Cloud based IT Infra with Central Identity

Phase I

Project Guide

T. Chandra Shekar

Lecturer – Dept. of CSE

Presenting by

Team r3b00+

Dept. of CSE, RGUKT – Nuzvid

December 12, 2014

About us

We are from team *r3b00+* {reboot}

T. Aneesh Kumar	N090247
P. Nageswarao	N091030
P. Anesh	N090977
P. Jyothi Ram	N090990
K. Naresh Chowdary	N090331
N. Venkata Sateesh	N090935
M. Sanyasirao	N090891

Outline

1 Objective
- Objective
- Motivation

2 Proposed System

3 Network Components

4 Single Sign-On

5 Identity Management

6 RBAC

7 Cloud Infra

Objective

Our objective is to create a private cloud and availing access of all its services using central identity with single sign on through dynamic role based management along with REST API to third party for application developers and users.

This can be developed by using open source tools like OpenStack, NFS, LDAP, Ubuntu and etc

Expecting to serve with virtual machines to the research, virtual labs rather than dedicated lab hardware.

Motivation

- No Central Identity, Central Storage & High capacity hardware resource pool.
- Failed to maintain large user load web services like ONB, Exam servers, etc.
- Dedicated computer course labs like Matlab, VLSI, etc.
- No proper Web Application Security & Standards.
- Inadequate resource requirements for Research.

Outline

1 Objective

2 Proposed System
- Users & IT Services
- Cloud Infrastructures

3 Network Components

4 Single Sign-On

5 Identity Management

6 RBAC

7 Cloud Infra

Users & IT Services

We are collaborating all IT Services that are required for University and identifing the users who is going to use them. All Users are catagorized into 4 groups [1]

- Studens, Developers, Staff, faculty & Researches

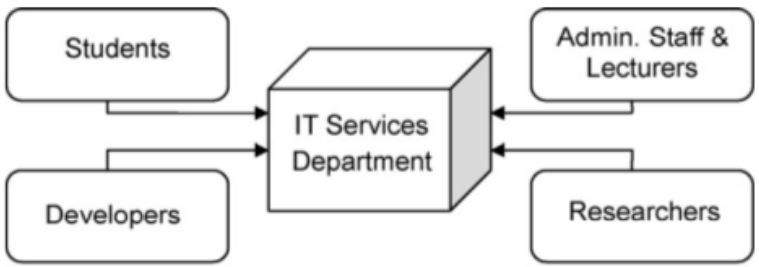

Figure : Simplified structure of the main users of IT services.

Cloud Infrastructures

All University IT Services are deployed in a private cloud, constructed over exsiting infrastructure, that can be broadly viewed as

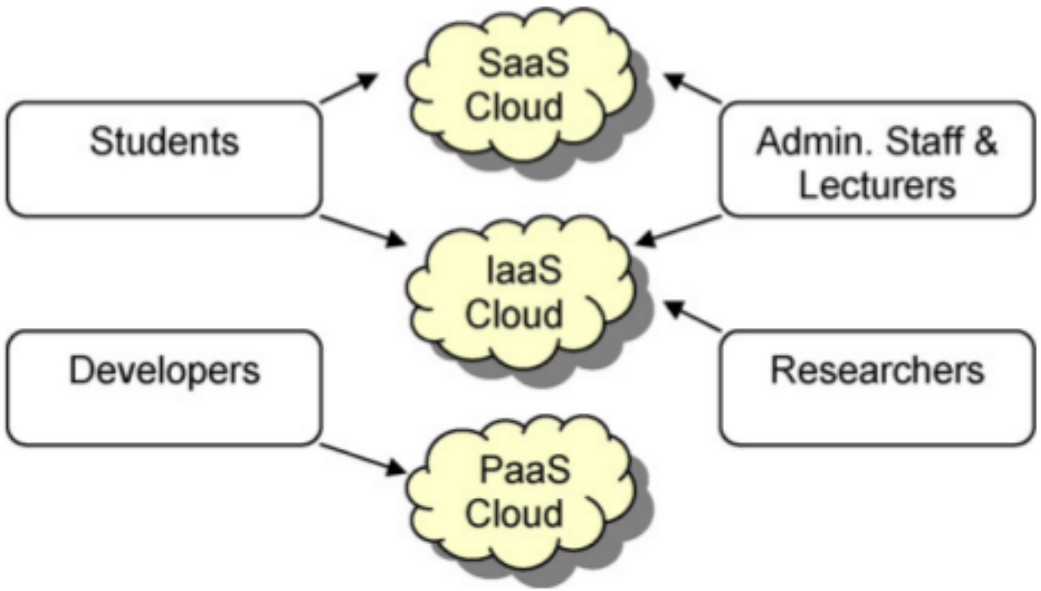

Figure : IT Services and Users in Cloud Computing

Proposed System - Main Components

- Network Components
  - AAA, LDAP, NFS
- Central Identiy
  - Single Sign on
  - Fedarated Identity
  - Dynamic Role Based Access Control
  - REST API to third party
- Cloud Infrastructure
  - Cloud Computing, Private Cloud, Open source tools

Outline

1 Objective

2 Proposed System

3 Network Components
- Introduction
- Why Network Components?
- AAA, LDAP and NFS Servers

4 Single Sign-On

5 Identity Management

6 RBAC

7 Cloud Infra

Introduction

To construct a Campus Area Network(CAN), building network is an important and complex task. So in order to make that complex task easier and very robust in nature we have to use some of these components such as LDAP, AAA and NFS.

Devices used to setup a Local Area Network (LAN) are the most common types of network devices used by the public. A LAN requires a router, switch, cabling or radio technology, network cards.

Why Network Components?

To build a robust and secure network we must use some of these components in constructing that network. These will simple our manual work by automating many complex tasks and very useful.

The given below are some basis network components

Router: A router is a device that forwards data packets along networks. A router is connected to at least two networks, commonly two LANs or WANs or a LAN and its ISP's network. Routers are located at gateways.

Switch: A network switch is a computer networking device that connects devices together on a computer network, by using a form of packet switching to forward data to the destination device.

AAA & LDAP

AAA: Authentication, Authorization and Accounting

- An AAA server is a server program that handles user requests for access to computer resources and, provides authentication, authorization, and accounting (AAA) services
- The current standard by which applications communicate with an AAA server is either RADIUS or TACACS+

LDAP: Lightweight Directory Access Protocol

- For accessing and maintaining distributed directory info
- LDAP provides a secure way to access information presented in directories by using some authentication methods

NFS Servers

NFS: Network File System

- It is a client/server system that allows users to access files across a network and treat them as if they resided in a local file directory
- Exporting: NFS Server provides clients with access to its files
- Mounting: File systems are made available to OS and the user

Outline

1 Objective

2 Proposed System

3 Network Components

4 Single Sign-On
- What is SSO
- Why SSO
- Advantages
- How well we will implement?
- What is JSON?
- What is REST?
- What is REST? (Contd...)

5 Identity Management

What is Single Sign-On?

- Single Sign-On (SSO) is an authentication process that allows a user to access multiple applications with one set of login credentials.
- One login. All of RGUKT.

Figure : Google SSO and Application

Why Single Sign-On?

- Signing Up everytime is troublesome
- I am a nerd, I can't remember all the passwords across multiple Apps.
- Is it possible to just sign-on once to perform all the actions?
- Is basic Authorization on many sites is secure? Any alternative?
- Yes!! Single sign-on can be used to answer all these Questions.

Advantages

- Ease burden on developers
- Improved user experience, no password lists to carry. Thus, improving productivity.
- Ease of Access through a single Central Database.
- Transfer of Sensitive Data across network is minimized.
- Enables users to login quickly and securely to all their applications.
- Auditing & Statistical history reviewing simplified.

How well we will implement?

- We want to develope well structured and documented REST API
- Designing neat and user friendly interface with Semantic UI
- Technologies to be Used

- Standards to be followed

What is JSON?

JSON

- JSON stands for **J**ava**S**cript **O**bject **N**otation
- An open standard format that uses plain text to transmit data.
- Used primarily to transmit data between a server and web application, as an alternative to XML.

Example

```
{
  "ID":1234,
  "Title":"Getting Started with PHP",
  "Description":"A definitive guide to learn PHP from
    scratch.",
  "Author":"Branko Ajzele",
  "Year":"2013"
}
```

Listing 1: JSON Example

What is REST?

REST API

- REST stand for **RE**presentational **S**tate **T**ransfer
- A Collection of simple URIs, and HTTP calls to those URIs and some JSON resources
- Basic CRUD Operations

operation	HTTP verb	action
Create	`POST /posts`	`create`
Read	`GET /posts/1`	`show`
Update	`PUT /posts/1`	`update`
Delete	`DELETE /posts/1`	`destroy`

Figure : CRUD Operations

REST API Example

Syntax

- http://it-ebooks-api.info/v1/book/:id/:author/

Example

- Request URI – http://it-ebooks-api.info/v1/book/1234/
- Response

```
{
  "ID":1234,
  "Title":"Getting Started with PHP",
  "Description":"A definitive guide to learn PHP from
    scratch.",
  "Author":"Branko Ajzele",
  "Year":"2013"
}
```

Listing 2: REST Example

Outline

1 Objective

2 Proposed System

3 Network Components

4 Single Sign-On

5 Identity Management
- Introduction
- Federated Identity Management
- User Centric Identity Management
- Relating to our University

6 RBAC

7 Cloud Infra

Introduction

- Users need to Manage different passwords to authenticate at various applications
- This is done at a central point to unify the authentication & authorization
- Organization has to take care of it, if not out-source it
- **Different Implementations**
  - Kerberos, LDAP Servers, Relational db's
  - X.509 Certificate based, FIdM (Shibboleth)
- **Secure Mechanisms**
  - User-Centric Identity
  - Federated Identity

Federated Identity Management

- SSO introduced on a large scale with Kerberos protocol
- SAML = Authentication + Authorization
- User can get access by using I-Card or Unique URL
- **Components**: SP, IdP, Federation(AAI), DS or WAYF Server, X.509 Certificates.
- **Advantages**
  - Secure, Prevalent, Fine-grained Authorization
- **Disadvantages**
  - Complexity of the protocol, Unable to manage privacy info

Federated Identity Management Continued

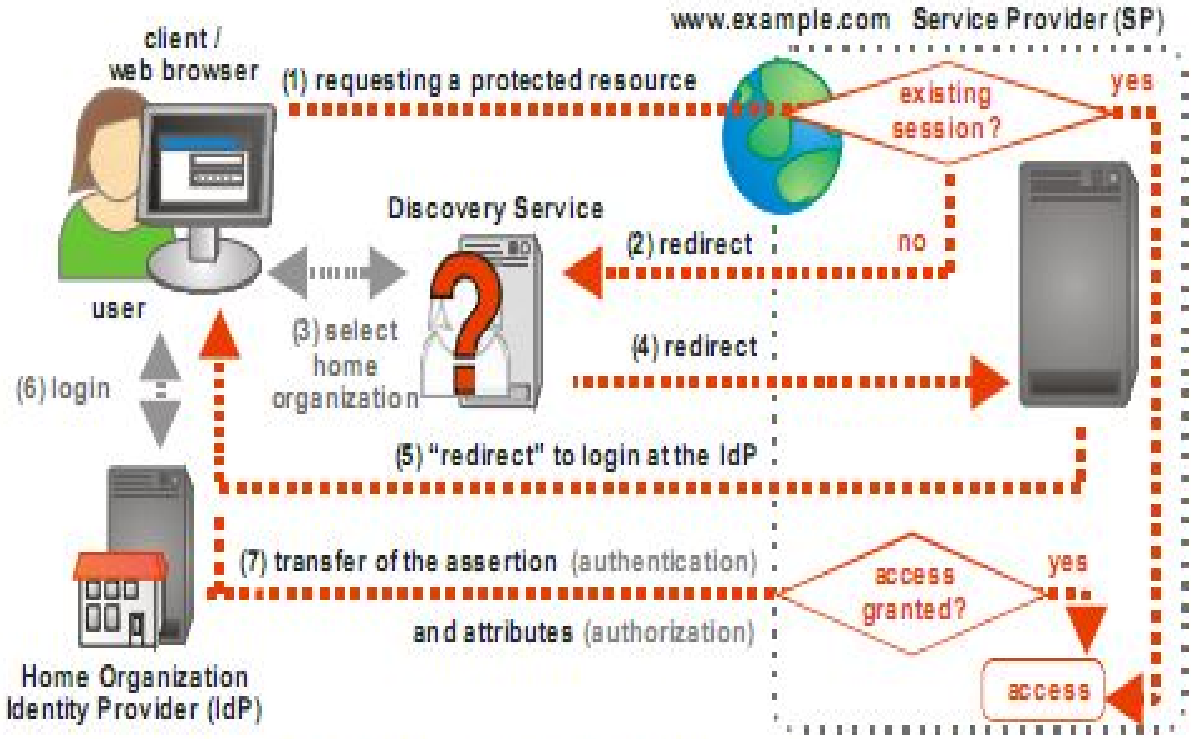

Figure 1. SAML-based federated identity management with Shibboleth 2.0.

User-Centric Identity Management

- The user presents the information as the I-Card
- There is no need of DS, its done on the user side
- User can get access by using I-Card or Unique URL
- **Components**: I-Card or Unique URL, SP, IdP, etc.
- **Advantages**
  - Usability,More Privacy, Simplification of Protocol
- **Disadvantages**
  - Security Risks(Phishing, XSS, CSRF),No single standard

User-Centric Identity Management Continued

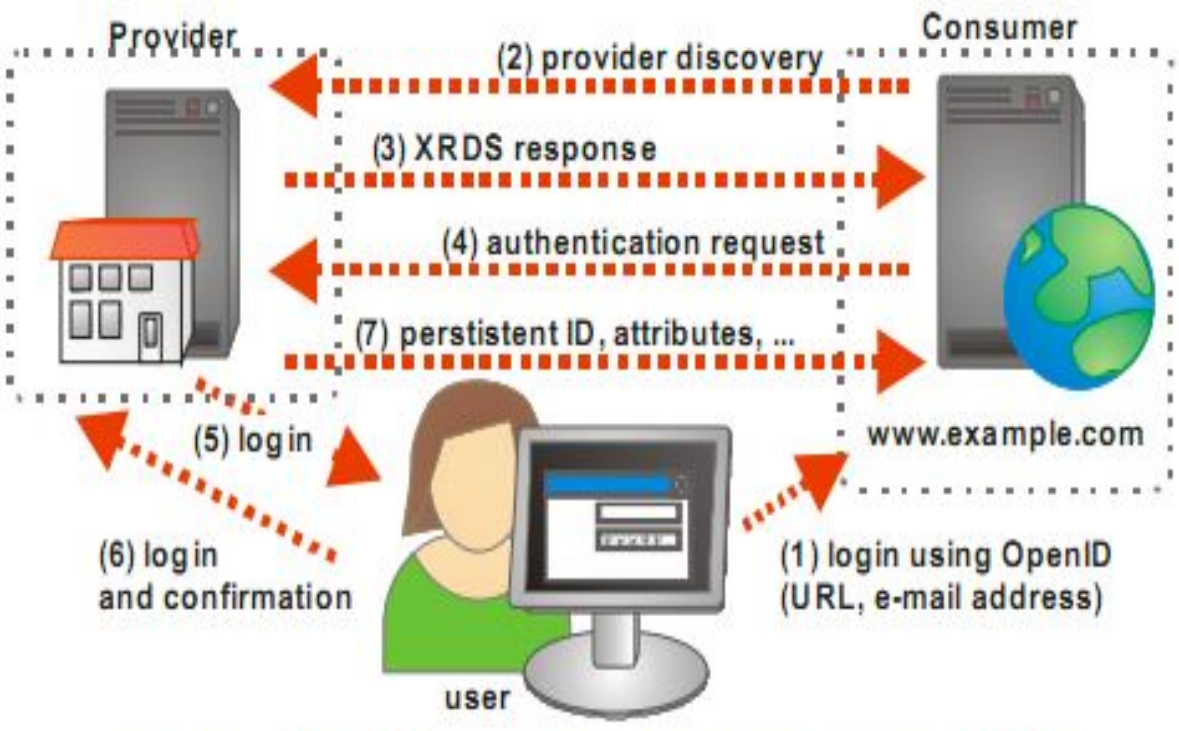

Figure 2. OpenID as an example for user-centric identity management

Intra & Inter campus Infrastructure

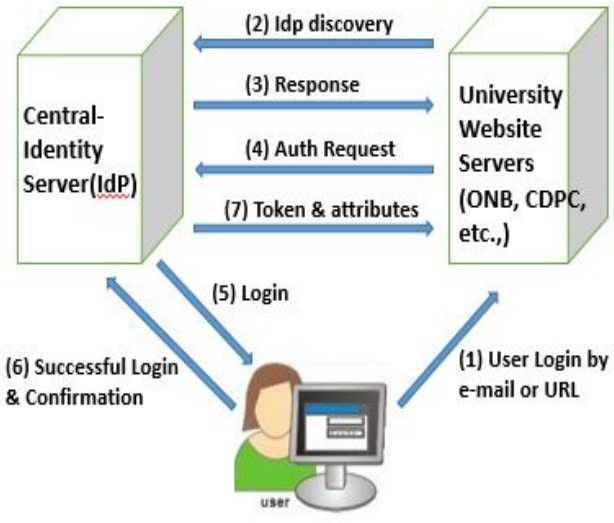

Figure : Intra Campus

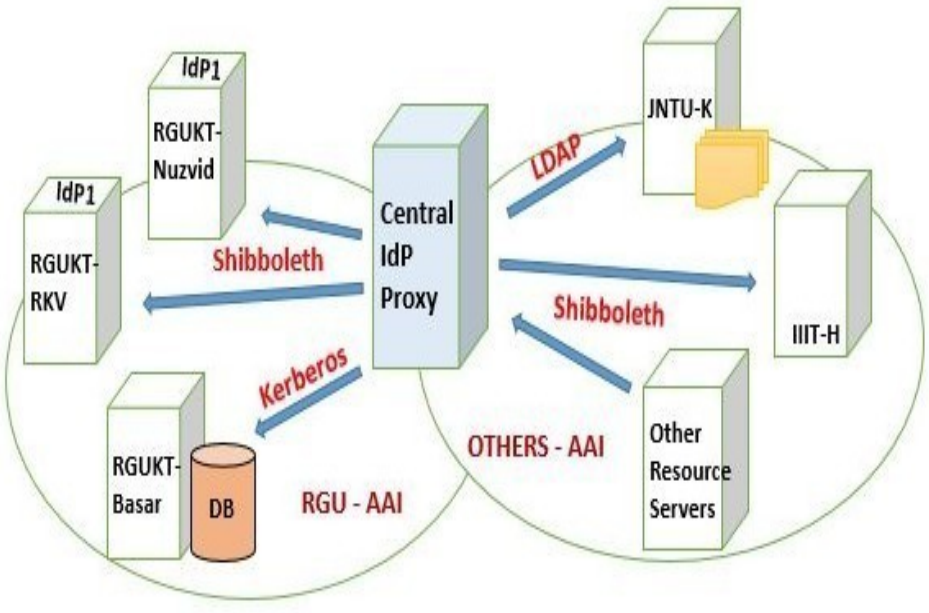

Figure : Inter Campus

Outline

1 Objective

2 Proposed System

3 Network Components

4 Single Sign-On

5 Identity Management

6 RBAC
- Introduction
- Idea of RBAC
- Structure of RBAC
- Dynamic RBAC

7 Cloud Infra

Introduction to RBAC

- Role Based Access Control(RBAC) assigns users to roles and then roles to permissions, It solves problems of least privilege, separation of duty and other security issues
- In RBAC model, these rights are defined based on the role that individuals are assigned to in an organization
- It overcomes the problems in DAC which is flexible but not secure and MAC which is Secure but not flexible

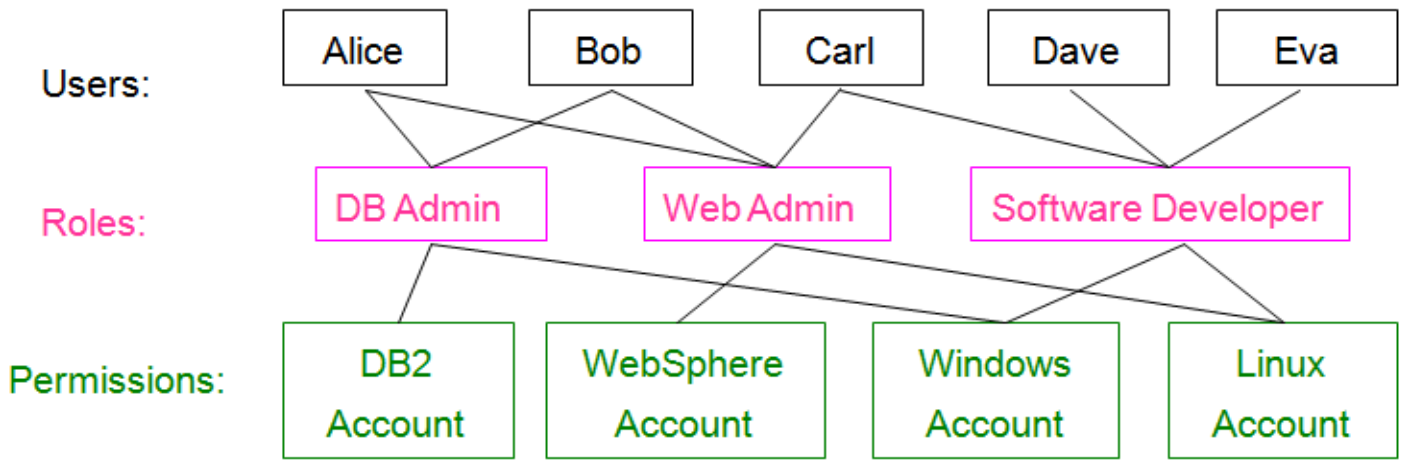

Figure : Scenario of RBAC

Basic Idea of RBAC

- Access Control policy is embodied in various components of RBAC such as,
  - Role-Permission relationships
  - User-Role relationships
  - Role-Role relationships
- Users get roles corresponding permissions by getting roles to operate on the objects
- RBAC model is defined in terms of three model components - Core RBAC, Hierarchical RBAC and Constraint RBAC

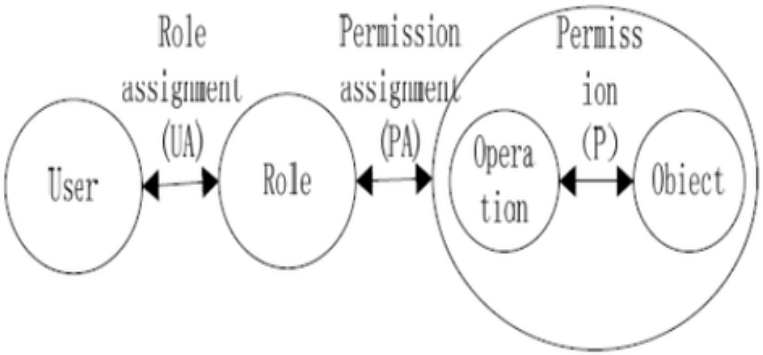

Fig 2. Core Idea of RBAC

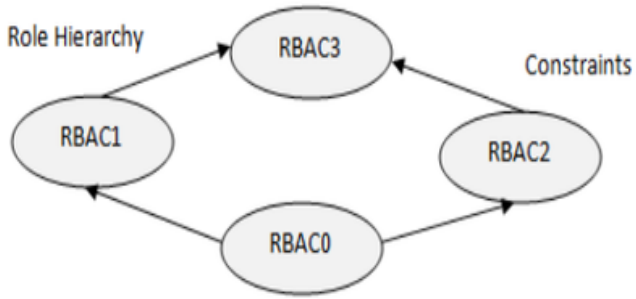

Fig 3. Family of RBAC Model

Structure Diagram of RBAC Model

- The Structure diagram of role based access control model consists of role hierarchies and constraints
- Role hierarchical relationship expresses the inheritance in roles permissions
  - User inheritance
  - Permission inheritance
  - Activation inheritance
- Constraints in RBAC adds separation of duty relations
  - Mutual exclusion
  - Pre-condition
  - Cardinality

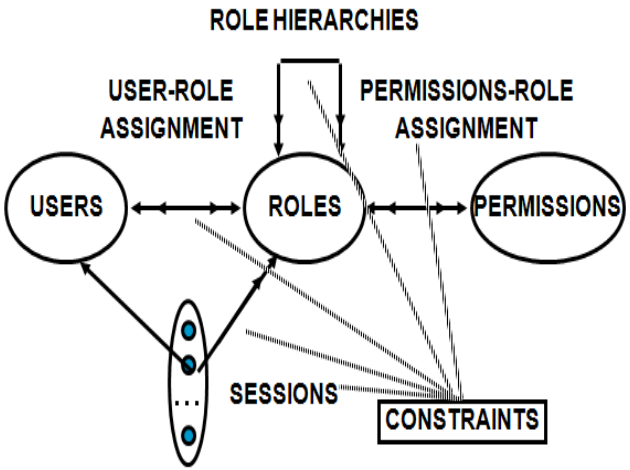

Figure : RBAC3 Mode

Dynamic RBAC

Dynamic RBAC overcomes the shortages of the traditional RBAC by adding with dynamic constraints and permissions

- It retains original static constraints of traditional RBAC
- The App creator no need to go for administration, himself he can add or create a role for users
- It supports each user has different levels of permission at different time

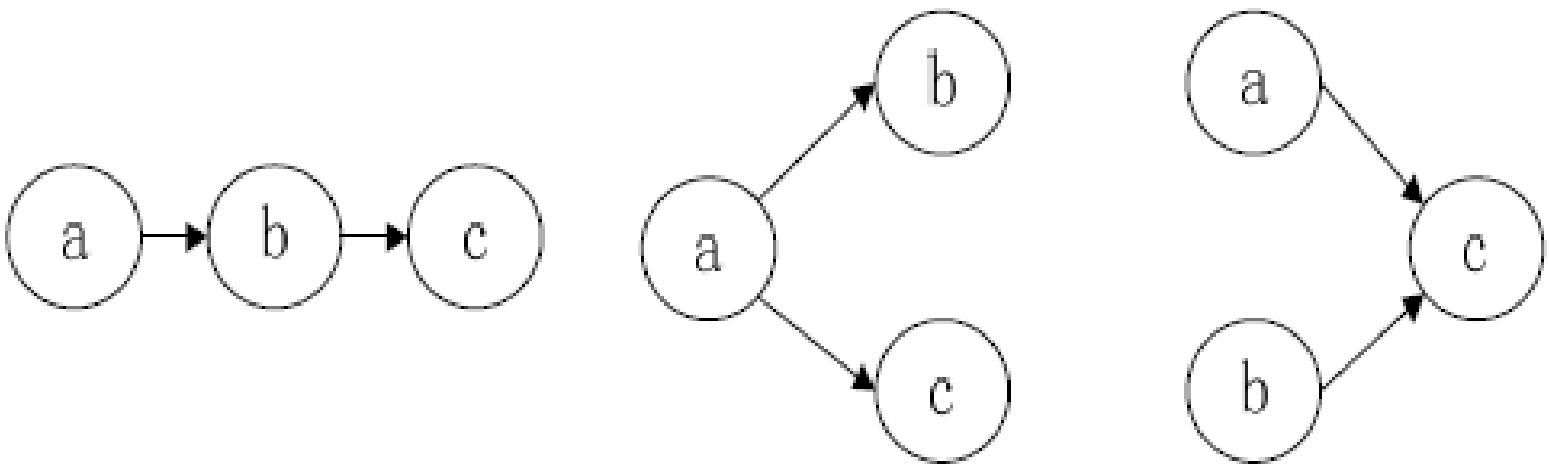

Figure : Different Types of Association

Outline

1 Objective

2 Proposed System

3 Network Components

4 Single Sign-On

5 Identity Management

6 RBAC

7 Cloud Infra
- Cloud Computing
- Private Clouds
- Cloud Infrastructure

Cloud Computing - Introduction

What is Cloud Computing ...?

"Cloud computing is a model for enabling convenient, on- demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" [1]

Essential Characteristics:

- On-demand self-service
- Broad network access
- Resource pooling
- Rapid elasticity
- Measured Service

Cloud Computing - Servcice & Deployment Models

Servcice Models:

Figure : Cloud Computing - Servcice Models

Deployment Models:

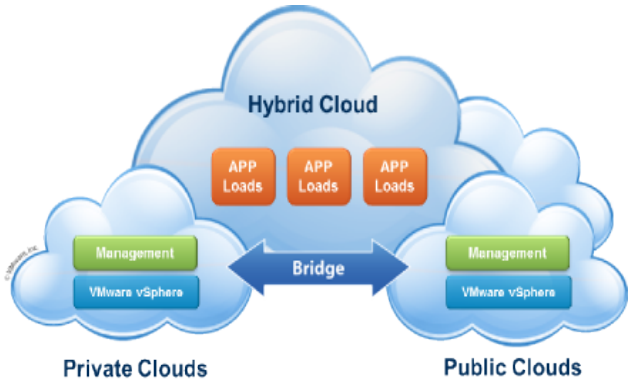

Figure : Cloud Computing - Deployment Models

Private Clouds – Definition & Opensource Tools

"Private Cloud"

– It is one of the cloud deployment model where the resources of small or medium organization are united and cattered to users of the that organization or outsourced through internet.

Opensource Tools:

Figure : Private Cloud - Open source tools

Architecture

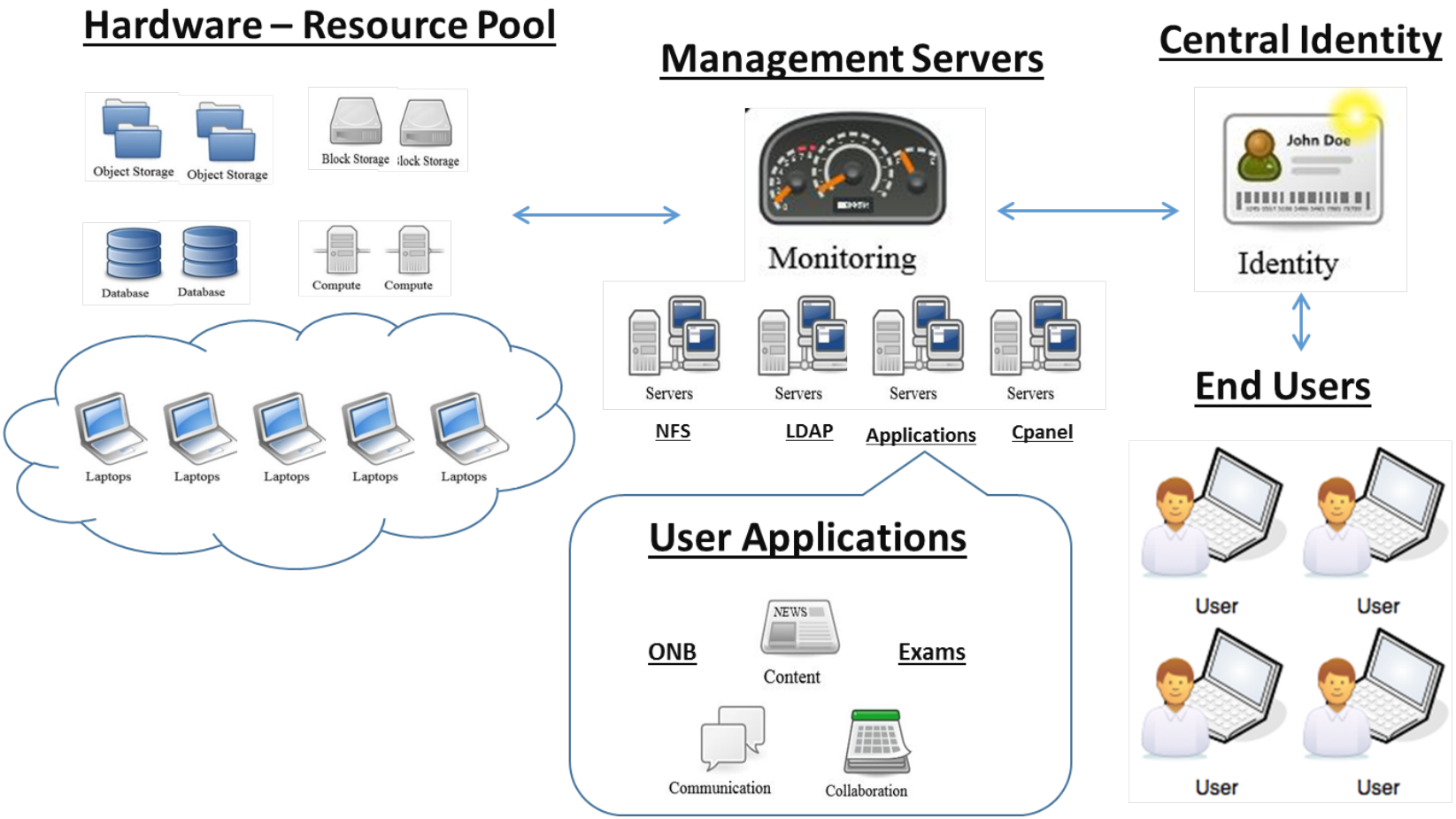

Figure : Cloud Infrastructure Archtitecture

References

- Nabil Sultan, "Cloud Computing for Education, International journal information management, 30, pp 109-116, 2010.
- Tharam Dillon, Chen Wu and Elizabeth Chang, "Cloud Computing: Issues and Challenges", 24thIEEE International Conference on Advanced Information Networking and Applications, 2010
- Sebastian Rieger, "User-centric Identity Management in Heterogeneous Federations", Fourth international conference on Internet and Web Applications and Services, 2009.
- Jun Zheng, Qikun Zhang ,Shangwen Zheng and Yuan Tan, "Dynamic Role Based Access Control", JOURNAL OF SOFTWARE, VOL. 6, NO. 6, JUNE 2011.
- R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E.Youman, "Role-Based Access Control Models," [C] IEEE Computer, vol. 29, pp. 38-47, 1996.

References

- Ivan Novakov, "Web Single Sign On Systems", CESNET technical report number 21/2006
- C.S.Yang, C.Y.Liu, J.H.Chen, C.Y.Sung, "Design and Implementations of Secure Web-based LDAP Management System".
- Ning Li, Qing Wang, Zhongliang Deng, "Authentication Framework of IIEDNS Based on LDAP and Kerberos", Proceedings of IC-BNMT2010.
- Salah A.Jaro Alabady, "Design implementation of Network Security Model Using Static VLAN and AAA Server
- Shengli Liu, Wenbing Wang, Yuefei Zhu, "A New-Style Domain Integrated Management of Windows and UNIX", The Ninth International Conference on Web-Age Information Management.

References

- Ubnutu OS, http://www.ubuntu.com/
- Openstack, https://openstack.org/
- Linux Bible, http://tuxnetworks.blogspot.com
- OAuth 2.0, http://oauth.net/
- Node.js https://nodejs.org
- Git, https://github.com
- Bootstrap, http://getbootstrap.com
- Stack Overflow http://stackoverflow.com/

End

Thank you and Any Queries ?