# MMBcloud-Tree: Authenticated Index For Verifiable Cloud Service Selection

**N.Sivasubramani[1], R.Prashanth[2], R.Paul Antony[3], Mr.T.John Berkmens M.Tech.,(Ph.D)[4]**
[1, 2, 3] Dept of IT
[4]Assistant professor, Dept of IT
[1, 2, 3, 4] JEPPIAAR SRR Engineering College

***Abstract***- *Cloud brokers have been recently introduced as an additional computational layer to facilitate cloud selection and service management tasks for cloud consumers. However, existing brokerage schemes on cloud service selection typically assume that brokers are completely trusted, and do not provide any guarantee over the correctness of the service recommendations. It is then possible for a compromised or dishonest broker to easily take advantage of the limited capabilities of the clients and provide incorrect or incomplete responses. To address this problem, we propose an innovative cloud service selection verification (CSSV) scheme and index structures (MMBcloud-tree) to enable cloud clients to detect misbehaviour of the cloud brokers during the service selection process. We demonstrate correctness and efficiency of our approaches both theoretically and empirically. After verification We block those fake brokers account. Our theoretical and experimental results demonstrate the effectiveness and efficiency of our schemes compared with the state-of-the art.*

***Keywords***- Communication, Database

## I. INTRODUCTION

Online chat may refer to any kind of communication over the Internet that offers a real-time transmission of text messages from sender to receiver. Chat messages are generally short in order to enable other participants to respond quickly. Thereby, a feeling similar to a spoken conversation is created, which distinguishes chatting from other text-based online communication forms such as Internet forums and email. This system is basically developed for the persons who are connected to the internet. In chat app, the user must see other messages by scroll the screen only. This paper has overcome the problem of scrolling option. Adding name, date time option in search command to retrieve a particular person messages.

## II. PROJECT DESCRIPTION

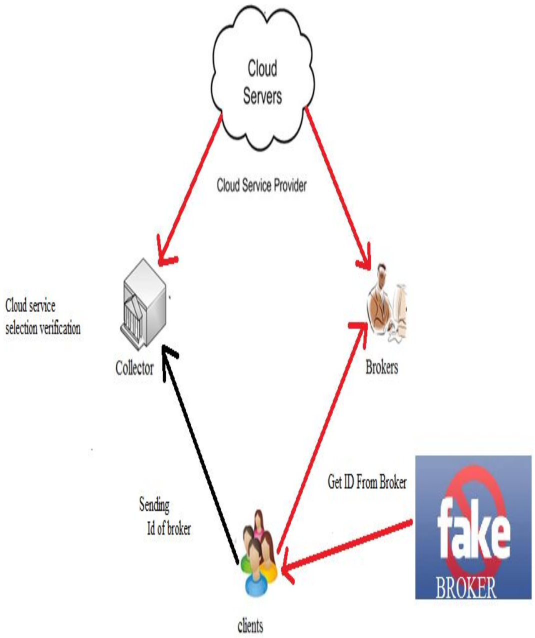
**Existing System:**

In existing brokerage schemes on cloud service selection typically assume that brokers are completely trusted, and do not provide any guarantee over the correctness of the service recommendations. It is then possible for a compromised or dishonest broker to easily take advantage of the limited capabilities of the clients and provide incorrect or incomplete responses.it difficult for potential cloud clients to weigh and decide which options suit their requirements the best. The challenges are twofold: 1) It is hard for cloud clients to gather information about all the CSPs available for their selections; 2) It is also computationally expensive to choose a suitable CSP from a potentially large CSP pool.Existing works on cloud service selection are focused only on how to select the services that satisfy customers' requirements. None of them considers security issues involved in the service selection, and none of them provides verifiable schemes to prove the correctness and completeness of their service selection results as addressed in our work

## III. PROPOSED SYSTEM

Some brokerage schemes on cloud service selection typically assume that brokers are completely trusted, and do not provide any guarantee over the correctness of the service recommendations. It is then possible for a compromised or dishonest broker to easily take advantage of the limited capabilities of the clients and provide incorrect or incomplete responses. To address this problem, we propose an innovative cloud service selection verification (CSSV) scheme and index structures (MMBcloud-tree) to enable cloud clients to detect misbehavior of the cloud brokers during the service selection process.we propose innovative authenticated index structures and verification protocols to allow clients to verify the completeness and authenticity of brokers' answers Our theoretical and experimental results demonstrate the effectiveness and efficiency of our schemes compared with the state-of-the art. After verification We block those fake brokers account.

## IV. ARCHITECTURAL DIAGRAM



**Hardware Requirements**

- ✓ Processor: Pentium IV
- ✓ RAM : 512 MB
- ✓ Hard Disk: 80 GB Computer system

**Software Requirements**

| | | |
|---|---|---|
| Front End | : | Java |
| Environment | : | Eclipse/Net Beans |
| Back End | : | MySQL |
| Operating System | : | Windows XP |

**Tools required**

- ✓ Eclipse
- ✓ HeidiSql

## V. MODULES

The modules involved in the proposed system is

1. User Login
2. Broker
3. Collector
4. Result Verification

### 5.1 User Login

First the user should register their personal details, using the username and password they will allow to login. Users may create their own accounts by giving their personal details like mobile number, address, date of birth etc…The accounts have login page to enter into the account, login page contains user name and password for security purpose. However, this assumption does not hold when a user's personal information can be acquired by an acquaintance, or by a stranger with access to public user profiles.

### 5.2 Broker

Cloud brokers have been recently introduced as an additional computational layer to facilitate cloud selection and service management tasks for cloud consumers. However, existing brokerage schemes on cloud service selection typically assume that brokers are completely trusted, and do not provide any guarantee over the correctness of the service recommendations. It is then possible for a compromised or dishonest broker to easily take advantage of the limited capabilities of the clients and provide incorrect or incomplete responses. Cloud Broker is the middleman between the cloud clients and CSPs. For example, we consider the case when the broker provides service selection services to the cloud clients. In these settings, our main problem is how to ensure that cloud brokers recommend authentic and complete services available to clients, according to the clients' specific requests (namely service selection queries).

### 5.3 Collector

The collector is responsible for collecting the profiles (including the properties discussed in Section 3 and some other information such as user ratings, etc) of CSPs, and constructing an authenticated CSP profile database that ensures the integrity of the CSPs' information. The collector sells the authenticated CSP profile database to multiple cloud brokers. Step 1 (Request). Upon receiving the latest information of a CSP, the collector contacts the broker and issues an update request for the CSP. The broker then identifies the leaf node storing the corresponding CSP, and sends the verification path of the leaf node to the collector. The verification path of a leaf node includes the nodes in leaf-to-root path as well as their sibling nodes. Step 2 (Update). The collector verifies the correctness of the verification path (as in Section 5.3), and updates the corresponding CSP entry to the latest information. The update operations in the MBcloud-tree and MMBcloud-tree are very similar to that in traditional Bþ-trees. The only difference is that the MBcloud-tree and MMBcloud-tree need to recompute the hash values for the updated nodes and their ancestor nodes.

### 5.4 Result Verification

In the CSSV scheme, we introduce one more entity, the collector, besides CSPs, cloud clients and cloud brokers. The collector acts like a certificate authority and is assumed fully trusted, which is inline with the recent work . The collector is associated with a pair of public and private keys, and its public key is made available to CSPs, cloud brokers and cloud clients. Specifically, the CSSV scheme includes the following three phases: 1) Database Construction by the collector. The collector sells the authenticated CSP profile database to multiple cloud brokers. 2) Service Selection by the broker. Each cloud broker handles a potentially large amount of online clients requests. Results Verification by the clients. The clients execute result verification algorithms to verify the correctness of the query results returned by the broker.

## VI. TESTING

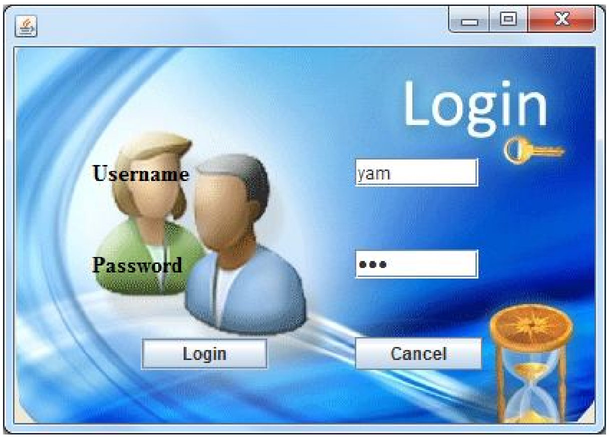
Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input produces valid outputs. .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

## ADVANTAGES

- To Prevent Data Leakage by involving third party.
- Data owner Receive secure data.

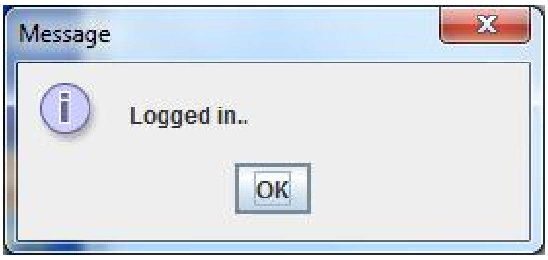
## APPLICATION DESCRIPTION

## USER LOGIN:



Username yam

Password •••

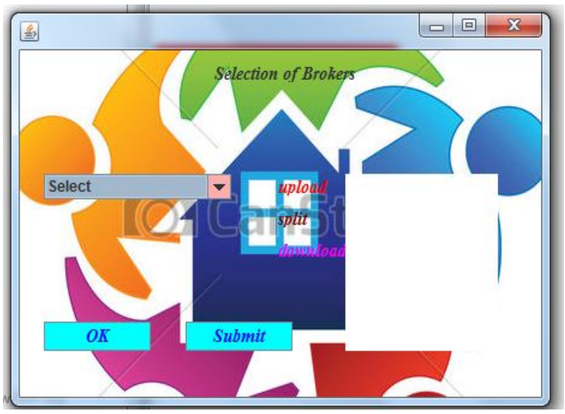


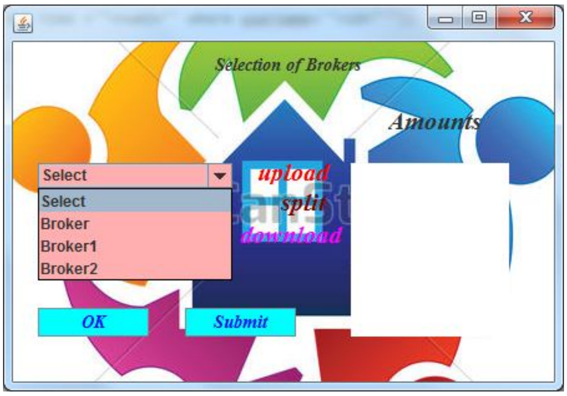
Message

Logged in..

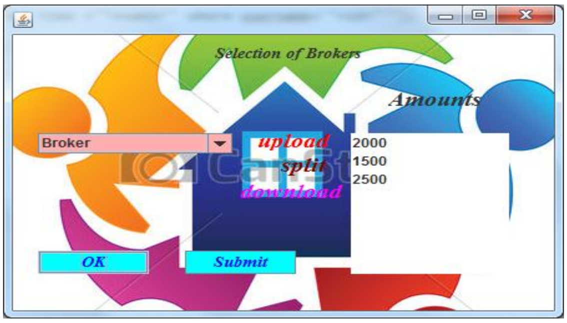
The accounts have login page to enter into the account, login page contains user name and password for security purpose.

## BROKER:



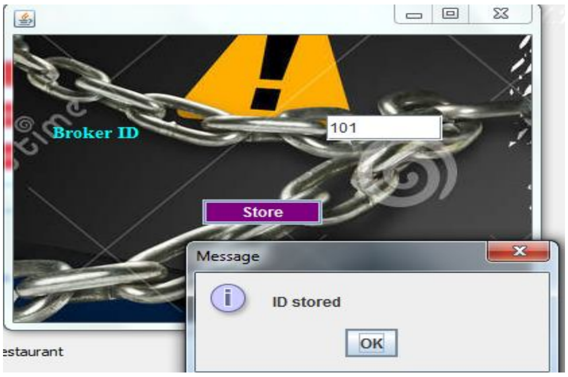


This figure contains Client will select the various types of brokers among the amounts.



Broker

2000
1500
2500

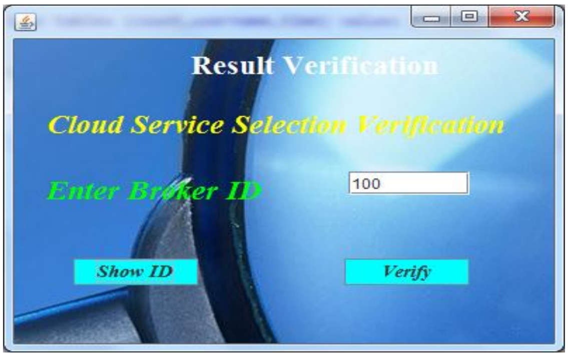


Broker ID 101

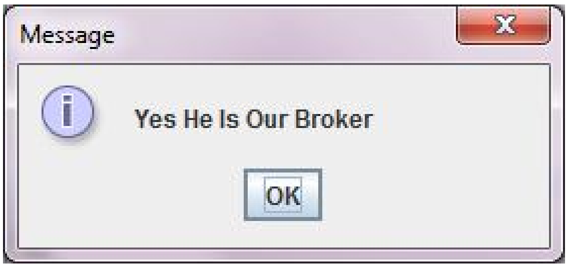
Message

ID stored

## RESULT VERIFICATION:



Enter Broker ID 100

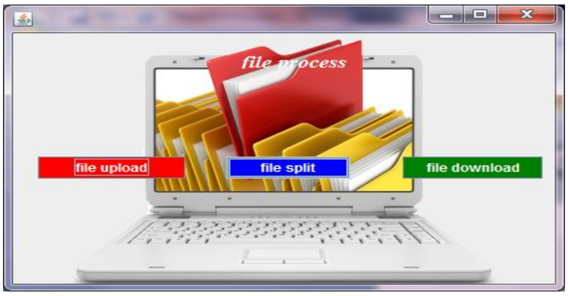


Message

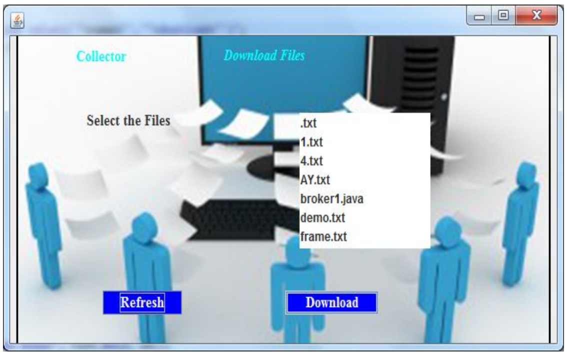
Yes He Is Our Broker

This figure contains Collector checks the result verification as the broker fake or real.

## FILE PROCESS:

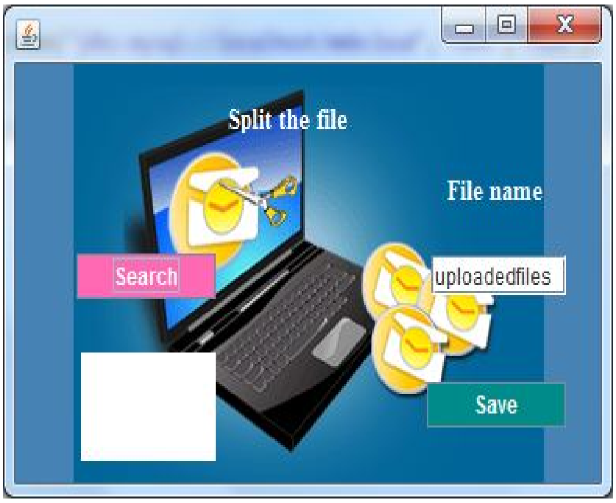


## FILE DOWNLOAD:



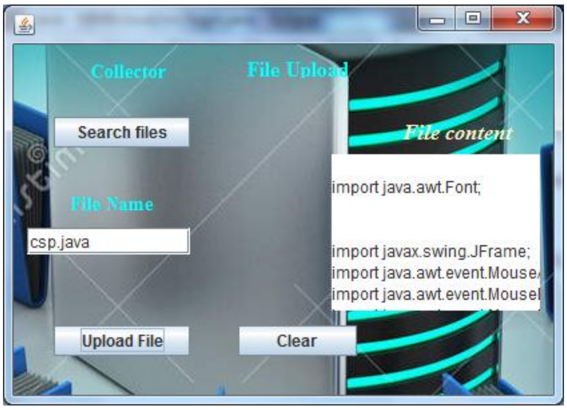
```
.txt
1.txt
4.txt
AY.txt
broker1.java
demo.txt
frame.txt
```

## FILE SPLIT:



File name uploadedfiles

## FILE UPLOAD:



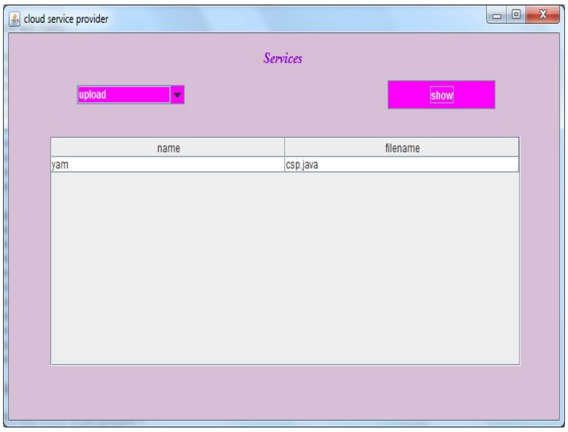
File Name csp.java

File content

```
import java.awt.Font;

import javax.swing.JFrame;
import java.awt.event.Mouse
import java.awt.event.Mousel
```

## CLOUD SERVICE PROVIDER:



| name | filename |
|---|---|
| yam | csp.java |

All database and files that cane be uploaded can be stored in CSP's entity.

## VII. CONCLUSION

We presented an innovative cloud service selection verification system to achieve cheating-free cloud service selection under a cloud brokerage architecture. The core of our system is an efficient authenticated index structure to ensure the authenticity, the satisfiability and the completeness of the service selection results. Our theoretical and experimental results demonstrate the effectiveness and efficiency of our schemes compared with the state-of-theart.

## VIII. FUTURE ENHANCEMENT

As part of our future work, we plan to consider a verifiable scheme for best service selection query whereby the broker returns only the best CSP instead of all candidate CSPs with respect to a client's request.

## REFERENCES

[1] I. Petri, M. Punceva, and O. F. Rana, "Broker emergence in social slouds," in Proc. 6th Int. Conf. Cloud Comput., 2016, pp. 669–676.

[2] L. Qu, Y. Wang, and M. A. Orgun, "Cloud service selection based on the aggregation of user feedback and quantitative performance assessment," in Proc. IEEE Int. Conf. Serv. Comput., 2013, pp. 152– 159.

[3] S. Sundareswaran, A. Squicciarini, and D. Lin, "A brokeragebased approach for cloud service selection," in Proc. IEEE 5th Int. Conf. Cloud Comput., Aug. 2012, pp. 558–565.

[4] Z. urRehman, O. K. Hussain, S. Parvin, and F. K. Hussain, "A framework for user feedback based cloud service monitoring," in Proc. 6th Int. Conf. Complex, Intell. Softw. Intensive Syst., 2012, pp. 257–262.

[5] A. Lenk, M. Menzel, J. Lipsky, S. Tai, and P. Offermann, "What are you paying for? Performance benchmarking for Infrastructure- as-a-Service offerings," in Proc. IEEE Int. Conf. Cloud Comput., 2011, pp. 484–491.

[6] L. Li and Y. Wang, "Subjective trust inference in composite services," in Proc. AAAI, 2010, pp. 1377–1384.

[7] A. Li, X. Yang, S. Kandula, and M. Zhang, "CloudCmp: Comparing public cloud providers," in Proc. 10th ACM SIGCOMM Conf. Internet Meas., New York, New York, USA, 2010, pp. 1–14.

[8] L. Xin and A. Datta, "On trust guided collaboration among cloud service providers," in Proc. 6th Int. Conf. Collaborative Comput.: Netw., Appl. Worksharing, 2010, pp. 1–8.

[9] C. Binnig, D. Kossmann, T. Kraska, and S. Loesing, "How is the weather tomorrow?: Towards a benchmark for the cloud," in Proc. 2nd Int. Workshop Testing Database Syst., Jun. 2009, pp. 1–6.

[10] F. Li, M. Hadjieleftheriou, G. Kollios, and L. Reyzin, "Dynamic authenticated index structures for outsourced databases," in Proc. ACM SIGMOD Int. Conf. Manage. Data, Jun. 2006, pp. 121–132.