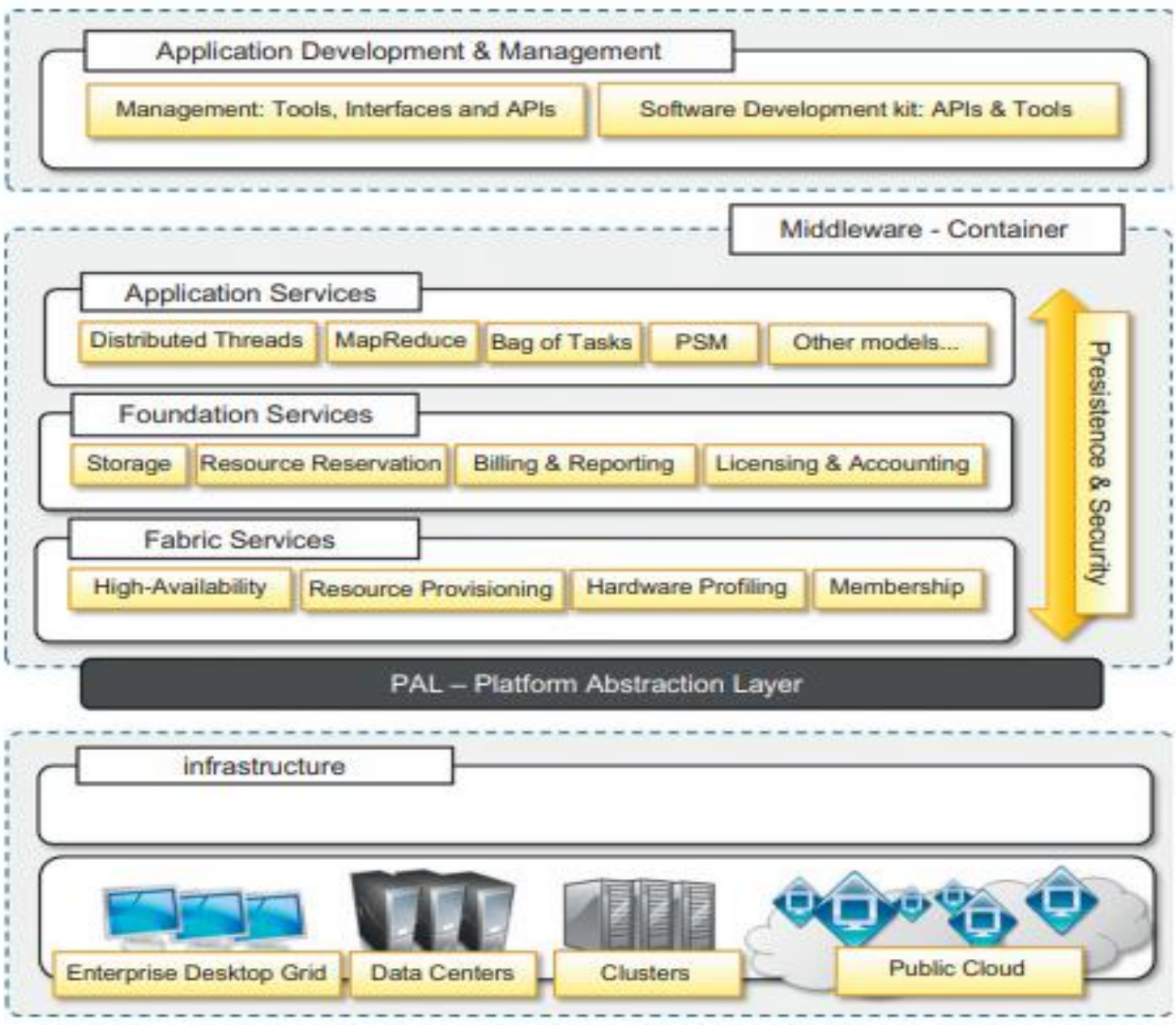
Module 2- Chapter 5
Aneka Cloud Application Platform

- Aneka is a software platform for developing cloud computing applications.
- Aneka is a pure PaaS solution for cloud computing.
- Aneka is a cloud middleware product that can be deployed on a heterogeneous set of resources: Like: a network of computers, a multi core server, data centers, virtual cloud infrastructures, or a mixture of all
- The framework provides both middleware for managing and scaling distributed applications and an extensible set of APIs for developing them.

Aneka Framework with Diagram



- A collection of interconnected containers constitutes 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,
  - Execution Services.
- Fabric services take care of infrastructure management for the Aneka Cloud
- Foundation Services take care of supporting services for the Aneka Cloud
- Application Services take care of application management and execution respectively

Various Services offered by Aneka Cloud Platform are:

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 run time machinery is responsible for keeping the infrastructure up and running and serves as a hosting environment for services.

Resource management: Aneka is an elastic infrastructure in which resources are added and removed dynamically according to application needs and user requirement

Application management: A specific subset of services is devoted to managing applications. These services include scheduling, execution, monitoring, and storage management.

User management: Aneka is a multi-tenant 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.

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 Aneka container constitutes the building blocks of Aneka Clouds and represents the runtime machinery available to services and applications
- The container is the unit of deployment in Aneka Clouds, and it is a lightweight software layer designed to host services and interact with the underlying operating system and hardware

The Aneka container can be classified into three major categories:
- Fabric Services
- Foundation Services
- Application Services
  - These services stack resides on top of the Platform Abstraction Layer (PAL) (Refer Diagram-5.2) it represents the interface to the underlying operating system and hardware.
  - PAL provides a uniform view of the software and hardware environment in which the container is running

Here is the functionality of each component of Aneka Framework

PAL –Platform Abstraction Layer

- In a cloud environment each operating system has a different file system organization and stores that information differently.
- It is The Platform Abstraction Layer (PAL) that addresses this heterogeneity problem with Operating systems and provides the container with a uniform interface for accessing the relevant information, thus the rest of the container can be operated without modification on any supported platform

The PAL provides the following features:
- Uniform and platform-independent implementation interface for accessing thehosting 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

Also The PAL is a small layer of software that comprises a detection engine, which automatically configures the container at boot time, with the platform-specific component to access the above information and an implementation of the abstraction layer for the Windows, Linux, and Mac OS X operating systems.

Following are the collectible data that are exposed by the PAL:
- 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 are in charge of 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

The two services of Fabric class are:

- Profiling and monitoring
- Resource management

Profiling and monitoring

Profiling and monitoring services are mostly exposed through following services

- Heartbeat,
- Monitoring,
- Reporting
- The Heart Beat makes the information that is collected through the PAL available
- The Monitoring and Reporting implement a generic infrastructure for monitoring the activity of any service in the Aneka Cloud;

Heartbeat Functions in detail

- The Heartbeat Service periodically collects the dynamic performance information about the node and publishes this information to the membership service in the Aneka Cloud
- It collects basic information about memory, disk space, CPU, and operating system
- These data are collected by the index node of the Cloud, and makes them available for reservations and scheduling services that optimizes them for heterogeneous infrastructure
- Heartbeat works with a specific component, called Node Resolver, which is in charge of collecting these data

Reporting & Monitoring Functions in detail

- The Reporting Service manages the store for monitored data and makes them accessible to other services 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

Many Built-in services use this channel to provide information, important built-inservices are:

- The Membership Catalogue service 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 obtains information about data transfer such as upload and download times, file names, and sizes.
- The Resource Provisioning Service tracks the provisioning and life time information of virtual nodes.

Resource management

Aneka provides a collection of services that are in charge of managing resources. These are

- Index Service (or Membership Catalogue)
- Resource Provisioning Service

Membership Catalogue features

- The Membership Catalogue is Aneka's fundamental component for resource management
- It keeps track of the basic node information for all the nodes that are connected or disconnected.
- It implements the basic services of a directory service, where services can be searched using attributes such as names and nodes

Resource Provisioning Service Features

- The resource provisioning infrastructure built into Aneka is mainly concentrated in the Resource Provisioning Service,
- It includes all the operations that are needed for provisioning virtual instances (Providing virtual instances as needed by users).
- The implementation of the service is based on the idea of resource pools.
- A resource pool abstracts the interaction with a specific IaaS provider by exposing a common interface so that all the pools can be managed uniformly.

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. These services cover:

- Storage management for applications
- Accounting, billing and resource pricing
- Resource reservation

Storage management

Aneka offers two different facilities for storage management:

- A centralized file storage, which is mostly used for the execution of compute- intensive applications,
- A distributed file system, which is more suitable for the execution of data-intensive applications.
- As the requirements for the two types of applications are rather different. Compute-intensive applications mostly require powerful processors and do not have high demands in terms of storage, which in many cases is used to store small files that are easily transferred from one node to another. Here, a centralized storage node is sufficient to store data
- Centralized storage is implemented through and managed by Aneka's Storage Service.
- The protocols for implementing centralized storage management are supported by a concept of File channel. It consists of a File Channel controller and File channel handler. File channel controller is a server component whereas File channel handler is a client component (that allows browsing, downloading and uploading of files).
- In contrast, data-intensive applications are characterized by large data files (gigabytes or terabytes, peta bytes) and here processing power required by tasks is not more.
- Here instead of centralized data storage a distributed file system is used for storing data by using all the nodes belonging to the cloud.
- Data intensive applications are implemented by means of a distributed file system.Google File system is best example for distributed file systems.
- Typical characteristics of Google File system are:
    - Files are huge by traditional standards (multi-gigabytes).
    - Files are modified by appending new data rather than rewriting existing data.
    - There are two kinds of major workloads: large streaming reads and small random reads.
    - It is more important to have a sustained bandwidth than a low latency.

Accounting, billing, and resource pricing

- Accounting Services keep track of the status of applications in the Aneka Cloud
- The information collected for accounting is primarily related to infrastructure usage and application execution
- Billing is another important feature of accounting.
- Billing is important since Aneka is a multitenant cloud programming platform in which the execution of applications can involve provisioning additional resources from commercial providers.

- Aneka Billing Service provides detailed information about each user's usage of resources, with the associated costs.
- Resource pricing is associated with the price fixed for different types of resources/nodes that are provided for the subscribers. Powerful resources are priced high and less featured resources are priced low
- Two internal services used by accounting and billing are Accounting service and Reporting Service

Resource reservation

Resource Reservation supports the execution of distributed applications and allows for reserving resources for exclusive use by specific applications.

Two types of services are used to build resource reservation:

- The Resource Reservation
- The Allocation Service
- Resource Reservation keeps track of all the reserved time slots in the Aneka Cloud and provides a unified view of the system. (provides overview of the system)
- The Allocation Service is installed on each node that features execution services and manages the database of information regarding the allocated slotson the local node.

Different Reservation Service Implementations supported by Aneka Cloud are:

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.

Libra Reservation: Represents a variation of the previous implementation that features the ability to price nodes differently according to their hardware capabilities.

Relay Reservation: This implementation is useful in integration scenarios in which Aneka operates in an inter cloud 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 distributedapplications on top of Aneka

Two types of services are:

1. The Scheduling Service :

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. 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

2. The Execution Service

Execution Services control the execution of single jobs that compose applications. They are in charge of setting up the runtime environment hosting the execution of jobs.

Some of the common operations that apply across all the range of supported models are:

• 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

The various Programming Models Supported by Execution Services of Aneka Cloud are:

1. Task Model. This model provides the support for the independent "bag of tasks" applications and many computing tasks. In this model application is modelled 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. Map Reduce Model. This is an implementation of Map Reduce as proposed by Google on top of Aneka.
4. Parameter Sweep Model. This model is a specialized form of Task Model for applications that can be described by a template task whose instances are created by generating different combinations of parameters.

Building Aneka clouds

Aneka Cloud can be realized by two methods:

1. Infrastructure Organization
2. Logical Organization

Infrastructure based organization of Aneka Cloud is given in the following figure-5.3:

The working mechanism of this model:

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

- The Management Console manages multiple repositories and select the one that best suits the specific deployment

- A **Repository** provides storage for all the libraries required to layout and install the basic Aneka platform, by installing images of the required software in particular Aneka Container through node managers by using various protocols like FTP, HTTP etc.
- A number of node managers and Aneka containers are deployed across the cloud platform to provision necessary services, The Aneka node manager are also known as AnekaDaemon
- The collection of resulting containers identifies the final AnekaCloud

Logical organization

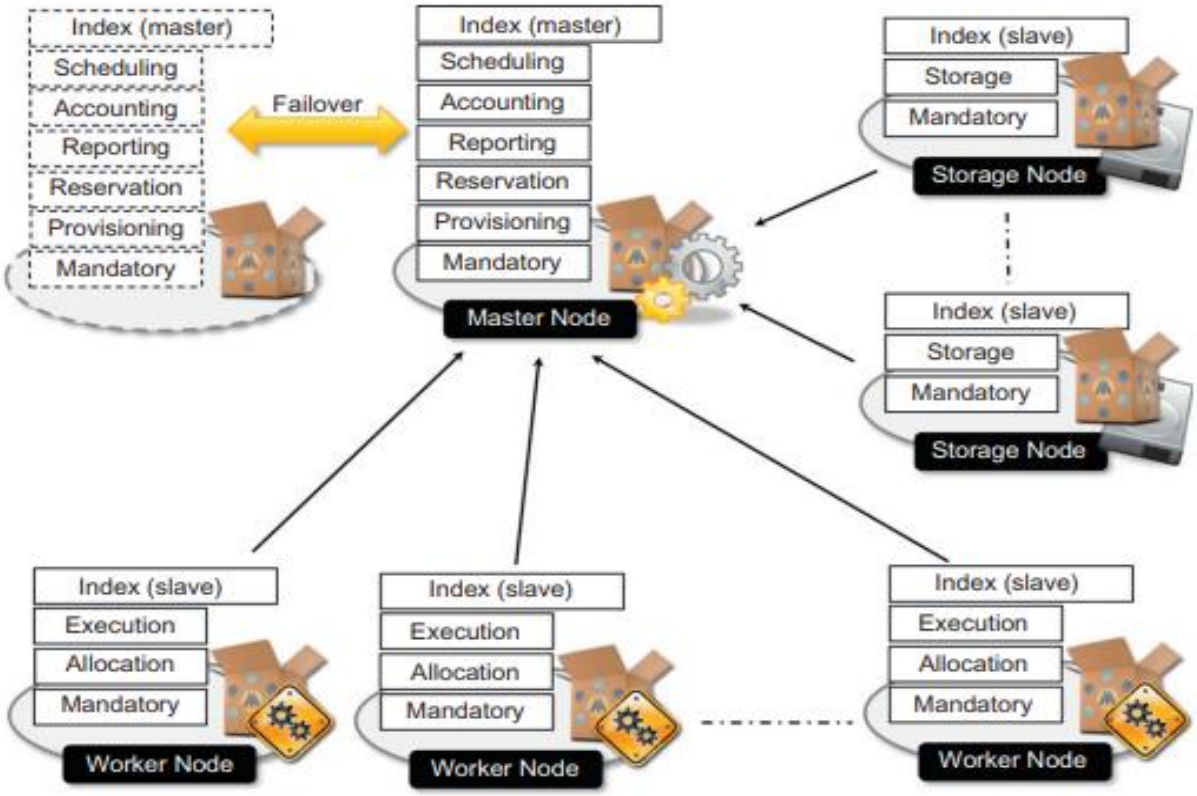
The logical organization of Aneka Clouds can be very diverse, since it strongly depends on the configuration selected for each of the container instances belonging to the Cloud.

Here is a scenario that has master-worker configuration with separate nodes for storage, the Figure 5.4. Portray

The master node comprises of following services:

- Index Service (master copy)
- Heartbeat Service
- Logging Service
- Reservation Service
- Resource Provisioning Service
- Accounting Service
- Reporting and Monitoring Service
- Scheduling Services for the supported programming models

Here Logging service and Heartbeat service and Monitoring service are considered as Mandatory services in all the block diagrams whereas other services are shown ditto.



Similarly, the Worker Node comprises of following services:

- Index Service
- Execution service
- Allocation service
- And mandatory (Logging, Heartbeat and monitoring services)

The Storage Node comprises of:

- Index service
- Storage Service
- And mandatory (Logging, Heartbeat and monitoring services)

In addition, all nodes are registered with the master node and transparently refer to any failover partner in the case of a high-availability configuration

Aneka Cloud Deployment Models

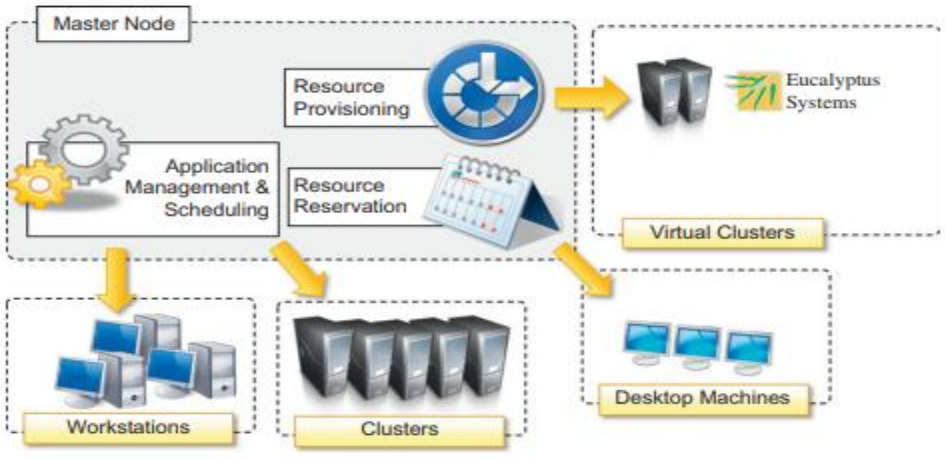
All the general cloud deployment models like Private cloud deployment mode, Public cloud deployment mode and Hybrid Cloud deployment mode are applicable to Aneka Clouds also.

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. This deployment is acceptable for a scenario in which the workload of the system is predictable and a local virtual machine manager can easily address excess capacity demand. Most of the Aneka nodes are constituted of physical nodes with a long lifetime and a static configuration and generally do not need to be reconfigured often. The different nature of the machines harnessed in a private environment allows for specific policies on resource management and usage that can be

accomplished by means of the Reservation Service. For example, desktop machines that are used during the day for office automation can be exploited outside the standard working hours to execute distributed applications. Workstation clusters might have some specific legacy software that is required for supporting the execution of applications and should be executed with special requirements



Note: In the master node: Resource Provisioning, Application Management & Scheduling and Resource Reservation are the primary services.

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.

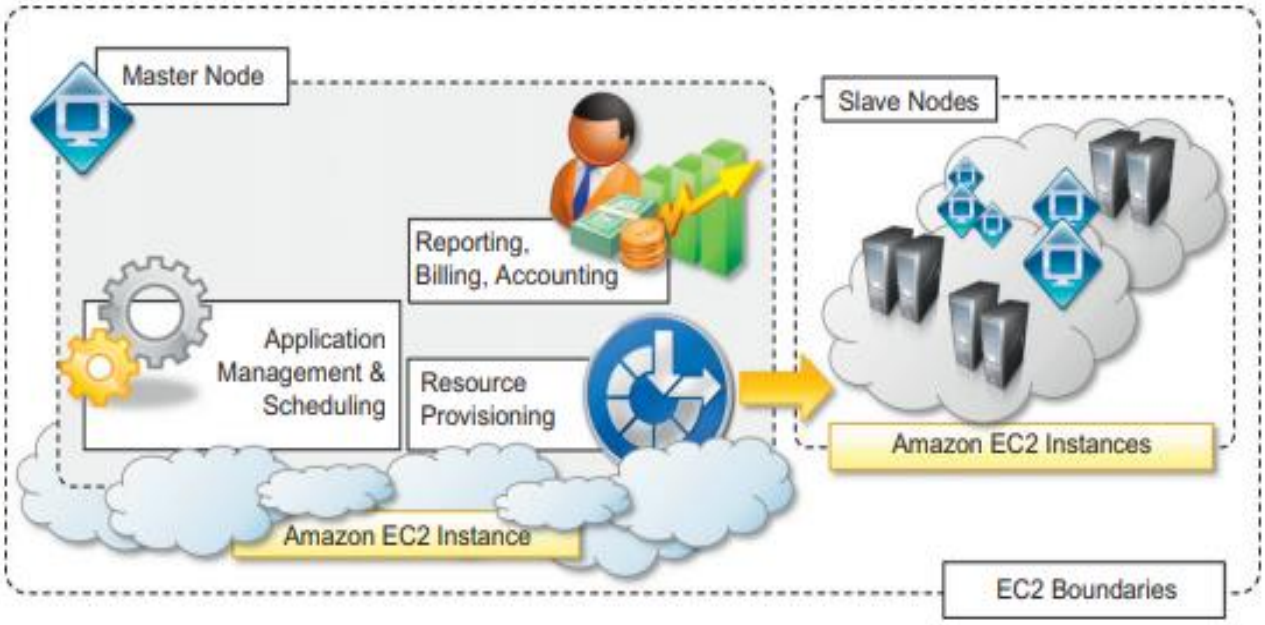
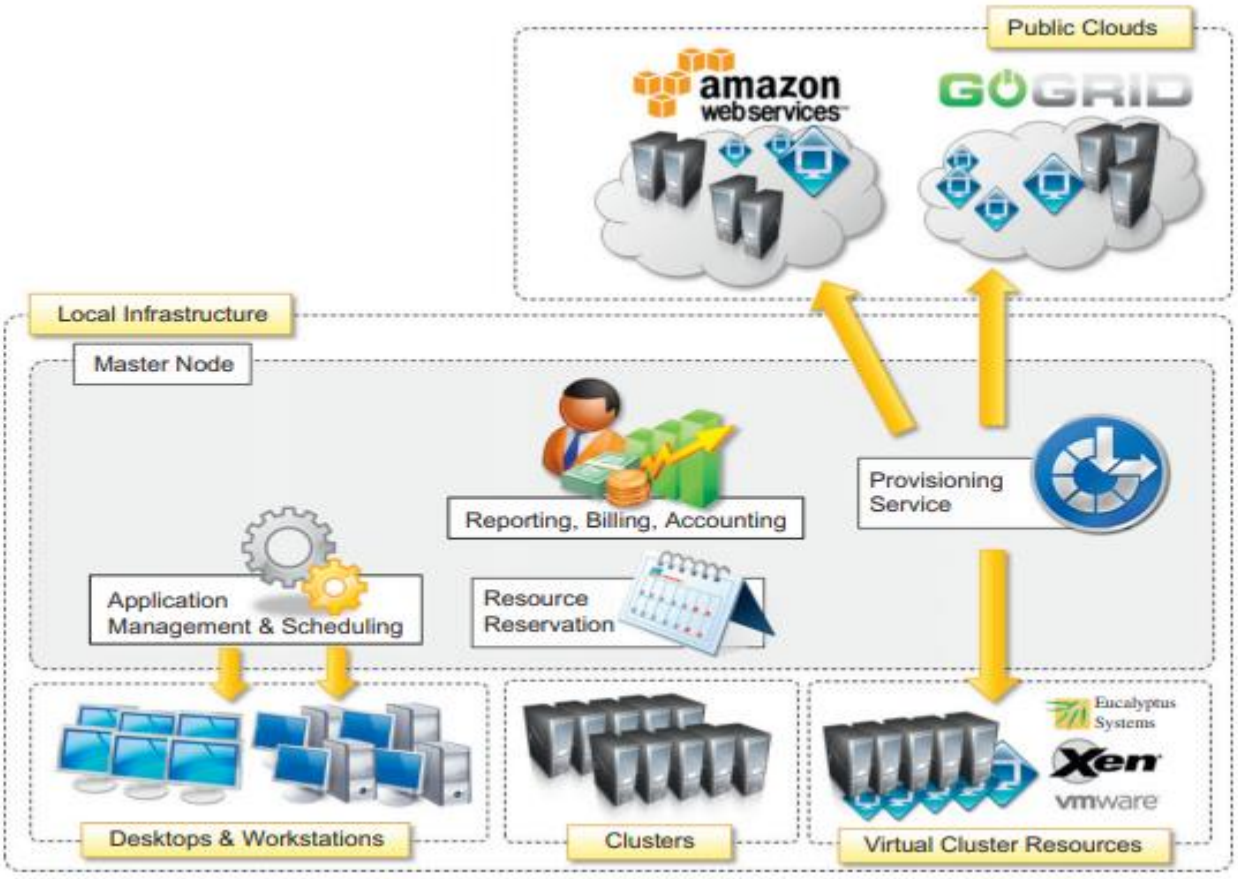


Figure 5.6 provides an overview of this scenario. The deployment is generally contained within the infrastructure boundaries of a single IaaS provider. The reasons for this are to minimize the data transfer between different providers, which is generally priced at a higher cost, and to have better network performance. In this scenario it is possible to deploy an Aneka Cloud composed of only one node and to completely leverage dynamic provisioning to elastically scale the infrastructure on demand. A fundamental role is played by the Resource Provisioning Service, which can be configured with different images and templates to instantiate. Other important services that have to be included in the master node are the Accounting and Reporting Services. These provide details about resource utilization by users and applications and are fundamental in a multitenant Cloud where users are billed according to their consumption of Cloud capabilities.

Note: Reporting, Billing, Accounting, Resource Provisioning and Application Management & Scheduling are the primary services in master node

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



This scenario constitutes the mostcomplete deployment for Aneka that is able to leverage all the capabilities of the framework:

- Dynamic Resource Provisioning
- Resource Reservation
- Workload Partitioning (Scheduling)
- Accounting, Monitoring, and Reporting

In a hybrid scenario, heterogeneous resources can be used for different purposes. As we discussed in the case of a private cloud deployment, desktop machines can be reserved for low priority work- load outside the common working hours. The majority of the applications will be executed on work- stations and clusters, which are the nodes that are constantly connected to the Aneka Cloud. Any additional computing capability demand can be primarily addressed by the local virtualization facili- ties, and if more computing power is required, it is possible to leverageexternal IaaS providers.

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.

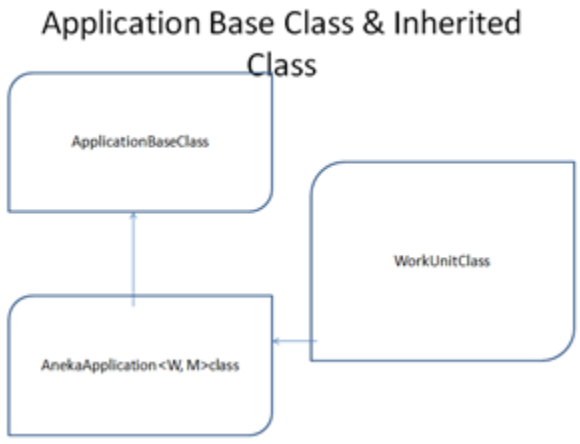
The SDK provides support for both programming models and services by

- The Application Model
- The Service Model.

Application Model

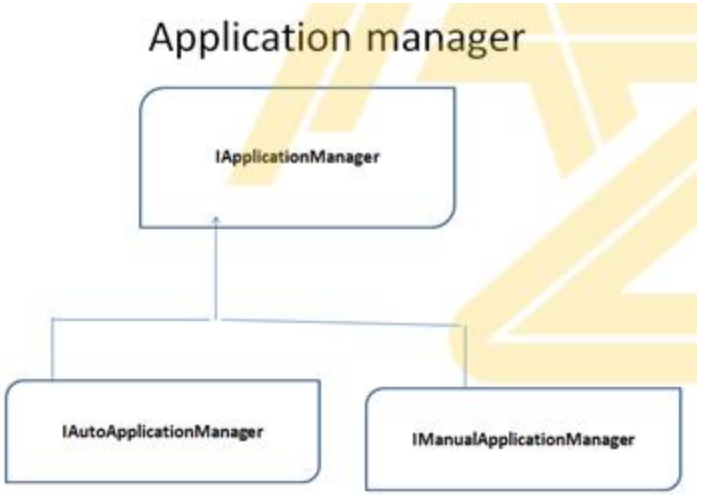
- The Application Model covers the development of applications and new programming models
- It Consists of Application Class & Application Manager
- Application Class – Provide user/developer view about distributed applications of the Aneka cloud
- Application Manager – Are Internal components that control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

The Application Class can be represented by following class diagram



Note: All the Aneka Application<W,M> class where W stands for Worker and M stands for Manager is inherited from base class and all Manual services are represented by WorkUnitClass. In addition there are two other classes in Application Class representation viz: Configuration Class and Application Data Class

The Application manager is represented with following class diagram:



Also, the table given below summarizes Application Class, The programming models supported and work units assigned to them.

Table 5.1 Aneka's Application Model Features

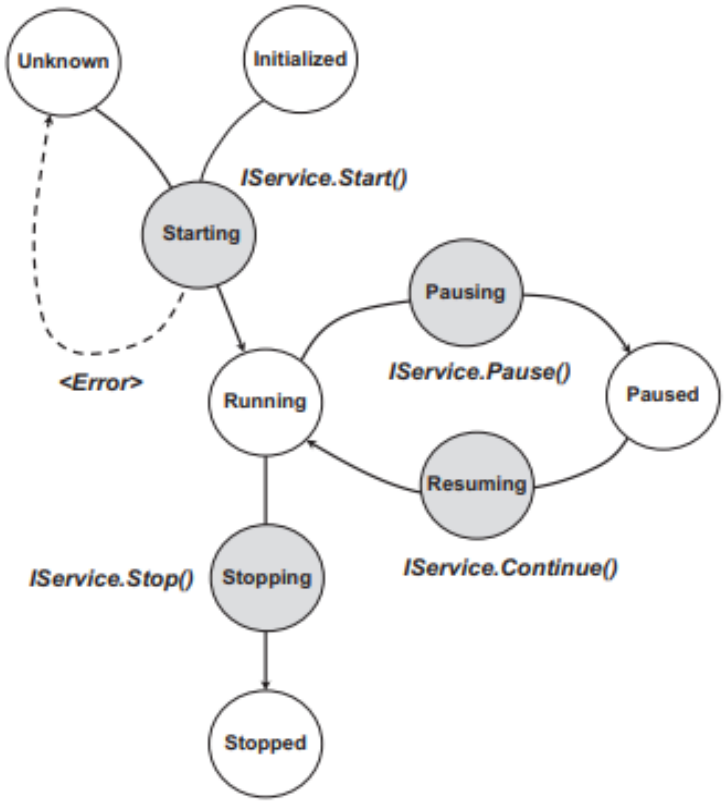
Category	Description	Base Application Type	Work Units?	Programming Models
Manual	Units of work are generated by the user and submitted through the application.	*AnekaApplication<W,M>* *IManualApplicationManager<W>* *ManualApplicationManager<W>*	Yes	Task Model Thread Model Parameter Sweep Model
Auto	Units of work are generated by the runtime infrastructure and managed internally.	*ApplicationBase<M>* *IAutoApplicationManager*	No	*MapReduce* Model

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

Figure the reference life cycle of each service instance in the Aneka container.
A service instance can initially be in the Unknown or Initialized state, a condition that refers to the creation of the service instance by invoking its constructor during the configuration of the container. Once the container is started, it will iteratively call the Start method on each service method. As a result, the service instance is expected to be in a Starting state until the startup process is completed, after which it will exhibit the Running state. This is the condition in which the service will last as long as the container is active and running. This is the only state in which the service is able to process messages. If an exception occurs while starting the service, it is expected that the service will fall back to the Unknown state, thus signalling an error. When a service is running it is possible to pause its activity by calling the Pause method and resume it by calling Continue.

As described in the figure, the service moves first into the Pausing state, thus reaching the Paused state. From this state, it moves into the Resuming state while restoring its activity to return to the Running state. Not all the services need to support the pause/continue operations, and the current implementation of the framework does not feature any service with these capabilities. When the container shutdown, the Stop method is iteratively called on each service running, and services move first into the transient Stopping state to reach the final Stopped state, whereall resources that were initially allocated have been released.



Note: Here all Unfilled Circles: Running, Unknown, Initialize, Paused and Stopped are Steady states. The filled Circles: Starting, Pausing, Resuming and Stopping are Transient States.

MANAGEMENT TOOLS

Aneka is a pure PaaS implementation and requires virtual or physical hardware to be deployed. Aneka's management layer, also includes capabilities for managing services and applications running in the Aneka Cloud.

Infrastructure management

Aneka leverages virtual and physical hardware in order to deploy Aneka Clouds. Virtualhardware is generally managed by means of the Resource Provisioning Service, which acquiresresources on demand according to the need of applications, while physical hardware is directlymanaged 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.