Cloud Computing Applications and Services

(Aplicações e Serviços de Computação em Nuvem)

Cloud Computing

University of Minho

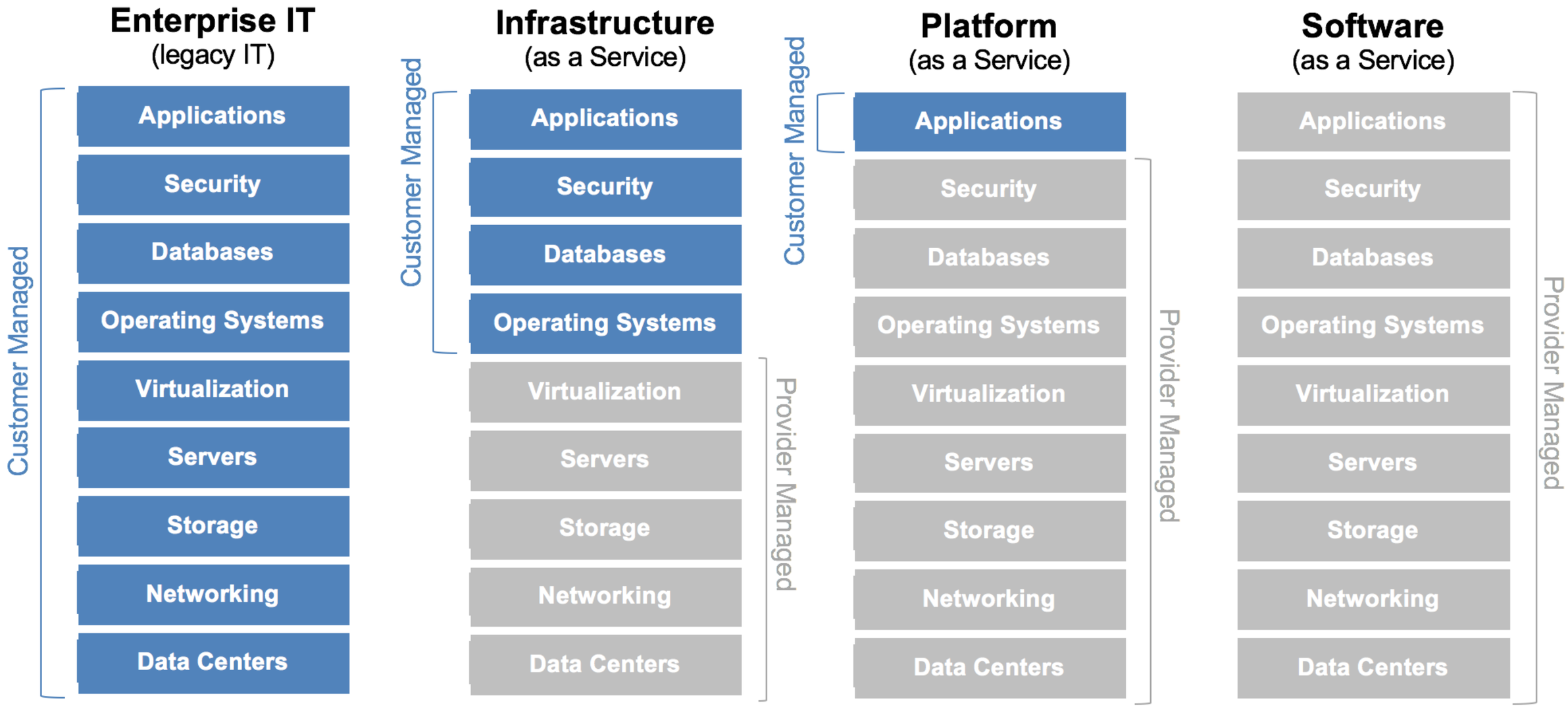
2026-2027



Cloud Services

Abstractions

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



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

IaaS
Infrastructure

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



IaaS
Goals

- Provides virtualized hardware resources such as **computing**, **storage** and **networking**
- Resources allocated **on demand** (i.e., as users require them) and in a **pay-per-use** fashion (i.e., users only pay for resources being actually used)
- Examples of IaaS services:
  - Amazon EC2 and Google Compute Engine (computing)
  - Amazon S3 (storage)

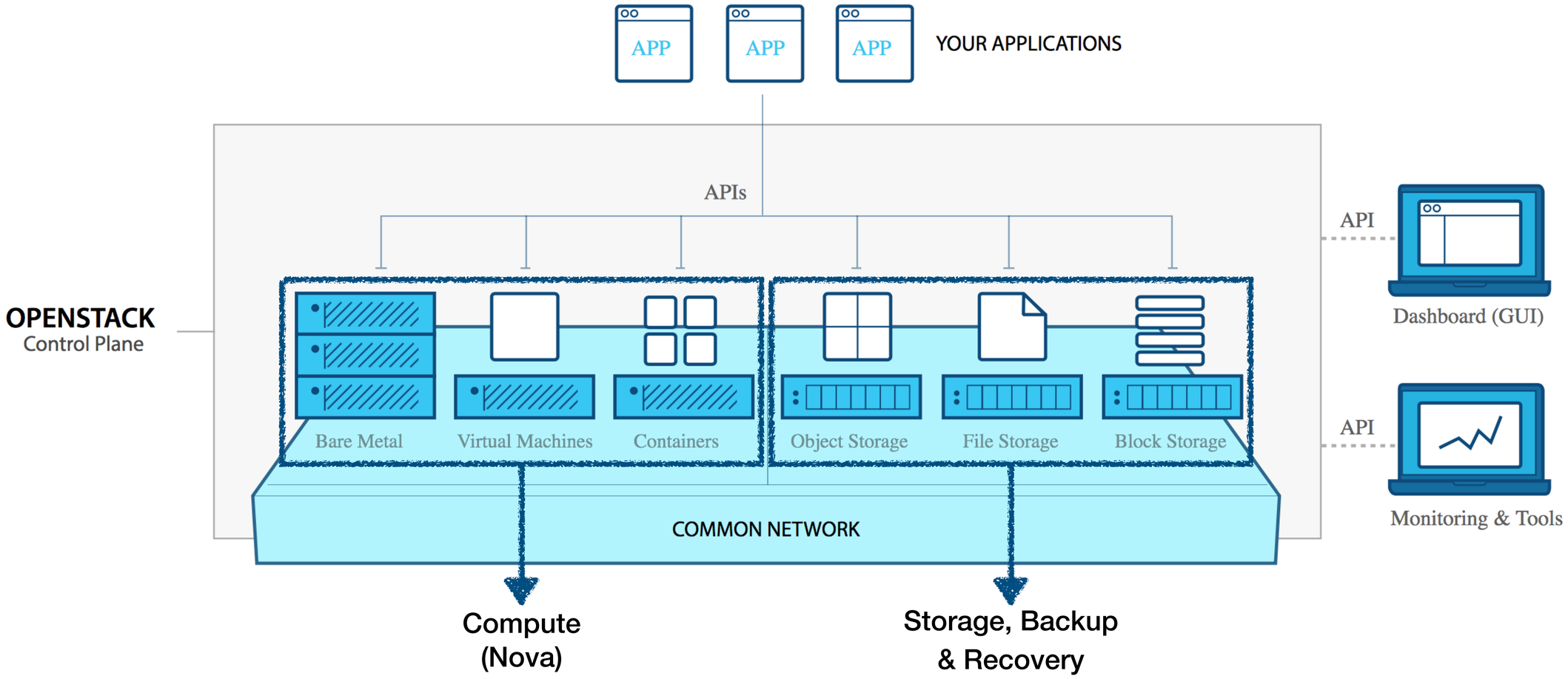
OpenStack

Example

- 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
Compute
(Nova)
Storage, Backup
& Recovery

OpenStack Compute

Nova service

- Provides different types of compute instances
  - Baremetal Servers
  - Virtual Machines (VMs)
  - Containers
- Requires integration with different other OpenStack services
  - **Keystone:** identity and authentication services
  - **Glance:** compute image repository (e.g., VM images)
  - **Neutron:** provides virtual or physical networks for compute instances
  - **Placement:** tracks available hardware resources in a cloud and allocates them, for instance, when creating virtual machines

OpenStack Storage
Cinder, Manila, and Swift services

- Compute instances have ephemeral storage (i.e., deleted when instances are terminated) provided by the OpenStack Nova service
- For persistent storage, several services exist
  - **Cinder:** block storage services
    - Available in the form of volumes (e.g., for Virtual Machines)
  - **Manila:** file system services
    - Ideal for sharing files/directories among compute instances
    - Similar to a Network File System (NFS) service
  - **Swift:** object storage services
    - Key-value API (imagine a distributed hash map…), ideal for unstructured data such as videos, images, …
    - Highly scalable and dependable

Ephemeral, Cinder, and Swift Storage

Comparison

	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 Networking and Monitoring

Neutron and Telemetry services

- **Neutron:** delivers Networking-as-a-Service (NaaS) in virtual compute environments
  - Follows Software-Defined-Network (SDN) principles
    i.e., dissociates network packets forwarding from routing decisions
- **Telemetry:** monitoring for cloud infrastructures and virtual resources
  - Collects monitoring metrics from physical and virtual resources
  - Persists metrics for subsequent retrieval and analysis
  - Triggers actions when defined rules are met
    - *Example*: launch more VM instances when a CPU threshold is met

Counterparts in Other Services

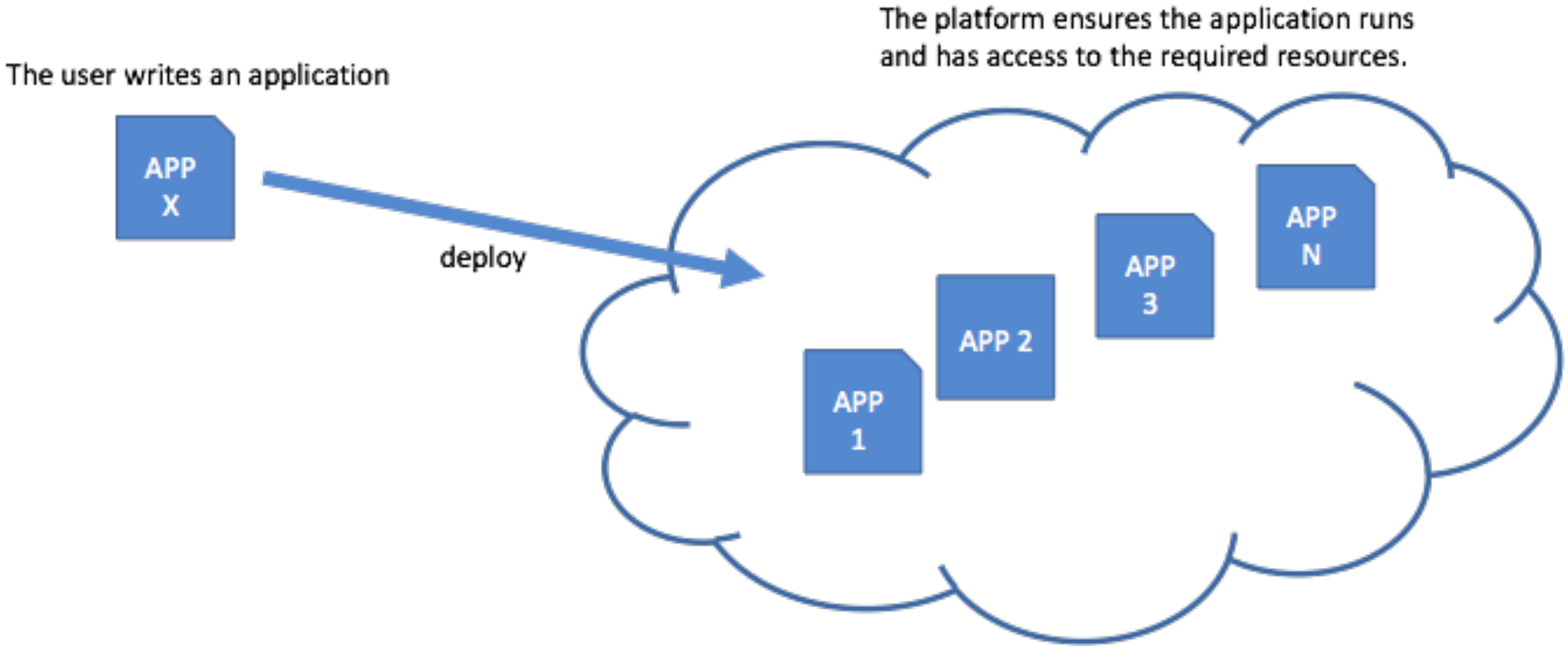
OpenStack	Amazon WebServices	Google Cloud Platform
Nova Compute	EC2	Google Compute Engine
Cinder	EBS (Elastic Block Storage)	Persistent disks
Swift	S3 (Simple Storage Service)	Cloud Storage
Neutron	Networking	Google Cloud Virtual Network
Telemetry	CloudWatch	Google Stackdriver

PaaS

Main differences from IaaS

- Users can focus on the functionality to deploy instead on what are the physical/virtual resource requirements to support it
  - Resources become transparent, while the focus is on the application, which is the deployment item
  - **Transparent** in this context means that users do not need to actually know the exact physical/virtual hardware resources where the application is deployed
- Offers a **development environment** that can be used to develop, deploy and run applications
  - The interface is typically a programming environment supporting different languages (e.g., python, Java, …) and APIs to use other cloud services

PaaS
Overview



PaaS
Example: Google App Engine

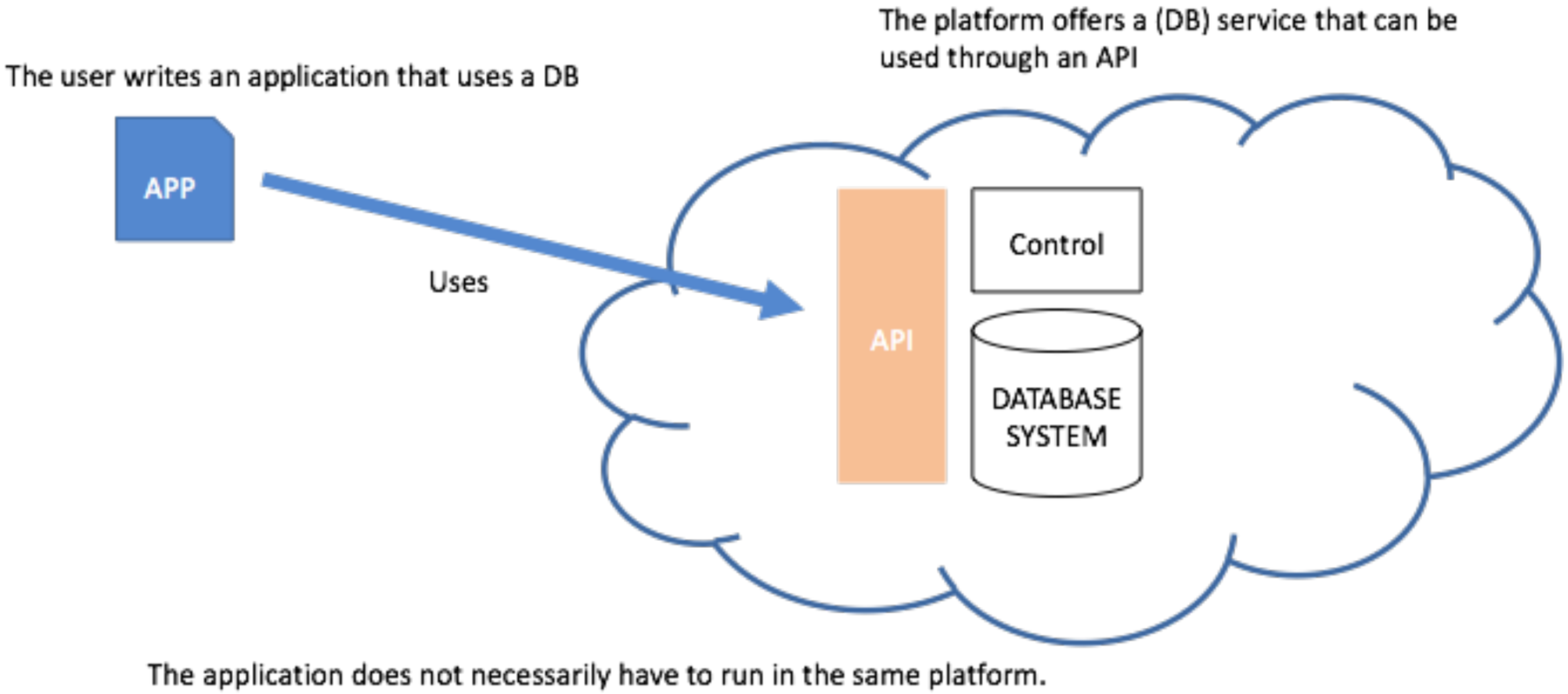
- Based on container instances
- Supports multiple languages (Java, Python, PHP, …)
  - And tools (Cloud SDK, IntelliJ IDEA, …)
  - And APIs (Google Cloud Storage, …)
- Versioning, testing, monitoring, and logging features

SaaS

Main differences from IaaS and PaaS

- Features full applications or generic software offered as a service
  - Some are software components that export their traditional APIs
  - Others are accessible as web services or through a web browser
- Examples
  - **Database (DB) 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
  - **SalesForce.com** and **Google Apps,** like Gmail, are well-known instances of web-based services
  - AI and Data analytics platforms, such as **Amazon SageMaker**

SaaS
Overview



Example: Amazon DynamoDB

IaaS, PaaS and SaaS

Complex Distributed Systems

- Managing (*cloud provider* task) and using (*cloud user* task) these services requires a deep understanding of several concepts and technologies
  - Distributed Systems
  - Virtualization
  - Networking
  - Storage
  - Provisioning
  - Monitoring
  - Reporting
  - Billing
  - …
  - And, interoperability between all the above…
- Our next classes will address some of these :)

Summary

Advantages

IaaS, PaaS and SaaS

Convenience

Developing, deploying and maintaining user applications

- **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

Developing, deploying and maintaining user applications

- **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]

When deploying and maintaining user applications

- **From IaaS**
  - Illusion of virtually infinite resources (computation, storage, …)
  - Users may increase/decrease computational power, storage space, …, according to demand
    - Done manually or by resorting to third-party tools
- **To PaaS**
  - The elasticity of computational, storage, and network resources is handled by the cloud provider
  - Usually the unit of scale is the application
    i.e., increase/decrease the number of app instances or components (DB, web server, …)
- **To SaaS**
  - Typically the cloud software or service being provided automatically handles elasticity

[1] Capacity to scale up or down resources (e.g., computational, storage, network) and/or components (e.g., databases, applications, web servers) according to applications/services' requirements.

Summary

Disadvantages

IaaS, PaaS and SaaS

Loss of control

When deploying and maintaining user applications

- **From IaaS**
  - No control over the hardware and virtualization software being used
    - No possibility of fine-tuning the infrastructure for the user application
- **To PaaS**
  - No control over specific hardware and the PaaS platform
    - User application's management, fine-tuning, and monitoring are restricted to the tools provided by the platform
- **To SaaS**
  - Third-party cloud software, which is optimized and mainly configured by the cloud provider

Dependability and Security

For applications using IaaS, PaaS and SaaS

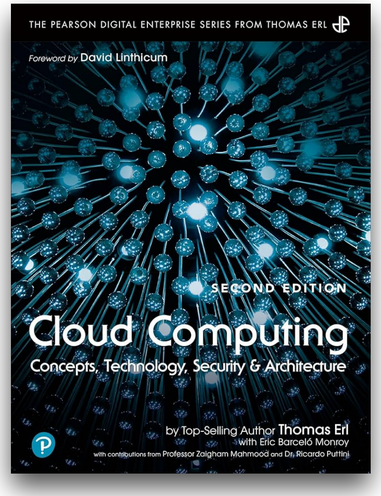
- **Dependability:** If (even if unlikely) the provider fails, the application fails and recovery is out of the control of the application owner
- **Security:** The application is at most as secure as the provider
  - any vulnerability of the provider is a vulnerability of the application
  - fixes to cloud services vulnerabilities must be done by the provider
- **Privacy:** The cloud provider holds data from users
  - Risk of data being leaked either by external or internal attacks
  - Critical data sometimes is protected by legislation (e.g., GDPR, HIPAA) that prevents some institutions (e.g., Hospitals) from using third-party services to store such information

Further Reading

- Want to test it out? Play with the **cloud simulator** done by Luís Ferreira! https://demo-iaas-paas-saas.netlify.app/

- 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***. Communications of the ACM 53, 4 (April 2010), 50-58.



Questions?