

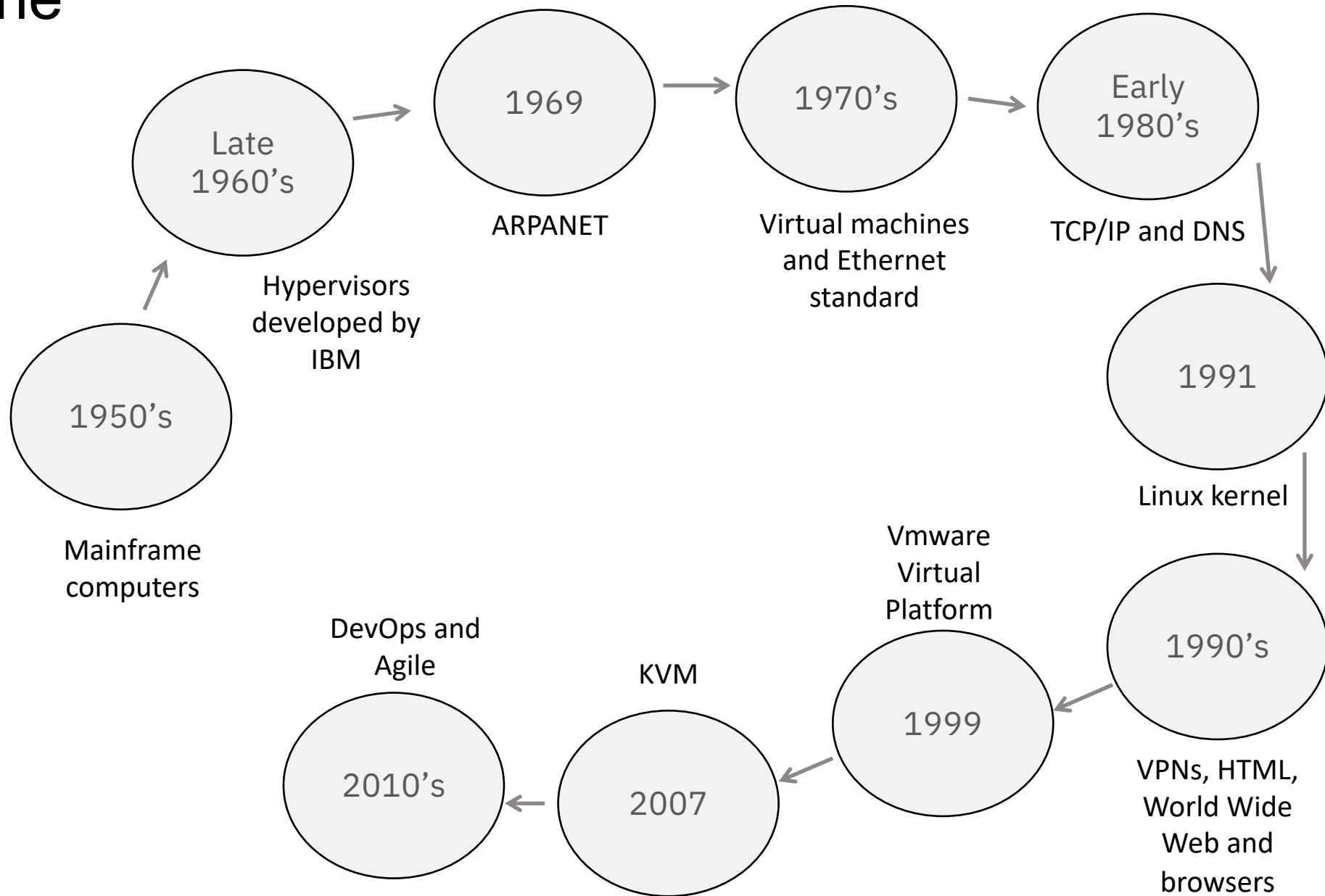
Adopting cloud technologies



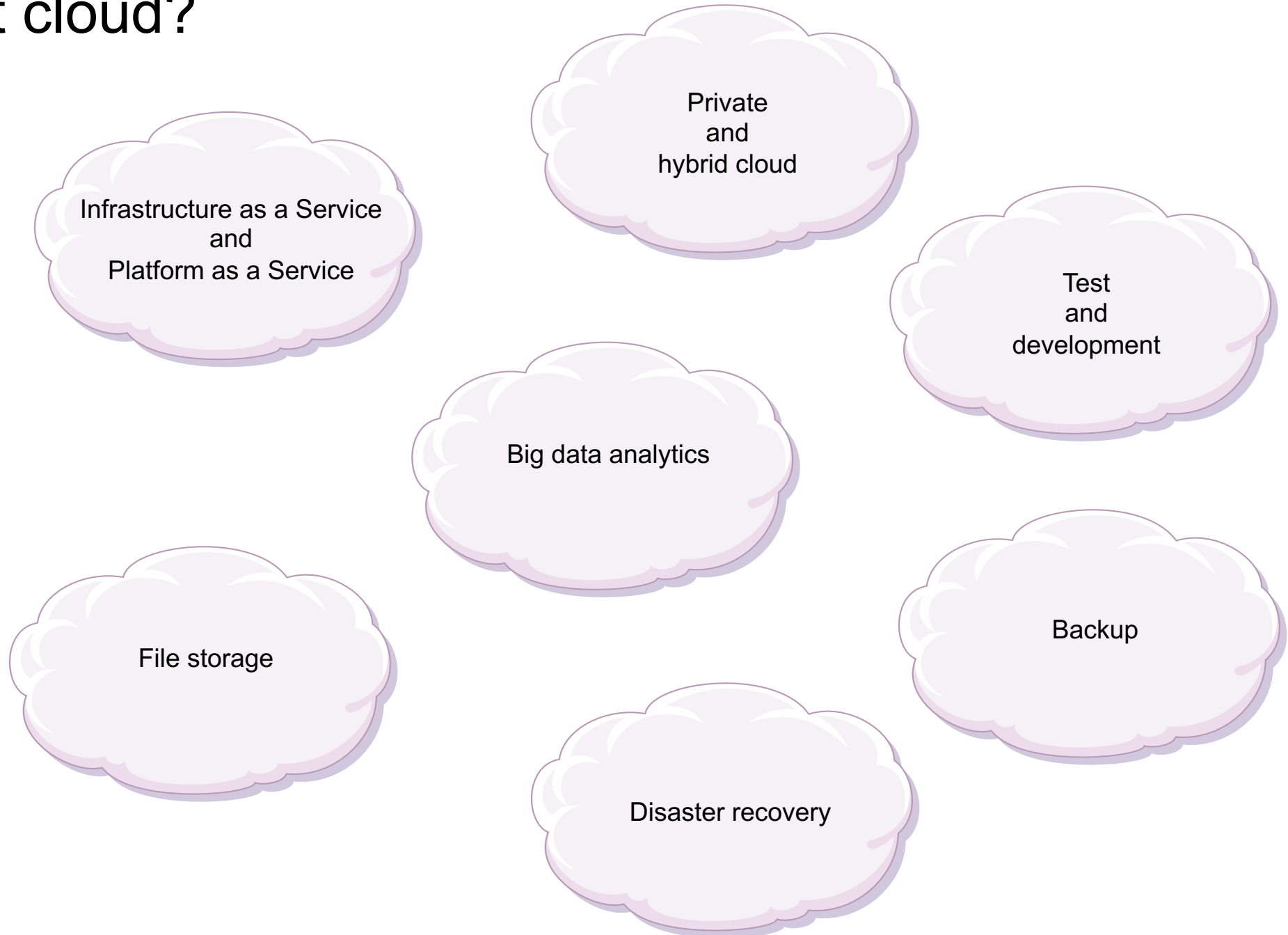
Agenda

- Explain the timeline that led up to enterprises adopting cloud technologies
- Discuss why enterprises adopt cloud technologies
- Describe the following basic cloud concepts:
 - Characteristics
 - Deployment models
 - Service models
- Describe the technologies necessary for the adoption of the cloud
- Discuss cloud computing provisioning

A timeline



Why adopt cloud?



Cloud – the basics

“The cloud” is the delivery of on-demand computing resources — everything from applications to data centers — over the internet on a pay-for-use basis.

- Private
- Public
- Hybrid

Deployment models

- (IaaS) Infrastructure as a Service
- (SaaS) Software as a Service
- (PaaS) Platform as a Service

Service models

The “Cloud”



- On-demand self-service
- Ubiquitous network access
- Resource pooling
- Rapid elasticity
- Measured service

Characteristics

Cloud characteristics

On-demand self-service:

Provisioning or de-provisioning of computing resources as needed in an automated fashion

Ubiquitous network access:

Computing resources can be accessed from anywhere over a network

Resource pooling:

Resources can be dynamically assigned, reassigned or de-allocated as per the requirement

Rapid elasticity: Illusion of infinite computing resources

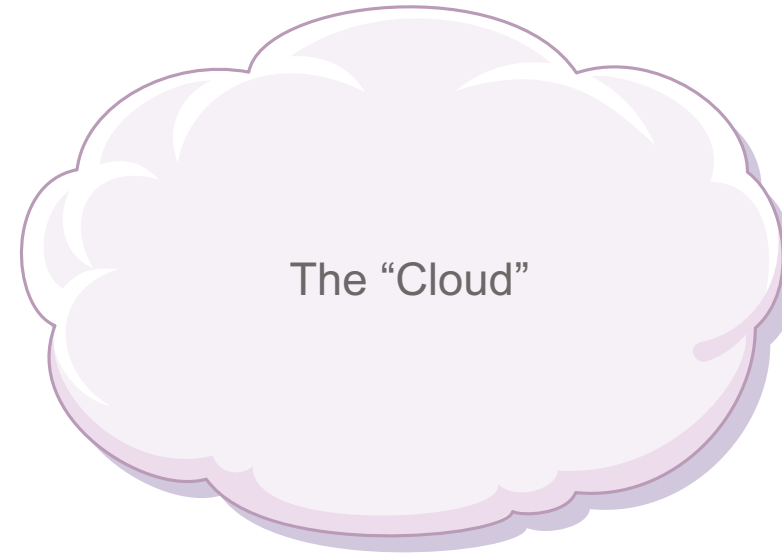
Measures service: Consumers only pay for the computing resources consumed

- Private
- Public
- Hybrid

Deployment models

- (IaaS) Infrastructure as a Service
- (SaaS) Software as a Service
- (PaaS) Platform as a Service

Service models



- On-demand self-service
- Ubiquitous network access
- Resource pooling
- Rapid elasticity
- Measured service

Characteristics

Cloud service models

IaaS: The delivery of computing infrastructure (networking, storage, compute) as a service

SaaS: The hosting of applications by a provider on a cloud infrastructure accessed from thin or thick clients over the network or a program interface

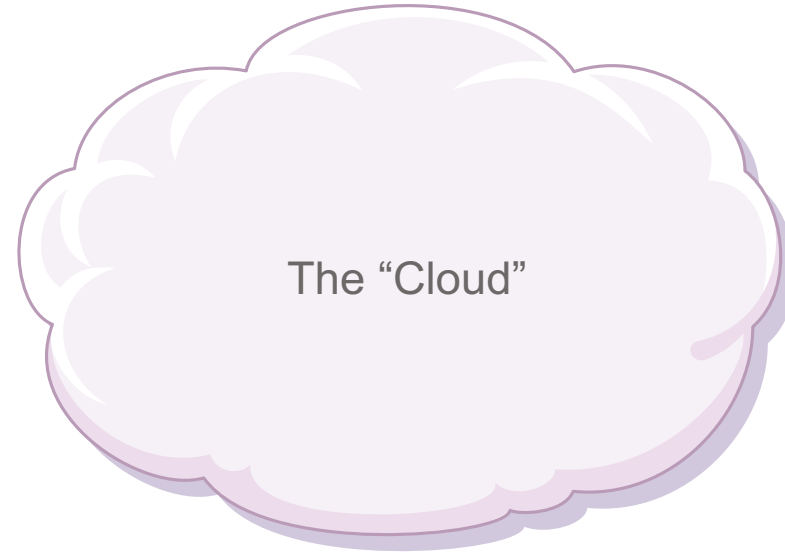
PaaS: The delivery of the infrastructure as well as middleware (databases, messaging engines and so on) and solution stacks for application build, development and deploy.

- (IaaS) Infrastructure as a Service
- (SaaS) Software as a Service
- (PaaS) Platform as a Service

Service models

- Private
- Public
- Hybrid

Deployment models



- On-demand self-service
- Ubiquitous network access
- Resource pooling
- Rapid elasticity
- Measured service

Characteristics

Cloud service models: IaaS

Integrates basic services into one platform for deploying and running applications:

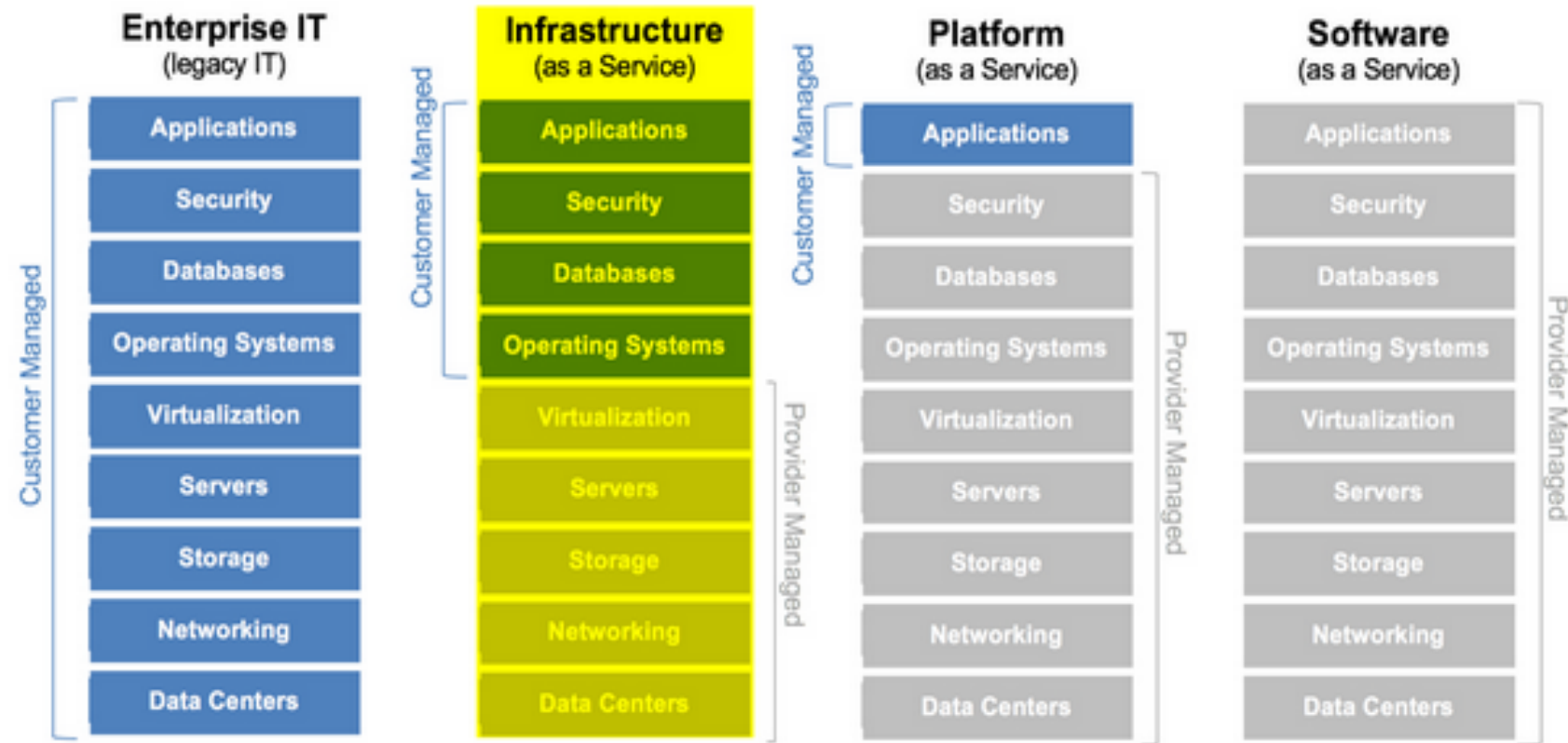
- Virtual servers
- Data storage
- Networking

Basic services are managed by the cloud provider.

Applications, security, databases and operating systems are managed by the customer.

Two primary characteristics:

- Elasticity
- Machine virtualization.



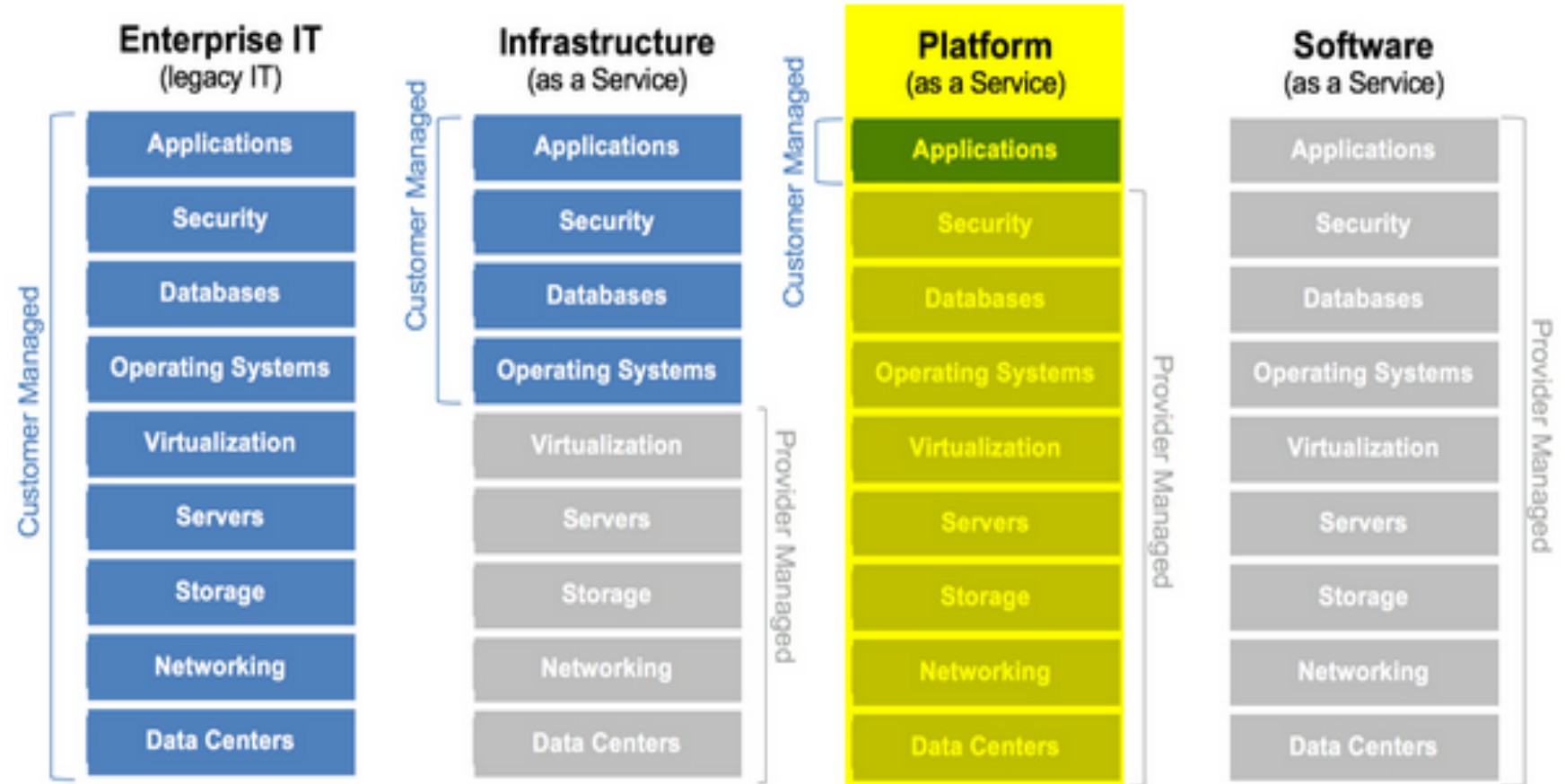
Cloud service models: PaaS

Enables developers to build and deploy web applications on a hosted infrastructure

Resources include everything managed by the provider that is necessary to support the application.

Two main components:

- Computing platform such as Windows or Linux
- Solution stack that consists of the apps used for development and deployment



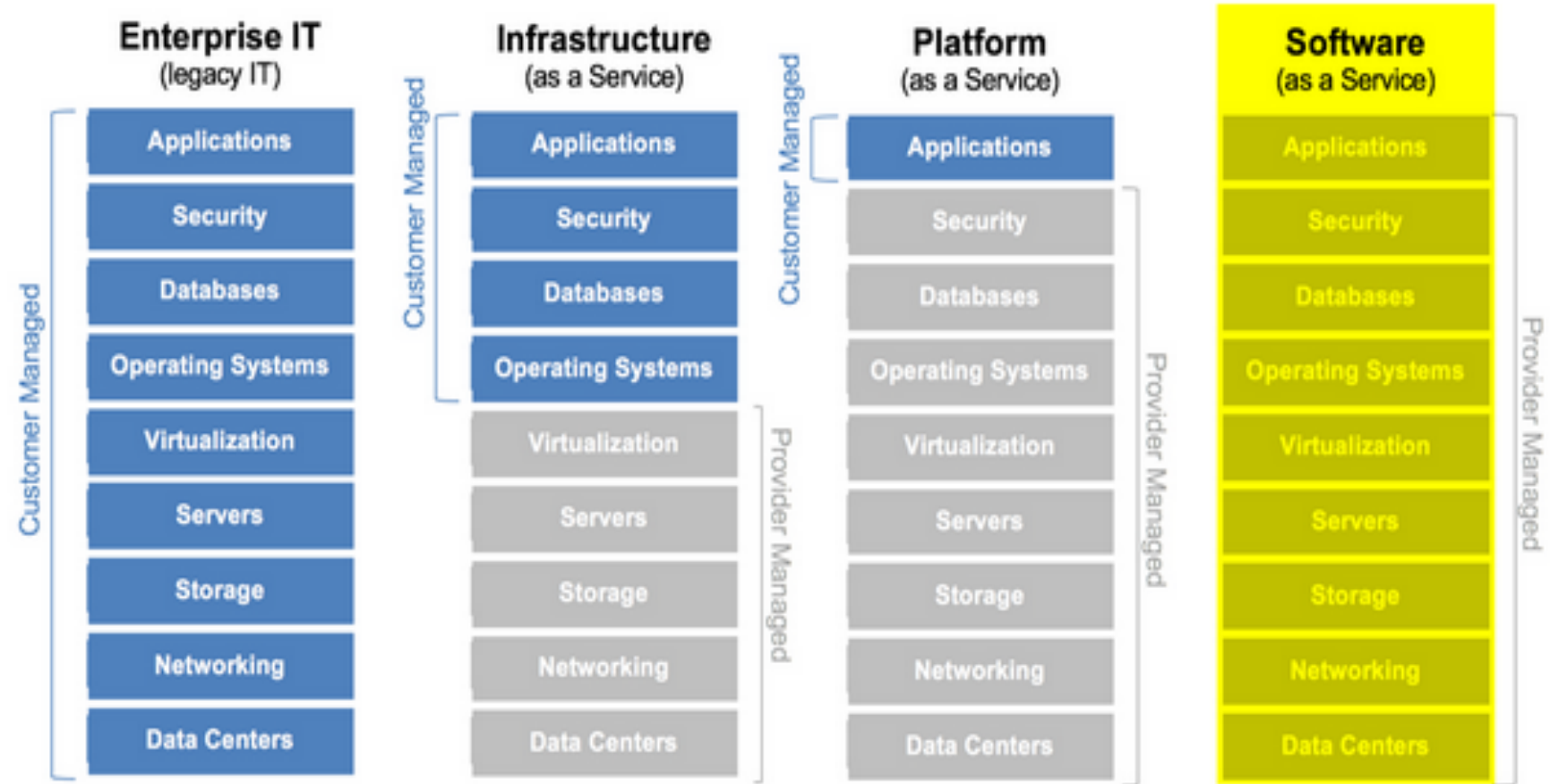
Cloud service models: SaaS

Provides network-based access to commercially available software.

Enables:

- Quick software deployment
- Faster user adoption of software
- Fewer support requirements
- Ease in implementation and upgrades

Common examples include Gmail and Google Docs.



Cloud service models: DBaaS

Software that enables users to provision, manage, consume, configure, and operate database software without having to know about the implementations of specific database software.

Example:

- Provision a MySQL database, manage, configure and operate it using the same set of API calls as if it were an Oracle or MongoDB database.
- Request a backup of the database using an API call or request a MySQL cluster or a MongoDB cluster,
- Resize that cluster without knowing how those operations were performed for each database technology.

Benefits to the enterprise:

- Developer agility
- DBA productivity
- Application reliability, performance, and security



Cloud deployment models

Private: Resources are owned by an organization and maintained either by the organization or a third party.

Public: Resources are provided by a "cloud provider" and used by different organizations through public Internet on a pay as you go model.

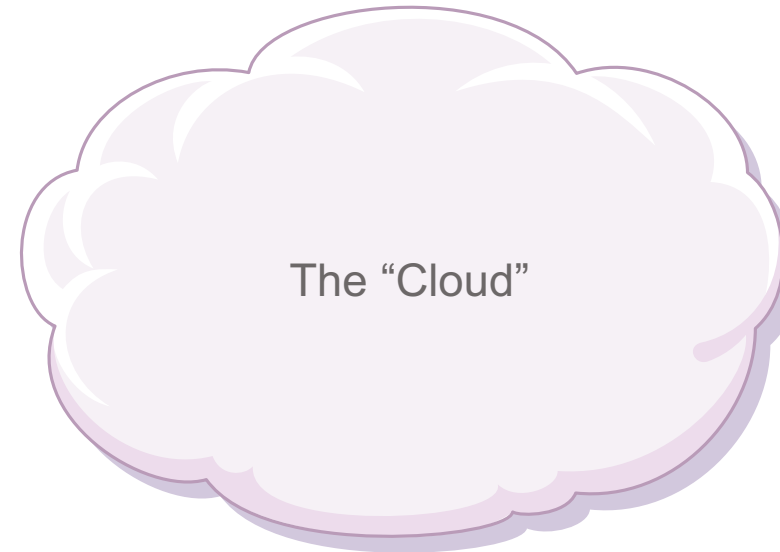
Hybrid: Combination of the public and private deployment models connected by a standardized or proprietary technologies.

- Private
- Public
- Hybrid

Deployment models

- (IaaS) Infrastructure as a Service
- (SaaS) Software as a Service
- (PaaS) Platform as a Service

Service models



- On-demand self-service
- Ubiquitous network access
- Resource pooling
- Rapid elasticity
- Measured service

Characteristics

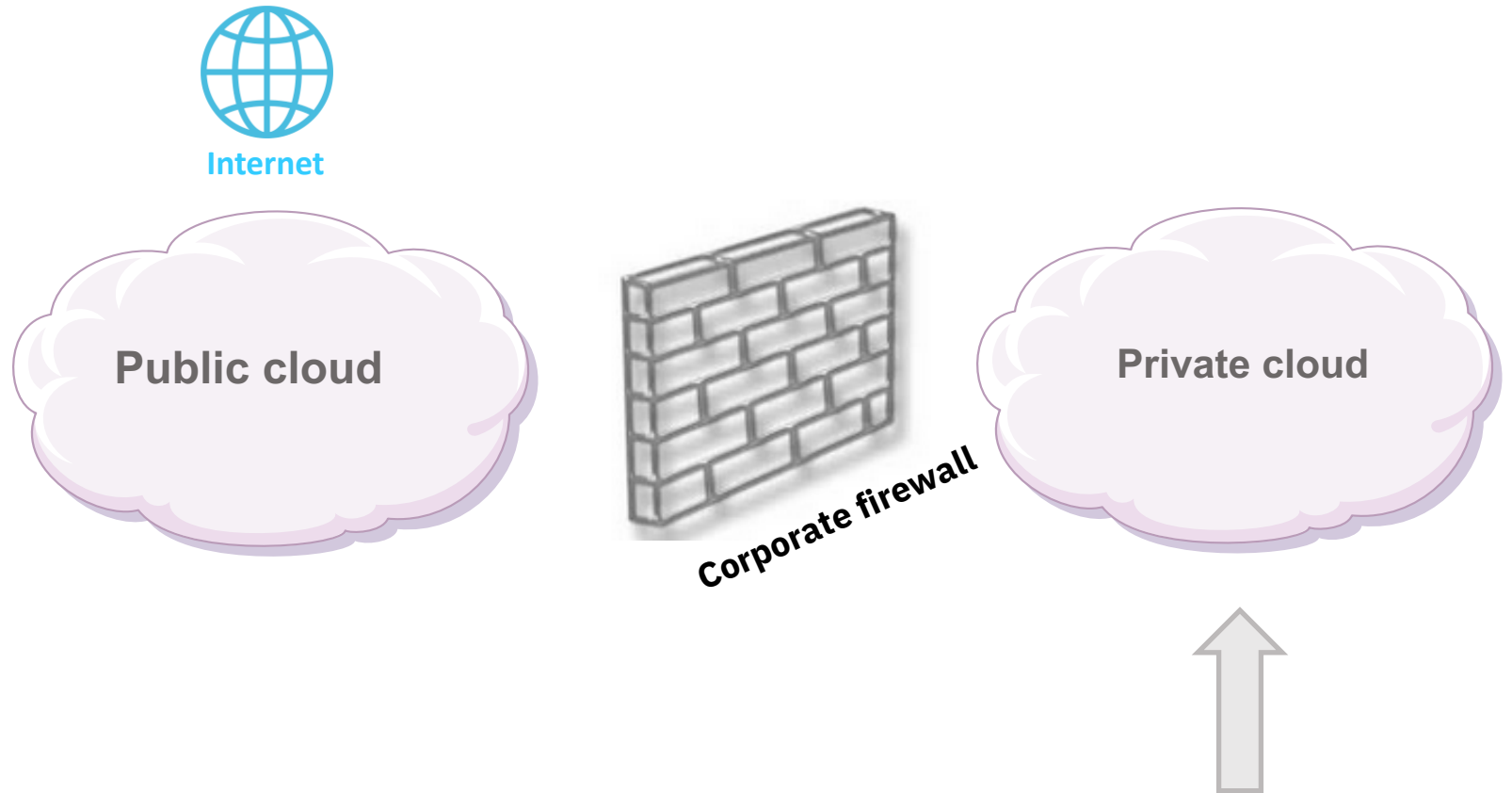
Cloud deployment models: Private

Software-defined data center

Combines essential hardware and other computing resources into a unified virtualized unit

Enables enterprises to scale and provision resources more dynamically than with traditional hardware-centric computing environments

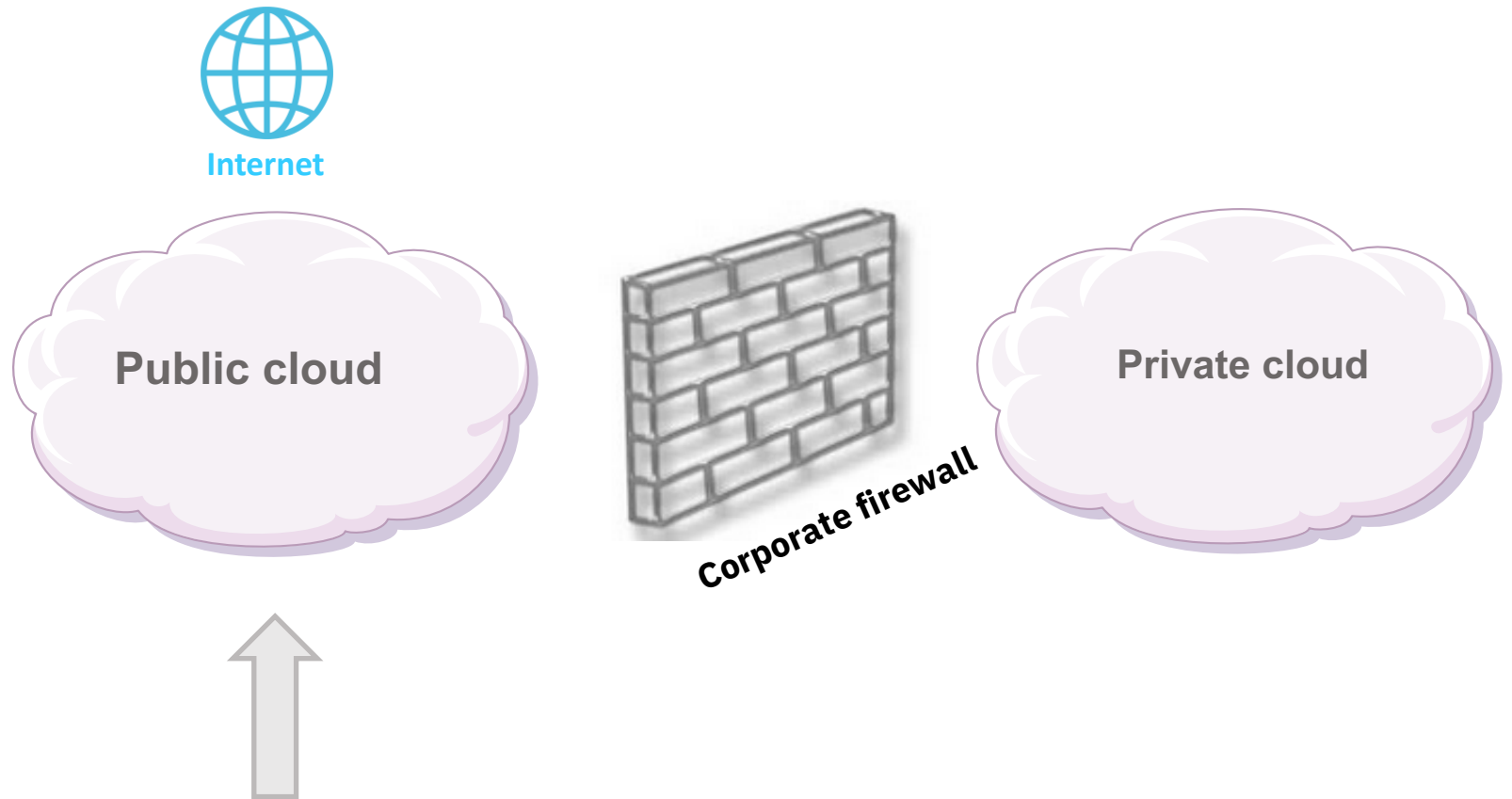
Typically hosted within a company's firewall



Cloud deployment models: Public

Scalable and elastic IT-enabled capabilities are provided as a service

Supports customers that are external to the provider's organization

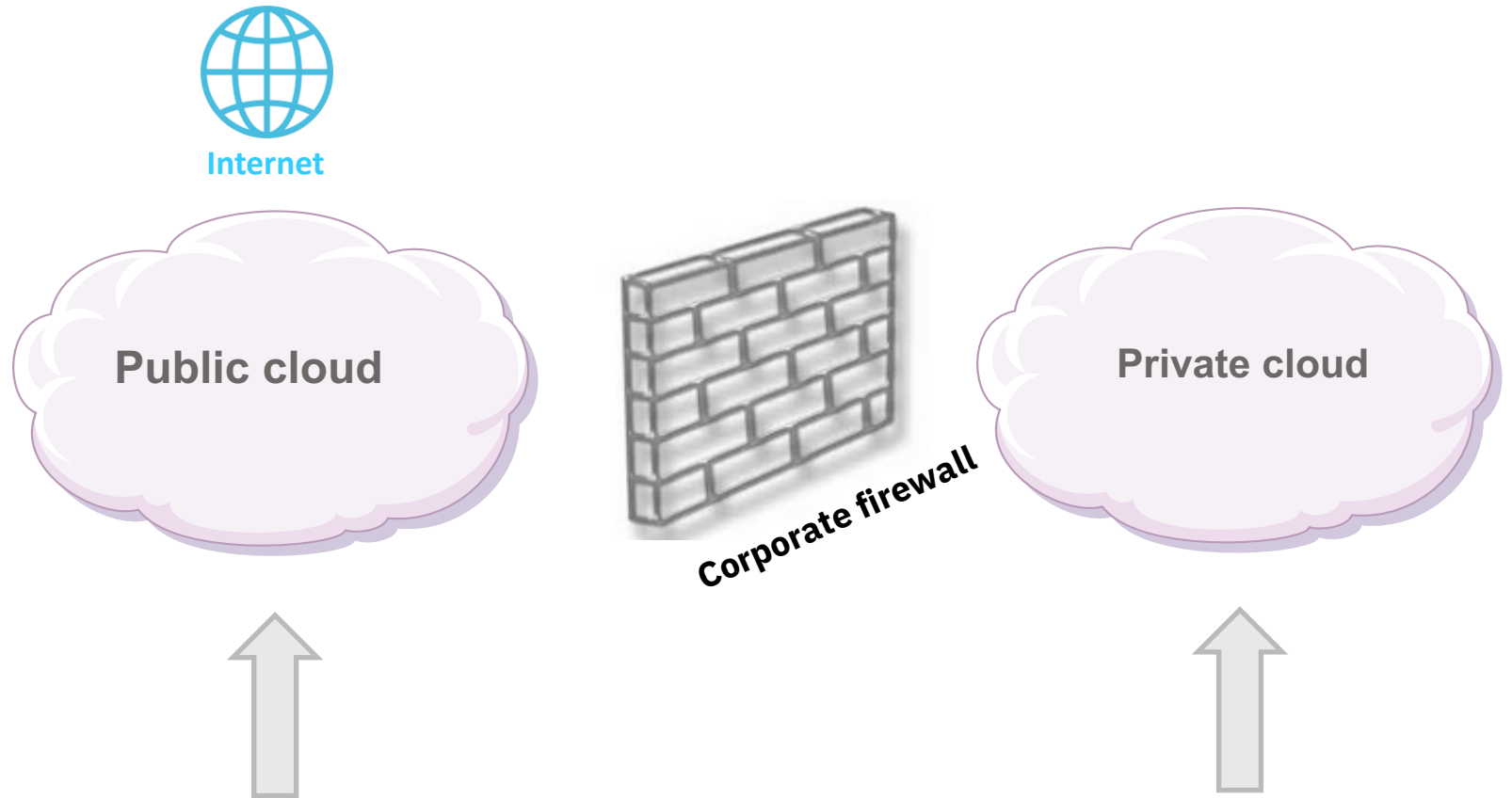


Cloud deployment models: Hybrid

Enterprise architecture that integrates public and private clouds

Maintain some data and applications in their own data centers, often for compliance or security reasons

Integrate with the public cloud services



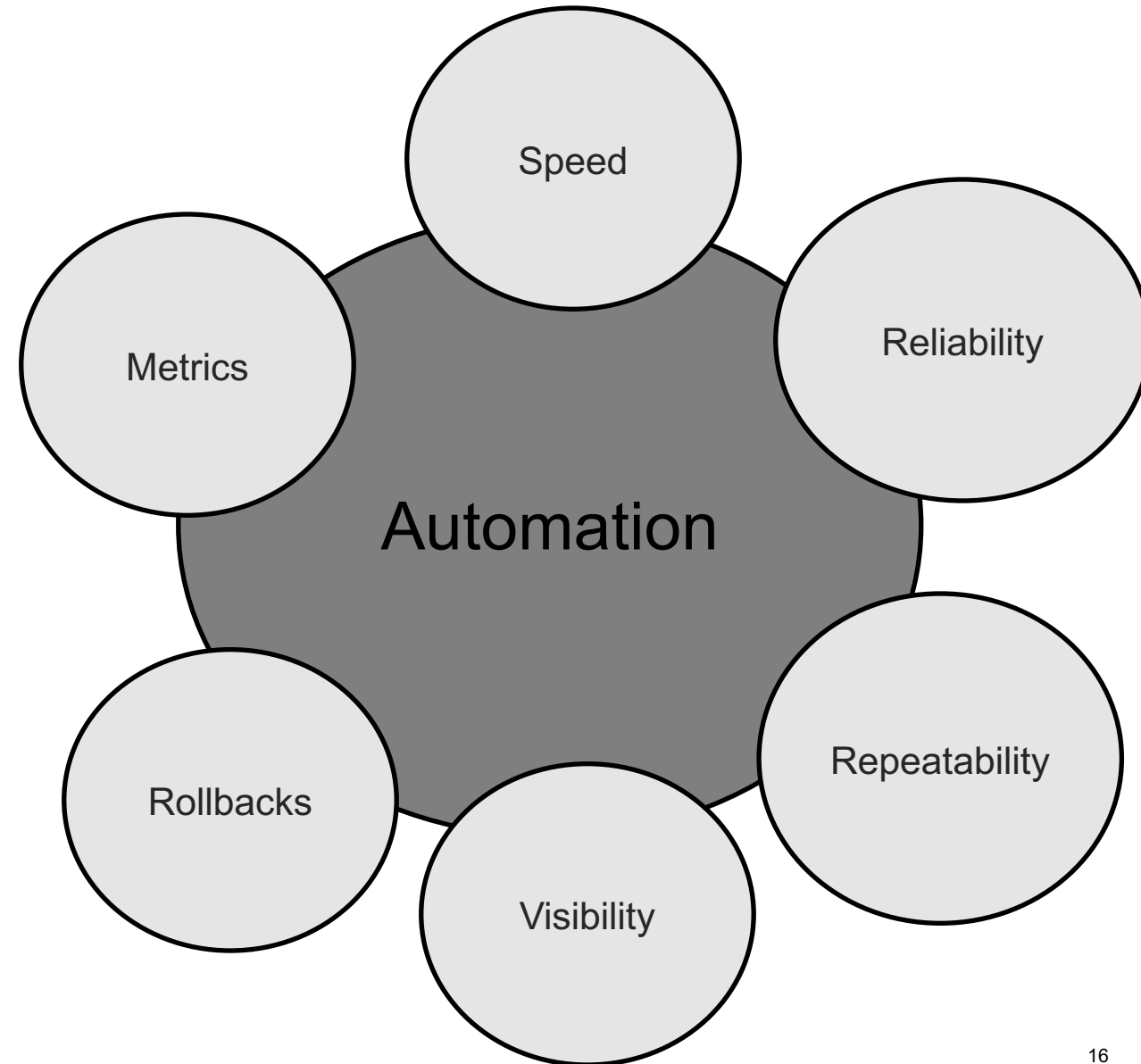
Automation

Automation can address the needs of several architectural layers:

At the **server layer**, the trend toward using highly automated configuration tools that work with large and dense server farms. Cloud automation tools allow you to effectively manage and provision installed capacity.

At the **software layer**, complex systems can be configured once and then rolled out in an agile manner. Intelligent systems architecture can balance the load among compute, network or storage resources, bringing systems online or offline as demand dictates.

Infrastructure automation involves template creation for infrastructure that can be replicated and deployed.

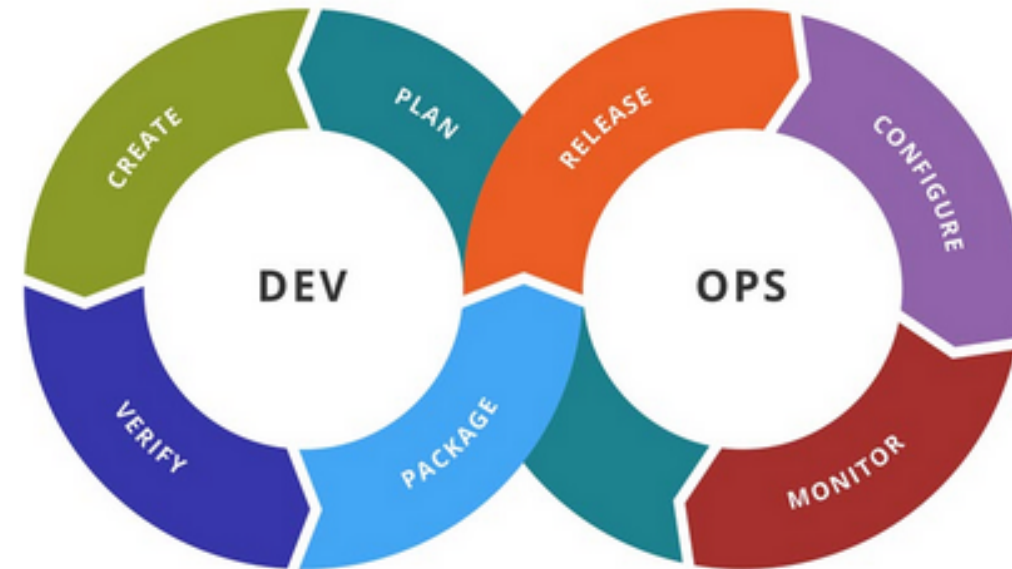


DevOps

... the practice of operations and development engineers participating together in the entire service lifecycle, from design through the development process to production support.

DevOps does not differentiate between different developer and sysadmin sub-disciplines. Generally...

- **Dev** is shorthand for developers, however in practice it means all the people involved in developing the product. Dev can include Product, Test, QA, and other disciplines.
- **Ops** is a term for systems engineers, system administrators, operations staff, release engineers, DBAs, network engineers, security professionals, and various other subdisciplines and job titles.

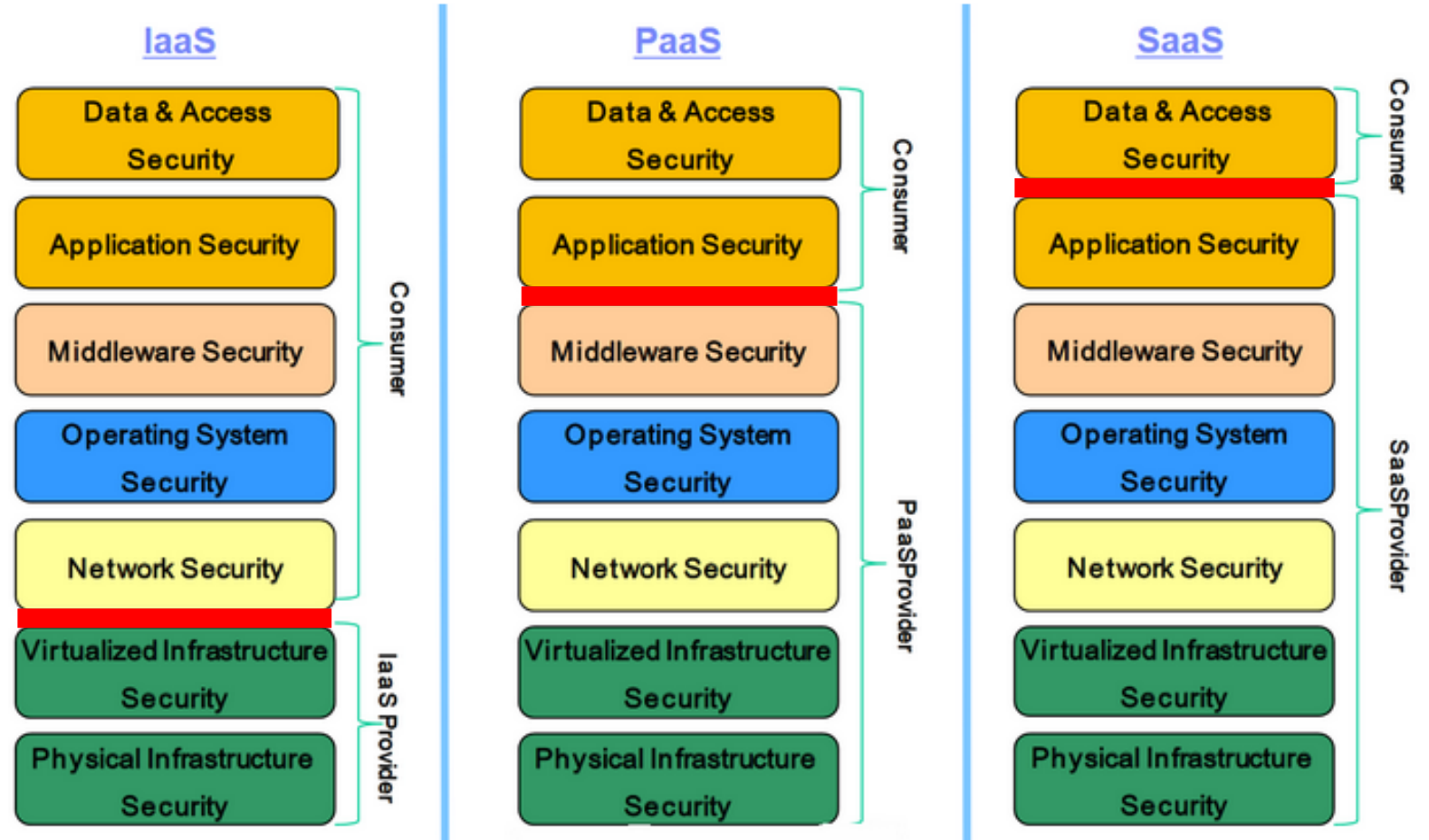


Security

The "security line of responsibility" is shown by a **red line** and indicates where an enterprise (consumer) can invest more money and resources.

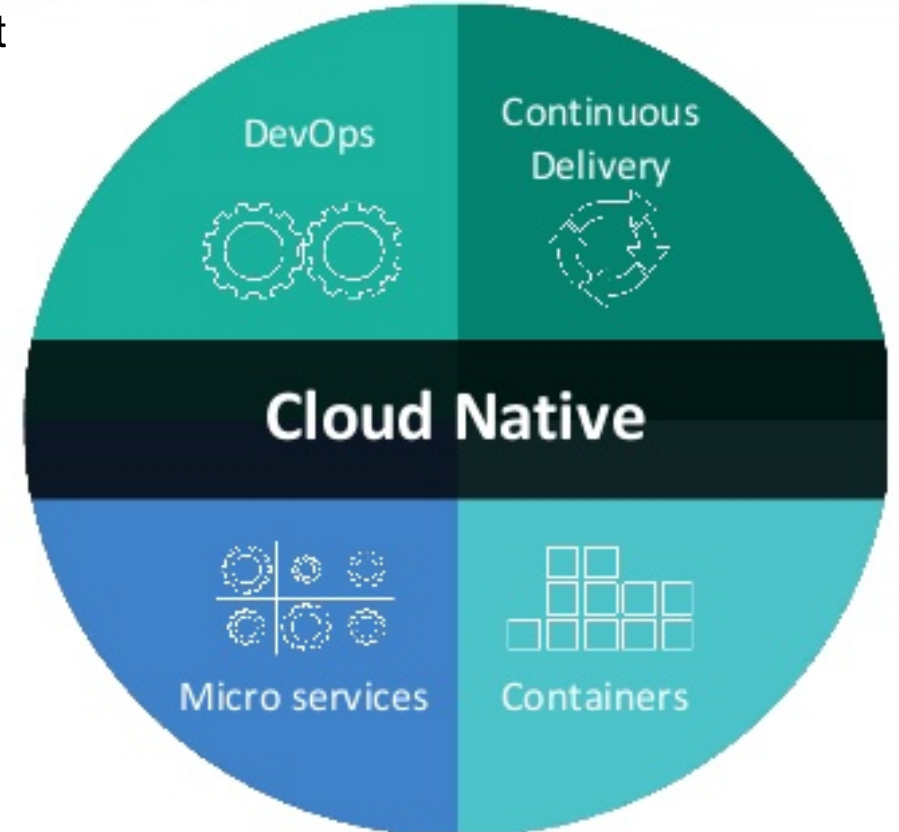
The line varies depending on which cloud service model is utilized.

Example: an enterprise that utilizes the SaaS service model can spend its resources on data security and access security, since the SaaS provider is responsible for securing the stack below the data level.



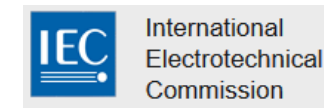
Cloud native

- Cloud native is about the application architecture and design...less about how you code the application
- An approach anchored in container technologies and DevOps practices for building and continuously delivering applications and microservices
- Applications are designed so they are decoupled from any specific physical resource
- Clouds provide an abstraction or virtualization layer between the application and the underlying physical (or virtual) resources, whether they are designed for cloud or not.



Standards

For cloud developers and designers, standards are a powerful addition to their toolbox that allows them to spend more time creating new apps engineered with elegant code, and less time working out compatibility issues.



Resources

Cloud computing

<http://www.redbooks.ibm.com/abstracts/redp5179.html?Open>

<https://www.ibm.com/developerworks/cloud/library/cl-cloudservicemodels/index.html>

Security

https://www.ibm.com/developerworks/community/blogs/ibmsyssw/entry/top_3_fundamental_cloud_security_considerations?lang=en

Cloud standards

<https://www.ibm.com/developerworks/cloud/library/cl-tools-to-ensure-cloud-application-interoperability/index.html>

[National Institute of Standards and Technology \(NIST\)](#)

[Open Cloud Consortium \(OCC\)](#)

[Cloud Standards Customer Council \(CSCC\)](#)

Cloud native

<https://developer.ibm.com/cloud-native/>

Cloud automation

<https://developer.ibm.com/cloudautomation/2017/11/14/intro-cam/>

<https://developer.ibm.com/tv/manage-cloud-automation/>