



# International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: [www.ijarcsse.com](http://www.ijarcsse.com)

## M University on Cloud

Ms. Anuradha Deokar, Prof. D. M. Thakore

Bharati Vidyapeeth Deemed University

College of Engineering, Pune, India

---

**Abstract**—*M-University on Cloud is completely mobile application which is based on Android Operating System. Android is a software stack for mobile devices that includes an operating system, middleware and key applications. M-University establishes the connection between the Students, Professors and administrator of the college through mobile. Here the Administrator task is to prepare and monitor the course contents, update/delete student. The Professor task is to prepare Test papers and monitors students by viewing their performance and the duration of listening to videos in graphical format. And the Student views the course contents, can listen to videos, giving comments and rating about particular session and even can view the comments and rating given by other students, giving test and viewing their performance, comparing with students through graphs.*

**Keywords**-*Android 2.2, cloud, mobile security, application*

---

### I. INTRODUCTION

Android is a software stack for mobile devices that includes an operating system, middleware and key applications. The Android SDK provides the tools and API necessary to begin developing applications on Android platform using Java programming language. It creates a mobile application for college where all professors, administrator, and students can interact with each other with various activities. Android operating system software stack consist of java application running on java based object oriented application running on a java based object oriented application on top of java core libraries. Libraries written in C include the surface manager, open core media framework, SQL relational database management system, OpenGL ES 2.0 3D graphics API, Web-Kit layout engine, SSL, Bionic labs. System has integration of mobile in parallel processing that gives new dimension of possibilities to share processing power.

#### A. Purpose

First M-University establishes the connection between the Students, Professor, and administrator of the college through mobile. Second Administrator task is to prepare and monitor the course contents, update/delete student and third Professor task is to prepare Test papers and monitors students by viewing their performance and the duration of listening to videos in graphical format.

#### B. Existing system

Presently there is no mobile application which provides communication between students and professors. It does not provide academic educational access to the student. Time required learning concept. Students have to wait for long time to give test and view their performance among others.

#### C. Proposed system

Provide Academic Educational access to the student through video lectures. Provides options to the college professor's to view and monitor the student performances. After completing the video Lecture/Session, the student can attend the test about that particular session and can view his performance among other students. Through this application users can enhance self directed Learning.

It focuses on the capabilities needed by Administrator, Professor and Student. The detailed of how the Android interface mobile is used to success the data on cloud fulfills these needs are detailed in the use case and supplementary specification. The process involves creating three modules for demonstrating the administrator task is implemented i.e., to prepare and monitor the course content, update/delete student. In the second module the professor task is implemented i.e., to prepare question papers i.e. adding question and monitors students by viewing their performance through graphs and the duration of listening to videos. In the third module Student Part is implemented i.e., Student has to create his own account with a unique user name and password and has to log in with the same; here authentication process is going to be implemented. Viewing the course contents, listening to videos, giving comments and rating about particular session and even can view the comments and rating given by other students, giving test and viewing their performance by comparing with students through graphs.

- Maintainability

All the modules must be clearly separate to allow different user interfaces to be developed in future.

- Performance

Care is taken so as to ensure a system with comparatively high performance.

- Usability  
The ease of use and training the end users of the system is usability. System should have qualities like- learning ability, efficiency, affect, control
- Portability  
It is the ability of the system to run under different computing environments. The environment types can be either hardware or software, but is usually a combination of two.
- Security  
It is the factors that protect the software from accidental or malicious access, use, modification, destruction, or disclosure. Security can be ensured as the project involves authenticating the users.

## II ARCHITECTURE

Project architecture represents no. of components we are using as part of our project. The architecture describes about the flow of the request processing in the project. Describes about the communication of the components

Generally 3-Tier Architecture is used

1. Client (Mobile)
2. Server (GlassFishV3.0)
3. Back end Database (My SQL).

So it explains requests will be taken from the Client and it will be process in GlassFishV3.0 server the server will contact the database and process the request.

The request will be getting back to the service method and in turn it will send to the client nothing but mobile.

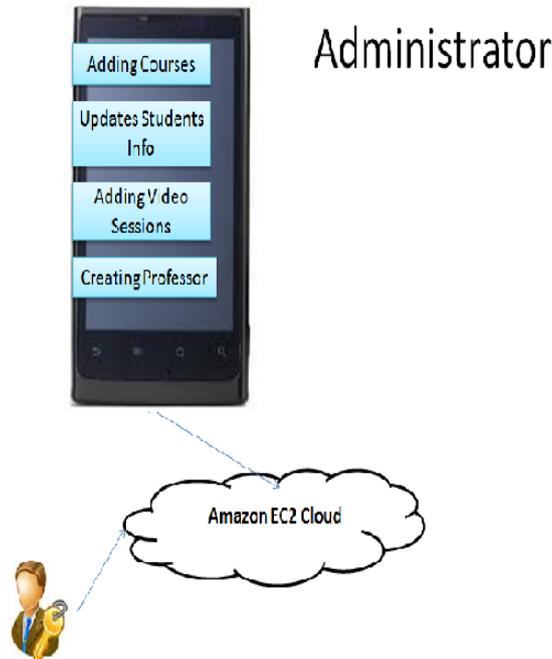


Fig. 1 Architecture of Administrator

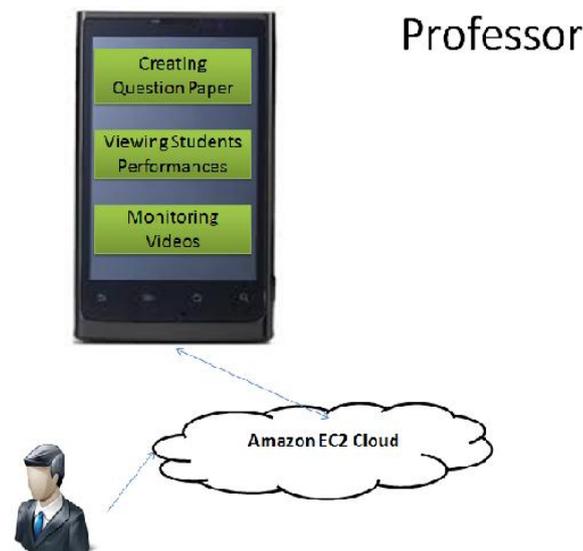


Fig. 2 Architecture of Professor

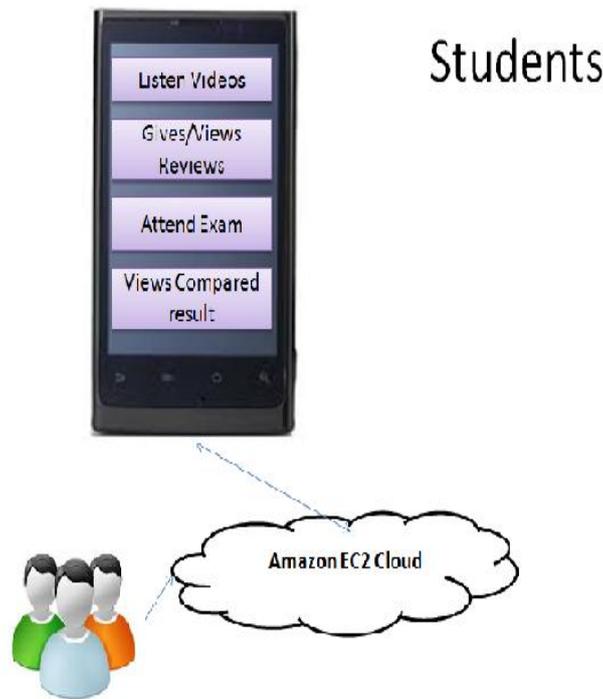


Fig. 3 Architecture of Students

### III CLOUD (AMAZON EC2 SERVICES)

Cloud computing is a technology that uses the internet and central remote servers to maintain data and applications.

Cloud infrastructure is an increasingly compelling option for many organizations and applications. This technology allows for much more efficient computing by centralizing storage, memory, processing and bandwidth



Figure4. Cloud Architecture

Cloud Architectures are designs of software applications that use Internet-accessible on-demand services. Applications built on Cloud Architectures are such that the underlying computing infrastructure is used only when it is needed (for example to process a user request), draw the necessary resources on-demand (like compute servers or storage), perform a specific job, then relinquish the unneeded resources and often dispose themselves after the job is done. While in operation the application scales up or down elastically based on resource needs.

#### A. Need of cloud architecture

Cloud Architectures address key difficulties surrounding large-scale data processing. In traditional data processing it is difficult to get as many machines as an application needs. Second, it is difficult to get the machines when one needs them. Third, it is difficult to distribute and co-ordinate a large-scale job on different machines, run processes on them, and provision another machine to recover if one machine fails. Fourth, it is difficult to auto-scale up and down based on dynamic workloads. Fifth, it is difficult to get rid of all those machines when the job is done. Cloud Architectures solve such difficulties. Applications built on Cloud Architectures run in-the-cloud where the physical location of the infrastructure is determined by the provider. They take advantage of simple APIs of Internet-accessible services that scale on-demand, that are industrial-strength, where the complex reliability and scalability logic of the underlying services

remains implemented and hidden inside-the-cloud. The usage of resources in Cloud Architectures is as needed, sometimes ephemeral or seasonal, thereby providing the highest utilization and optimum bang for the buck.

#### B. *Benefits of Cloud Architectures*

- Almost zero upfront infrastructure investment: If you have to build a large-scale system it may cost a fortune to invest in real estate, hardware (racks, machines, routers, backup power supplies), hardware management (power management, cooling), and operations personnel. Because of the upfront costs, it would typically need several rounds of management approvals before the project could even get started. Now, with utility-style computing, there is no fixed cost or startup cost.
- Just-in-time Infrastructure: In the past, if you got famous and your systems or your infrastructure did not scale you became a victim of your own success. Conversely, if you invested heavily and did not get famous, you became a victim of your failure. By deploying applications in-the-cloud with dynamic capacity management software architects do not have to worry about pre-procuring capacity for large-scale systems. The solutions are low risk because you scale only as you *grow*. Cloud Architectures can relinquish infrastructure as quickly as you got them in the first place (in minutes).
- More efficient resource utilization: System administrators usually worry about hardware procuring (when they run out of capacity) and better infrastructure utilization (when they have excess and idle capacity). With Cloud Architectures they can manage resources more effectively and efficiently by having the applications request and relinquish resources only what they need (on-demand).
- Usage-based costing: Utility-style pricing allows billing the customer only for the infrastructure that has been used. The customer is not liable for the entire infrastructure that may be in place. This is a subtle difference between desktop applications and web applications. A desktop application or a traditional client-server application runs on customer's own infrastructure (PC or server), whereas in a Cloud Architectures application, the customer uses a third party infrastructure and gets billed only for the fraction of it that was used.
- Potential for shrinking the processing time: Parallelization is the one of the great ways to speed up processing. If one compute-intensive or data-intensive job that can be run in parallel takes 500 hours to process on one machine, with Cloud Architectures, it would be possible to spawn and launch 500 instances and process the same job in 1 hour. Having available an elastic infrastructure provides the application with the ability to exploit parallelization in a cost-effective manner reducing the total processing time.

#### C. *Amazon EC2 services*

- Automate as Much As Possible- One of the biggest features of Amazon EC2 is that you can provision any number of compute instances by making a simple web service call. Automation will empower the developer to run a dynamic programmable datacenter that expands and contracts based on his needs. For example, automating your build-test-deploy cycle in the form of an Amazon Machine Image (AMI) and then running it automatically on Amazon EC2 every night (using a CRON job) will save a lot of time. By automating the AMI creation process, one can save a lot of time in configuration and optimization.
- Add Compute Instances on-The-Fly- with Amazon EC2, we can fire up a node within minutes. Hadoop supports the dynamic addition of new nodes and task tracker nodes to a running cluster. One can simply launch new compute instances and start Hadoop processes on them, point them to the master and dynamically grow (and shrink) the cluster in real-time to speed up the overall process.

## IV ANDROID MOBILE SECURITY

### A. *How Android security model works*

The Open Source Android phone has a significantly different model rather than other commercially available Mobile Phones in the market. Typically, the security in Android resides on the Linux Kernel and uses a method such as Sandbox approach.

- Phase1- Security Analysis with Android

The open nature of Android phone and well constructed sandbox security provides a quite different way of accessing the data on the phone. Unlike Windows platform, Android's Applications runs on its own instances of the Dalvik Virtual machine in which each instance represents a Linux kernel process. And the instances running on each application is completely isolated from the other application and memory

- Phase 2- Security Analysis for permission

First, the Application can give a certain type of permission to visible all database and files to other application by a general permission

- Phase 3- Security Analysis

There are many security discussions still undergoing on the Android security discussion forum that no one is clear whether the security certificates used in Android phone is secured or not. For this purpose, we applied the security principle from Public Key Infrastructure to the Android phone's permission based security. There are many security implications to note that Certificates used in Android phone.

## V. CONCLUSIONS

We proposed a system for college students to get academic details and students can view the courses and listen to video lectures through these users can enhance self directed Learning. If the students are having doubt in any subject it can be instant solved by the professor. The proposed mechanism can reduce low overhead can be deployed in Android mobile

**REFERENCES**

- [1] Machigar Ongtang, Stephen McLaughlin, William Enck and Patrick McDaniel *Semantically Rich Application-Centric Security in Android*, Department of Computer Science and Engineering, The Pennsylvania State University, University Park, PA 16802
- [2] SHANE CONDER AND LAUREN DARCEY, *ANDROID MOBILE APPLICATION DEVELOPMENT*
- [3] Marc Benioff, Carley Adler, *Behind the cloud*.
- [4] Reto Meier, *Professional Android 2 Application Development*.
- [5] <http://aws.amazon.com/ec2>
- [6] Jousha Price, *Getting started with Amazon EC2*.