

# Cloud based IT Infra with Central Identity

## {Project reboot} - Phase I

### Project Guide

T. Chandra Shekar

Presenting by

*Team r3b00+*

*Dept. of CSE, RGUKT - Nuzvid*

December 10, 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. Sanyasi Rao	N090891

# Objective

The main objective of “Cloud based IT Infra with Central Identity” is to utilize existing hardware, turn them into private clouds and access all of its services using Central Identity, which can be available to third party developers as API with dynamic role management and service endpoints.

New private cloud based IT Infra is aimed to develop using some opensource tools like OpenStack, NFS, LDAP, Ubuntu and etc

Expecting to serve with high computational virtual machines to the research, academic, learning purpose, 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.

# Users & IT Services

We are grouping all IT Services that are required for University into one and identifying the user who will going to use them. All Users are catagorized into 4 groups <sup>[1]</sup>

- Studens, Developers, Staff, faculty & Researches

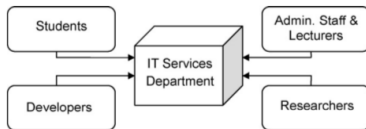


Figure : Simplified structure of the main users of IT services.

# Cloud Infrastructures

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

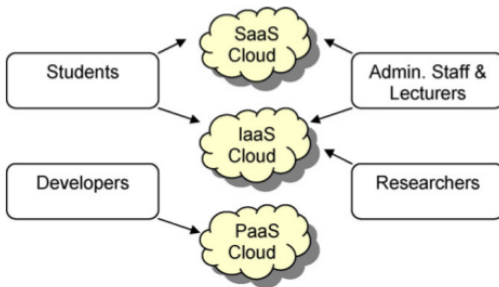


Figure : IT Services and Users in Cloud Computing

# Proposed System - Main Components

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

# Introduction

To construct a Campus Area Network(CAN), building network existed in that 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 simplify our manual work by automating many complex tasks and are very useful.

The given below are some basic 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

# 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 develop well structured and documented REST API
- Designing neat and user friendly interface with Semantic UI
- Technologies to be Used



- Standards to be followed



REST API

# 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

```
1 {  
2   "ID":1234 ,  
3   "Title":"Getting Started with PHP" ,  
4   "Description":"A definitive guide to learn PHP from  
5     scratch." ,  
6   "Author":"Branko Ajzele" ,  
7   "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

```
1 {  
2   "ID":1234 ,  
3   "Title":" Getting Started with PHP" ,  
4   "Description":"A definitive guide to learn PHP from  
5     scratch." ,  
6   "Author":"Branko Ajzele" ,  
7   "Year":"2013"  
}
```

Listing 2: REST Example

# 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

# 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.

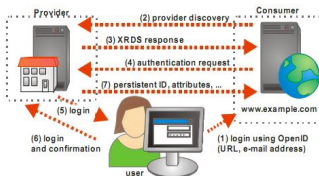


Figure 2. OpenID as an example for user-centric identity management

- Advantages
  - Usability, More Privacy, Simplification of Protocol
- Disadvantages
  - Security Risks(Phishing, XSS, CSRF), No single standard

# Federated Identity Management

- SSO introduced on a large scale with Kerberos protocol
- SAML = Authentication + Authorizatio
- User can get access by using I-Card or Unique URL
- **Components:** SP, IdP, Federation(AAI), DS or WAYF Server, X.509 Certificates.

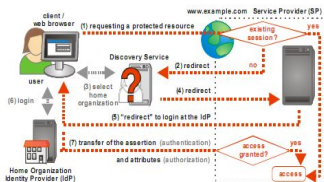


Figure 1. SAML-based federated identity management with Shibboleth 2.0.

- **Advantages**
  - Usability, More Privacy, Simplification of Protocol
- **Disadvantages**
  - Security Risks (Phishing, XSS, CSRF), No single standard

# Intra & Inter campus Infrastructure

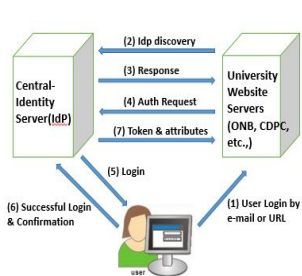


Figure : Intra Campus

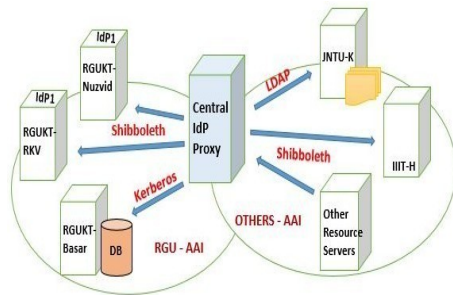


Figure : Inter Campus

# 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

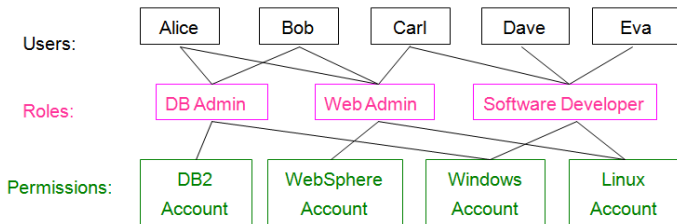


Figure : Scenario of RBAC

# 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

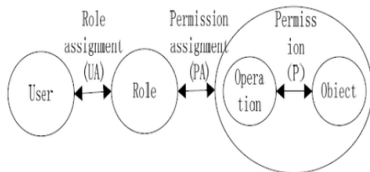


Fig 2. Core Idea of RBAC

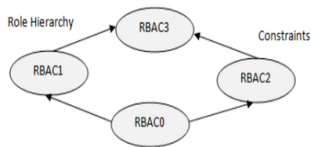


Fig 3. Family of RBAC Model



# 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

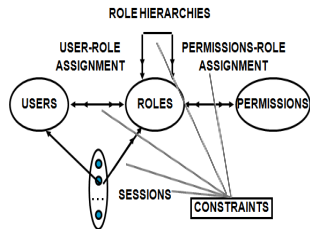


Figure : RBAC3 Mode

# 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

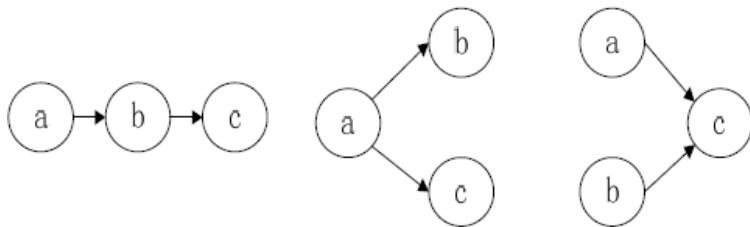


Figure : Different Types of Association

# Cloud Computing - Definition

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]*

# Cloud Computing - Characteristics

One can define Cloud Computing with essential characteristics like

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

# Cloud Computing - Service Models

If we providing any thing as a service comes, that will comes into Cloud Computing. Various Service Delivery Models listed bellow.

- Software as a Service (SaaS)
- Platform as a Service (PaaS)
- Infrastructure as a Service (IaaS)
- Anything as a Service (XaaS)



Figure : Cloud Computing - Service Models

# Cloud Computing - Deployment Model

We can deploy the cloud in various ways.

- Public Cloud
- Private Cloud
- Hybrid cloud

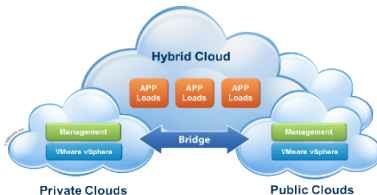


Figure : Cloud Computing - Deployment Models

# Private Clouds – Introduction

As per our concern we mainly focused about private clouds in order to ensure Organizational data security & High resource utilization

## “Private Cloud”

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

# Private Clouds – Open Source Tools

We can construct private cloud using some open source tools like Openstack, Cloudstack, OpenNebula.

We can use this private cloud to deploy various services like Departmental Websites, Notice Boards, Events portal, High Computational Virtual Machines for Virtual Labs, High Performance Computing, Big data analytics.



Figure : Private Cloud - Open source tools



# 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

- Ubuntu 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/>

# One word to say ..

Thank you and Any Queries ?