Cloud Architecture

Nishant Saurabh

Utrecht University

Agenda for today

- 13:15 – 14:15: Cloud Architecture
- 14:30 – 14:45: Lab Assignment: I
- 14:45 – 15:45: Lab-I Assignment time
- 16:00 – 16:45: Assignment time
- 16:45 – 17:00: Wrap-up

Service-oriented Architecture?

Utrecht University

Definition 1

Service –oriented architectures (SOAs) are way of developing distributed systems where system components are stand-alone services , executing on geographically distributed servers

Utrecht University

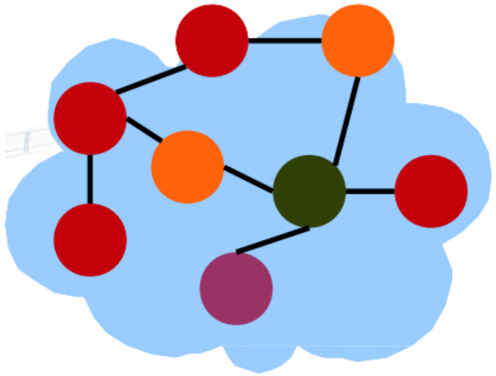
Definition 2

A paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. It provides uniform means to offer, discover, interact with and use capabilities to produce desired effects consistent with measurable preconditions and expectations

OASIS (Organization for the Advancement of Structured Information Standards)

Utrecht University

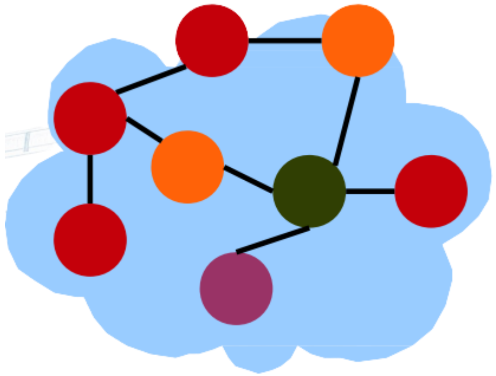
Components of SOA



- Services
  - **Loosely-coupled reusable software component**
  - **Encapsulates discrete functionalities**
  - **Distributed and programmatically accessed**
- Messages
  - **Exchange of information**
- Meta-data
  - **Service description**
  - **Service interface**
  - **Service metadata**

Utrecht University

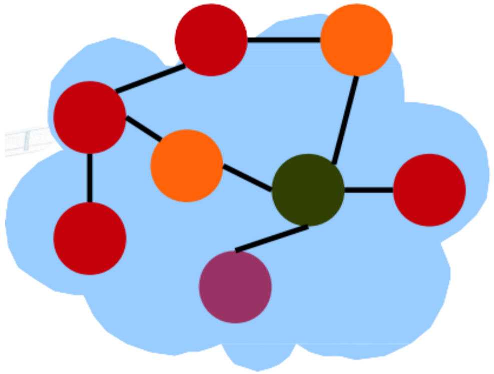
SOA characteristics



- Integration of SOA architecture
- Application
- Scope
- Dependency
- Stateful vs stateless
- Loose coupling
- Reusability
- Platform and implementation independent
- Discoverable

Utrecht University

SOA considerations?



- Computational penalties
- Extra layers
- Higher communication latency
- Supporting native applications
- Service granularity (reusability or performance)
- Fault tolerance (partial or full failure)
- Service agreements (performance and cost)
- Governance (how to manage and orchestrate)

Utrecht University

Cloud Architecture

You have house to rent

- What does the tenant want?
- What can you offer ?

Utrecht University

You have house to rent

- What does the tenant want?
- What can you offer ?

- Is it affordable ?
- Is it spacious?
- Will I be disturbed by outsiders?
- Will energy cost be billed separately?

You have a computer to rent

- What does the tenant want?
- What can you offer ?

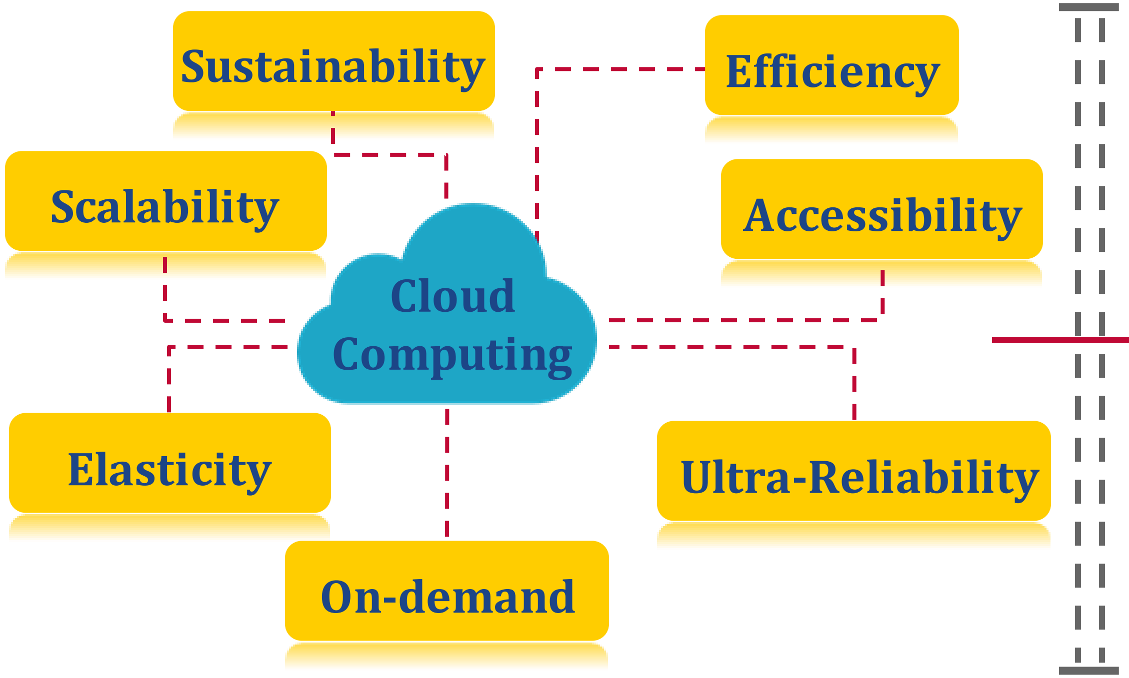
- Is it affordable to rent?
- Is there enough CPU/Memory/Disk?
- Is network connection efficient?
- Do I pay for what I use?

NIST Definition of Cloud

A model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g. networks, servers, storage, applications and services) that can rapidly provisioned and released with minimal management effort or service provider interaction

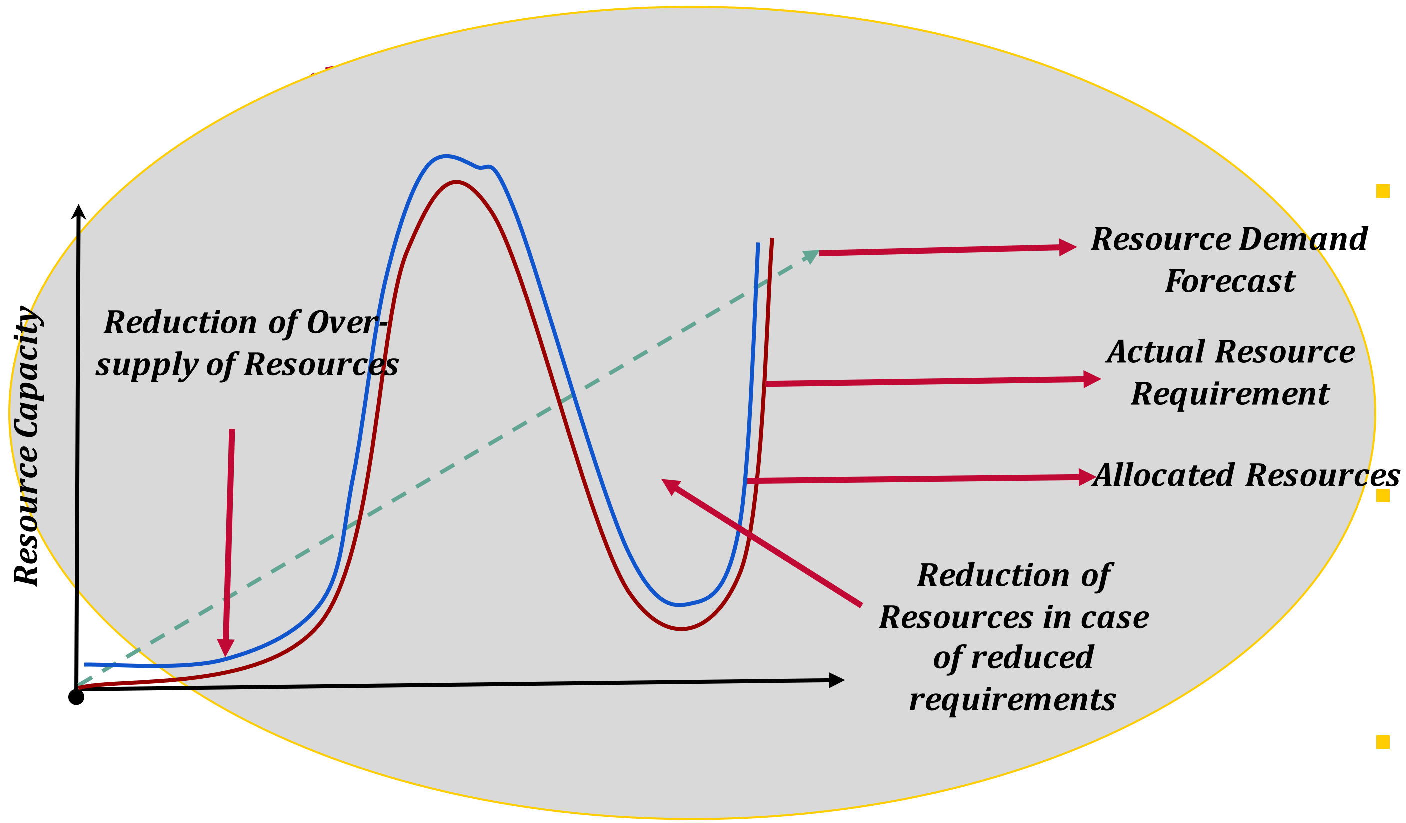
Utrecht University

What makes Cloud Ubiquitous?



- On-demand
- Access to resources
- Outside the premises?
- Shared by others
- Simple to use
- Cheap
- Pay as you use
- Accessed over the internet
- Security
- Reliability

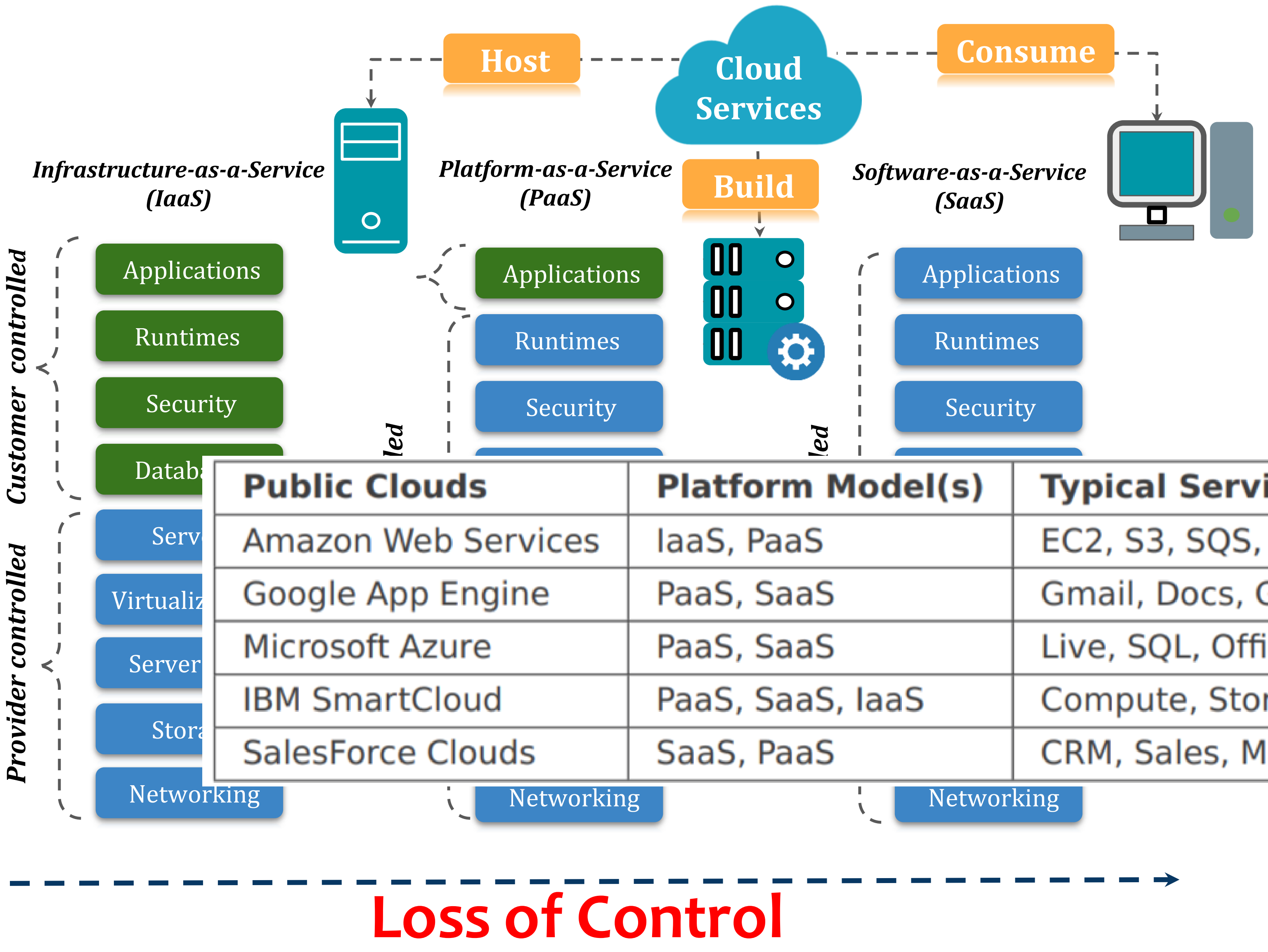
What makes Cloud Ubiquitous?



- On-demand self service
  - **Unilaterally provision computing capabilities without requiring human interactions**
- Broad network access
  - **Available over the network**
  - **Heterogenous thin or thick client platforms (mobile phones, laptops)**
- Resource pooling
  - **Compute resources serve multiple users**
  - **Multi-tenant model**
- Metering capability
  - **Monitoring services**
  - **Automatic control and optimization**
- Rapid elasticity
  - **Scale out**
  - **Scale in**

What are Cloud service models?

Cloud service models

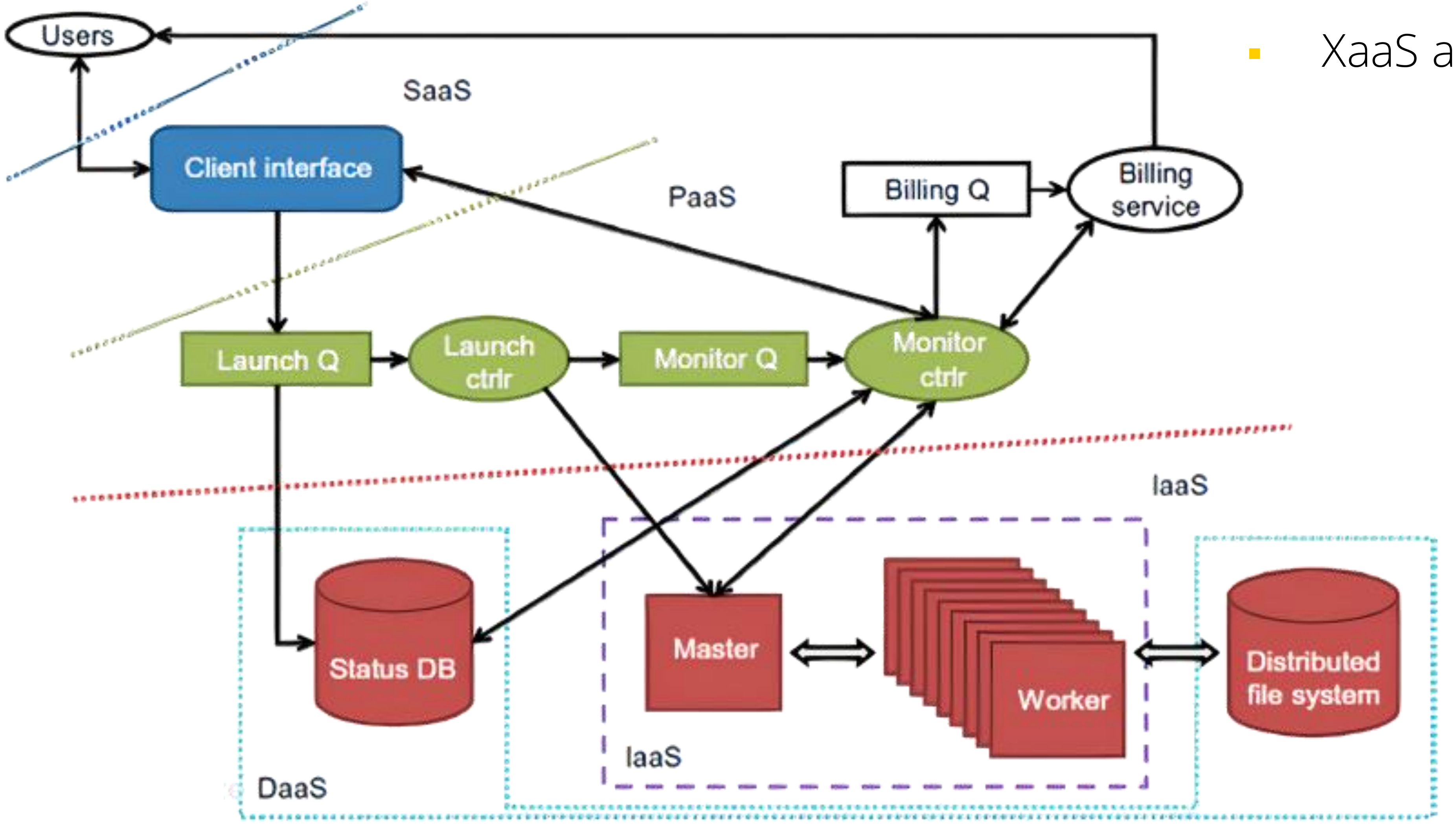


- Software-as-a-Service
- Platform-as-a-Service
- Infrastructure-as-a-Service

Public Clouds	Platform Model(s)	Typical Service Offerings
Amazon Web Services	IaaS, PaaS	EC2, S3, SQS, EMR, VPC, EBS, SNS, Cloudfront, etc.
Google App Engine	PaaS, SaaS	Gmail, Docs, GFS, BigTable, Chubby
Microsoft Azure	PaaS, SaaS	Live, SQL, Office 365, Dynamic CRM
IBM SmartCloud	PaaS, SaaS, IaaS	Compute, Storage, Backup, Networking, Virtualization
SalesForce Clouds	SaaS, PaaS	CRM, Sales, Marketing, Apex, VisualForce

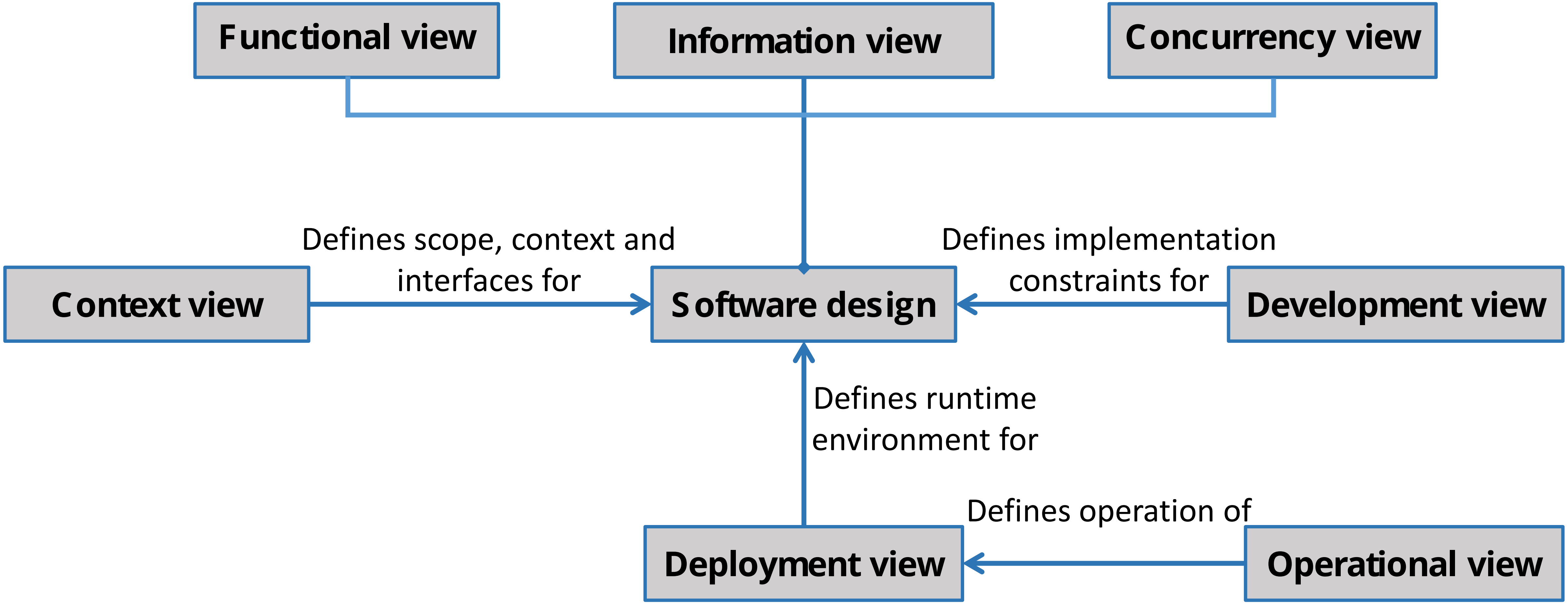
Utrecht University

XaaS: Everything-as-Service



- XaaS architectural hierarchy

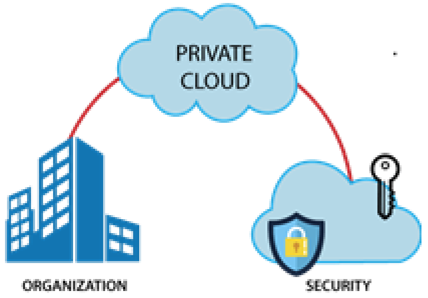
Where does Cloud fit?



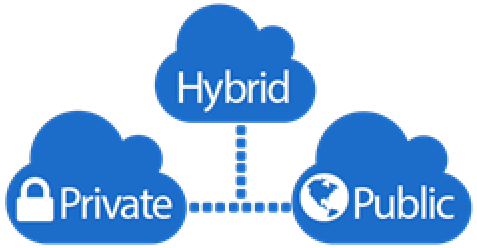
Cloud deployment models

- Public Cloud
  Available to all
  Example: AWS, GCP

- Private Cloud
  On-premise
  Example: OpenStack



- Hybrid Cloud
  Private + Public
  E.g. Rackspace Cloud



- Community Cloud
  Shared by organisations with similar goals
  Example: salesforce

Utrecht University

Why do we need Cloud?

Economies of Scale

- Large data centers cheaper to operate
- Large means 100, 000+ servers
- Small means less than 10,000 servers

Economies of Scale

- Large data centers cheaper to operate
- Large means 100, 000+ servers
- Small means less than 10,000 servers

Microsoft datacenter: 11.5 times the size of a football field ? Still economic ?

Utrecht University

Economies of Scale

- Cost of power
  - **Lower per server**
- Infrastructure labor costs
  - **Administrator services more than 1000 servers**
- Security and reliability investment
  - **Disaster recovery**
- Hardware costs
  - **Large scale hardware purchase**

Microsoft datacenter: 11.5 times the size of a football field ? Still economic ?

Utrecht University

Utilization

- Collocated services and applications
- Variations in workload
- Time of the day, year
  - **Resource usage patterns**
  - **Uncertainty (e.g. news events, marketing events)**

Microsoft datacenter: 11.5 times the size of a football field ? Still economic ?

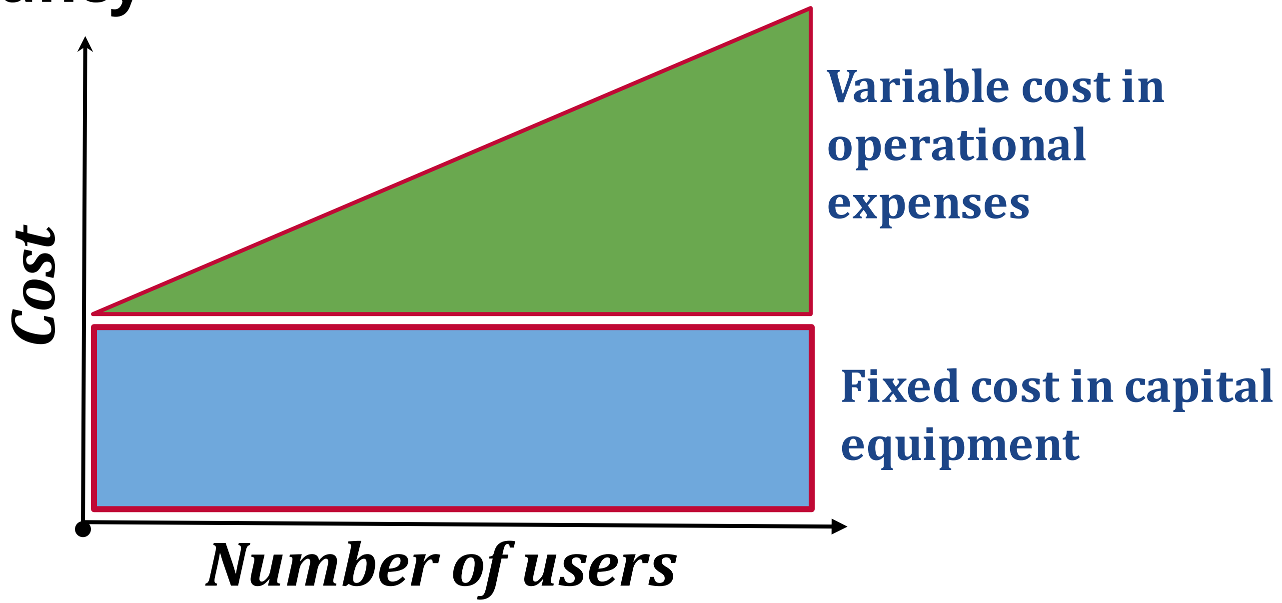
Multitenancy

- Salesforce
  - **Single application for multiple users**
- Reduced costs
  - **help desk support**
  - **upgradation costs**
  - **software development and maintenance costs**

Microsoft datacenter: 11.5 times the size of a football field ? Still economic ?

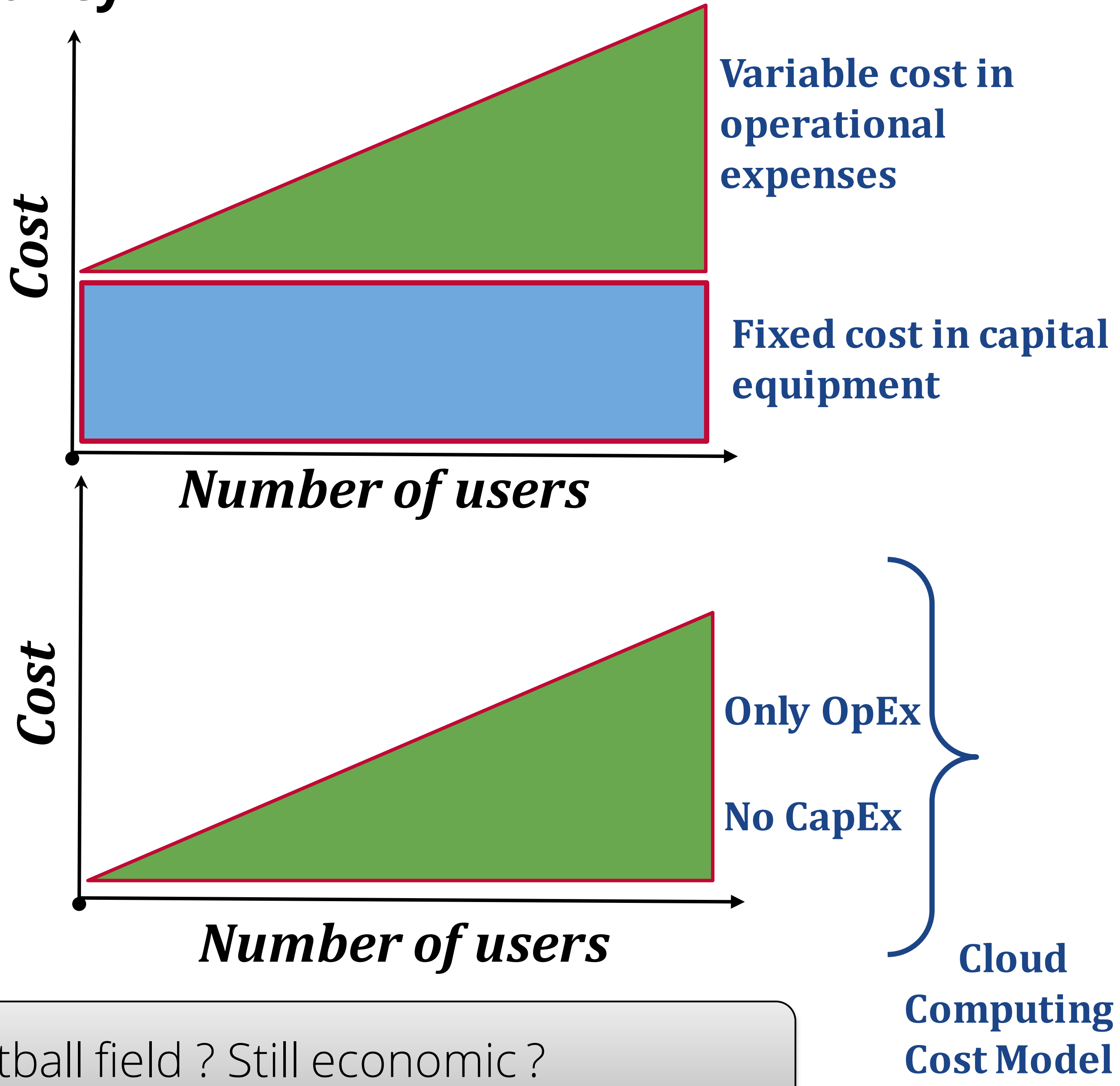
Utrecht University

Multitenancy



Microsoft datacenter: 11.5 times the size of a football field ? Still economic ?

Multitenancy



Microsoft datacenter: 11.5 times the size of a football field ? Still economic ?

Utrecht University

Virtualization: VMs to Serverless

Utrecht University

Virtualization

- Create a software-based-or virtual-representation of applications, servers, storage and networks to reduce IT expenses while boosting efficiency and agility.

Virtualization

- Create a software-based-or virtual-representation of applications, servers, storage and networks to reduce IT expenses while boosting efficiency and agility.

Virtualization != Cloud computing ?

Virtualization abstraction levels

Level of Virtualization	Functional Description	Example Packages	Merits, App Flexibility/ Isolation, Implementation Complexity
Instruction Set Architecture	Emulation of a guest ISA by host	Dynamo, Bird, Bochs, Crusoe	Low performance, high app flexibility, median complexity and isolation
Hardware-Level Virtualization	Virtualization on top of bare-metal hardware	XEN, VMWare, Virtual PC	High performance and complexity, median app flexibility, and good app isolation
Operating System Level	Isolated containers of user app with isolated resources	Docker Engine, Jail, FVM	Highest performance, low app flexibility and best isolation, and average complexity
Run-Time Library Level	Creating VM via run-time library through API hooks	Wine, vCUDA, WABI, LxRun	Average performance, low app flexibility and isolation, and low complexity
User Application Level	Deploy HLL VMs at user app level	JVM, .NET CLR, Panot	Low performance and app flexibility, very high complexity and app isolation

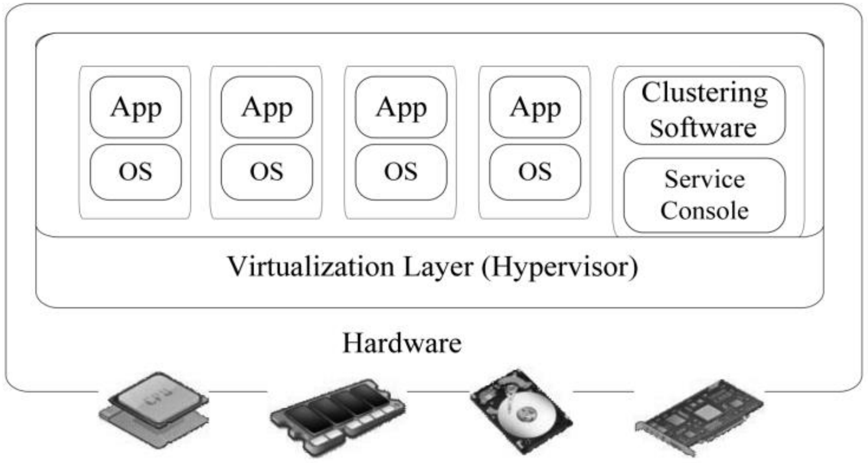
Hwang, Kai. Cloud computing for machine learning and cognitive applications. MIT press 2017.

Utrecht University

Hardware and OS-level virtualization

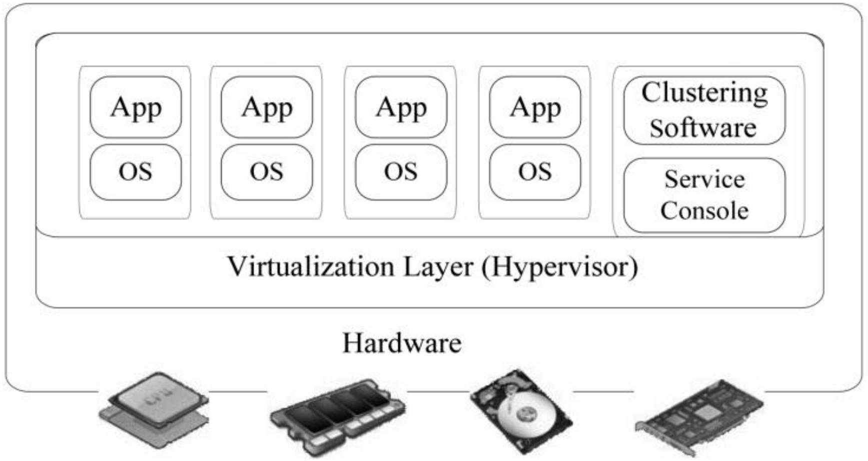
Hardware-level virtualization

- Virtual hardware environment for VMs
  - **Xen, KVM, Hyper-V, Virtual Box**
- Virtual Machine (VM)
  - **Tightly isolated software container with an OS and application inside,stateful in nature**
- Hypervisor
  - **Thin layer of software decouples VMs from host and dynamically allocates computing resources to each VM**



Hwang, Kai. Cloud computing for machine learning and cognitive applications. MIT press 2017.

Hardware-level virtualization



- Virtual hardware environment for VMs
  - **Xen, KVM, Hyper-V, Virtual Box**
- Virtual Machine (VM)
  - **Tightly isolated software container with an OS and application inside, stateful in nature**
- Hypervisor
  - **Thin layer of software decouples VMs from host and dynamically allocates computing resources to each VM**

Drawbacks: Long boot delay, Takes upto tens of GBs, VM sprawl issues

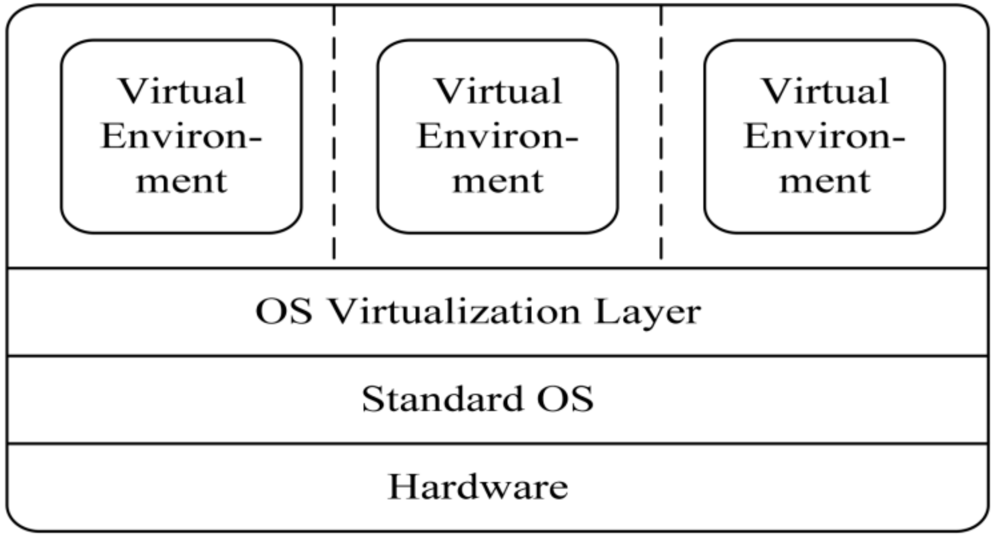
Saurabh, Nishant, et al. "Semantics-aware virtual machine image management in IaaS clouds." 2019 IEEE International Parallel and Distributed Processing Symposium (IPDPS). IEEE, 2019.

Saurabh, Nishant, et al. "Expelliarmus: Semantic-centric virtual machine image management in IaaS Clouds." *Journal of Parallel and Distributed Computing* 146 (2020): 107-121.

Utrecht University

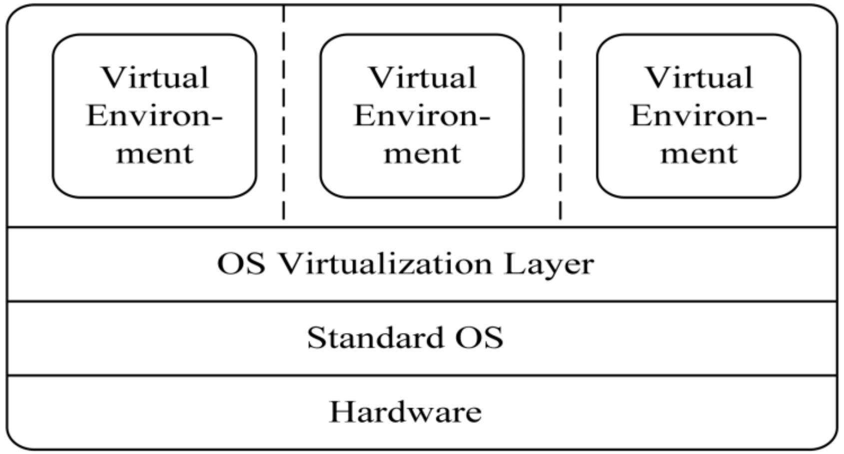
OS-level virtualization

- Containers
  - **Creates isolated containers on a single physical server or a VM to utilize hardware and software in datacenters**
  - **E.g. Dockers**



Hwang, Kai. Cloud computing for machine learning and cognitive applications. MIT press 2017.

OS-level virtualization



- Containers
  - **Creates isolated containers on a single physical server or a VM to utilize hardware and software in datacenters**
  - **E.g. Dockers, stateless in nature**
- Pods
  - **Group of related containers**
  - **Used for deploying, managing and scaling containers**

Advantages: Lower startup time compared to VMs
Drawbacks: Performance interference, Container image sizes are growing upto few GBs, more start up time

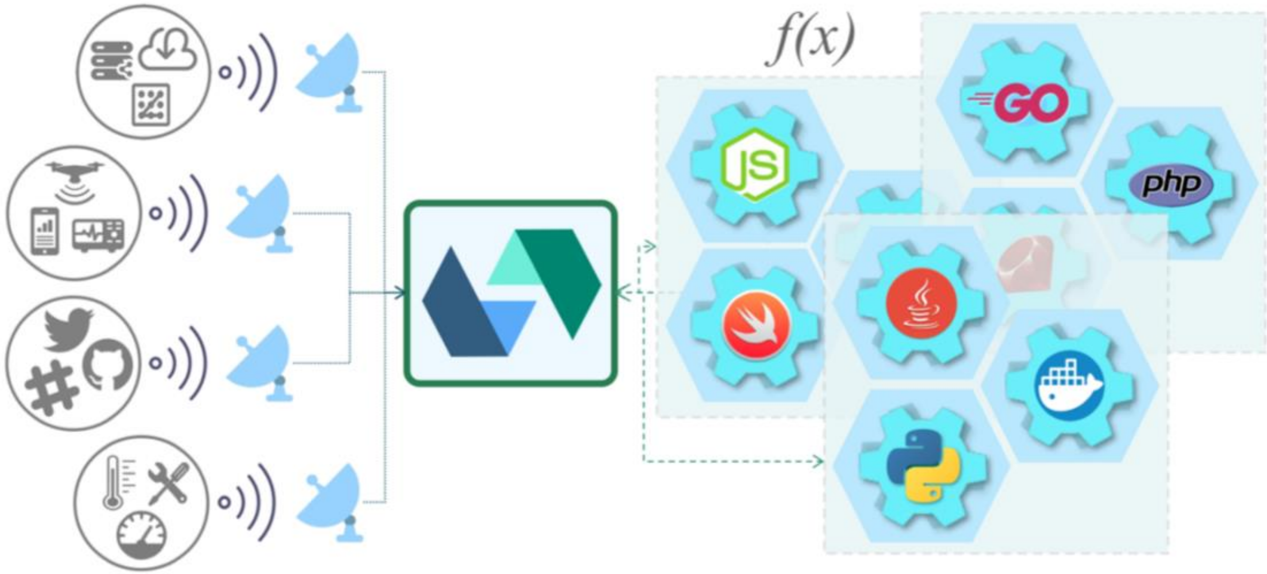
Sharma, Prateek, et al. "Containers and virtual machines at scale: A comparative study." Proceedings of the 17th international middleware conference. 2016.

Utrecht University

Serverless

Event-driven

- Datastores (e.g. S3)
- Message Queues
- Applications
- Sensors
- Scheduled tasks (via Alarms)



- Function as a Service (FaaS)
  - **Light weight functions**
    - **Can be stored in a repository**
  - **Runtime system to run any kind of code**
    - **Java, Node.js, Python etc.**
- Place functions in containers
- Real pay as you go model
  - **Use resources only when you run functions**
- E.g. AWS Lambda, Azure functions

VMs/Containers vs. Serverless

- VM and containers reserve resources without a load
  - **Can be costly**
- Serverless uses resources only when a function is invoked
  - **Provider waits for requests**
- Serverless computing
  - **Also serverless DB, etc.**
- Disadvantages
  - **Cold start problem**

Architecting in a Cloud environment

Utrecht University

Architecting in a Cloud Environment

- Quality attributes that are different in a cloud
  - **Security and Consistency**
  - **Performance**
  - **Availability**

Security and Consistency

- Multi-tenancy concerns over non-cloud environments
  - **Inadvertent information sharing : shared use of resources**
  - **Virtual machine escape : breaking the hypervisor**
  - **Side channel attacks: monitoring cache**
  - **Denial of service attacks**
- Some systems require:
  - **state management, time coordination and data coordination**
- You need to consider risks when deciding what applications to host in the cloud
- Also which services need consistency measures

Performance

- Auto scaling provides additional performance when load grows
  - **CPU utilization, network and IO bandwidth utilization thresholds**
  - **Response time for new resources may not be adequate**
- As an architect you need to be aware of resource requirements for you system
  - **Build and integrate the knowledge**
  - **Make system self-aware and proactive**
  - **Use of load balancers**

Availability

- Failure: a regular phenomena
  - **Response time for new resources may not be adequate**
- Cloud providers promise an SLA (Service level agreement)
  - **99.999999…**
  - **More 9s after decimal makes your system more reliable**
- As a software architect, you must assume failures
  - **Build detection and correction mechanisms**
- Tactics:
  - **Timeout**
    - **Can not differentiate between failures and slow response**
    - **Might be Long tail latency**

Agenda for today

- 13:15 – 14:15: Cloud Architecture
- 14:30 – 14:45: Lab Assignment: I
- 14:45 – 15:45: Lab-I Assignment time
- 16:00 – 16:45: Assignment time
- 16:45 – 17:00: Wrap-up

Utrecht University

For Next Time

- Future computing architectures
  - Edge and Quantum architectures
- Deadline Lab Assignment-I: March 8th AOE

Utrecht University

DISCLAIMER