

## **CLOUD BASED COMPILER**

**Sajid Abdulla, Srinivasan Iyer, Sanjay Kutty**

S.I.E.S, Graduate School of Technology, Nerul, Navi Mumbai, Maharashtra, India.  
sajid.qwerty@gmail.com, srinivasan.iyer91@gmail.com, sanjay\_kutty2003@yahoo.com

### *Abstract*

*Compilers are used to run programs and convert them from a text format to executable format. A compiler that is to be installed manually on every system physically requires a lot of space and also configuring of it if not installed using default parameters. Also once a program is compiled it becomes platform dependent. It is also not easy to carry the same program code to multiple systems if situation doesn't permit the usage of a single system. Another drawback is that we would need to install a different compiler on each language on which we wish to work.*

*We propose a solution to this in the form of a cloud based compiler.*

*Cloud computing is a model for enabling convenient, on demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort. Our project aims to create an online compiler which helps to reduce the problems of portability of storage and space by making use of the concept of cloud computing. The ability to use different compilers allows the programmer to pick up the fastest or the most convenient tool to compile the code and remove the errors. Moreover a web based application can be used remotely through any network connection which is platform independent. The errors/Output of the compiled program can be stored in a more convenient way. Also the trouble of installing a compiler on each computer is avoided. Thus these advantages make this application ideal for conducting online examinations.*

*We would be implementing a private cloud on which the software would be hosted. The software would be provided to the end user using a SAAS cloud. The software would contain a system that has a text editor and a terminal. The user would be given an option to select the language in which he wants to compile the program. The software will compile the program and return the output to the user. Additional functionalities such as monitoring of the system, user usage, user forums, and collaborative development can be added as needed.*

**Keywords:** Compiler, Cloud, Azure, Multiple Language Support.

### **I. PROBLEM DEFINITION**

Compilers are used to run programs and convert them from a text format to executable format. A compiler that is to be installed manually on every system physically requires a lot of space and also configuring of it if not installed using default parameters. Also once a program is compiled it becomes platform dependent. It is also not easy to carry the same program code to multiple systems if situation doesn't permit the usage of a single system. Another drawback is that we would need to install a different compiler on each language on which we wish to work.

Our implementation is a private cloud on which the application would be hosted. The software would be provided to the end user using a SAAS cloud. The software would contain a system that has a text editor and a terminal. The user would be given an option to select the language in which he wants to compile the program. The software will compile the program and return the output to the user. Additional functionalities such as monitoring of the system, user usage, user forums, and collaborative development can be added as needed.

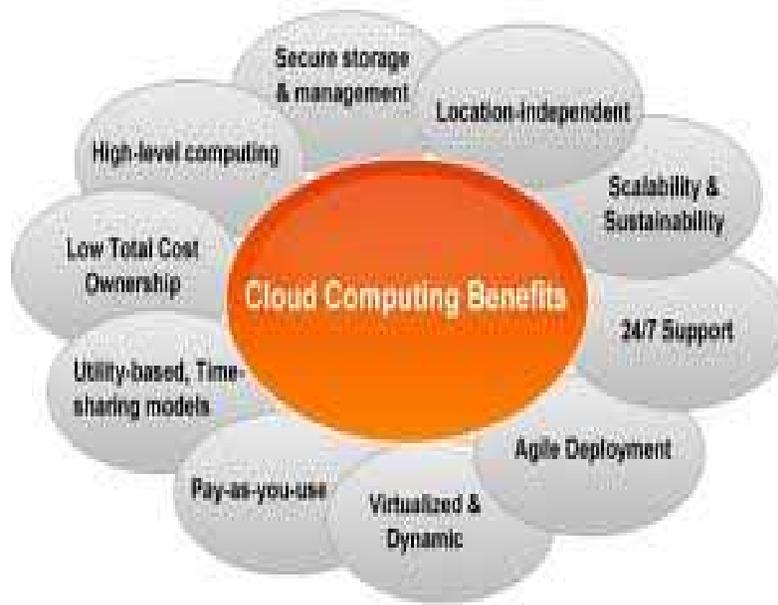
## **II. MOTIVATION AND OBJECTIVE**

As Cloud computing is a model for enabling convenient, on demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort. We decided to make a project that aims to create an online compiler which helps to reduce the problems of portability of storage and space by making use of the concept of cloud computing. The ability to use different compilers allows the programmer to pick up the fastest or the most convenient tool to compile the code and remove the errors. Moreover a web based application can be used remotely through any network connection which is platform independent. The errors/Output of the compiled program can be stored in a more convenient way. Also the trouble of installing a compiler on each computer is avoided. Thus these advantages make this application ideal for conducting online examinations.

Cloud based compiler mainly deals with providing a platform to compile and execute programs that is not dependent on any platform related restriction or complication.

The compiler that we are going to implement would be a Java compiler that is hosted on a private cloud implemented on Windows Azure platform. The compiler can be used to implement and run java programs and get the executable files or directly view the output.

## **III. ADVANTAGES OF USING CLOUD COMPUTING**



Cloud computing exhibits the following key characteristics:

- **Agile development** improves with users' ability to re-provision technological infrastructure resources.
- **Application programming interface (API)** accessibility to software that enables machines to interact with cloud software in the same way the user interface facilitates interaction between humans and computers. Cloud computing systems typically use REST-based APIs.
- **Low total Cost** is claimed to be reduced and in a public cloud delivery model capital expenditure is converted to operational expenditure. This is purported to lower barriers to entry, as infrastructure is typically provided by a third-party and does not need to be purchased for one-time or infrequent intensive computing tasks. Pricing on a utility computing basis is fine-grained with usage-based options and fewer IT skills are required for implementation (in-house
- **Device and location independence** enable users to access systems using a web browser regardless of their location or what device they are using (e.g., PC, mobile phone). As infrastructure is off-site (typically provided by a third-party) and accessed via the Internet, users can connect from anywhere.
- **Virtualized and dynamic** technology allows servers and storage devices to be shared and utilization be increased. Applications can be easily migrated from one physical server to another.
- **Multi-tenancy, scalable and sustainable** enables sharing of resources and costs across a large pool of users thus allowing for multiple advantages

- **Centralization** of infrastructure in locations with lower costs (such as real estate, electricity, etc.)
- **Peak-load capacity High level computing** increases users need not engineer for highest possible load-levels.
- **Utilization and efficiency** improvements for systems that are often only 10–20% utilized.
- **Reliability** is improved if multiple redundant sites are used, which makes well designed cloud computing suitable for business continuity and disaster recovery.
- **Scalability and elasticity** via dynamic ("on-demand") provisioning of resources on a fine-grained, self-service basis near real-time, without users having to engineer for peak loads.
- **Performance** is monitored and consistent and loosely coupled architectures are constructed using web services as the system interface.
- **Secure storage and management** could improve due to centralization of data, increased security-focused resources, etc., but concerns can persist about loss of control over certain sensitive data, and the lack of security for stored kernels. Private cloud installations are in part motivated by users' desire to retain control over the infrastructure and avoid losing control of information security.
- **Maintenance** of cloud computing applications is easier, because they do not need to be installed on each user's computer and can be accessed from different places

#### IV. EXTENSIONS OVER TRADITIONAL COMPILERS

Cloud based compiler mainly deals with providing a platform to compile and execute programs that is not dependent on any platform related restriction or complication.

The cloud compiler can be used by any user who has subscribed to the compiler and can use it for a specific period of time. The functionalities that are provided by the cloud are:-

- **Compile on the Go:** The cloud will have an IDE where the source code can be written. The code can be implemented on the cloud itself and the output can be viewed at same time.
- **File Management:** The source code can be saved in a text format or in .java format in the space that is allocated to each user. The user can extended his storage space by buying a bigger storage space.
- **Forums:** The user can log on into a forum where he can discuss the various problems or the solution to problems he is facing.

- **Security:** We provide by security by implementing a sandbox that is a security mechanism for separating running programs. It is often used to execute untested code, or untrusted programs from unverified third-parties, suppliers, untrusted users and untrusted websites.

Also we use signature testing mechanism i.e. a type of antivirus program that searches a system for virus signatures that have attached to executable programs and applications such as e-mail clients. A virus scanner can either search all executable codes when a system is booted or scan a file only when a change is made to the file as viruses will change the data in a file.

The various advantage and disadvantage of the systems are:-

### **ADVANTAGES**

- No need to download the SDK of any compiler.
- No issues of setting path variables.
- Elegant and simple to use GUI for better coding.
- Support from other users in forums.

### **DISADVANTAGE**

- Not cost Effective.
- Requirement of an Internet connections

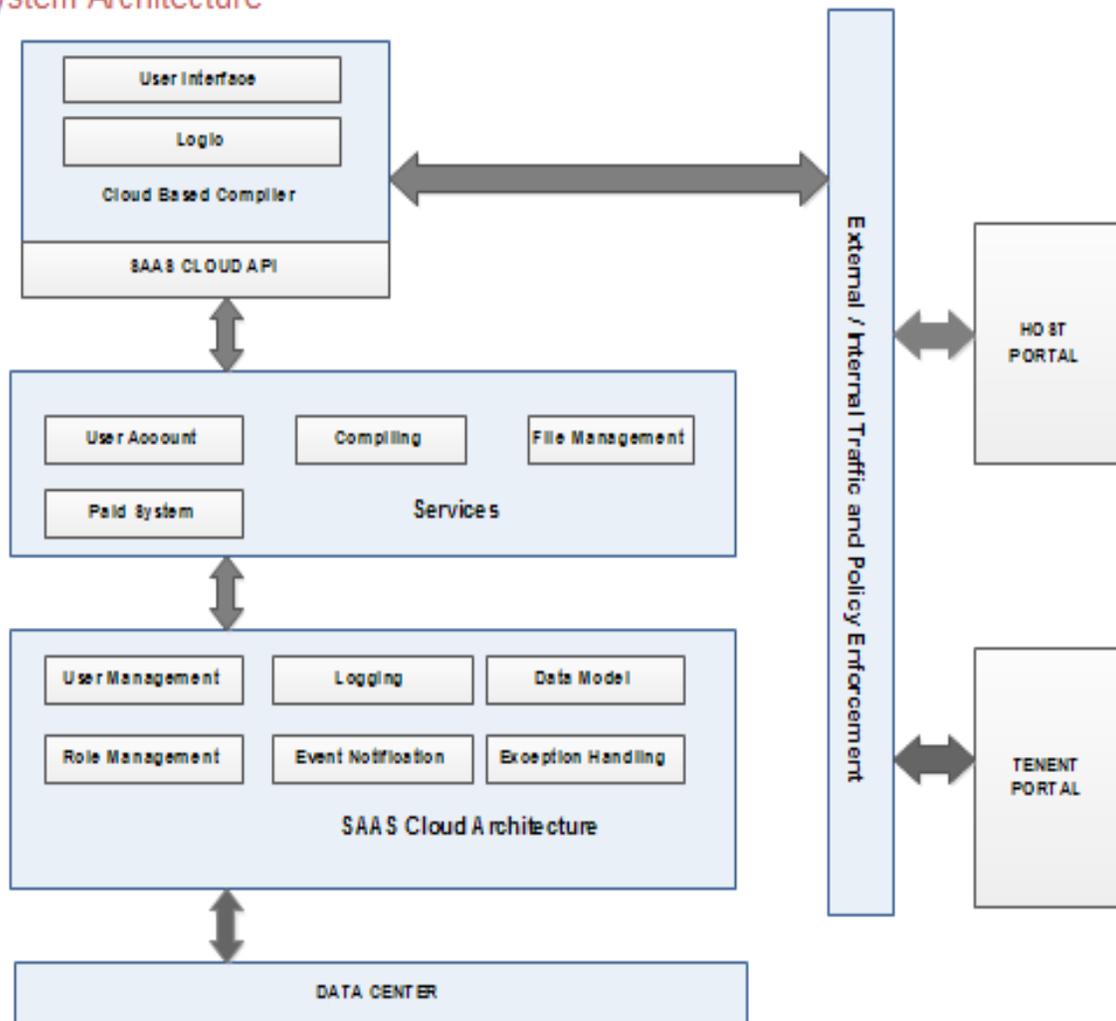
## **V. MODELLING DIAGRAMS**

### **System Architecture**

An architectural model (in software) is a rich and rigorous diagram, created using available standards, in which the primary concern is to illustrate a specific set of tradeoffs inherent in the structure and design of a system or ecosystem. An architectural model is an expression of a viewpoint in software architecture.

The System Architecture of the proposed system is as shown.

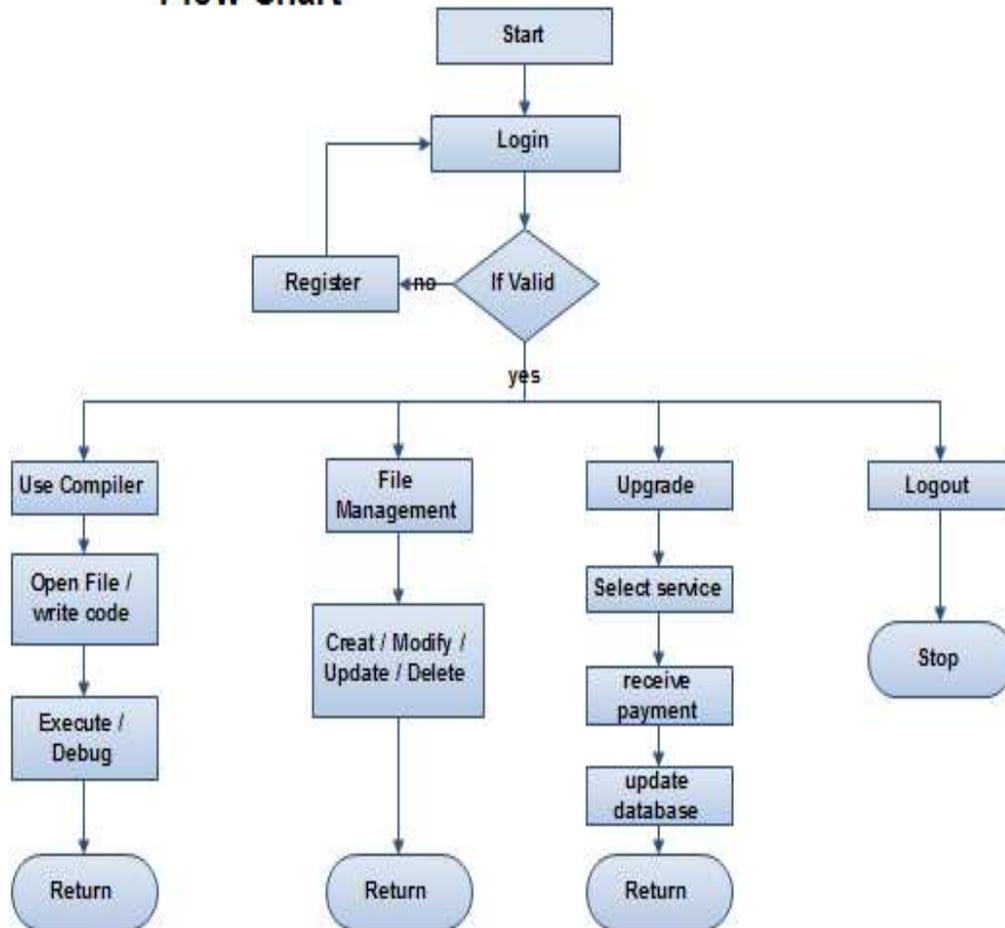
### System Architecture



### Flow Chart

A flowchart is a type of diagram that represents an algorithm or process, showing the steps as boxes of various kinds, and their order by connecting them with arrows. This is a diagrammatic solution to the stated problem. Process operations are implied by the sequencing of operations.

## Flow Chart

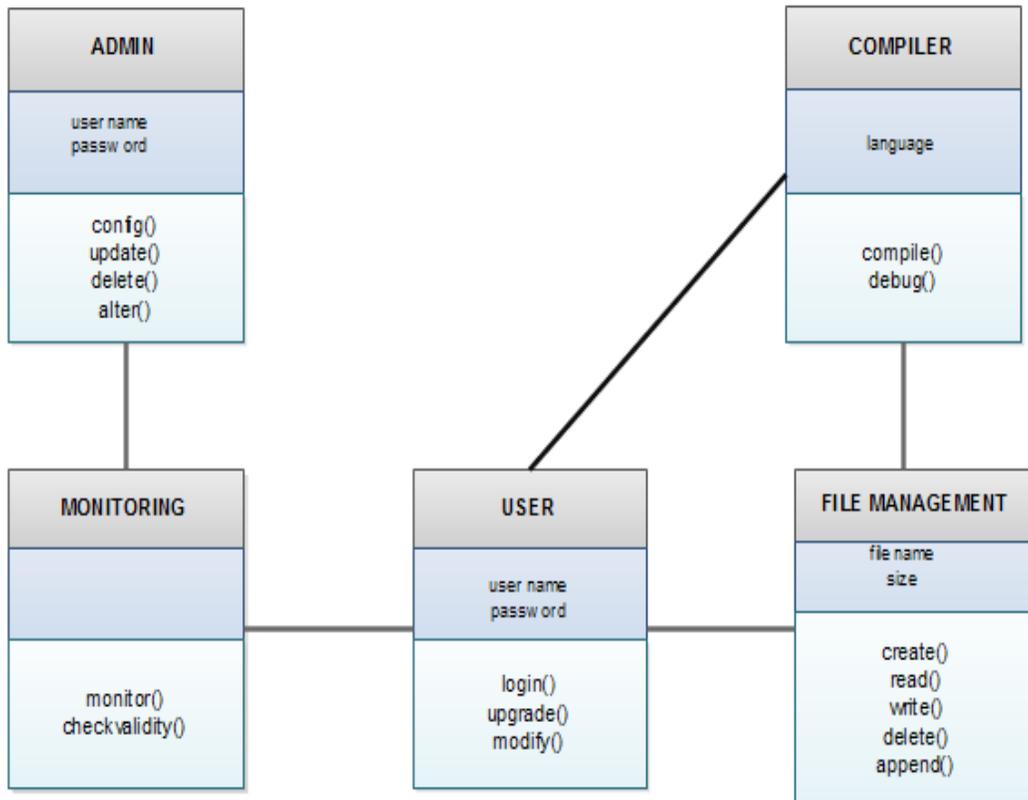


## Class Diagram

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes.

Class Diagram for the system is as follows

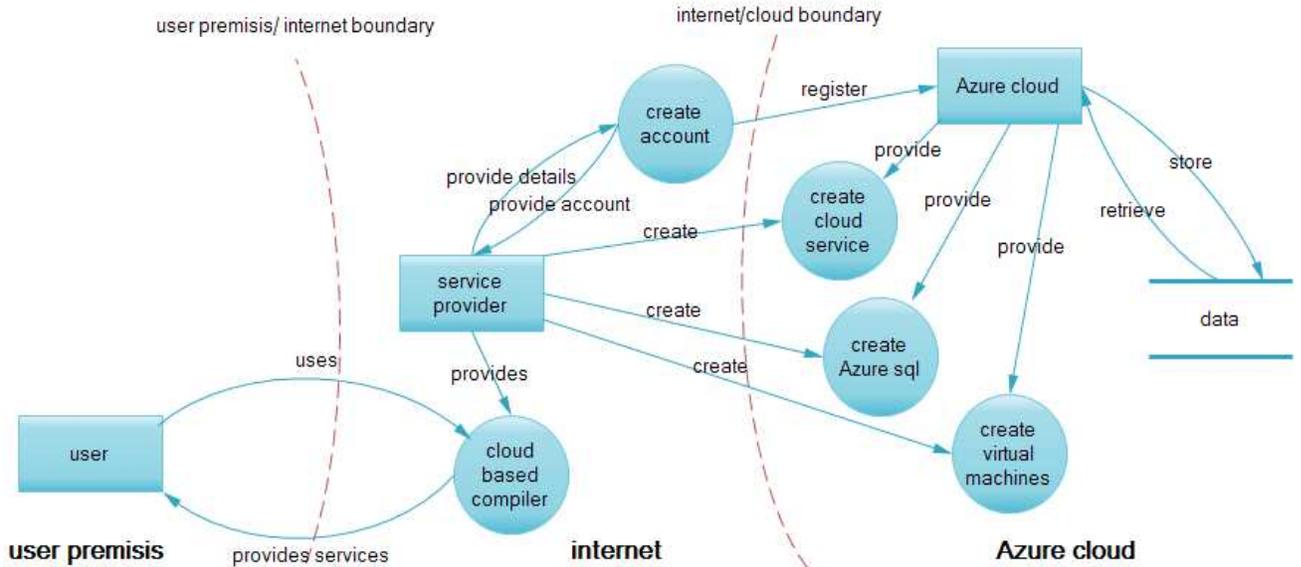
## Class Diagram



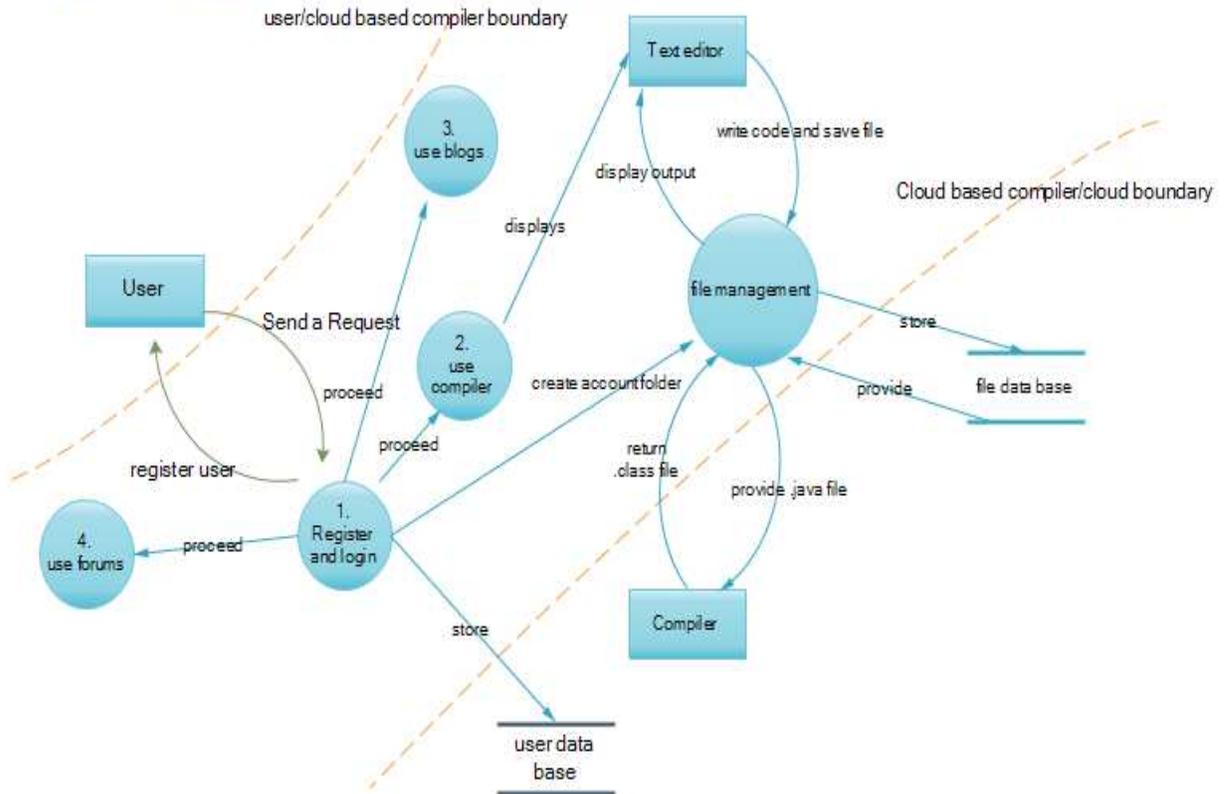
## Data Flow Diagram

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. Often they are a preliminary step used to create an overview of the system which can later be elaborated.

### Data Flow Diagram

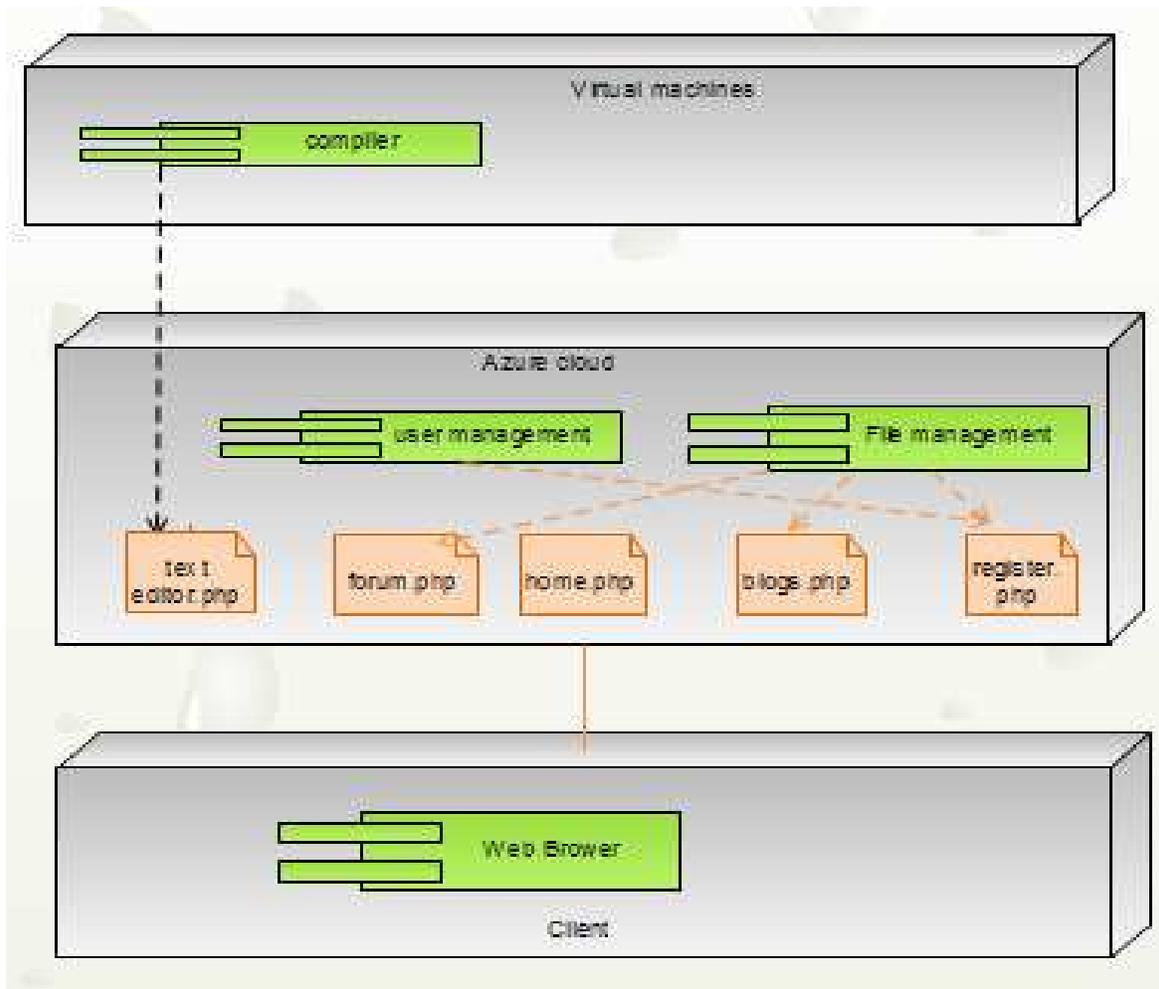


### Data Flow Diagram for Cloud Based Compiler



## Deployment Diagram

A deployment diagram in the Unified Modeling Language models the physical deployment of artifacts on nodes. To describe a web site, for example, a deployment diagram would show what hardware components ("nodes") exist (e.g., a web server, an application server, and a database server), what software components ("artifacts") run on each node (e.g., web application, database), and how the different pieces are connected



## VI. SYSTEM REQUIREMENTS

### Hardware requirements:

- AMD Athlon X2 Dual core processor

- Virtual Storage disk 10 GB
- 4GB RAM

**Software Requirements:**

- Linux Distribution (Ubuntu 12.04 used)
- LAMP: Linux, Apache, MySQL, PHP.
- Compilers for the languages to be supported.
  - IcedTeajdk 7 (for JAVA)
  - G++ (for C++)
  - GCC (for C)
  - PyDEV (for python)

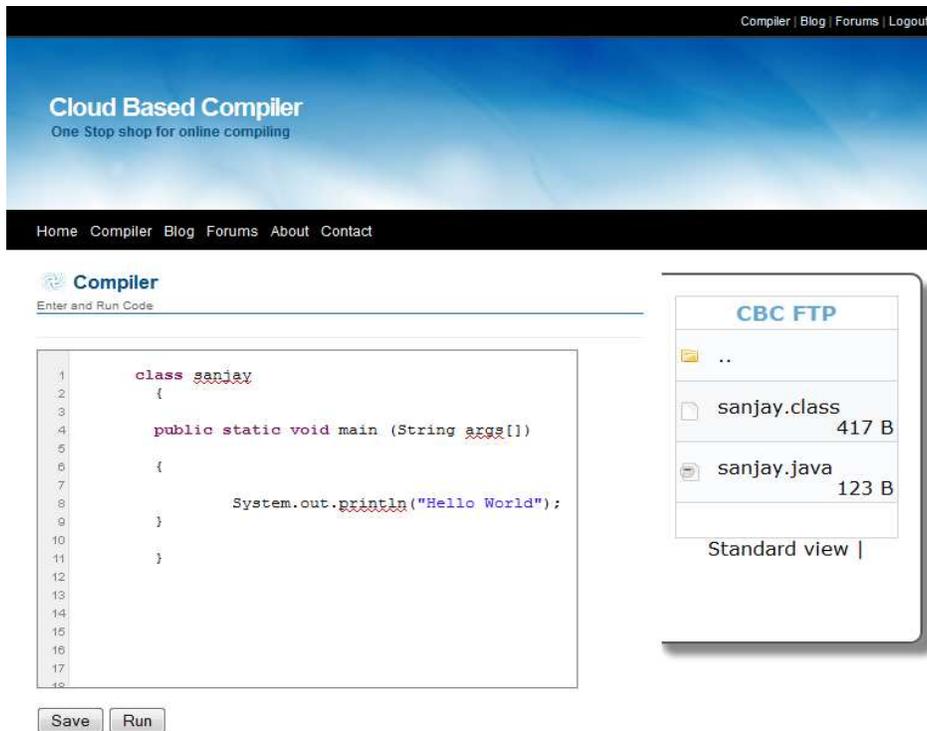
**VII. FINAL IMPLEMENTATION**

The extensive support for deploying PHP based applications in Windows Azure triggered the idea of building the application using PHP. Also the web application to create an online compiler was developed. The code is a server side script which sets the path and gives variety options. The only dilemma lied in the deploying of compilers. The compilers are hosted on virtual machines created in our windows Azure cloud account. The path is set to these compilers using environment variables on the virtual machine.

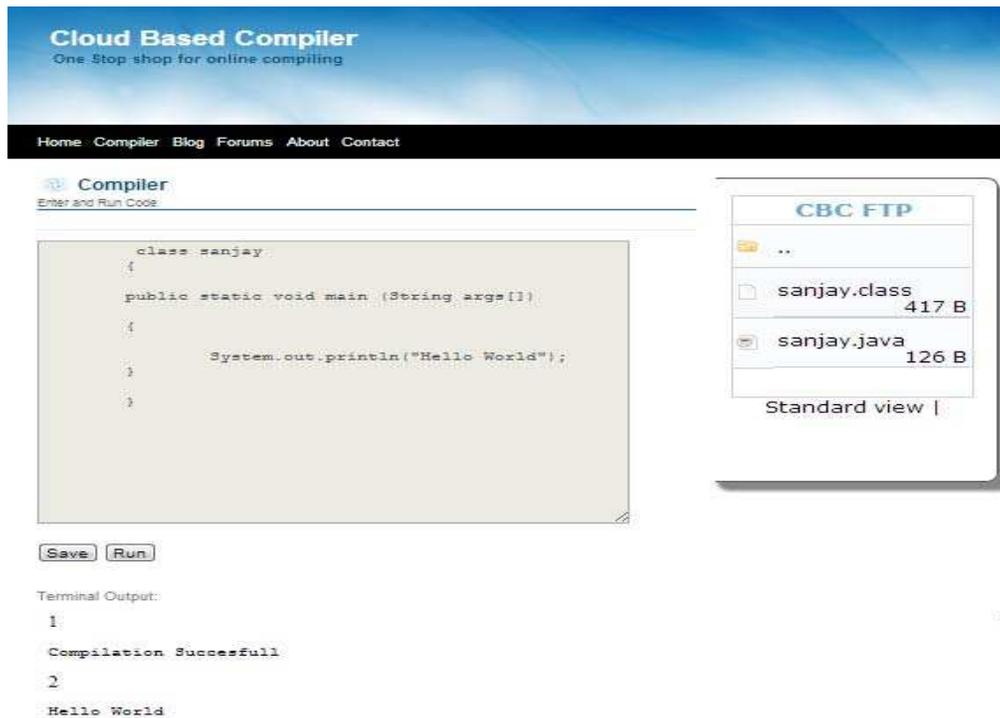
**Screenshots**



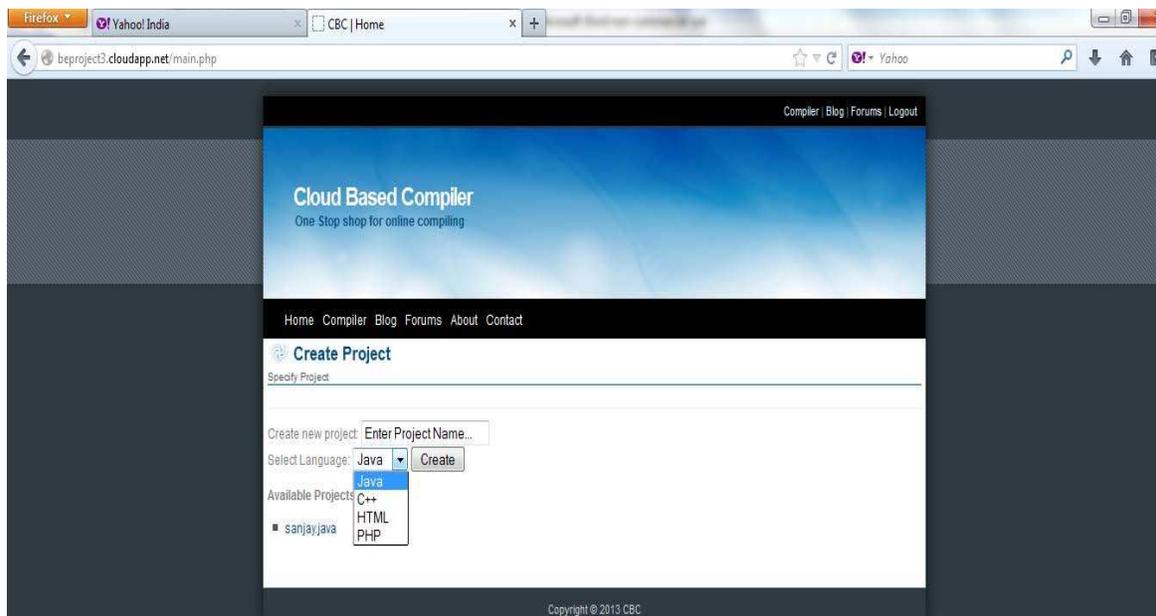
**Home Page**



**File Access**



## Compilation



## Multiple Language Support

Security mechanism implemented to avoid malicious programs from being run on the compilers using a virtual machine as sand box and virus signature scanning API of VirusTotal.com.

## VIII. CONCLUSION

The main reason for creating the project is to provide a centralized compiling scheme. Also, it will act as a centralized repository for all the codes written. The other major advantage that this system will have over the others is that it will make the users system lightweight i.e. there will be no need to maintain separate compilers at the client side. Also, the process of maintenance and distribution of dynamic usernames and passwords will be greatly simplified. Also, authentication and personalized task distribution will be made possible. A compiler, which is the heart of any computing system, transforms source code from a higher level language to a lower, machine level language. This is mainly done in order to create executable files which can then be run in order to execute the program and its instructions. As compared to the current scenario where each compilers required to be installed on each machine separately this would eliminate the need to install compilers separately. So we can check our code at the centralized server. Another advantage of such project is that whenever the compiler package is to be upgraded it can be done easily without again installing it on each and every machine.

## IX. FUTURE WORK

- 1) **Provide more compilers:** The application can be extended to provide compilers for Python, FORTRAN, COBOL, C# etc. Most compilers require authorizing their certain load libraries and System name parameters.
- 2) **Implement a security mechanism:** This is an issue that may question the survivability of the application. The user premises are usually secure by firewall protection. And data is secure in the cloud as Azure provides various data confidentiality and integrity mechanisms. Any malicious could be run on the compiler by an attacker to corrupt the entire system directed to the centralized compiler. User authentication could be maintained using passwords, passTicket authentication or FTP authentication. A complex system providing Sandbox Testing mechanism can be used to avoid malicious program attacks.
- 3) **Provide efficiency meters and code optimizer add ons:** They would provide the complexity of the code and help the user to improve writing better codes by providing a measure to understand their coding skills

- 4) **Create Web API's:** We could also provide the above project by using API's in the cloud. This helps to create a more interactive way of providing software as a service.
- 5) **Mobile Applications:** Mobile Applications can be developed so that the users can create and execute their applications using their mobiles with greater ease. This would provide them with a better ease of use as they would not have to go through the hassle of logging into a Desktop optimized site through their mobiles.
- 6) **Collaborative Editing:** Collaborative editing features can be added so that large project groups can work on the project online and with ease.

## REFERENCES

1. **A.Rabiyathul and Basariya k.Tamil Selvi**, “*Centralized C# Compiler Using Cloud Computing*”, International Journal of Communications and Engineering Volume 06– No.6, Issue: 02 March 2012.
2. “*Challenges in deploying SaaS applications*”, Imaginea Inc. white paper
3. “*Future of cloud computing*”, [www.roseindia.net](http://www.roseindia.net)
4. **Donovan Kretsman**, “*SaaS | don't let the Cloud rain on your parade*”, [www.focalscope.com](http://www.focalscope.com)'s blog on SaaS
5. “*Advantages of SaaS*”, [www.cloudtweaks.com](http://www.cloudtweaks.com)
6. **M. Tim Jones**, “*Anatomy of a cloud storage infrastructure*”, [www.ibm.com](http://www.ibm.com)
7. “*Quickly build and deploy Software as a Service applications*”, Ironspeed Inc. white paper
8. “*Tutorials Windows Azure*” <http://www.windowsazure.com/en-us/develop/net/tutorials/get-started>
9. Online multiple platform provider: [www.compilr.com](http://www.compilr.com)
10. Online compiler: [www.cloudcompiling.com](http://www.cloudcompiling.com)