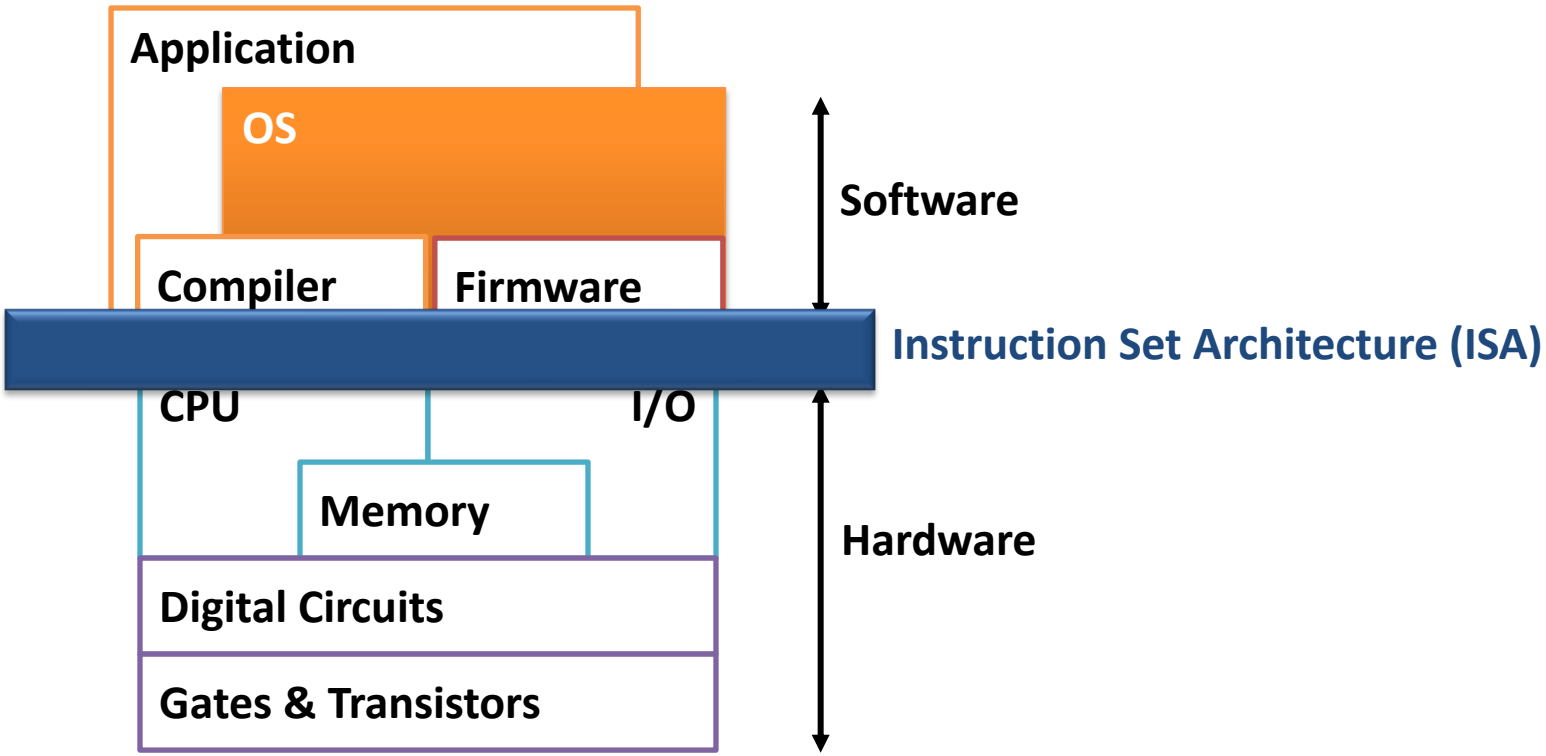
Sistemas, Virtualización y Seguridad

https://www.ce.unican.es/course/svs/
https://discord.gg/EFuM2eYgAp

Motivation of The course

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



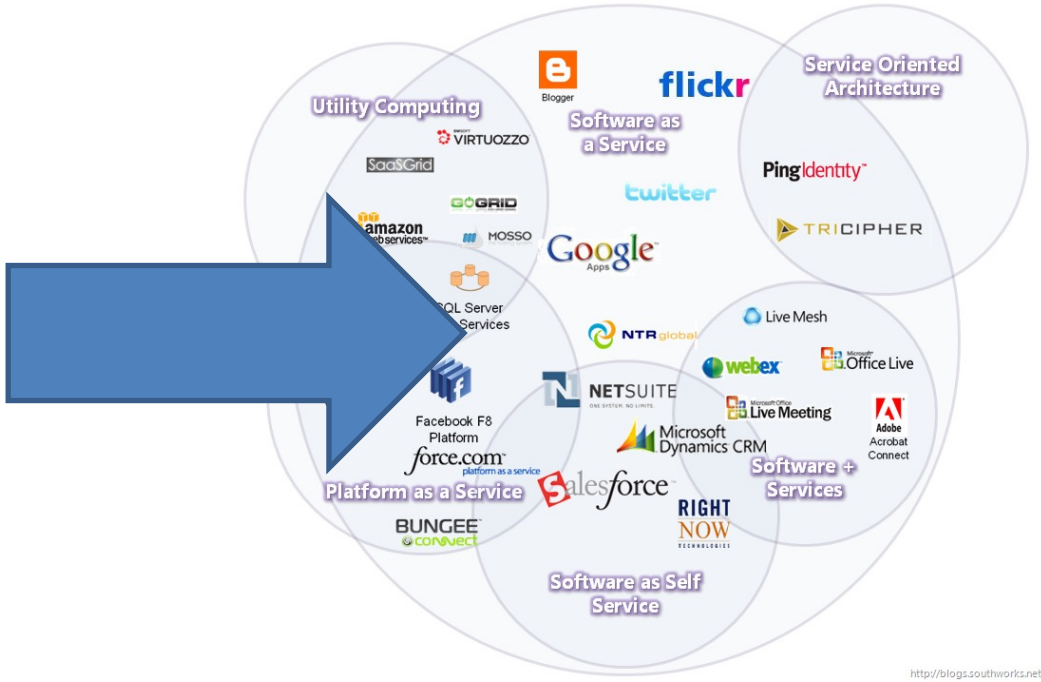
Trends in current computing

- 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



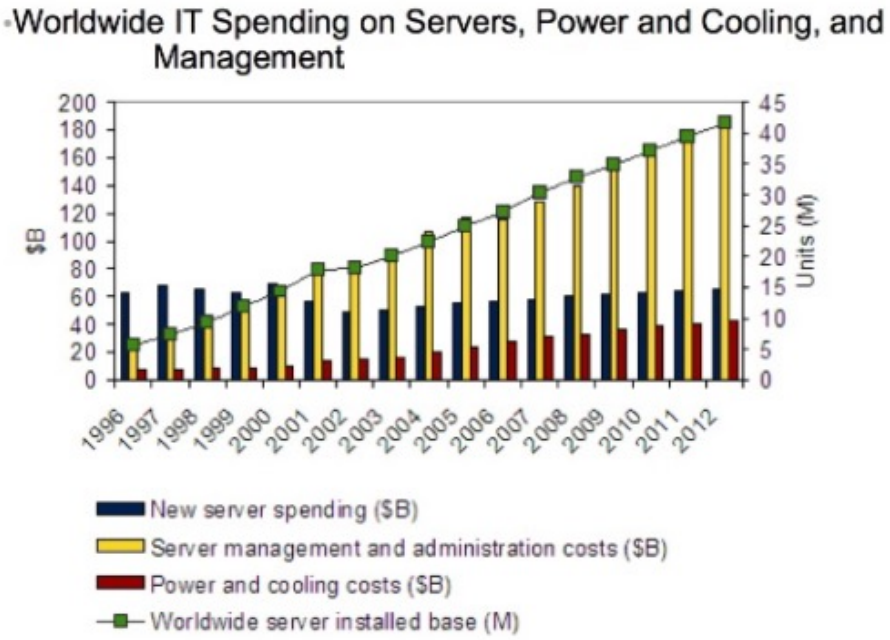
Cloud Characteristics

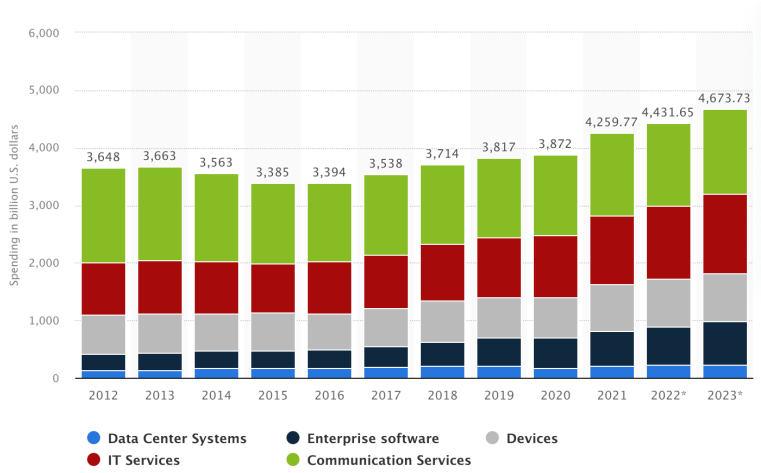
- On demand self-service
- Ubiquitous network access: Anywhere, Any device, Any Time
- Location Independent resource pooling
- Rapid Elasticity
- Pay-as-you go

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

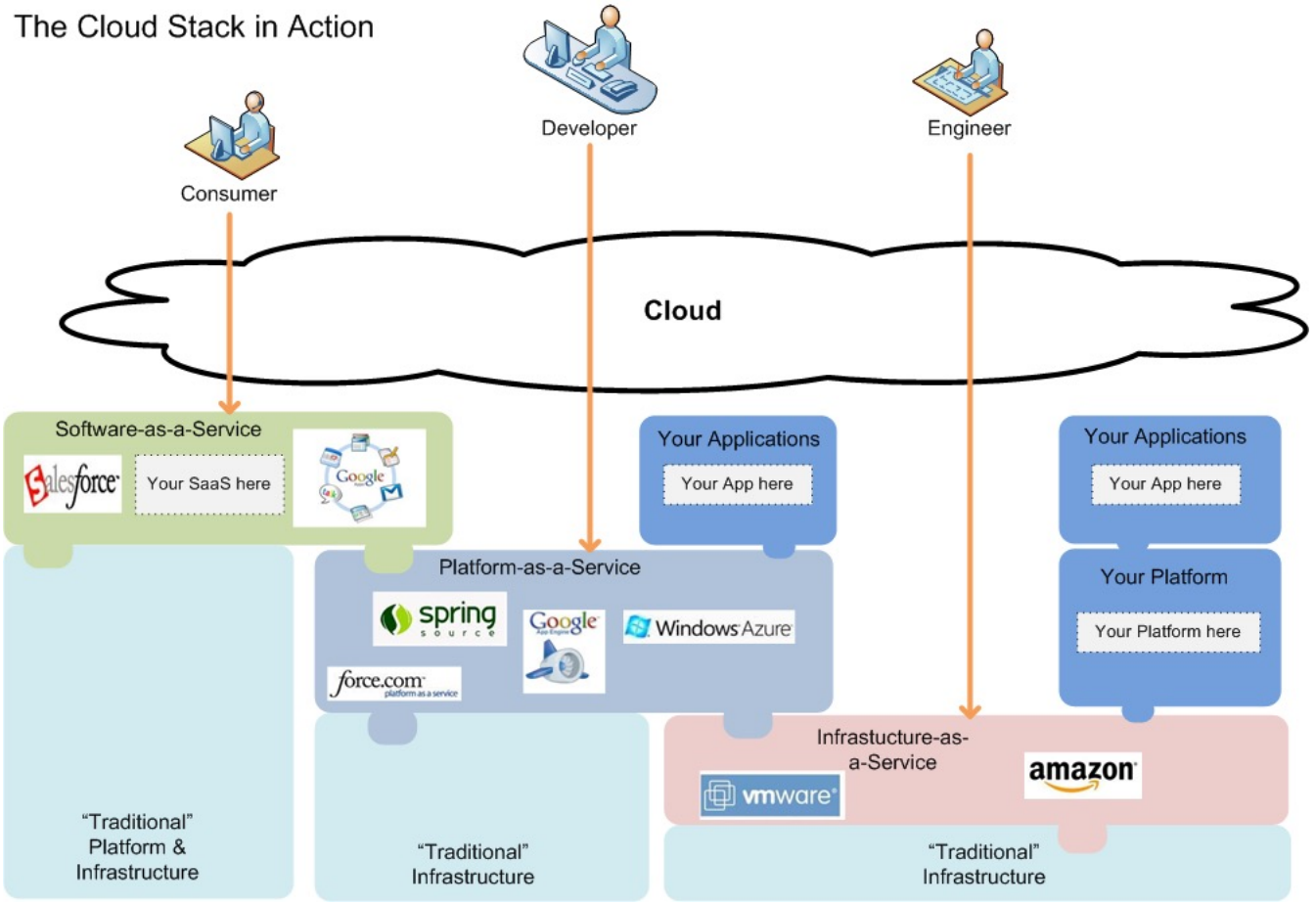
FaaS (Function as a Service)

- 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

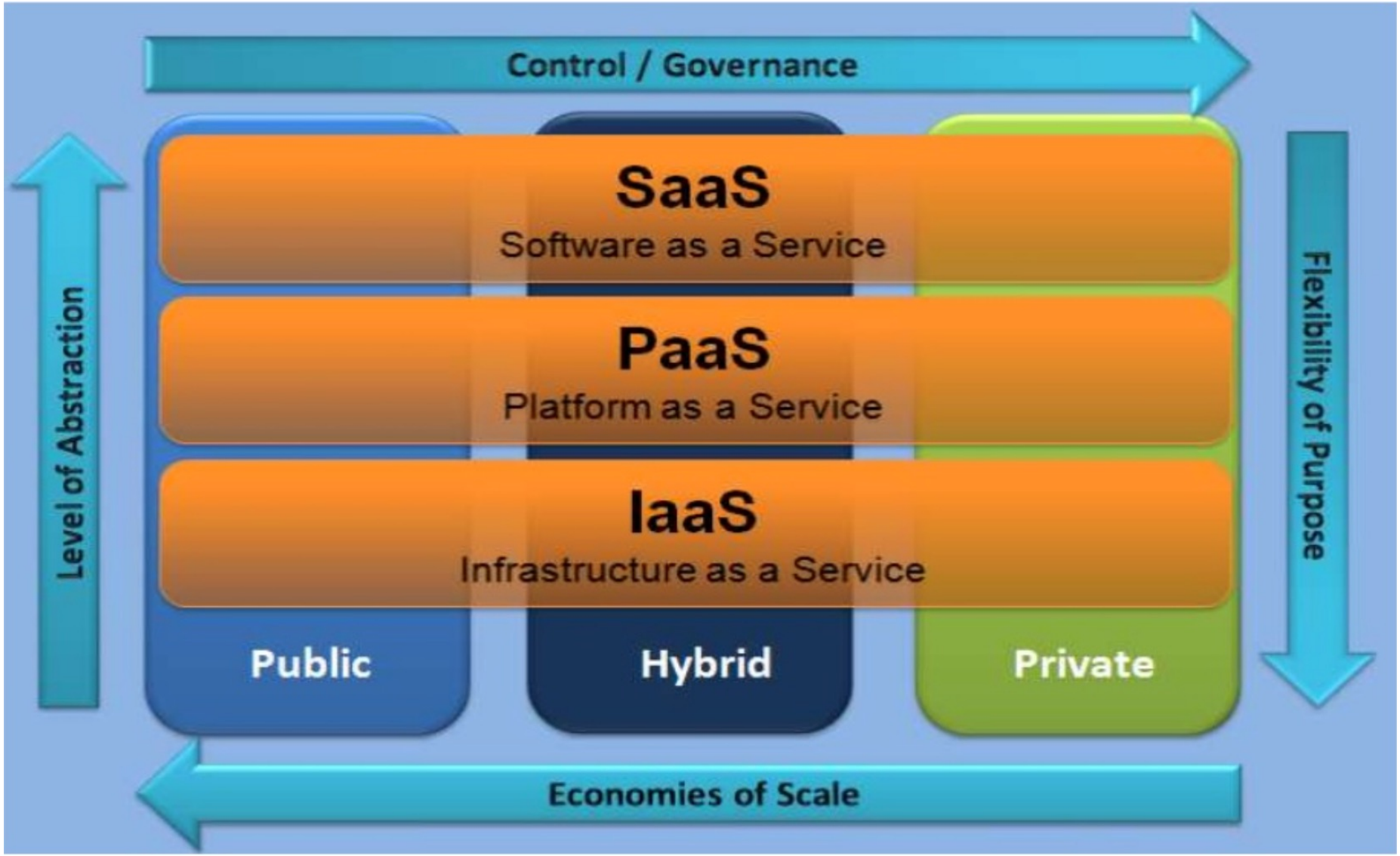
IaaS

- 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

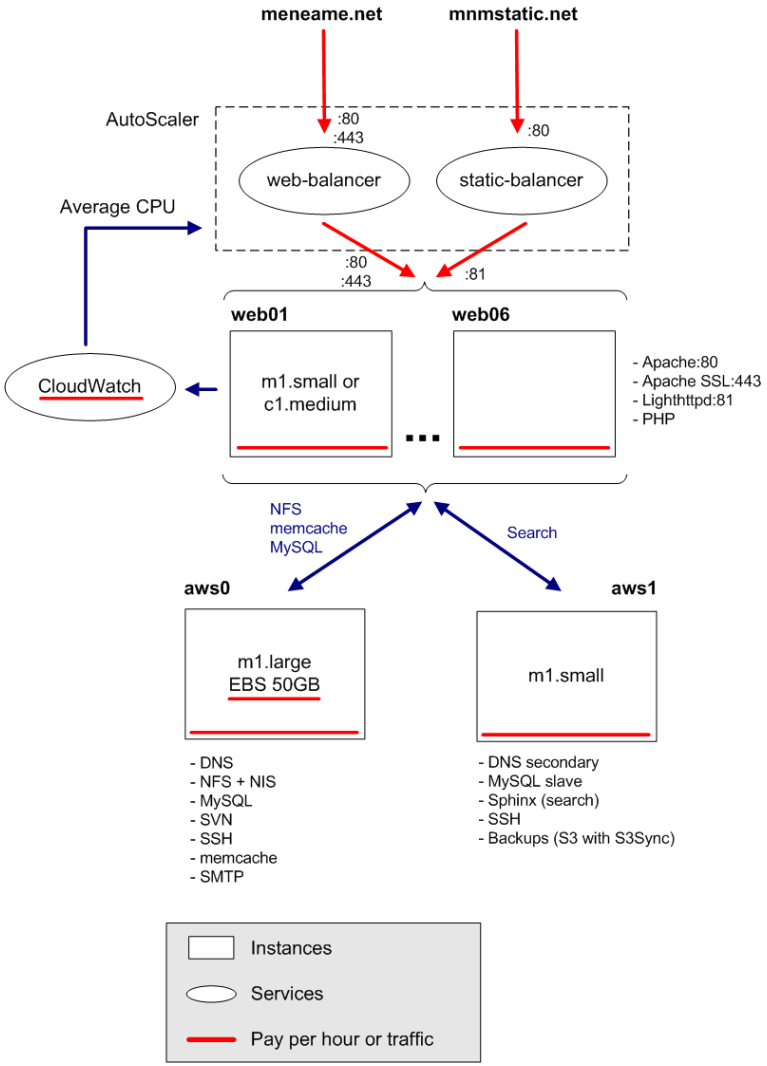


Cloud Computing Service Model



A practical example of "cost optimizations": Meneame over AWS/EC2 (IaaS practical use)

- Two static backbends
  - MySql+memcache
  - 2009: m1.large (0.38$/h), m1.small backup (0.095$/h) & 50GB EBS
- Front-ends (a.k.a. web servers)
  - Running ngix
  - m1.small or c1.medium
  - Scaled up and down according the traffic
- Where area the savings?
  - At 100% system load cloud is not cost effective
    - Amazon is not an ONG!
  - The reality is that the load is lower (~10-20%), I can share the common offsets
    - TCO is shared with other users

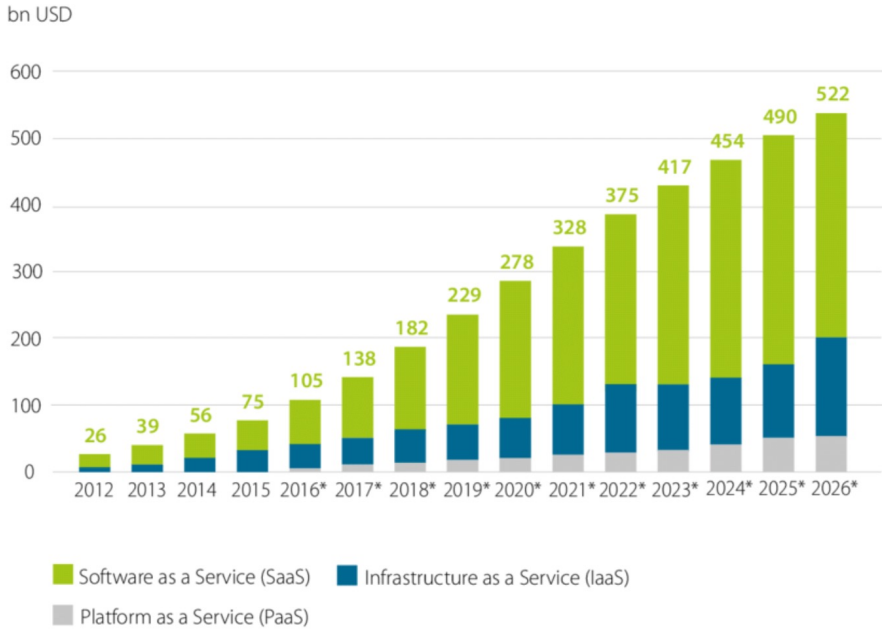


Impact of Cloud Computing

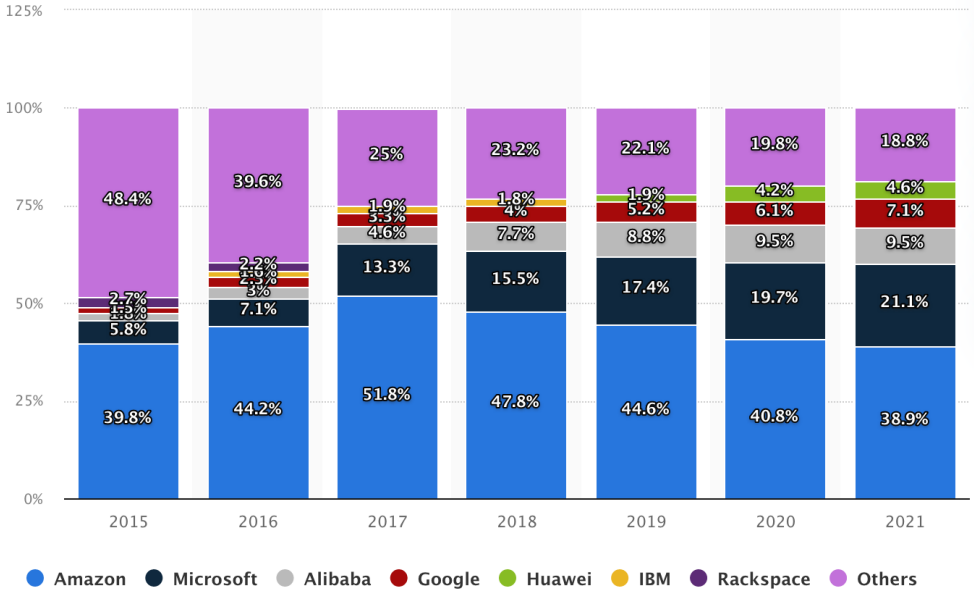
- More software companies and less hardware providers
  - Really easy to provide a service (if you have a ground-breaking idea)
- Start-up costs for service providing are almost zero
  - No CapEX to begin with
- Privacy is a big issue
  - Current support in hardware, start to address it (e.g., Secure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

Cloud Market

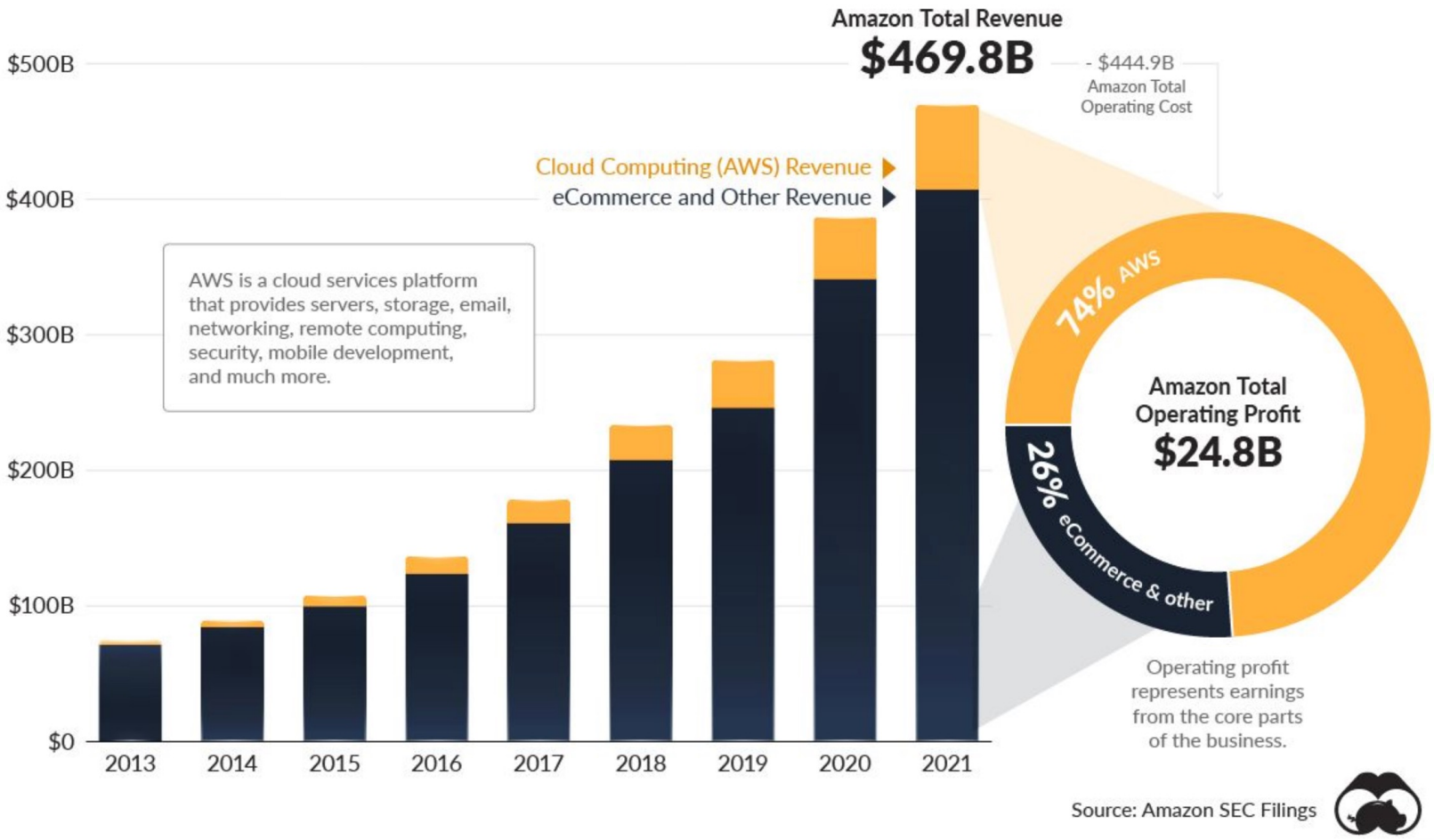
Revenue



Market Share



Amazon vs AWS



It is cloud Important for “you”?

- Two standpoints: provider and customer
  - The companies should be “aware” of the cost optimizations that the “cloud” provides
    - Focus on **admins**
  - There is an open market for cloud providers
    - Through specialization, you can compete with big players (Amazon, Google, Microsoft)
- Specialization requires a vertical vision of the system

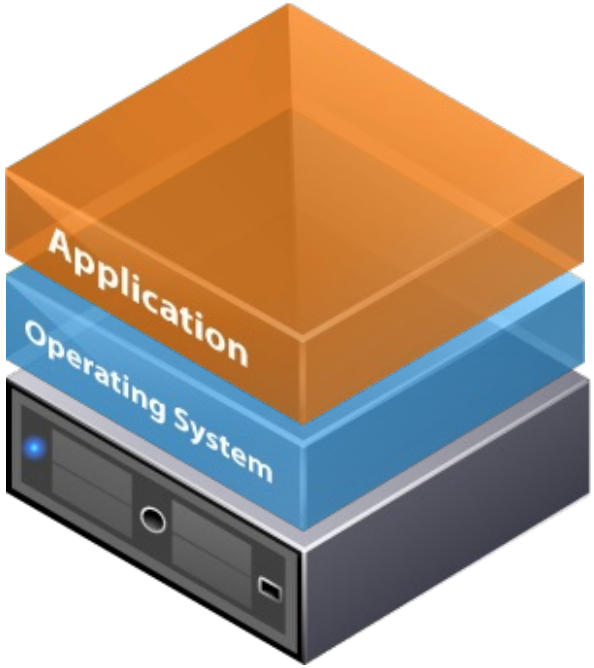
What it's next?

- Big Data & Economy of data
  - Turn knowledge (aware or not) in economical benefits
  - Somehow connected to cloud computing
    - Massive services (big data producers) runs on top of it
  - Will turn back hardware as "key" (?)
  - Deep Learning

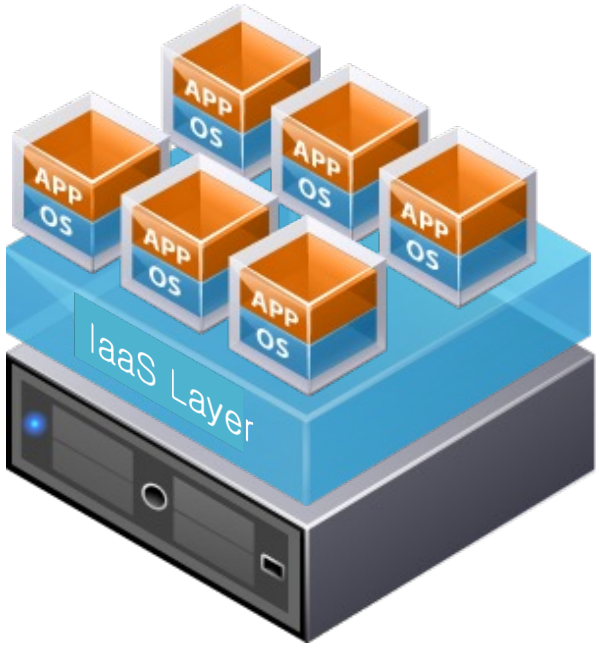
- *Security and/or Privacy*
  - *Hardware support*
  - *Hardware Issues*

The Focus of this Course

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



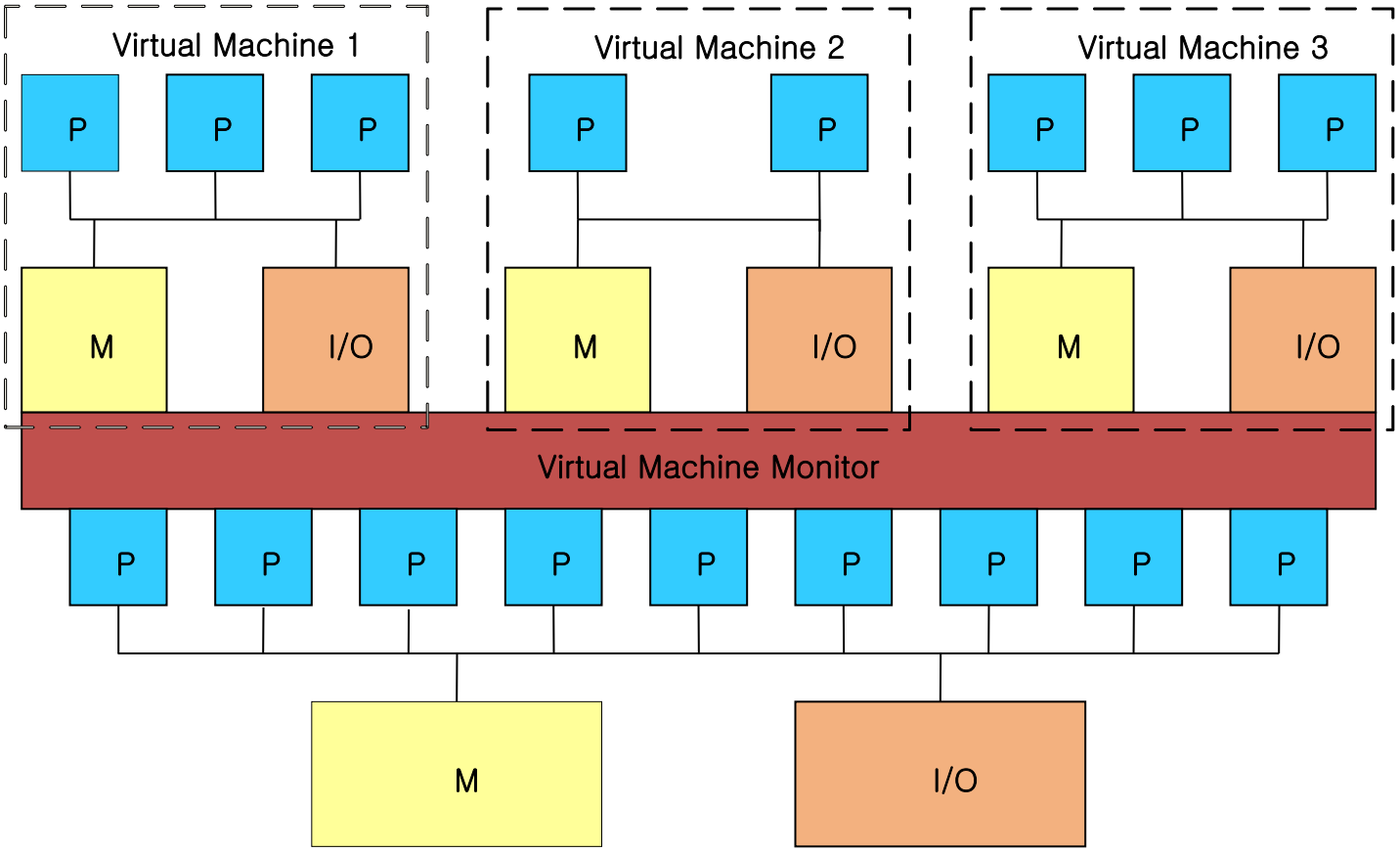
Traditional Architecture



Virtual Architecture

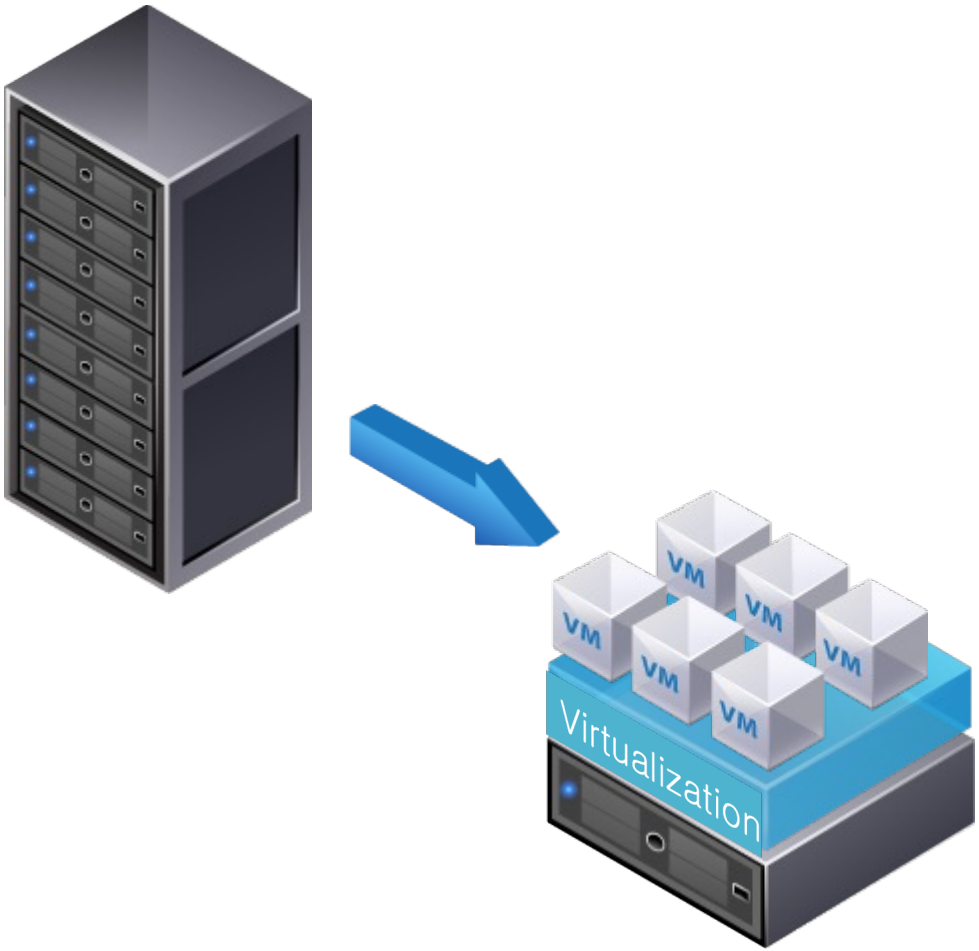
The actual view of computing systems

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



Virtualization is the key component

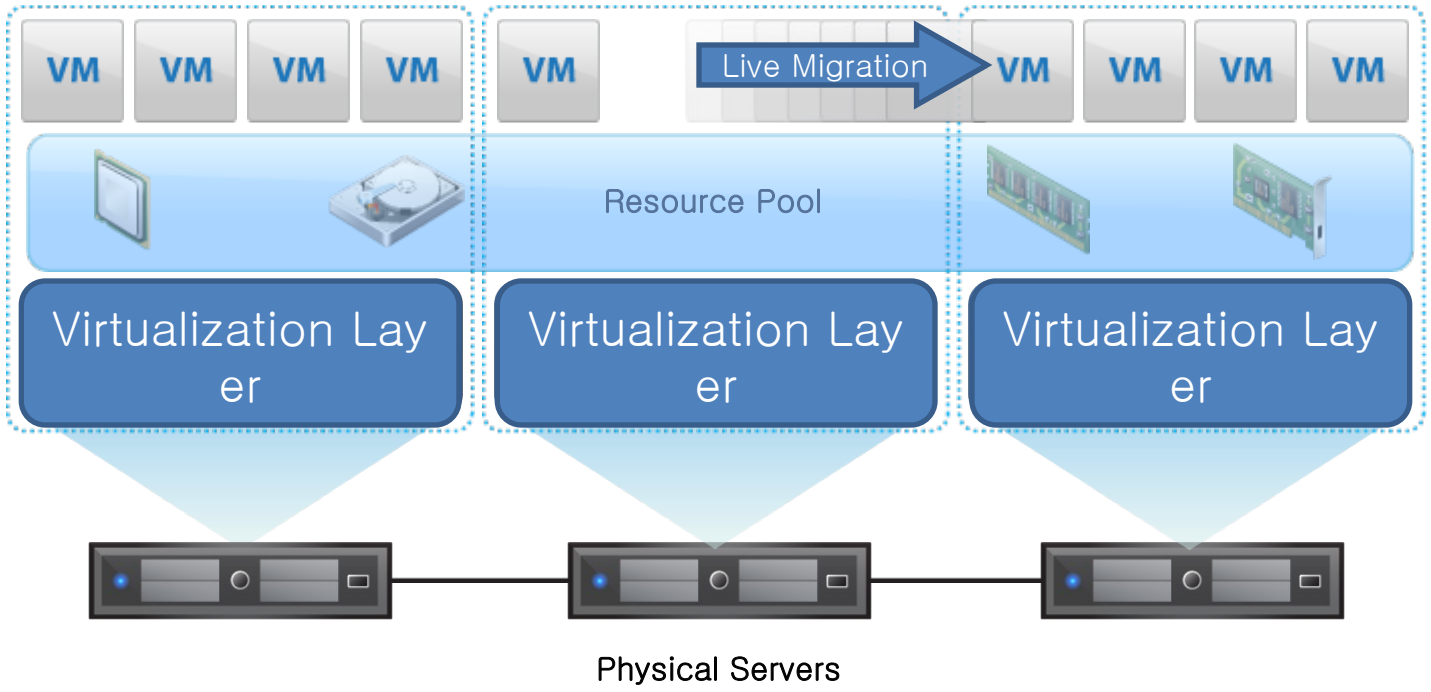
- Consolidation, fault tolerance, easy deployment, ….



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?

- Familiarize yourself with cloud computing
  - Performance & Costs
  - Ease of use
- Virtualization
  - What is the performance overhead of the virtualization?
  - How modern hardware eases the impact?
- Security and/or Security
  - Modern hardware issues
  - Support in state-of-the-art hardware for security
  - *Public clouds security*

Objectives & Approach

- OS becomes another app more
- Provide the basic foundations to understand where contemporary computing systems are going
- Top-down approach
  - From Cloud to Architecture

Outline

- Performance and Complex workload deployment
- Public Clouds: GCE + Azure
- Containers
- Virtualization
- Virtualization without architectural support
- Virtualization in x86
- Security (architecture perspective)

Lab

- Two parts
  - Labs (warmup) (1/3) of course
  - Project (2/3) of course

- Projects Options
  - Individual:
    - Survey paper, Case study, etc…
  - Group:
    - Original research proposal

Labs (warmup)

- Designed to perform some simple task in IaaS
- Although specific, quite open ended given the cloud richness
- Can be done in group (ideally two persons)
- Will be evaluated according the provided report and presentation quality

Option I: Survey, Case study, etc…

- In the second option, you may **work alone** to write a paper that surveys a related area.
- The paper should: summarize work in an area, giving extensive references, present opinions of others for and against various options (with references), and conclude with your opinion of the strengths and weaknesses of arguments presented above.
- You will be graded on the completeness of your survey, the accuracy of your summaries, the support you give for your opinions, and the quality of your presentation.
- Since a survey paper is "safer" than a research project, I will hold survey papers to a higher standard of completeness and analysis of the literature.
- Examples: compare different virtualization tools, cloud providers, containers, etc…

Option 2: Project

- The first option is to do some original work/research related to cloud computing (e.g., analyse cost saving of using cloud, implement some functionality, etc…).
- You will be graded on how well you define your problem, survey previous work, design and conduct experiments, and present your results.
- Since time is limited, however, the above goal is hard to reach, and I will reward those that aim high even if they do not completely succeed. The key is insuring that some aspects of your work are completely done; it is very hard to grade a project where the simulator did not quite work.
- This type of project should use groups of **three (preferred) or two.**
- **Any topic related with the course will be ok. Align your interests with the project and/other subjects**

Some ideas for Projects (bottom to top)

- Explore the virtualization capabilities in risc-v
- Use or extend teaching OS such as https://github.com/mit-pdos/xv6-public or hypervisors such as https://github.com/zeling/lvisor
  - Support for multicore, Nested virtualization in lvisor
  - Support for containers in xv6
- Use unikernels and/or lightweight VM to deploy some application (js, networking, etc..) running on top of Xen, KVM, etc…
- Accelerate hardware simulators using virtual machines
- Explore and use Infrastructure as a Code tools
- Large scale cloud deployment (e.g., LLM)
- Deploy **your** App in the cloud
- …

More (nice and challenging) ideas

- Practical use of FHE
  - https://spiralwiki.com/
- Use secure enclaves to prevent "~~Call of Duty~~" online game cheaters
  - http://web.cse.ohio-state.edu/~lin.3021/file/SYSTEX16.pdf
- Run Linux x86 binaries on Apple Silicon
  - https://developer.apple.com/documentation/virtualization/running_intel_binaries_in_linux_vms_with_rosetta
- Virtual machines, emulation Android and iOS

Some ideas for Surveys

- Hypervisor comparisons: Xen, KVM, Hyper-V, VMWare, …
- Hardware support for virtualization comparison: Power, riscv, x86 (AMD), …
- Hardware support for secure clouds (Intel SGX, AMD SEV, Intel TME, etc..)
- Cloud providers comparisons: GCE, AWS, Azure…
- Cost reduction in some “realistic” case from cloud: (v.gr. Deep Learning Apps)
- …

Proposals

- Before to do the work, you need to present a proposal

- Should include a couple of pages
  - A description of the topic
  - Discuss why the proposal is relevant/interesting
  - What might be challenging,
  - What you plan to do, along with tentative timeline

- Should be presented in the classroom

Project Progress Report (10% of project)

- A revised version of the proposal plus one page describing the accomplishments and (if needed) modifications in the goals of the original proposal

- Should be presented in the classroom

- All projects has to have a "**github**" project and vpuente@unican.es should be Developer (Other groups should be guests)

Proposal and/or Survey Presentation

- Final report
  - Reports should consist of an abstract, body and optional appendices, much like a conference paper.
  - The abstract should summarize the contributions of the report in one or two paragraphs.
  - The length of the body should be ~8000 words.
  - Additional supporting material of any length can be put in appendices.
- Presentation
  - The author/s present the work to the class
  - Practice before to present the work
  - Conference paper format
- Work ethics in Projects
  - Its your responsibility to organize the work. The grade will be equal to all authors.

Grading

- 50% Lab
  - 10% Warmup Labs
  - 40 % Project/Survey
- 50% Final exam
  - Includes material from the regular lectures

Disclaimer

- 3-to-1