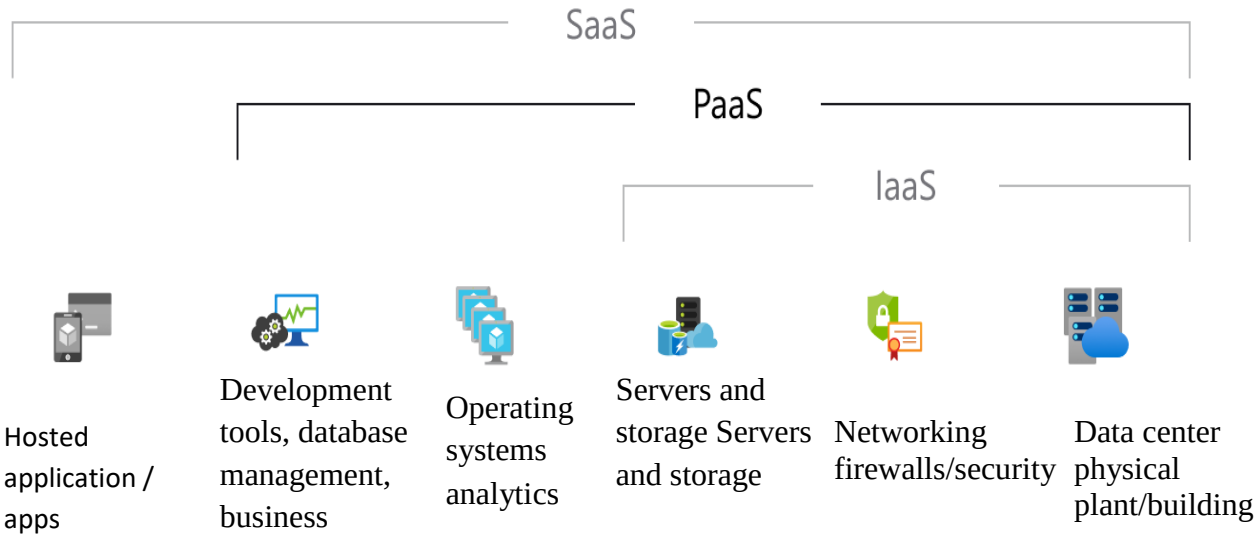
Unit 2: Cloud Computing Architecture

Platform as service

Platform as a service (PaaS) is a complete development and deployment environment in the cloud, with resources that enable you to deliver everything from simple cloud-based apps to sophisticated, cloud-enabled enterprise applications. You purchase the resources you need from a cloud service provider on a pay-as-you-go basis and access them over a secure Internet connection.

Like IaaS, PaaS includes infrastructure—servers, storage, and networking—but also middleware, development tools, business intelligence (BI) services, database management systems, and more. PaaS is designed to support the complete web application lifecycle: building, testing, deploying, managing, and updating.

PaaS allows you to avoid the expense and complexity of buying and managing software licenses, the underlying application infrastructure and middleware, container orchestrators such as Kubernetes, or the development tools and other resources. You manage the applications and services you develop, and the cloud service provider typically manages everything else.



Common PaaS scenarios

Development framework. PaaS provides a framework that developers can build upon to develop or customize cloud-based applications. Similar to the way you create an Excel macro, PaaS lets developers create applications using built-in software components. Cloud features such as scalability, high-availability, and multi-tenant capability are included, reducing the amount of coding that developers must do.

Analytics or business intelligence. Tools provided as a service with PaaS allow organizations to analyze and mine their data, finding insights and patterns and predicting outcomes to improve forecasting, product design decisions, investment returns, and other business decisions.

Additional services. PaaS providers may offer other services that enhance applications, such as workflow, directory, security, and scheduling.

Advantages of PaaS

By delivering infrastructure as a service, PaaS offers the same advantages as IaaS. But its additional features—middleware, development tools, and other business tools—give you more advantages:

Cut coding time. PaaS development tools can cut the time it takes to code new apps with pre-coded application components built into the platform, such as workflow, directory services, security features, search, and so on.

Add development capabilities without adding staff. Platform as Service components can give your development team new capabilities without your needing to add staff having the required skills.

Develop for multiple platforms—including mobile—more easily. Some service providers give you development options for multiple platforms, such as computers, mobile devices, and browsers making cross-platform apps quicker and easier to develop.

Use sophisticated tools affordably. A pay-as-you-go model makes it possible for individuals or organizations to use sophisticated development software and business intelligence and analytics tools that they could not afford to purchase outright.

Support geographically distributed development teams. Because the development environment is accessed over the Internet, development teams can work together on projects even when team members are in remote locations.

Efficiently manage the application lifecycle. PaaS provides all of the capabilities that you need to support the complete web application lifecycle: building, testing, deploying, managing, and updating within the same integrated environment.

Characteristics of PaaS

1. Multi-tenant architecture

A PaaS offering must be multi-tenanted. A multi-tenant platform is one that uses common computing resources including hardware, operating system, software (i.e. application code), and a single underlying database with a shared schema to support multiple customers simultaneously. This is in direct contrast to the traditional client/server architecture, which requires an entire stack of hardware and software to be dedicated to each tenant (customer).

2. Customizable /Programmable User Interface

PaaS offering should provide the ability to construct highly flexible user interfaces via a simple "drag & drop" methodology that permits the creation and configuration of UI components on the fly. Furthermore, given the growing set of Web devices, additional flexibility to use other technologies such as CSS, AJAX and Adobe Flex to specify the appearance of the application's interface should be available to the UI designer.

3. Unlimited Database Customizations

Database used by application should have option of customization for more flexibility in application development. Specifying relationships between objects, a key requirement of any

sophisticated business application, must be possible through the declarative Web-based interface. Other mandatory functions include the ability to incorporate validation rules and permissions at the object/field level and the ability to specify auditing behavior.

4. Automation

PaaS environments automate the process of deploying applications to infrastructure, configuring

application components, provisioning and configuring supporting technology like load balancers and databases, and managing system change based on policies set by the user. While IaaS is known for its ability to shift capital costs to operational costs through outsourcing, only PaaS is able to cut down costs across the development, deployment and management aspects of the application lifecycle.

5. Security

The PaaS offering should provide a flexible access control system that allows detailed control over what users of the SaaS application can see and the data each user can access. Definition of access from the application level (including tabs, menus, objects, views, charts, reports and workflow actions) to the individual field level should be possible. Defining an access control model should be possible through the creation of groups and roles and the assignment of users to either groups or roles.

6. Runtime Framework:

This is the “software stack” aspect of PaaS, and perhaps the aspect that comes first to mind for most people. The PaaS runtime framework executes end-user code according to policies set by the application owner and cloud provider. PaaS runtime frameworks come in many flavors, some based on traditional application runtimes, others based on 4GL and visual programming concepts, and some with pluggable support for multiple application runtimes.

Software as a Service

Software as a service , sometimes referred to as "on-demand software", is a software delivery model in which software and associated data are centrally hosted on the cloud. SaaS is typically accessed by users using a thin client via a web browser. In this model, the software is not hosted on the customers' individual computers. Under the SaaS model, a vendor is responsible for the creation, updating, and maintenance of software. Customers buy a subscription to access it, which includes a separate license, or seat, for each person that will use the software. SaaS has become a common delivery model for many business applications, including accounting, collaboration, customer relationship management (CRM), management information systems(MIS), enterprise resource planning (ERP), invoicing, human resource management (HRM), content management (CM) and service desk management. The emergence of SaaS as an effective software-delivery mechanism creates an opportunity for IT departments to change their focus from deploying and supporting applications to managing the services that those applications provide.

Unlike traditional software which is conventionally sold as a perpetual license with an up-front cost (and an optional ongoing support fee), SaaS providers generally price applications using a subscription fee, most commonly a monthly fee or an annual fee. Consequently, the initial setup cost for SaaS is typically lower than the equivalent enterprise software. SaaS vendors typically price their applications based on some usage parameters, such as the number of users ("seats") using the application. However, because in a SaaS environment customers' data reside with the SaaS vendor, opportunities also exist to charge per transaction, event, or other unit of value.

Benefits of SaaS

- Save money by not having to purchase servers or other software to support use.
- Focus Budgets on competitive advantage rather than infrastructure.
- Monthly obligation rather than up front capital cost.
- Reduced need to predict scale of demand and infrastructure investment up front as available
- capacity matches demand.

- Multi-Tenant efficiency
- Flexibility and scalability
- Security

Characteristics of SaaS

- **Simple and quick system implementation:** As Saas sits on top of PaaS and IaaS, deploying any enterprise level software becomes easy and quick as SaaS inherits all the features of underlying infrastructure. Also since SaaS is scalable, any user request can be addressed with required elasticity.
- **Transparent and low pricing:** With common infrastructure, cloud vendor can leverage their infrastructure with lower cost and transparency. End result of this directly impacts customer cost of software implementation as they get it cheaper with enhanced security and high availability.
- **Multitenant Architecture:** A multitenant architecture, in which all users and applications share a single, common infrastructure and code base that, is centrally maintained. Because SaaS vendor clients are all on the same infrastructure and code base, vendors can innovate more quickly and save the valuable development time previously spent on maintaining numerous versions of outdated code.
- **Easy software maintenance and Customization:** The ability for each user to easily customize applications to fit their business processes without affecting the common infrastructure. Because of the way SaaS is architected, these customizations are unique to each company or user and are always preserved through upgrades. That means SaaS providers can make upgrades more often, with less customer risk and much lower adoption cost.
- **Better Access:** Improved access to data from any networked device while making it easier to manage privileges, monitor data use, and ensure everyone sees the same information at the same time.

Infrastructure as a Service

Infrastructure as a Service (IaaS) is one of the three fundamental service models of cloud computing alongside Platform as a Service (PaaS) and Software as a Service (SaaS). Infrastructure as a Service is a provision model in which an organization outsources the equipment used to support operations, including storage, hardware, servers and networking components. The service provider owns the equipment and is responsible for housing, running and maintaining it. The clienttypically pays on a per-use basis. Clients are able to self-provision this infrastructure, using a Web-based graphical user interface that serves as an IT operations management console for the overall environment. API access to the infrastructure may also be offered as an option.

A typical Infrastructure as a Service offering can deliver the following features and benefits:

- **Scalability;** resource is available as and when the client needs it and, therefore, there are no delays in expanding capacity or the wastage of unused capacity
- **No investment in hardware;** the underlying physical hardware that supports an IaaS service is set up and maintained by the cloud provider, saving the time and cost of doing so on the client side
- **Utility style costing;** the service can be accessed on demand and the client only pays for the resource that they actually use
- **Location independence;** the service can usually be accessed from any location as long as there is an internet connection and the security protocol of the cloud allows it
- **Physical security of data center locations;** services available through a public cloud, or private clouds hosted externally with the cloud provider, benefit from the physical security afforded to the servers which are hosted within a data center
- **No single point of failure;** if one server or network switch, for example, were to fail, the broader service would be unaffected due to the remaining multitude of hardware resources and redundancy configurations. For many services if one entire data center were to go offline, never mind one server, the IaaS service could still run successfully.

On Demand Computing: On-demand (OD) computing is an increasingly popular enterprise model in which computing resources are made available to the user as needed. The resources may be maintained within the user's enterprise, or made available by a service provider. The on-demand model evolved to overcome the challenge of being able to meet fluctuating resource demands efficiently. Because demand for computing resources can vary drastically from one time to another, maintaining sufficient resources to meet peak requirements can be costly.

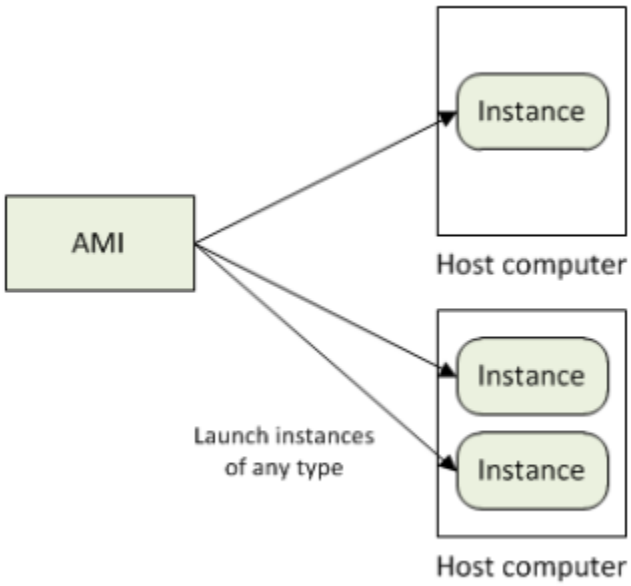
On-demand computing products are rapidly becoming prevalent in the marketplace. Computer Associates, HP, IBM, Microsoft, and Sun Microsystems are among the more prominent on-demand vendors. These companies refer to their on-demand products and services by a variety of names. Concepts such as grid computing, utility computing, autonomic computing, and adaptive management seem very similar to the concept of on-demand computing. The major advantage of On Demand Computing (ODC) is low initial cost, as computational resources are essentially rented when they are required. This provides cost savings over purchasing them outright.

(Amazon EC2): Amazon Elastic Compute Cloud (EC2) is a central part of Amazon cloud Computing platform Amazon Web Services (AWS). EC2 allows users to rent virtual computers on which to run their own computer applications. EC2 allows scalable deployment of applications by providing a Web service through which a user can boot an Amazon Machine Image to create a virtual machine, which Amazon calls an "instance", containing any software desired. A user can create, launch, and terminate server instances as needed, paying by the hour for active servers, hence the term "elastic".

It reduces the time required to obtain and boot new server instances to minutes, allowing customers to quickly scale capacity as their computing demands dictate. It changes the economics of computing by allowing clients to pay only for capacity they actually use. It

provides developers the tools needed to build failure-resilient applications and isolate themselves from common failure scenarios.

To use the EC2, a subscriber creates an Amazon Machine Image (AMI) containing the operating system, application programs and configuration settings. Then the AMI is uploaded to the Amazon Simple Storage Service (Amazon S3) and registered with Amazon EC2, creating a so-called AMI identifier (AMI ID). Once this has been done, the subscriber can requisition virtual machines on an as-needed basis. Capacity can be increased or decreased in real time from as few as one to more than 1000 virtual machines simultaneously. Billing takes place according to the computing and network resources consumed.



The idea of EC2 is to lighten the cost of buying servers to host a system, but more importantly to eliminate the wasted time systems engineers devote to managing hard assets. Instead of buying servers to increase capacity or add new features, you simply buy more gigabytes on EC2. Amazon sells it in subscription form, with subscriptions based on how much capacity you use.

Characteristics of Amazon EC2

a. Persistent Storage

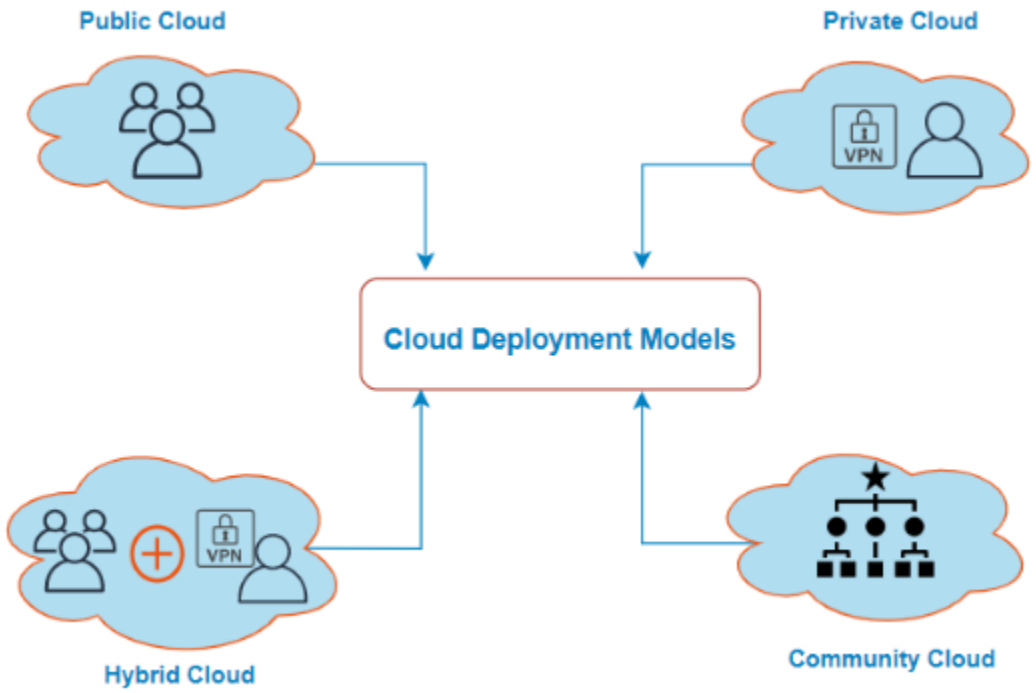
b. Elastic IP Addresses

c. Amazon Cloud Watch

d. Automated Scaling

e. Reliability

Cloud Computing Deployment Model



Public Cloud: This is the deployment model that most commonly described as cloud computing. In this model, all of the physical resources are owned and operated by a third party cloud computing provider. The provider services multiple clients that may consist of

individuals or corporations utilizing these resources through the public Internet. Services can be dynamically provisioned and are billed based on usage alone. This model provides the highest degree of cost savings while requiring the least amount of overhead. This model is best suited for business requirements wherein it is required to manage load spikes, host SaaS applications, utilize interim infrastructure for developing and testing applications, and manage applications which are consumed by many users that would otherwise require large investment in infrastructure from businesses.

Private Cloud: Private cloud computing systems emulate public cloud service offerings

within an organization's boundaries to make services accessible for one designated organization. Private cloud computing systems make use of virtualization solutions and focus on consolidating distributed IT services often within data centers belonging to the company. The chief advantage of these systems is that the enterprise retains full control over corporate data, security guidelines, and system performance. This model doesn't bring much in terms of cost efficiency: it is comparable to buying, building and managing your own infrastructure. Still, it brings in tremendous value from a security point of view. In addition to security reasons, this model is adopted by organizations in cases where data or applications are required to conform to various regulatory standards, which may require data to be managed for privacy and audits that govern the corporation.

Hybrid Cloud: This can be a combination of private and public clouds that support the requirement to retain some data in an organization, and also the need to offer services in the cloud. A company may use internal resources in a private cloud maintain total control over its proprietary data. It can then use a public cloud storage provider for backing up less sensitive information. At the same time, it might share computing resources with other organizations that have similar needs. By combining the advantages of the other models, the hybrid model offers organizations the most flexibility. This model is also used for handling cloud bursting, which refers to a scenario where the existing private cloud infrastructure is not able to handle load spikes and requires a fallback option to support the load. Hence, the cloud migrates workloads between public and private hosting without any inconvenience to the users.

Community Cloud: In the community deployment model, the cloud infrastructure is shared by several organizations with the same policy and compliance considerations. This helps to further reduce costs as compared to a private cloud, as it is shared by larger group.

A community cloud contains features of the public and private cloud models. Like a public cloud, the community cloud may contain software, data storage, and computing resources that are utilized by multiple organizations. Where this model differs from the public model is that the infrastructure is only utilized by a group of organizations that are known to each other. Similarly to a private cloud, these organizations are responsible for the operation of their own infrastructure. The community cloud model can provide greater cost savings than the private cloud while offering some of its security features. This model is best suited for organizations that share common requirements such as security or legal compliance policies. It can be managed by the member organizations or by a third party provider.

Cloud design and implementation using SOA

Service Oriented Architecture (SOA) is an architectural style and a combination of methodologies that aims to achieve interoperability of remotely or locally located homogeneous and heterogeneous applications by utilizing reusable service logic. Service orientation shows variation in adopting technology for implementation, rather than focusing on the technology itself, as SOA considers the description of the problem domain before concentrating on the usage of a specific execution environment. Although SOA does not a direct implication of a certain technology and has been applied to software development before the invention of Web services, the capable architectures that realize the SOA vision in a more applicable way are built with Web service technologies. Driven by these competent technologies, the enterprise is practicing open standards for communication over network, messaging and description of service interfaces. SOA with Web services are at the present gaining momentum, as with Web services there is fundamental improvement in SOA based application development. It is required to follow new approaches and particular methodologies when building service based application structure, rather than tracking the traditional approaches to software development. SOA needs unique development strategies, which

replace the conventional approaches to building software architectures and promise the development of plug-and-play application structures and building modules capable of expressing definite business functionalities and problem domains.

SOA Components

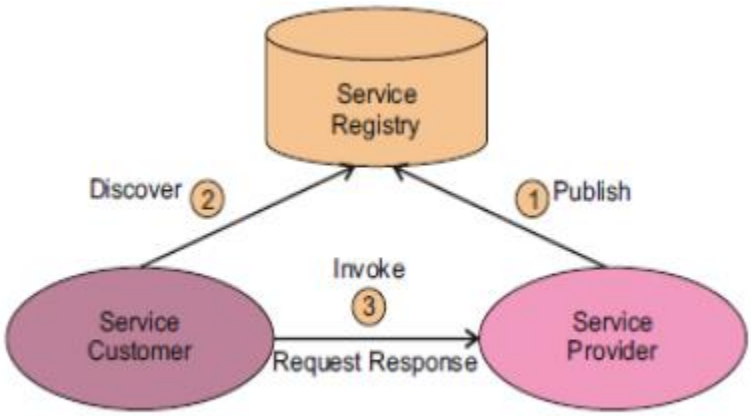


Figure 1 Display the SOA component

SOA provides a strong architectural discipline and focus area centered on service creation, modeling and development, formation of process information, and service oriented integration approaches and strategies. Services are the building blocks of SOA and new applications can be constructed through consuming these services and orchestrating them within a business process. Services are reusable units for articulating common business and technology functionalities. To implement a successful SOA in enterprise requires consideration of various concepts and implementation strategies, which formulate the essential characteristics of service oriented enterprise. A complete SOA implementation reflects on not only the deployment of services, but also the possibility of using them to integrate diverse application logics, and building of composite applications.

Characteristics of SOA

- In SOA, Services should be **independent** of other services. Altering a service should not affect calling service.

- Services should be **self-contained**. When we talk about a Register Customer service it means, service will do all the necessary work for us, we are not required to care about anything.
- Services should be able to **define themselves**. Services should be able to answer a question what is does? It should be able to tell client what all operations it does, what all data types it uses and what kind of responses it will return.
- Services should be **published** into a location (directory) where anyone can search for it.
- As said, SOA comprises of collection services which communicate via **standard Messages**.
- Standard messages make them platform independent. (Here standard doesn't mean standard across Microsoft it means across all programming languages and technologies.)
- Services should be able to communicate with each other **asynchronously**.
- Services should support **reliable messaging** Means there should be a guarantee that request will be reached to correct destination and correct response will be obtained.
- Services should support **secure communication**.

Privacy

The concept of privacy is not universally defined, since it is a cultural construct and, hence, subjective. It refer to informational privacy , which is related to the person's right to determine when, how and to what extent information about him or her is communicated to others The privacy requirements related to cloud computing are closely related to the set of applicable legislation and regulations . In Europe, they include a series of European Commission directives. Those directives regulate the processing of personal data, protection of privacy and data retention in the EU. In addition, a comprehensive reform of the data protection rules was recently proposed in The reform includes a single set rules on data protection to be enforced throughout the EU, including users' right to access their own data, to transfer personal data from one service to another and the "right to be forgotten". The new directive and regulation will result, if approved, in a new set of privacy requirements that the telecommunication providers will need to comply with. The principle of necessity of data collection and processing, which is one of the most essential privacy requirements, determines that the collection and processing of personal data should only be allowed if it is

necessary for the tasks falling within the responsibility of the data processing agency. Hence, personal information should ideally not be collected or used for identification purposes when not absolutely necessary. The best strategy to enforce such a requirement is the avoidance or minimization of personal data. Thus, privacy is best protected if no personal identifiable information (PII) is stored or processed in the cloud computing platform or transferred to or from it. Hence, the telecommunication providers must consider the implementation of privacy enhancing technologies (PETs) for the anonymization or pseudonymization of data. It is, nevertheless, expected that a significant share of data being transmitted, processed or stored on the cloud computing platform will have PII associated to it. Therefore, other PETs, such as those based on obfuscation or on the concept of privacy by design , in which privacy is integrated into all phases of the design and implementation of the cloud computing platform, are required to embed privacy protections into the system. The implementation of cloud marketplaces and their use by telecommunication providers must also take privacy aspects into consideration, as the EU legislation obliges the data controller to have knowledge about which entity processes which PII and the purpose of such processing. Hence, a telecommunication provider must necessarily have knowledge about all cloud computing providers that process PII of its customers.

Security

Cryptographic techniques are essential to provide information separation and data confidentiality in cloud computing platforms. At the same time, such techniques greatly reduce the ability to provide simple caching and filtering/screening services that increase efficiency and robustness against unwanted traffic. Cryptographic techniques, such as holomorphic encryption, may solve, in principle, some problems but it is still rather unexplored and untested in practice. Cryptography separation is realized with strong compartmentalization techniques provided in secure operational systems, virtualization or hardware. Virtualization using hypervisors is a technique that is available in open-source solutions, such as Xen and KVM (Kernel-based Virtual Machine) . Moreover, access control mechanisms can also be used to protect information. Despite the huge success of cloud computing as a computing model and as an economic platform, many enterprises still hesitate to join the cloud in full because of security concerns. Many major organizations only allow

their employees to use cloud services for less sensitive data. Other organizations use private cloud computing platforms to overcome some of these problems. This hesitation comes from concerns about the security of cloud solutions as well as loss of control of data and computation. Such a situation creates a clear advantage for the telecommunications industry, as it can be the enabler of the security services. For instance, the problem of cross-authenticating customers is a key problem in cloud computing platforms, and telecommunication providers are already able to provide strong and practical solutions to it. Thus, the telecommunication providers can potentially take a great share of the business role of cloud computing providers. Security requirements in cloud computing platforms are a well-documented subject. Standard security concerns are, for example, vulnerabilities that appear in hypervisors or other virtualization technologies that are used by cloud computing vendors; network attacks on the user's connection to the cloud providers; weaknesses in the user authentication process; and availability issues, including denial of services (DoS) attacks. Another key problem is the loss of control of data. From a security perspective, this has to do with confidentiality and data integrity issues. The cloud computing provider may take part of a customer's PII directly, or may use it indirectly (for example, the cloud computing provider may provide statistical analysis of the data for advertising purposes). This problem has accelerated the search for technical solutions to provide confidentiality and integrity in cloud computing. Cryptographic techniques can be applied, but have negative practical implications regarding computational performance, as discussed in. In cloud storage, operations on the data are commonplace, such as searching for keywords. Decrypting an entire data set when searching for a single keyword is of course not an acceptable solution. Searchable Encryption algorithms offer the possibility of searching over encrypted data. In cloud computing platforms, it might be possible to perform computation on encrypted data without decrypting it beforehand by using holomorphic encryption techniques. However, current techniques for holomorphic encryption may not be practical enough yet for deployment in cloud computing platforms. Inter virtual machine attacks need to be evaluated when considering virtualization. Methods based on Chinese-wall techniques mitigate the risks of such attacks. VM image protection is also an important aspect when considering secure procedures for migration and backup.

Trust

The cloud computing ecosystem consists of several entities, e.g., cloud providers, resellers, carriers and customers. This ecosystem is highly distributed and customers possess only a limited and abstract view of it. This situation is rather similar to what customers face with telecommunication networks in which base stations are co-owned by different telecommunication providers. Again, customers possess only a limited view of the telecommunication infrastructure. Introducing cloud computing technologies on top of telecommunication networks pushes such abstraction even further, and with it the trust that customers invest in their telecommunication providers. The abstract and distributed nature of cloud computing environments represents a considerable obstacle for the acceptance and market success of cloud based services. Recent surveys have demonstrated that cloud consumers are concerned about their outsourced data and the related services provided by cloud providers. Such concerns can be mitigated by using preventive controls, e.g., encryption and authentication. These preventive controls would permit users to establish trust with cloud providers from a technical point of view, which is known as a "hard trust" mechanism. "Hard trust" mechanisms make assumptions about an information system's security, dependability and reliability based on existence of different primitives, e.g., certificates, audits and secure hypervisors. However, "hard trust" mechanisms are not necessarily sufficient for cloud consumers to trust cloud providers. Trust involves aspects such as intrinsic human emotions and perceptions, interaction and exchange of experiences, and loyalty to a brand. Such aspects are fundamental when establishing trust with cloud providers. This kind of trust is known as "soft trust" . "Soft trust" assumes that no "hard trust" mechanisms are perfect and errors exist no matter how rigid the design procedures are. "Soft trust" mechanisms can be complemented with "hard trust" to support telecommunication providers in establishing trustworthy cloud computing environments. Moreover, trust mechanisms require knowledge of the architecture of the system and the trustworthiness of its subsystems and components when evaluating trustworthiness of cloud providers