Abstract
Cloud computing has displayed many intriguing possibilities since it was launched. Few of the cloud computing systems are centralized and might lead to bottlenecks under heavy traffic. This situation will influence security and leads to blockage of a greater part in the network especially when important routers and servers are not working. In the present years there has been a great development on Peer-to-Peer (P2P) cloud computing. In this work, we are providing a protocol to allow cloud servers easily interact and transport session state data between one another. The proposed methodology has been implemented using Visual Studio platform. The platform has been extensively tested and it has demonstrated promising results.

Introduction
A. Main Players:
Physical hardware, software or even datacenters are not necessary required in the case of cloud computing at the bottom layers and its software has been virtualized and is available when needed from providers of service. Amazon, Goggle, Microsoft, IBM are the major sources of cloud computing such as Hadoop, Eucalyptus, Enomaly ECP, Sector and Sphere, Abiquo and MongoDB.

B. Cloud Computing’s Services:
Taking into consideration the type of service provided, there are three types:
- Structure of the service
- Bottom as a service
- Software as a service
Providing competence with large datacenters to provide efficient computing requires large scale heterogeneous components. Its scalability allows more efficiency and more power when it is needed by the consumer. Applications are also distributed to systems. Cloud computing has established the theory of loose oriented services which gives rise to distinct free services.

Proposed Solution
The existing technologies are extremely valuable in cloud computing. However, their restrictions raise security concerns. They do not offer a quite data integration as they should.

In this work, we provide a methodology for deploying a secure cloud architecture. The global cloud servers on a network securely communicate and pass session state data amongst each other as needed, as shown below in Figure 1.

The implemented framework is based on the three search terms provided in Table 1

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Any</td>
<td>This will help in showing clouds that are</td>
</tr>
<tr>
<td>All</td>
<td>connected.</td>
</tr>
<tr>
<td>Global</td>
<td>This will present the computer’s connection to</td>
</tr>
<tr>
<td>Local</td>
<td>global cloud.</td>
</tr>
</tbody>
</table>

This design improves scalability because Local Cloud servers can share multiple global cloud servers which eliminates single point of failure. Through using data mitigation and dynamic placement, we can have a control and create a methodology for load balancing on the network: Mimicking Peer to peer network can help in implementing a load balanced cloud network. Implementation of Peer to peer design using IP6 protocol can be easily incorporated in the proposed framework.

Conclusion
This poster provides a methodology for the mitigation of the cloud data on a wireless LAN network using a hierarchy concept while inheriting all the added advantages of the cloud system and keeping it secure. Extensive test results show that the implementation is highly secure as entity has complete copy of the data required via a node. In case that any of the nodes is compromised, a section of the global cloud is disconnected from the network while the majority of the network is continue to be connected and secure.

References