# Cloud Computing Applications and Services
## (Aplicações e Serviços de Computação em Nuvem)

# Cloud Computing

University of Minho
2022/2023

# Cloud Services

- Cloud services are divided into 3 main abstractions:
  - Infrastructure-as-a-Service (IaaS)
  - Platform-as-a-Service (PaaS)
  - Software-as-a-Service (SaaS)

# Cloud infrastructure

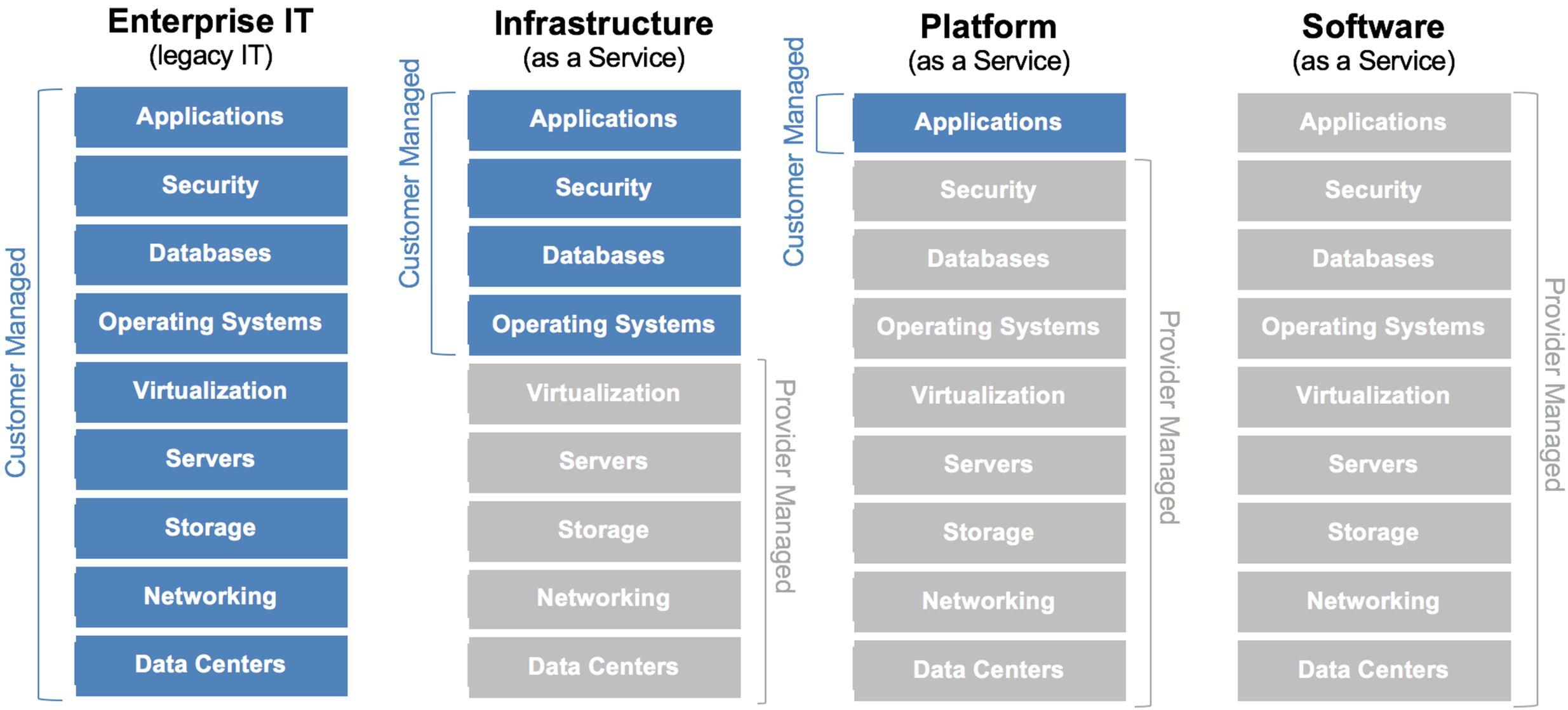
- Infrastructure-as-a-Service (IaaS):
  - provides virtualized hardware resources such as computing, storage and networking
  - resources are allocated on demand and in a pay-per-use fashion
  - examples of IaaS services:
    Amazon EC2 and Google Compute Engine (for computing), Amazon S3 (for storage)

# Cloud infrastructure

- Platform-as-a-Service (PaaS):
  - offers an encapsulation of a development environment abstraction that can be used to develop, deploy and run applications
  - examples include the Google App Engine

# Cloud infrastructure

- Software-as-a-Service (SaaS):
  - features full applications or generic software like databases
  - offered as a service and accessible as a web service or through a web browser
  - SalesForce.com and the Google Apps like Gmail are some well known instances of this type




https://mycloudblog7.wordpress.com/2013/06/19/who-manages-cloud-iaas-paas-and-saas-services/

# Infrastructure-as-a-Service

- Data centers scattered around the world (Asia, Europe and, North and South America)
- Each with around 80 000 servers
- Top players: Google, Amazon and Microsoft

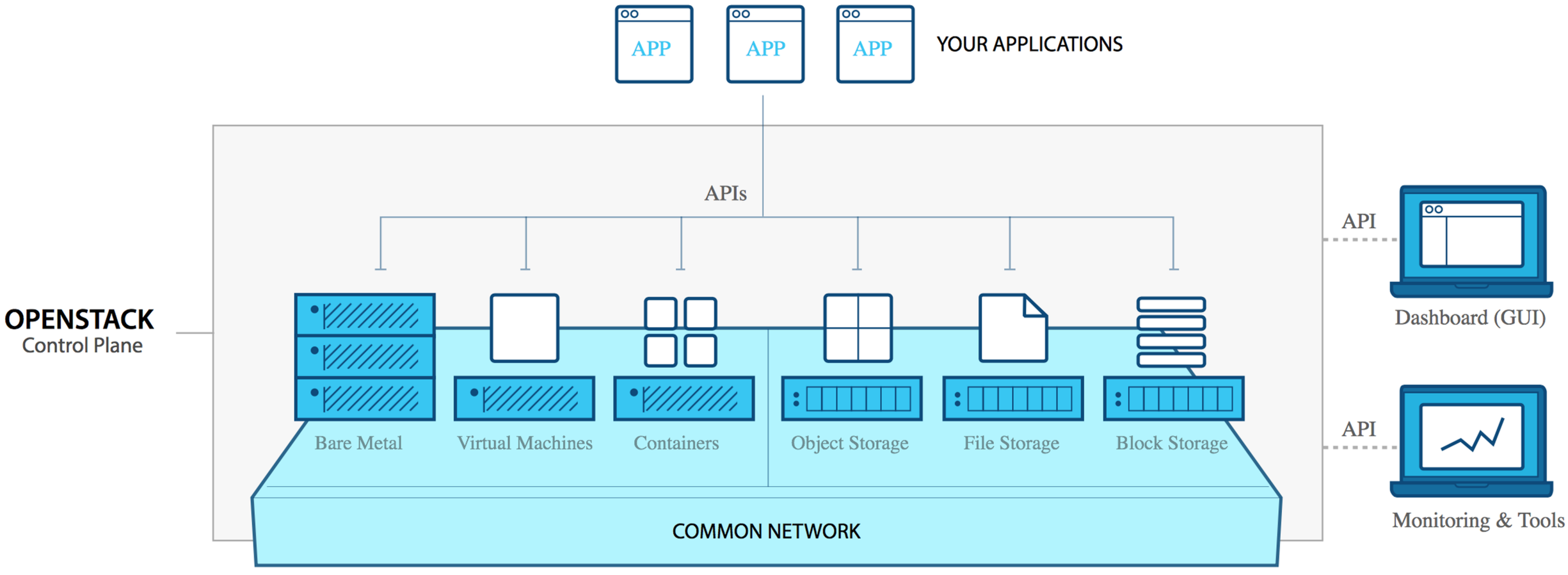
# OpenStack

- Open-source software for creating private and public clouds

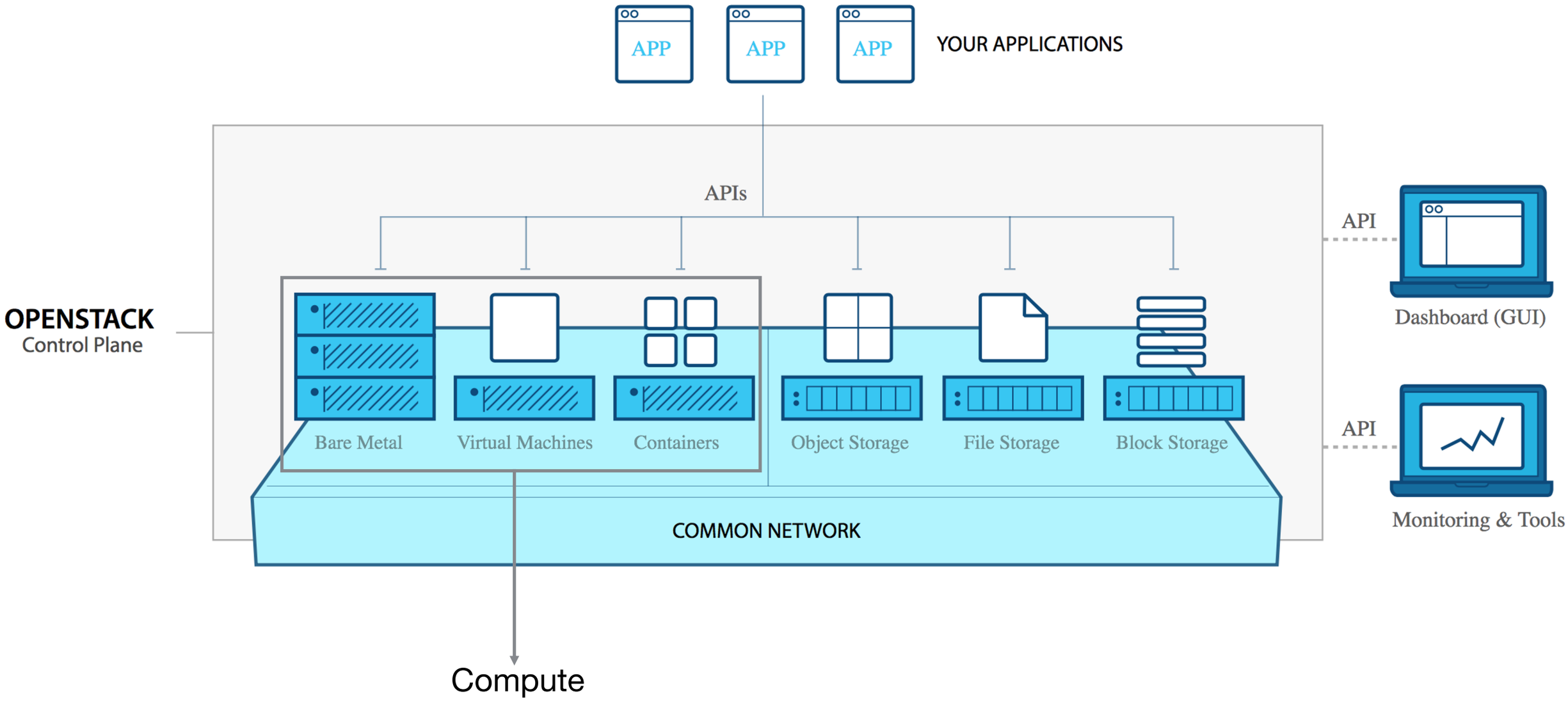
- Controls large pools of compute, storage, and networking resources throughout a datacenter

- Managed through a dashboard or via the OpenStack API

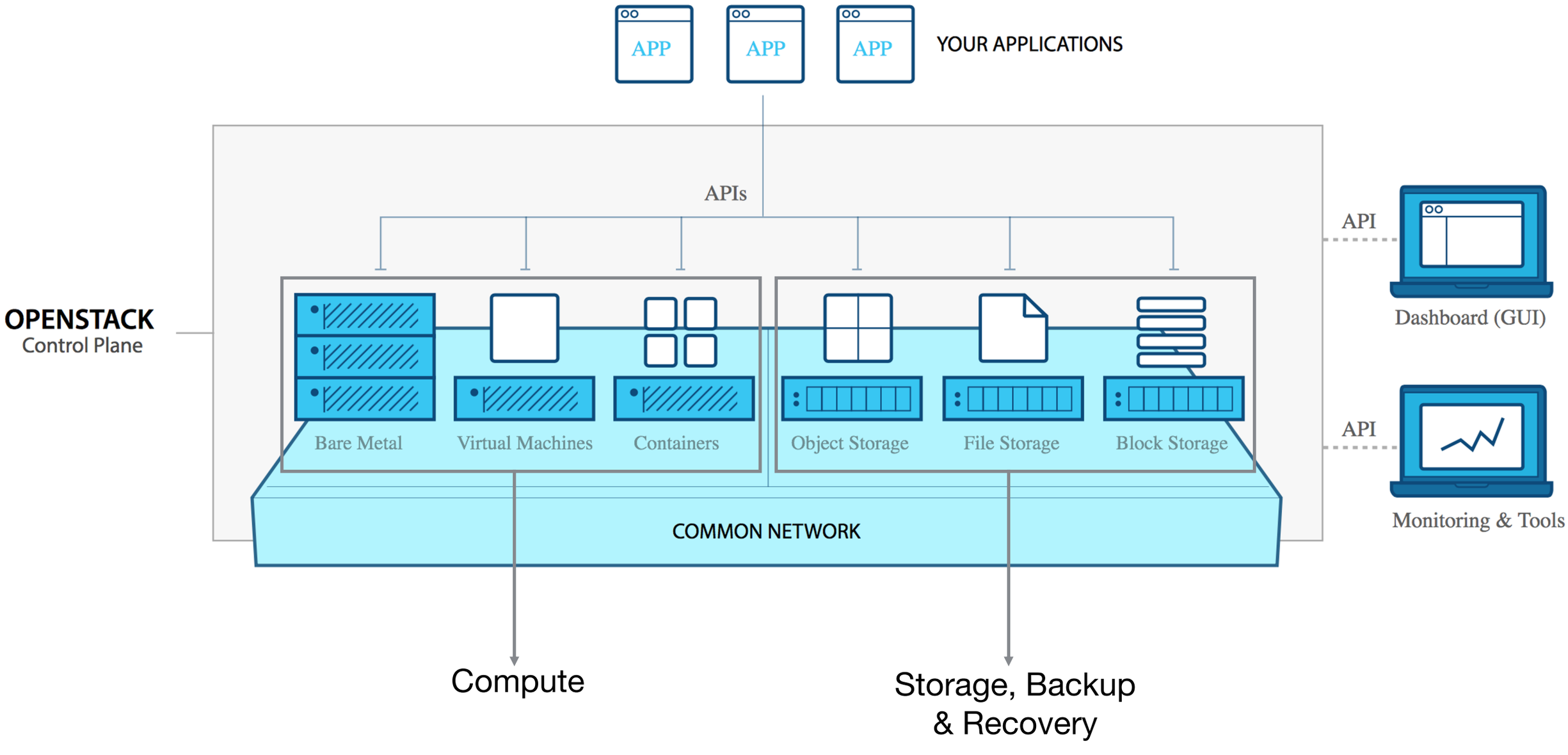
# OpenStack


APP
APP
APP
YOUR APPLICATIONS
APIs
OPENSTACK
Control Plane
Bare Metal
Virtual Machines
Containers
Object Storage
File Storage
Block Storage
COMMON NETWORK
API
Dashboard (GUI)
API
Monitoring & Tools

# OpenStack


APP
APP
APP
YOUR APPLICATIONS
APIs
OPENSTACK
Control Plane
Bare Metal
Virtual Machines
Containers
Object Storage
File Storage
Block Storage
COMMON NETWORK
Compute
API
Dashboard (GUI)
API
Monitoring & Tools

# OpenStack


APP
APP
APP
YOUR APPLICATIONS
APIs
OPENSTACK
Control Plane
Bare Metal
Virtual Machines
Containers
Object Storage
File Storage
Block Storage
COMMON NETWORK
API
Dashboard (GUI)
API
Monitoring & Tools
Compute
Storage, Backup
& Recovery

# OpenStack - Cinder

- Block storage
  - virtualizes the management of block storage devices
  - provides end users with a self service API to request and consume those resources without requiring any knowledge of where their storage is actually deployed or on what type of device

# OpenStack - Swift

- Highly available, distributed, eventually consistent object/blob store
- Ideal for storing unstructured data (VM images, videos) that can grow without bound

# Ephemeral storage vs Cinder vs Swift

| | Ephemeral storage | Block storage | Object storage |
|---|---|---|---|
| Used to… | Run operating system and scratch space | Add additional persistent storage to a virtual machine (VM) | Store data, including VM images |
| Accessed through… | A file system | A block device that can be partitioned, formatted and mounted (such as, /dev/vdc) | REST API |
| Accessible from… | Within a VM | Within a VM | Anywhere |
| Managed by… | OpenStack Compute (Nova) | OpenStack Block Storage (Cinder) | OpenStack Object Storage (Swift) |
| Persists until… | VM is terminated | Deleted by user | Deleted by user |
| Sizing determined by… | Administrator configures size settings, known as *flavors* | Specified by user in initial request | Amount of available physical storage |
| Example of typical usage… | 10 GB first disk, 30GB second disk | 1 TB disk | 10s of TBs of dataset storage |

# Ephemeral storage vs Cinder vs Swift

| | Ephemeral storage | Block storage | Object storage |
|---|---|---|---|
| Used to… | Run operating system and scratch space | Add additional persistent storage to a virtual machine (VM) | Store data, including VM images |
| Accessed through… | A file system | A block device that can be partitioned, formatted and mounted (such as, /dev/vdc) | REST API |
| Accessible from… | Within a VM | Within a VM | Anywhere |
| Managed by… | OpenStack Compute (Nova) | OpenStack Block Storage (Cinder) | OpenStack Object Storage (Swift) |
| Persists until… | VM is terminated | Deleted by user | Deleted by user |
| Sizing determined by… | Administrator configures size settings, known as *flavors* | Specified by user in initial request | Amount of available physical storage |
| Example of typical usage… | 10 GB first disk, 30GB second disk | 1 TB disk | 10s of TBs of dataset storage |

# Ephemeral storage vs Cinder vs Swift

| | Ephemeral storage | Block storage | Object storage |
|---|---|---|---|
| Used to… | Run operating system and scratch space | Add additional persistent storage to a virtual machine (VM) | Store data, including VM images |
| Accessed through… | A file system | A block device that can be partitioned, formatted and mounted (such as, /dev/vdc) | REST API |
| Accessible from… | Within a VM | Within a VM | Anywhere |
| Managed by… | OpenStack Compute (Nova) | OpenStack Block Storage (Cinder) | OpenStack Object Storage (Swift) |
| Persists until… | VM is terminated | Deleted by user | Deleted by user |
| Sizing determined by… | Administrator configures size settings, known as *flavors* | Specified by user in initial request | Amount of available physical storage |
| Example of typical usage… | 10 GB first disk, 30GB second disk | 1 TB disk | 10s of TBs of dataset storage |

# OpenStack - Neutron

- Software-Defined-Network (SDN)
- focused on delivering networking-as-a-service (NaaS) in virtual compute environments
- abstracts the network topology and addressing

# OpenStack - Telemetry

- Monitoring service
  - collects monitoring metrics for physical and virtual resources
  - persists metrics for subsequent retrieval and analysis
  - triggers actions when defined rules are met (e.g., launch more VM instances when a CPU threshold is met)

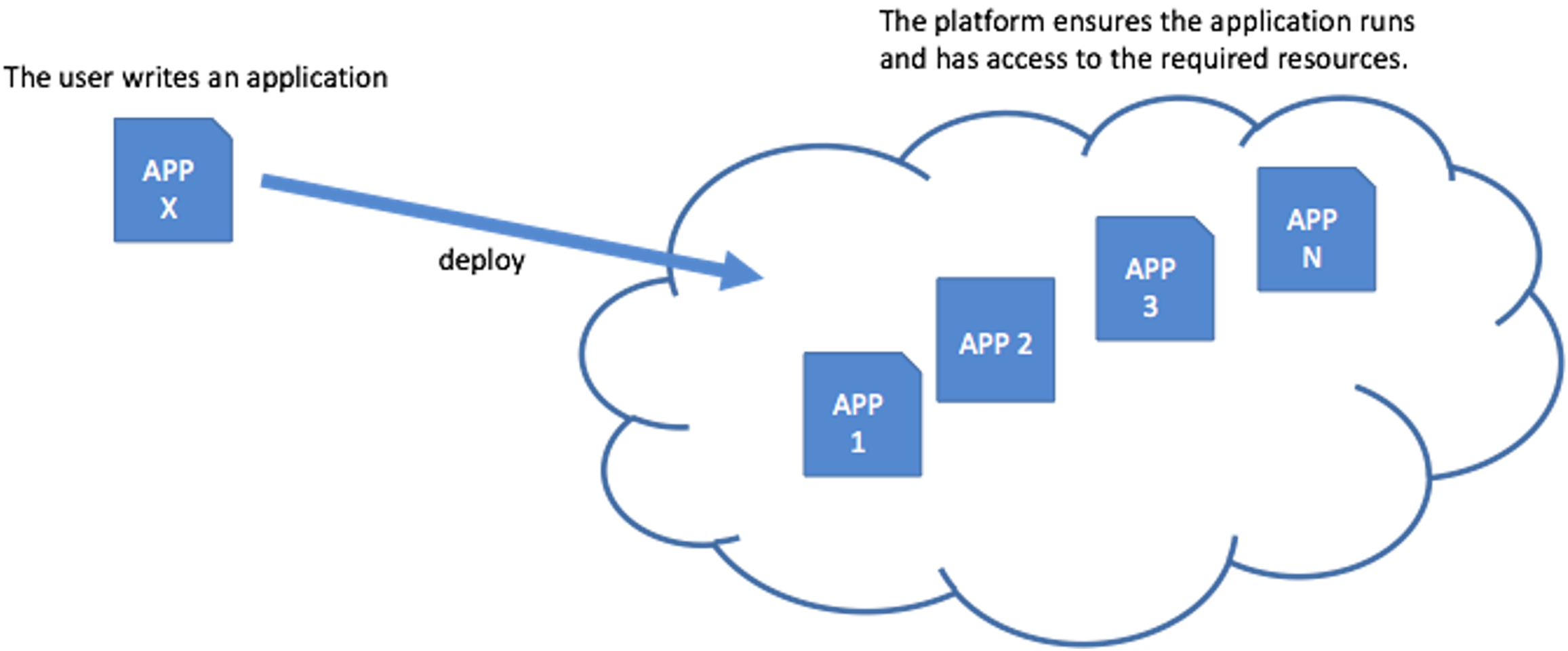
# Counterparts in other services

| OpenStack | Amazon WebServices | Google Cloud Platform |
|---|---|---|
| **Nova Compute** | EC2 | Google Compute Engine |
| **Cinder** | EBS (Elastic Block Storage) | Persistent disks |
| **Swift** | S3 (Simple Storage Service) | Cloud Storage |
| **Neutron** | Networking | Google Cloud Virtual Network |
| **Telemetry** | CloudWatch | Google Stackdriver |

# From IaaS to PaaS

- From managed allocation and provision of resources to managed infrastructure
- Actual resources become transparent
- Focus on the application, which is the deployment item
- The interface is a programming environment, with APIs to IaaS/SaaS services
- The user can focus on the functionality to deploy instead on what are the resource requirements to support it

# PaaS


The user writes an application
APP
X
deploy
The platform ensures the application runs
and has access to the required resources.
APP
1
APP 2
APP
3
APP
N

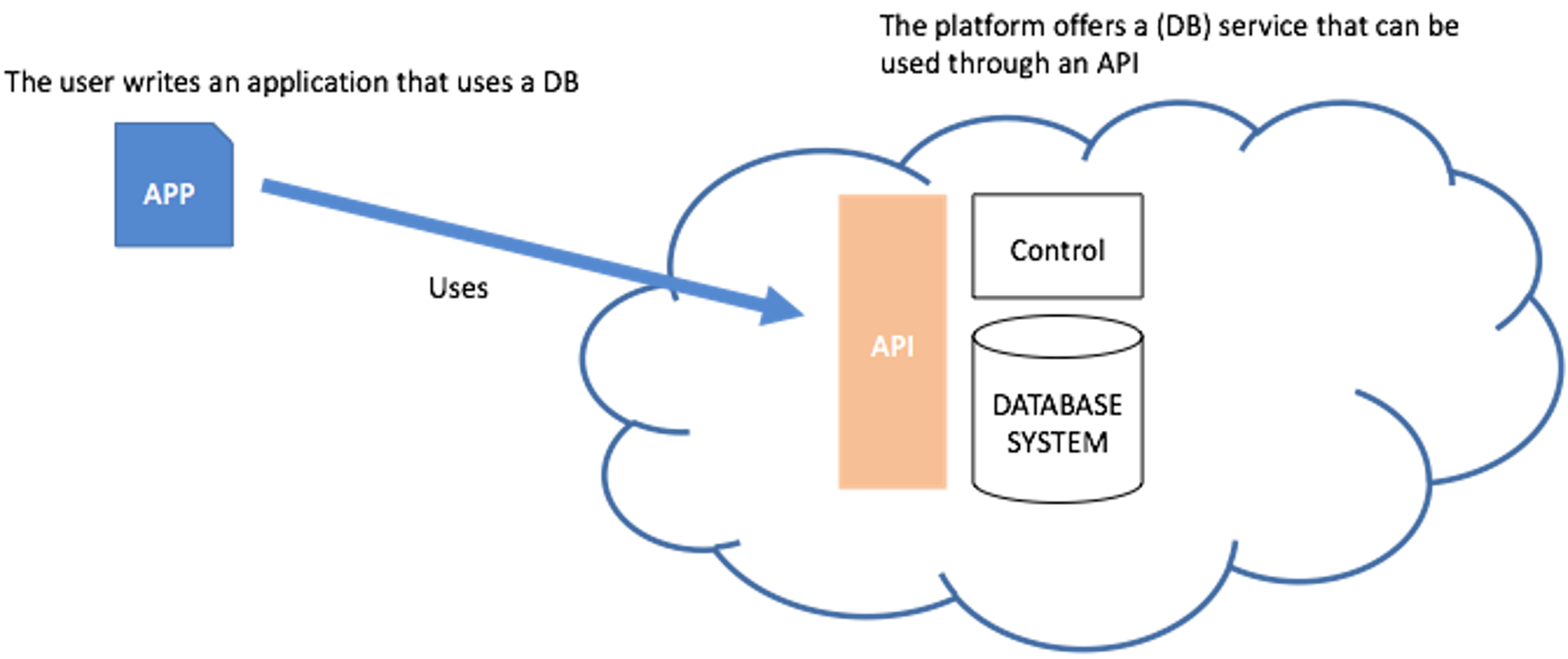
# PaaS

- Example: Google App Engine
  - based on container instances
  - supports multiple languages (Java, Python, PHP, …)
  - and tools (Cloud SDK, IntelliJ IDEA, …)
  - and APIs (Google Cloud Storage, …)
  - versioning, testing, monitoring, logging features

# From PaaS to SaaS

- Specific services are provided
- Managed software components that export their traditional APIs
- Examples are database management systems
- There is no deployment item – the DB is exposed through a client and used as a traditional DB but with minimal configuration needed and with remote access

# SaaS




The application does not necessarily have to run in the same platform.

## Example: Amazon DynamoDB

# IaaS, PaaS and SaaS: complex distributed systems

- Virtualization
- Provisioning
- Monitoring
- Reporting
- Billing
- ...
- Interoperability between all the above

# Summary

Advantages
IaaS, PaaS and SaaS

# Convenience

- From IaaS:
  - avoid upfront costs on infrastructure management and hardware
  - “easily” deploy legacy applications
- to PaaS
  - focus on the application development itself and its requirements
  - powerful development, deployment, debugging, and benchmarking tools already in place
- to SaaS
  - leverage existing components (databases, web/application servers)

# Speed

- From IaaS:
  - infrastructure is already installed and configured
- to PaaS
  - a development framework is already installed and configured
- to SaaS
  - quick integration of different cloud software solutions

# Elasticity[1]

- From IaaS
  - illusion of virtually infinite resources
  - increase and decrease computational power, storage space and other resources according to demand
    - needs to be done manually or by resorting to third-party tools
- to PaaS and SaaS
  - no need to manually manage elasticity

[1]Capacity to scale up or down resources (e.g., computational, storage, network) according to applications' requirements.

Summary

Disadvantages

IaaS, PaaS and SaaS

# Loss of control

- From IaaS
  - no control over specific hardware and virtualization software
  - no possibility of fine tuning and optimizing the infrastructure
- to PaaS
  - no control over specific hardware and the PaaS platform
  - management, fine tuning, and monitoring is reduced to the tools provided by the platform
- to SaaS
  - third-party cloud applications

# Security

- IaaS, PaaS and SaaS
  - as secure as the provider – any vulnerability of the provider is a vulnerability of the application
  - fixes to vulnerabilities must be done by the provider
  - if (even if unlikely) the provider fails, the application fails and recovery is out of the control of the application owner
  - Data privacy at third party infrastructures

# Further reading


- T. Erl, R. Puttini e Z. Mahmood. **Cloud Computing: Concepts, Technology and Architecture**. Prentice Hall, 2013

- M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. 2010. **A view of cloud computing**. Commun. ACM 53, 4 (April 2010), 50-58.

# Questions?