**2. Cloud Native Services**

Shivakumar R Goniwada[**1**](https://learning.oreilly.com/library/view/cloud-native-architecture/9781484272268/html/511610_1_En_2_Chapter.xhtml#Aff2)

(1)

Bangalore, India

In the previous chapter, I discussed cloud native architecture and its importance in the IT industry. In this chapter, I will explore that topic further, but in the context of supporting cloud services.

Cloud services were developed to support a cloud native architecture. Many organizations start their journey by adopting various cloud services; as a result, the organization’s business goals, objectives, and processes greatly affect how they will provision resources and develop cloud native applications.

This chapter focuses on various cloud services and the evolution of each service. We will look at services in detail and how to adopt these services.

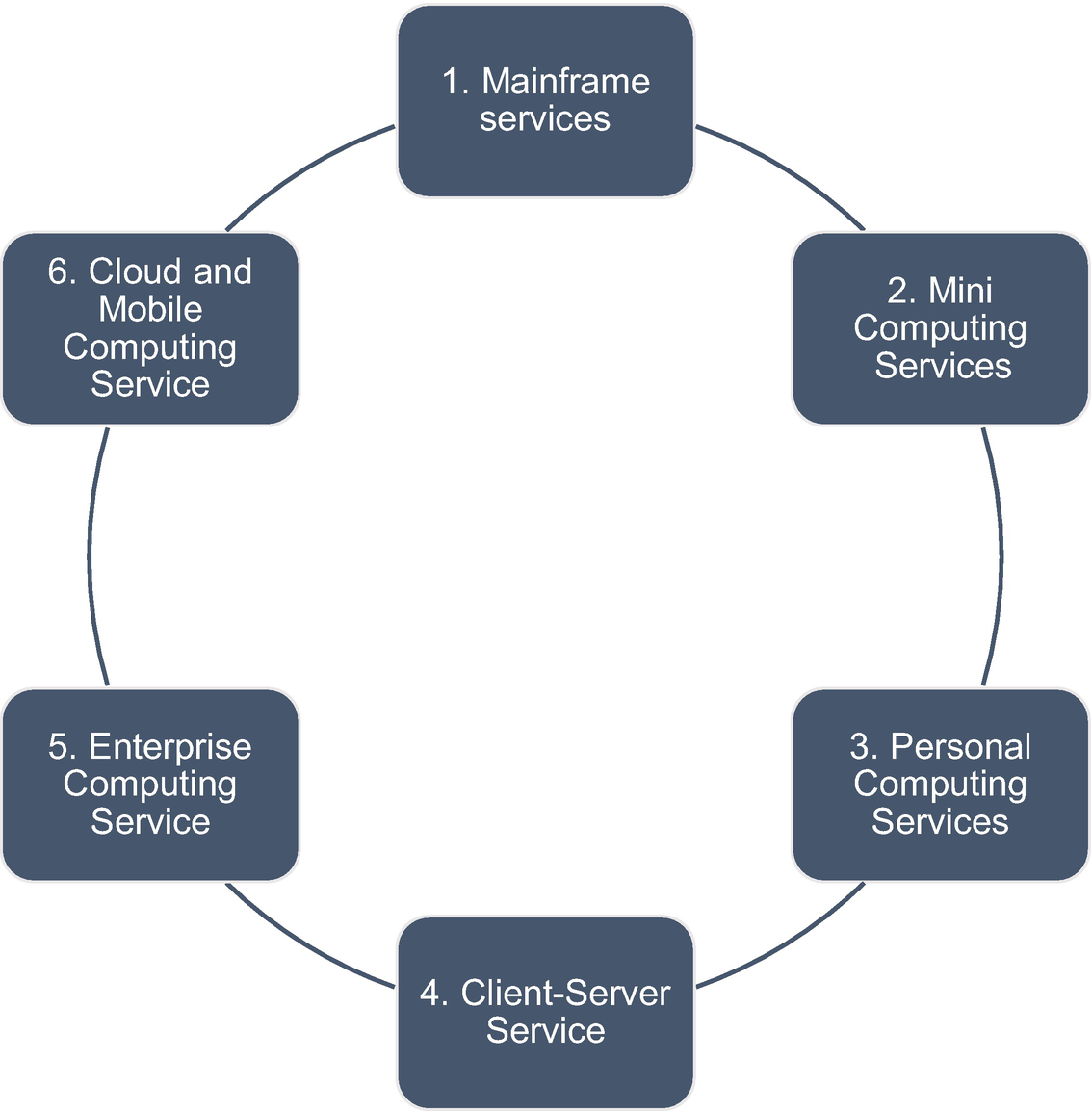
In this chapter, we will cover the following topics:

* Evolution of infrastructure as a service (IaaS)
* IT infrastructure laws
* Evolution of server technology
* What is containerization?
* What is IaaS?
* What is platform as a service (PaaS)?
* What is software as a service (SaaS)?

**Evolution of Infrastructure Services**

An IT infrastructure service is the shared technology resources that provide the services to the applications. Infrastructure services include the hardware, software, operating system (OS), networking services, telecommunication services, Internet services, etc.

Infrastructure services are a result of five decades of evolution in computing. To reach the present-day container technology level, the infrastructure has undergone six stages of evolution, each representing different subservices. Figure [2-1](https://learning.oreilly.com/library/view/cloud-native-architecture/9781484272268/html/511610_1_En_2_Chapter.xhtml#Fig1) shows the six stages.

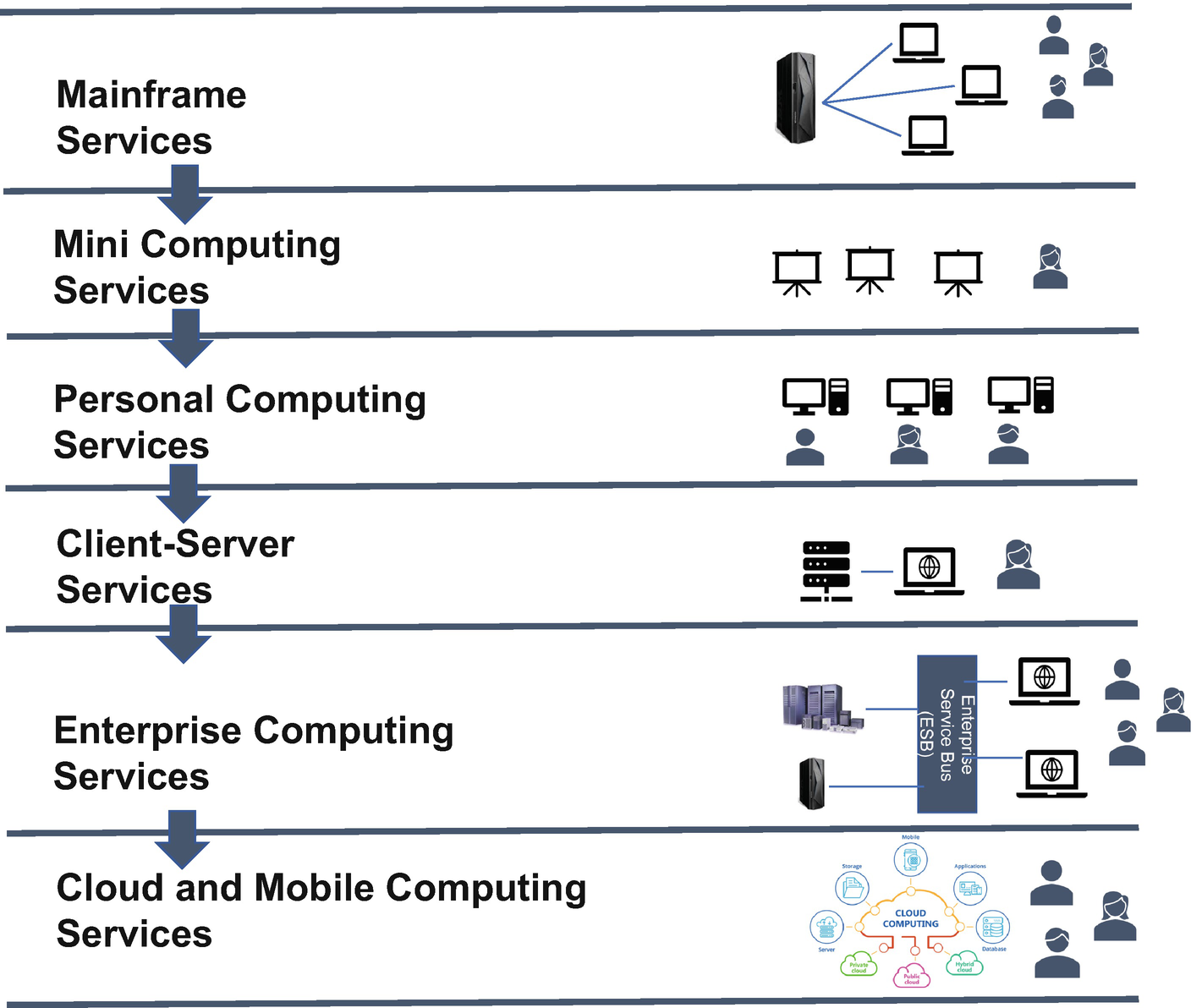


***Figure 2-1***

Six stages of infrastructure services

Technologies that are used in one stage may also be used in another stage for other business services. For example, a lot of financial- and insurance-sector business processes are using mainframe services, and these mainframe services are consumed by container-based microservice use cases. The mainframe shared model demonstrates a stage 1 evolution used by a stage 6 evolution in many enterprises as shown in above figure.

Figure [2-2](https://learning.oreilly.com/library/view/cloud-native-architecture/9781484272268/html/511610_1_En_2_Chapter.xhtml#Fig2) shows the stages of IT infrastructure.



***Figure 2-2***

Stages in IT evolution

**Mainframe Services**

The first general-purpose automatic digital computer was built by IBM around 1944. It was an electromechanical machine developed in conjunction with Harvard University. In 1952, IBM announced its first fully electronic data processing system, the IBM 701; in the next few years, the IBM 650 was created. In 1959, IBM introduced two of its most important computers. These were the 1401 Data Processing System, widely used for business applications, and the 1620 Data Processing System, a small scientific and engineering computer used for such diverse applications as automatic typesetting, highway design, etc.

The IBM introduced the large-scale 7000 series, the 1410, and Stretch (IBM 7030), the most powerful scientific computer ever designed. In the 1960s, IBM announced System/360, which was the first system where companies integrated all of their data processing systems.

In the 1990s, IBM introduced System/390 with high-speed fiber-optic channels, ESCON architecture, ultra-dense circuits, and circuit packaging for higher performance.

Currently, a mainframe runs with the Z series, z900, which includes the newly designed 64-bit z/architecture; most enterprises use a mainframe for their core business.

**Minicomputer Services**

The small computer was developed in the 1960s and sold for a much lower price than mainframes. Examples of minicomputers are Control Data’s CDC 160A and CDC 1700, HP 3000 series and HP 2000 series, IBM midrange computer, Texas Instrument T1-990, etc.

**Personal Computing Service**

The PC started with the IBM PC in 1981 and was widely adopted by the business community; later the Macintosh (Apple) computer and Intel-based Windows PCs came on the scene. A personal computer works in a stand-alone state with its CPU and is used by an individual. Worldwide sales at the end of the third quarter of 2020 were $71.4 million, which is a 3.6 percent increase from the previous year. Predominantly PCs are used by end users to connect various ancillaries and servers. In recent years, the PC has become more and more difficult to pin down. A PC can be any personal device with a microprocessor.

**Client-Server Service**

Client-server architecture is a computing model in which the server hosts and manages most of the services to be consumed by the client. This type of architecture has one or more client computers connected to a central server (the central server can be Linux, Solaris, AIX, or Windows) over a local or wide area network or the Internet. Currently we are calling this *legacy software*; these legacy software packages are based on a client-server architecture. The server is a single monolithic application and provides services to a thick client hosted on a PC, and the data is exchanged between the client and server over the network by using RPC.

**Enterprise Computing Service**

Enterprise computing was among the most important developments in information technology in the 1990s. Nearly every top company has implemented some form of enterprise system. Enterprise computing involves the use of computers in networks, such as LANs and WANs, or a series of interconnected networks encompassing a variety of different operating systems, protocols, and network architectures.

The enterprises turned to network standards and software tools that could integrate disparate networks and applications within and across business units (BUs) over the TCP/IP protocol. The commonly used tools in enterprise computing include enterprise resource planning (ERP), customer relationship manager (CRM), reporting, order systems, etc. All these systems are in a monolithic single unit and running on a single CPU in memory.

**Cloud and Mobile Computing Services**

Cloud computing as a term has been around since the 2000s, but the concept of computing as a service has been around for much longer, since the 1960s, when IBM allowed companies to rent time on a mainframe, rather than have to buy one themselves.

The growing bandwidth power of the Internet and disruption in business and technology pushed the client-server model to the cloud computing model. Cloud computing is the result of the evolution and adoption of existing technologies and paradigms. The goal of cloud computing is to allow users to get the benefits of all the services without the need for deep knowledge about or expertise in each one of them.

According to Wikipedia, cloud computing is the on-demand availability of computer system resources, especially cloud storage and computing power, typically over the Internet and on a pay-as-you-go basis.

Rather than each enterprise owning its infrastructure or data centers, companies can rent services from cloud providers. This helps enterprises to outsource servers, space, resources, etc., with the most security possible.

Cloud computing services provide a vast range of options starting with infrastructure, software, storage, platform, networking, natural language process, and artificial intelligence, and also provide traditional software like ERP, CRM, etc.

Today, cloud computing is becoming the de facto standard for all enterprises, and some software providers are discontinuing on-prem licenses and provide only cloud service licenses.

Enterprises can use single cloud provider services or a combination of multicloud provider services or hybrid services or private cloud services. Cloud providers are now in competition, so each provider provides free tools and solutions to port from one cloud provider to another provider seamlessly.

In the future, the cloud will become the de facto standard for all computing. Especially after the COVID-19 pandemic, most enterprises (even financial enterprises) are moving toward the cloud. Various research institutes predict that half of all global enterprises use the cloud now.

According to Gartner, global spending on cloud services will reach $350 billion by 2021 and will reach $500 billion by 2023.

**IT Infrastructure Laws and Prediction**

As the stages progress as shown in Figure [2-1](https://learning.oreilly.com/library/view/cloud-native-architecture/9781484272268/html/511610_1_En_2_Chapter.xhtml#Fig1), infrastructure services are becoming cheaper, with exponentially increased computing power. The following are the theories that predict the IT infrastructure changes in the years to come.

**Moore’s Law**

Moore’s law is a prediction made by American engineer Gordon Moore in 1965 that the number of transistors per silicon chip will double every year. He observed that the number of transistors on a computer chip was doubling about every 18–24 months. This is an observation and projection based on historical trends, rather than a law of physics.

There are three interpretations of Moore’s law .

* The power of microprocessors doubles every 18 months.
* Computing power doubles every 18 months.
* The price of computing falls by half every 18 months.

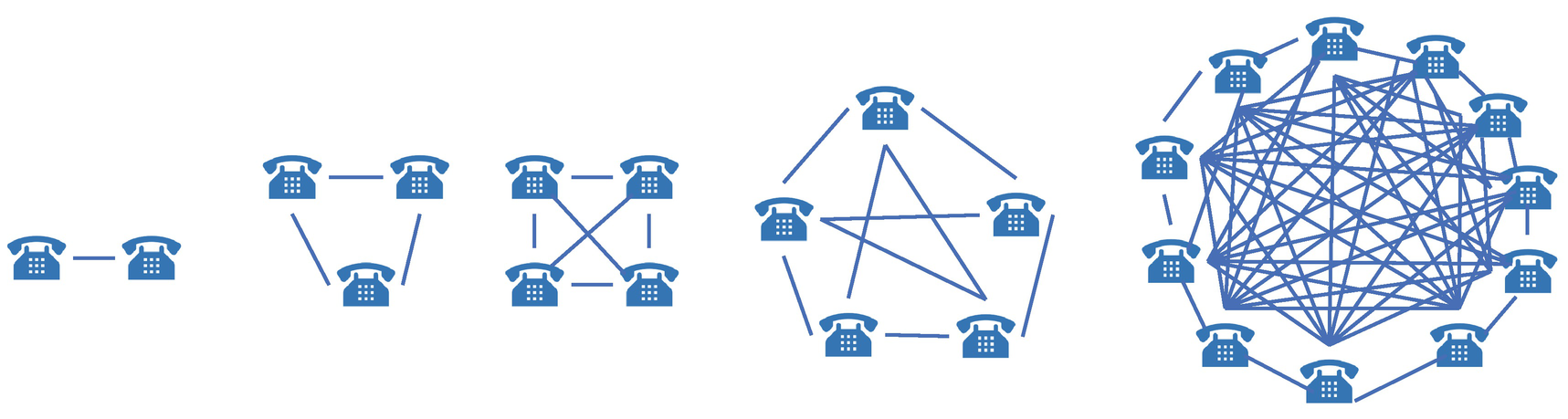
For example, Moore’s law means we get ever-more powerful personal computers for less and less money. A computer chip that contained 2,000 transistors and cost $1,000 in 1970, $500 in 1972, $250 in 1974, $0.97 in 1990, and less than $0.02 to manufacture today.

**The Laws of Mass Digital Storage**

The amount of information is roughly doubling every year, and the cost of storing digital information is falling at an exponential rate. Currently, the compound annual growth rate is roughly around 60 percent, with an exponential decrease in the cost of storing data.

**Metcalfe’s Law**

Metcalfe’s law states that the effect of a telecommunication network is proportional to the square of the number of connected users of the system. The law shows that a network’s value to participants grows exponentially as the network takes on more members. The increasing scale of that network grows exponentially as more and more people join the network, as shown in Figure [2-3](https://learning.oreilly.com/library/view/cloud-native-architecture/9781484272268/html/511610_1_En_2_Chapter.xhtml#Fig3). As the number of members in a network grows linearly, the value of the entire system grows exponentially and continues to grow as members increase.



***Figure 2-3***

Network increases linearly

**Communication Cost and Internet**

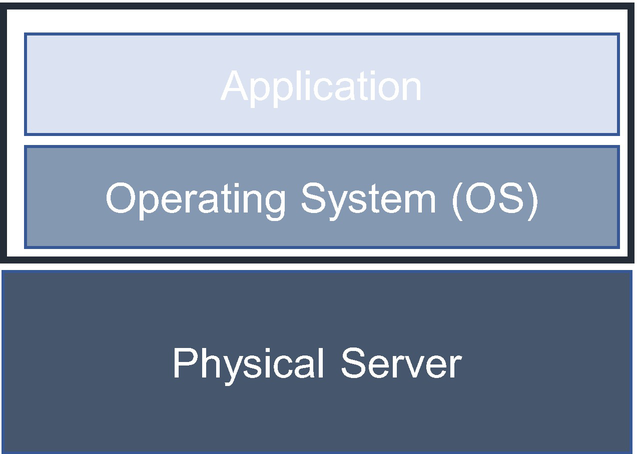
There has been a rapid decline of the cost of communication and an exponential growth in the size of the Internet. Estimated Internet access is around 4.12 billion, which means more than 50 percent of the global population is connected to the Internet. As communication costs fall, the utilization of communication and computing facilities grows.

**Evolution of Servers**

The servers used have evolved from bare-metal physical servers to virtual servers to cloud servers and containers to serverless.

**Bare-Metal Servers**

We began with a bare-metal server/physical server , as shown in Figure [2-4](https://learning.oreilly.com/library/view/cloud-native-architecture/9781484272268/html/511610_1_En_2_Chapter.xhtml#Fig4). Each server offered for rental is a distinct physical piece of hardware that is a functional server on its own; in other words, each physical box hosts one piece of hardware.



***Figure 2-4***

Bare-metal architecture

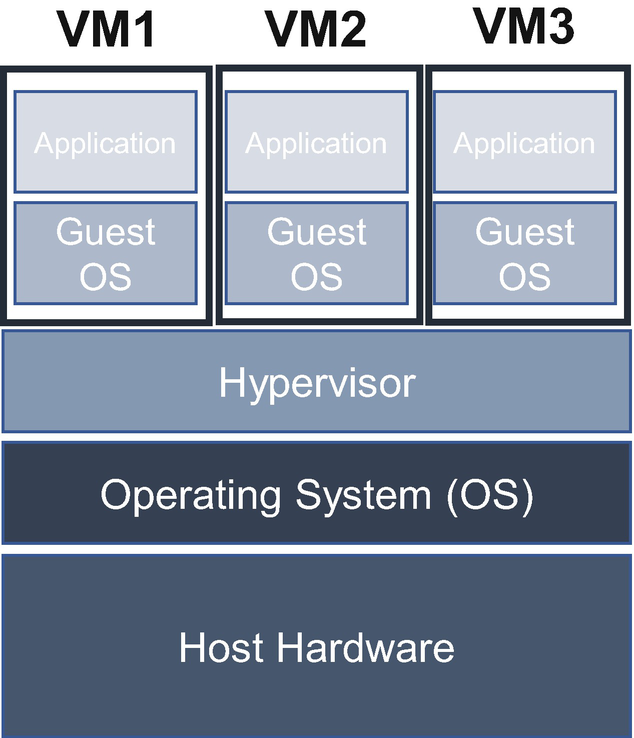
These servers require a physical box and deploy an OS on it, after which we layer on specific application software to perform the required business functionality on that system.

In the early 2000s, it became evident that enterprises were not getting appropriate value for their server dollar. The CEOs of enterprises questioned why so many expensive servers were running such low utilization rates.

**Virtual Machine Revolution**

Virtualization uses the same physical hardware, but rather than installing a single OS and running a single workload on that physical box, install a hypervisor OS and set it up to support multiple virtual machines or virtualized servers that can run many different business applications all at the same time on one physical server.

The VMs are hosted with their CPU, memory, network interface, and storage on physical hardware, as shown in Figure [2-5](https://learning.oreilly.com/library/view/cloud-native-architecture/9781484272268/html/511610_1_En_2_Chapter.xhtml#Fig5). The hypervisor separates the single physical server resources from the hardware and provisions them appropriately so they can be used by the VM. The VMs that use physical server resources are guest machines, guest computers, and guest OSs. The hypervisor treats compute resources such as CPU, memory, and storage as a pool of resources.



***Figure 2-5***

Virtual machine architecture

Virtualization technology allows you to share resources with many virtual environments. The hypervisor manages the hardware and separates the physical resources from the virtual environments. Resources are partitioned as needed from the physical environment to the VMs. When the VM is running and a user or program issues an instruction that requires additional resources from the physical environment, the hypervisor schedules the request to the physical system’s resources so that the VM’s OS and applications can access the shared pool.

Each VMs is isolated from the rest of the VMs and can be co-located on a single piece of hardware, and VMs can allow multiple OSs to run simultaneously on a single computer, such as a Linux distro on a macOS laptop or a Linux distro on a Windows OS laptop.

**Adoption of Virtual Machines**

Organizations soon recognized that they could get much more value by virtualizing applications. Virtualization provides a significant improvement in enterprise computing and realizes benefits from managing servers and the applicable costs, as mentioned here:

* Better utilization of server, network, and storage resources
* Better return on investment on infrastructure
* Better portability
* Better management of server setup, network, etc.

**Virtual Machines in the Cloud**

The introduction of virtual machine technology into the organization’s own data center was hardly a cure-all, from a financial and resource point of view. The organization still faced the high costs of data center real estate, electricity, and environmental conditioning, of computer storage and networking hardware, and of managing the software and platforms.

To alleviate some of these issues in the data center and managing platforms, enterprises turned to co-location, third-parties, and a service model; this means renting data center space from a third party and outsourcing the server management to the third party in their data center. This strategy relieved enterprises of the cost of maintaining real estate and facilities, but companies directly or indirectly own the servers and other accessories.

With the availability of the public cloud, a new cost model—renting capacity—emerged, which allowed companies to think of computing as an on-demand resource. A few enterprises started to migrate a few systems from their own data centers to VMs in the cloud; this process is called *lift and shift*. This type of model started with AWS in 2006 and followed by others like Google’s GCP and Microsoft’s Azure. These infrastructure as a service solutions provided a way to run your VMs on their cloud. Thus, this was the beginning of the cloud revolution.

Let’s discuss a few benefits and drawbacks of VM migration to the cloud.

Here are the benefits:

* Less management required because VMs in the cloud are developed and maintained like VMs on-premises
* Vastly reduced the real-estate expenditure
* New cost model based on a rental approach to computing instead of a capital expense
* New usage model for scaling up and down depending on the demand

Here are the drawbacks:

* Costs can increase with VMs if care is not taken in the planning stage.
* Without changes, some applications are not an ideal fit for the cloud.
* It is a better, but still suboptimal, use of server resources.

**Container Revolution**

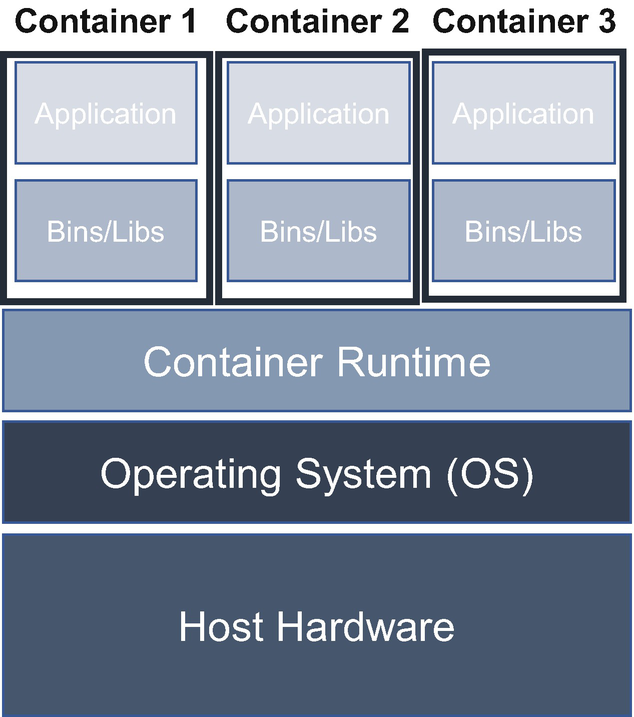
Many organizations first experienced the cloud by migrating virtual machines from the data center to the cloud and then facing costs that were higher than expected or business value that was less than anticipated. Over the last few years, a new form of virtualization has arisen, called *containers*.

The concept of containers has been around for a while; for example, IBM supported the notion of web application containers. However, these early approaches suffered from an increasing scope, compounding rather than alleviating the issue of idle capacity on servers.

The philosophy of a container is to put in what you need to make it whole, and nothing more. Easy-to-use and compact container technology supporting a higher degree of isolation has been mainstream only since mid-2014, when Docker introduced version 1.0. Since then, containers have become widely popular.

A container is a standard unit of software that packages the application code and all its dependencies, so the application runs quickly and reliably and is abstracted from the environment in which it runs. It is a lightweight, standalone, executable package of software that includes everything needed to run an application. This decoupling allows container-based applications to be deployed easily and consistently, regardless of whether the target environment is a private data center, the public cloud, or even a PC. The container uses the same physical hardware and OS but in an isolated, lightweight silo for running an application on the host OS. Each container can host a single service or multiple services depending on the nature of services.

As mentioned in Figure [2-6](https://learning.oreilly.com/library/view/cloud-native-architecture/9781484272268/html/511610_1_En_2_Chapter.xhtml#Fig6), in VMs, the guest OS such as Linux or Windows runs on top of the host OS with virtualized access to the underlying host hardware. Like VMs, the containers allow you to package your application together with libraries and dependencies, providing isolated environments for running software services.



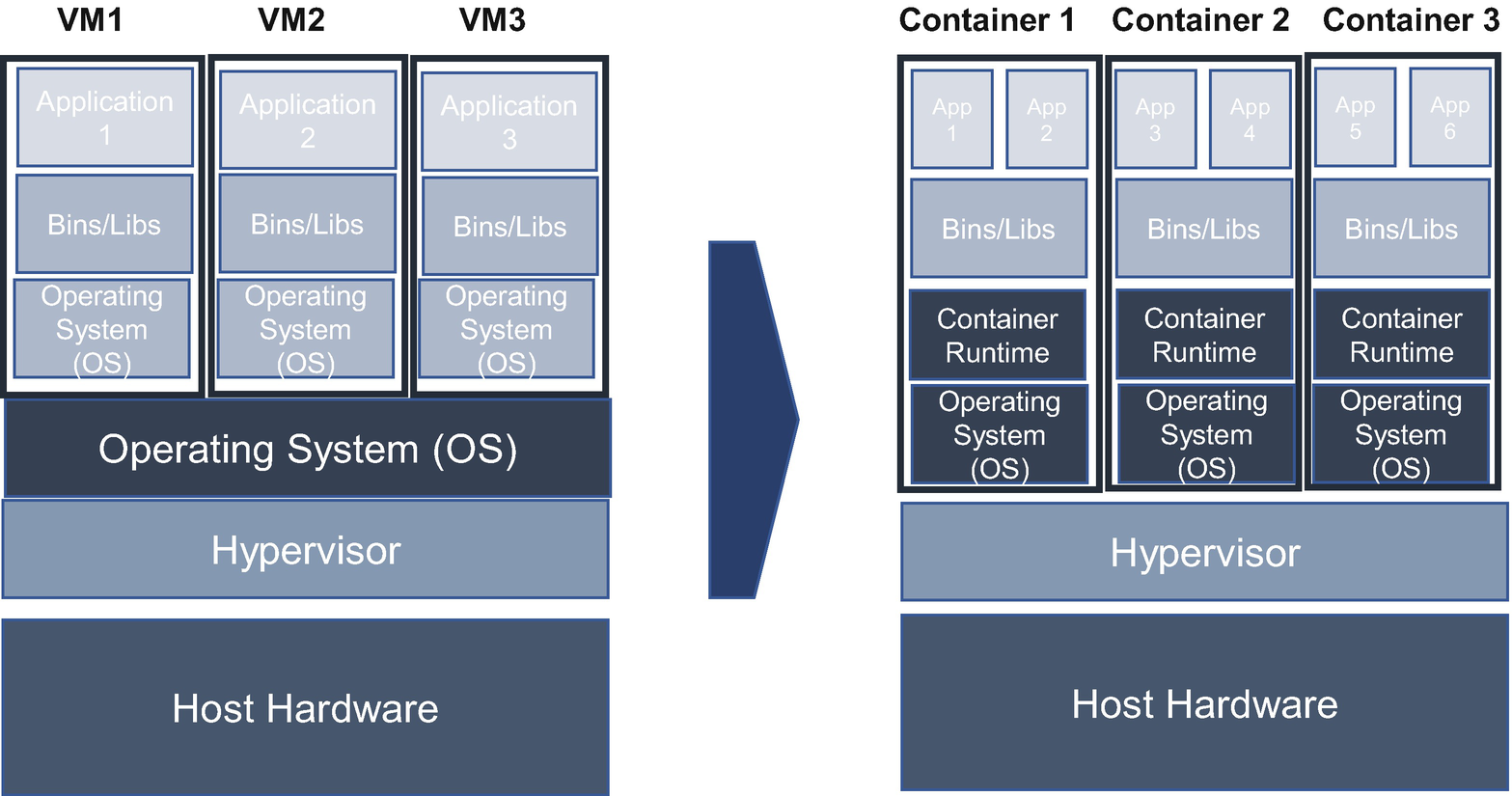
***Figure 2-6***

Container environment architecture

The virtual machines are virtualized on the hardware stack, and containers virtualize at the OS kernel, start much faster, and use a fraction of the memory compared to booting an entire OS. A virtual machine virtualizes CPU, memory, storage, and network resources at the OS level.

Containers can run virtually anywhere: on Linux, Windows, and macOS; on VMs or on bare metal. All cloud native applications will use a container to host business use cases. There are many companies that provide container images other than Docker such as Mesos, Open VZ, CoreOS rkt, and more.

Figure [2-7](https://learning.oreilly.com/library/view/cloud-native-architecture/9781484272268/html/511610_1_En_2_Chapter.xhtml#Fig7) shows a high-level comparison between virtual machines and containers.



***Figure 2-7***

Virtual machine and container comparison

Since their introduction, containers have become wildly popular for several reasons, including the following:

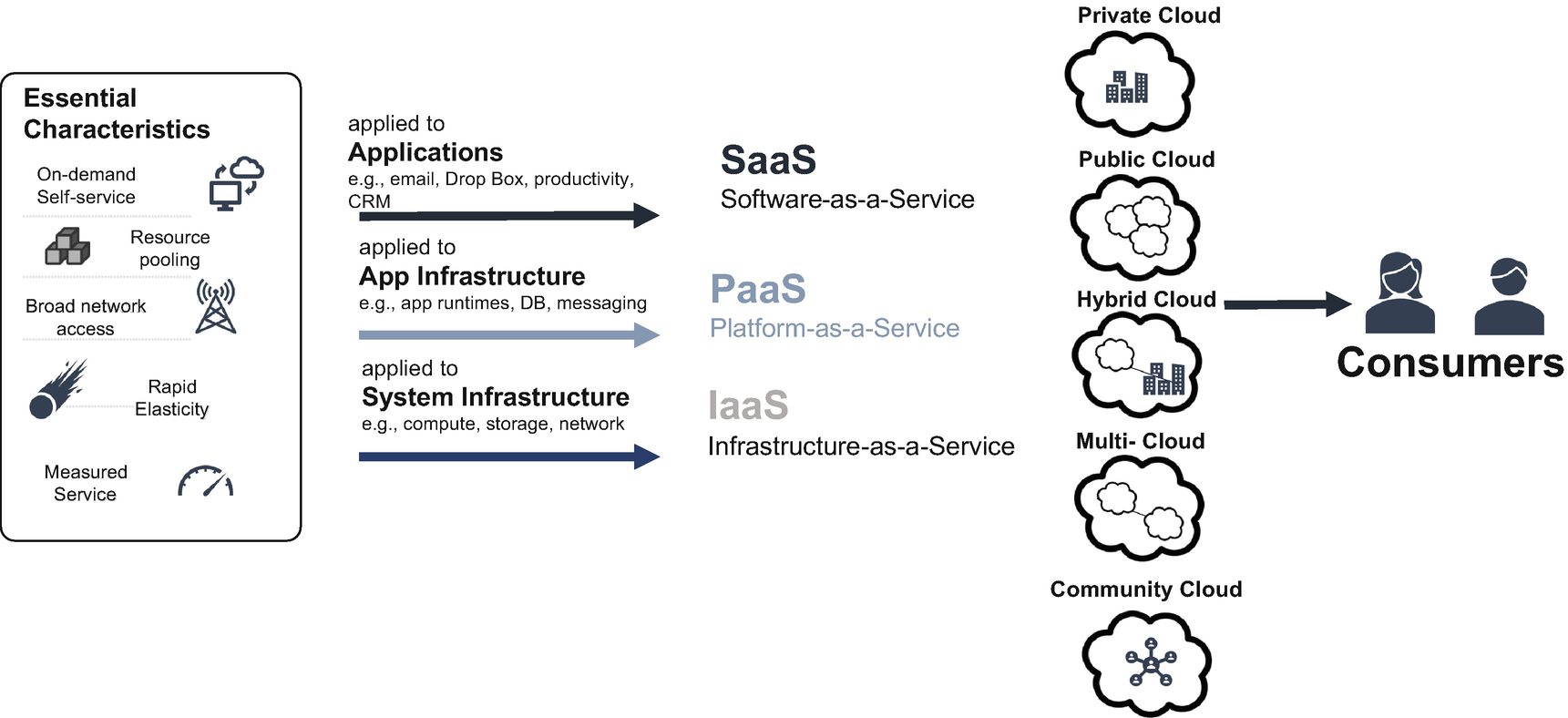
* *Server density*: There is only one copy of the OS; you can often create many container images on a given VM server.
* *Startup time*: You do not need any initialization required like the OS; startup time is much faster.
* *Portability*: Containers can run in a variety of environments including public, private, hybrid, or on-prem.
* *Scalability*: They can scale up and down in few seconds without any human intervention.
* *Wide support*: All the cloud providers support containers, and wide community support is available.

VMs have a few limitations. Since there is only a copy of the operating system, you can’t run an application written in a different OS on the same server. For example, if your application is running on Windows, Red Hat, or Ubuntu, then all applications in a container run this as a full VM.

**Understanding Cloud Services**

Cloud computing provides various services delivered on-demand to the customers over the Internet. These services are designed to provide easy, affordable access to applications and resources, without the need for internal infrastructure or hardware.

The cloud services are fully managed by cloud computing providers, as shown in Figure [2-8](https://learning.oreilly.com/library/view/cloud-native-architecture/9781484272268/html/511610_1_En_2_Chapter.xhtml#Fig8); they’re made available to customers or enterprises from the provider’s data center, so there’s no need for a company to host the application on its on-premises servers.



***Figure 2-8***

Cloud services

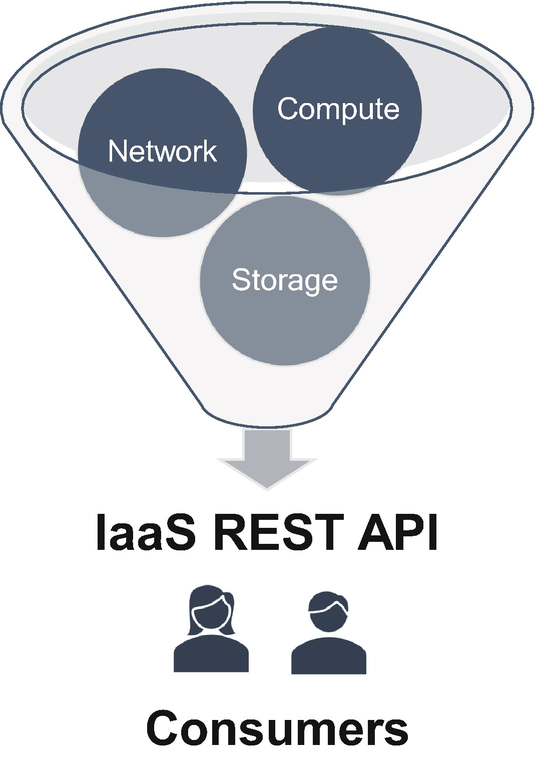
The benefits of cloud services are the ability to scale, increased flexibility, lowered cost, etc.

The following are the types of services offered by various cloud providers.

**Infrastructure as a Service**

IaaS is a form of cloud computing that delivers fundamental compute, network, and storage resources to consumers on-demand over the Internet and on a pay-as-you-go basis. IaaS enables consumers to scale and shrink resources on an as-needed basis. This service reduces an enterprise’s need for high, up-front capital expenditure or unnecessary procured infrastructure.

IaaS consists of a collection of physical and virtualized resources, as shown in Figure [2-9](https://learning.oreilly.com/library/view/cloud-native-architecture/9781484272268/html/511610_1_En_2_Chapter.xhtml#Fig9), that provide consumers with the building blocks needed to run business applications.



***Figure 2-9***

IaaS

IaaS providers manage large data centers across geographies that contain the physical and virtual machines and create a layer on top of these servers over the web. The end users do not interact directly with the physical infrastructure in the data center but are provided as REST services to them with parameters.

IaaS is typically a virtualized environment; IaaS providers manage the hypervisors, and end consumers can provision them through program and REST APIs with the desired amount of compute and memory for different types of use cases.

Networking in a cloud environment is defined by the software, and the consumer can access the required networking resources such as routers and switches through REST APIs. A consumer can create a virtual private cloud (VPC) on a single DC or multi-DC.

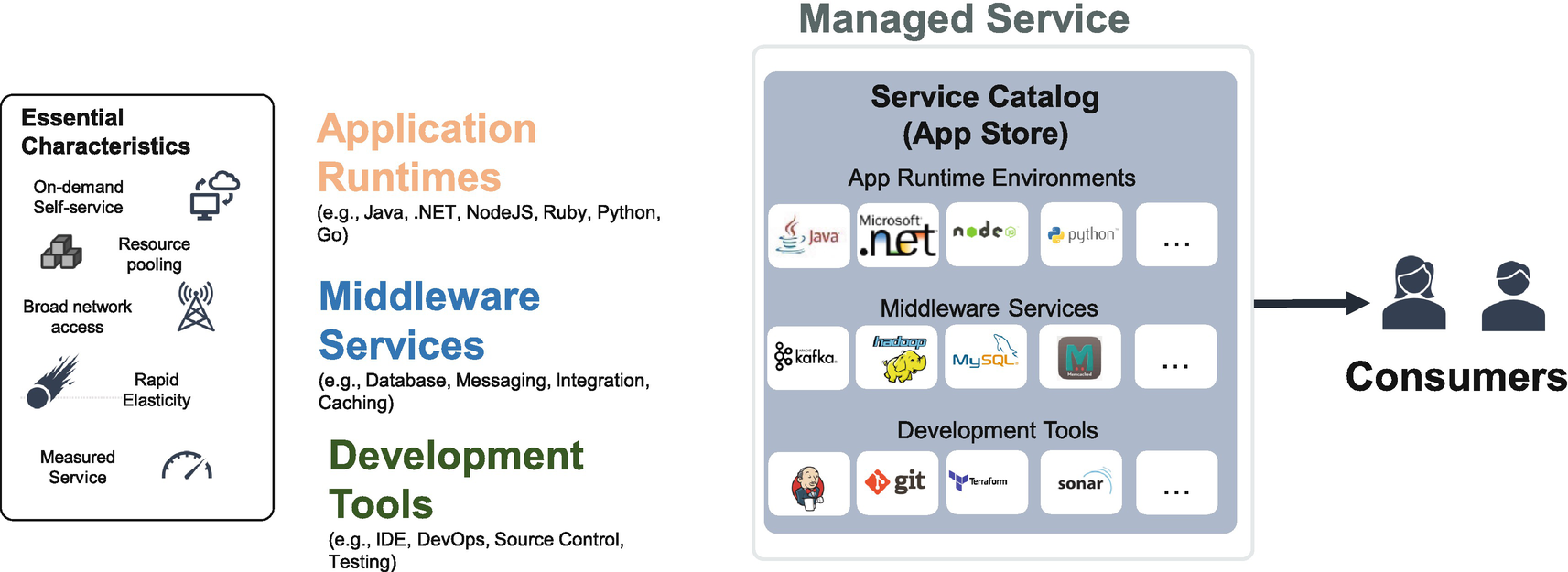
IaaS offers three types of storage options, listed here:

* *Block storage*: Block storage is used to store data files on a storage area network (SAN).
* *File storage*: File storage is a hierarchical storage methodology like the file or folder organization on your PC. The file storage can be organized on hardware or a network-attached storage (NAS) device.
* *Object storage*: This type of storage stores large amounts of unstructured data such as images, documents, etc. This data is organized in a folder called a *bucket*.

**Platform as a Service**

With PaaS , the consumer can deploy onto the cloud infrastructure consumer-created or acquired applications written using programming languages, libraries, services, and tools supported by the cloud providers. The consumer does not manage or control the underlying cloud infrastructure, including the network, servers, operating system, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment.

Like IaaS, the PaaS includes the infrastructure: servers, storage, and networking. The middleware services, development tools, BI services, data services, etc., as shown in Figure [2-10](https://learning.oreilly.com/library/view/cloud-native-architecture/9781484272268/html/511610_1_En_2_Chapter.xhtml#Fig10), will be used IaaS services.

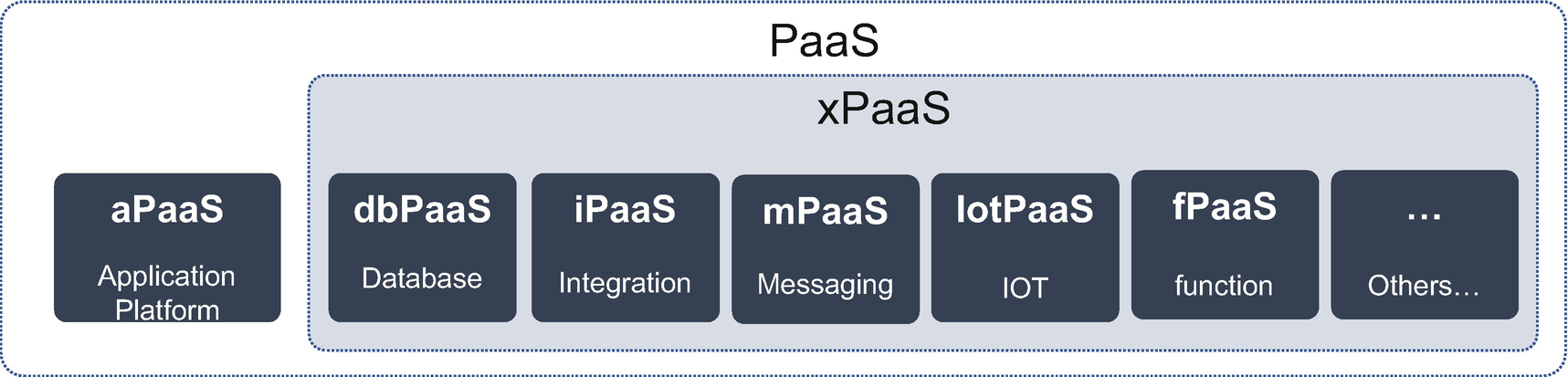


***Figure 2-10***

Platform as a service

**PaaS Taxonomy**

As of 2020, Gartner has identified 21 categories of xPaaS offerings, as shown in Figure [2-11](https://learning.oreilly.com/library/view/cloud-native-architecture/9781484272268/html/511610_1_En_2_Chapter.xhtml#Fig11), which refer to a particular type of application infrastructure functionality. aPaaS and xPaaS can be used together or independently.



***Figure 2-11***

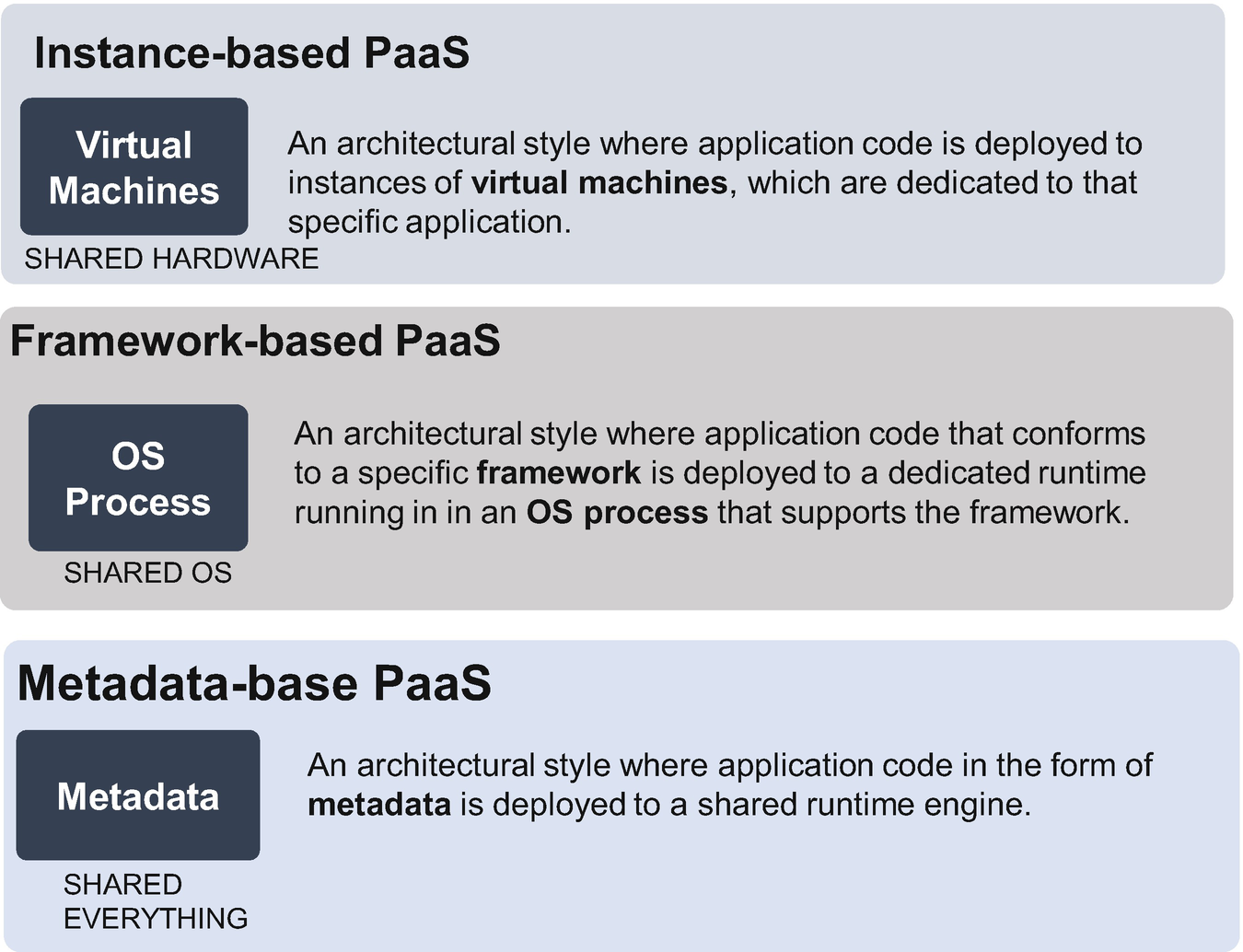
PaaS taxonomy

The application platform as a service (aPaaS) platform consists of the following:

* Automated deployment of code (IaC)
* Multitenant platform
* Service catalog
* Provisioning middleware services
* Provisioning development tools
* Tenant management
* Role-based access
* DevOps tools

**PaaS Architecture Styles**

The PaaS architecture style is a family of architectures that share certain characteristics. The architecture styles don’t require the use of particular technologies, but some technologies or services are well suited for certain styles. The architecture styles in Figure [2-12](https://learning.oreilly.com/library/view/cloud-native-architecture/9781484272268/html/511610_1_En_2_Chapter.xhtml#Fig12) are not related to any xPaaS and can be used across xPaaS services.



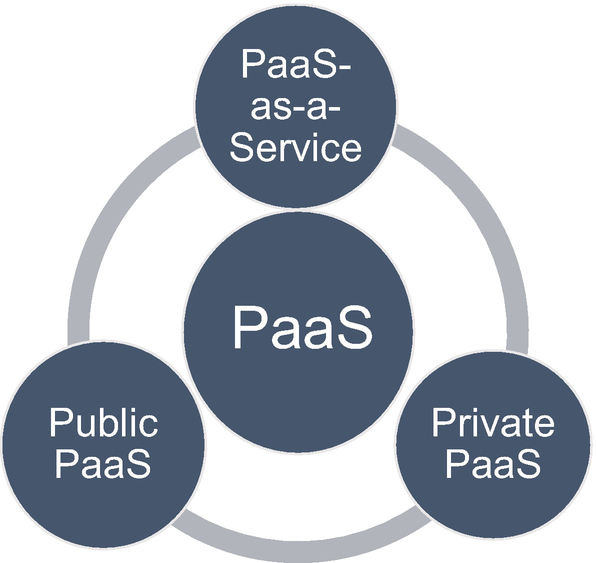
***Figure 2-12***

PaaS architecture styles

**PaaS Deployment Model**

The PaaS model can be deployed in three ways, as shown in Figure [2-13](https://learning.oreilly.com/library/view/cloud-native-architecture/9781484272268/html/511610_1_En_2_Chapter.xhtml#Fig13).

* *Public cloud model*: The PaaS capability provisioned for use by the general public over the Internet. The characteristics of the public cloud model are elasticity, utility pricing, and leverage of expertise, and a public cloud can be shared with all tenants with limited customization.



***Figure 2-13***

PaaS deployment model

* *Private PaaS model*: The PaaS capability provisioned for exclusive use by a single organization. It offers the same characteristics of the public cloud, such as elasticity and resource utilization but with total control and regulation flexibility. This type of model can run in the cloud or on-premises.
* *PaaS-as-a-service model*: The PaaS capability is provisioned for exclusive use by a single organization and managed by a third party. It offers the same characteristics of both public and private PaaS but is managed privately by a single organization.

The following are some PaaS limitations and concerns:

* *Data security*: The enterprises can run their developed apps and APIs using PaaS platforms. The data will reside on the vendor platform. The vendor-controlled cloud servers pose a concern. Every enterprise needs to do a thorough security review with the vendor before using services.
* *Vendor lock-in*: The enterprises make decisions based on certain requirements that drive them to use certain PaaS solutions; these decisions may not apply in the future due to business and technology disruption. Enterprises need to review the migration policies when moving to another vendor.

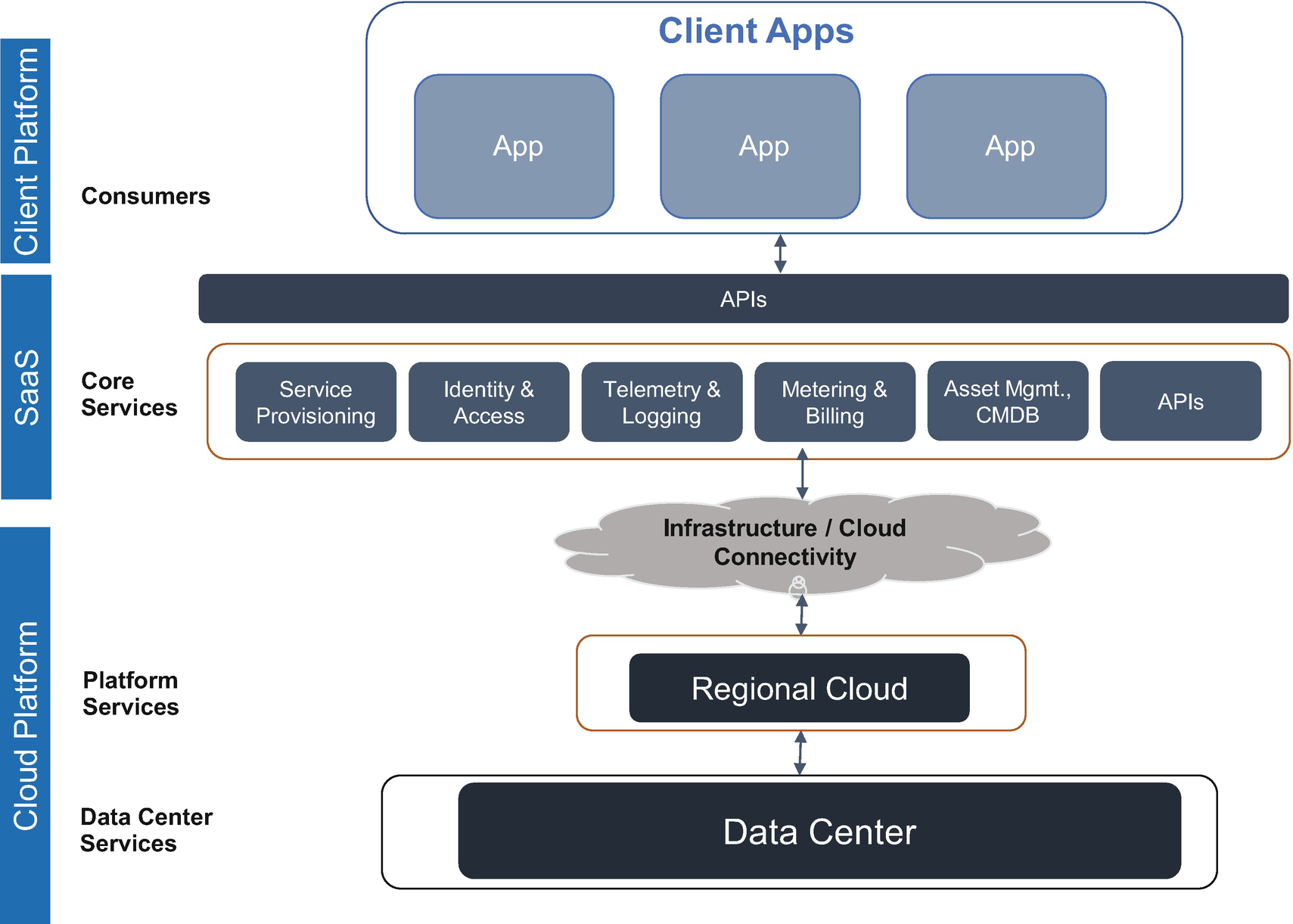
**Software as a Service**

SaaS is essentially on-demand software that is provided to the client over the Internet. SaaS has found more traction from small and midsize enterprises primarily due to its low capital and operational overhead.

According to Wikipedia, “SaaS is a software licensing and delivery model in which software is licensed on a subscription basis and is centrally hosted. It is sometimes referred to as on-demand software and was formerly referred to as software plus services by Microsoft. SaaS applications are also known as web-based software, on-demand software, and hosted software.”

Figure [2-14](https://learning.oreilly.com/library/view/cloud-native-architecture/9781484272268/html/511610_1_En_2_Chapter.xhtml#Fig14) shows a high-level view of the SaaS architecture. SaaS utilizes the Internet to provide services that are managed by third-party vendors. Most SaaS applications run directly through your web browser via an API.

* *Client apps*: Application modules/features that a client subscribed to as a service, provided by SaaS.
* *SaaS layer*: Provides common and cross-product functional and technical services and abstracts the SaaS services.



***Figure 2-14***

SaaS architecture

* *Platform layer*: Provides virtualized computing power, storage, networking, and cloud availability zones.
* *Data center services*: Provides the physical facility, network services, etc.

SaaS provides numerous advantages to enterprises by reducing operational activity such as installing and managing applications.

Common software offered vi a SaaS model includes sales software, CRM solutions, tax software, etc.

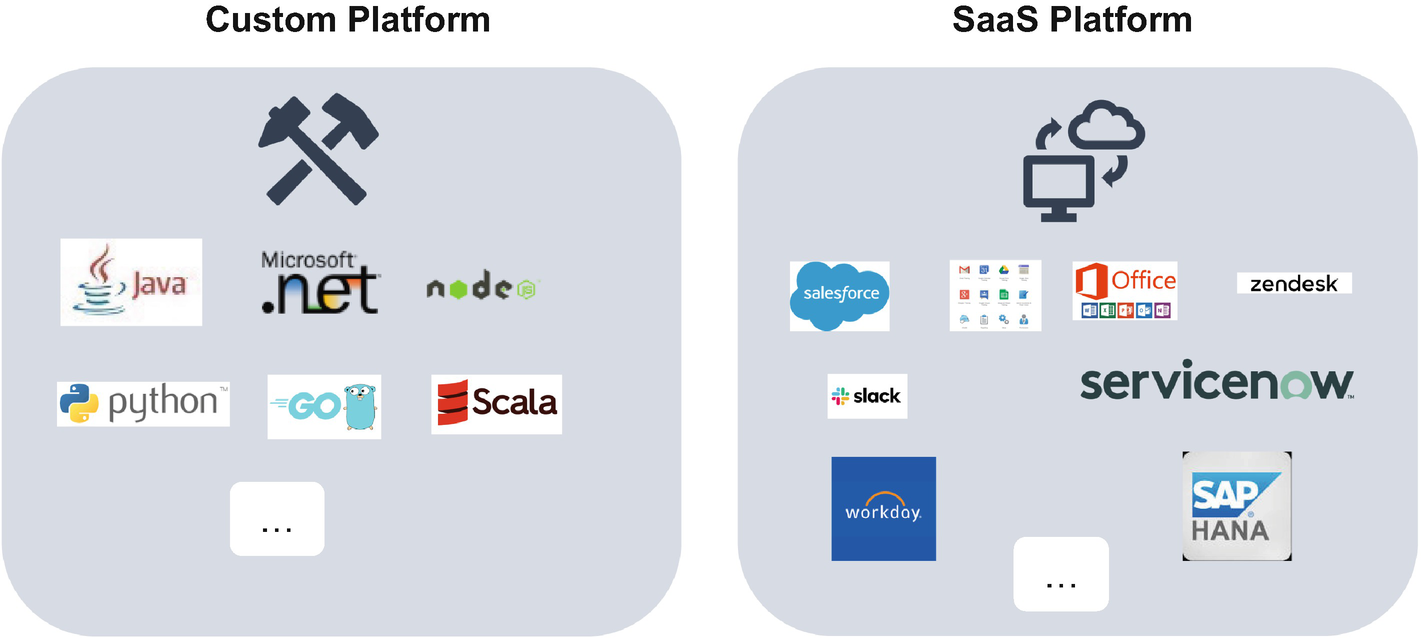
**SaaS Limitations**

The following are the limitations of SaaS:

* *Vendor lock-in*: It is easy to join SaaS services but difficult to leave. Most SaaS providers are not flexible on portability and compatibility, so it takes a huge amount of time to port from one vendor application to another.
* *Interoperability*: It is not easy to integrate SaaS applications with an existing legacy application. You need to modify existing applications to the vendor’s SaaS APIs. Sometimes you may require a separate integration layer for consumer SaaS services.
* *Data security*: SaaS software holds most of your business data. Security at rest and security in transit are concerns, even though all the APIs are security enabled and data is stored with encryption.

**Architectural Considerations: How to Decide on a Custom vs. SaaS Platform**

You can compare a custom platform to a SaaS platform solution, as shown in Figure [2-15](https://learning.oreilly.com/library/view/cloud-native-architecture/9781484272268/html/511610_1_En_2_Chapter.xhtml#Fig15). Identify the key contrasting dimensions and compare the architect’s involvement in those dimensions across the two platforms.



***Figure 2-15***

Custom and SaaS platforms

Table [2-1](https://learning.oreilly.com/library/view/cloud-native-architecture/9781484272268/html/511610_1_En_2_Chapter.xhtml#Tab1) shows the key factors to use when comparing custom and SaaS platforms.

***Table 2-1***

Custom vs. SaaS Platform

| Factors | Custom Platform | SaaS Platform |
| --- | --- | --- |
| **Customization** | By definition is fully customized to your needs. | Meets most needs based on your requirements but does not necessarily meet them 100 percent. |
| **Continuously evolving for business disruption** | A custom platform can keep up-to-date with the business requirements and optimization and standards. However, every enhancement must undergo development, which has its pros and cons. | A SaaS platform keeps close watch on business disruption and continuously improves its product features. |
| **Integration** | You can design your custom-built software to integrate with any software and with any protocols. | Open APIs allow most SaaS solutions to integrate with a wide range of third-party software. |
| **Cost impact** | It is expensive due to team setup and must go through the entire SDLC process. | It is cheap and based on the subscription cost model. This eliminates up-front cost, and SaaS providers indirectly distribute the development costs across all subscribers. |
| **Delivery and deploy maturity** | Custom platforms have a mature DevOps process with the presence of continuous integration and continuous delivery. Implementation takes its own time. | May not have high DevOps maturity; a SaaS platform allows custom language and platform code storage. The DevOps process is dependent on the built-in tooling of the SaaS provider. Implementation is quick. |
| **Platform upgrades** | Upgrades can be seamless based on requirements. | The upgrades are vendor driven, and sometimes certain vendors are required for upgrades. |
| **Data** | You are owning your data. | You leave your data with the vendor you are subscribed to. |

The chart in Table [2-2](https://learning.oreilly.com/library/view/cloud-native-architecture/9781484272268/html/511610_1_En_2_Chapter.xhtml#Tab2) indicates the extent of various indicators on both the custom and SaaS platforms.

***Table 2-2***

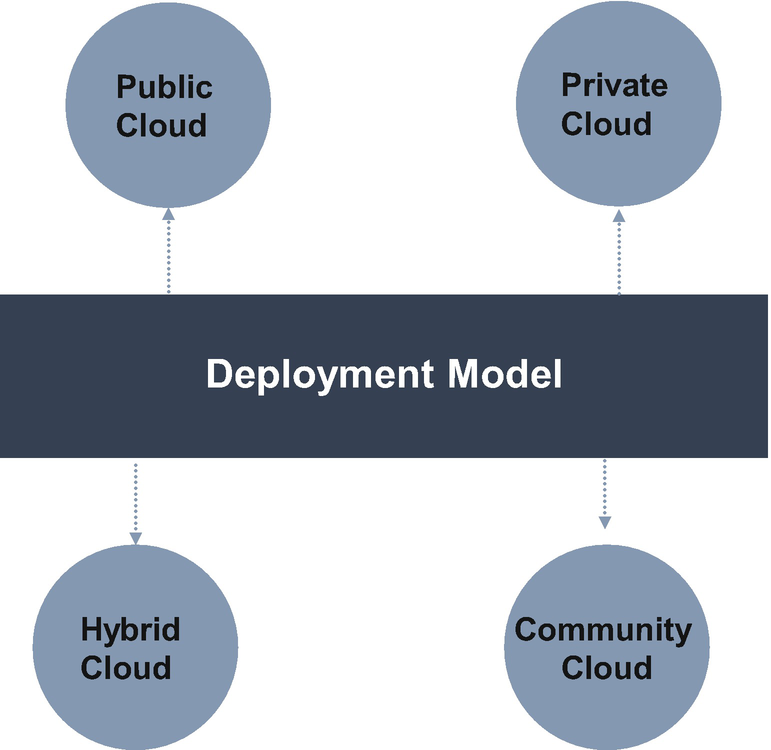
Custom vs. SaaS Platform Indicators

|  |
| --- |
| ../images/511610_1_En_2_Chapter/511610_1_En_2_Figa_HTML.gif |

Both custom and SaaS are good solutions, but you need to evaluate which one is best for you based on your business requirements and other needs. In some cases, SaaS is good. As mentioned, you do not have any operational and management headaches compared to custom, and in some cases customization is good for custom software. It is important to evaluate alternatives on both sides of the development spectrum and choose the option that is best for your business requirements. SaaS services like Gmail, Office 365, etc., are difficult to customize, but it is better to use these SaaS services instead of reinventing the wheel.

**Cloud Computing Deployment Models**

As cloud computing adoption has increased, several different deployment strategies have emerged to help you to meet the specific needs of your enterprise. Each type of cloud service and deployment method provides you with different levels of control, flexibility, and management. Various models can be utilized to deploy the application in production. Understanding different deployment models, as shown in Figure [2-16](https://learning.oreilly.com/library/view/cloud-native-architecture/9781484272268/html/511610_1_En_2_Chapter.xhtml#Fig16), will help you to decide what set of services is right for your needs.



***Figure 2-16***

Cloud deployment model

A cloud deployment model is defined according to where the infrastructure for the deployment resides and who has control over the infrastructure.

Deciding which deployment model you will choose is one of the important organization strategies. Each cloud deployment model satisfies different organizational needs, so it is important to choose a model that meets your strategy. One of the main strategies is based on the value proposition and cost associated with it.

**Public Cloud**

In public cloud computing, the use of computing resources and software is based on the subscription model. The resources are provisioned for open use by the public and various organizations. The architecture of the public cloud is a multitenant type. This kind of architecture allows you to share resources across organizations and applications, but data is isolated from each other, and there will be a stricter security firewall between tenants. A multitenant is like houses on one floor of an apartment, with each apartment separated by walls, but with the common area being shared by all apartments on that floor.

The flexibility, reliability, scalability, and cost are advantages of the public cloud.

**Private Cloud or On-Premises Cloud**

In private cloud computing, the infrastructure is used by a single organization. Such an infrastructure is managed within an organization or dedicated infrastructure in the cloud. Technically, there is no difference between the public and private clouds. We often think that a private cloud is only on-premises, but that assumption is not true; you can deploy your application entirely on a private cloud at various cloud providers and provision exclusive use by a single organization comprising multiple portfolios within an organization.

The private cloud offers some benefits and features the public cloud, but there is no compromise of security on both the public and private clouds. A private cloud may be necessary due to various compliances and regulations like HIPAA, PCI DSS, etc.

The benefits of the private cloud are flexibility in the deployment and customization of infrastructure based on your requirements, but all these come with a cost, so the price of this model is higher than the public cloud.

**Community Cloud**

In community cloud computing, multiple organizations share computing resources that are part of a community. The community must have a shared concern, for example, shared policies, SLAs, shared security requirements, etc. The concern may be owned by one organization or a third-party provider, and it may exist on or off-premises.

**Hybrid**

A hybrid model is a combination or composition of two or more distinct cloud infrastructures, such as private, public, or community. It is a way to connect infrastructure and applications between cloud-based resources and existing enterprise resources that reside in your data center. This model is most adopted across organizations. A hybrid cloud architecture helps organizations to integrate their on-premises and cloud operations to support various use cases using a set of cloud service tools and APIs on-premises and across cloud environments.

**Cloud Services**

Table [2-3](https://learning.oreilly.com/library/view/cloud-native-architecture/9781484272268/html/511610_1_En_2_Chapter.xhtml#Tab3) lists the main feature services of three cloud providers: AWS, Azure, and Google.

***Table 2-3***

Main Services of Top Three Cloud Providers

| Product Type | **AWS** | **Azure** | **GCP** |
| --- | --- | --- | --- |
| **Compute** | EC2 | Azure VM | Compute Engine |
| **Serverless** | Lambda | Azure Functions | App Engine  Cloud Function |
| **Containers** | Elastic Container Service (ECS)  Elastic Kubernetes Service (EKS) | Container Instances  Azure Kubernetes Service (AKS) | Google Kubernetes Engine (GKE) |
| **RDBMS** | Aurora | Azure SQL | Cloud Spanner |
| **NoSQL** | DynamoDB | Cosmos DB | Cloud BigTable |
| **Object storage** | S3 | Blob | Cloud Storage |
| **Caching** | ElastiCache | Azure Redis | Memorystore |
| **Managed database (MySQL/PostgreSQL)** | RDS | Azure Database | Cloud SQL |
| **Event-driven** | SQS, MQ, SNS | Event Hubs | Pub/Sub |
| **Streaming** | Kinesis, Kafka |  | Kafka |
| **Data warehouse** | Redshift, EMR |  | Bigquery |
| **Developer tools** | AWS DevOps |  | GCP Developer Tools |
| **Monitoring and OpsWorks** | CloudWatch, CloudTrail, OpsWork |  | Cloud Monitoring, Cloud Trace, logging |
| **Security** | Identity and access  Cognito  CloudHSM  WAF |  | Cloud key management, workloads |
| **API integration** | Gateway | API management | Cloud endpoints |

Cloud service providers offer all types of services to develop and deploy cloud native applications. Depending on the maturity of the organization and the skills of its employees, starting the cloud native journey might mean leveraging basic services such as the ones mentioned in the table. Adopting these services is a bare-minimum step to developing cloud native applications. You can use a combination of various cloud services to develop a cloud native application. For example, you can use containers, NoSQL, event-driven architecture, monitoring, and API integration to develop a microservices application; all these combinations depend on the use cases and resources within your organization. Along with the major cloud providers, there are various vendors such as Alibaba, IBM, OpenShift, and VMware Tanzu. Lots of financial clients and other industries are embracing OpenShift and Tanzu. Recently I used OpenShift for payment infrastructure.

**Summary**

This chapter covered how infrastructure has evolved from mainframes to the present-day containers. We also looked at some principles behind this evolution that helped you to reduce the infrastructure costs.

To illustrate the evolution of servers, we covered bare-metal servers, VMs, and containers, and we provided the scenarios for which you would adopt them.

We explained types of cloud services such as IaaS, PaaS, and SaaS, and when to use SaaS over custom services, etc. We talked about their usage and provided the details of the featured cloud services of major cloud providers.

We explained the different cloud deployment options like the public cloud, private cloud, community cloud, and hybrid cloud.

The next chapter provides more about the principles of cloud native architecture.