ADC501

Cloud Computing

Module 3

Cloud Computing Services and Business Value

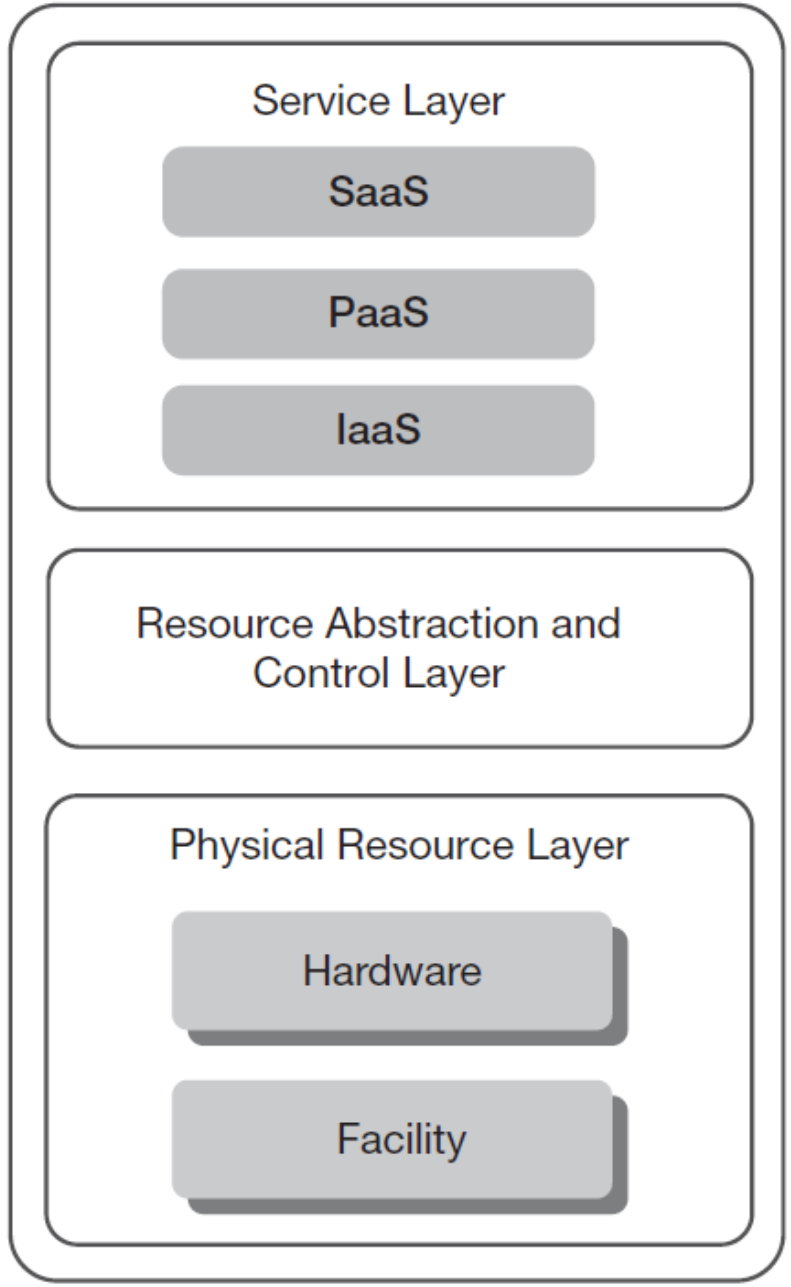


FIG 5.1: Service Layer as part of Cloud Service Orchestration[6]

Infrastructure-as-a-Service

- *Hardware-as-a-Service* (HaaS)
- underlying infrastructures such as compute, network, and storage are managed by the service provider
- delivers virtualized-hardware (not physical, but simulated software) resources to consumers known as virtual resources or virtual components
- Consumers no longer need to manage or control the underlying computing infrastructure
- the **simulated (or virtual) hardware component delivered as per consumers' requirement** is the uniqueness of IaaS model
- IaaS vendors generally offer custom made *virtual machines*
- **Amazon EC2** and Google Compute Engine
- **Amazon S3 is a popular storage service** available as IaaS

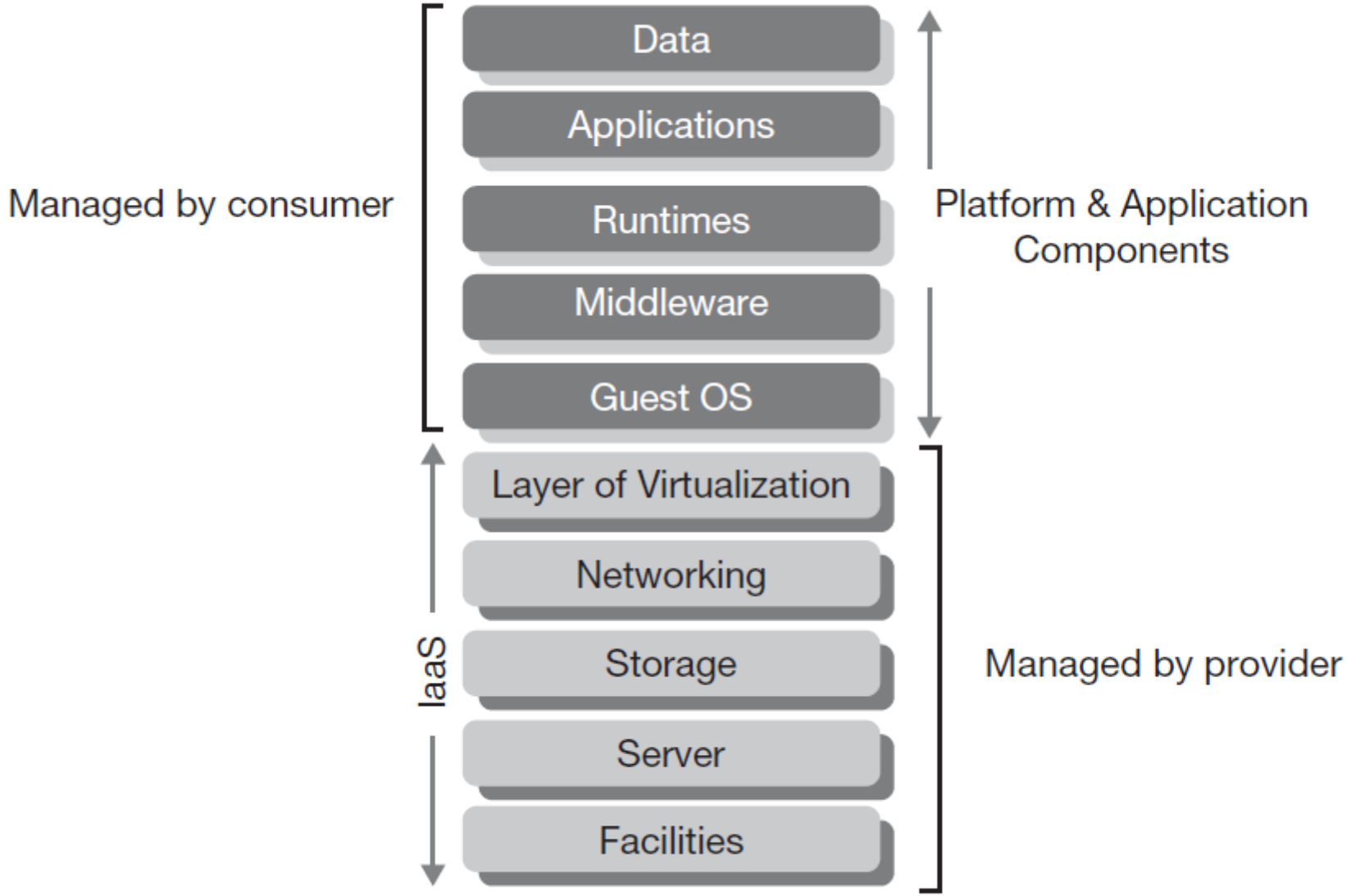


FIG 5.2: IaaS component stack

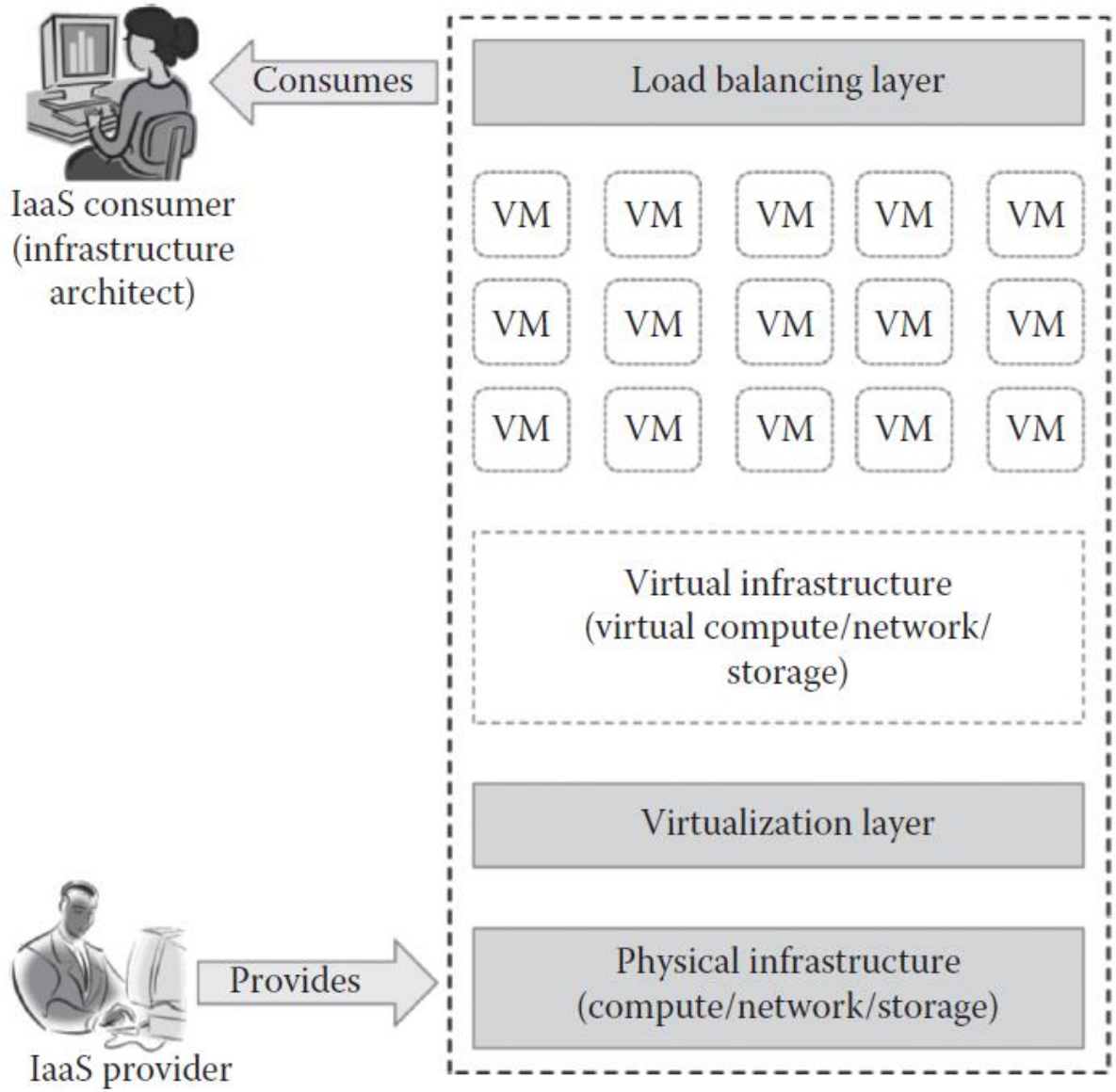


FIGURE 5.4
Overview of IaaS.

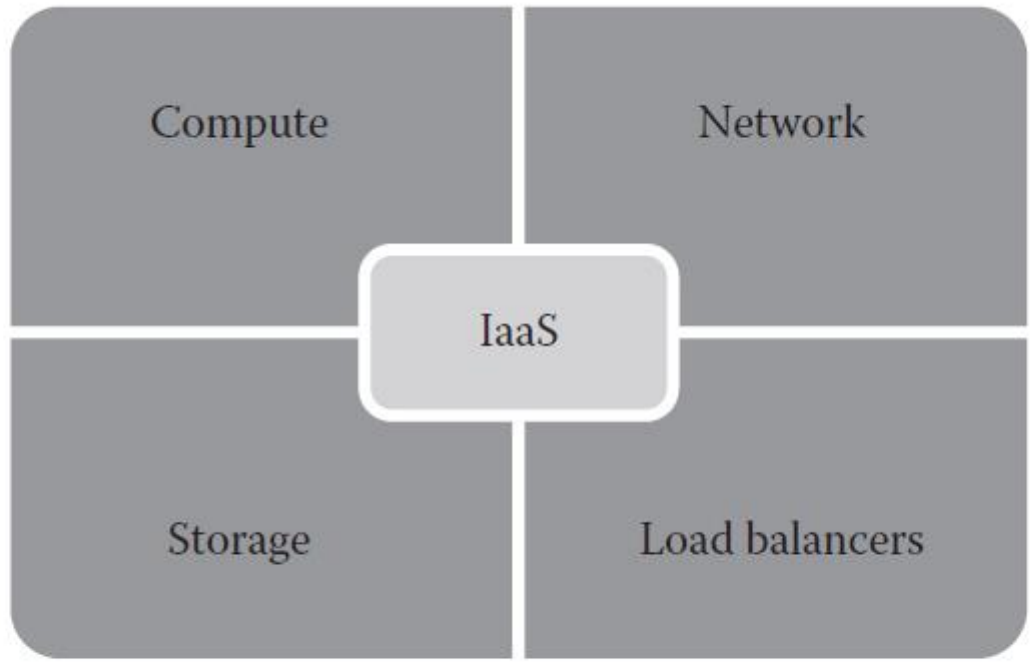


FIGURE 5.5
Services provided by IaaS providers.

Characteristics of IaaS

- Web access to the resources - IaaS model enables the IT users to access infrastructure resources over the Internet
- Centralized management - ensures effective resource management and effective resource utilization
- Elasticity and dynamic scaling
- Shared infrastructure - IaaS follows a one-to-many delivery model and allows multiple IT users to share the same physical infrastructure
- Preconfigured VMs - IaaS providers offer preconfigured VMs with operating systems (OSs), network configuration, etc
- Metered services - IaaS allows the IT users to rent the computing resources instead of buying it

Suitability of IaaS

- Unpredictable spikes in usage
- Limited capital investment
- Infrastructure on demand

IT users should avoid using the IaaS when

- When regulatory compliance does not allow off-premise hosting
- When usage is minimal
- When better performance is required - network latency
- When there is a need for more control on physical infrastructure

Benefits of IaaS

- Pay-as-you-use model - This model eliminates the unnecessary spending on buying hardware
- Reduced TCO - total cost of ownership
- Elastic resources
- Better resource utilization - increase the ROI (return on investment)
- Supports Green IT

Drawbacks of IaaS

- Security issues - Most of the IaaS providers are not able to provide 100% security to the VMs and the data stored on the VMs , attacks that target the hypervisors
- Interoperability issues - no common standards followed among the different IaaS providers
- Performance issues - distributed servers are connected over the network , Latency of the network

TABLE 5.1

Summary of Popular IaaS Providers

Provider	License	Deployment Model	Host OS	Guest OS	Supported Hypervisor(s)
Amazon Web Services	Proprietary	Public	Not available	Red Hat Linux, Windows Server, SuSE Linux, Ubuntu, Fedora, Debian, CentOS, Gentoo Linux, Oracle Linux, and FreeBSD	Xen
Google Compute Engine	Proprietary	Public	Not available	Debian 7 Wheezy, CentOS 6, Red Hat Enterprise Linux, SUSE, Windows Server, CoreOS, FreeBSD, and SELinux	KVM
Microsoft Windows Azure	Proprietary	Public	Not available	Windows Server, CentOS, FreeBSD, openSUSE Linux, and Oracle Enterprise Linux	Windows Azure hypervisor

Platform-as-a-Service

- platform means the underlying system on which software applications can be installed (and also developed)
- A computing platform comprises hardware resources, operating system, middleware (if required) and runtime libraries
- Cloud service provider takes care of the lower level (infrastructure level) resource management and provisioning
- PaaS comes with IaaS capability integrated into it
- collaborative application development becomes easier where multiple users can work from different geographical locations
- reduces the *total cost of ownership* (TCO) as computing platform becomes available on rent basis
- PaaS model lets the users focus only on development and deployment of application without having the tension of arranging and managing the underlying hardware and software
- Microsoft Azure Platform, GoGrid Cloud Center, Force.com are very popular, AWS Elastic Beanstalk
- Open-source PaaS - *Cloud foundry* is one such which is developed by VMware

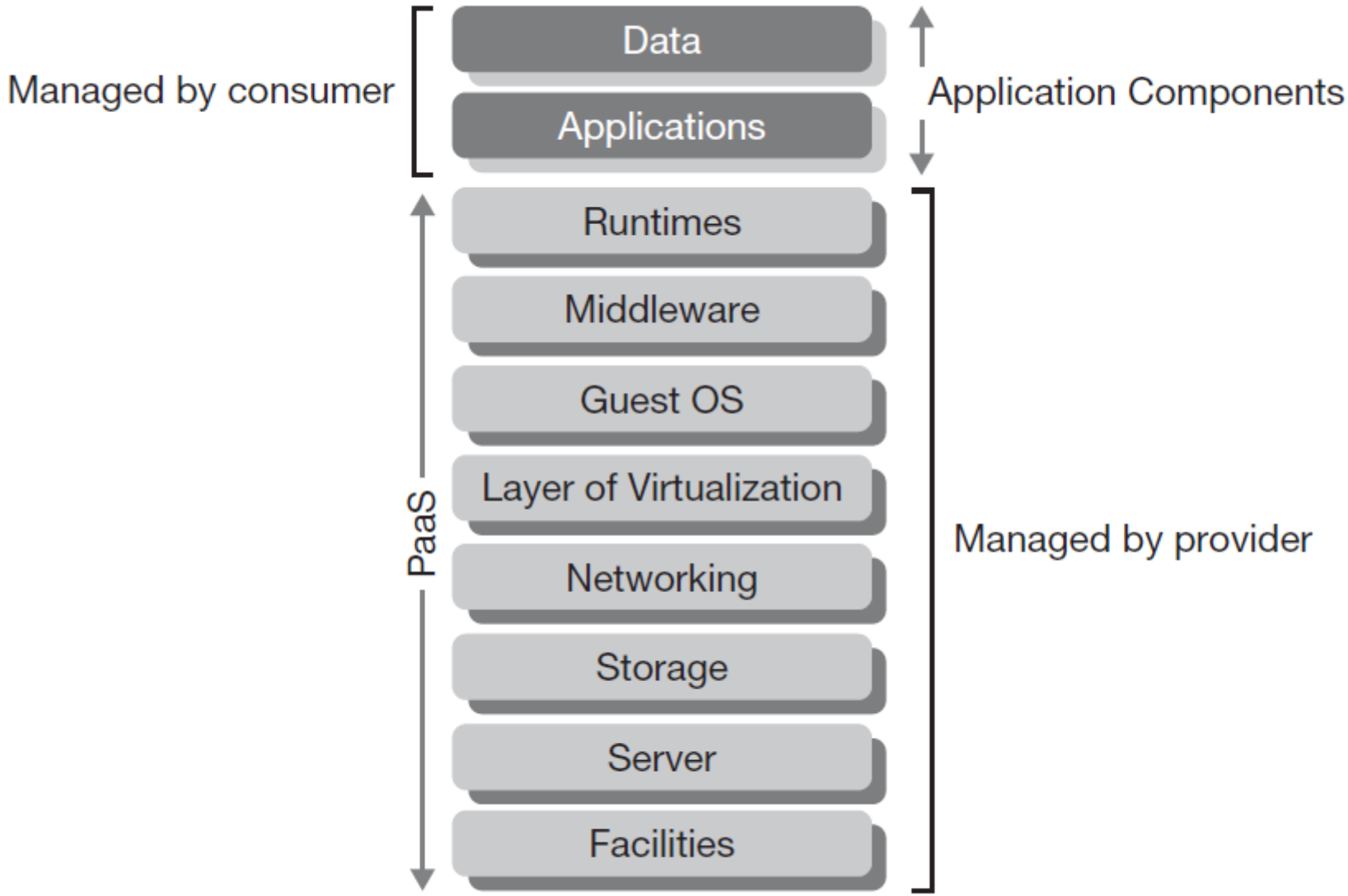
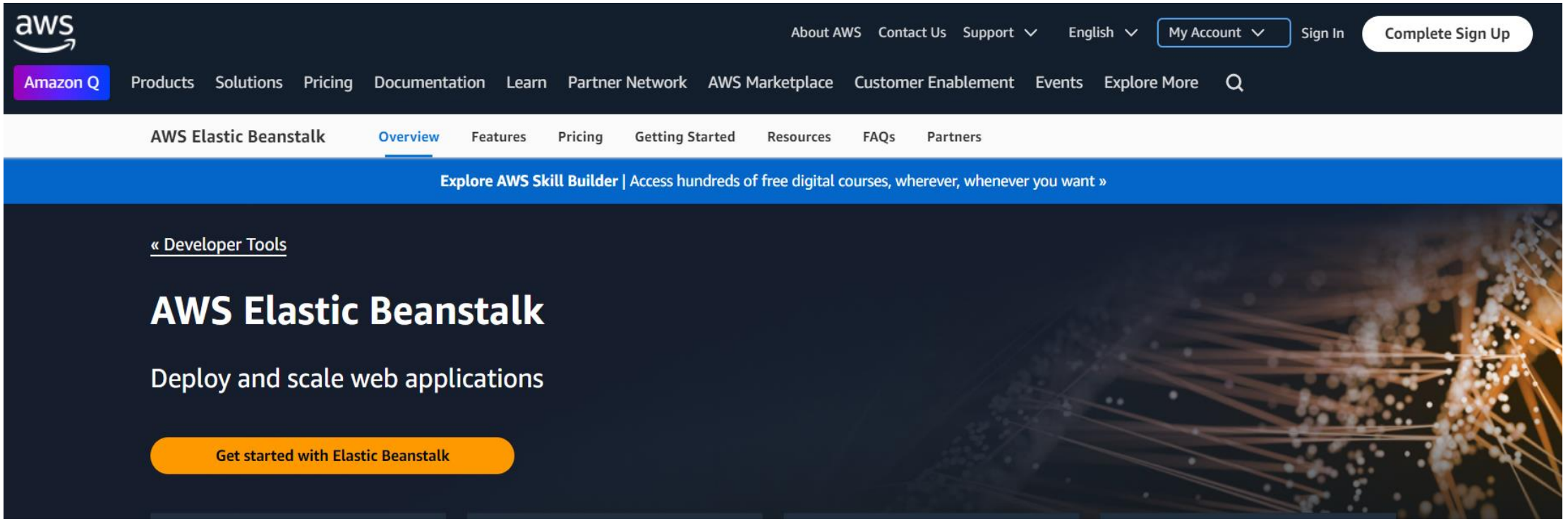


FIG 5.3: PaaS component stack

Platform-as-a-Service

- Amazon Web Services (AWS) comprises over one hundred services
- automatically handles the details of capacity provisioning, load balancing, scaling, and application health monitoring
- Elastic Beanstalk supports applications developed in Go, Java, .NET, Node.js, PHP, Python, and Ruby
- There is no additional charge for Elastic Beanstalk. You pay only for the underlying AWS resources that your application consumes



Platform-as-a-Service

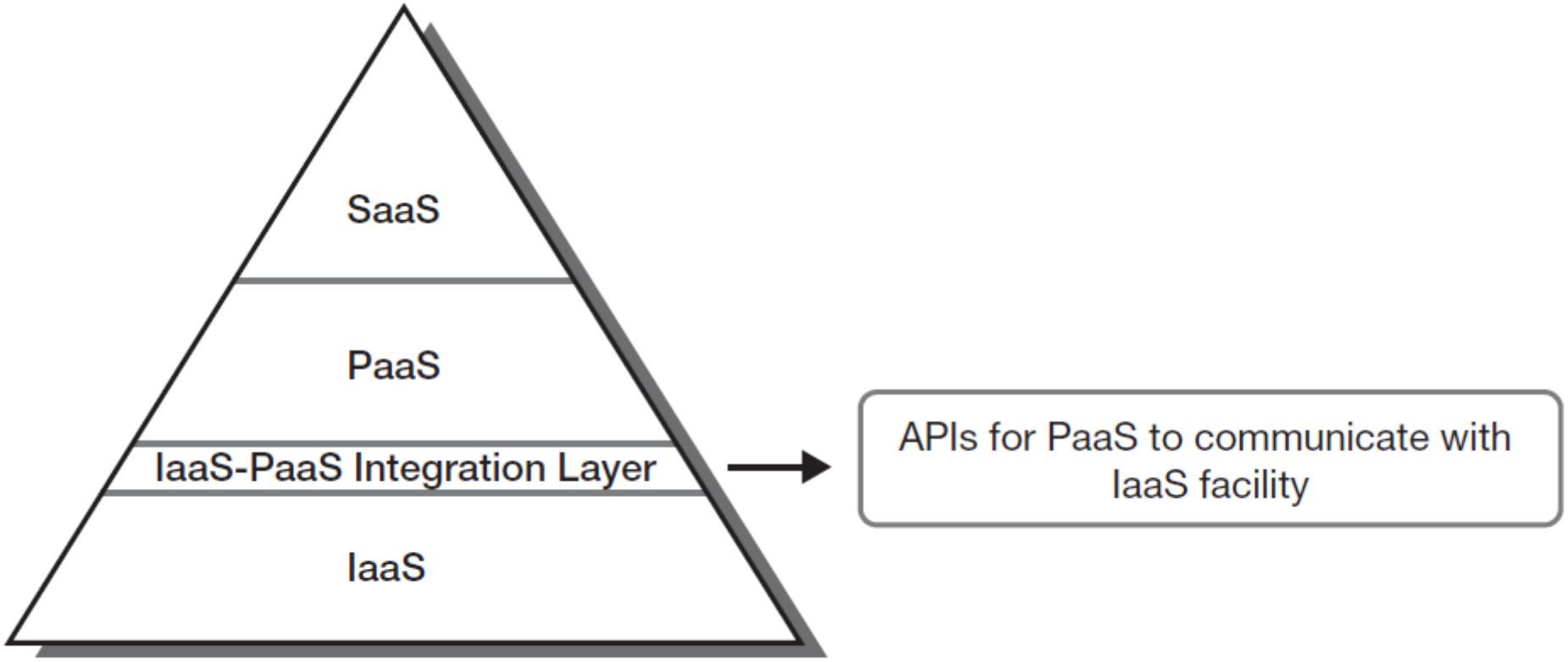


FIG 5.4: Integration of PaaS with IaaS

Platform-as-a-Service

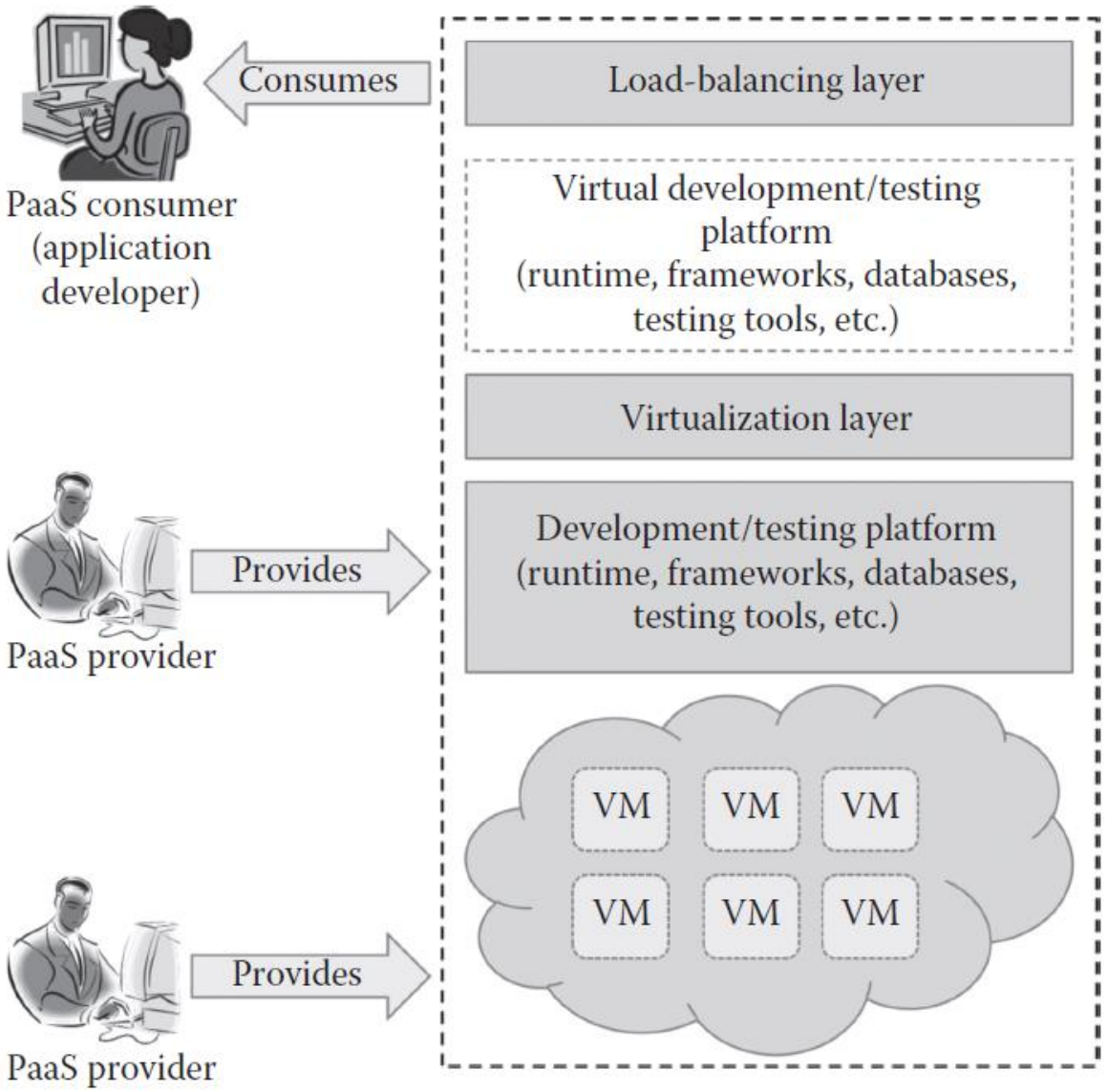


FIGURE 5.6
Overview of PaaS.

Platform-as-a-Service

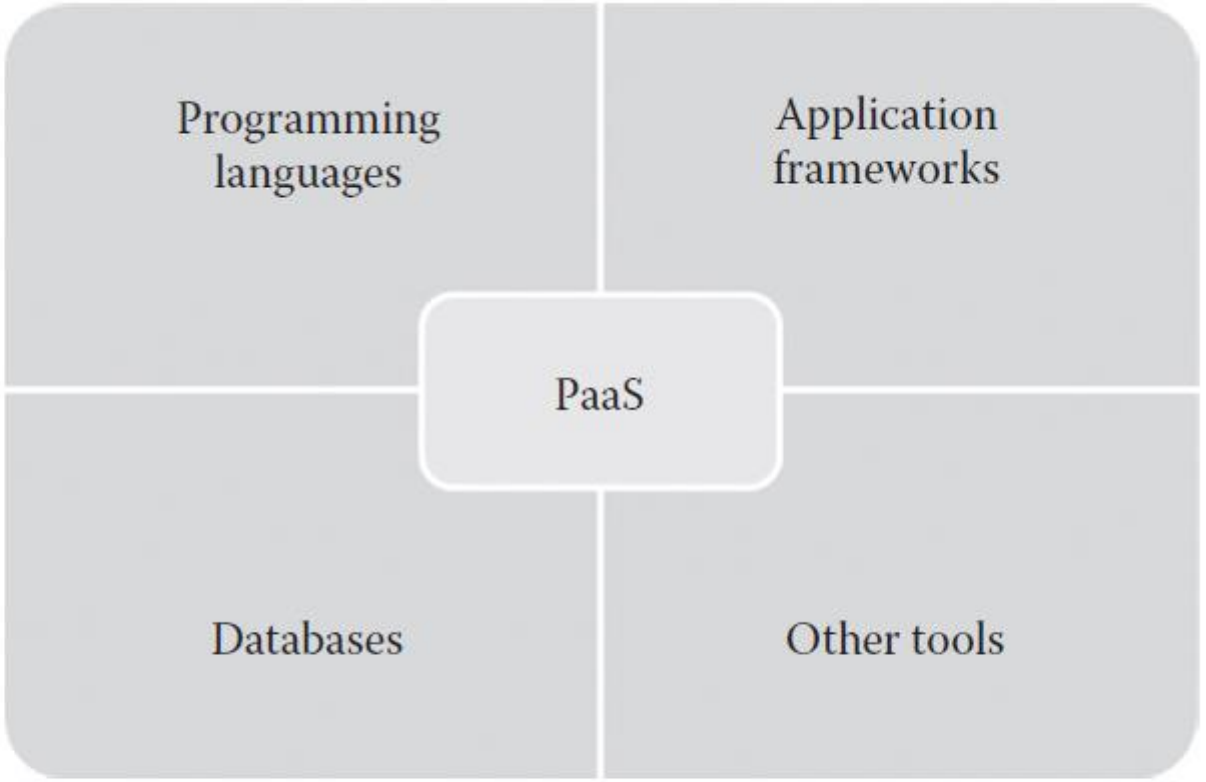


FIGURE 5.7
Services provided by PaaS providers.

Platform-as-a-Service

PaaS providers also provide build tools, deployment tools, and software load balancers as a service

- Programming languages - Java, Perl, PHP, Python, Ruby, Scala, Clojure, and Go etc.
- Application frameworks - simplify the application development - Node.js, Rails, Drupal, Joomla, WordPress, Django, EE6, Spring, Play, Sinatra, Rack, and Zend etc.
- Database - ClearDB, PostgreSQL, Cloudant, Membase, MongoDB, and Redis etc.
- Other tools: PaaS providers provide all the tools that are required to develop, test, and deploy an application

Characteristics of PaaS

- All in one - PaaS providers offer services to develop, test, deploy, host, and maintain applications in the same IDE
- Web access to the development platform
- Offline access - some of the PaaS providers allow the developer to synchronize their local IDE with the PaaS services, The developers can develop an application locally and deploy it online whenever they are connected to the Internet
- Built-in scalability - ensures that the application is capable of handling varying loads efficiently
- Collaborative platform - the development team consists of developers who are working from different places , PaaS providers provide tools for project planning and communication
- Diverse client tools - client tools include CLI, web CLI, web UI, REST API, and IDE etc

Suitability of PaaS

- Collaborative development - To increase the time to market and development efficiency, there is a need for a common place where the development team and other stakeholders of the application can collaborate with each other

- Automated testing and deployment

- Time to market

where PaaS may not be the best option

- Frequent application migration - The major problem with PaaS services are vendor lock-in

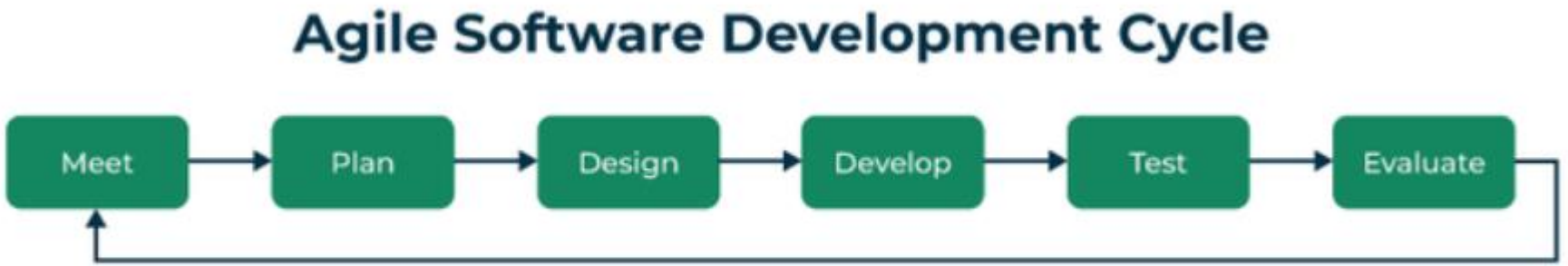
- Customization at the infrastructure level

- Flexibility at the platform level - PaaS provides template-based applications where all the different programming languages, databases, and message queues are predefined. It is an advantage if the application is a generic application.

- Integration with on-premise application

Pros and Cons of PaaS

- Quick development and deployment
- Reduces TCO
- Supports agile software development - PaaS services support agile methodologies
- Different teams can work together - online common development platform provided by PaaS providers
- Ease of use
- Less maintenance overhead
- Produces scalable applications



Agile software development cycle

Drawbacks of P-a-a-S

- Vendor lock-in
- Security issues
- Less flexibility - PaaS providers do not give much freedom for the developers to define their own application stack
- Depends on Internet connection

TABLE 5.2

Summary of Popular PaaS Providers

Provider	License	Deployment Model	Supported Languages	Supported Frameworks	Supported Databases	Client Tools
Cloud Foundry	Open source and proprietary	Public	Python, PHP, Java, Groovy, Scala, and Ruby	Spring, Grails, Play, Node.js, Lift, Rails, Sinatra, and Rack	MySQL, PostgreSQL, MongoDB, and Redis	cf. CLI, IDEs, and build tools
Google App Engine	Proprietary	Public	Python, Java, Groovy, JRuby, Scala, Clojure, Go, and PHP	Django, CherryPy, Pyramid, Flask, web2py, and webapp2.	Google Cloud SQL, Datastore, BigTable, and Blobstore	APIs
Heroku	Proprietary	Public	Ruby, Java, Scala, Clojure and Python, PHP, and Perl	Rails, Play, Django, and Node.js.	ClearDB, PostgreSQL, Cloudant, Membase, MongoDB, and Redis	CLI and RESTful API
Microsoft Windows Azure	Proprietary	Public	.Net, PHP, Python, Ruby, and Java	Django, Rails, Drupal, Joomla, WordPress, DotNetNuke, and Node.js.	SQL Azure, MySQL, MongoDB, and CouchDB	RESTful API and IDEs

Software-as-a-Service

- way of delivering application as a service over the network/Internet that users can directly consume without the tension of installing or configuring an application
- cheaper way of using application
- Users can access the applications through a *thin client interface* (usually a browser) from any location
- CRM (customer relationship management) package of Salesforce
- Microsoft 365 – online documentation service

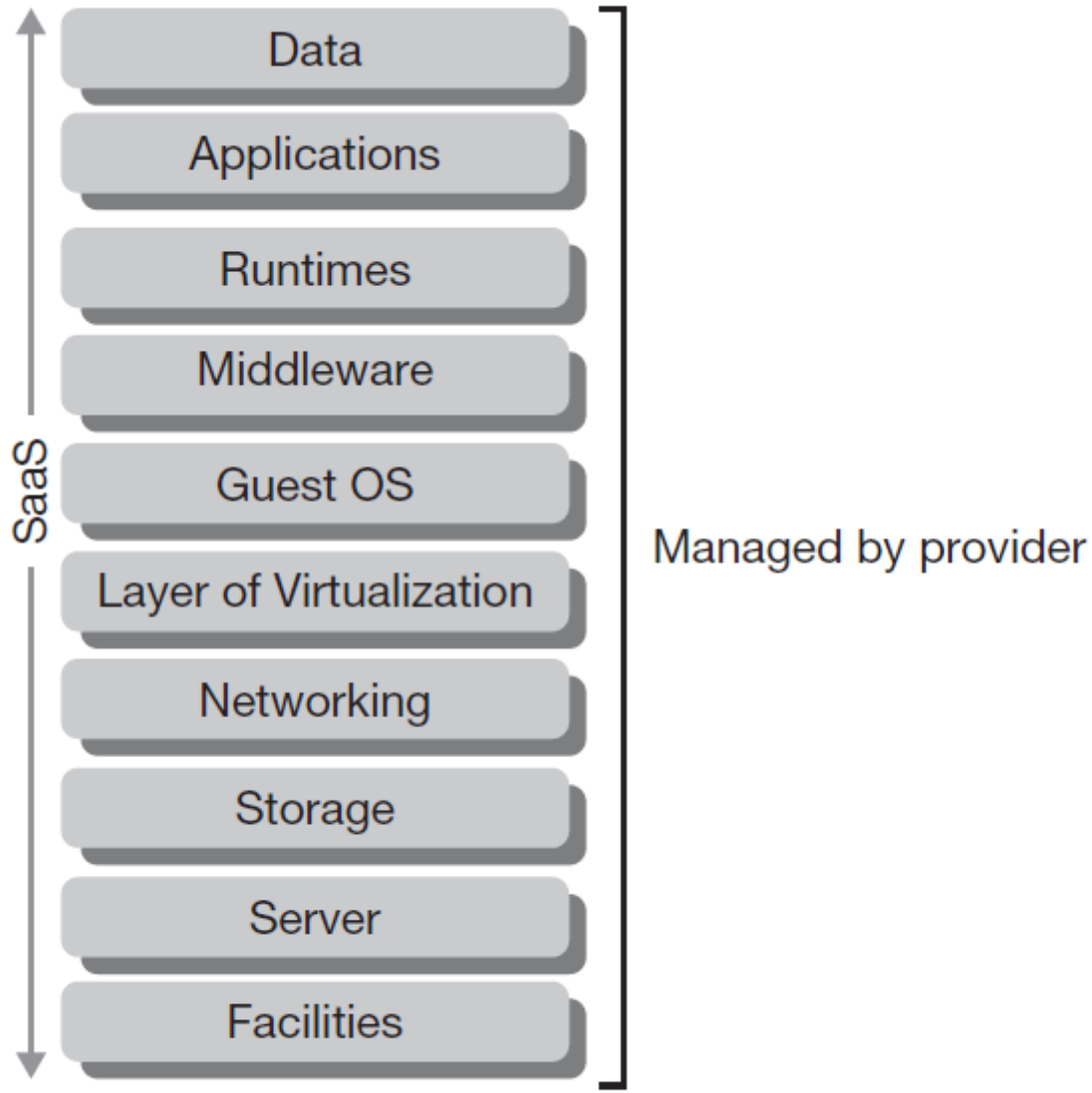


FIG 5.5: SaaS component stack

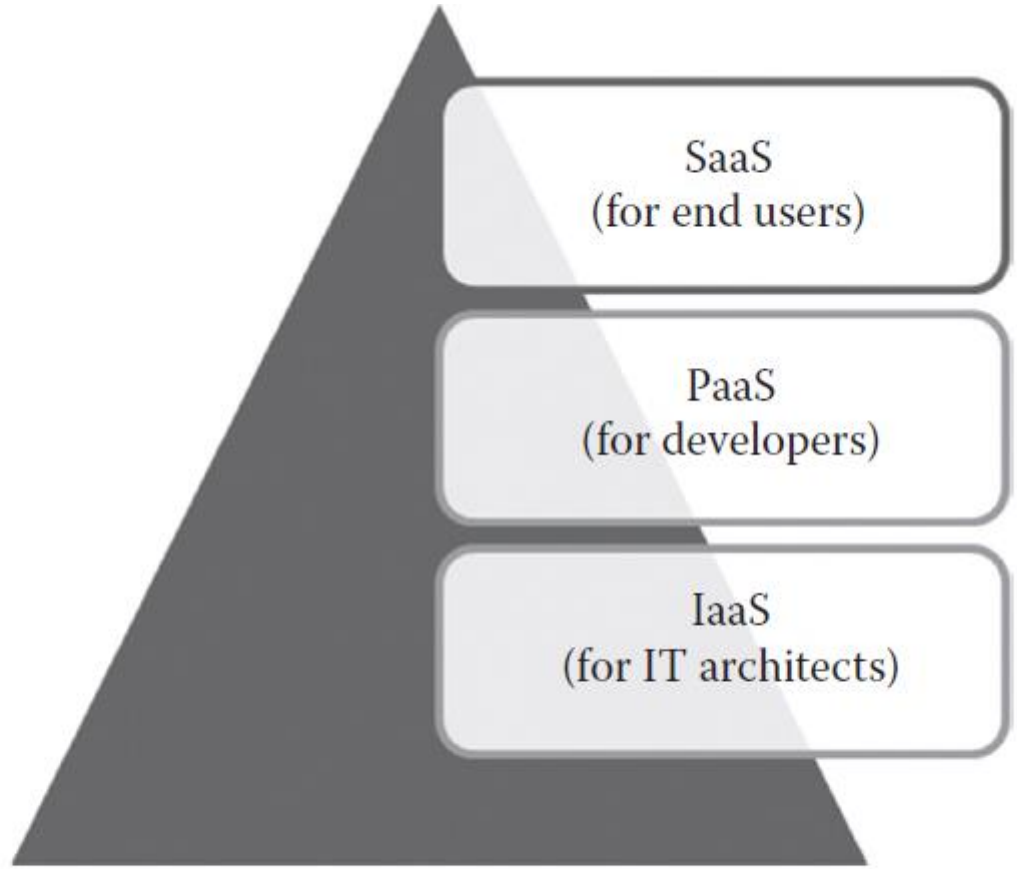


FIGURE 5.1
Basic cloud service models.

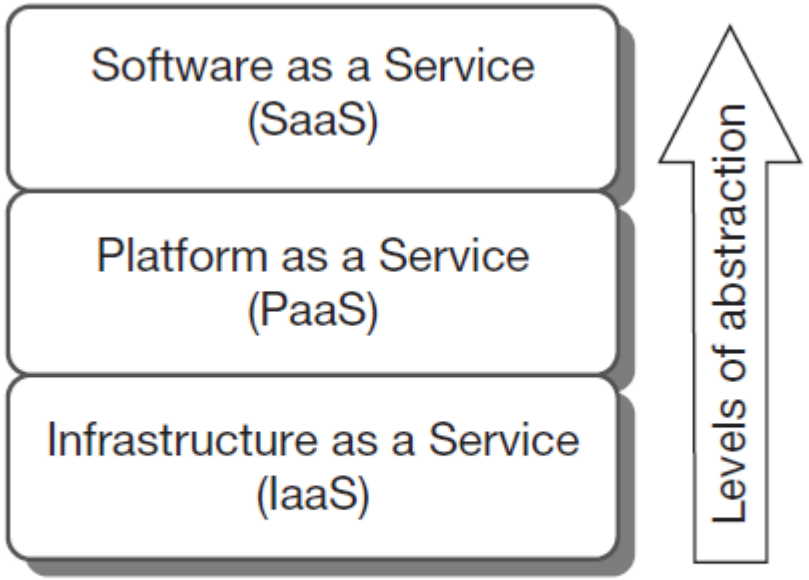


FIG 5.7: The levels of service abstraction

- Cloud computing abstracts the underlying computing system from the users. The degree of abstraction increases with the layers of services and abstraction is maximum at the SaaS level
- Any delivery model can exist in any deployment scenario and thus any delivery/deployment pairing is possible
- SaaS offerings are mostly public Services

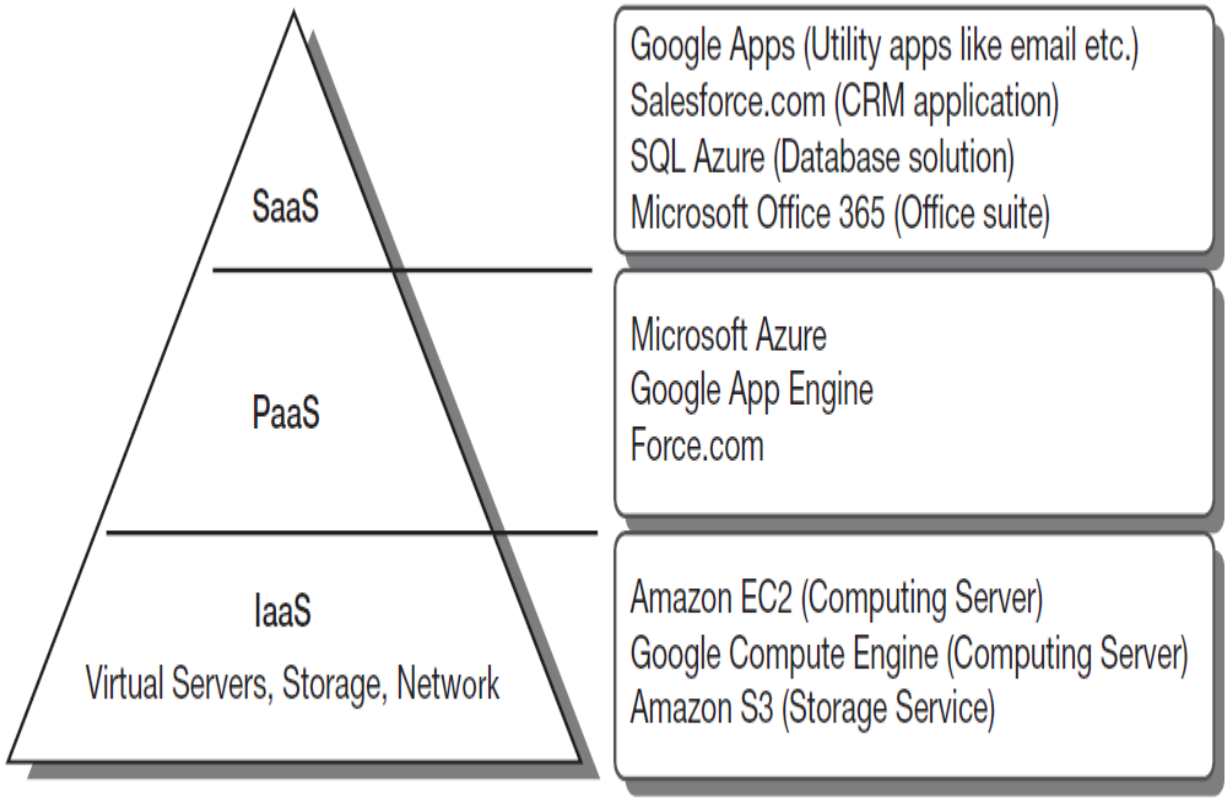


FIG 5.8: The layered cloud service model

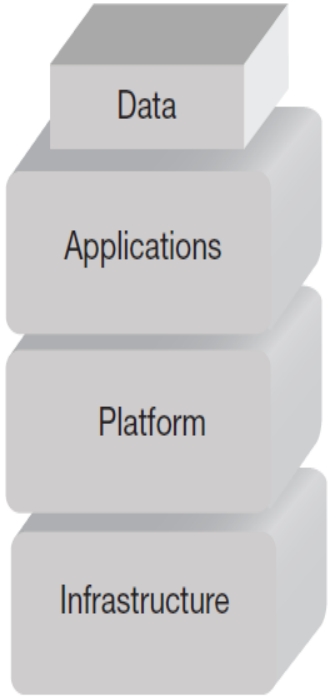


FIG 5.9: The working model of traditional computing system

- Service means abstraction of a lot of complex functionalities
- users' responsibility (of managing the environment) decreases as they move gradually from IaaS to SaaS
- rapid elasticity in the capability and on-demand self-service facility, it has to be a measured service

Benefits of SaaS over Traditional Applications

- **Licensing**
  Consumers need not to purchase any software license for fixed period

- **Risks of software acquisition**
  The resource, time, budget and expertise required for managing deployment of such applications are some of its critical task

- **Business focus**
  can focus more on high-value activities to support their own business goals

Table 14.4 Some SaaS examples

SaaS provider	Important services
A2Zapps.com (2010)	Marketing Automation, School Automation (ERP)
Envysion.com (2010)	Video Management
Learn.com (2010)	Training, HR, Online Courses
Microsoft (2010)	Office Live Meeting, Dynamics CRM, SharePoint
OpenID (2010)	Log in Identification
Zoho (2010)	Mail, Docs, Wiki, CRM, Meeting, Business

Among them, **Amazon is considered as pioneer in IaaS**,

the efforts of Google are mainly focused on SaaS and PaaS delivery and

later on, Microsoft chose the PaaS and SaaS as the core of their interests

Table 14.5 The Amazon Web services (prices from Feb 2010)

Amazon web service	Brief description	Geographical regions[1]	Pricing range
Amazon Elastic Compute Cloud (Amazon EC2)	"Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud" (Amazon EC2, 2010).	US – N. Virginia US – N. California EU – Ireland	$0.085 – $3.18 per hour (vary for different Instance and regions)
Amazon Simple Storage Service (Amazon S3)	"Amazon S3 provides a simple web services interface that can be used to store and retrieve any amount of data, at any time, from anywhere on the web" (Amazon S3, 2010).	US – N. Virginia US – N. California EU – Ireland	0.055 – 0.150 Per GB (vary for different regions)
Amazon SimpleDB (2010)	"Amazon SimpleDB is automatically indexing your data and providing a simple API for storage and access and requiring no schema" (Amazon SimpleDB, 2010).	US – N. Virginia US – N. California EU – Ireland	$0.140 – $0.154 per Hour (vary to different regions)
Amazon CloudFront (2010)	"Amazon CloudFront is a web service for content delivery. It delivers the static and streaming content using a global network of edge locations. By routing the requests for any objects to the nearest edge location, so content is delivered with the best possible performance. Amazon CloudFront works seamlessly with (Amazon S3)" (Amazon CloudFront, 2010).	United States Europe Hong Kong Japan	$0.050 – $0.221 per GB (vary for different Data transfer per month and regions)
Amazon Simple Queue Service (Amazon SQS) (2010)	"Amazon Simple Queue Service (Amazon SQS) offers a reliable, highly scalable, hosted queue for storing messages as they travel between computers. Amazon SQS makes it easy to build an automated workflow, working in close conjunction with the Amazon EC2" (Amazon SQS, 2010).	US – N. Virginia US – N. California EU – Ireland	$0.01 per 10,000 Amazon SQS Requests ($0.000001 per Request)

Table 14.5 (continued)

Amazon web service	Brief description	Geographical regions[1]	Pricing range
Amazon Elastic MapReduce (2010)	"Amazon Elastic MapReduce is a web service that enables businesses, researchers, data analysts, and developers to easily and cost-effectively process vast amounts of data. It utilizes a hosted Hadoop framework running on the web-scale infrastructure of Amazon EC2 and Amazon S3" (Amazon Elastic MapReduce, 2010).	US – N. Virginia US – N. California EU – Ireland	$0.015 – $0.42 per hour (vary for different instance and regions)
Amazon Relational Database Service (Amazon RDS) (2010)	"Amazon Relational Database Service (Amazon RDS) gives you access to the full capabilities of a familiar MySQL database. This means the code, applications, and tools you already use today with your existing MySQL databases work seamlessly with Amazon RDS" ((Amazon RDS, 2010).	US Region	$0.11 – $3.10* (vary for different instances)

[1] Amazon web services provide multiple regions and "availability zones" so customers can connect to the most convenient service, and also choose services in multiple zones to maximize failure-independence.

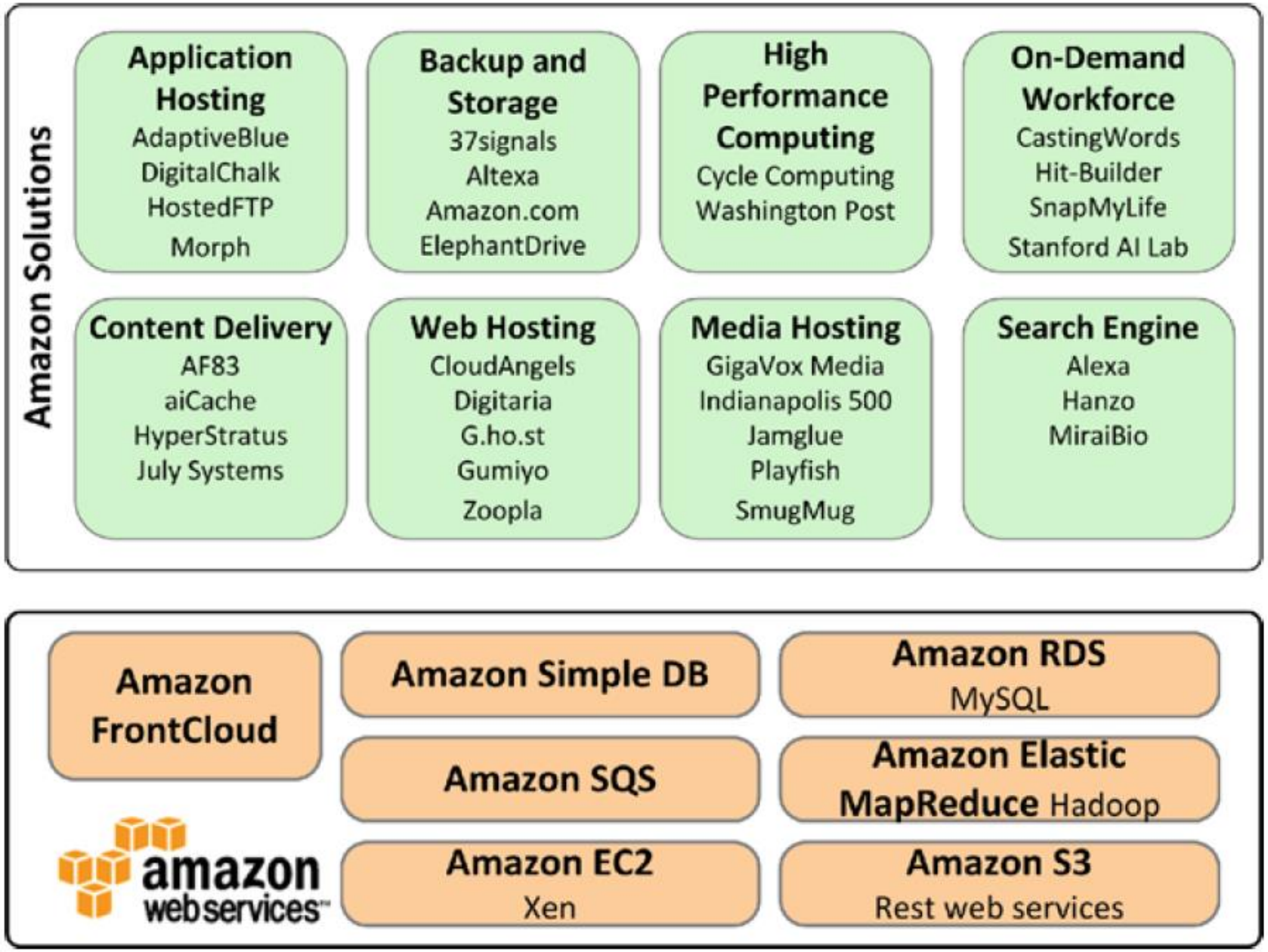


Fig. 14.7 The Amazon Family: Amazon Web services and their Different Solutions and customers (Amazon AWS, 2010)

History on Enterprise IT Services

- **Mainframes** – (60's decade) designed for handling and processing very large amounts of data quickly with high reliability, availability and serviceability,
  Mainframes scale vertically by adding more computing,storage or connectivity resources

- Decentralized applications - **client/server architectures**

Distributed systems present some advantages over mainframes

- Cost reduction
- Flexibility
- Latency reduction
- Interactive services
- Unlimited resource addition

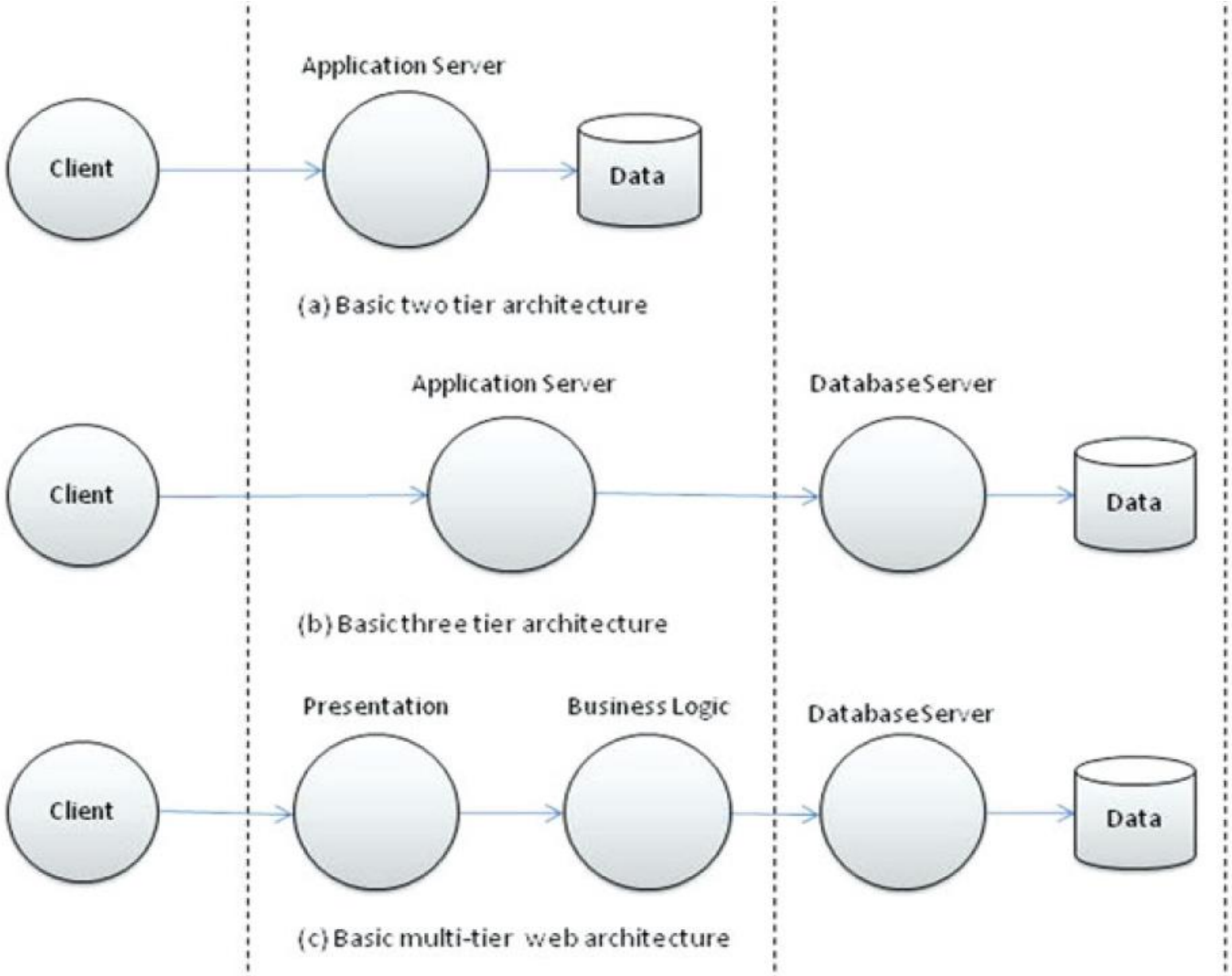


Fig. 15.1 Basic client/server architectures (Orfali et al., 1996)

But, during the 90's, distributed systems presented some **disadvantages** as compared to mainframes

- Low use of server nodes
- More operation costs
- Less energy efficiency
- More physical space required
- Potentially less performance with I/O
- Potentially more difficult to be fault tolerant
- Inability to share resources in distributed nodes

At the beginning of the 2000's these disadvantages made some IT managers reconsider coming back to renewed mainframes architectures which enabled to consolidate hardware resources while being able to run multiple OS instances - These architectures introduced virtualization technologies

Grid and Cloud technologies have emerged to allow resource sharing between organizations

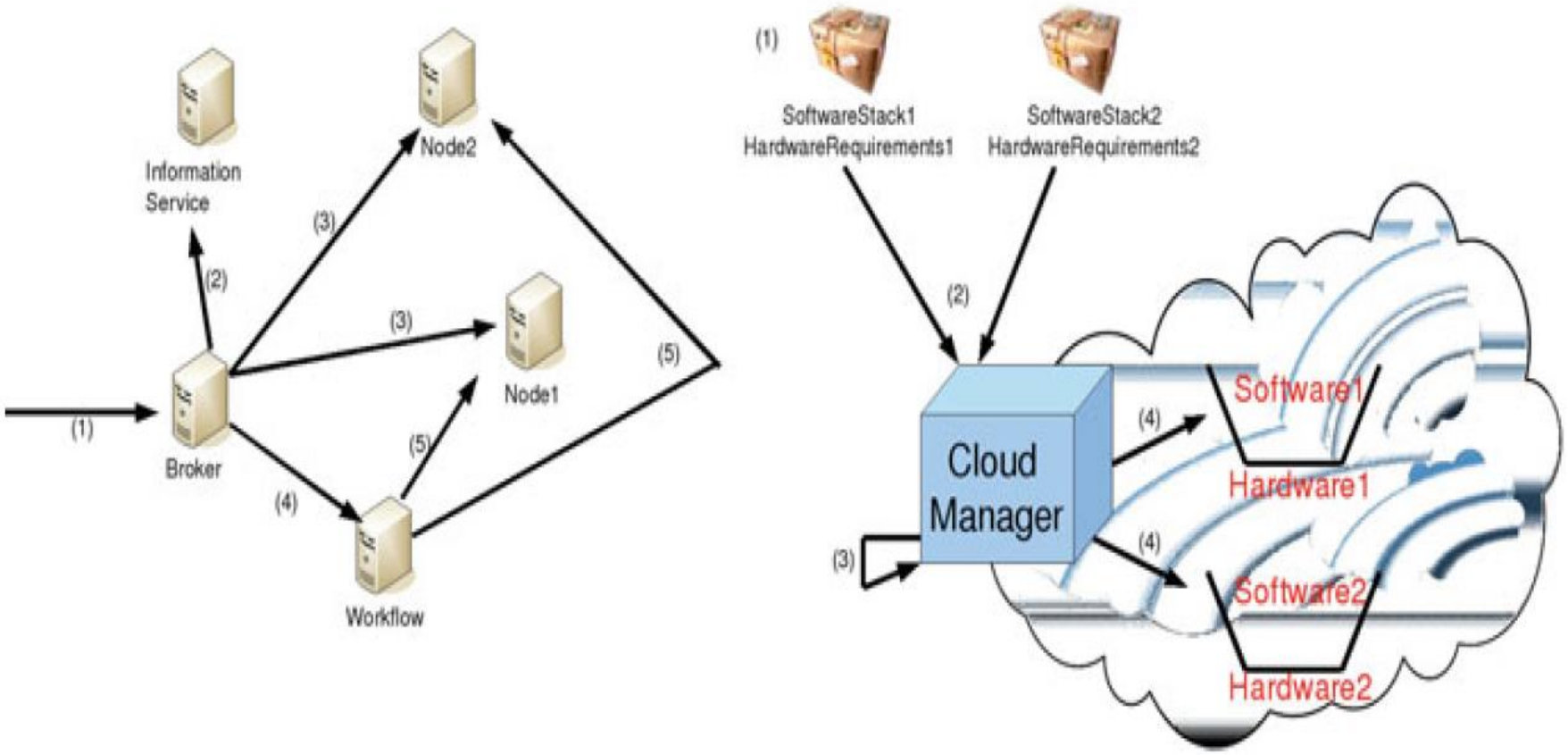
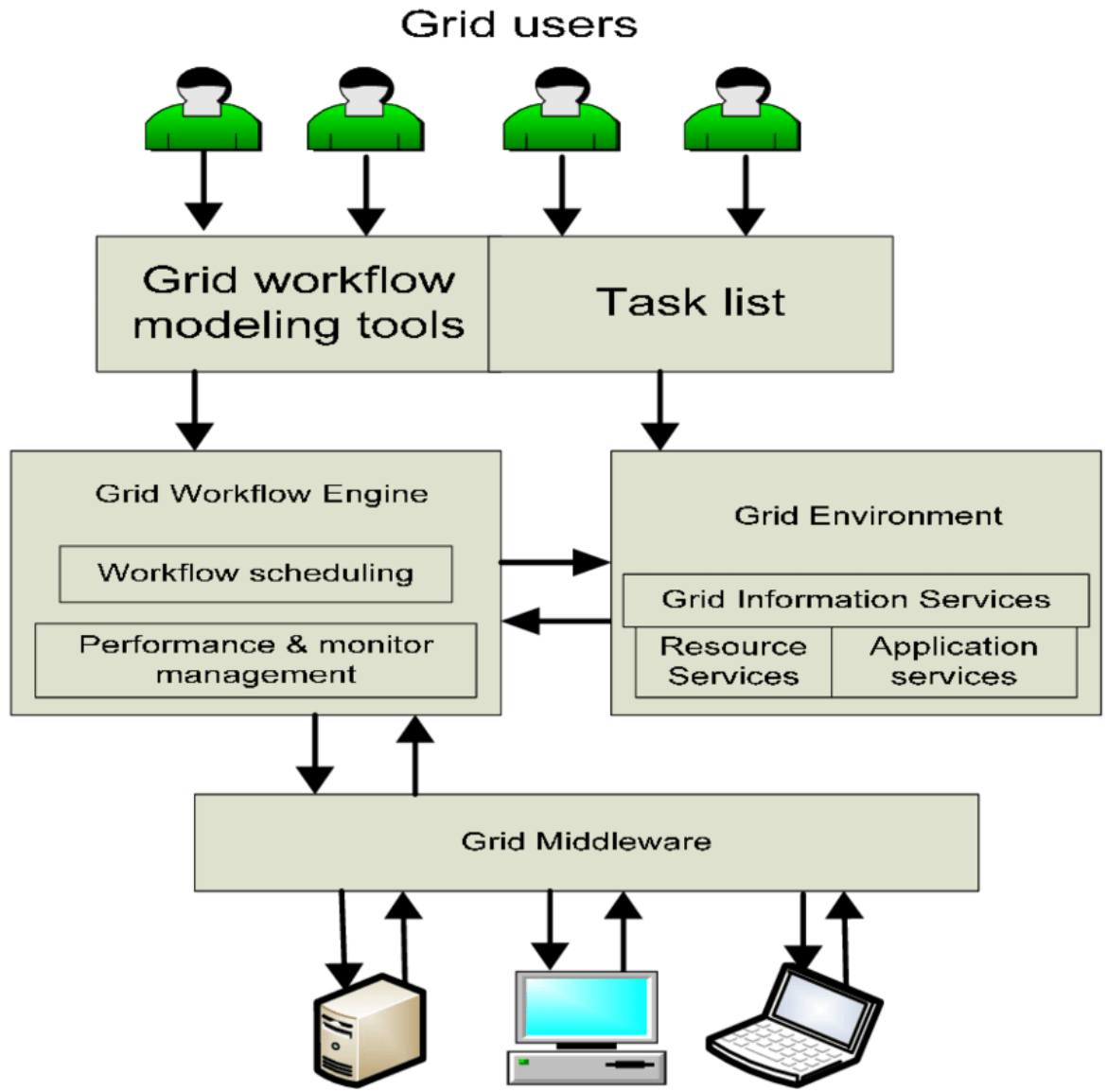


Fig. 15.3 Grid vs cloud provisioning models


Grid users
Grid workflow modeling tools
Task list
Grid Workflow Engine
Workflow scheduling
Performance & monitor management
Grid Environment
Grid Information Services
Resource Services
Application services
Grid Middleware

- Grid is defined as a “system that coordinates resources which are not subject to centralized control, using standard, open, general-purpose protocols and interfaces to deliver nontrivial qualities of service”

- the ability to combine resources from different organizations for a common goal

- no mechanisms for helping Grid Service developers to scale their systems in accordance with changes in demand

- an administrator should detect service overloads and manually scale the system

- There are three main types of nodes: control, provider, and user nodes

- Cloud, came into the scene to help increase the scalability provided to the end user.

- Grids scale at the vertical level - adding more re-sources to a virtual machine

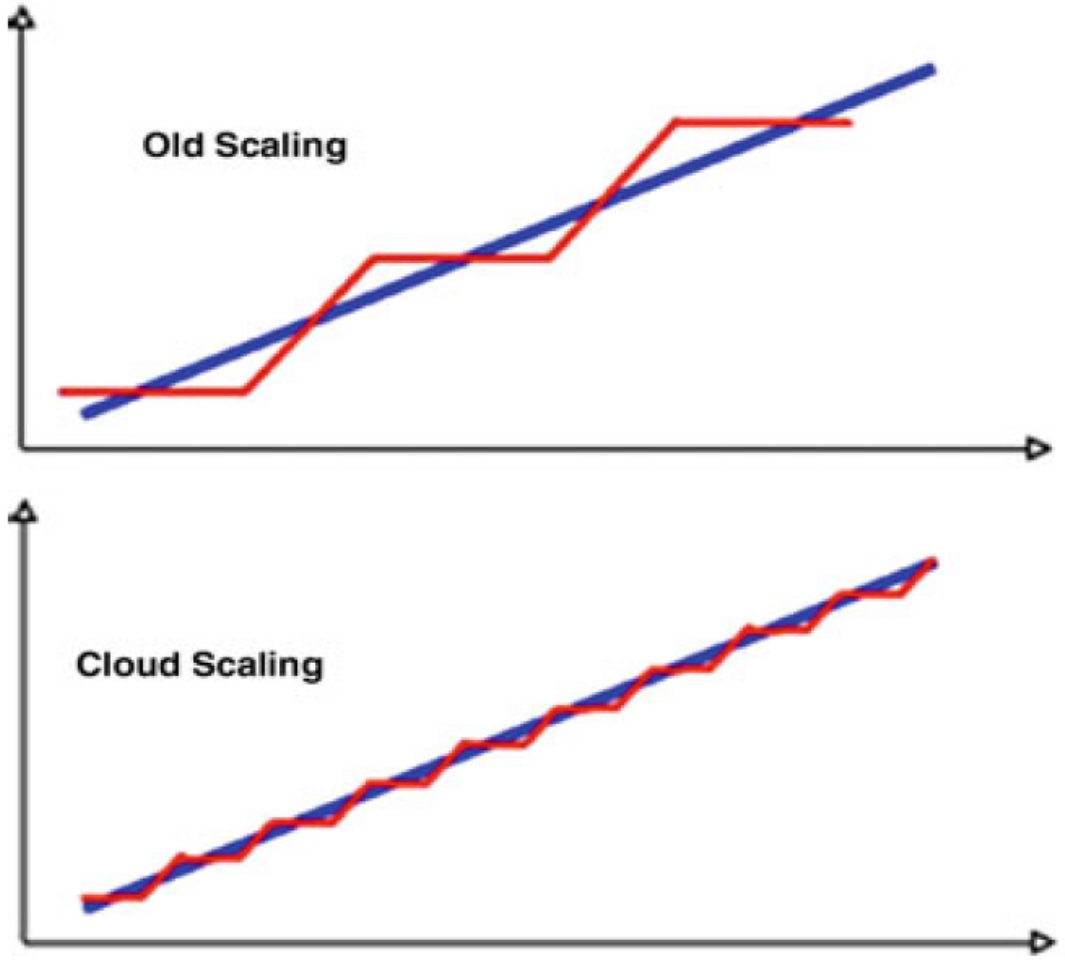


Fig. 15.4 Cloud computing techniques to reduce over- and under-provisioning

Table 15.2 Scaling potential of current grid and IaaS clouds

Grid	Cloud
• Adding new organizations (and their shared resources) to the virtual organization. • Manual service scaling.	• "Renting" resources on-demand. • Preliminary automation means to enforce scaling rules. • Horizontal scaling thanks to hypervisor technology.

- This ability of expanding and shrinking of a system as per workload is known as scaling.

- Static scaling requires system shut-down (system to restart)

In the traditional static scaling approach, computing system requires a ‘restart’ for the scaling effect to take place which causes service disruption

- Dynamic scaling - **dynamic-automatic scaling**, the system resource capacity can be altered while a system is running

- Dynamic scaling enables a system to keep performing consistently during times of massive demand by expanding it at pace with growing demand.

- It should be noted that in any computing environment, the scaling-down is as important as scaling-up from business point of view.

- Scaling in Cloud is Reversible

- Implementation of reversible scaling is a critical act as system performance should not be hampered while releasing the resources.

Service Scalability Over the Cloud

- The ability to scale up a system may depend on its design, the types of data structures, algorithms or communication mechanisms used to implement the system components

Characterization of different types of scalability

- **Load Scalability** – when a system has the ability to make good use of available resources at different workload levels (i.e. avoiding excessive delay, unproductive consumption or contention)

- **Space Scalability** - The system has the ability to keep the consumption of system's resources (i.e. memory or bandwidth) between acceptable levels when the workload increases.

- **Structural scalability** -

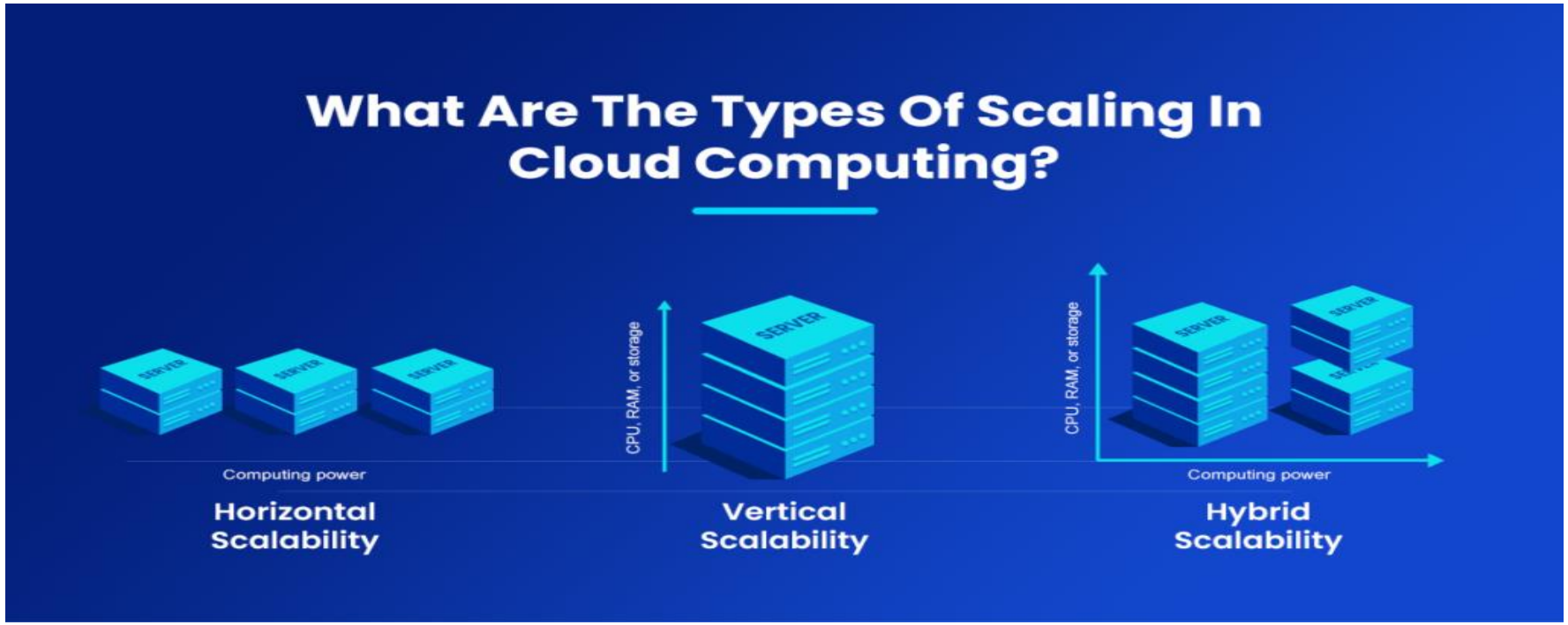
Sometimes the actions taken to improve one of these capabilities could spoil others

Service Scalability Over the Cloud

The actions to scale may be classified in

- **Vertical scaling -** by adding more horsepower (more processors, memory, bandwidth, etc.) to equipments used by the systems - add CPU, RAM, or storage to an existing server

- **Horizontal scaling -** by adding more of the same software or hardware resources - refers to the ability to add more instances of the same resource to a cloud environment. For example, you can add more servers to your environment if you need more computing power.



Cloud scalability is based in **three basic pillars**

- **Virtualization**
- **Resource sharing**
- **Dynamic provisioning**

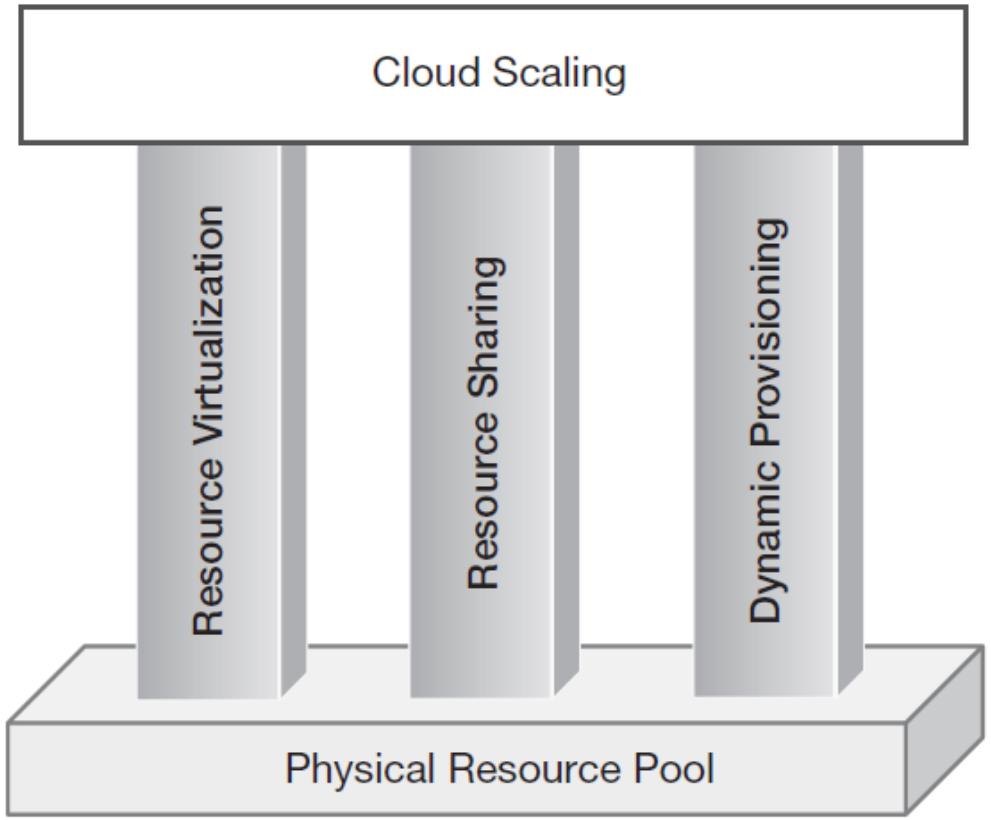


FIG 9.1: Foundations of cloud scaling

Application Scalability

- Web applications can be scaled horizontally and vertically
- scaling of application depends on two layers which are, scalable application architecture and scalable system architecture
- Application's scalability depends on its design, the types of algorithms applied and on the data structures used, and database system architecture
- It is not possible to take full advantage of the scalable computing infrastructure if the application architecture is not scalable. Both have to work together to maximize the gain.
- SaaS scaling is not just about having a scalable underlying (virtual) hardware, but also about writing scalable applications

To Do

- Sharding (partitioning) data set into smaller fragments and distributing them among a large number of servers
- Dynamic Load Balancing by biasing the sharding policy to equalize the workload per node
- Health checking and watchdog timers
- Integrity checks to avoid data corruption
- Application-specific compression

Auto-scaling can be implemented in two different ways:

- Scaling based on a predefined schedule known as *proactive scaling*.
- Scaling based on current actual demand known as *reactive scaling*.

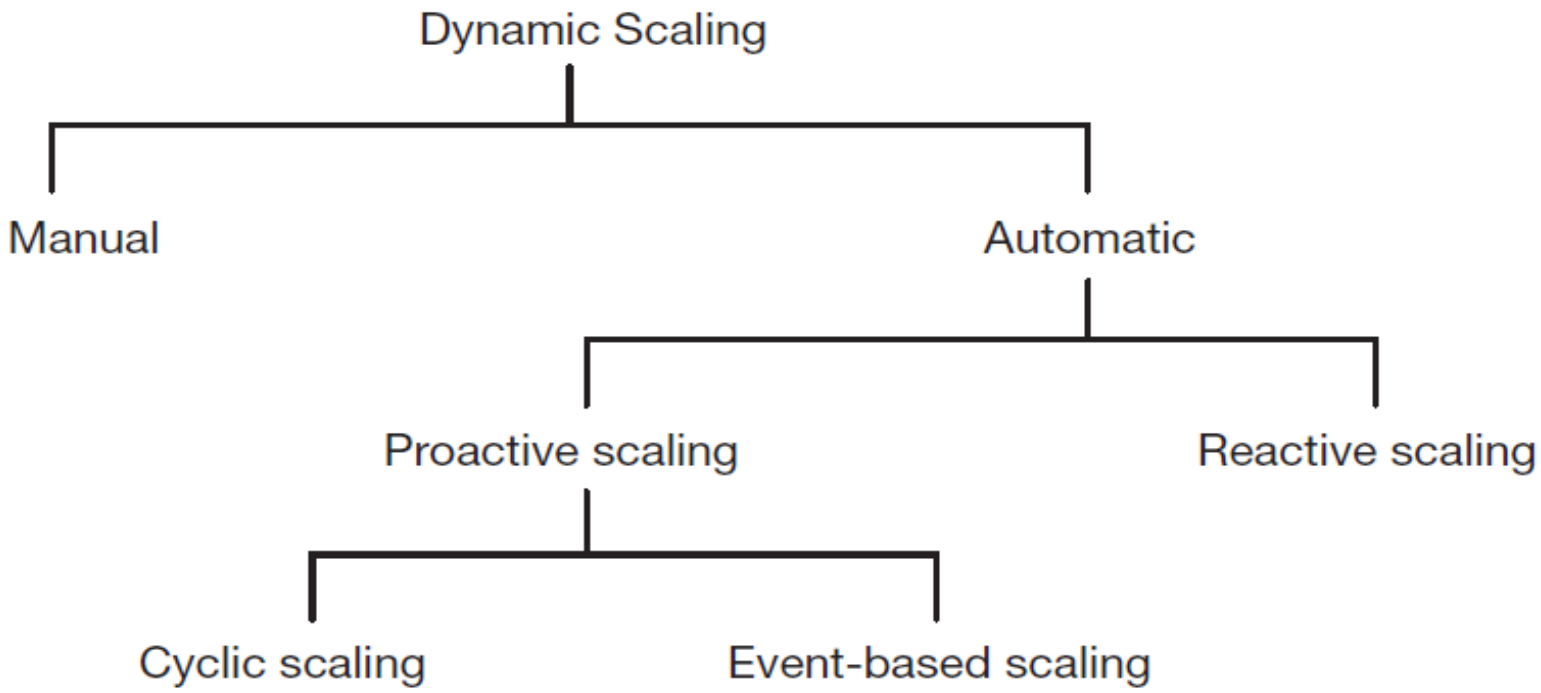


FIG 9.2: Classification of dynamic scaling approaches

Levels of Business Value from Cloud Computing

From a business viewpoint, cloud computing offers value to an organization at three different levels.

key benefits for **SMBs (Small Medium Businesses)**

	Level of Value from Using the Cloud	Description
1.	**Basic Level (Utility Level Value)**	Consumer organizations benefit from fundamental features of cloud such as lower IT costs, higher service levels, scalability to meet peak loads, absence of fixed or capital expenses, and pay-per-use billing. Focus is on labour, IT resources and power
2.	**Intermediate Level (Process Transformation Level Value)**	Enterprises find it difficult to improve business processes, because they are usually ineffectively supported by traditional in-house IT infrastructure and teams. A cloud, on the other hand, allows business units to regulate to meet their specific requirements. For example, sales teams can use cloud-based CRM to improve sales tracking and customer relations. Human Resource (HR) departments can use cloud-based human capital management applications. Cloud users can introduce new processes by taking advantage of pooled and scalable resources in the cloud. It facilitates better collaboration between geographically-dispersed teams and users with mobile and remote access.
3.	**Advanced Level (Business Innovation Level Value)**	The business innovation level aims to create new value chains between organizations and customers and novel and pioneering business models. This is achieved by rewiring the way organizations can operate using cloud resources, making choices of competitive advantage, and deriving new values from cloud-based services. It can be used to collaborate between users, customers, and partners.