Constraints for Attribute Based Access Control with Application in Cloud IaaS

Khalid Zaman Bijon

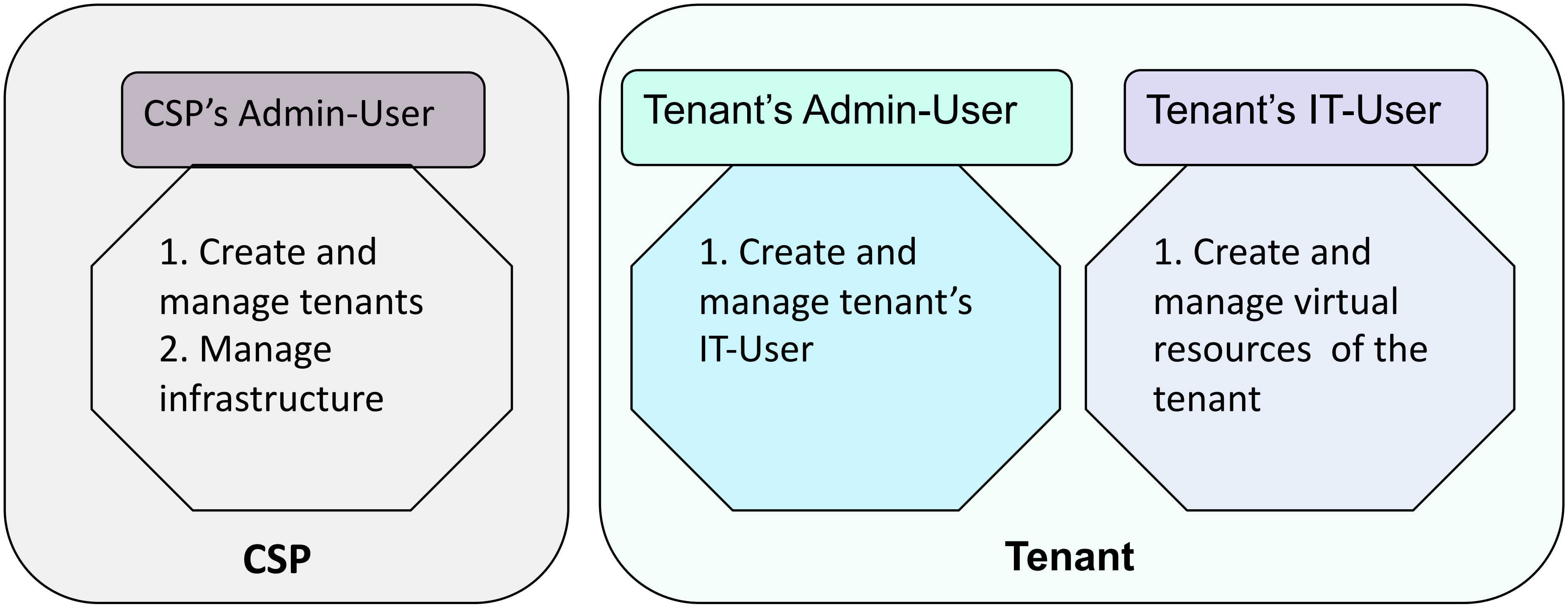
Department of Computer Science & Institute for Cyber Security
University of Texas at San Antonio

Dissertation Defense

Committee:
Dr. Ravi Sandhu (Advisor)
Dr. Ram Krishnan (Co-Advisor)
Dr. Gregory B. White
Dr. Shouhuai Xu
Dr. Weining Zhang


Its OK to have your head in the cloud,
if your feet are on the ground
- Adapted from Wilferd Arlan Peterson

Cloud IaaS



Cloud Service Provider (CSP)

- e.g., AWS, Rackspace.
- Offers virtualized computing resources to enterprises

Enterprises (Tenants)

- e.g., netflix, expedia.
- Consume virtualized computing resources

I·C·S
The Institute for Cyber Security

Cloud IaaS

UTSA

Control access of the IT-User to resources

(e.g., who can stop virtual machine vm1, who can connect virtual network vn1 to virtual machine vm1)

Received interests from academia and industry

1. Jin et. al. ABAC for cloud IaaS
2. Wu et. al. RBAC for AWS cloud

3. AWS IAM, OpenStack Keystone

Constrain the mappings between resources.

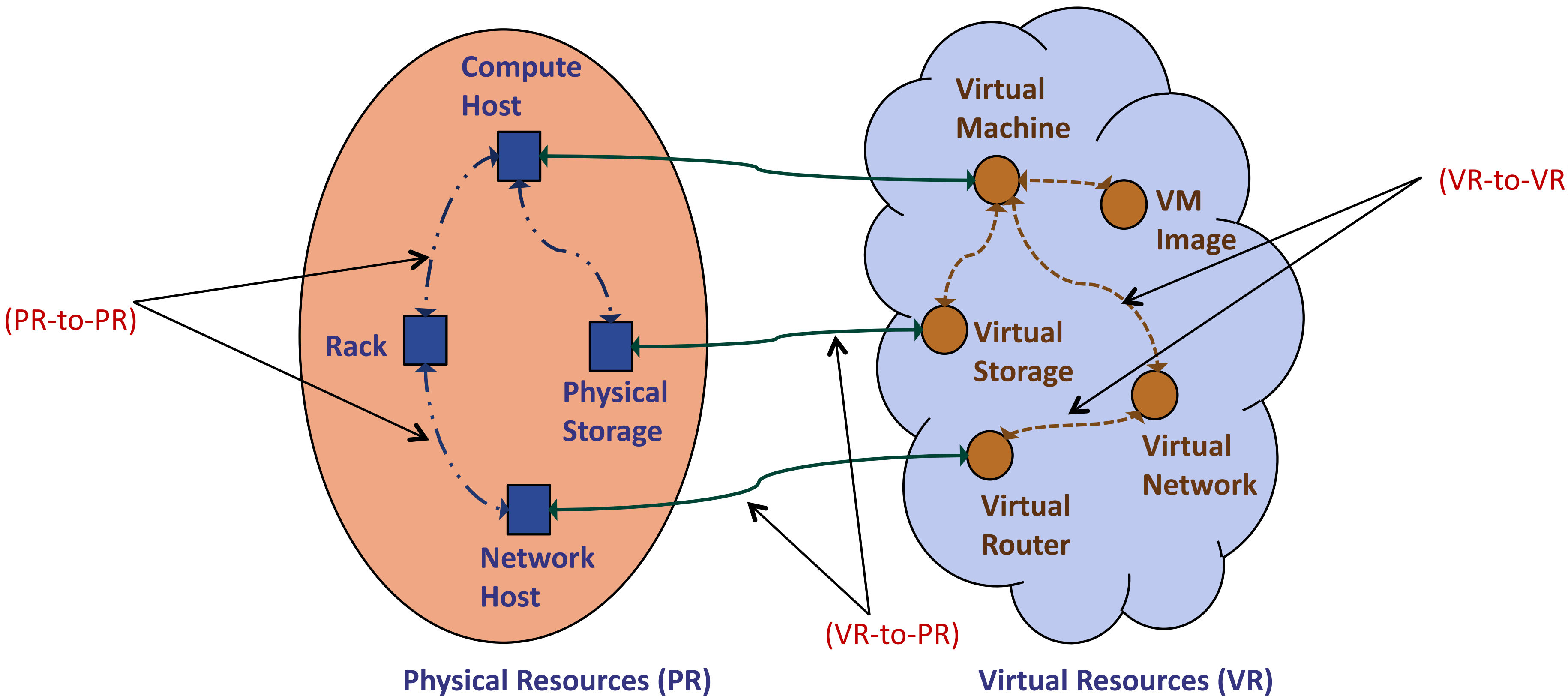
(e.g., if a virtual network vn1 can connect to the virtual machine vm2

No significant research

Mandatory Constraint

Focus of this dissertation

Problem Space



- Mapping between Resources in Cloud IaaS
- Shared Responsibility : CSP and Tenants
- Dissertation Scope: VR-to-VR and VR-to-PR Mappings

Thesis Statement

A suitably devised attribute based constraints specification mechanism can provide effective and expressive capabilities in laying out higher-level security policies for a traditional organization that exercises attribute based access control as well as for the mapping configuration management of virtual resources in cloud infrastructure-as-a-service.

Outline

1. Constraints for VR-to-VR Mapping

2. Constraints for VR-to-PR Mapping

3. Constraints for Attribute Based Access Control

Outline

1. Constraints for VR-to-VR Mapping

- Constraint Specification and Enforcement
- Automated Constraint Construction

2. Constraints for VR-to-PR Mapping

3. Constraints for Attribute Based Access Control

Paper

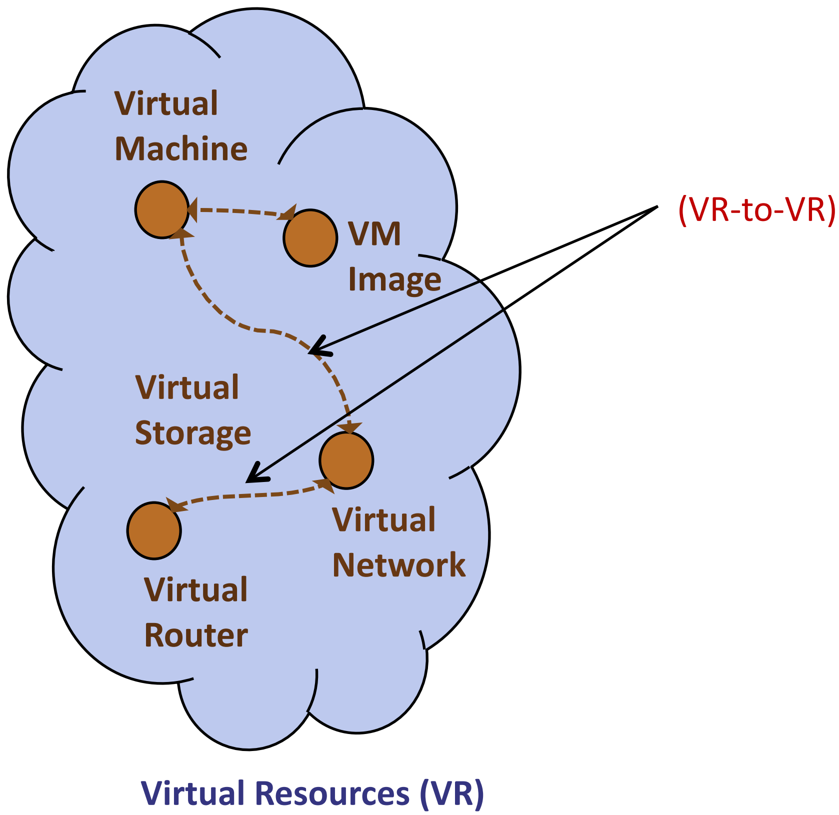
1. **Khalid Bijon, Ram Krishnan, and Ravi Sandhu.**

Virtual Resource Orchestration Constraints in Cloud Infrastructure as a Service. ACM CODASPY'15.

2. **Khalid Bijon, Ram Krishnan, and Ravi Sandhu.**

Automated Constraints Construction in Cloud Infrastructure as a Service. Under Preparation (will be submitted to IEEE TDSC).

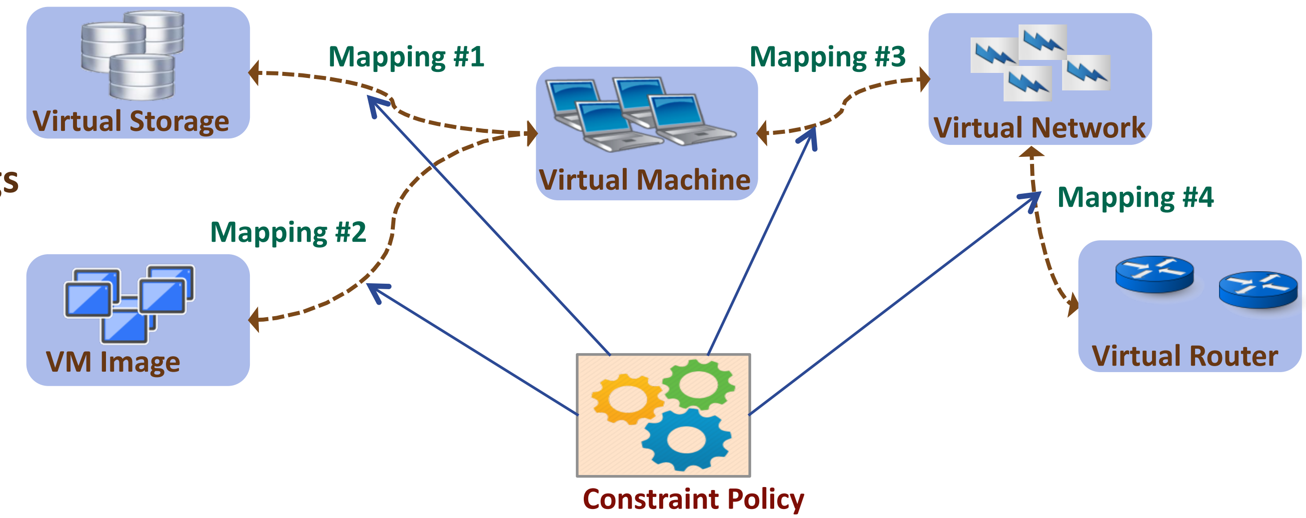
VR-to-VR Mapping



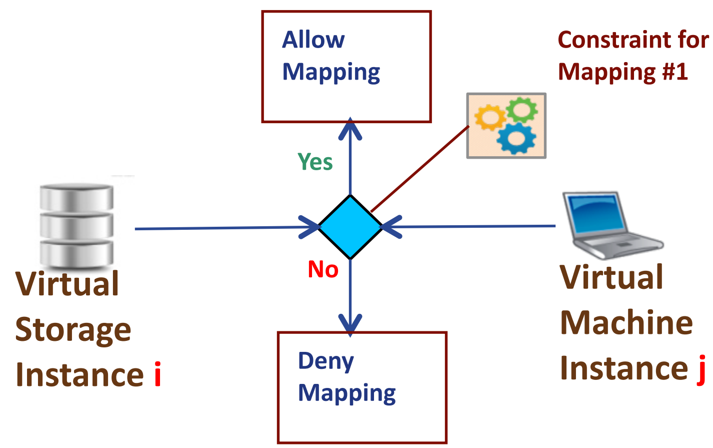
- Complex Management Process
- Scope: Intra-Tenant
- Goal: Diversity of Tenant

Constraint Specification and Enforcement

- **Constraint Policy**
  - For each VR-to-VR mappings



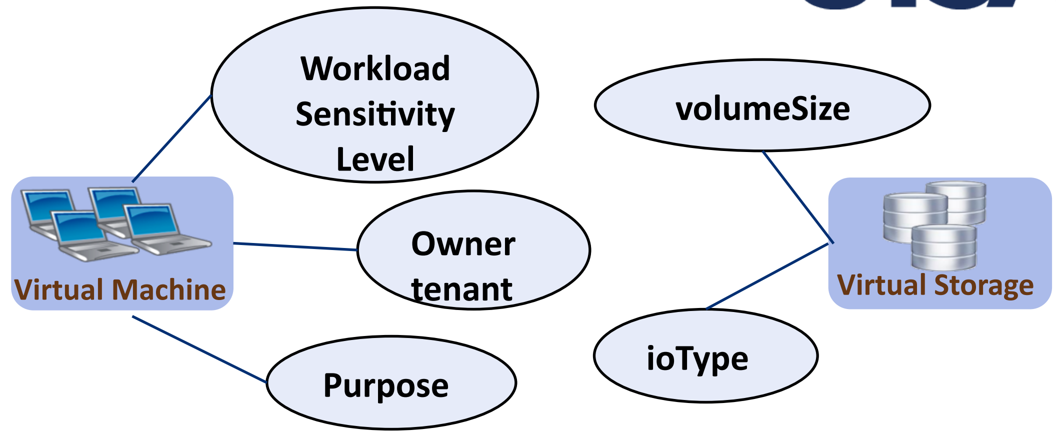
- **Satisfied By**
  - Individual virtual resources



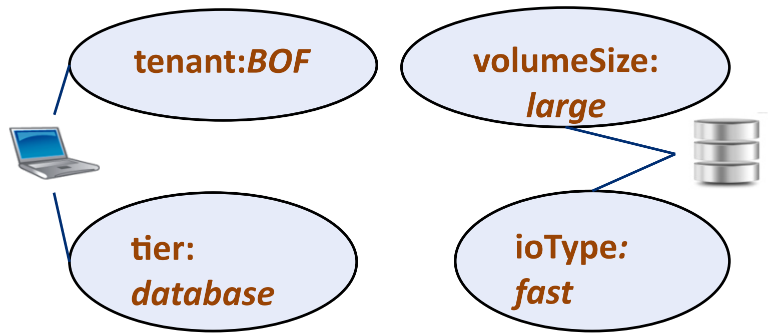
Credit: www.iconarchive.com

An Attribute Based Approach

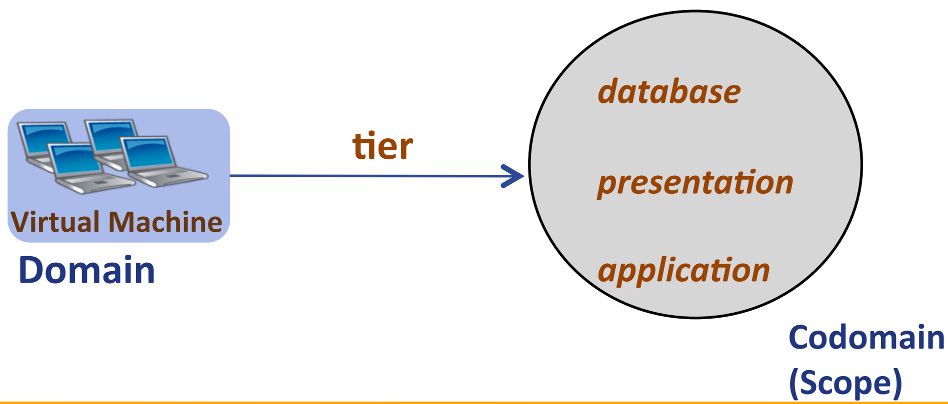
- **Attribute Specifies Virtual Resource Properties**



- **A name:*value* Pair**

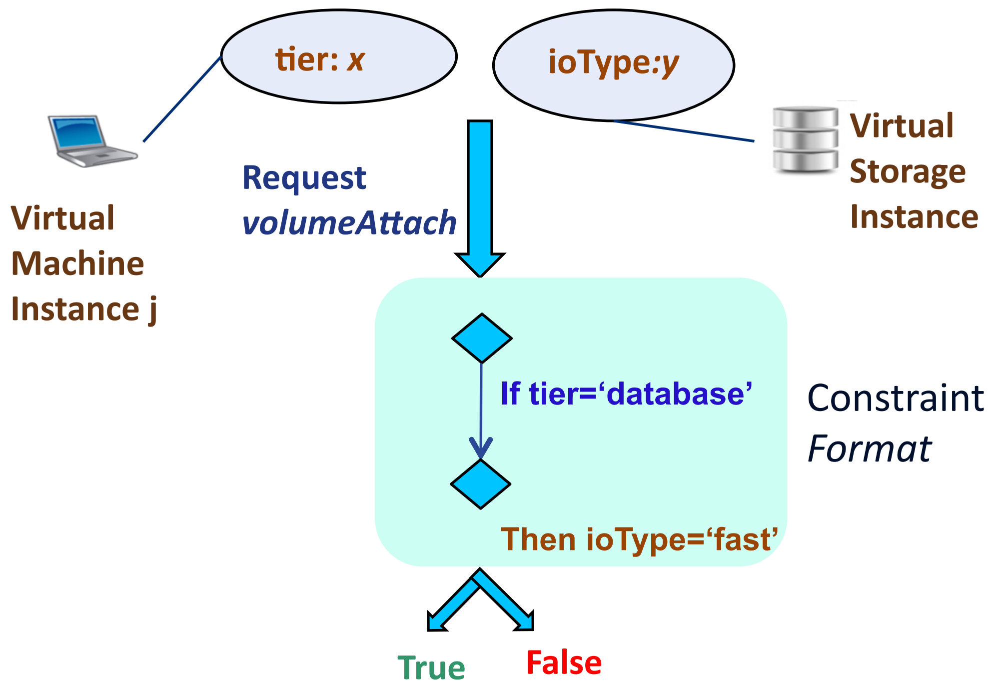


- **Designed as Functions**

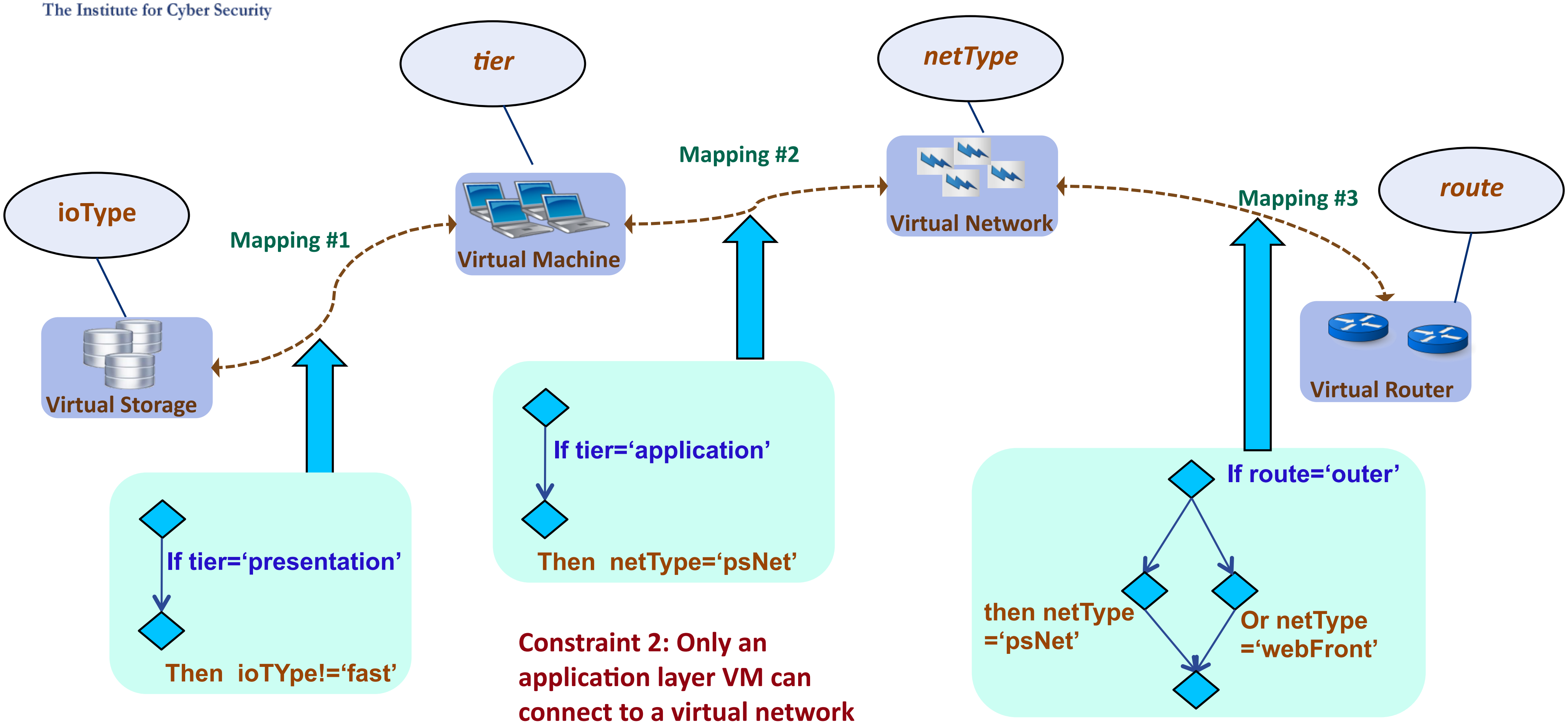


Constraint Policy

- A Constraint
  - Logical Formula
  - Compares Certain Attribute Values



Use Case (3-Tier System)

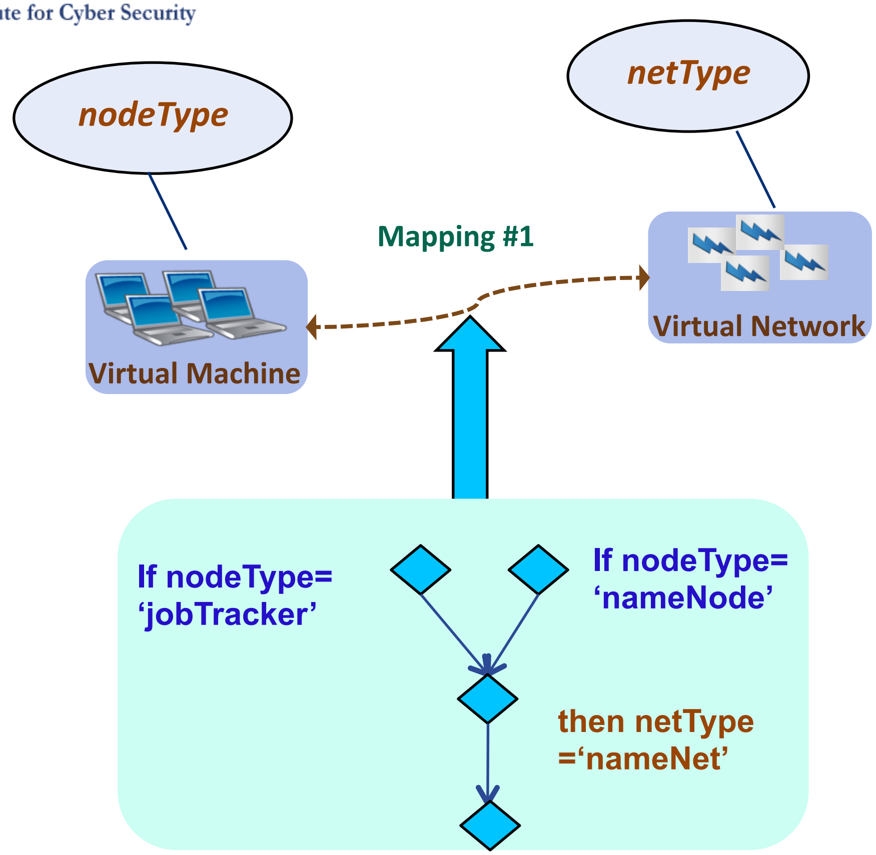


Constraint 2: Only an application layer VM can connect to a virtual network which is created for passing application layer data.

Constraint 1: If a VM is for presentation layer, attaching storage’s ioType cannot be fast.

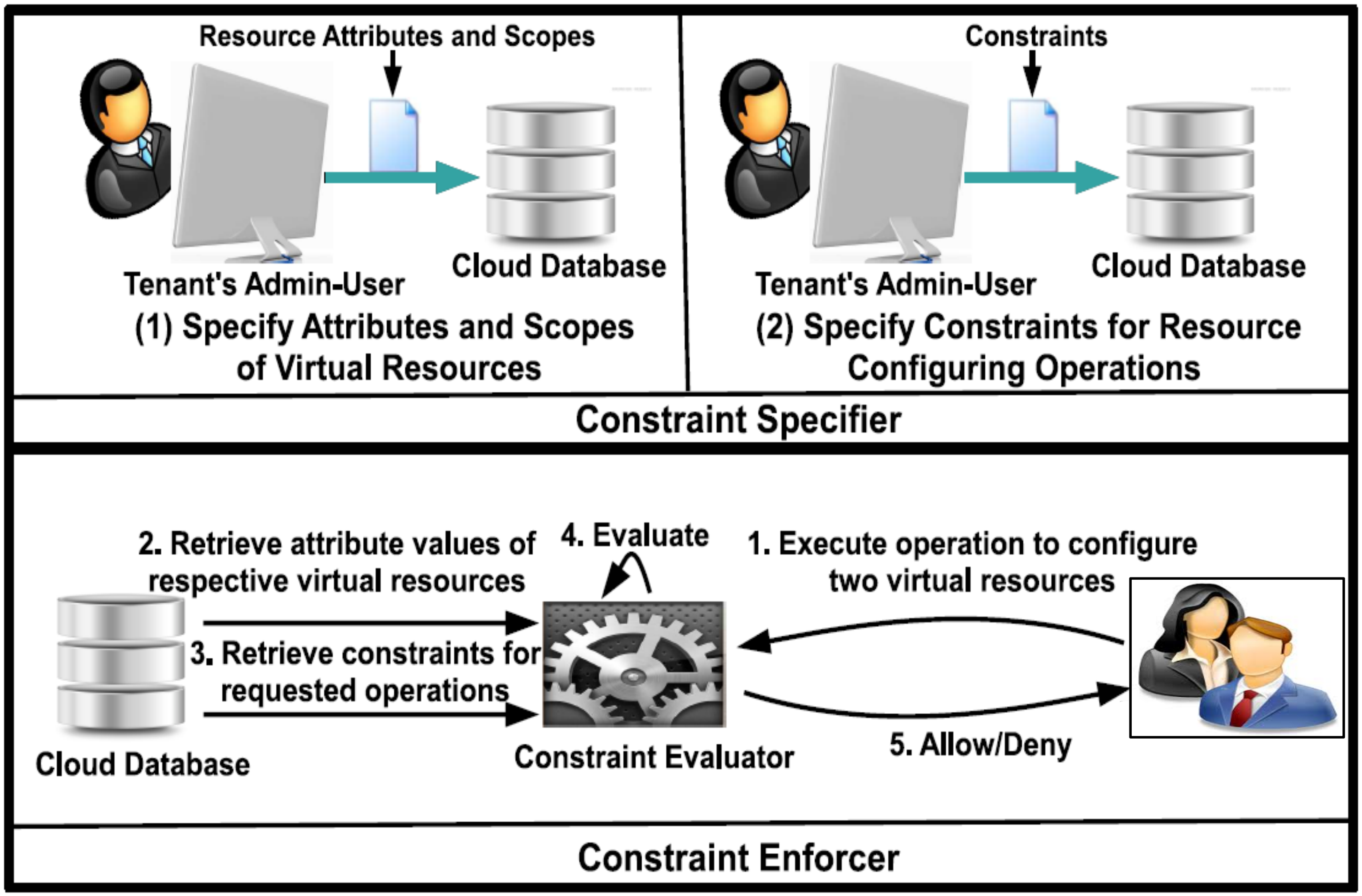
Constraint 3: If a router is for connecting to out-side internet, only presentation layer network or web-fornt network can connect to it.

Use Case (Hadoop)



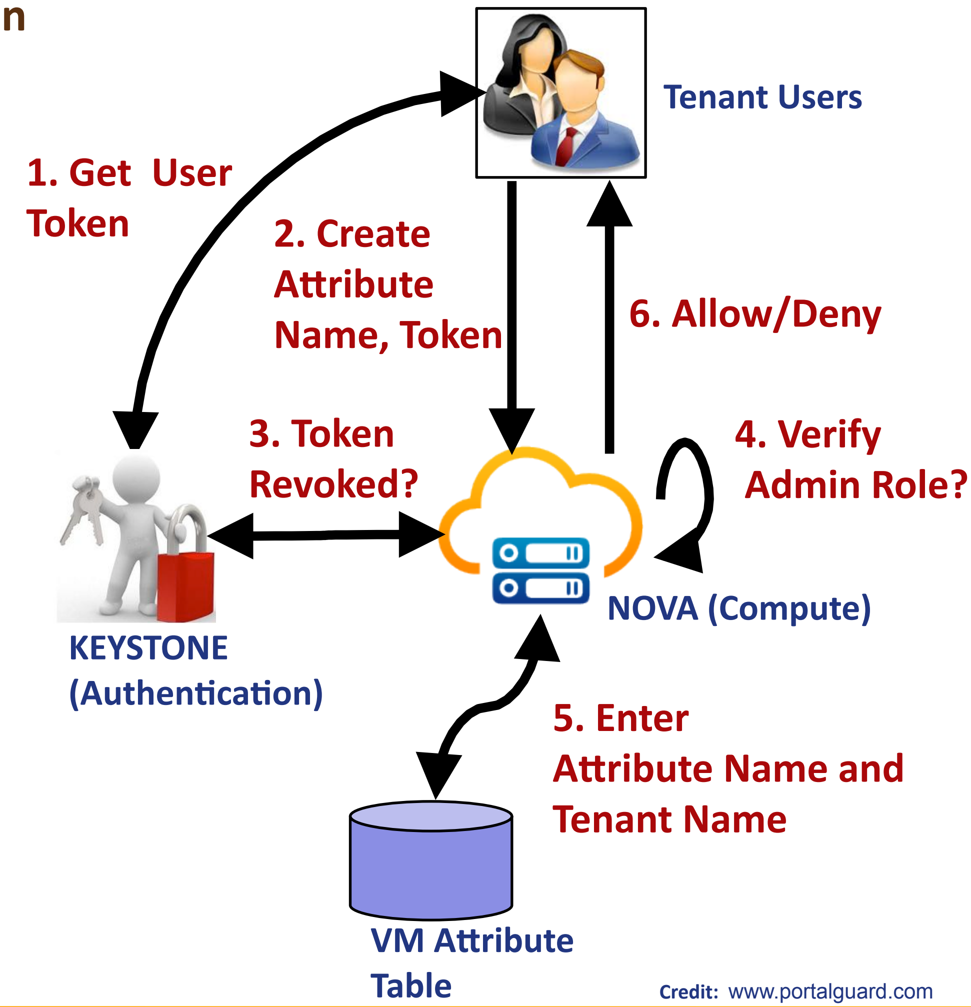
Constraint 1: Only jobTracker and nameNode VMs can connect to a network created for passing data to/from name Nodes.

Specify and Enforce



Specifier

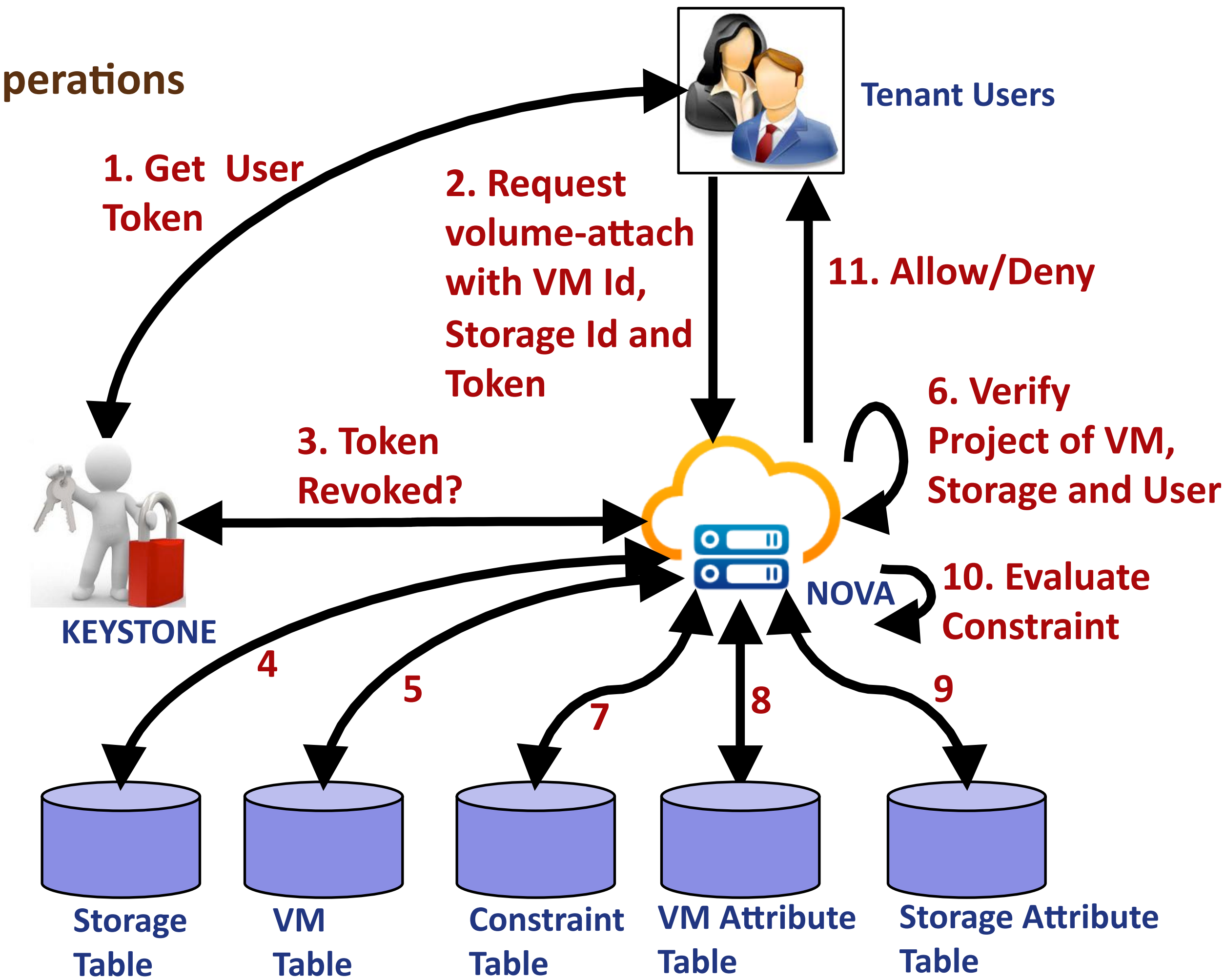
- Implemented in OpenStack
- Execution of “*attribute-creation*” operation
- Similarly,
  - Attribute-value specification
  - Constraint Specification
  - Attribute-value assignment



Credit: www.portalguard.com

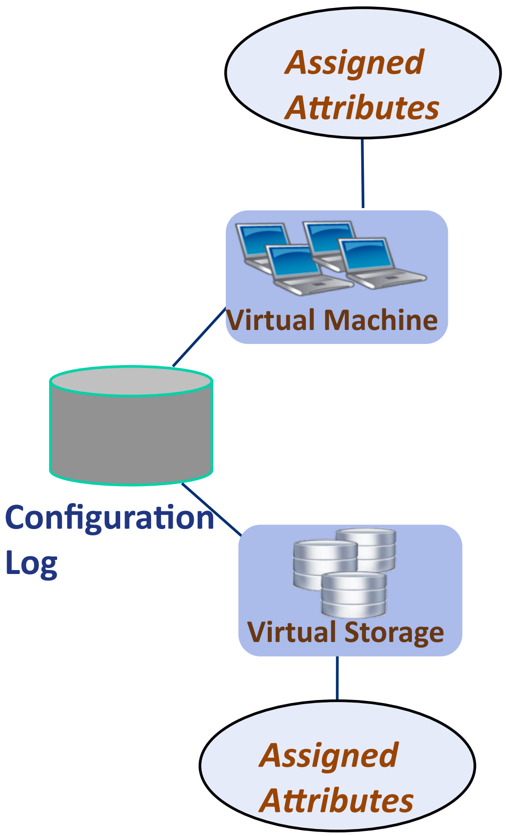
Enforcer Implementation

- Implemented in OpenStack
- A Constraint Parser
- Invoked by Resource Mapping Operations (e.g., *volume-attach*)

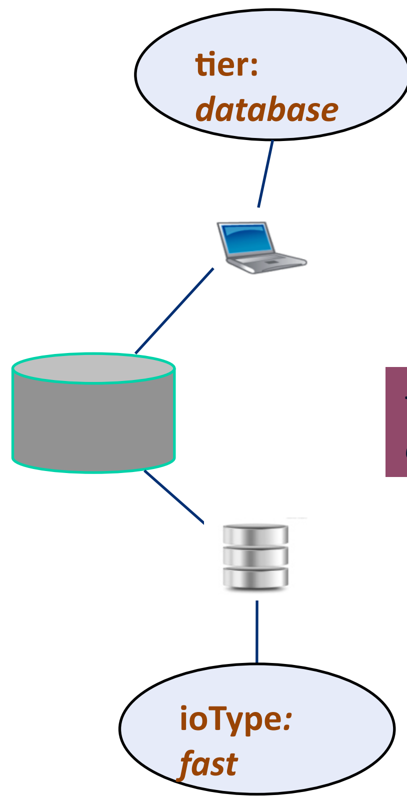


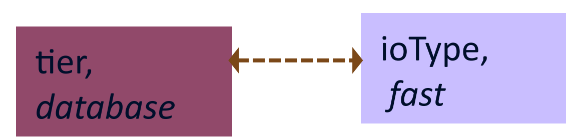
Automated Constraint Construction

- Helps the tenants to find policy
- From Previous Configurations



- Construct Relation between values of two attributes

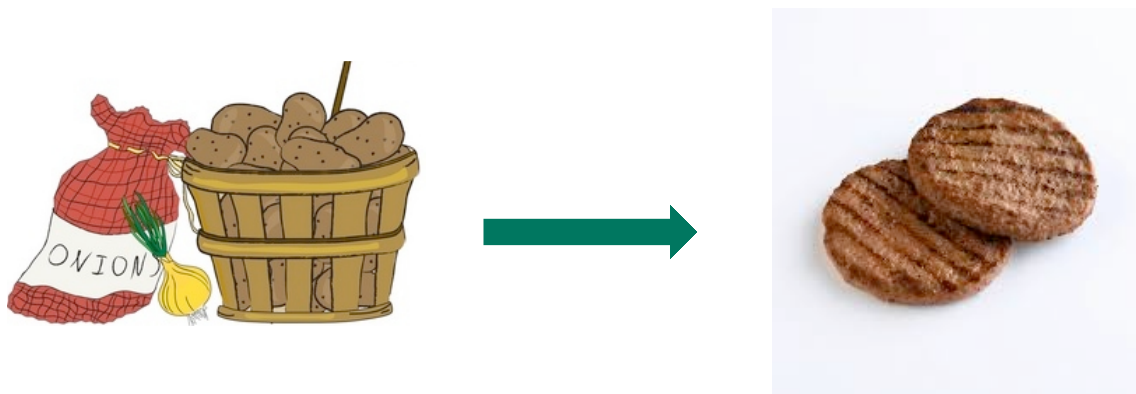




Credit: www.iconarchive.com

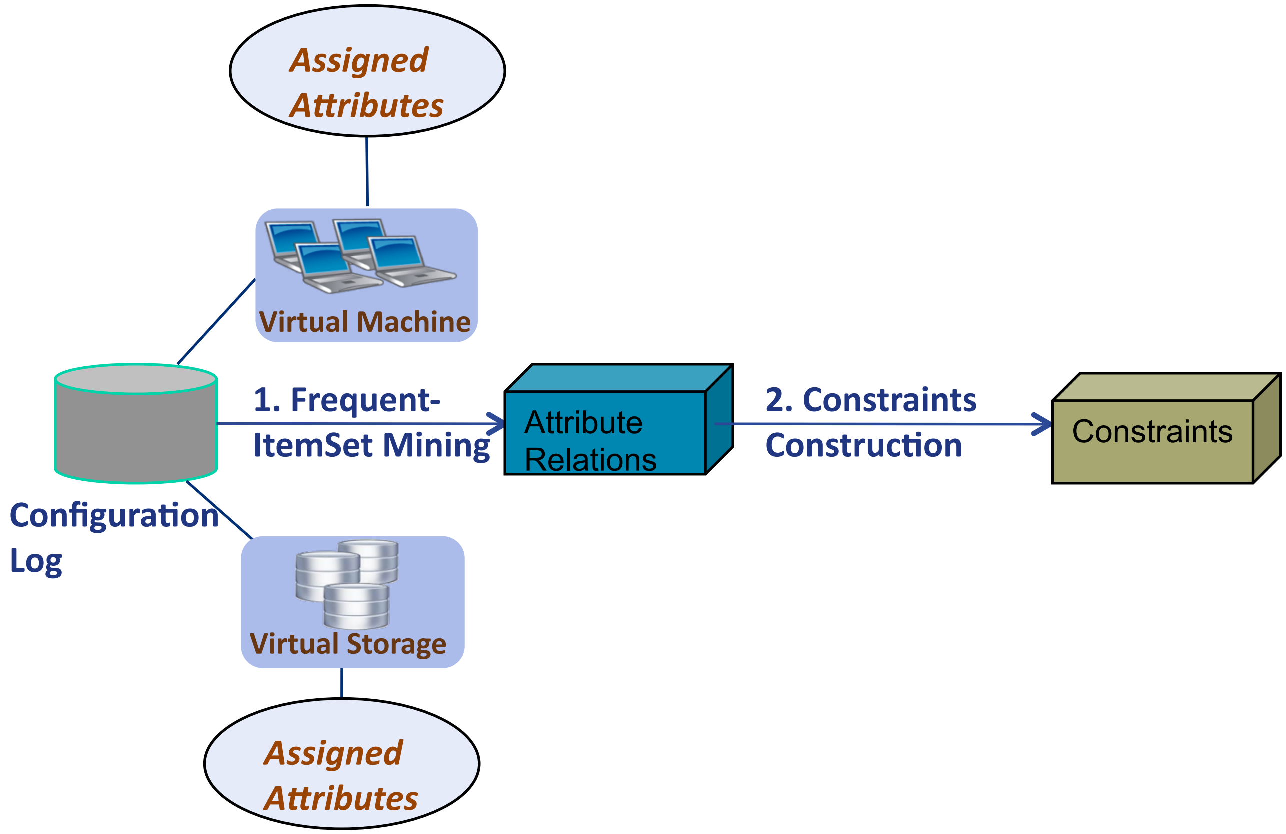
Automated Constraint Construction

- **Association Rule Mining (Frequent-Itemset Mining)**
  - **relations between variables in large databases**



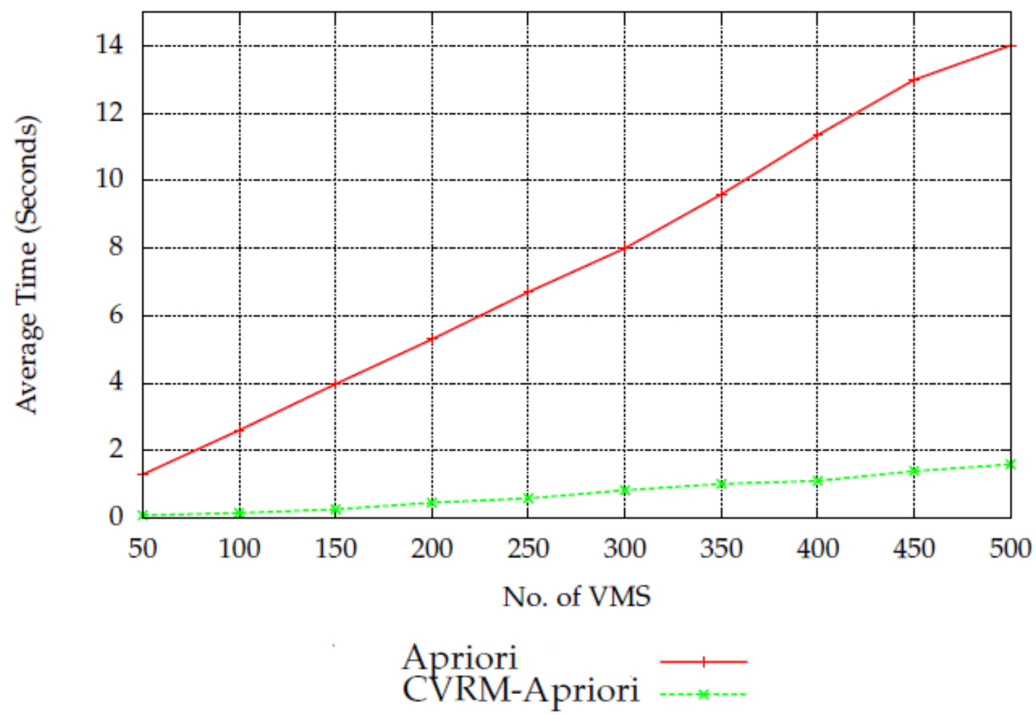
- **Apriori Algorithm**
  - **Consider relations between all combination of values**
- **With customization for cloud IaaS (CVRM-Apriori)**
  - **Only consider relations between every pair of values of two attribute**

CVRM-Apriori



Evaluation

- Policy for VM-Network Connectivity Mapping
- From VM-Network Table (table *virtual_interfaces* in Nova, OpenStack)
- 10 Attributes each with 10 values
- 10 Virtual Networks
- At least three Networks per VM
- Mine relations between every pair of attribute values



Outline

1. Constraints for VR-to-VR Mapping

2. Constraints for VR-to-PR Mapping

- Conflict-Free Virtual Resource Scheduling
- Physical Resource Optimization
- Experimental Analysis

3. Constraints for Attribute Based Access Control

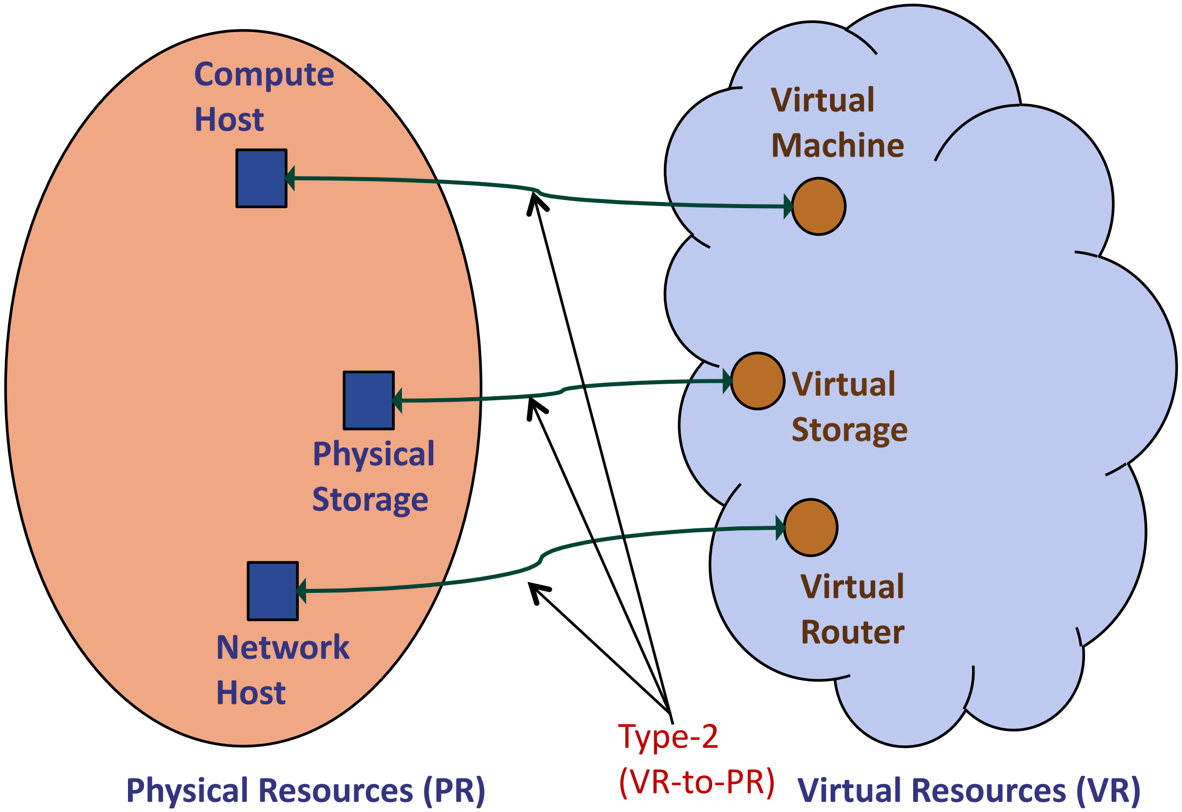
Paper

Khalid Bijon, Ram Krishnan and Ravi Sandhu.

Mitigating Multi-Tenancy Risks in IaaS Cloud Through Constraints-Driven Virtual Resource Scheduling.

ACM SACMAT'15.

Problem Space



- **Shared Responsibility: CSP and Tenant**
- **Tenant: Control Placement of Virtual Resource**
- **CSP: Optimize the Physical Resources**

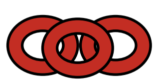
Goal

- Restrict VR-to-PR Mapping
  - For security and performance

- Security Example (DoD Cloud)
  - Should not co-locate conflicting vms to same server
  - E.g., VM processing top-secret for Navy might not want to co-locate with top-secret Air Froce

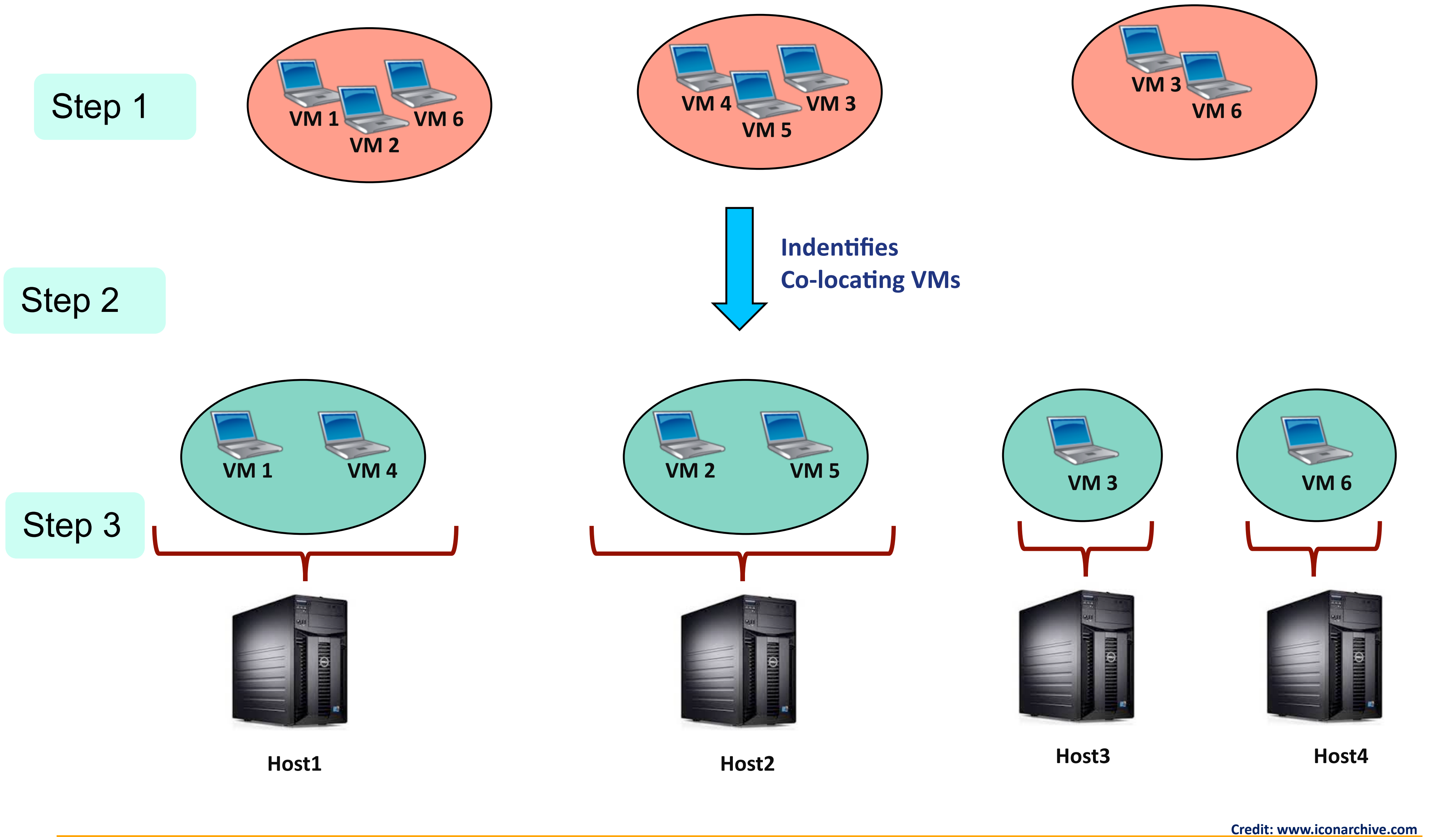
- Host Optimization
  - Increase host utilization

- Scope
  - Focus on virtual machine to compute host mapping
  - Anti-Affinity (Must-not co-locate)



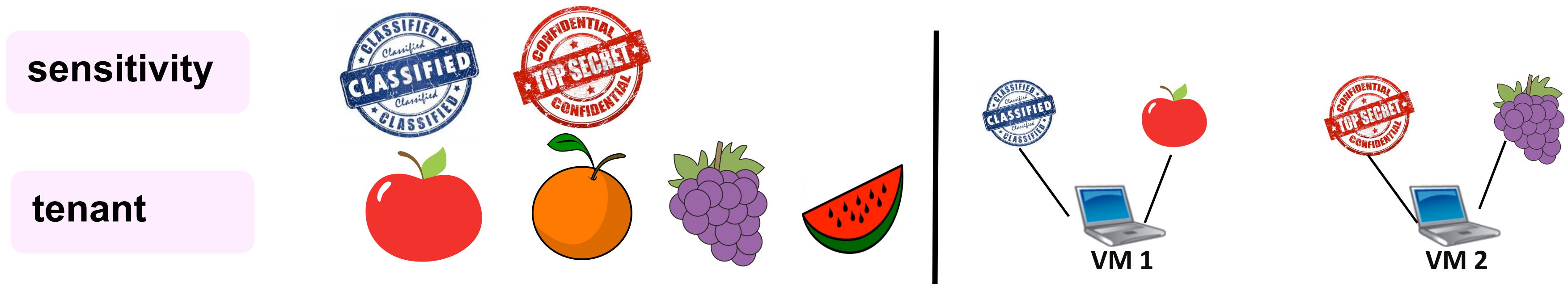
Credit:
www.bartley.hants.sch.uk
www.opsrules.com

Process


Step 1
VM 1
VM 2
VM 6
VM 4
VM 5
VM 3
VM 3
VM 6
Step 2
Indentifies
Co-locating VMs
Step 3
VM 1
VM 4
VM 2
VM 5
VM 3
VM 6
Host1
Host2
Host3
Host4
Credit: www.iconarchive.com

An Attribute Based Approach

- **Attribute Specifies Virtual Resource Properties**



- **Attribute-based conflict-free Virtual Machine Scheduling**
  - **Specify conflicts between values of attributes**

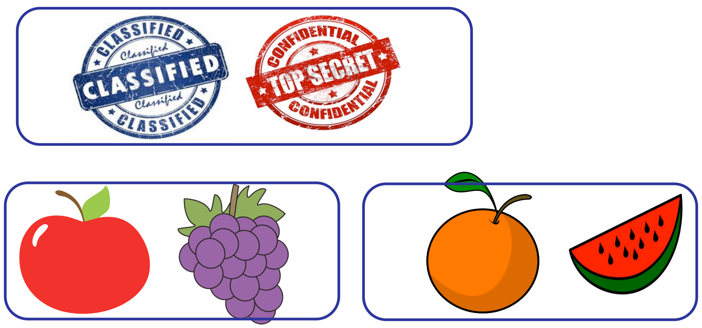
Credit: www.iconarchive.com

Conflict-free VM Scheduling

- **Step 1:** Specify Conflicts among attribute values of each attribute

Conflict Set Sensitivity

Conflict Set Tenant



- **Step 2:** Create Conflict-free partitions of the values of each attribute

Partition Sensitivity

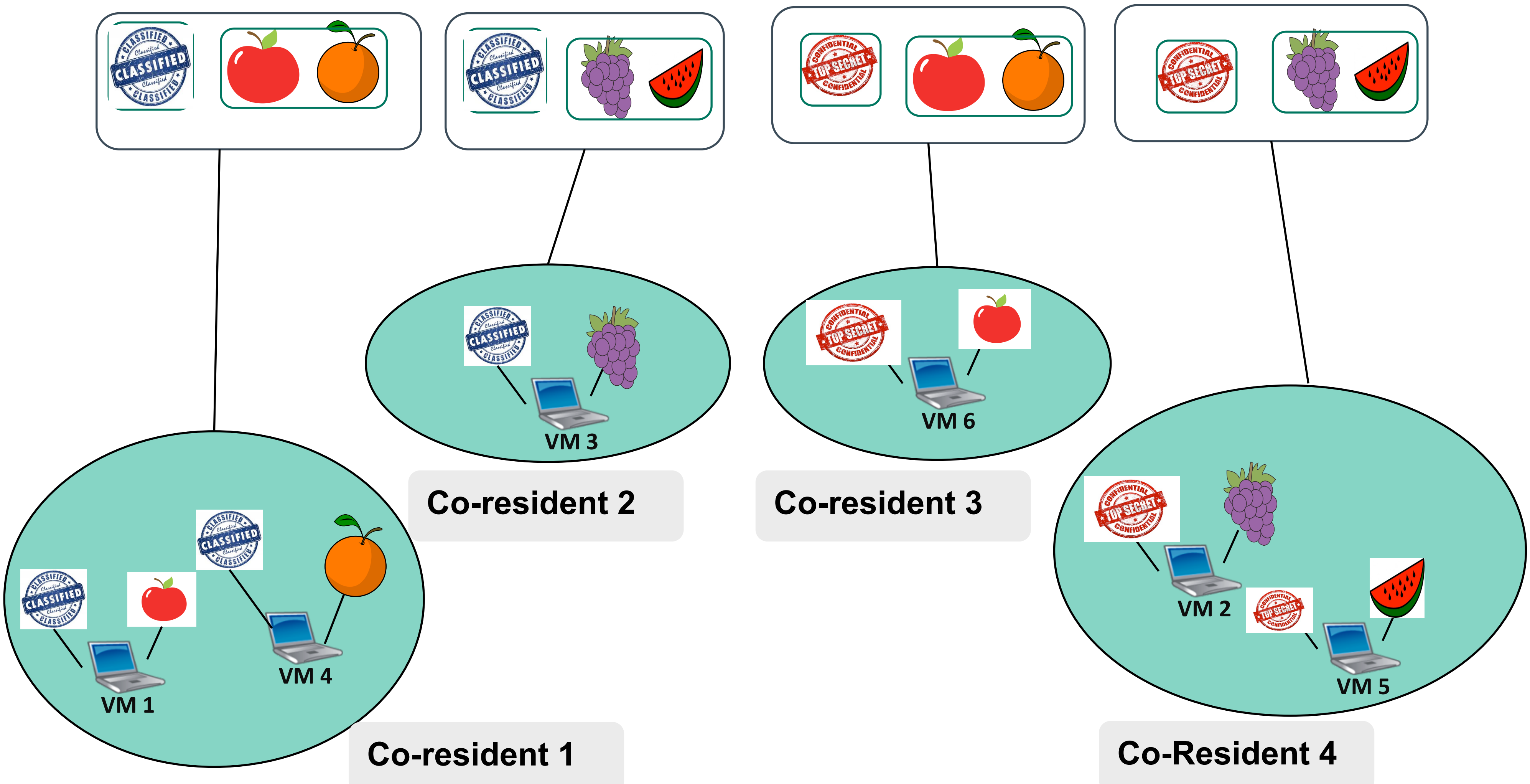
Partition Tenant

- **Step 3:** Create Conflict-free Segments (each segment contains an element of the conflict-free partition of each attribute)

Credit: www.iconarchive.com

Conflict-free VM Scheduling

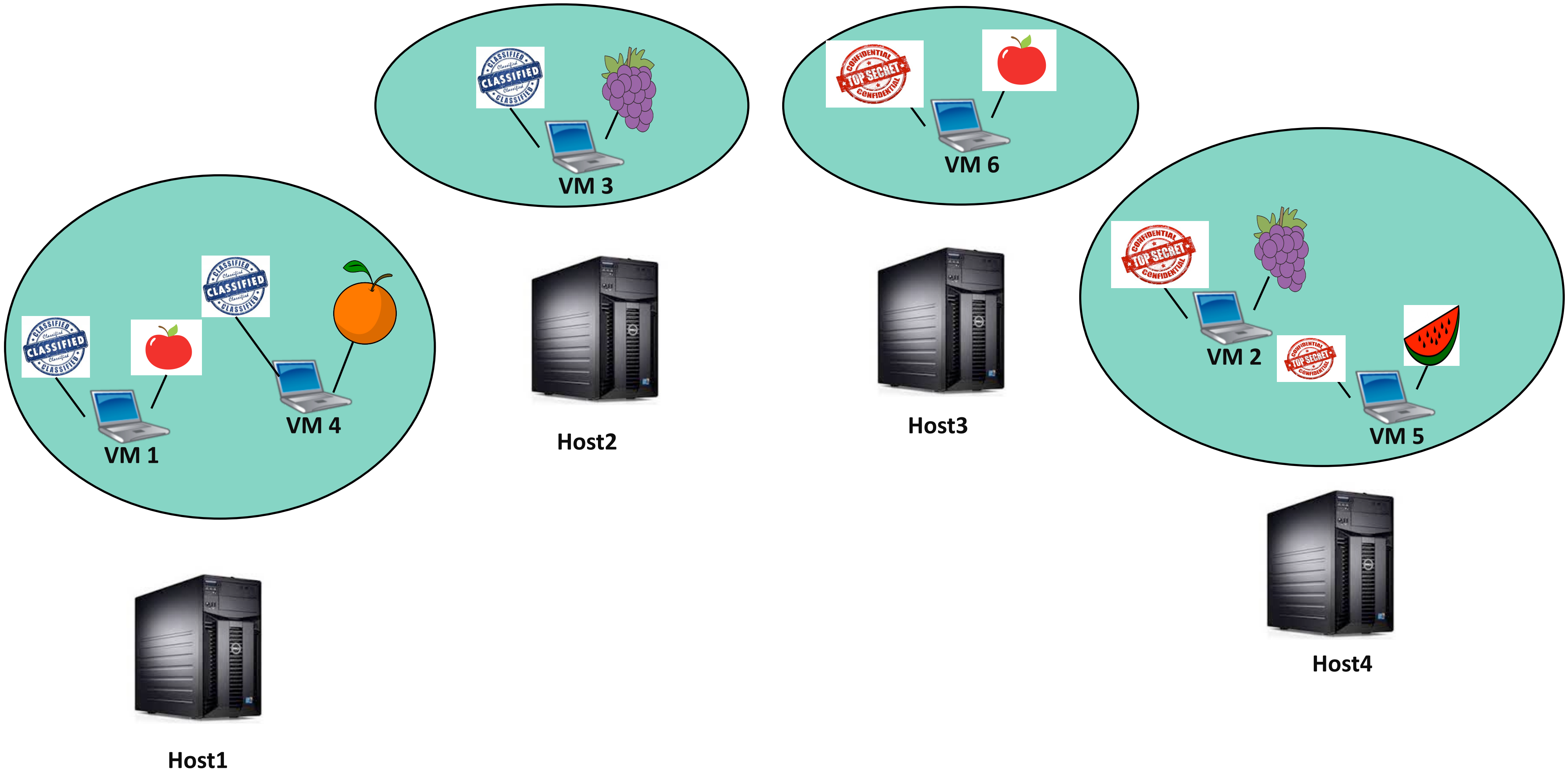
- Step 4: Create VM partition that can co-reside



Credit: www.iconarchive.com

Conflict-free VM Scheduling

- Step 5: Allocate Separate Hosts for each VM Partition



Credit: www.iconarchive.com

Outline

1. Constraints for VR-to-VR Mapping

2. Constraints for VR-to-PR Mapping

- Conflict-Free Virtual Resource Scheduling
- **Physical Host Optimization**
- Experimental Analysis

3. Constraints for Attribute Based Access Control

Physical Host Optimization

- Step 1: Specify Conflicts among attribute values
- Step 2: Create Conflict-free partitions (Crucial)
  - Minimum number of conflict-free partitions
  - Minimum number of conflict-free segments
  - Minimum number of VM partitions
- Step 3: Create Conflict-free Segments
- Step 4: Create VM partition that can co-reside
- Step 5: Allocate Separate Hosts for each VM Partition

- Optimization Problem:
  - Input-conflicts among values of an attribute
  - Output-minimum number of partitions

- K-Partition:
  - Input-conflicts among values and K
  - Output-if there is K number of partitions

K-Partition

I·C·S The Institute for Cyber Security

Physical Host Optimization

UTSA

- **K-Partition is NP-Complete**
  - **Reduction from k-coloring**

K-Partition

$\leq_p$

K-Coloring

- **Approximation Algorithms for Graph Coloring can Apply**

- **Develop an Exact Algorithm (Backtracking)**
  - **Useful for small number of attribute-values**

Outline

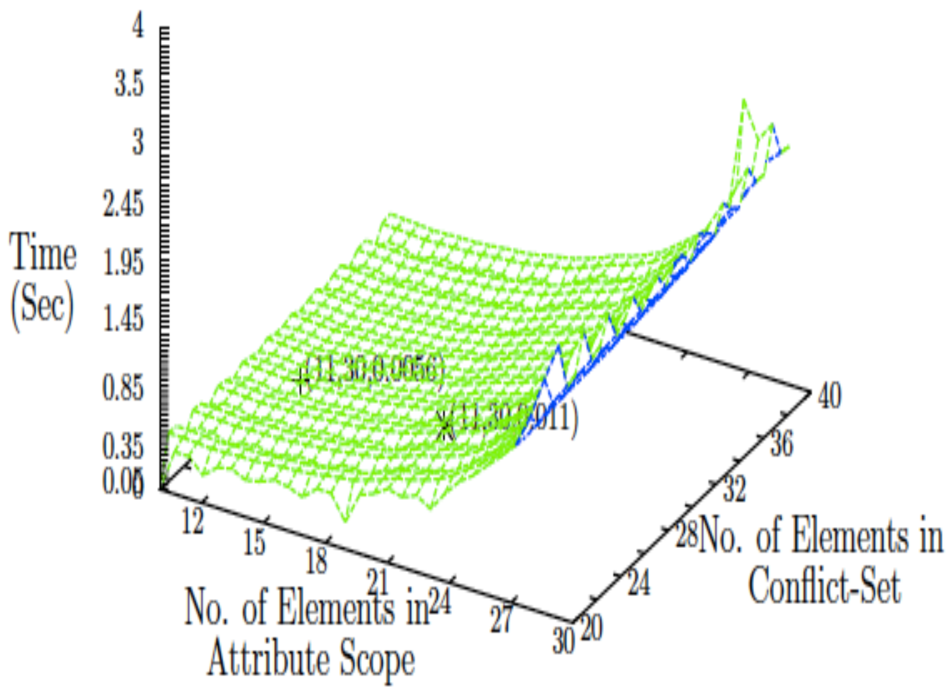
1. Constraints for VR-to-VR Mapping

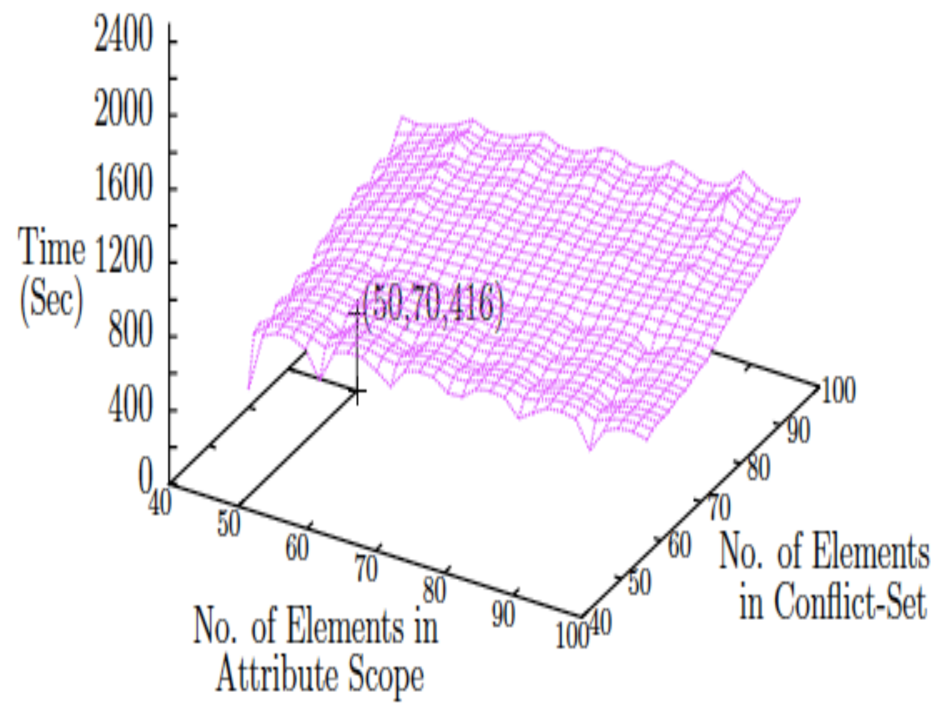
2. Constraints for VR-to-PR Mapping

- Conflict-Free Virtual Resource Scheduling
- Physical Host Optimization
- Experimental Analysis

3. Constraints for Attribute Based Access Control

1. Performance of Backtracking algorithm

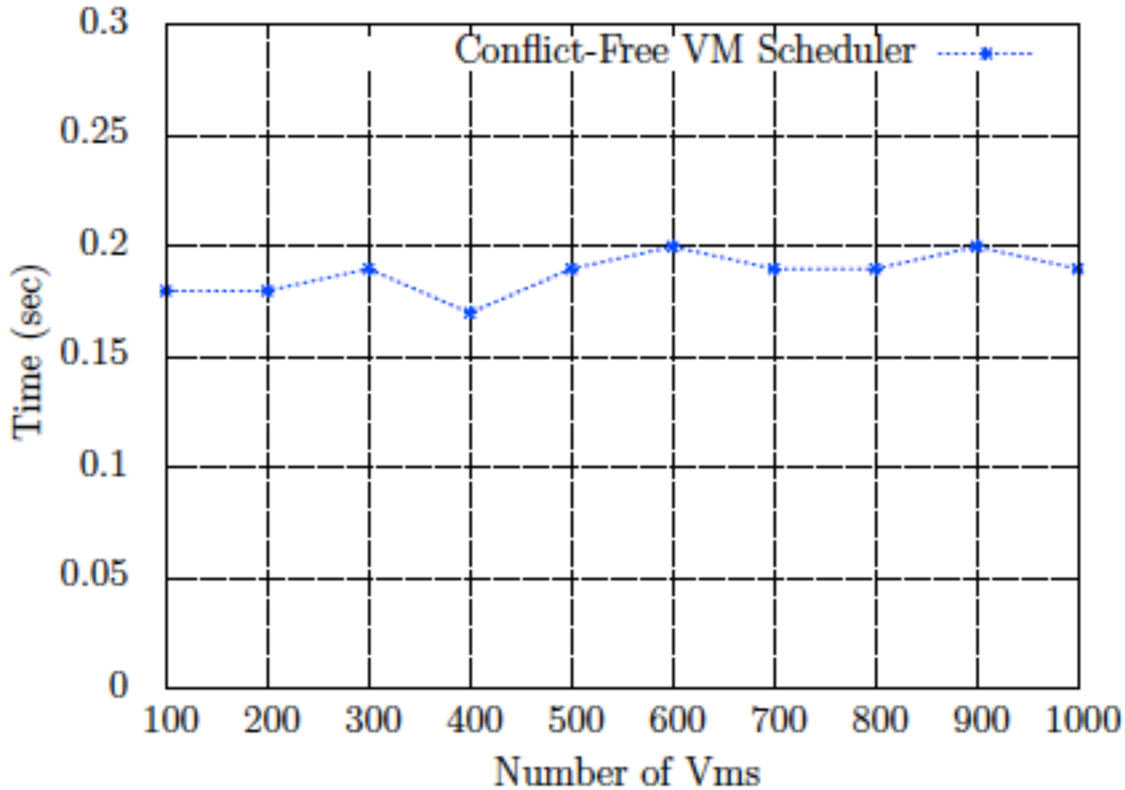




Required Time for Small Scope and Conflict-Set

Required Time for Large Scope and Conflict-Set

2. Scheduling Latency



Less than 0.2 seconds for scheduling (once the conflict-free partitions are created)

Outline

1. Constraints for VR-to-VR Mapping

2. Constraints for VR-to-PR Mapping

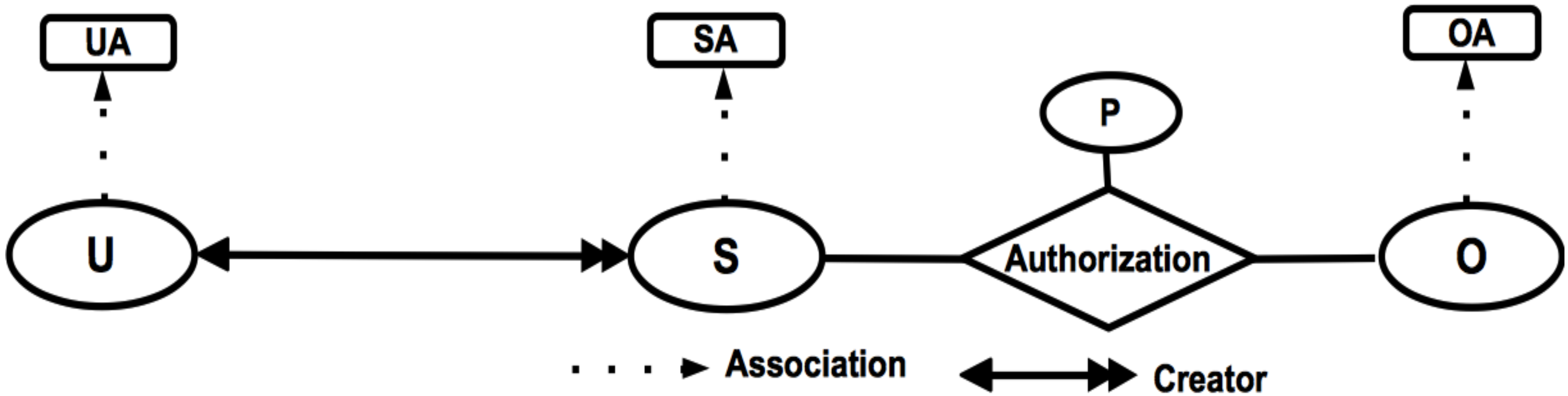
3. Constraints for Attribute Based Access Control

Paper

1. Khalid Bijon, Ram Krishnan, and Ravi Sandhu.
   Towards An Attribute Based Constraints Specification Language.
   IEEE PASSAT'13.

2. Khalid Bijon, Ram Krishnan, and Ravi Sandhu.
   Constraint Specification in Attribute Based Access Controls.
   ASE Science Journal'13.

Overview of an ABAC Model



- **Basic Entities**
  - User (U), Subject (S) and Object (O)
  - Their Attributes (UA, SA, OA)
- **Attribute can be atomic or set valued (in cloud IaaS it was only atomic value)**
  - e.g., clearance vs. role
- **Permission has Authorization policy**
  - Verify subject and object attributes

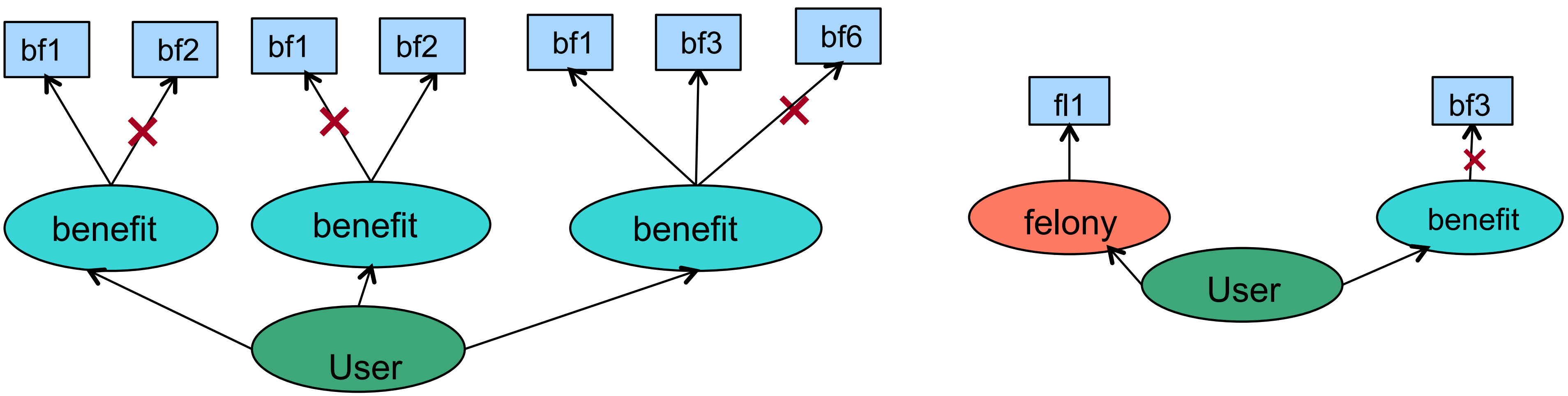
Motivation

- ABAC is policy neutral
  - Subject with required attribute can access
- Proper attribute assignment to the entities
  - Need to ensure authorized access
- Constraints for the attribute assignment
  - Verify subject and object attributes
  - configure high level security policy

Attribute Based Constraints Specification

- Develop an attribute based constraints specification language (ABCL)
  - Identify relation between values (of same attribute or across attributes)

    (across attribute (VR-to-VR) and same attribute (VR-to-PR))
  - A relation restricts an entity to get certain values of an attribute.
    - *Benefit* attribute represents customers' assigned benefits in a Bank
    - A customer cannot get both *benefits* 'bf1' and 'bf2' (mutual exclusion)
    - Cannot get more than 3 benefits from 'bf1', 'bf3' and 'bf6' (cardinality on mutual exclusion)



Attribute Based Constraints Specification

- A mechanism to represent different types of such relationships as a set

1. Mutual-Exclusive relation of the ***benefit*** attribute values (single attribute conflict)

Attribute_Set$_{U,benefit}$ *UMEBenefit*
UMEBenefit={avset1, avset2} where
avset1=({'bf1','bf2'}, 1) and
avset2=({'bf1','bf3','bf4'}, 2)

2. Mutual-Exclusive relation of the ***benefit*** *and* ***felony*** (cross attribute conflict)

Cross_Attribute_Set$_{U,Aattset,Rattset}$ *UMECFB*
Here, *Aattset*= {felony} and *Rattset*= {benefit}
UMECFB={attfun1} where
attfun1(felony)=(attval, *limit)*
where *attval={'fl1',* 'fl2'} and *limit=1*
attfun1(benefit)=(*attval, limit)*
where *attval={'bf1'}* and *limit=0*

ABCL Syntax

- A grammar in Backus Normal Form (BNF)
  - Declaration of the Attribute_Set and Cross_Attribute_Set
  - Constraint Expression

Declaration of the *Attribute_Set* and *Cross_Attribute_Set*:

<attribute_set_declaration> ::= <atribute_set_type> <set_identifier>

<attribute_set_type> ::= ***Attribute_Set***$_{U,<\mathrm{attname}>}$ | ***Attribute_Set***$_{S,<\mathrm{attname}>}$ | ***Attribute_Set***$_{O,<\mathrm{attname}>}$

<cross_attribute_set_type> ::= ***Cross_Attribute_Set***$_{U,<Aattset>,<Rattset>}$ | ***Cross_Attribute_Set***$_{S,<Aattset>,<Rattset>}$
| ***Cross_Attribute_Set***$_{O,<Aattset>,<Rattset>}$

<*Aattset*> ::= {<attname>, <attname>*}

<*Rattset*> ::= {<attname>, <attname>*}

<set_identifier> ::= <letter> | <set_identifier><letter> | <set_identifier><digit>

<digit> ::= *0|1|2|3|4|5|6|7|8|9*

<letter> ::= *a|b|c|....|x|y|z|A|B|C|...|X|Y|Z*

Constraint Expressions:

<statement> ::= <statement> <connective> <statement> | <expression>

<expression> ::= <token> <atomiccompare> <token> | <token> <atomiccompare> <size>
| <token> <atomiccompare>|<set>| | <token> <atomiccompare> <set> | <token>

<token> ::= <token> <setoperator> <term> | <term> | |<term>|

<term> ::= <function> (<term>) | <attributefun> (<term>) | **OE** (<relationsets>).<item>
| **OE** (<term>) | **OE** (<set>) | **AO** (<term>) | **AO** (<set>) | <attval>

<connective> ::= $\wedge \mid \Rightarrow$

<setoperator> ::= $\in \mid \cup \mid \cap \mid \notin$

<atomicoperator> ::= $+ \mid < \mid > \mid \leq \mid \geq \mid \neq \mid =$

<set> ::= *U*| *S*| *O*

<relationsets> ::= <set_identifier>

<attname> ::= $ua_1 \mid ua_2 \mid \ldots \mid ua_x \mid sa_1 \mid sa_2 \mid \ldots \mid sa_y \mid oa_1 \mid \ldots \mid oa_z$

<attval> ::= ‘ua$_1$val$_1$’ | ‘ua$_1$val$_2$’ | ... | ‘ua$_x$val$_r$’ | ‘sa$_1$val$_1$’ | ‘sa$_1$val$_2$’ | ... | ‘sa$_y$val$_s$’ | ‘oa$_1$val$_1$’ | ... | ‘oa$_z$val$_t$’

<size> ::= ϕ | 1 | ... | N

<item> ::= *limit*| *attval*| **attfun**(<attname>).*limit*| **attfun**(<attname>).*attval*

<attributefun> ::= $ua_1 \mid ua_2 \mid \ldots \mid ua_x \mid sa_1 \mid sa_2 \mid \ldots \mid sa_y \mid oa_1 \mid \ldots \mid oa_z$

<function> ::= **SubCreator** | **assignedEntities**$_{U,<\mathrm{attname}>}$ | **assignedEntities**$_{S,<\mathrm{attname}>}$ | **assignedEntities**$_{O,<\mathrm{attname}>}$

Example

1. A customer cannot get both benefits ‘bf1’ and ‘bf2’

Expression: $|OE(UMEBenefit).attset \cap benefit(OE(U))| \leq OE(UMEBenefit).limit$

2. If a customer committed felony ‘fl1’, She can not get more than one benefit from ‘bf1’, ‘bf2’ and ‘bf3’

Expression: $OE(UMECFB)(felony).attset \cap felony(OE(U))| \geq OE(UMECFB)(felony).limit \Rightarrow |OE(UMECFB)(benefit).attset \cap benefit(OE(U))| \leq OE(UMECFB)(benefit).limit$

Use Cases

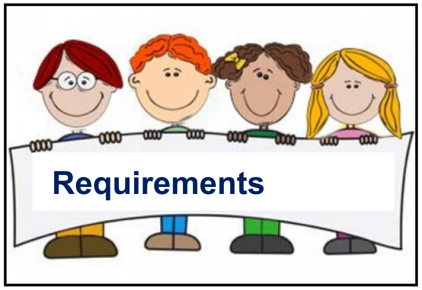
- ABCL can configure well-known RBAC constraints
  - Role can be considered as a single attribute
  - Can express SSOD and DSOD constraints
  - Just need to declare conflict-relation sets for conflicting roles

- It can configure several security requirements of traditional organization (e.g. banking organization)
  - E.g. Constraints on benefit attribute assignment

Conclusion

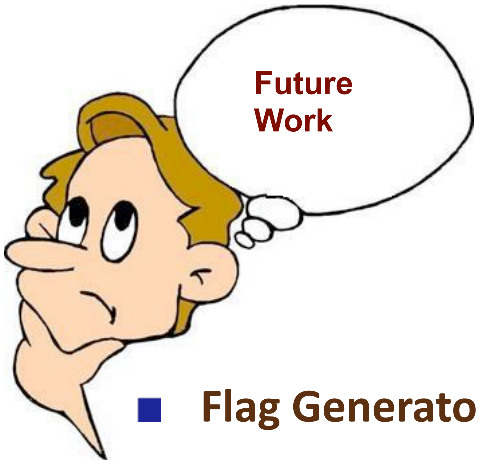
- A Constraint Specification Framework for ABAC and Cloud IaaS

- Easily manageable and generic



Tenants

- Automatic Generation of Constraints



- Flag Generator System

- Improve mining (incorporate noise)

- Analysis for other VR-to-PR

Credit: www.iconarchive.com

Publications (related to dissertation)

Khalid Bijon, Ram Krishnan and Ravi Sandhu
Automated Constraint Constructions Cloud Infrastructure as a Service.
Under Preparation (target IEEE TDSE)

Khalid Bijon, Ram Krishnan and Ravi Sandhu
Mitigating Multi-Tenancy Risks in IaaS Cloud Through Constraints-Driven Virtual Resource Scheduling.
ACM Symposium on Access Control Models and Technologies, 2015.

Khalid Bijon, Ram Krishnan and Ravi Sandhu
Virtual Resource Orchestration Constraints for Cloud Infrastructure as a Service.
ACM Conference on DATA and Application Security and Privacy, 2015.

Khalid Bijon, Ram Krishnan and Ravi Sandhu
A Formal Model for Isolation Management in Cloud Infrastructure-as-a-Service.
International Conference on Network and System Security , 2014.

Khalid Bijon, Ram Krishnan and Ravi Sandhu
Towards An Attribute Based Constraints Specification Language.
IEEE International Conference on Privacy, Security and Trust, 2013.

Khalid Bijon, Ram Krishnan and Ravi Sandhu
Constraints for Attribute Based Access Control
ASE Science Journal, 2013.

Publications (outside dissertation)

Khalid Bijon, MM Haque and Ragib Hasan
A TRUst based Information Sharing Model (TRUISM) in MANET in the Presence of Uncertainty.
International Conference on Privacy, Security and Trust, 2014.

Khalid Bijon, Ram Krishnan and Ravi Sandhu
A Framework for Risk-Aware Role Based Access Control.
IEEE Symposium on Security Analytics and Automation, 2013.

Khalid Bijon, Ram Krishnan and Ravi Sandhu
Risk-Aware RBAC Sessions.
International Conference on Information Systems Security, 2012.

Khalid Bijon, Tahmina Ahmed, Ravi Sandhu and Ram Krishnan
A Lattice Interpretation of Group-Centric Collaboration with Expedient Insiders.
IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing, 2012.

Yuan Cheng, Dang Nguyen, Khalid Bijon, Ram Krishnan, Jaehong Park and Ravi Sandhu
Towards Provenance and Risk-Awareness in Social Computing.
ACM International Workshop on Secure and Resilient Architectures and Systems, 2012.

Khalid Bijon, Ravi Sandhu and Ram Krishnan
A Group-Centric Model for Collaboration with Expedient Insiders in Multilevel Systems.
IEEE International Symposium on Security in Collaboration Technologies and Systems, 2012.

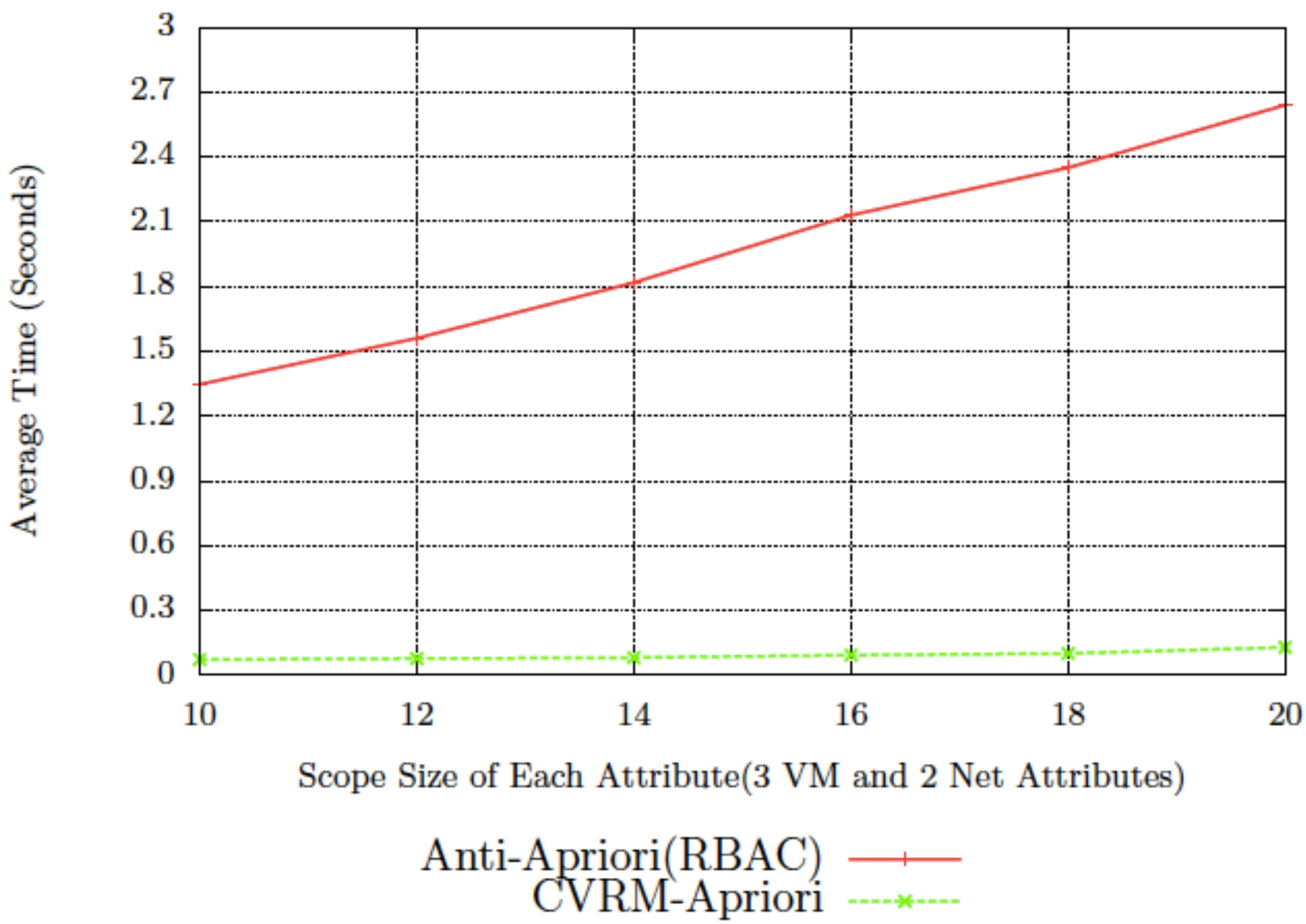
Tahmina Ahmed, Ravi Sandhu, Khalid Bijon, and Ram Krishnan
Equivalence of Group-Centric Collaboration with Expedient Insiders (GEI) and LBAC with Collaborative Compartments (LCC).
Technical Report CS-TR-2012-010, Department of Computer Science, 2012

Ravi Sandhu, Khalid Zaman Bijon, Xin Jin and Ram Krishnan
RT-Based Administrative Models for Community Cyber Security Information Sharing.
IEEE International Workshop on Trusted Collaboration, 2011.

Thank You!

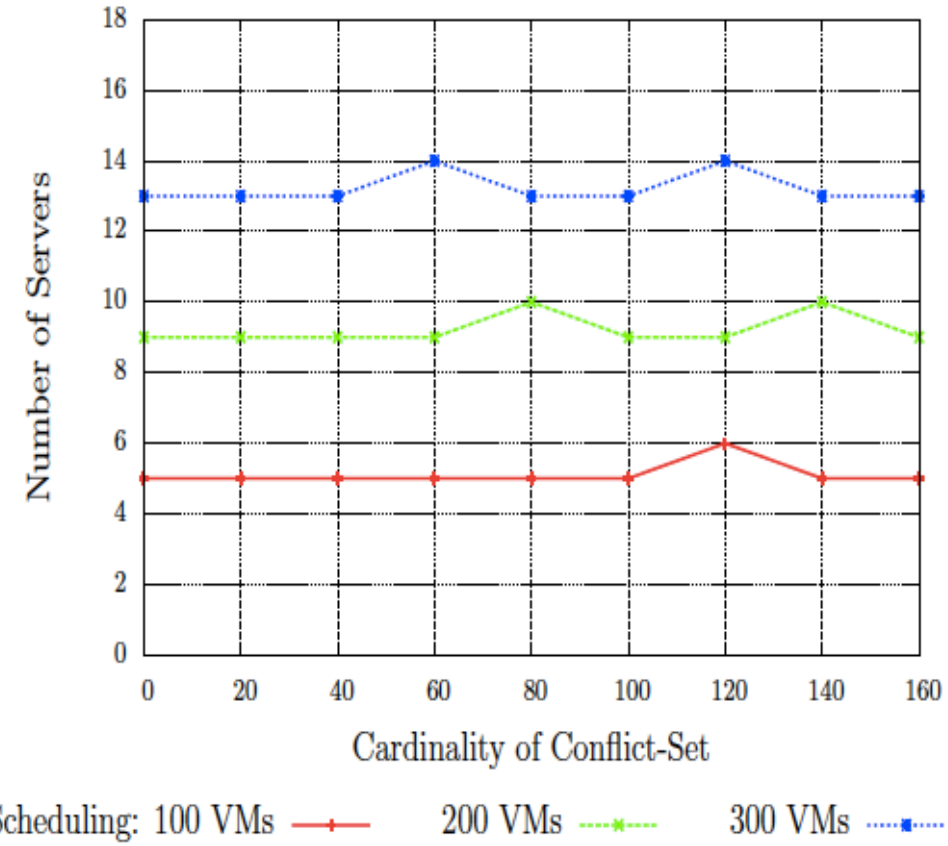
Evaluation

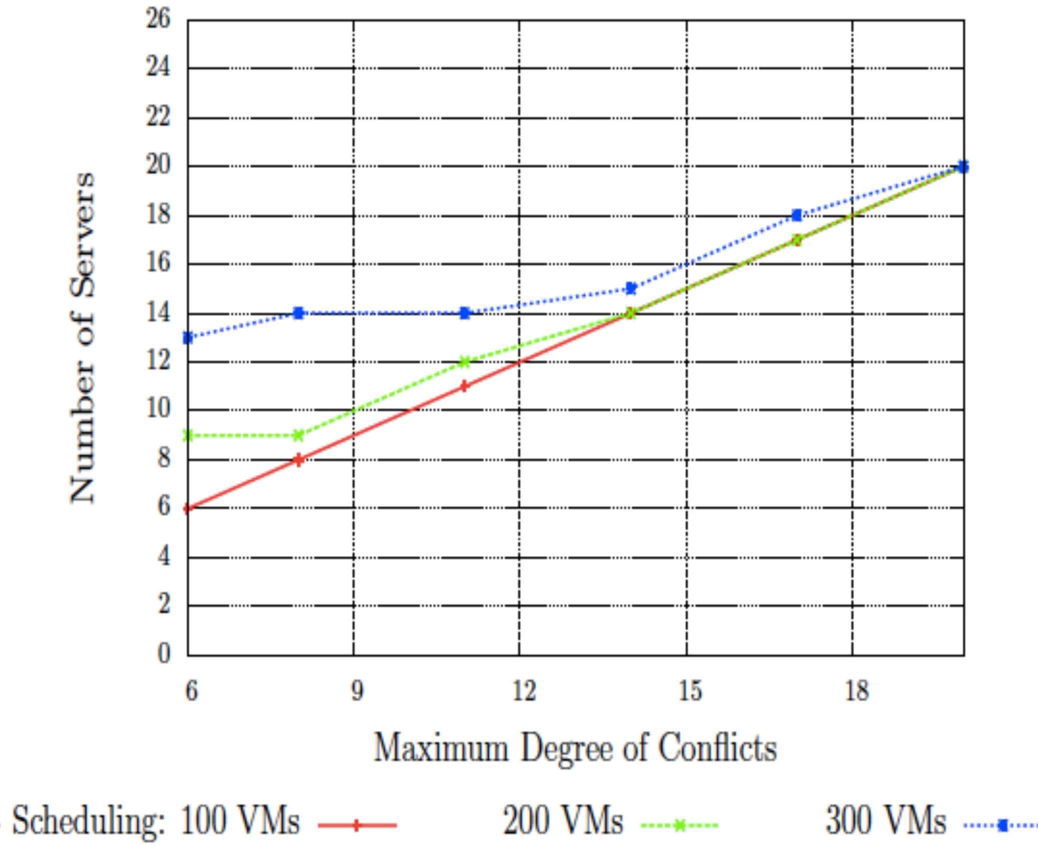
- Mining Time with Increasing Scope



Experimental Analysis

3. Required Number of Hosts





Required Number of hosts for Varying Number of Conflicts

Required Number of hosts for Max Degree of Conflicts

Vision

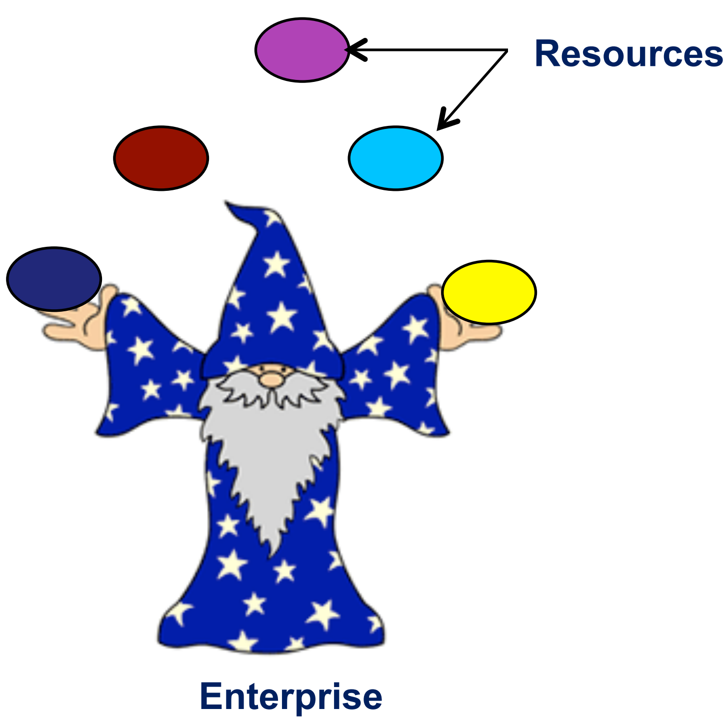


Figure 3-A: On Premise

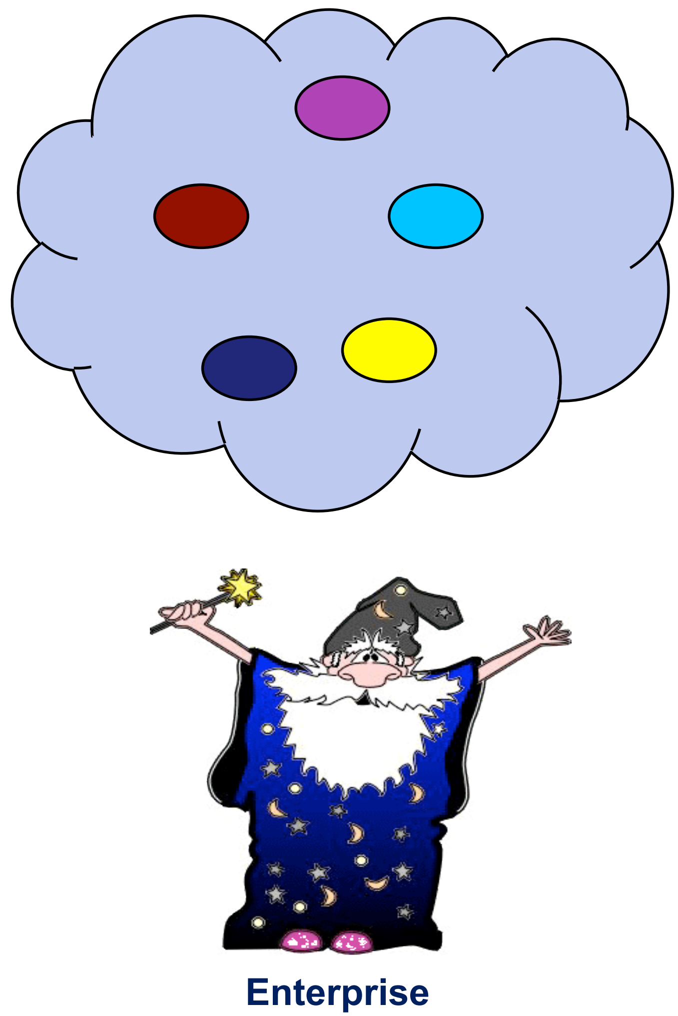
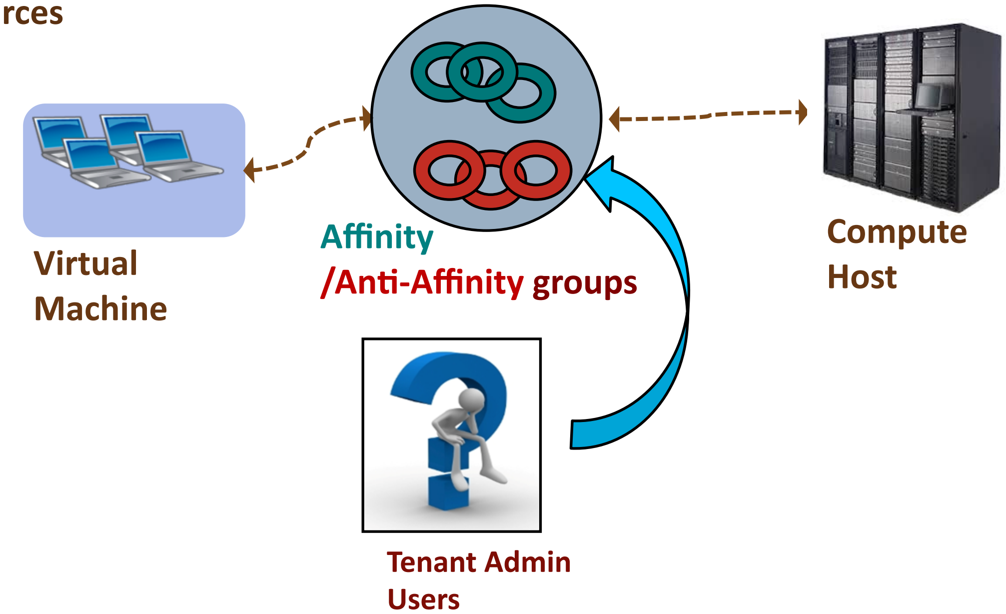


Figure 3-B: In Cloud

Credit: www.erindorpress.com/

Motivation

- **Not Scalable**
  - Manual Groupings of Virtual Resources



- **Inefficient Scheduler (e.g., filter-scheduler in OpenStack)**
  - Host Exhaustion problem

Credit: www.iconarchive.com
www.consulting.ky

Conclusion

- **A Constraint Specification for Attribute Based Access Control**

- **Mechanism for High Level Security Policy Specifications for an Organization**

Conclusion

- Scalable Constraint-Aware Scheduling
- Host Optimization

Tenants

- Affinity Constraints
- Combine both of them

Credit: www.iconarchive.com