAaaS Platform Prototype

- Centralized (Chosen)
  - Centralized PDP with distributed PEP
    - Pros: easy management
    - Cons: volume of requests may be high
- Decentralized
  - Distributed PDP and PEP
    - Pros: requests handling
    - Cons: keep decision consistent

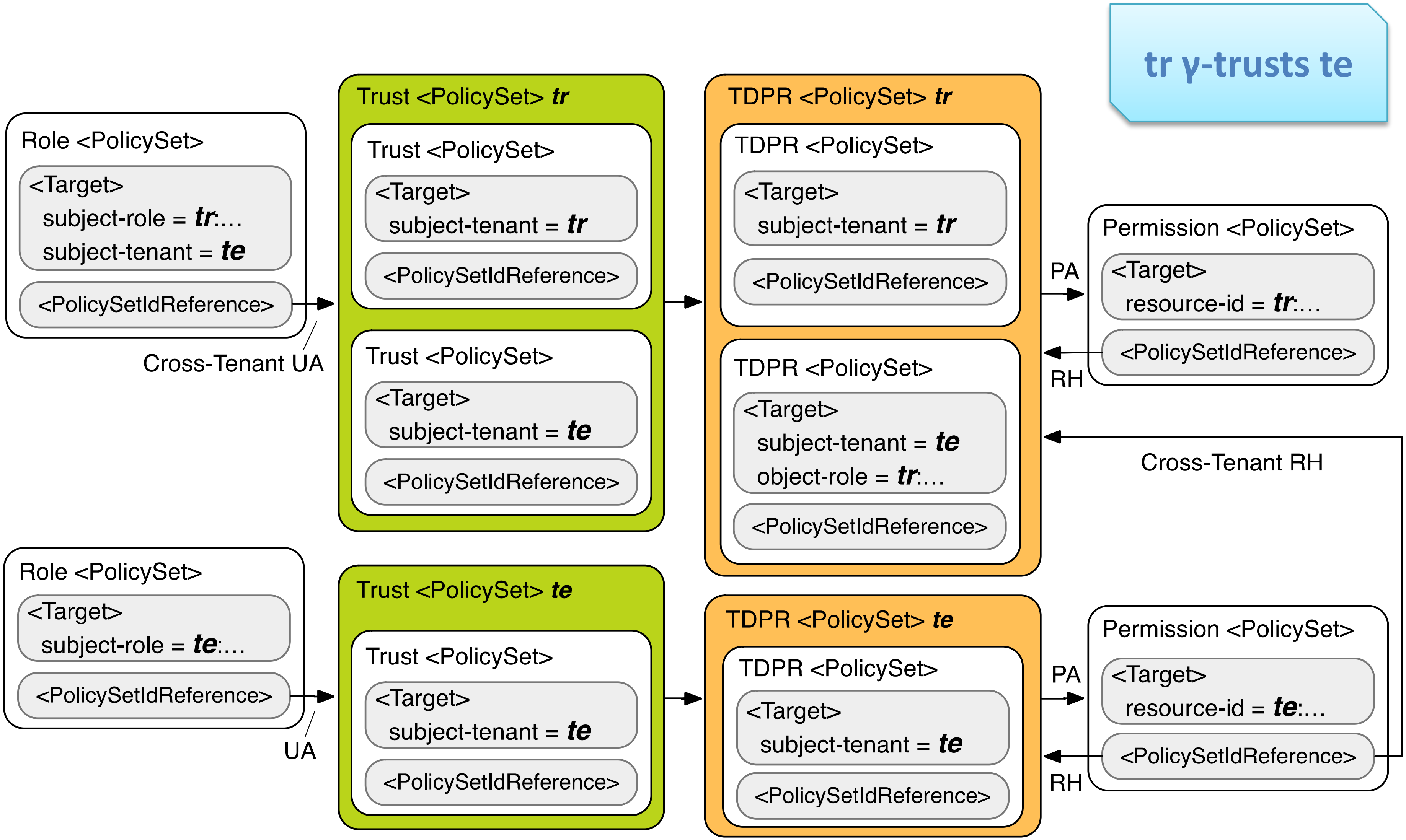


Example MTAS policy structure

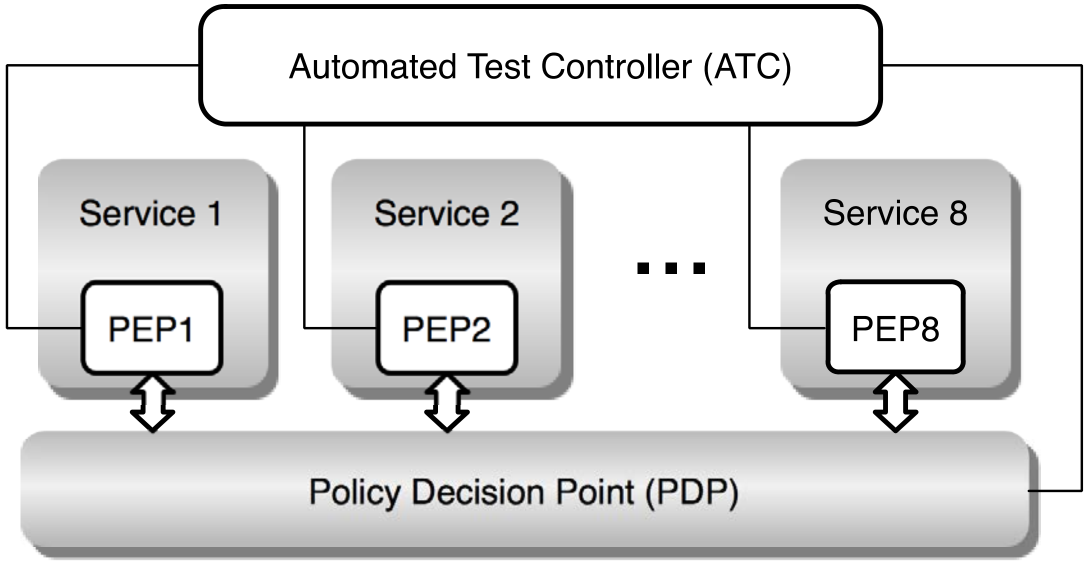
OS β-trusts E



MT-RBAC2 Policy Example



Experiment Environment



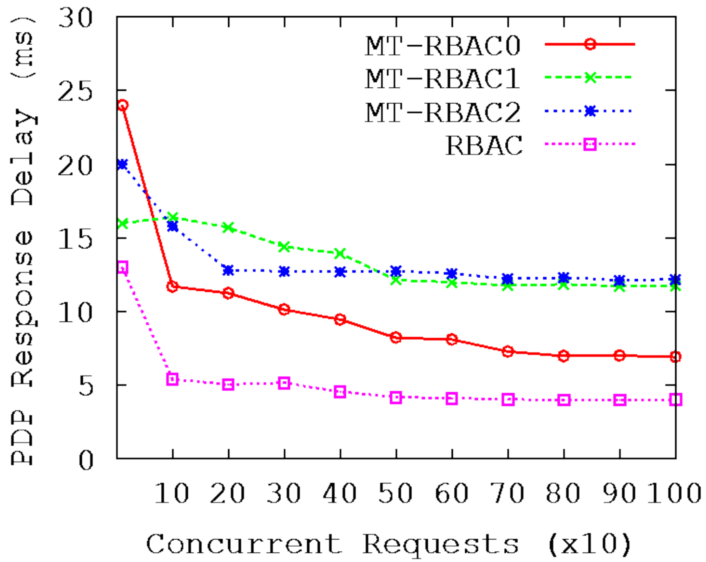
- FlexCloud Testbed

1 unit = 1CPU/1GB RAM

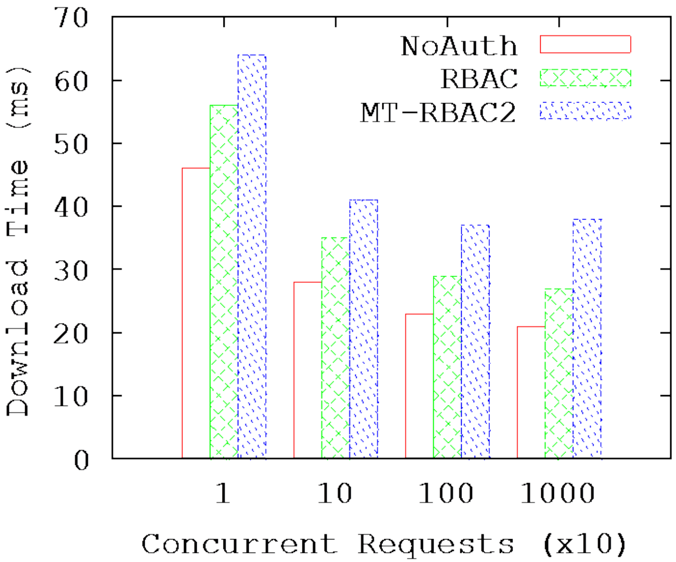
  - PEP × 8: SmartOS 1.8.1 / CPU Cap=350 / 256MB RAM
  - PDP: 64-bit CentOS 6 / 1-, 2-, 4-, 8-, 16-Units
  - ATC: SmartOS 1.8.4 / CPU Cap=350 / 1GB RAM
  - PEPs in a same network which is different with PDP's

Evaluation: Performance

- MT-RBAC vs RBAC
  - More policy references incur more decision time
- MT-RBAC2 introduces 12 ms authz. overhead.



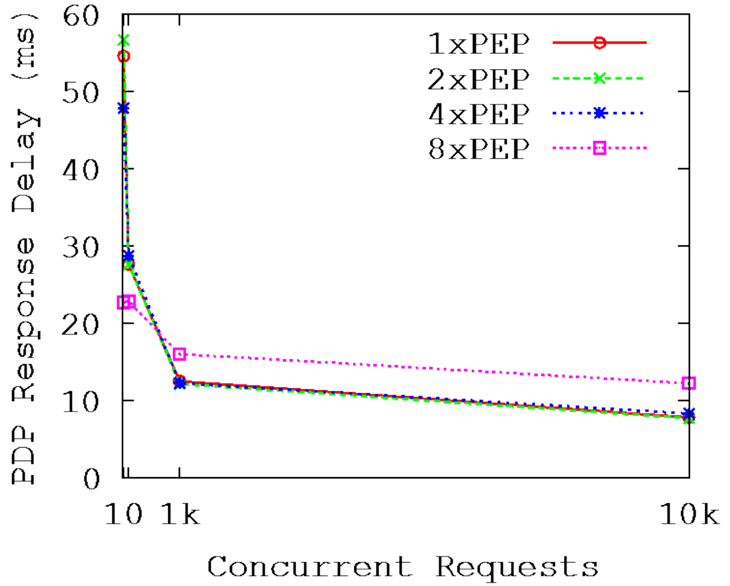
PDP Performance



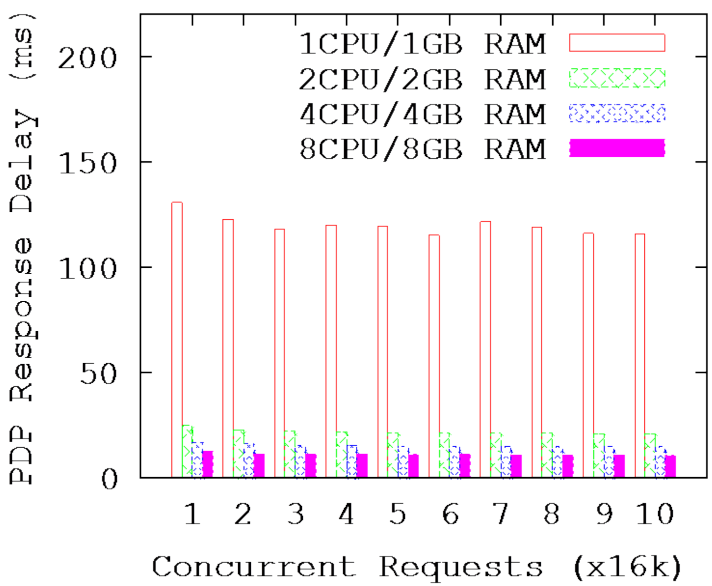
Client-End Performance when downloading 1KB file

Evaluation: Performance

➢ MTAS introduces 12 ms authz. overhead.



PDP Response Delay with various PEP amount

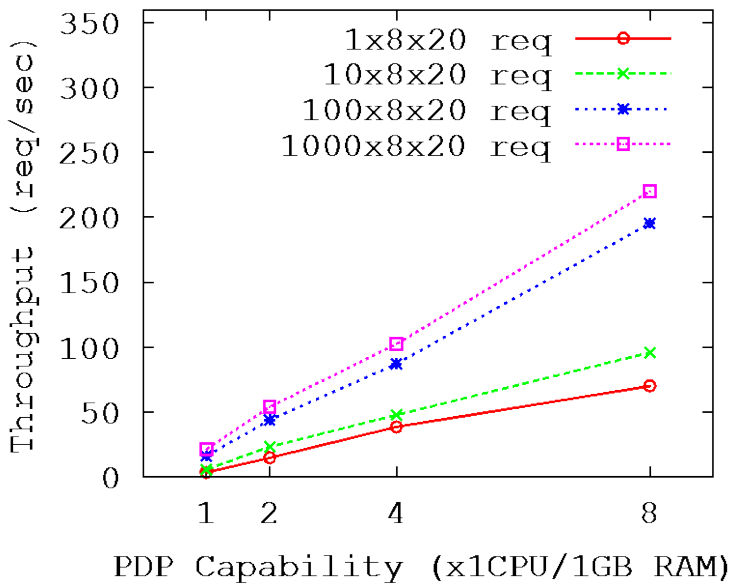


PDP Response Delay with various hardware capability and 1k tenants

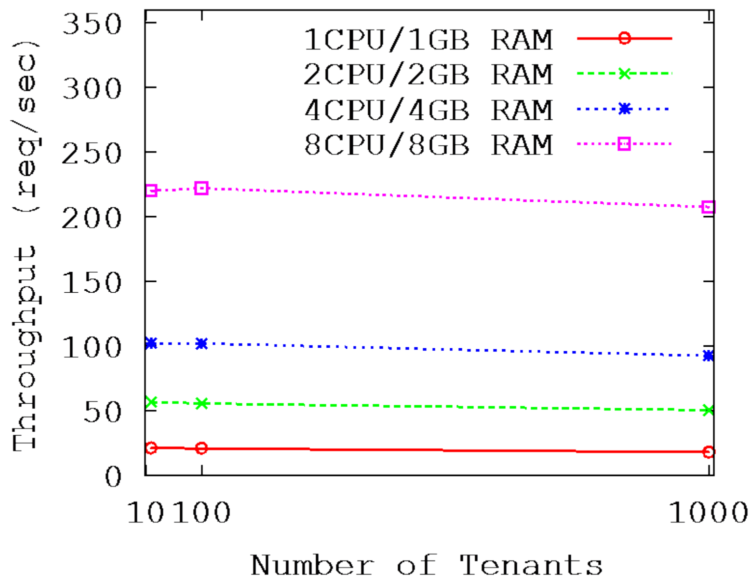
Evaluation: Scalability

- Scalable in terms of both
  - PDP hardware capacity
  - Policy complexity

$$Throughput = \frac{1}{Average\_Delay \times CPU\_Utilization}$$

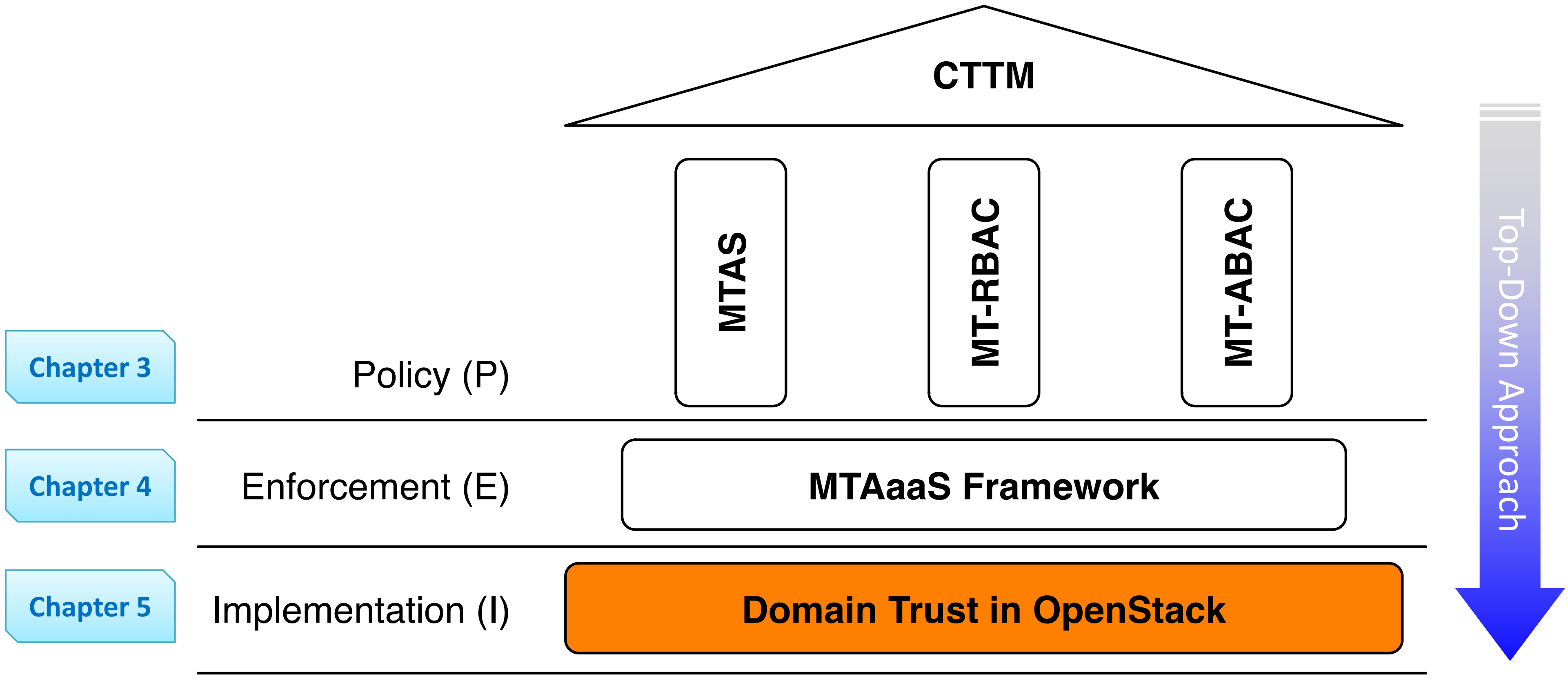


Policy Complexity Scalability Results

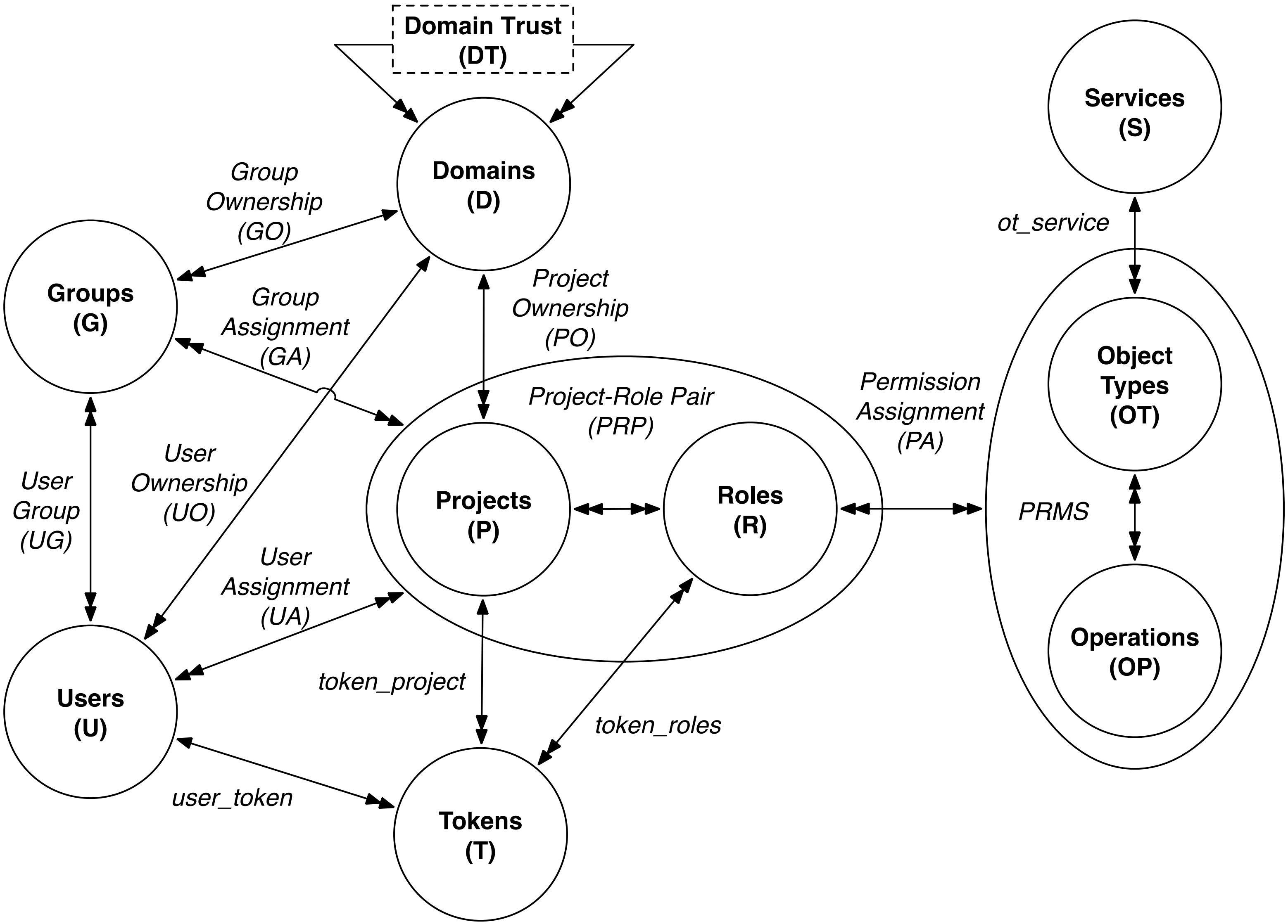


Policy Complexity Scalability Results

Multi-Tenant Access Control (MTAC)

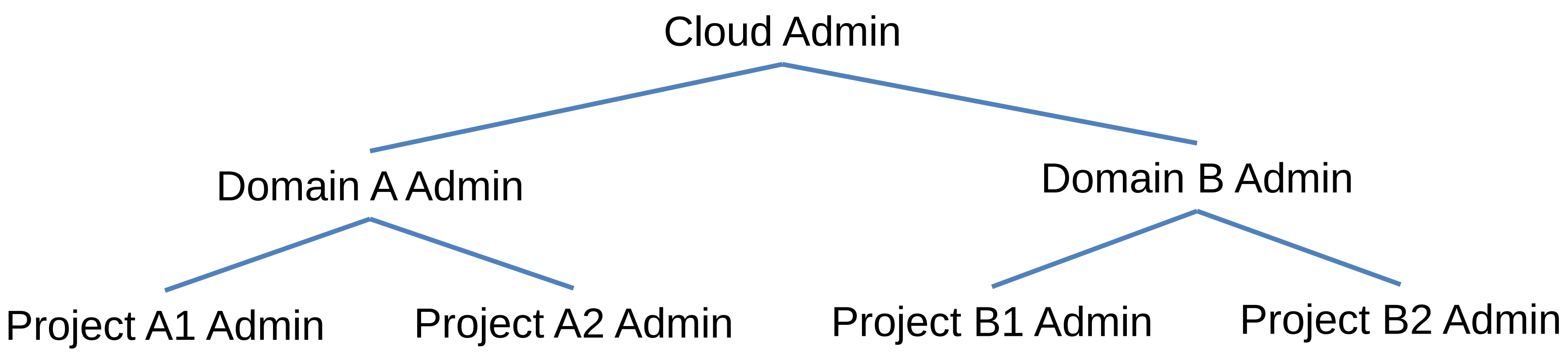


OSAC



Domain Trust
(DT)
Services
(S)
Domains
(D)
Group
Ownership
(GO)
ot_service
Groups
(G)
Group
Assignment
(GA)
Project
Ownership
(PO)
Object
Types
(OT)
Project-Role Pair
(PRP)
Permission
Assignment
(PA)
User
Group
(UG)
User
Ownership
(UO)
Projects
(P)
Roles
(R)
PRMS
User
Assignment
(UA)
Operations
(OP)
Users
(U)
token_project
token_roles
user_token
Tokens
(T)

AOSAC

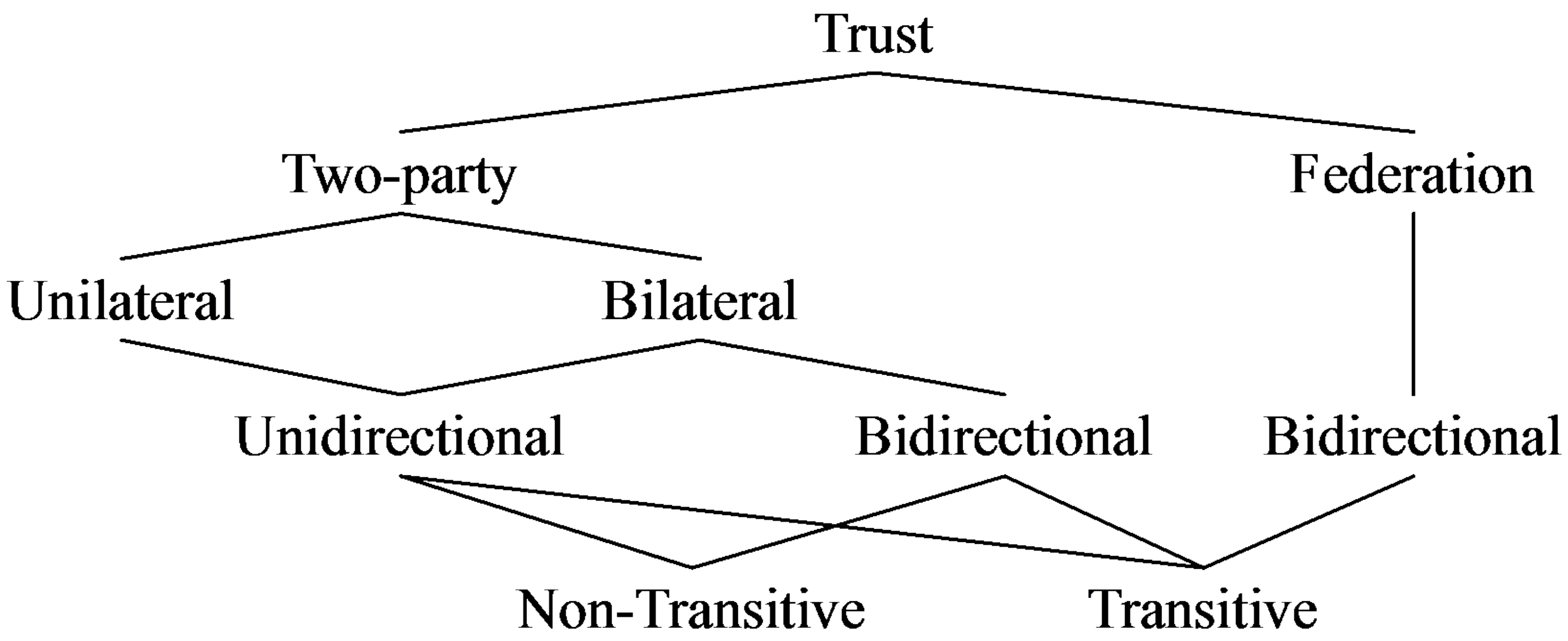



```
rule:add_user_to_tenant -> (role:keystone_admin ||
    (role:tenant_admin && tenant_id:%(target_tenant_id)s) ||
    (domain_role:domain_admin && domain_id:%(target_domain_id)s))
```

```
rule:add_tenant_to_domain -> (role:keystone_admin ||
    (domain_role:domain_admin && domain_id:%(target_domain_id)s))
```

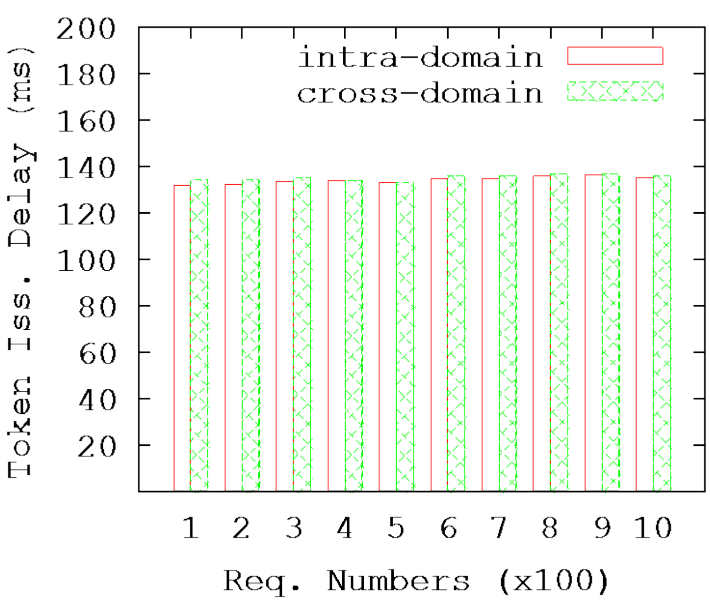
Source: https://wiki.openstack.org/wiki/Domains

Trust Framework

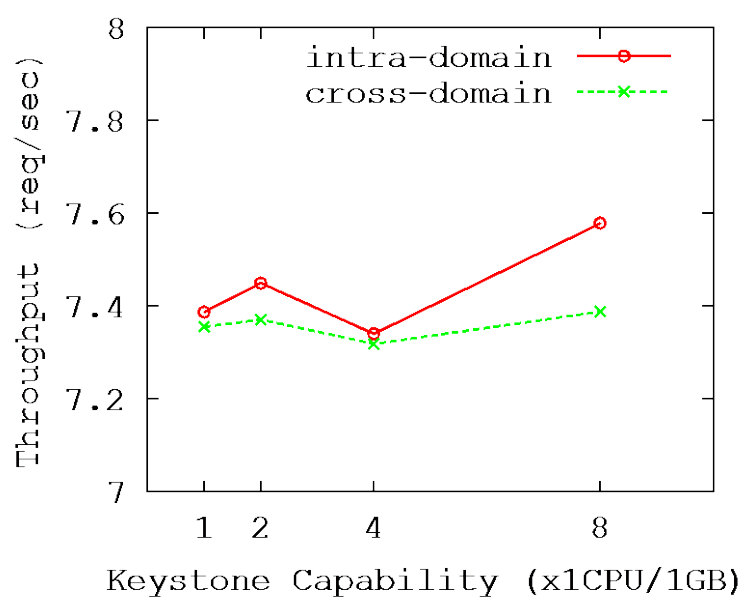


Prototype & Evaluation

- Sequential request handling (Queuing)
  - Domain trust introduces 0.7% authz. Overhead
  - Scalability changes little with domain trust



Performance



Scalability

Chapter 6: Conclusion

- Policy
  - MTAS: role-based Type-β trust
  - MT-RBAC: role-based Type-γ trust
  - CTTM: trust type taxonomy for role-based models
  - MT-ABAC: attribute-based model trusts
- Enforcement
  - MTAaaS: centralized PDP with distributed PEP
- Implementation
  - Domain Trust in OpenStack

Chapter 6: Future Work

- MT-ABAC
  - Finer-grained extensions
  - Administration, enforcement and implementation.
- More and finer-grained trust models
  - Trust negotiation and graded trust relations
- More MTAC models
  - MT-PBAC, MT-RAdAC, etc.
- Attribute-based MTAC models in OpenStack

Publications

- **Bo Tang** and Ravi Sandhu. *Extending OpenStack Access Control with Domain Trust*. In Proceedings 8th International Conference on Network and System Security (NSS), Xi'an China, October 2014.
- **Bo Tang**, Ravi Sandhu and Qi Li. *Multi-Tenancy Authorization Models for Collaborative Cloud Services*. Concurrency and Computation: Practice & Experience (CCPE), WILEY, 2014. (under review)
- **Bo Tang** and Ravi Sandhu. *Cross-Tenant Trust Models in Cloud Computing*. In Proceedings 14th IEEE Conference on Information Reuse and Integration (IRI), San Francisco, California, August 2013.
- **Bo Tang**, Qi Li and Ravi Sandhu. *A Multi-Tenant RBAC Model for Collaborative Cloud Services*. In Proceedings 11th IEEE Conference on Privacy, Security and Trust (PST), Tarragona, Spain, July 2013.
- **Bo Tang**, Ravi Sandhu and Qi Li. *Multi-Tenancy Authorization Models for Collaborative Cloud Services*. In Proc. 14th IEEE Conference on Collaboration Technologies and Systems (CTS), San Diego, California, May 2013.

Institute for Cyber Security

Q & A

Thank You!