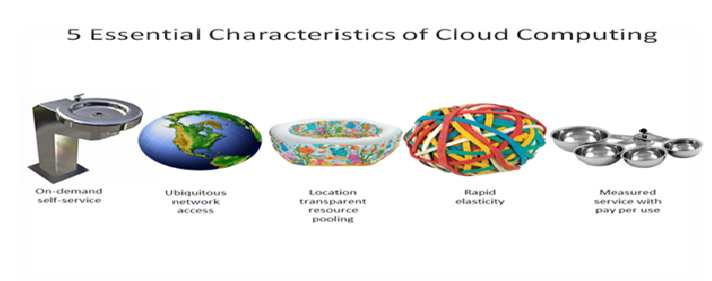
**INTRODUCTION**

In the cloud computing paradigm, providing database-as a- service (DaaS) allows a third party service provider to host database as a service, providing its customers seamless mechanisms to create, store, and access databases at cloud with adequate storage resource, convenient data access and reduced management and infrastructure costs. But database outsourcing also raises data confidentiality and privacy concerns due to data owner’s loss of physical data control. To provide privacy guarantees for sensitive data such as personal identities, health records, financial data, etc., a straightforward approach is to encrypt the sensitive data locally before outsourcing .While providing strong end-to-end privacy, encryption becomes a hindrance to data computation or utilization, i.e., it is hard to retrieve data files based on content as in the plaintext search domain. In addition, clients are also concerned about their query privacy, expecting that the database server cannot learn the data content nor the query in plaintext form. To address these challenges, the notion of searchable symmetric encryption (SSE) was first introduced by Song et al. Roughly speaking, a SSE scheme encrypts data in such a way that it can be privately queried through the use of a query-specific token generated with knowledge the secret key. In recent years, researchers have put great efforts to make SSE solutions practical. Kamara et al. designed an inverted index based searchable encryption scheme where index can be incrementally updated. Afterwards, a parallelizable and dynamic SSE scheme based on the red-black tree data structure is proposed in .In another work Cash et al. also proposed a parallelizable and dynamic SSE scheme by leveraging the generic dictionary structure, which stores newly-added document-keyword pairs in an auxiliary encrypted dictionary and records the pair to be deleted in a revocation list. Moreover, their scheme shows good scalability when it searches on datasets with billions of document keyword pairs. Hahn et al. Proposed a dynamic SSE construction which uses linked lists to construct the inverted index. However, the client needs to maintain an additional structure called history which stores previous search tokens, and the server has to traverse the encrypted index with time cost linear in the number of document-keyword pairs for a new search. The above dynamic SSE schemes cannot achieve forward privacy, means that the server cannot learn whether an updated data file contains a keyword w that has been searched not in the past. In a recent work by leveraging such update leakage, the authors showed some devastating file injection which can be run on almost all the existing SSE schemes. As a consequence, this work underlines the need for dynamic SSE constructions to provide the forward privacy guarantee. To address this problem, forward privacy was for the first time explicitly considered by Stefanov et al. in . Their main idea is to store document-keyword pairs in a hierarchical structure of logarithmic levels by using techniques in oblivious RAM. Unfortunately, their search time becomes very long with the increasing number of document keyword pairs, meanwhile, the updates require multiple rounds nof communication between the server and the client with non negligible overheads.

**Characteristics and Services Models:**

The salient characteristics of cloud computing based on the definitions provided by the National Institute of Standards and Terminology (NIST) are outlined below:

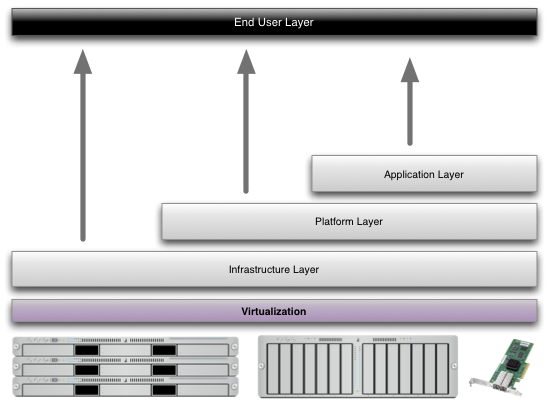
* **On-demand self-service**: A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service’s provider.
* **Broad network access**: Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, laptops, and PDAs).
* **Resource pooling**: The provider’s computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location-independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or data center). Examples of resources include storage, processing, memory, network bandwidth, and virtual machines.
* **Rapid elasticity**: Capabilities can be rapidly and elastically provisioned, in some cases automatically, to quickly scale out and rapidly released to quickly scale in. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be purchased in any quantity at any time.
* **Measured service**: Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be managed, controlled, and reported providing transparency for both the provider and consumer of the utilized service.



Characteristics of cloud computing

**Services Models:**

  Cloud Computing comprises three different service models, namely Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS). The three service models or layer are completed by an end user layer that encapsulates the end user perspective on cloud services. The model is shown in figure below. If a cloud user accesses services on the infrastructure layer, for instance, she can run her own applications on the resources of a cloud infrastructure and remain responsible for the support, maintenance, and security of these applications herself. If she accesses a service on the application layer, these tasks are normally taken care of by the cloud service provider.



Structure of service models

**Benefits of cloud computing:**

1. **Achieve economies of scale** – increase volume output or productivity with fewer people. Your cost per unit, project or product plummets.
2. **Reduce spending on technology infrastructure.** Maintain easy access to your information with minimal upfront spending. Pay as you go (weekly, quarterly or yearly), based on demand.
3. **Globalize your workforce on the cheap.** People worldwide can access the cloud, provided they have an Internet connection.
4. **Streamline processes.** Get more work done in less time with less people.
5. **Reduce capital costs.** There’s no need to spend big money on hardware, software or licensing fees.
6. **Improve accessibility.** You have access anytime, anywhere, making your life so much easier!
7. **Monitor projects more effectively.** Stay within budget and ahead of completion cycle times.
8. **Less personnel training is needed.** It takes fewer people to do more work on a cloud, with a minimal learning curve on hardware and software issues.
9. **Minimize licensing new software.** Stretch and grow without the need to buy expensive software licenses or programs.
10. **Improve flexibility.** You can change direction without serious “people” or “financial” issues at stake.

**Advantages:**

1. **Price:** Pay for only the resources used.
2. **Security**: Cloud instances are isolated in the network from other instances for improved security.
3. **Performance:** Instances can be added instantly for improved performance. Clients have access to the total resources of the Cloud’s core hardware.
4. **Scalability:** Auto-deploy cloud instances when needed.
5. **Uptime:** Uses multiple servers for maximum redundancies. In case of server failure, instances can be automatically created on another server.
6. **Control:** Able to login from any location. Server snapshot and a software library lets you deploy custom instances.
7. **Traffic:** Deals with spike in traffic with quick deployment of additional instances to handle the load.