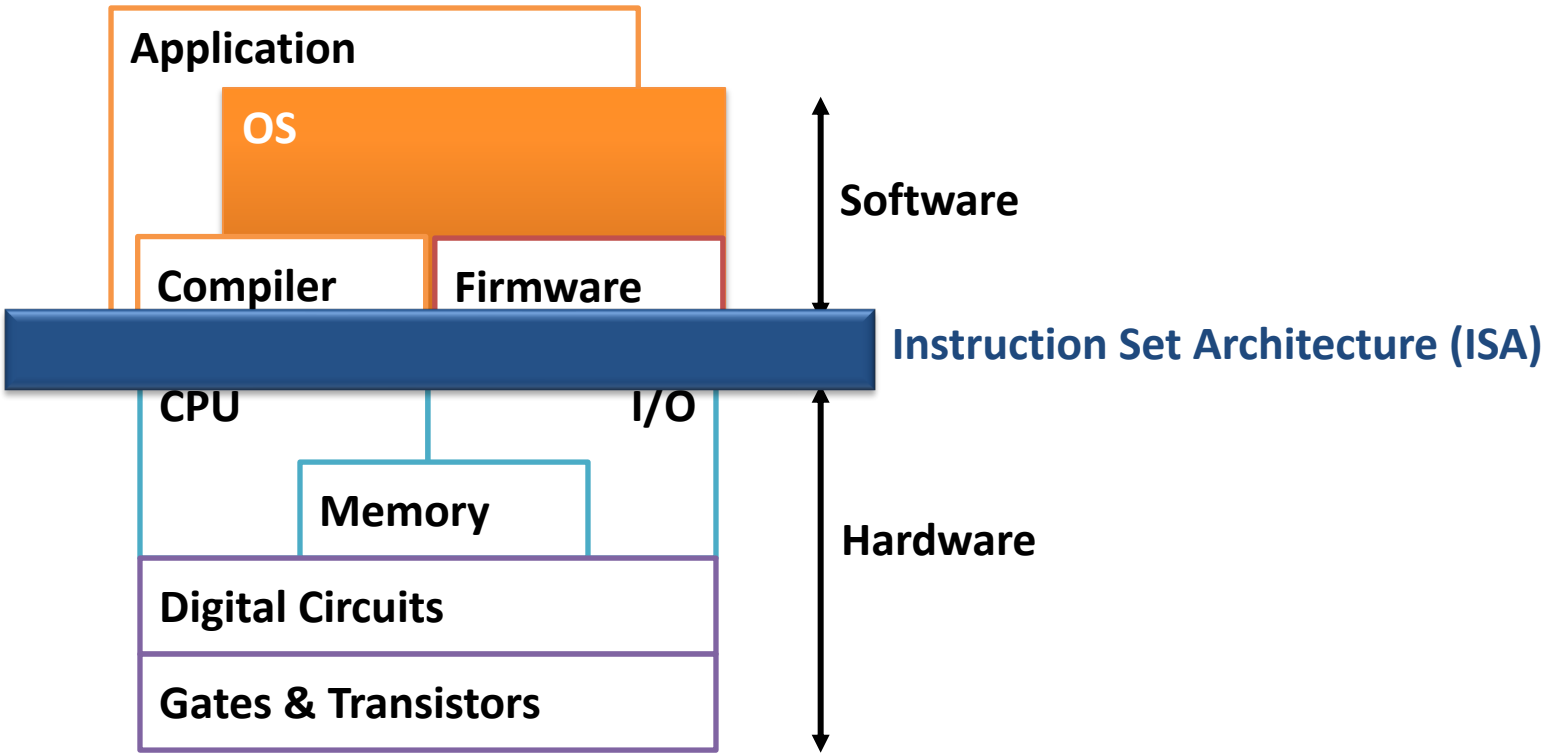
# Sistemas, Virtualización y Seguridad

https://github.com/valentinpuente/SVS

# Motivation of The course

- This figure is no longer “representative” for many computing system



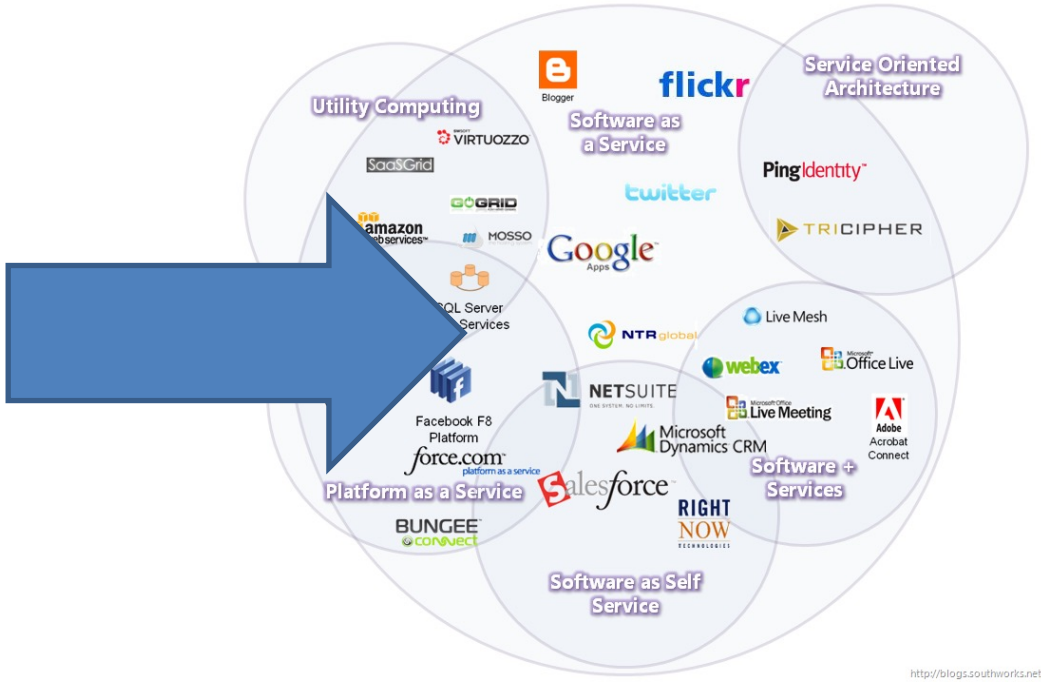
# Trends in current computing

- Mobility
- Cost optimization
- Productivity

- Rise the level of abstraction!
  - Conceptualize the computing resources like the electric "grid": just wall sockets no more gasoline generators!
  - Actually, in the origin (~2000), it was called "Grid Computing"

# The Cloud

# “Software” Computing Infrastructure



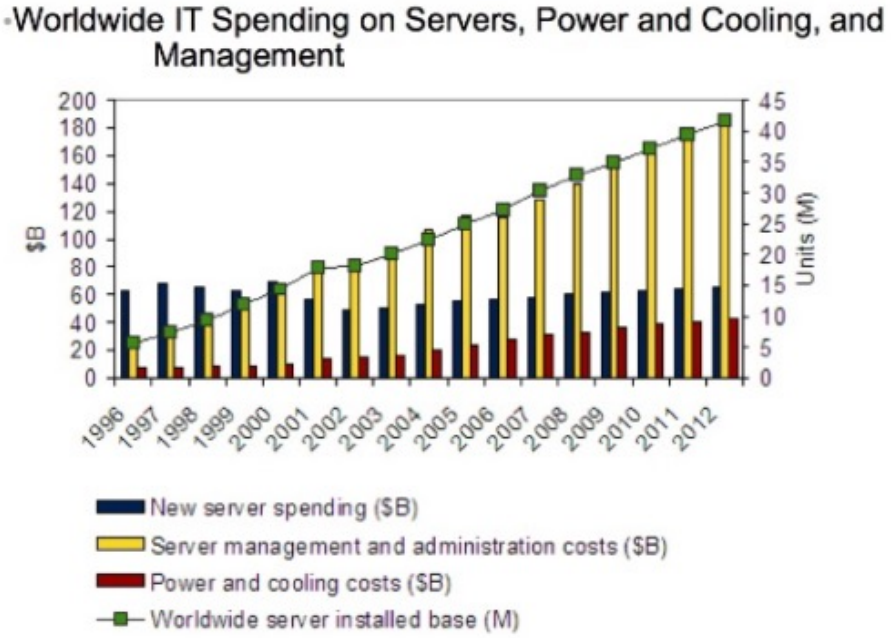
# Cloud Characteristics

- On demand self-service
- Ubiquitous network access: Anywhere, Any device, Any Time
- Location Independent resource pooling
- Rapid Elasticity
- Pay-as-you go

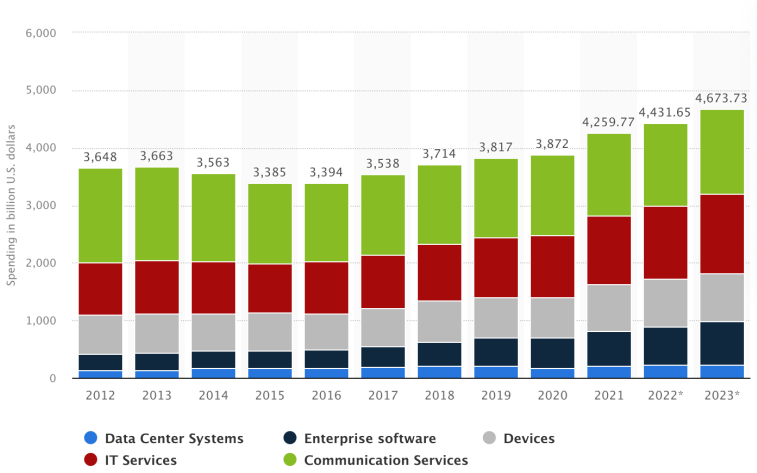
# Supporting Factors of Cloud Computing

- Processor advancements
- Networking Technology
- Virtualization Technology
- Automated management
- Fast and inexpensive hardware

# The impact of cloud computing




- Hardware cost reduction is noticeable
  - Moore's law+Cloud computing
  - Little to no overprovision
- Services are dominating
  - New market models
  - Optimization of HW resources



# The Cloud Computing Stack

**SaaS**

- Software as a Service
- *Book a room in a hotel*

**PaaS**

- Platform as a Service
- *Rent a furnished apartment*

**IaaS**

- Infrastructure as a Service
- *Rent an unfurnished apartment*

# SaaS

- Characteristics
  - Highest level of abstraction
  - No hardware of software to manage
  - Services delivered through browser or custom clients
- Advantages
  - Pay per use
  - Scalability, reliability, security
  - Minimum management costs
- Examples
  - Salesforce (CRM)
  - GotoMeeting (collaboration)
  - Dropbox (Storage)
  - Google Docs (Office Docs)

# PaaS

- Characteristics
  - Medium level of abstraction
  - Service provider supply OS and software-stack to deploy customer tools
  - Services delivered through custom environment

- Advantages
  - Pay per use
  - Scalability, reliability, security
  - APIs

- Examples
  - Google App Engine
  - Windows Azure
  - AWS RDS

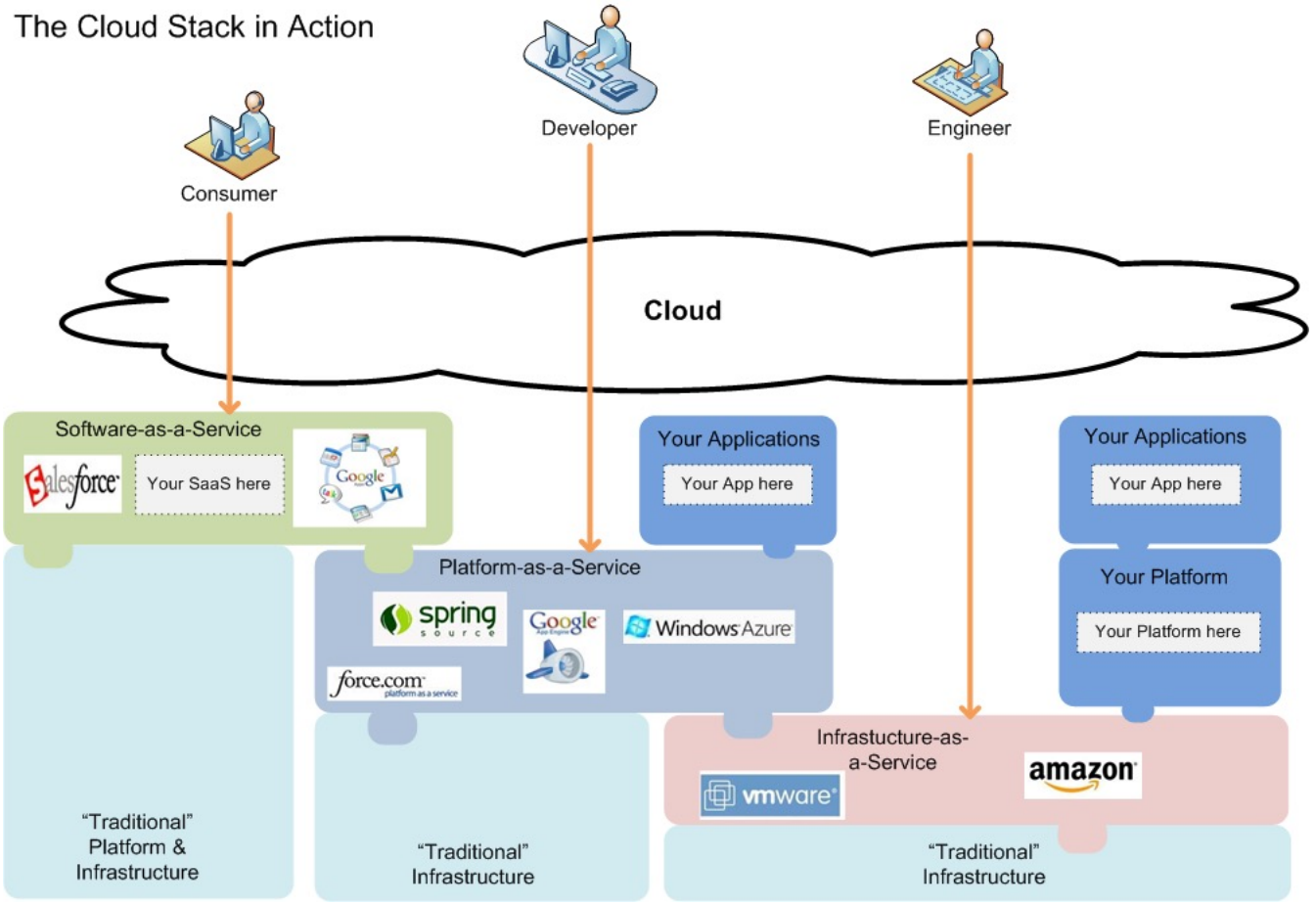
# FaaS (Function as a Service)

- Characteristics
  - Like PaaS but without maintaining any infrastructure (some people says is a particular case of Pass)
  - Serverless architecture
  - Typically used when building microservices applications.
  - Billed by transaction
- Advantages
  - Lower granularity in the cost
  - Zero Maintenance cost
  - API
- Examples
  - AWS Lambda
  - GCE Function
  - Apache OpenWisk

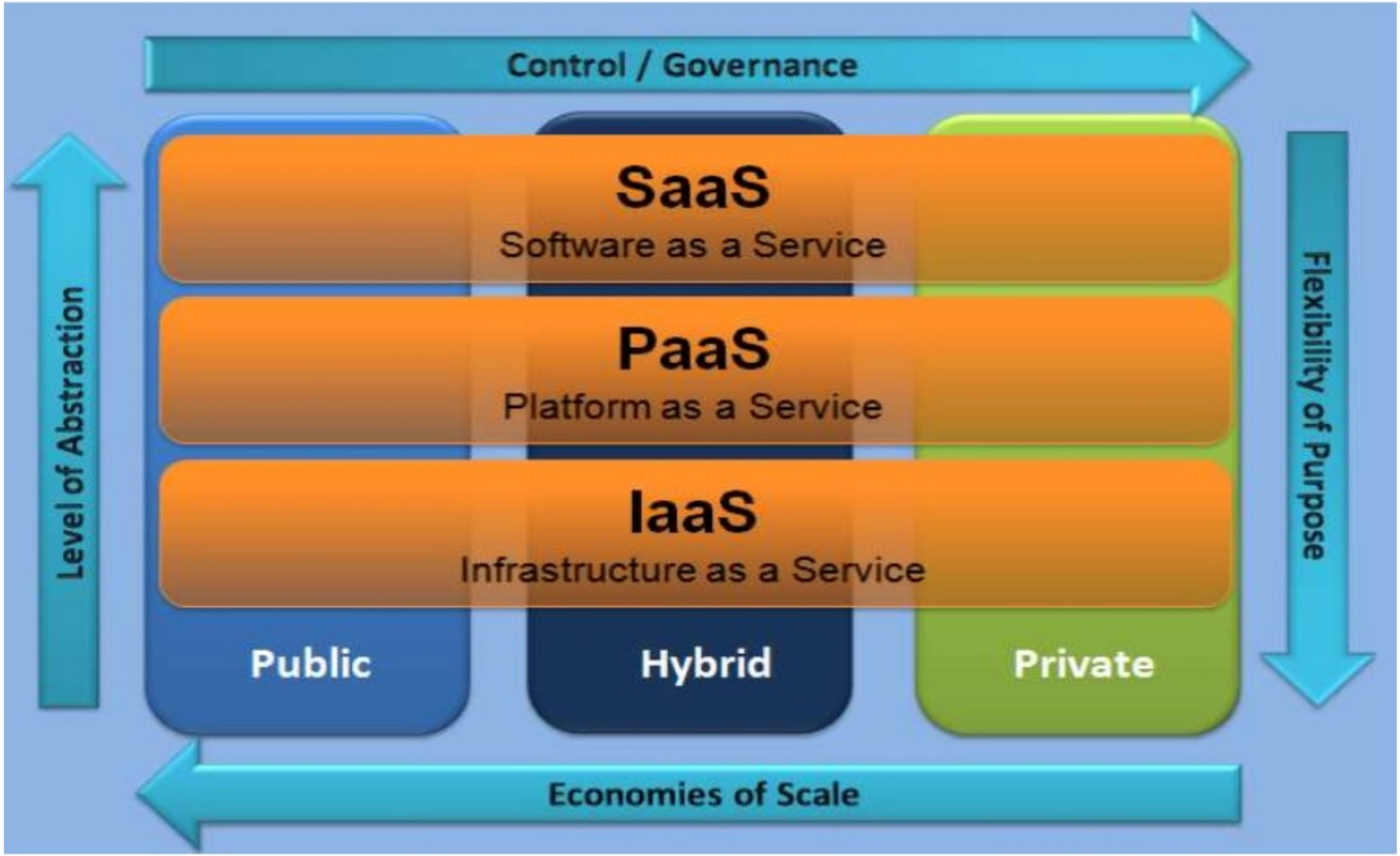
# IaaS

- Characteristics
  - Lowest level of abstraction
  - Service provider supply computing resources, i.e. CPU, Memory, Network and Storage
  - Services delivered through customized virtual machines, software defined network, etc…
- Advantages
  - Pay per use
  - Scalability, reliability, security
  - Flexibility
- Examples
  - AWS EC2
  - AWS EBS
  - AWS VPC

# Perspectives in Cloud Computing



# Cloud Computing Service Model

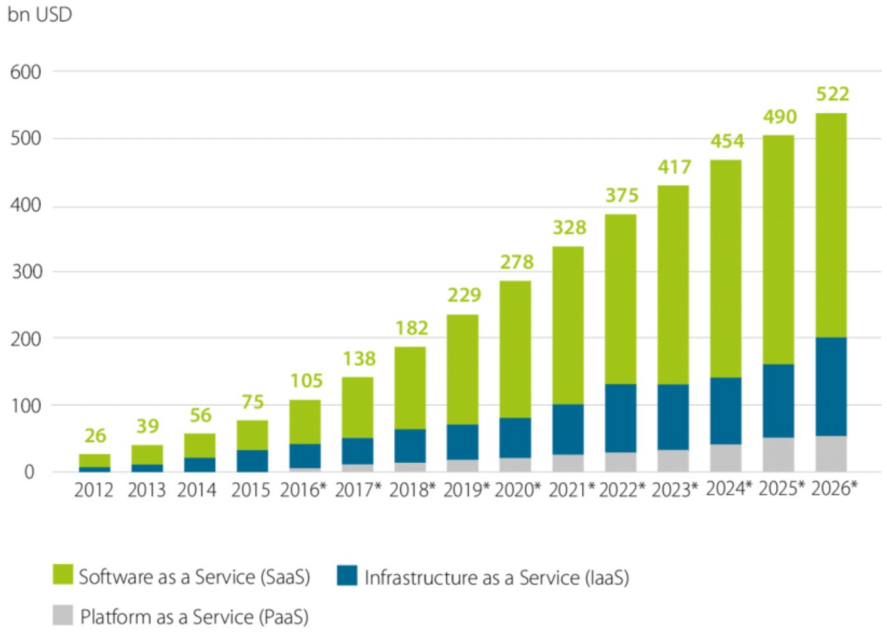


# Impact of Cloud Computing

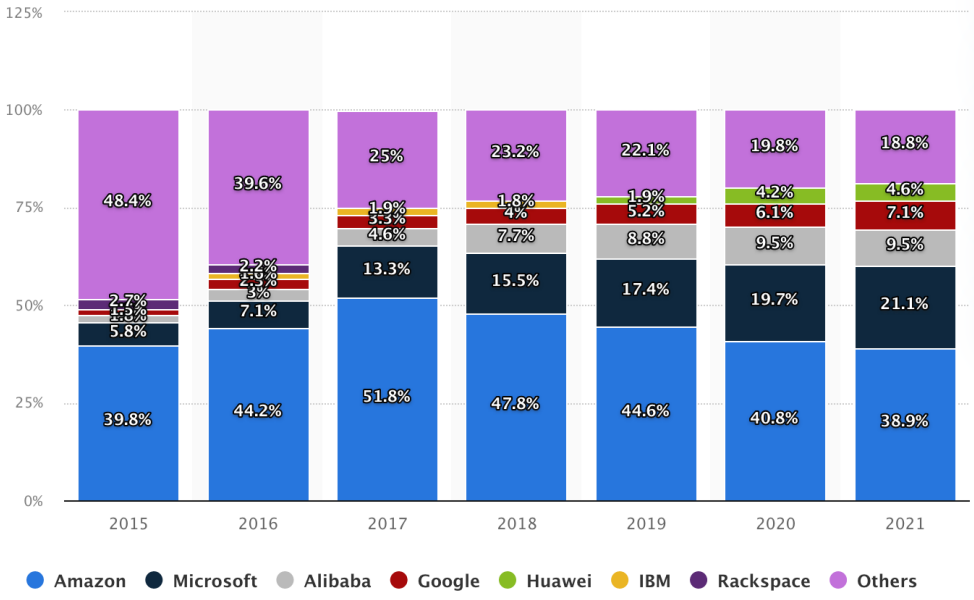
- More software companies and less hardware providers
  - Really easy to provide a service (if you have a ground-breaking idea)
- Start-up costs for service providing are almost zero
  - No CapEX to begin with
- Privacy is a big issue
  - Current support in hardware, start to address it (e.g., Secure Cloud in GCE)
  - Hardware issues

# Cloud Market

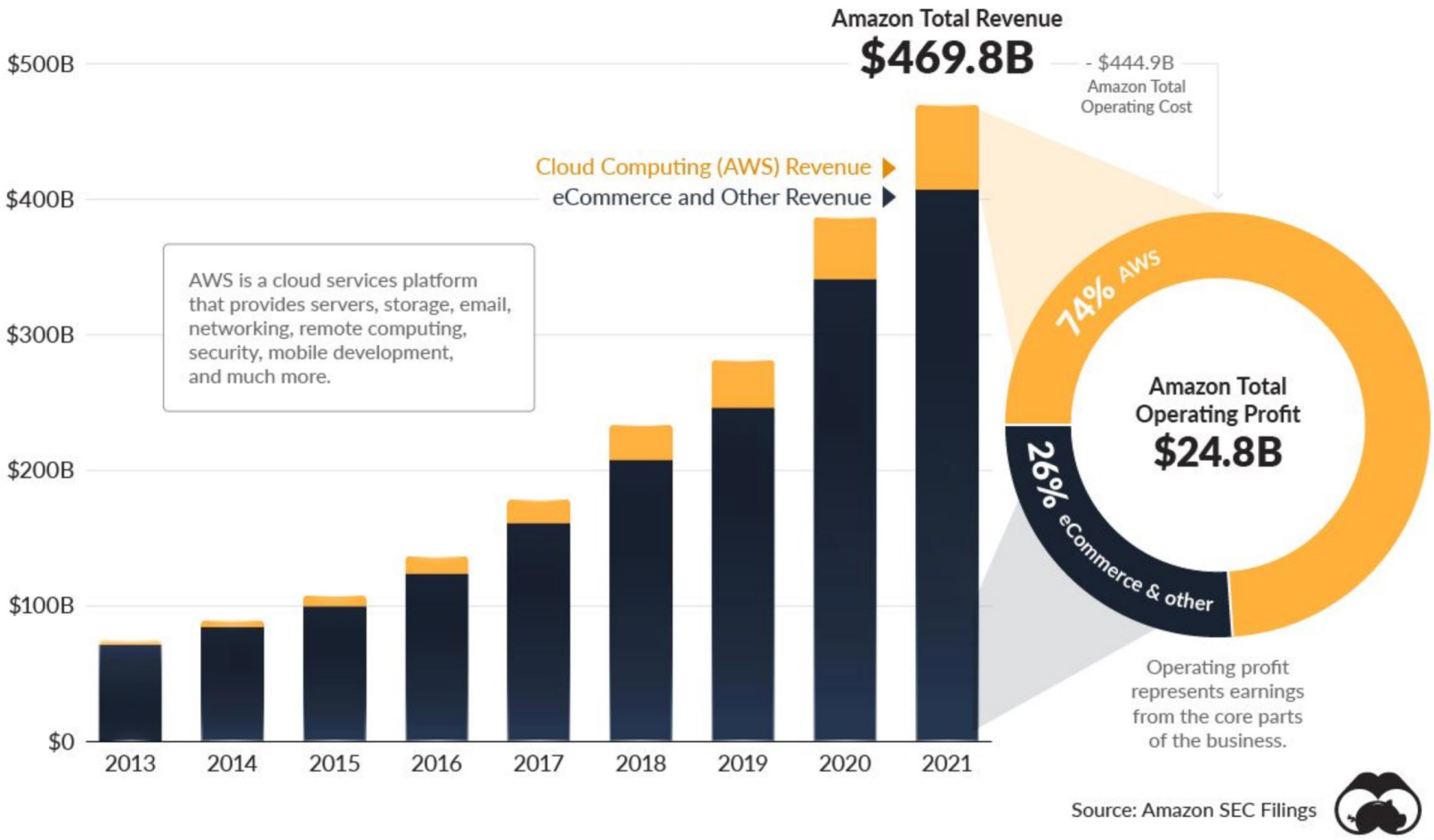
## Revenue




## Market Share



# Amazon vs AWS



# It is cloud Important for “you”?

- Two points of view: provider and customer
  - Companies should be "aware" of the cost optimizations that the "cloud" offers.
    - Focus on administrators
  - There is an open market for cloud providers
  - Specialize to compete with the big players (Amazon, Google, Microsoft)

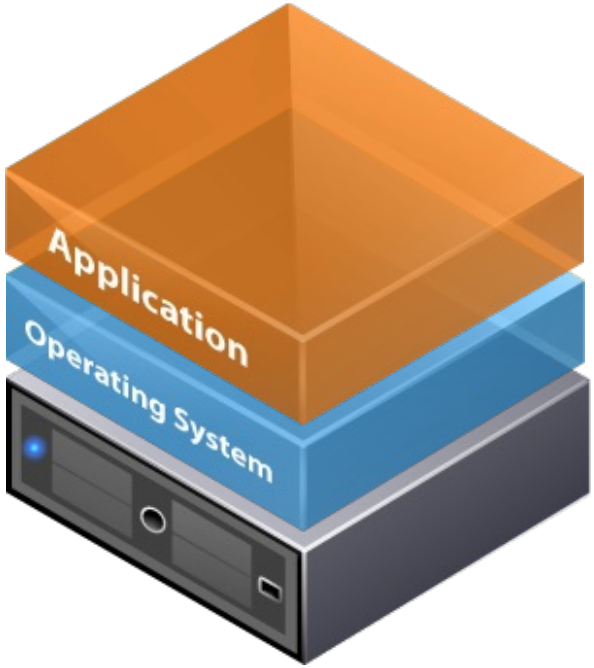
- Specialization requires a vertical view of the system

# What it's next?

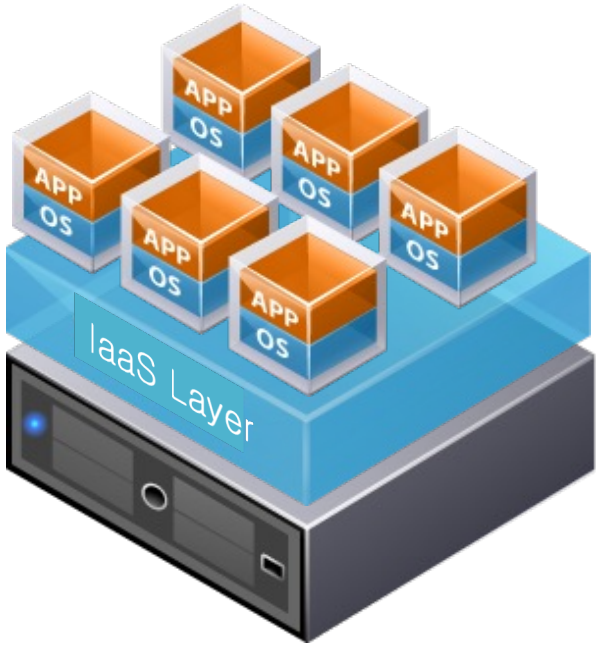
- Big Data & Data Economy
  - Turning knowledge (conscious or not) into economic benefits
    - Somehow related to cloud computing
  - Massive services (big data producers) running on it
  - Will turn back hardware as "key" (?)
  - Deep Learning

# The Focus of this Course

- IaaS
  - Closest layer to the "classical" computing system layering



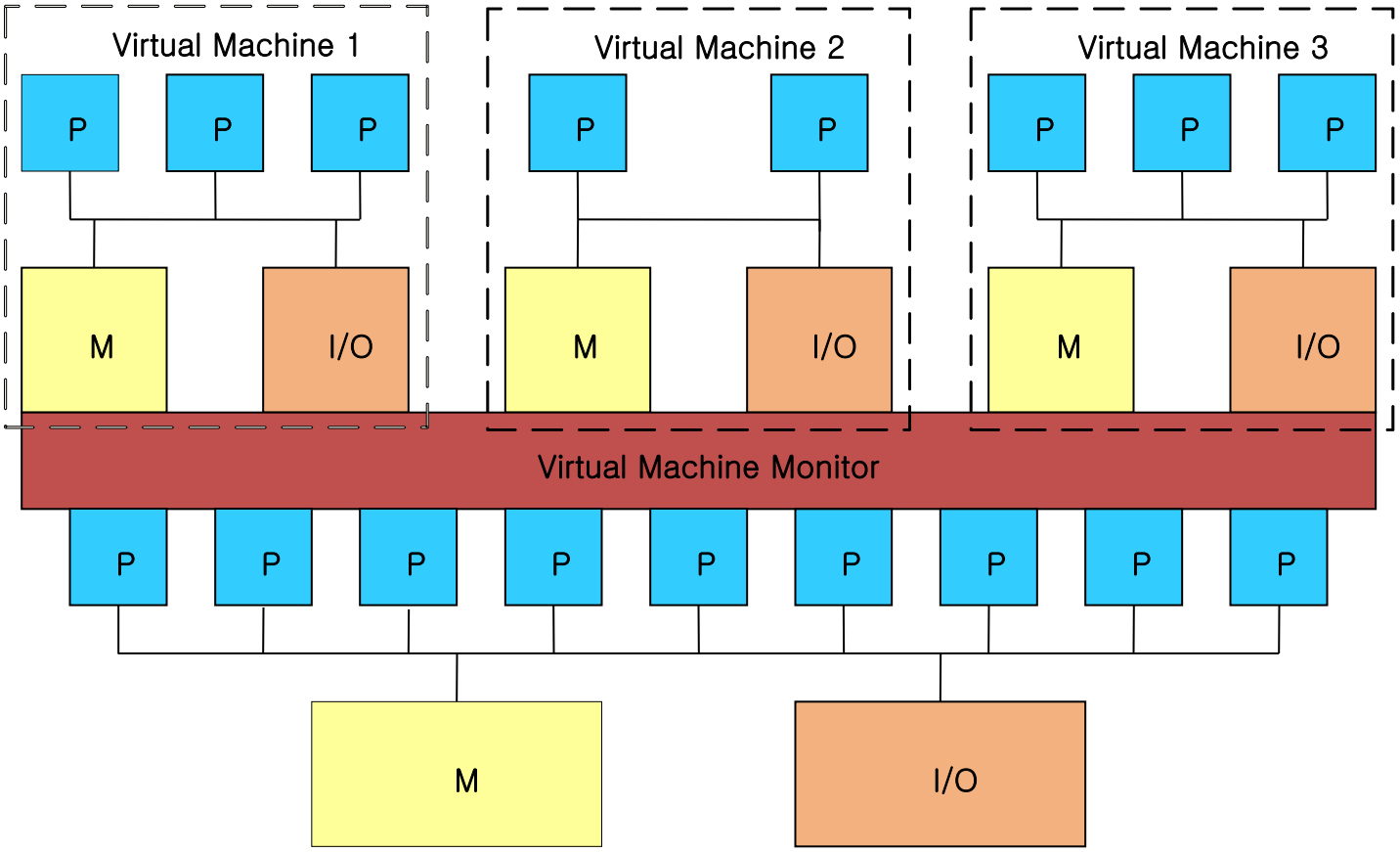

Traditional Architecture




Virtual Architecture

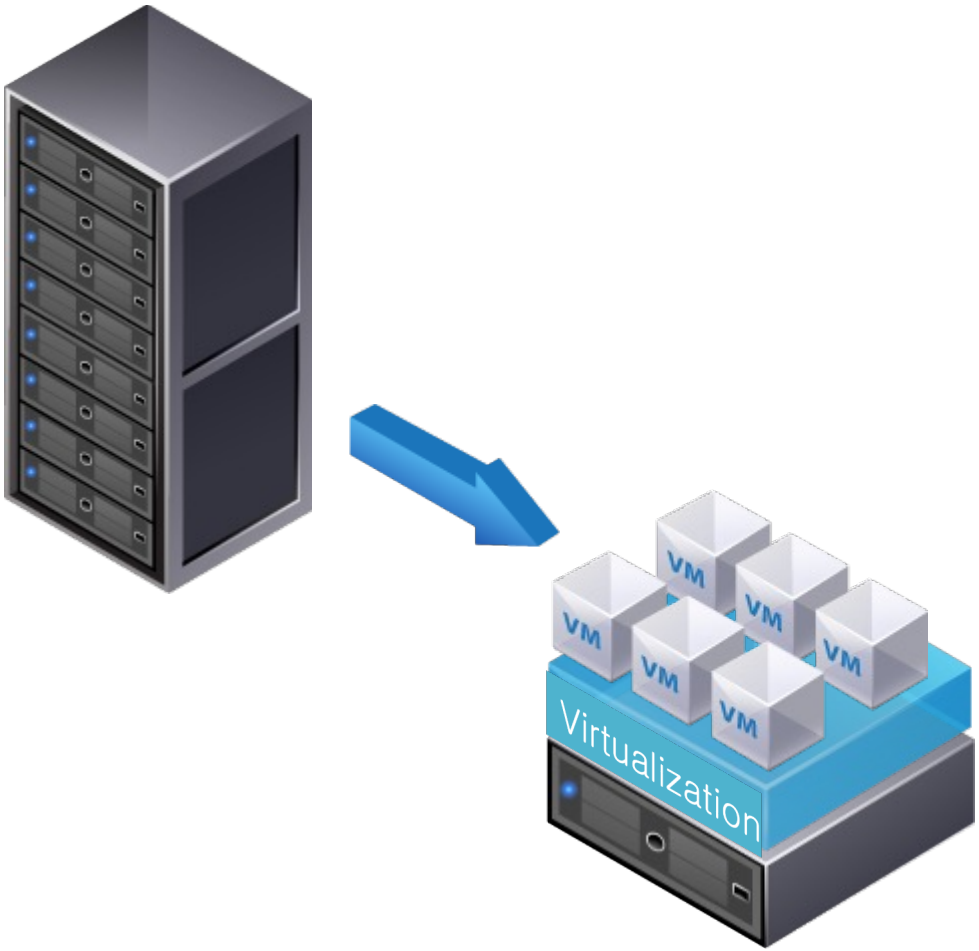
# The actual view of computing systems

- Understand the "support" for "Software Defined" Computing Systems



# Virtualization is the key component

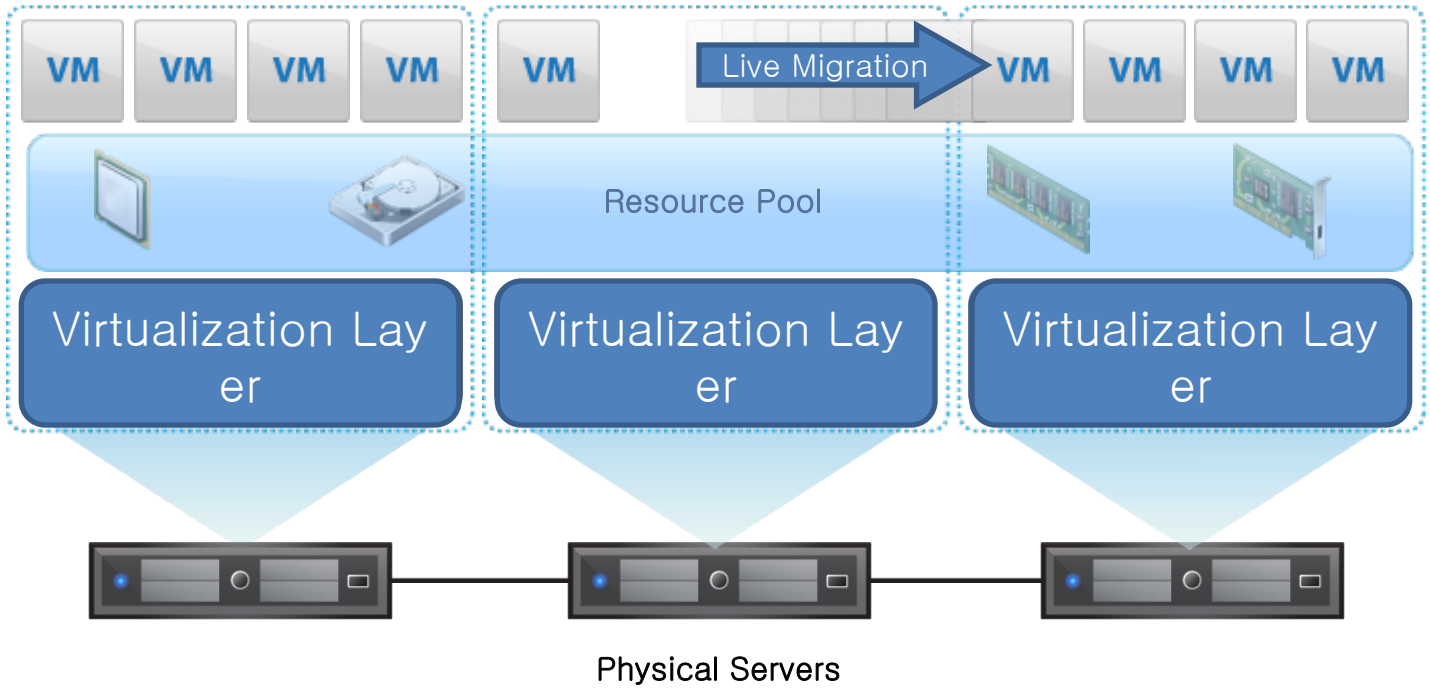
- Consolidation, fault tolerance, easy deployment, ….



# Virtualization Layer

- Is the key component to make “believe” the hosted OS/App that has “real” hardware resources

- Virtualization introduces flexibility to allocate hardware resources under demand: scalability and availability



# Issues we want to address?

- Understanding Key Operating System Concepts
  - Resource Virtualization
  - I/O Handling
- Understand Virtualization
  - What is the performance overhead of virtualization?
  - How does modern hardware mitigate the impact?
- Security (from an architectural perspective)
  - Support in stat-of-the-art hardware for security
  - Modern hardware issues

# Objectives & Approach

- OS becomes another app more

- Provide the basic foundations to understand the state-of-art computing infrastructure works

# Outline

- Review of Operating Systems
- Virtualization
- Virtualization without architectural support
- Virtualization in x86
- Architectural support for security
- Security in modern processors

# Lab

- Focus on papers to review
  - Set of selected papers for review
  - Available in repo
  - Owner assigned by pull request
  - Invite vpuente@unican.es to collaborate on forked repo
  - All personal work should be available in forked repo

- Evaluation
  - Depends on the quality of the available information in the repo and presentation

# Material

- OS review
  - Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau, "Operating Systems: Three Easy Pieces", Arpaci-Dusseau Books, March, 2020
- Virtualization
  - E. Bugnion, J. Nieh, and D. Tsafrir, "Hardware and Software Support for Virtualization," *Synth. Lect. Comput. Archit.*, vol. 12, no. 1, pp. 1–206, Feb. 2017.
- Security
  - J. Szefer, "Principles of secure processor architecture design," *Synth. Lect. Comput. Archit.*, vol. 13, no. 3, pp. 1–173, 2019.

# Disclaimer

- 3-to-1