MODEL 2

Ranjitha H.M
Department of ISE
Acharya Institute Of Technology

Module-2	Cloud Computing Architecture	8 hours

Cloud Computing Architecture, Introduction, Cloud Reference Model, Architecture, Infrastructure / Hardware as a Service, Platform as a Service, Software as a Service, Types of Clouds, Public Clouds, Private Clouds, Hybrid Clouds, Community Clouds, Economics of the Cloud, Open Challenges, Cloud Definition, Cloud Interoperability and Standards Scalability and Fault Tolerance Security, Trust, and Privacy Organizational Aspects Aneka: Cloud Application Platform, Framework Overview, Anatomy of the Aneka Container, From the Ground Up: Platform Abstraction Layer, Fabric Services, foundation Services, Application Services, Building Aneka Clouds, Infrastructure Organization, Logical Organization, Private Cloud Deployment Mode, Public Cloud Deployment Mode, Hybrid Cloud Deployment Mode, Cloud Programming and Management, Aneka SDK, Management Tools

Textbook 1: Ch. 4,5

INTRODUCTION

- Utility-oriented data centers are the first outcome of cloud computing, and they serve as the infrastructure through which the services are implemented and delivered.
- Any cloud service, whether virtual hardware, development platform, or application software, relies on a distributed infrastructure owned by the provider or rented from a third party.
- Clouds are built by relying on one or more datacenters. In most cases hardware resources are virtualized to provide isolation of workloads and to best exploit the infrastructure.
- *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 reference model

Cloud computing supports any IT service that can be consumed as a utility and delivered through a network, most likely the Internet. 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 (figure next slide), from hardware appliances to software systems.

* This layer is implemented using a datacenter in which hundreds and thousands of nodes are stacked together.

* Cloud infrastructure can be heterogeneous in nature because a variety of resources.

* Database systems and other storage services can also be part of the infrastructure.

The cloud reference model

* 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.

*Hardware virtualization is most commonly used at this level. Hypervisors manage the pool of resources and expose the distributed infrastructure as a collection of virtual machines.

*By using virtual machine technology it is possible to finely partition the hardware resources such as CPU and memory and to virtualize specific devices, thus meeting the requirements of users and applications.

* Infrastructure management is the key function of core middleware, which supports capabilities such as

1. negotiation of the quality of service,
2. admission control,
3. execution management and
4. monitoring, accounting, and
5. billing.

- The combination of cloud hosting platforms and resources is generally classified as a Infrastructure-as-a-Service (IaaS) solution.
- We can organize the different examples of IaaS into two categories:
  * Some of them provide both the management layer and the physical infrastructure;
  * Others provide only the management layer (IaaS (M)). In this second case, the management layer is often integrated with other IaaS solutions that provide physical infrastructure and adds value to them.
- IaaS solutions are suitable for designing the system infrastructure but provide limited services to build applications.
- PaaS- Users develop their applications specifically for the cloud by using the API exposed at the user-level middleware.
- This approach is also known as Platform-as-a-Service (PaaS) because the service offered to the user is a development platform rather than an infrastructure

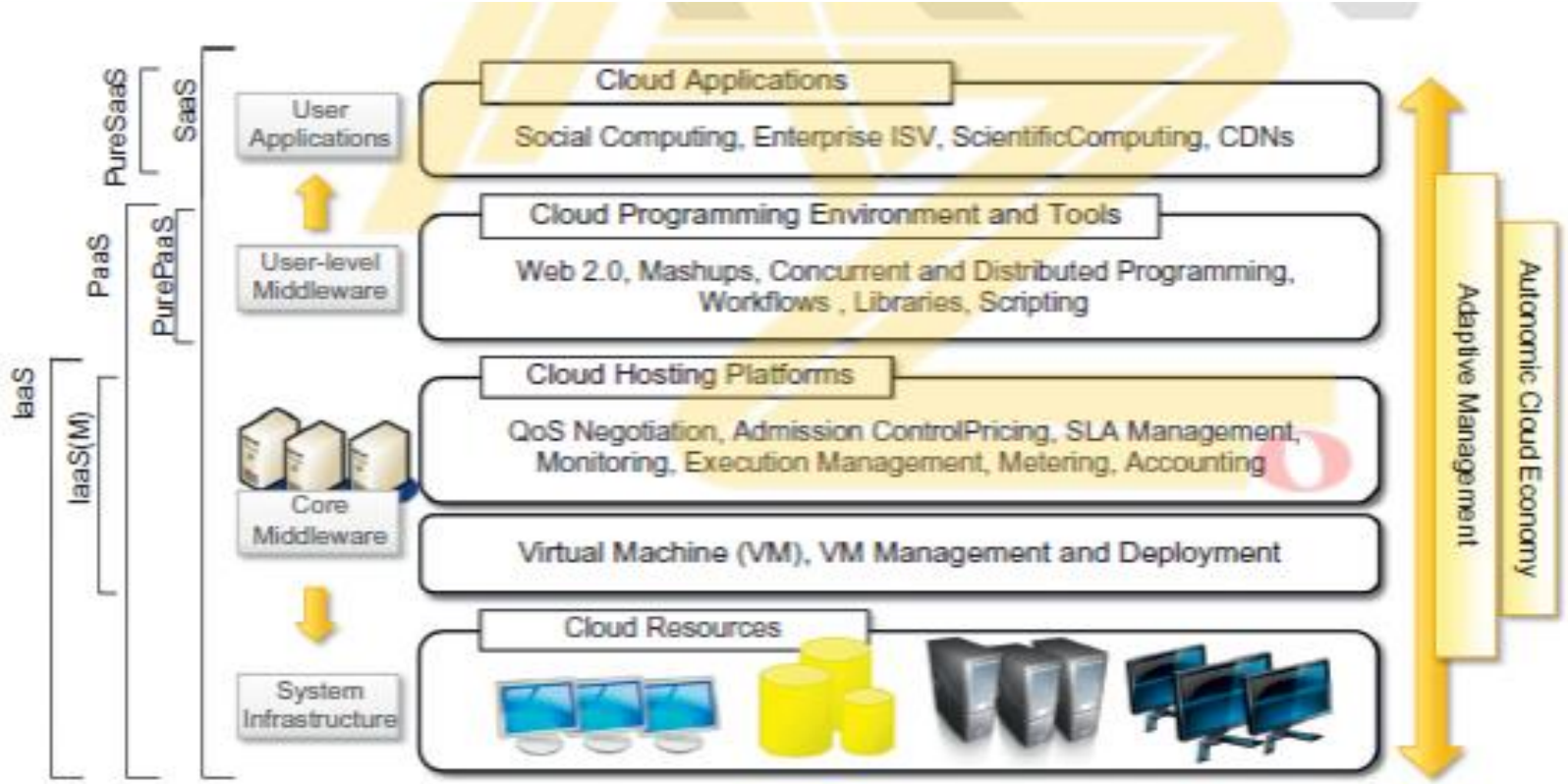


Figure 3.1 The cloud computing architecture.

* PaaS solutions generally include the infrastructure as well, which is bundled as part of the service provided to users.

* In the case of Pure PaaS, only the user-level middleware is offered, and it has to be complemented with a virtual or physical infrastructure.

- SaaS- Web-based applications that rely on the cloud to provide service to end users.

*The horsepower of the cloud provided by IaaS and PaaS solutions allows independent software vendors to deliver their application services over the Internet.

*SaaS implementations should feature adaptively change and expose an autonomic behavior automatically, whereas PaaS and IaaS generally provide this functionality as a part of the API exposed to users.

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.
- They deliver customizable infrastructure on demand. IaaS offering range from single servers to entire infrastructures, including network devices, load balancers, and database and Web servers.
- The main technology used to deliver and implement these solutions is hardware virtualization: one or more virtual machines opportunely configured and interconnected define the distributed system on top of which applications are installed and deployed
- IaaS/HaaS solutions bring all the benefits of hardware virtualization: workload partitioning, application isolation, sandboxing, and

Infrastructure- and hardware-as-a-service

- From the perspective of the customer it reduces the administration and maintenance cost as well as the capital costs allocated to purchase hardware.
- At the same time, users can take advantage of the full customization offered by virtualization to deploy their infrastructure in the cloud.
- Virtual machines come with only the selected operating system installed and the system can be configured with all the required packages and applications.
- Other solutions provide prepackaged system images that already contain the software stack required for the most common uses: Web servers, database servers, or LAMP stacks.
- Additional services can be provided: SLA resource-based allocation, workload management, support for infrastructure design through advanced Web interfaces, and the ability to integrate third-party IaaS solutions.

Infrastructure- and hardware-as-a-service

- It is possible to distinguish three principal layers:

->the physical infrastructure,

->the software management infrastructure, and

->the user interface.

1. User interface : It provides access to the services exposed by the software management infrastructure.

- Such an interface is generally based on Web 2.0 technologies: Web services, RESTful APIs, and mash-ups. These technologies allow either applications or final users to access the services.
- Web services and RESTful APIs allow programs to interact with the service without human intervention

Infrastructure- and hardware-as-a-service

2. Infrastructure management software layer.

The core features of an IaaS solution are implemented in the infrastructure management software layer.

- 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:
- 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 will be executed in the future.

Infrastructure- and hardware-as-a-service

- A QoS/SLA management component will maintain a repository of all the SLAs made with the users; together with the monitoring component.
- The VM repository component provides a catalog of virtual machine images that users can use to create virtual instances.
- A VM pool manager component is responsible for keeping track of all the live instances.
- 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.

3. Physical infrastructure: The bottom layer is composed of the physical infrastructure, on top of which the management layer operates

A cloud infrastructure developed in house, in a small or medium-sized enterprise or within a university department, will most likely rely on a cluster

Platform as a service

- 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.
- They 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.
- 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.\
- The specific development model decided for applications determines the interface exposed to the user.
- Other implementations of the PaaS model provide a complete object model for representing an application and provide a programming language-based approach.

Table 3.1 Cloud Computing Services Classification

Category	Characteristics	Product Type	Vendors and Products
SaaS	Customers are provided with applications that are accessible anytime and from anywhere.	Web applications and services (Web 2.0)	SalesForce.com (CRM) Clarizen.com (project management) Google Apps
PaaS	Customers are provided with a platform for developing applications hosted in the cloud.	Programming APIs and frameworks Deployment systems	Google AppEngine Microsoft Azure Manjrasoft Aneka Data Synapse
IaaS/HaaS	Customers are provided with virtualized hardware and storage on top of which they can build their infrastructure.	Virtual machine management infrastructure Storage management Network management	Amazon EC2 and S3 GoGrid Nirvanix

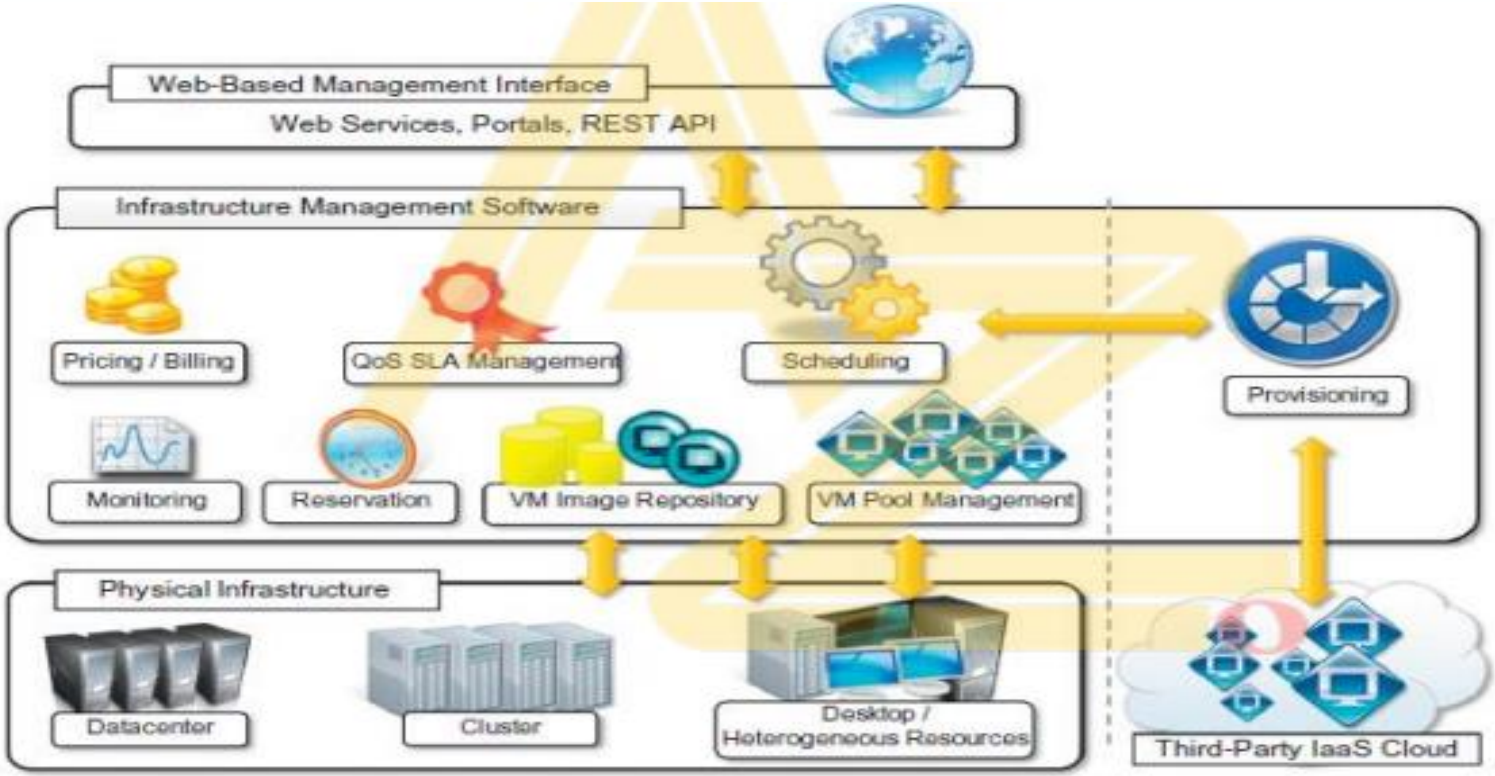


Figure 3.2 IaaS Reference implementation

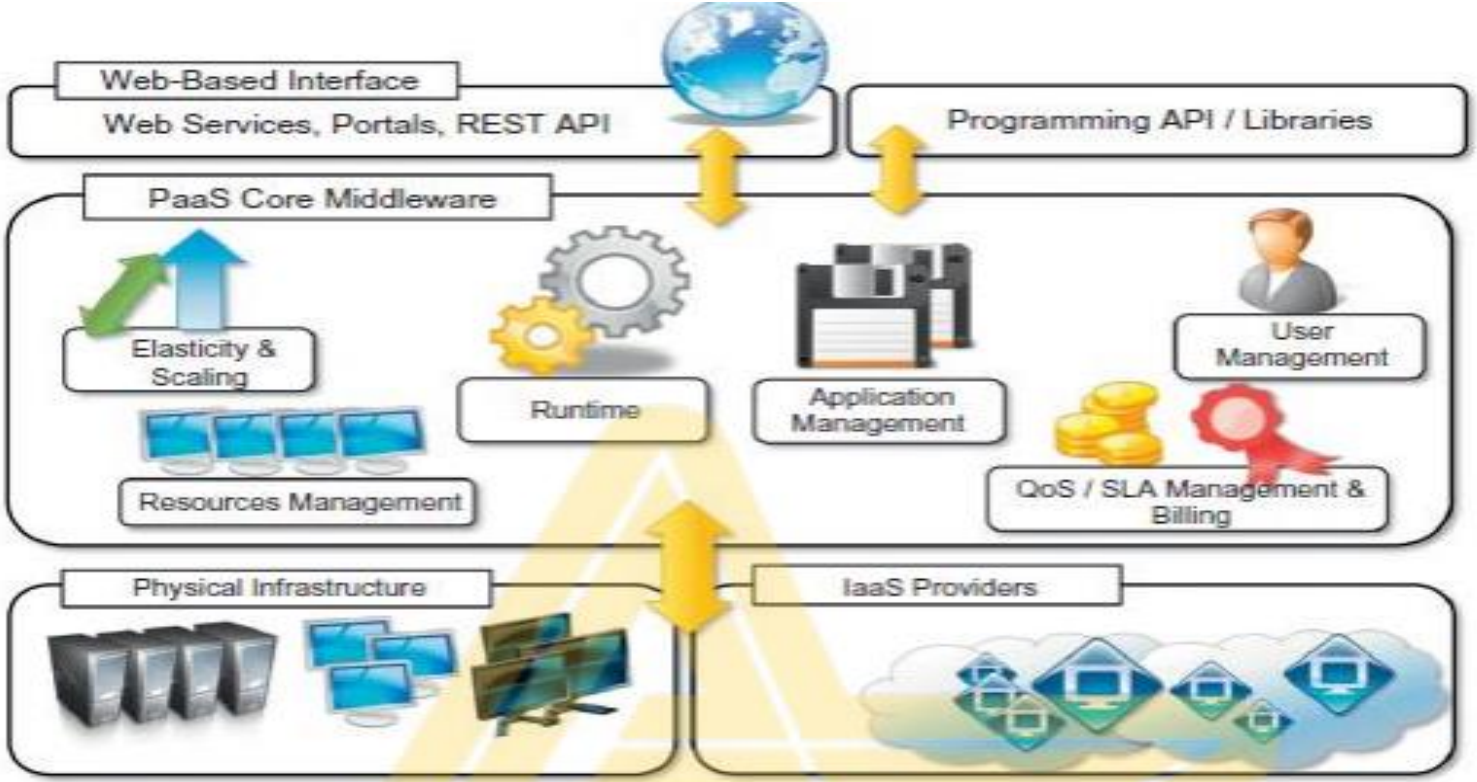


Figure 3.3 The Platform-as-a-Service reference model

Platform as a service

- Developers generally have the full power of programming languages such as Java, .NET, Python, or Ruby, with some restrictions to provide better scalability and security.
- 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, referred to in this book as Pure PaaS.

Platform as a service

- PaaS-I: The first category(PaaS-I) identifies PaaS implementations that completely follow the cloud computing style for application development and deployment.
- They offer an integrated development environment hosted within the Web browser where applications are designed, developed, composed, and deployed.
- Ex: Force.com and Longjump. Both deliver as platforms the combination of middleware and infrastructure.
- PaaS-II: we can list all those solutions that are focused on providing a scalable infrastructure for Web application, mostly websites.
- In this case, developers generally use the providers' APIs, which are built on top of industrial runtimes, to develop applications
- ex: Google AppEngine is the most popular product in this category
- It provides a scalable runtime based on the Java and Python programming languages .Joyent Smart Platform provides a similar approach to Google AppEngine. A different approach is taken by Heroku and Engine Yard, which provide scalability support for Ruby- and Ruby on Rails-based Websites

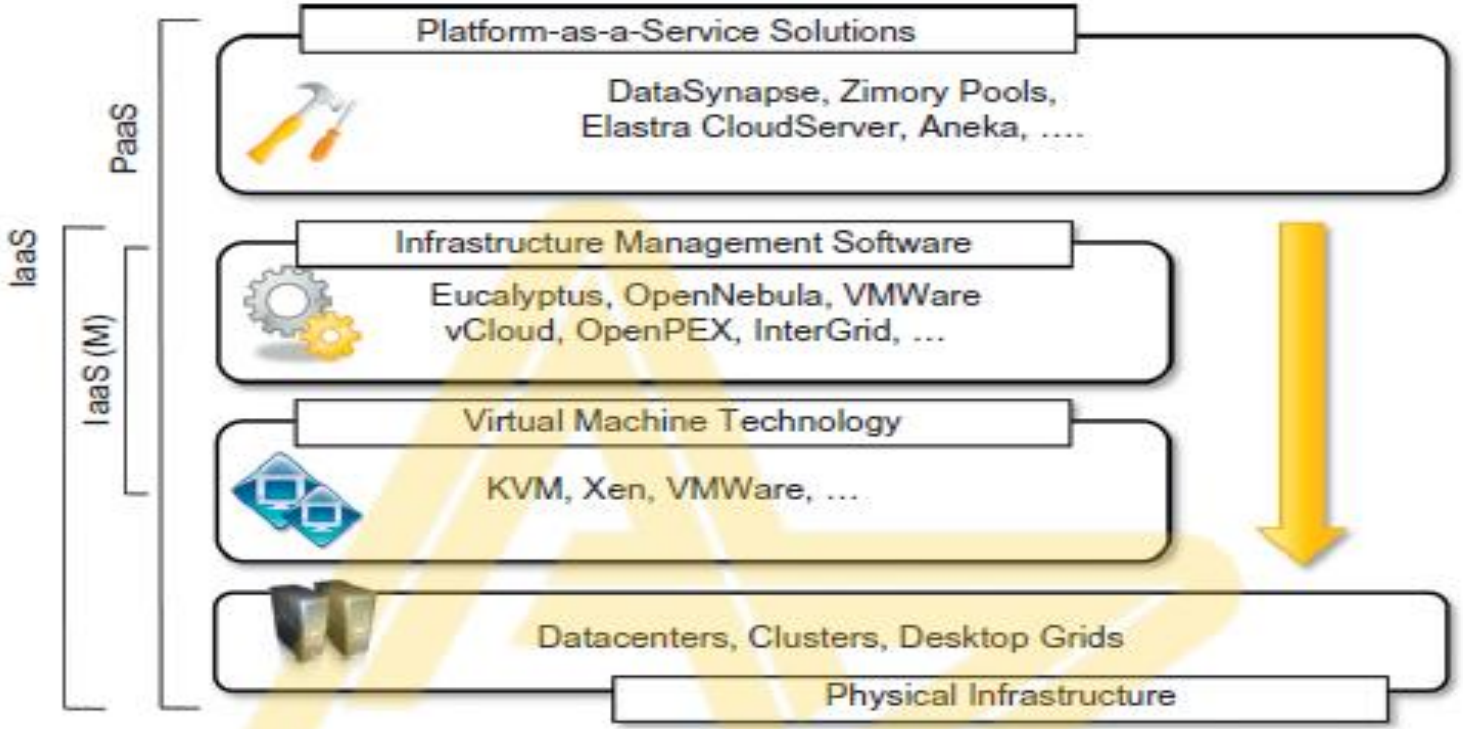


Figure 3.4 Private cloud software and hardware stack

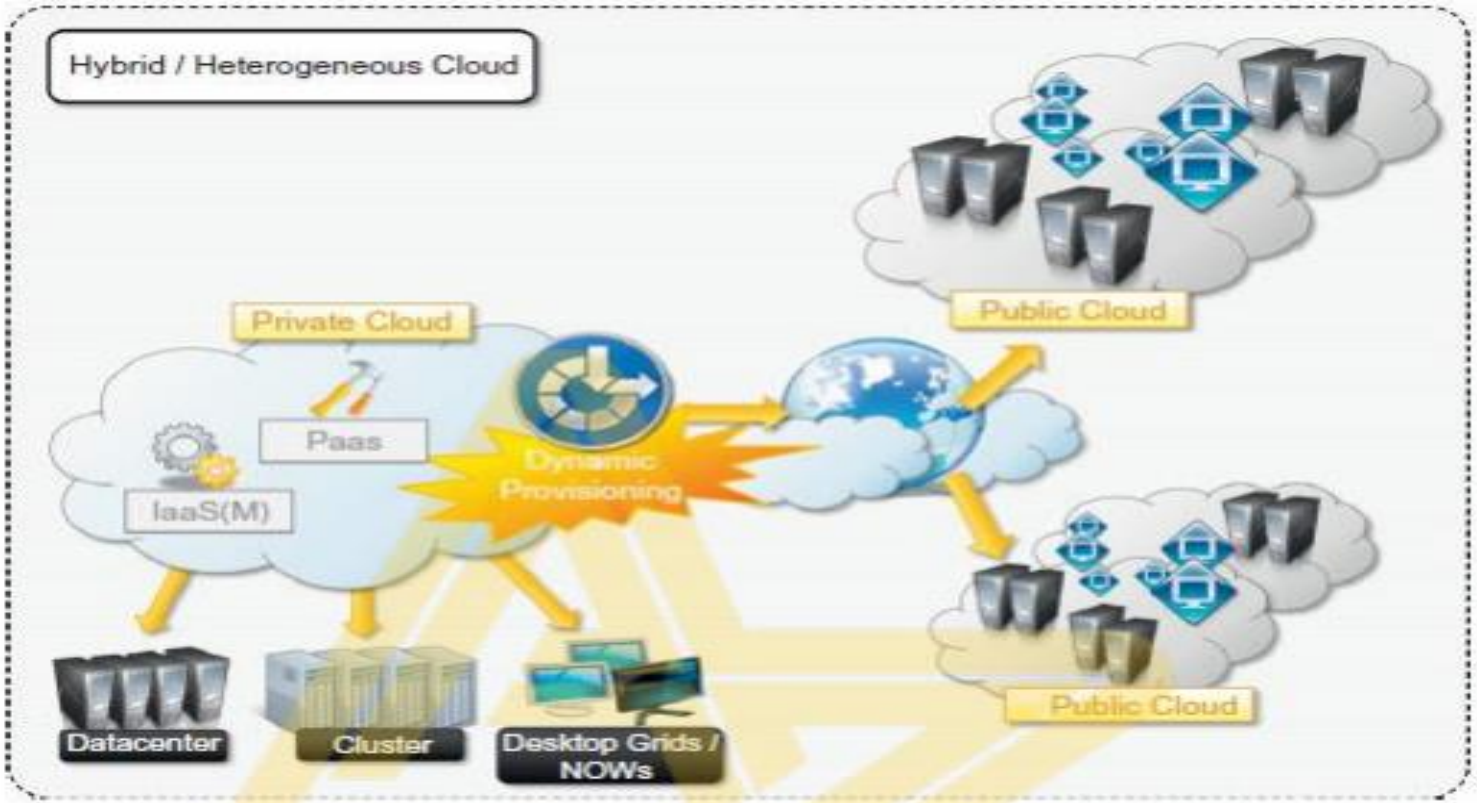


Figure 3.5 Hybrid/heterogeneous cloud overview.

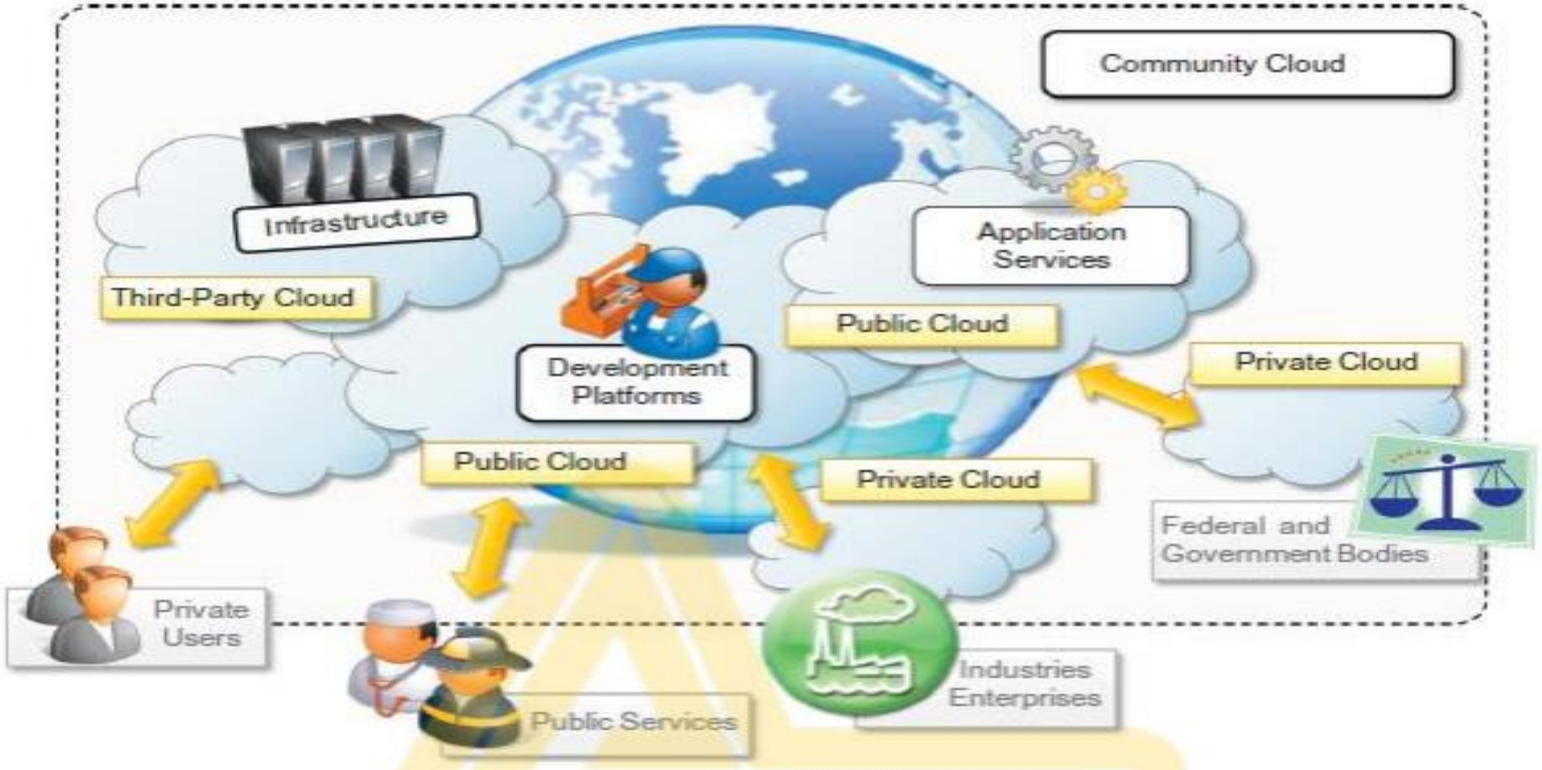


Figure 3.5 A Community Cloud

Economics of the cloud

The main drivers of cloud computing are economy of scale and simplicity of software delivery and its operation. In fact, the biggest benefit of this phenomenon is financial: *the pay-as-you-go model* offered by cloud providers.

In particular, cloud computing allows:

*Reducing the capital costs associated to the IT infrastructure

*Eliminating the depreciation or lifetime costs associated with IT capital assets

*Replacing software licensing with subscriptions

*Cutting the maintenance and administrative costs of IT resources

•A capital cost is the cost occurred in purchasing an asset that is useful in the production of goods or the rendering of services. Capital costs are one-time expenses that are generally paid up front and that will contribute over the long term to generate profit.

Economics of the cloud

- Enterprise will definitely have an IT department that is used to automate many of the activities that are performed within the enterprise: payroll, customer relationship management, enterprise resource planning, tracking and inventory of products, and others.
- In the case of IT capital costs, the depreciation costs are represented by the loss of value of the hardware over time and the aging of software products that need to be replaced because new features are required.
- Many enterprises own a small or medium-sized datacenter that introduces several operational costs in terms of maintenance, electricity, and cooling.Moreover, other costs are triggered by the purchase of potentially expensive software. With cloud computing these costs are significantly reduced or simply disappear according to its penetration.
- Cloud computing also introduces reductions in administrative and maintenance costs leverage
- In the case of a small startup, it is possible to completely leverage the cloud for many aspects, such as:
- IT infrastructure
- Software development

Economics of the cloud

- In terms of the pricing models introduced by cloud computing, we can distinguish three different strategies that are adopted by the providers:

- Tiered pricing. In this model, cloud services are offered in several tiers, each of which offers a fixed computing specification and SLA at a specific price per unit of time. This model is used by Amazon for pricing the EC2 service, which makes available different server configurations in terms of computing capacity (CPU type and speed, memory) that have different costs per hour.
- Per-unit pricing. This model is more suitable to cases where the principal source of revenue for the cloud provider is determined in terms of units of specific services, such as data transfer and memory allocation. In this scenario customers can configure their systems more efficiently according to the application needs. This model is used, for example, by GoGrid, which makes customers pay according to RAM/hour units for the servers deployed in the GoGrid cloud.
- Subscription-based pricing. This is the model used mostly by SaaS providers in which users pay a periodic subscription fee for use of the software or the specific component services that are integrated in their applications.

Cloud Application Programming and the Aneka Platform

Framework overview

- Aneka is a software platform for developing cloud computing applications.
- It allows harnessing of disparate computing resources and managing them into a unique virtual domain—the Aneka Cloud—in which applications are executed.
- According to the Cloud Computing Reference Model ,it is a pure PaaS solution for cloud computing.
- Aneka is a cloud middleware product that can be deployed on a heterogeneous set of resources:

a network of computers, a multicore server, datacenters,

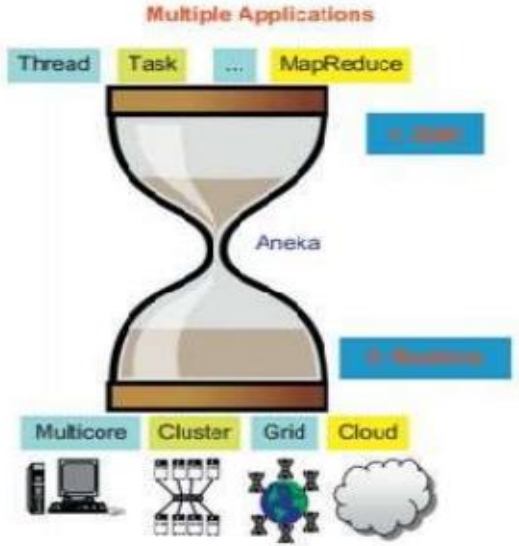
virtual cloud infrastructures, or a mixture of these.

- The framework provides both middleware for managing and scaling distributed applications and an extensible set of APIs for developing them.

Framework overview

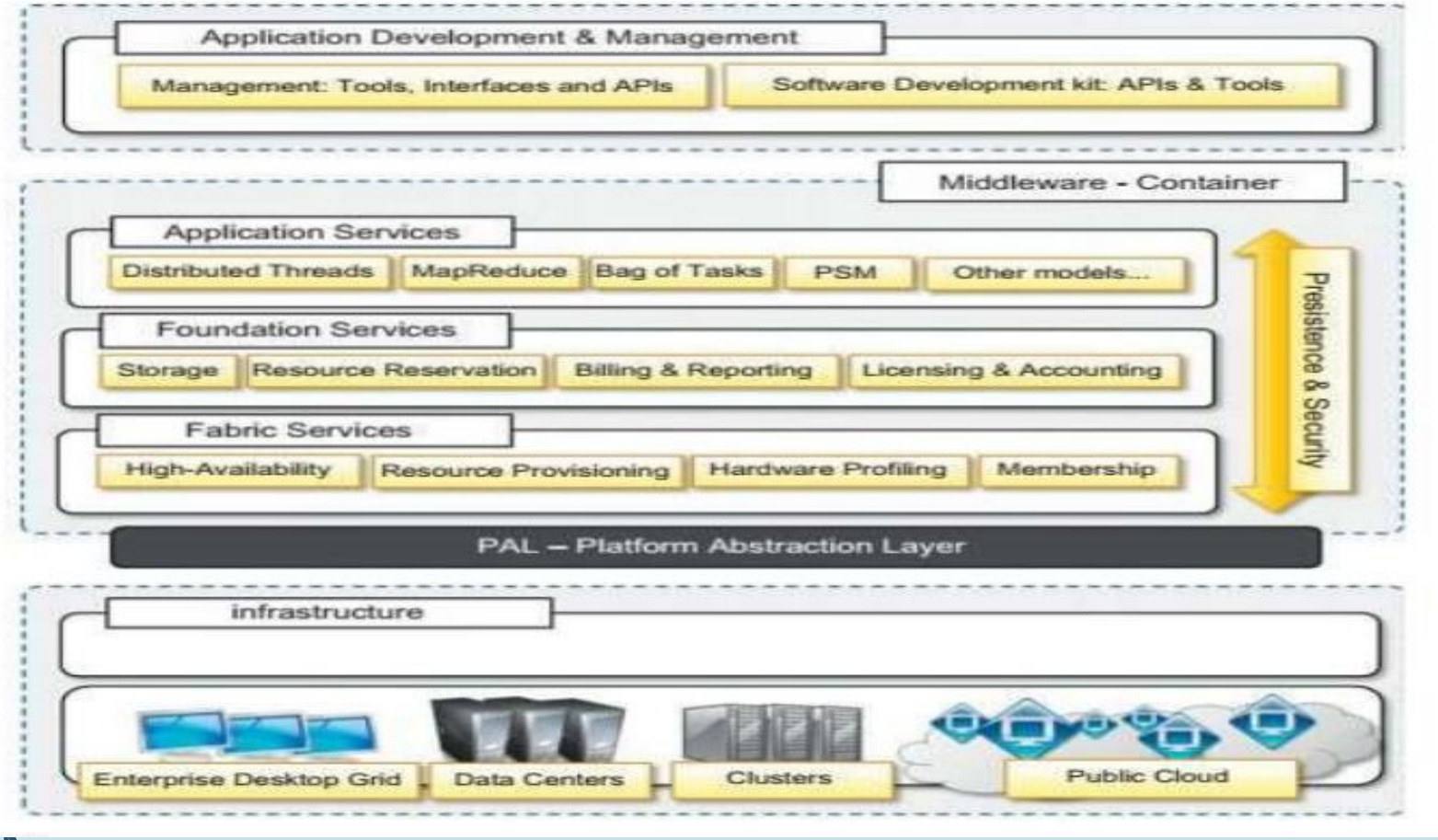
- One of Aneka's key advantages- its extensible set of APIs associated with different types of programming models—such as Task, Thread, and MapReduce

—used for developing distributed applications, integrating new capabilities into the cloud, and supporting different types of cloud deployment models: public, private, and hybrid (see Figure)



Framework overview

- Figure 5.2 provides a complete overview of the components of the Aneka framework. The core infrastructure of the system allows the framework to be deployed over different platforms and operating systems.
- Aneka container, which is installed on each node and constitutes the basic building block of the middleware.
- A collection of interconnected containers constitute the Aneka Cloud: a single domain in which services are made available to users and developers.
- The container features three different classes of services: Fabric Services, Foundation Services, and Execution Services.
- These take care of infrastructure management, supporting services for the Aneka Cloud, and application management and execution, respectively.
- These services are made available to developers and administrators by means of the application management and development layer, which includes
  - * interfaces and APIs for developing cloud applications and
  - * management tools and interfaces for controlling Aneka Clouds.


Application Development & Management
Management: Tools, Interfaces and APIs
Software Development kit: APIs & Tools
Middleware - Container
Application Services
Distributed Threads
MapReduce
Bag of Tasks
PSM
Other models...
Foundation Services
Storage
Resource Reservation
Billing & Reporting
Licensing & Accounting
Fabric Services
High-Availability
Resource Provisioning
Hardware Profiling
Membership
Presistence & Security
PAL – Platform Abstraction Layer
infrastructure
Enterprise Desktop Grid
Data Centers
Clusters
Public Cloud

ACHARYA

Framework overview

- Aneka implements a service-oriented architecture (SOA), and services are the fundamental components of an Aneka Cloud.
- Services operate at container level and, except for the PAL, they provide developers, users, and administrators with all features offered by the framework.
- Services also constitute the extension and customization point of Aneka Clouds:
  - 1. The infrastructure allows for the integration of new services or replacement of the existing ones with a different implementation.
  - 2. The framework includes the
    - basic services for infrastructure and node management,
    - application execution,
    - accounting, and
    - system monitoring;
    - existing services can be extended and new features can be added to the cloud by dynamically plugging new ones into the container.

Framework overview

- Aneka implements a service-oriented architecture (SOA), and services are the fundamental components of an Aneka Cloud.
- Services operate at container level and, except for the PAL, they provide developers, users, and administrators with all features offered by the framework.
- Services also constitute the extension and customization point of Aneka Clouds:
  - 1. The infrastructure allows for the integration of new services or replacement of the existing ones with a different implementation.
  - 2. The framework includes the
    - basic services for infrastructure and node management,
    - application execution,
    - accounting, and
    - system monitoring;
    - existing services can be extended and new features can be added to the cloud by dynamically plugging new ones into the container.

ACHARYA

Framework overview

- Aneka services involve:
- Elasticity and scaling. By means of the dynamic provisioning service, Aneka supports dynamically upsizing and downsizing of the infrastructure available for applications.
- Runtime management. The runtime machinery is responsible for keeping the infrastructure up and running and serves as a hosting environment for services. (achieve through container)
- Resource management. Aneka is an elastic infrastructure in which resources are added and removed dynamically according to application needs and user requirements.

  To provide QoS-based execution, the system not only allows dynamic provisioning but also provides capabilities for reserving nodes for exclusive use by specific applications.
- Application management. A specific subset of services is devoted to managing applications. These services include scheduling, execution, monitoring, and storage management.

Framework overview

- User management. Aneka is a multitenant distributed environment in which multiple applications, potentially belonging to different users, are executed.

* The framework provides an extensible user system via which it is possible to define users, groups, and permissions. The services devoted to user management build up the security infrastructure of the system.

- QoS/SLA management and billing. Within a cloud environment, application execution is metered and billed.

* Aneka provides a collection of services that coordinate together to take into account the usage of resources by each application and to bill the owning user accordingly.

Anatomy of the Aneka container

- The services installed in the Aneka container can be classified into three major categories:
  - Fabric Services
  - Foundation Services
  - Application Services
- The services stack resides on top of the Platform Abstraction Layer (PAL), representing the interface to the underlying operating system and hardware.

1. ***From the ground up: the platform abstraction layer***

- The core infrastructure of the system is based on the .NET technology and allows the Aneka container to be portable over different platforms and OS.
- Any platform featuring an ECMA-334 and ECMA-335 compatible environment can host and run an instance of the Aneka container.
- The Common Language Infrastructure (CLI), which is the specification introduced in the ECMA-334 standard, defines a common runtime environment (for executing programs) but does not provide any interface.

From the ground up: the platform abstraction layer

- Moreover, each OS has a different file system organization and stores that information differently.
- The Platform Abstraction Layer (PAL) addresses this heterogeneity and provides the container with a uniform interface for accessing the relevant hardware and operating system information.
- The PAL is responsible for detecting the supported hosting environment(detection engine)
- The PAL provides the following features:
  - Uniform and platform-independent implementation interface for accessing the hosting platform
  - Uniform access to extended and additional properties of the hosting platform
  - Uniform and platform-independent access to remote nodes
  - Uniform and platform-independent management interfaces
- The PAL is a small layer of software that comprises a detection engine, which automatically configures the container at boot time.

From the ground up: the platform abstraction layer

- The collectible data that are exposed by the PAL are the following:
  - Number of cores, frequency, and CPU usage
  - Memory size and usage
  - Aggregate available disk space
  - Network addresses and devices attached to the node

Fabric services

- Fabric Services define the lowest level of the software stack representing the Aneka Container.
- They provide access to the resource-provisioning subsystem and to the monitoring facilities implemented in Aneka.
  - Resource-provisioning services- dynamically providing new nodes on demand by relying on virtualization technologies,
  - Monitoring services - allow for hardware profiling and implement a basic monitoring infrastructure that can be used by all the services installed in the container.

1. ***Profiling and monitoring***

- Profiling and monitoring services are mostly exposed through the

  * *Heartbeat-* makes available the information that is collected through the PAL

  * *Monitoring, and Reporting Services* - implement a generic infrastructure for monitoring the activity of any service in the Aneka Cloud.

Fabric services

- Heartbeat: The Heartbeat Service periodically collects the dynamic performance information about the node and publishes this information to the membership service in the Aneka Cloud.
  * These data are collected by the index node of the Cloud, which makes them available for services such as reservations and scheduling.
  * Basic information about memory, disk space, CPU, and operating system is collected.
- A specific component, called Node Resolver, is in charge of collecting these data and making them available to the Heartbeat Service.
- Reporting and Monitoring Services. The Reporting Service manages the store for monitored data and makes them accessible to other services or external applications for analysis purposes.
  * On each node, an instance of the Monitoring Service acts as a gateway to the Reporting Service and forwards to it all the monitored data that has been collected on the node. Any service that wants to publish monitoring data can leverage the local monitoring service.

Fabric services

- Currently several built-in services provide information through this channel:
  * *The Membership Catalogue* tracks the performance information of nodes.
  * *The Execution Service monitors* several time intervals for the execution of jobs.
  * *The Scheduling Service* tracks the state transitions of jobs.
  * The *Storage Service* monitors and makes available information about data transfer, such as upload and download times, filenames, and sizes.
  * The *Resource Provisioning Service* tracks the provisioning and lifetime information of virtual nodes.

5.2.2.2 Resource management

- Resource management is another fundamental feature of Aneka Clouds. It comprises several tasks: resource membership, resource reservation, and resource provisioning.

Aneka provides a collection of services that are in charge of managing resources. These are the Index Service (or Membership Catalogue), Reservation Service, and Resource Provisioning Service

Fabric services

- The Membership Catalogue –

* keeps track of the basic node information for all the nodes that are connected or disconnected.

* implements the basic services of a directory service, allowing the search for services using attributes such as names and nodes.

* During container startup, each instance publishes its information to the Membership Catalogue and updates it constantly during its lifetime.

* Services and external applications can query the membership catalogue to discover the available services and interact with them.

* collector of the dynamic performance data of each node, which are then sent to the local monitoring service.

- Resource provisioning –

* allows the integration and management of virtual resources leased from IaaS providers into the Aneka Cloud.

* changes the structure of the Aneka Cloud by allowing it to scale up and down according to different needs: handling node failures, ensuring the quality of service for applications, or maintaining a constant performance and throughput of the Cloud

Foundation services

- Foundation Services are related to the logical management of the distributed system built on top of the infrastructure and provide supporting services for the execution of distributed applications.
- All the supported programming models can integrate with and leverage these services to provide advanced and comprehensive application management. These services cover:
  - * Storage management for applications
  - * Accounting, billing, and resource pricing
  - * Resource reservation
- Foundation Services provide a uniform approach to managing distributed applications and allow developers to concentrate only on the logic that distinguishes a specific programming model from the others.
- Together with the Fabric Services, Foundation Services constitute the core of the Aneka middleware.
- These services are mostly consumed by the execution services and Management Consoles. External applications can leverage the exposed capabilities for providing advanced application management.

Foundation services

•5.2.3.1 Storage management

- File/Data transfer management is an important aspect of any distributed system, even in computing clouds(data persisted and moved in the format of files).
- Aneka offers two different facilities for storage management: Centralized file storage and Distributed file system.

•*A Centralized File Storage-* used for the execution of compute-intensive applications (appln require powerful processors and do not have high demands in terms of storage, used to store small files that are easily transferred from one node to another..applicable for a centralized storage node, or a pool of storage nodes)

*It consitutes Aneka's data-staging facilities- It provides distributed applications with the basic file transfer facility and abstracts the use of a specific protocol (FTP)to end users

* To support different protocols, the system introduces file channel components: a file channel controller and a file channel handler.

Foundation services

- **5.2.3.1 Storage management(contd)**

  * The file channel controller constitutes the server component of the channel, where files are stored and made available; the file channel handler represents the client component, which is used by user applications or other components of the system to upload, download, or browse files

  * The storage service uses the configured file channel factory to first create the server component that will manage the storage and then create the client component on demand.

  * File Channel Abstraction is the ability to chain two different channels to move files by using two different protocols. For example, an output file should store in S3 bucket. worker node -> Storage Service (using the internal FTP protocol) -> S3 (by the FTP channel controller)

- ***Distributed File System*** -for the execution of data-intensive applications(appln are characterized by large data files (gigabytes or terabytes), and the processing power required by tasks does not constitute a performance bottleneck. Applicable for a distributed file system)

Foundation services

- **5.2.3.1 Storage management(contd)**
  - * The reference model for the distributed file system is the Google File System.
  - * The architecture of the file system is based on a master node, which contains a global map of the file system and keeps track of the status of all the storage nodes, and a pool of chunk servers, which provide distributed storage space in which to store files.
  - * Files are logically organized into a directory structure.Each file is organized as a collection of chunks that are all of the same size. File chunks are assigned unique IDs and stored on different servers, eventually replicated to provide high availability and failure tolerance.
  - * the only programming model that makes use of the distributed file system is MapReduce
- **5.2.3.2 Accounting, billing, and resource pricing**

Foundation services

- **5.2.3.2 Accounting, billing, and resource pricing(contd)**

•Billing is another important feature of accounting. Aneka is a multitenant cloud programming platform. Aneka Billing Service provides detailed information about each user's usage of resources, with the associated costs. Each resource can be priced differently according to the set of services that are available.

•The accounting model provides an integrated view of budget spent for each application, a summary view of the costs associated to a specific user, and the detailed information about the execution cost of each job

•Difference b/w Accounting Service and the Reporting Service.

Accounting- Keeps track of the information that is related to application execution, such as the distribution of jobs among the available resources, the timing of each of job, and the associated cost.

Reporting -makes available the information collected from the monitoring services for accounting purposes: storage utilization and CPU performance.

Foundation services

5.2.3.3 Resource reservation

- Resource Reservation supports the execution of distributed applications and allows for reserving resources for exclusive use by specific applications.
- Resource reservation is built out of two different kinds of services:

1. Resource Reservation- It keeps track of all the reserved time slots in the Aneka Cloud and provides a unified view of the system.
2. Allocation Service. It is installed on each node that features execution ser-vices and manages the database of information regarding the allocated slots on the local node.

- At the moment, the framework supports three different implementations:

1. Basic Reservation. Features the basic capability to reserve execution slots on nodes and implements the alternate offers protocol, which provides alternative options in case the initial reservation requests cannot be satisfied.
2. Libra Reservation. Represents a variation of the previous implementation that features the ability to price nodes differently according to their hardware capabilities.
3. Relay Reservation. Constitutes a very thin implementation that allows a resource broker to reserve nodes in Aneka Clouds and control the logic with which these nodes are reserved. This implementation is useful in integration scenarios in which Aneka operates in an intercloud environment.

Application services

- Application Services manage the execution of applications and constitute a layer that differentiates according to the specific programming model used for developing distributed applications on top of Aneka.
- It is possible to identify two major types of activities that are common across all the supported models: scheduling and execution.

- **5.2.4.1 Scheduling**
- Scheduling Services are in charge of planning the execution of distributed applications on top of Aneka and governing the allocation of jobs composing an application to nodes. They also constitute the integration point with several other Foundation and Fabric Services, such as the Resource Provisioning Service, the Reservation Service, the Accounting Service, and the Reporting Service. Common tasks that are performed by the scheduling component are the following:

- Job to node mapping
- Rescheduling of failed jobs
- Job status monitoring
- Application status monitoring

Application services

- **5.2.4.1 Scheduling (contd)**
  - Aneka does not provide a centralized scheduling engine.
  - Different scheduling engines for different models gives great freedom in implementing scheduling and resource allocation strategies .
  - common situations that have to be appropriately managed are the following: *multiple jobs sent to the same node at the same time; jobs without reservations sent to reserved nodes; and jobs sent to nodes where the required services are not installed.*
  - Aneka's Foundation Services provide sufficient information to avoid these cases.
  - it is possible to enforce that only one job per programming model is run on each node at any given time, but the execution of applications is not mutually exclusive unless Resource Reservation is used.
- **5.2.4.2 Execution**
- Execution Services control the execution of single jobs that compose applications.

Application services

•They are in charge of setting up the runtime environment hosting the execution of jobs. As happens for the scheduling services, each programming model has its own requirements, but it is possible to identify some common operations that apply across all the range of supported models:

*Unpacking the jobs received from the scheduler

*Retrieval of input files required for job execution

*Sandboxed execution of jobs

*Submission of output files at the end of execution

*Execution failure management (i.e., capturing sufficient contextual information useful to identify the nature of the failure)

*Performance monitoring

*Packing jobs and sending them back to the scheduler

•Execution Services handle less information and are required to integrate themselves only with the Storage Service and the local Allocation and Monitoring Services.

•Application Services constitute the runtime support of the programming model in the Aneka Cloud

Application services

- Currently there are several supported models:

1. Task Model. This model provides the support for the independent "bag of tasks" applications and many computing tasks. In this model, an application is modeled as a collection of tasks that are independent from each other and whose execution can be sequenced in any order.
2. Thread Model. This model provides an extension to the classical multithreaded programming to a distributed infrastructure and uses the abstraction of Thread to wrap a method that is executed remotely.
3. MapReduce Model. This is an implementation of MapReduce as proposed by Google on top of Aneka.
4. Parameter Sweep Model. This model is a specialization of the Task Model for applications that can be described by a template task whose instances are created by generating different combinations of parameters, which identify a specific point into the domain of interest.

- Other programming models Dataflow Model, the Message-Passing Interface, and the Actor Model developed for internal use.

Building Aneka clouds

Aneka Cloud can be realized by two methods:

- Infrastructure Organization
- Logical Organization

Infrastructure organization

Infrastructure based organization of Aneka Cloud is given in the following figure-5.3: The working mechanism of this model:

FIGURE 5.3
Aneka cloud infrastructure overview.

It contains **Aneka Repository, Administrative Console, Aneka Containers & Node Managers as major components.**

Logical Organization

The logical organization of Aneka Clouds can be very diverse, since it strongly depends on the configuration selected for each of the container instances belonging to the Cloud.

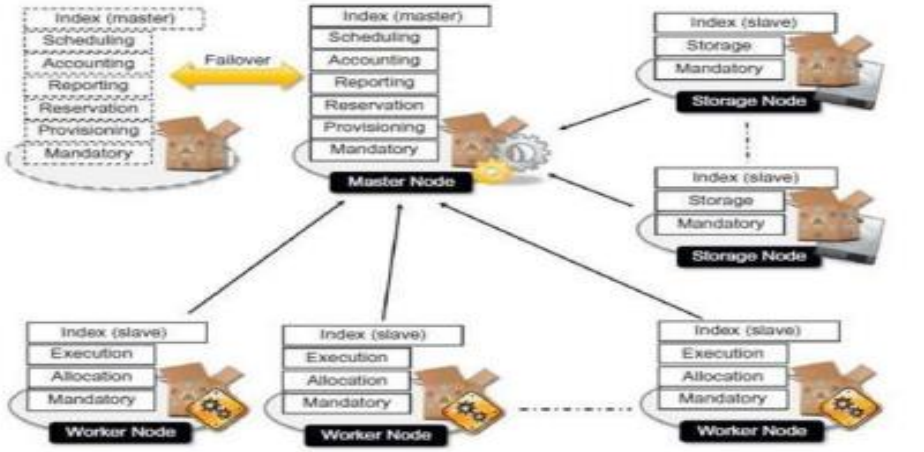


FIGURE 5.4
Logical organization of an Aneka cloud.

Private cloud deployment mode

Aneka Cloud Deployment Models

All the general cloud deployment models like Private cloud deployment mode, Public cloud deployment mode and Hybrid Cloud deployment mode are applicable to Aneka Clouds also.

Private cloud deployment mode

A private deployment mode is mostly constituted by local physical resources and infrastructure management software providing access to a local pool of nodes, which might be virtualized.

Figure 5.5 shows a common deployment for a private Aneka Cloud.

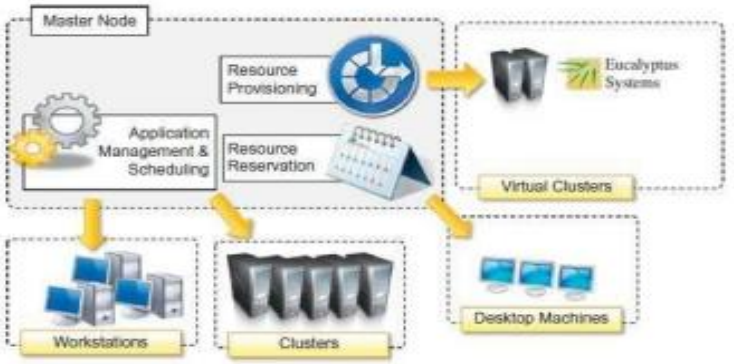


FIGURE 5.5
Private cloud deployment.

Public cloud deployment mode

Public Cloud deployment mode features the installation of Aneka master and worker nodes over a completely virtualized infrastructure that is hosted on the infrastructure of one or more resource providers such as Amazon EC2 or GoGrid.

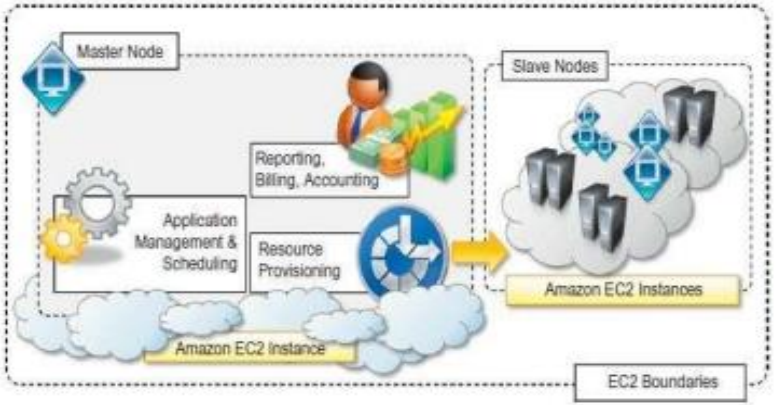


FIGURE 5.6

Public Aneka cloud deployment.

Hybrid cloud deployment mode

- The hybrid deployment model constitutes the most common deployment of Aneka. In many cases, there is an existing computing infrastructure that can be leveraged to address the computing needs of applications.
- This infrastructure will constitute the static deployment of Aneka that can be elastically scaled on demand when additional resources are required

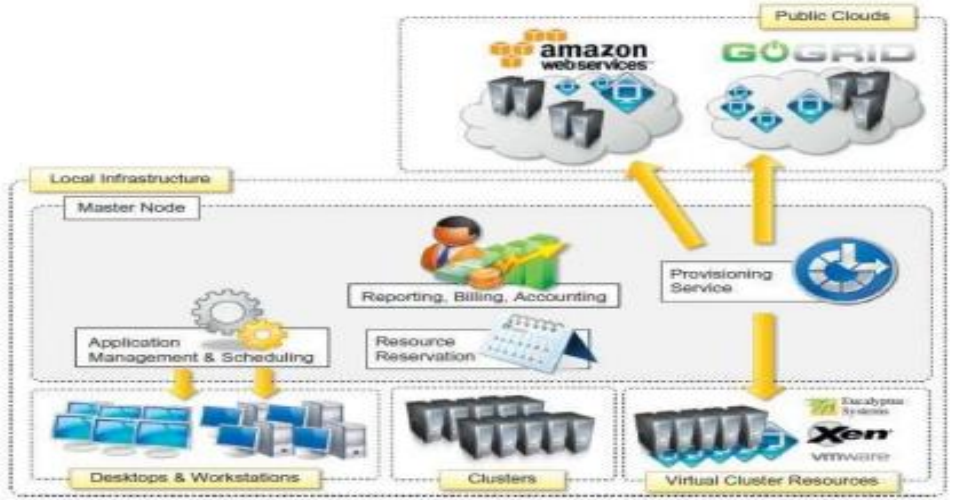


FIGURE 5.7
Hybrid cloud deployment.

Cloud programming and management

- Aneka's primary purpose is to provide a scalable middleware product in which to execute distributed applications.
- Application development and management constitute the two major features that are exposed to developers and system administrators.
- Aneka provides developers with a comprehensive and extensible set of APIs and administrators with powerful and intuitive management tools.
- The APIs for development are mostly concentrated in the Aneka SDK; management tools are exposed through the Management Console

Aneka SDK

- Aneka provides APIs for developing applications on top of existing programming models, implementing new programming models, and developing new services to integrate into the Aneka Cloud.
- The SDK provides support for both programming models and services by
  - The Application Model
  - The Service Model.

Application Model

- The Application Model covers the development of applications and new programming models
- It Consists of Application Class & Application Manager
- Application Class – Provide user/developer view about distributed applications of the Aneka cloud
- Application Manager – Are Internal components that control and monitor the execution of Aneka clouds

The Service Model

- defines the general infrastructure for service development.
- The Aneka Service Model defines the basic requirements to implement a service that can be hosted in an Aneka Cloud.
- The container defines the runtime environment in which services are hosted. Each service that is hosted in the container must use IService interface, which exposes the following methods and properties:
  - Name and status
  - Control operations such as Start, Stop, Pause, and Continue methods
  - Message handling by means of the HandleMessage method

MANAGEMENT TOOLS

Infrastructure management

Aneka leverages virtual and physical hardware in order to deploy Aneka Clouds. Virtual hardware is generally managed by means of the Resource Provisioning Service, which acquires resources on demand according to the need of applications, while physical hardware is directly managed by the Administrative Console by leveraging the Aneka management API of the PAL.

Platform management

The creation of Clouds is orchestrated by deploying a collection of services on the physical infra- structure that allows the installation and the management of containers. A collection of connected containers defines the platform on top of which applications are executed. The features available for platform management are mostly concerned with the logical organization and structure of Aneka Clouds.

Application management

Applications identify the user contribution to the Cloud. This is an important feature in a cloud computing scenario in which users are billed for their resource usage. Aneka exposes capabilities for giving summary and detailed information about application execution and resource utilization.


Thank You!