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 exisiting 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, virtal 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 identifing the user who will going to use them. All Users are catagorized into 4 groups [1]

- Studens, Developers, Staff, faculty & Researches

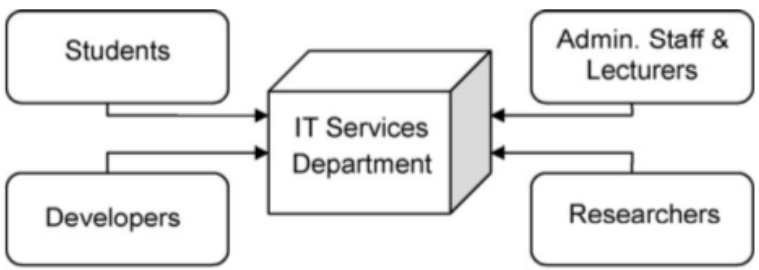

Figure : Simplified structure of the main users of IT services.

Cloud Infrastructures

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

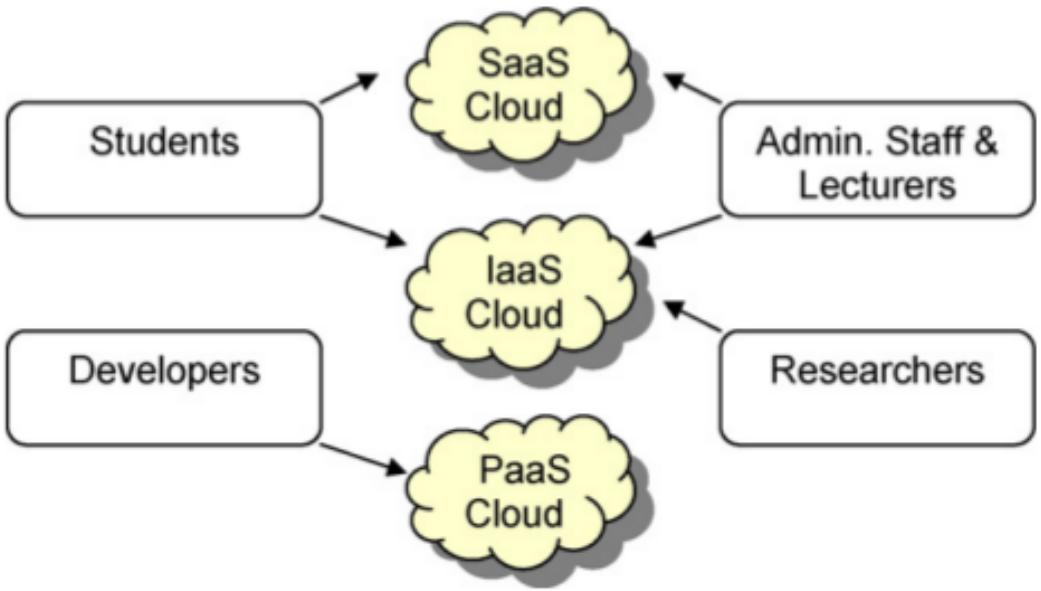

Figure : IT Services and Users in Cloud Computing

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

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

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

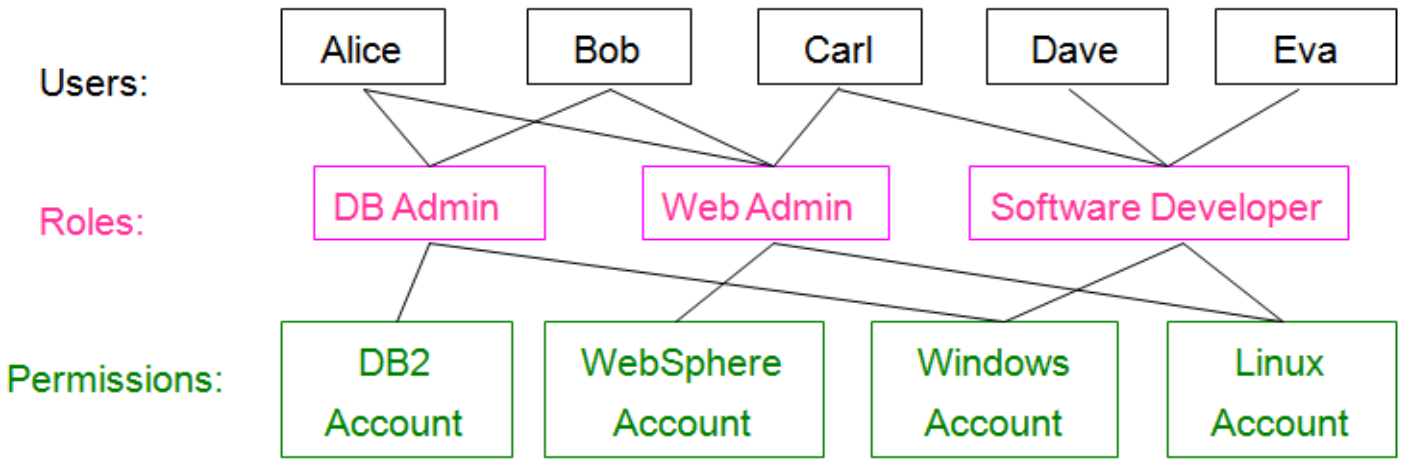

Figure : Scenario of RBAC

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

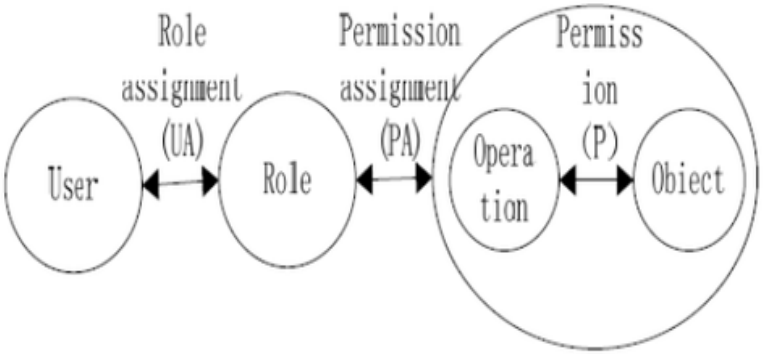

Fig 2. Core Idea of RBAC

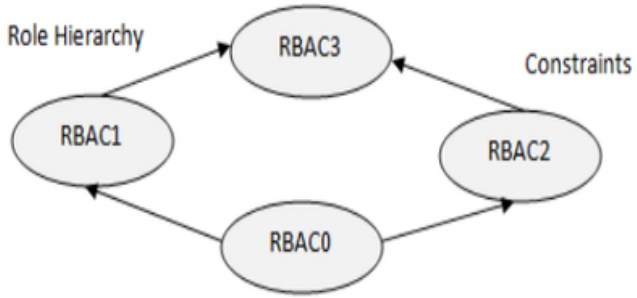

Fig 3. Family of RBAC Model

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

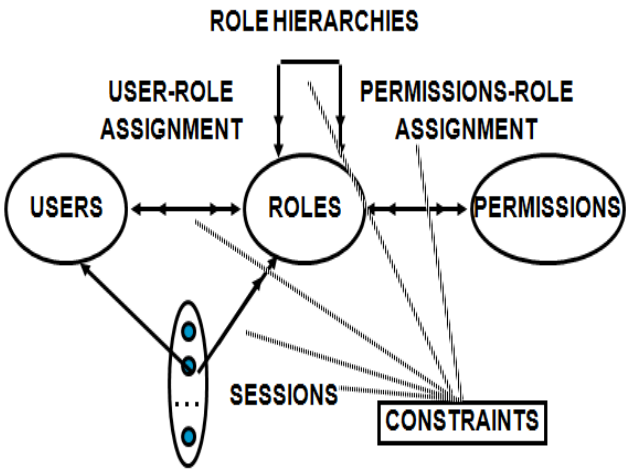

Figure : RBAC3 Mode

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

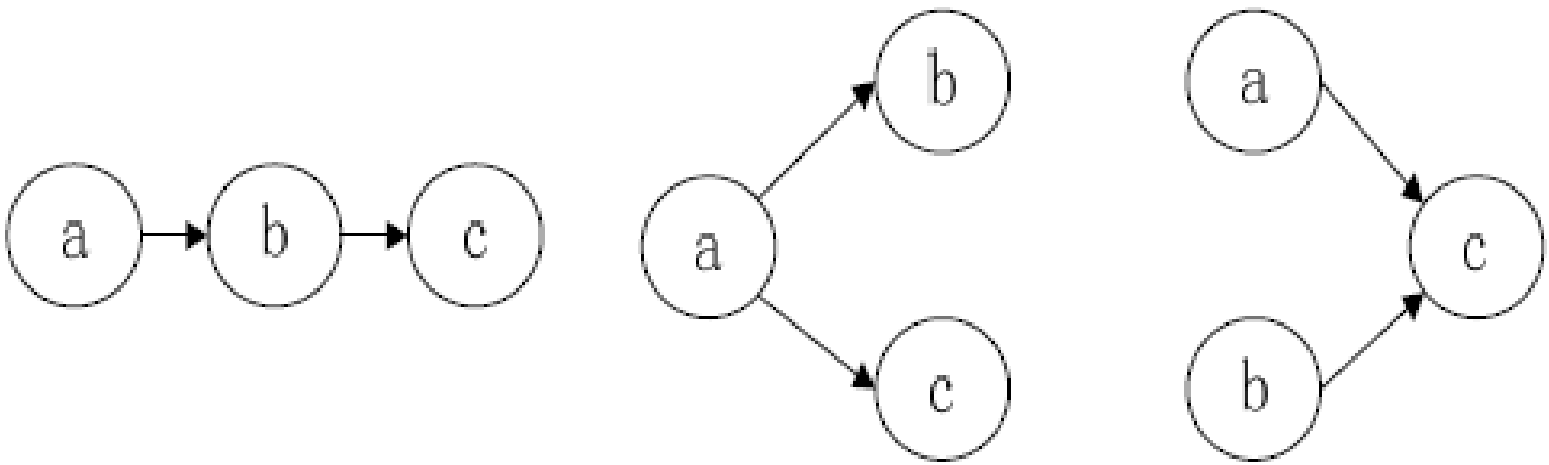

Figure : Different Types of Association

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 - Servcice 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 - Servcice Models

Cloud Computing - Deployment Model

We can deploy the cloud in various ways.

- Public Cloud
- Private Cloud
- Hybrid cloud

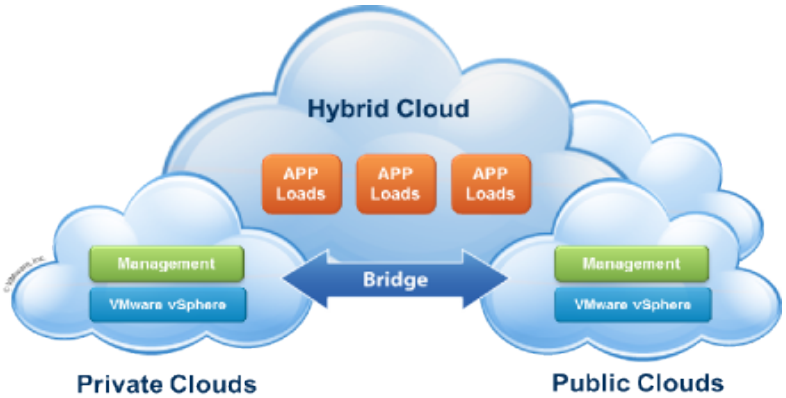

Figure : Cloud Computing - Deployment Models

Private Clouds – Introduction

As per our concern we mainly focused about private clouds inorder 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 cattered to users of the that organization or outsourced through internet.*

Private Clouds – Open Soruce 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

Questions?

Questions ?

One word to say ..

Thank you