Chapter - 1
cloud architecture

Prepared by: Sushil Sah

Cloud Computing Architecture

- The term cloud computing is a wide umbrella encompassing many different things; lately it has become a buzzword that is easily misused to refurbish existing technologies and ideas for the public.
- Cloud computing is a utility-oriented and Internet-centric way of delivering IT services on demand. These services cover the entire computing stack: from the hardware infrastructure packaged as a set of virtual machines to software services such as development platforms and distributed applications.
- The cloud computing paradigm emerged as a result of the convergence of various existing models, technologies, and concepts that changed the way we deliver and use IT services.

The cloud reference model

- Cloud computing supports any IT service that can be consumed as a utility and delivered through a network, most likely the Internet.
- Cloud computing ‘s such characterization includes quite different aspects:
  - infrastructure,
  - development platforms,
  - application and services.
- It is possible to organize all the concrete realizations of cloud computing into a layered view covering the entire stack from hardware appliances to software systems as architectural view.

Architecture

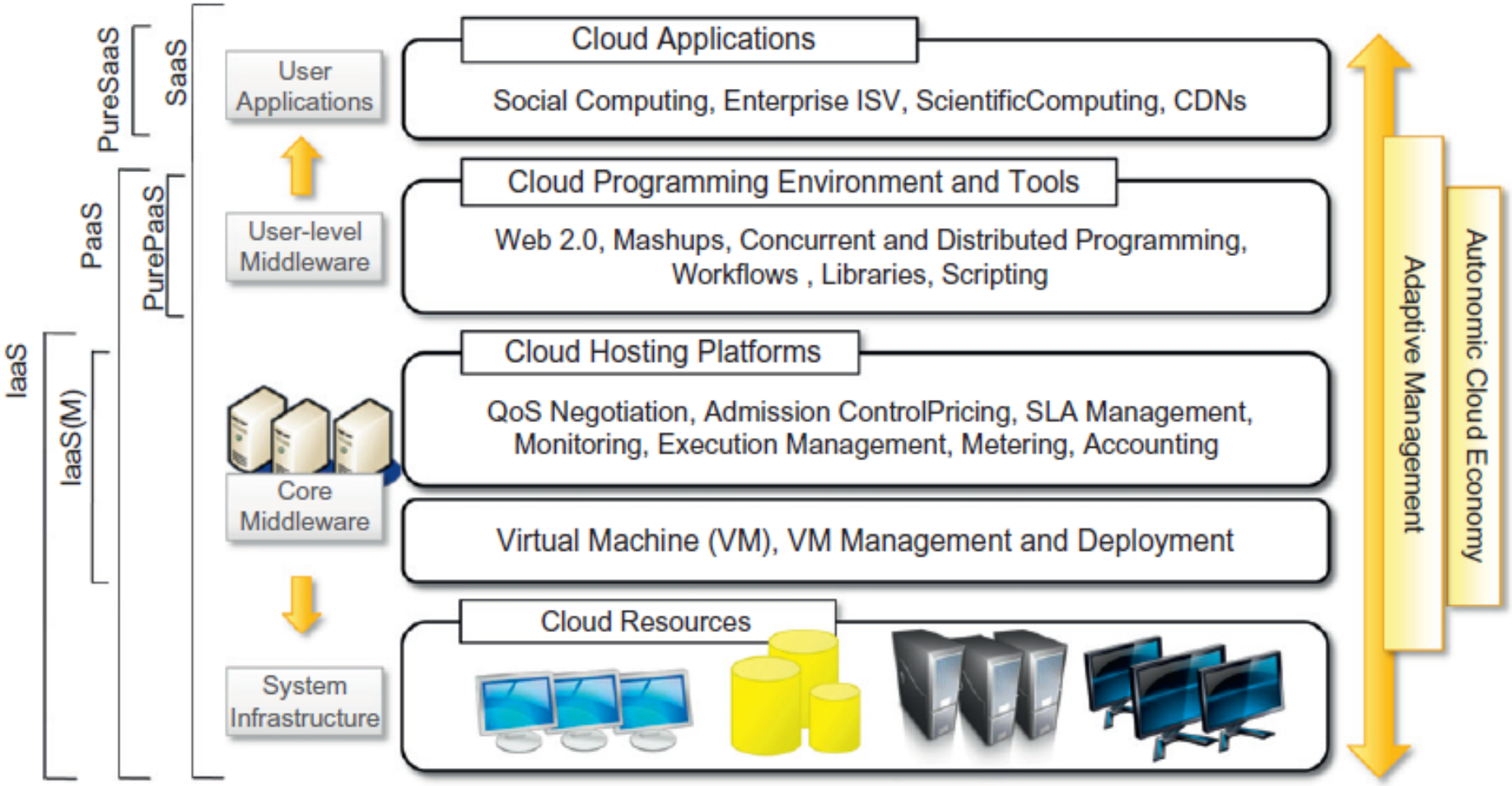


fig: cloud computing architecture

Architecture

- The physical infrastructure is managed by the core middleware, the objectives of which are to provide an appropriate runtime environment for applications and to best utilize resources.
- At the bottom of the stack, virtualization technologies are used to guarantee runtime environment customization, application isolation, sandboxing, and quality of service.
- Hypervisors manage the pool of resources and expose the distributed infrastructure as a collection of virtual machines. By using VT, CPU and memory and to virtualize specific devices, thus meeting the requirements of users and applications.

Architecture

- This solution is generally paired with storage and network virtualization strategies, which allow the infrastructure to be completely virtualized and controlled.
- The combination of cloud hosting platforms and resources is generally classified as a Infrastructure-as-a-Service(IaaS) solution.
- IaaS solutions are suitable for designing the system infrastructure but provide limited services to build applications.
- The top layer of the reference model contains services delivered at the application level, are mostly referred to as Software-as-a-Service(SaaS).
- The horsepower of the cloud provided by IaaS and PaaS solutions allows software vendors to deliver their application services over the Internet.

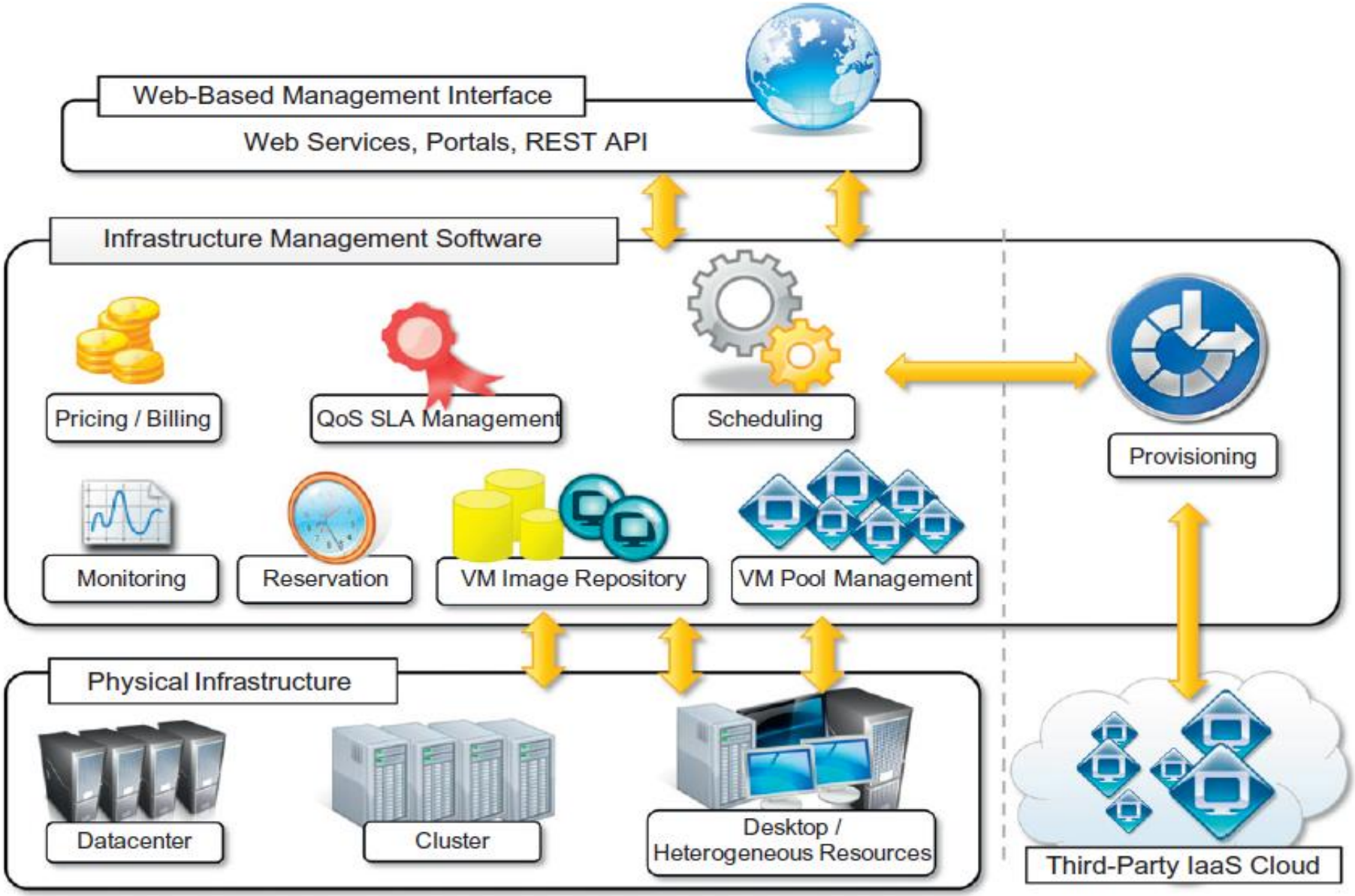
Architecture

- The reference model described here also introduces the concept of everything as a Service (XaaS).
- This is one of the most important elements of cloud computing: Cloud services from different providers can be combined to provide a completely integrated solution covering all the computing stack of a system.
- Three major categories used to classify cloud computing solutions.
  - Infrastructure- and hardware-as-a-service
  - Platform as a service
  - Software as a service

Infrastructure- and hardware-as-a-service

- Infrastructure- and Hardware-as-a-Service (IaaS/HaaS) solutions are the most popular and developed market segment of cloud computing.
- Deliver customizable infrastructure on demand. The available options within the IaaS offering umbrella range from single servers to entire infrastructures, including network devices, load balancers, and database and Web servers.
- IaaS/HaaS solutions bring all the benefits of hardware virtualization: workload partitioning, application isolation, sandboxing, and hard- ware tuning.

IaaS reference implementation



IaaS reference implementation

- SLA resource-based allocation, workload management, support for infrastructure design through advanced Web inter- faces, and the ability to integrate third party IaaS solutions.
- Three principal layers forming an Infrastructure-as-a-Service solution:
  - the physical infrastructure,
  - the software management infrastructure,
  - the user interface.
- In particular, management of the virtual machines is the most important function performed by this layer.
- A central role is played by the scheduler, which is in charge of allocating the execution of virtual machine instances. The scheduler interacts with the other components that perform a variety of tasks.

Scheduler tasks

- The pricing and billing component takes care of the cost of executing each virtual machine instance and maintains data that will be used to charge the user.
- The monitoring component tracks the execution of each virtual machine instance and maintains data required for reporting and analyzing the performance of the system.
- The reservation component stores the information of all the virtual machine instances that have been executed.
- If support for QoS-based execution is provided, a QoS/SLA management component will maintain a repository of all the SLAs made with the users.
- The VM repository component provides a catalog of virtual machine images that users can use to create virtual instances.

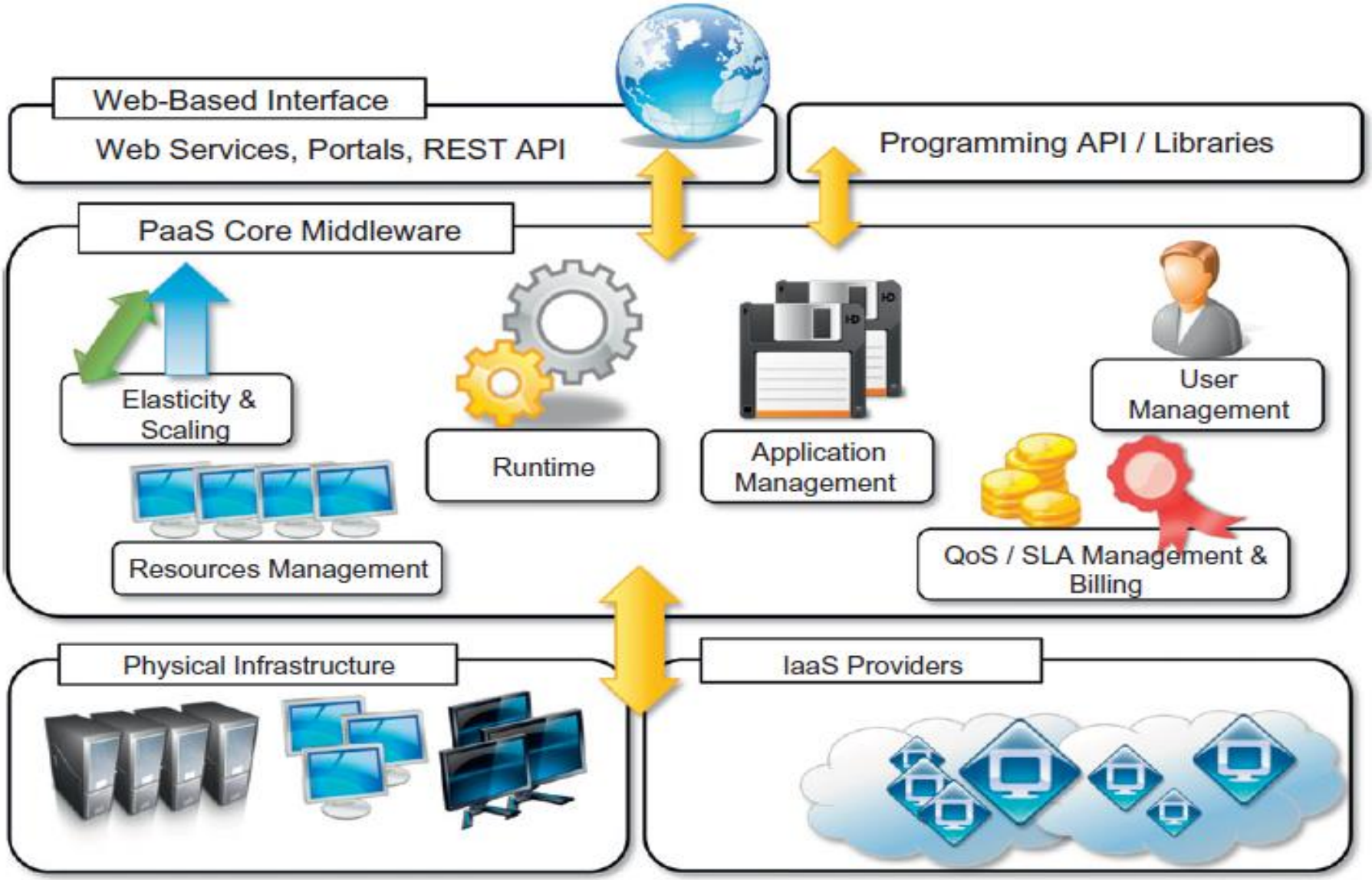
Scheduler tasks

- A **VM** pool manager component is responsible for keeping track of all the live instances.
- Finally, if the system supports the integration of additional resources belonging to a third-party IaaS provider, a provisioning component interacts with the scheduler to provide a virtual machine instance that is external to the local physical infrastructure directly managed by the pool.

Platform as a service

- Platform-as-a-Service (PaaS) solutions provide a development and deployment platform for running applications in the cloud.
- PaaS constitute the middleware on top of which applications are built. **Application management** is the core functionality of the middleware.
- PaaS implementations provide applications with a runtime environment and do not expose any service for managing the underlying infrastructure.
- Application management automate the process of deploying applications to the infrastructure, configuring application components, provisioning and configuring supporting technologies such as load balancers and databases, and managing system change based on policies set by the user.

Platform as a service



Platform as a service

- Developers design their systems in terms of applications and are not concerned with hardware (physical or virtual), operating systems, and other low-level services.
- The core middleware is in charge of managing the resources and scaling applications on demand or automatically, according to the commitments made with users.
- From a user point of view, the core middleware exposes interfaces that allow programming and deploying applications on the cloud. These can be in the form of a Web-based interface or in the form of programming APIs and libraries.

Platform as a service

- PaaS solutions can offer middleware for developing applications together with the infrastructure or simply provide users with the software that is installed on the user premises.
- In the first case, the PaaS provider also owns large datacenters where applications are executed; in the second case, PaaS, the middleware constitutes the core value of the offering.
- It is also possible to have vendors that deliver both middleware and infrastructure and ship only the middleware for private installations.

Platform as a service

Table 4.2 Platform-as-a-Service Offering Classification

Category	Description	Product Type	Vendors and Products
PaaS-I	Runtime environment with Web-hosted application development platform. Rapid application prototyping.	Middleware + Infrastructure Middleware + Infrastructure	Force.com Longjump
PaaS-II	Runtime environment for scaling Web applications. The runtime could be enhanced by additional components that provide scaling capabilities.	Middleware + Infrastructure Middleware Middleware + Infrastructure Middleware + Infrastructure Middleware + Infrastructure Middleware	Google AppEngine AppScale Heroku Engine Yard Joyent Smart Platform GigaSpaces XAP
PaaS-III	Middleware and programming model for developing distributed applications in the cloud.	Middleware + Infrastructure Middleware Middleware Middleware Middleware Middleware	Microsoft Azure DataSynapse Cloud IQ Manjrasof Aneka Apprenda SaaSGrid GigaSpaces DataGrid

Platform as a service

- The PaaS umbrella encompasses a variety of solutions for developing and hosting applications in the cloud. some essential characteristics that identify a PaaS solution:
  - Runtimeframework - This framework represents the “softwarestack” of the PaaS model and the most intuitive aspect that comes to people’s minds when they refer to PaaS solutions.
  - Abstraction - PaaS solutions are distinguished by the higher level of abstraction that they provide. This means that PaaS solutions offer a way to deploy and manage applications on the cloud.
  - Automation – PaaS environments automate the process of deploying applications to the infrastructure, scaling them by provisioning additional resources when needed. This process is performed automatically and according to the SLA made between the customers and the provider.
  - Cloud services. PaaS offerings provide developers and architects with services and APIs, helping them to simplify the creation and delivery of elastic and highly available cloud applications.

Software as a service

- SaaS is a software delivery model that provides access to applications through the Internet as a Web-based service.
- Free the users from complex hardware and software management by offloading such tasks to third parties, which build applications accessible to multiple users through a Web browser.
- It is also possible to have vendors that deliver both middleware and infrastructure and ship only the middleware for private installations.
- In this scenario, customers neither need install anything on their premises nor have to pay considerable up-front costs to purchase the software and the required licenses.

Software as a service

- The SaaS model is appealing for applications serving a wide range of users and that can be adapted to specific needs with little further customization.
- Every enterprise will have the same requirements for the basic features concerning CRM and ERP; different needs can be satisfied with further customization.
- This scenario facilitates the development of software platforms that provide a general set of features and support specialization and ease of integration of new components.
- In this scenario, customers neither need install anything on their premises nor have to pay considerable up-front costs to purchase the software and the required licenses.

Software as a service

- Multitenancy – which is a feature of SaaS compared to traditional packaged software, allows providers to centralize and sustain the effort of managing large hardware infrastructures, maintaining and upgrading applications transparently to the users, and optimizing resources by sharing the costs among the large user base.
- **Core characteristics of SaaS:**
- The product sold to customer is application access.
- The application is centrally managed.
- The service delivered is one-to-many.
- The service delivered is an integrated solution delivered on the contract, which means provided as promised.

SaaS benefit

- Software cost reduction and total cost of ownership (TCO) were paramount
- Service-level improvements
- Rapid implementation
- Standalone and configurable applications
- Rudimentary application and data integration
- Subscription and pay-as-you-go (PAYG) pricing

- Thus, SaaS application providing users with a Web space and profile that can be enriched and extended with third-party applications such as office automation, integration with CRM-based solutions, social Websites, and photo editing.

Types of clouds

- Four different types of cloud:
- Public clouds- The cloud is open to the wider public.
- Private clouds – The cloud is implemented with in the private premises of an institution and generally made accessible to the members of the institution or a subset of them.
- Hybrid or heterogeneous clouds- The cloud is a combination of the two previous solutions and most likely identifies a private cloud that has been augmented with resources or services hosted in a public cloud.
- Community clouds- The cloud is characterized by a multi-administrative domain involving different deployment models(public, private, and hybrid),and it is specifically designed to address the needs of a specific industry.

Public clouds

- Public clouds constitute the first expression of cloud computing in which the services offered are made available to anyone, from anywhere, and at any time through the Internet.
- Public clouds are a distributed system, most likely composed of one or more datacenters connected together, on top of which the specific services offered by the cloud are implemented.
- A fundamental characteristic of public clouds is multitenancy, means to serve a multitude of users, not a single customer. Any customer requires a virtual computing environment that is separated, and most likely isolated, from other users.

Public clouds

- **QoS** management is a very important aspect of public clouds where significant portion of the software infrastructure is devoted to monitoring the cloud resources, to bill them according to the contract made with the user, and to keep a complete history of cloud usage for each customer.
- A public cloud can offer any kind of service: infrastructure, platform, or applications. For example, **Amazon EC2** is a public cloud that provides infrastructure as a service; **Google AppEngine** is a public cloud that provides an application development platform as a service; and **SalesForce.com** is a public cloud that provides software as a service.

Private clouds

- Public clouds are not applicable in all scenarios. In the case of public clouds, the provider is in control of the infrastructure and, eventually, of the customers' core logic and sensitive data.
- Even though there could be regulatory procedure in place that guarantees fair manage- ment and respect of the customer's privacy, this condition can still be perceived as a threat or as an unacceptable risk that some organizations are not willing to take.
- In particular, institutions such as government and military agencies will not consider public clouds as an option for processing or storing their sensitive data..

Private clouds

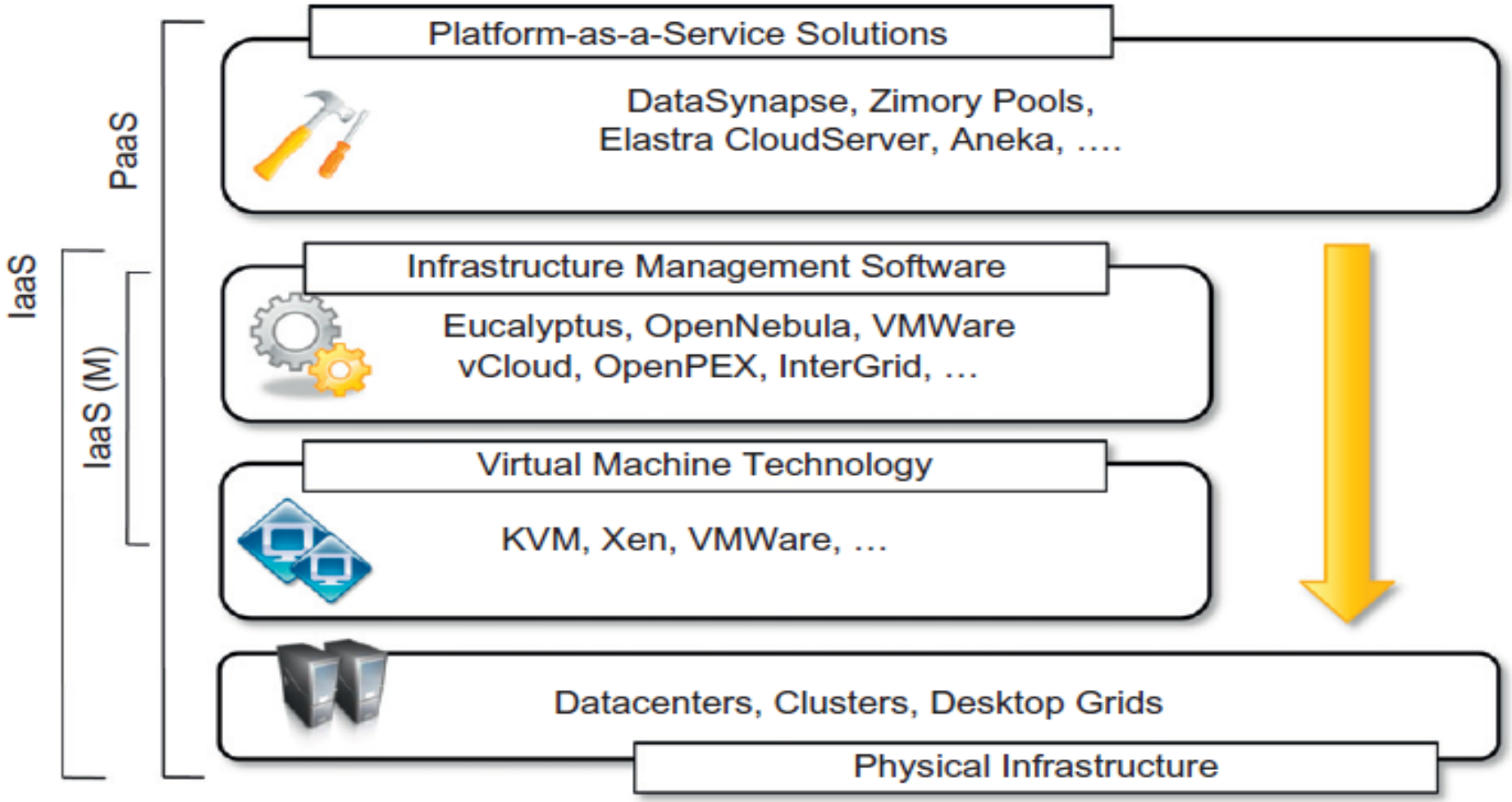
- Private clouds are virtual distributed systems that rely on a private infrastructure and provide internal users with dynamic provisioning of computing resources. Instead of a pay-as-you-go model as in public clouds taking into account the usage of the cloud and proportionally billing the different departments or sections of an enterprise.
- Private clouds have the advantage of keeping the core business operations in-house by relying on the existing IT infrastructure and reducing the burden of maintaining it once the cloud has been set up.
- In this scenario, security concerns are less critical, since sensitive information does not flow out of the private infrastructure. Moreover, existing IT resources can be better utilized because the private cloud can provide services to a different range of users.

Advantages of Private clouds

- Customer information protection - espite assurances by the public cloud leaders about security, In-house security is easier to maintain and rely on.
- Infrastructure ensuring SLAs - Quality of service implies specific operations such as appropriate clustering and failover, data replication services can be commensurate to the application needs.
- Compliance with standard procedures and operations - If organizations are subject to third-party compliance standards, specific procedures have to be put in place when deploying and executing applications.
- Private clouds can provide in-house solutions for cloud computing, but if compared to public clouds they exhibit more limited capability to scale elastically on demand.

Private clouds

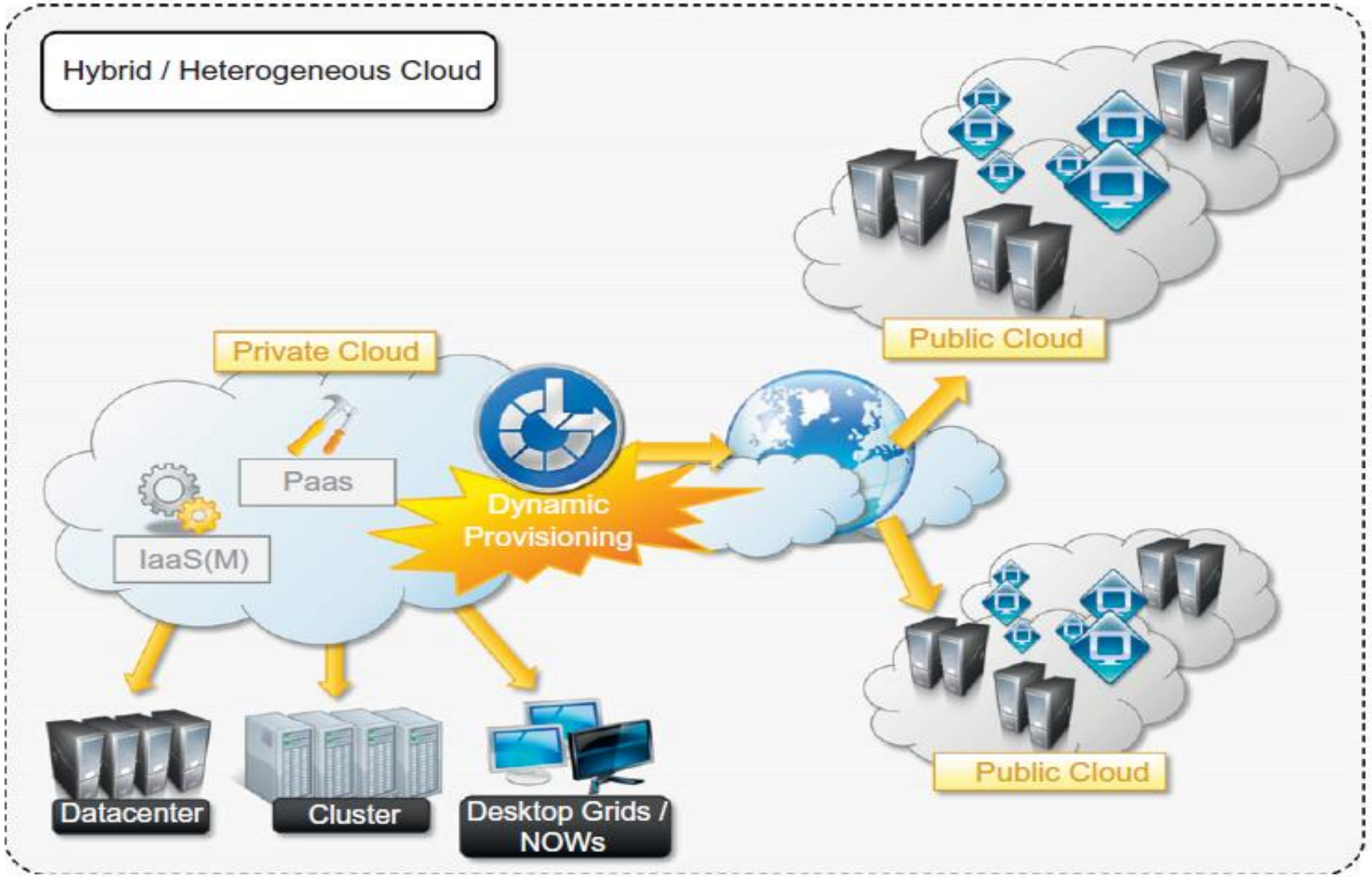
- From an architectural point of view, private clouds can be implemented on more heterogeneous hardware.



Hybrid clouds

- Hybrid clouds allow enterprises to exploit existing IT infrastructures, maintain sensitive information within the premises, and naturally grow and shrink by provisioning external resources and releasing them when they're no longer needed.
- Security concerns are then only limited to the public portion of the cloud that can be used to perform operations with less stringent constraints but that are still part of the system workload.
- Hybrid cloud is a heterogeneous distributed system resulting from a private cloud that integrates additional services or resources from one or more public clouds. For this reason they are also called heterogeneous clouds.

Hybrid clouds



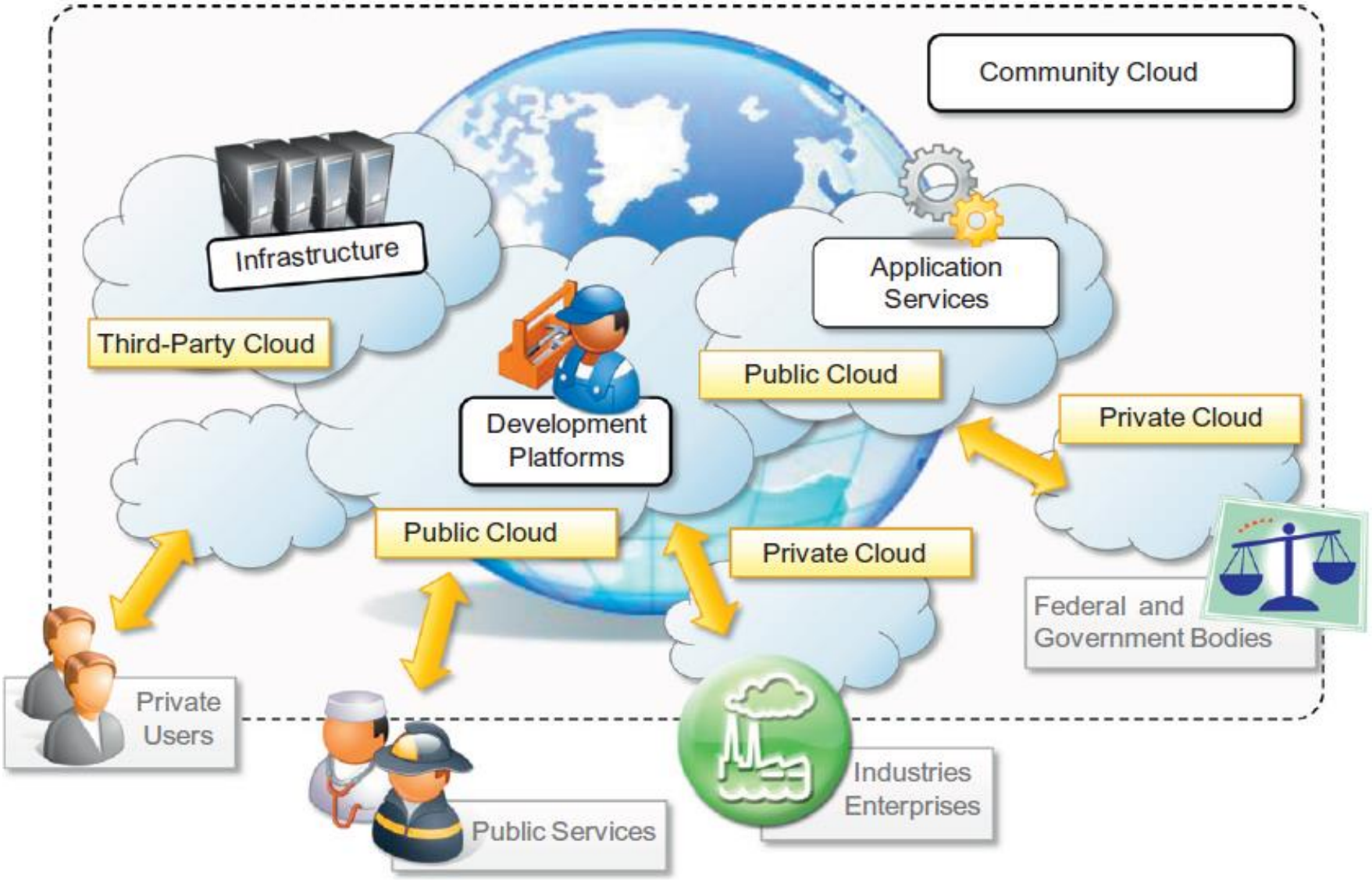
Hybrid clouds

- Dynamic provisioning is a fundamental component in hybrid clouds, address scalability issues by leveraging external resources for exceeding capacity demand.
- These resources or services are temporarily leased for the time required and then released. This practice is also known as cloudbursting.
- Dynamic provisioning is most commonly implemented in PaaS solutions that support hybrid clouds.
- **Aneka** provides a provisioning service that leverages different IaaS providers for scaling the existing cloud infrastructure. The provisioning service cooperates with the scheduler, which is in charge of guaranteeing a specific QoS for applications.

Community clouds

- Community clouds are distributed systems created by integrating the services of different clouds to address the specific needs of an industry, a community, or a business sector.
- According to NIST, *“The infrastructure is shared by several organizations and supports a specific community that has shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be managed by the organizations or a third party and may exist on premise or off premise. “*
- The users of a specific community cloud fall into a well-identified community, sharing the same concerns or needs; they can be government bodies, industries, or even simple users, but all of them focus on the same issues for their interaction with the cloud.

Community clouds



Sectors for community clouds

- Media industry – In the media industry, companies are looking for low-cost, agile, and simple solutions to improve the efficiency of content production..
- Health care industry - In the health care industry, there are different scenarios in which community clouds could be of use.
- Energy and other core industries - In these sectors, community clouds can bundle the comprehensive set of solutions that together vertically address management, deployment, and orchestration of services and operations.
- Public sector - Legal and political restrictions in the public sector can limit the adoption of public cloud offerings.
- Scientific research - Science clouds are an interesting example of communityclouds.

Private clouds

- **QoS** management is a very important aspect of public clouds where significant portion of the software infrastructure is devoted to monitoring the cloud resources, to bill them according to the contract made with the user, and to keep a complete history of cloud usage for each customer.
- A public cloud can offer any kind of service: infrastructure, platform, or applications. For example, **Amazon EC2** is a public cloud that provides infrastructure as a service; **Google AppEngine** is a public cloud that provides an application development platform as a service; and **SalesForce.com** is a public cloud that provides software as a service.

Benefits of community clouds

- Openness - By removing the dependency on cloud vendors, community clouds are open systems in which fair competition between different solutions can happen.
- Community - Being based on a collective that provides resources and services, the infrastructure turns out to be more scalable because the system can grow simply by expanding its user base.
- Graceful failures - Since there is no single provider or vendor in control of the infrastructure, there is no single point of failure.
- Convenience and control - Within a community cloud there is no conflict between convenience and control because the cloud is shared and owned by the community.
- Environmental sustainability - The community cloud is supposed to have a smaller carbon footprint because it harnesses underutilized resources.

cloud interoperability and standards

- Cloud computing is a service-based model for delivering IT infrastructure and applications like utilities such as power, water, and electricity.
- To fully realize this goal, introducing standards and allowing interoperability between solutions offered by different vendors are objectives of fundamental importance.
- Vendor lock-in can prevent a customer from switching to another competitor's solution, or when this is possible, it happens at considerable conversion cost and requires significant amounts of time.
- This can occur either because the customer wants to find a more suitable solution for customer needs or because the ven- dor is no longer able to provide the required service.

cloud interoperability and standards

- The presence of standards that are actually implemented and adopted in the cloud computing community could give room for interoperability and then lessen the risks resulting from vendor lock-in.
- The challenge is providing standards for supporting the migration of running instances, thus allowing the real ability of switching from one infrastructure vendor to another in a completely transparent manner.
- Another direction in which standards try to move is devising a general reference architecture for cloud computing systems and providing a standard interface through which one can interact with them.
- There is no consistent trend in devising some common APIs for interfacing with IaaS (and, in general, XaaS), and this constitutes one of the areas in which a considerable improvement can be made in the future.

Scalability and fault tolerance

- The ability to scale on demand constitutes one of the most attractive features of cloud computing.
- Clouds allow scaling beyond the limits of the existing in-house IT resources, whether they are infrastructure (compute and storage) or applications services.
- To implement such a capability, the cloud middleware has to be designed with the principle of scalability along different dimensions in mind - for eg, performance, size, and load.
- The ability to tolerate failure becomes fundamental, sometimes even more important than providing an extremely efficient and optimized system.
- The challenge in this case is designing highly scalable and fault-tolerant systems that are easy to manage and at the same time provide competitive performance.