cphbusi

COPENHAGEN BUSINESS ACADEMY





Virtualisation and services

Jens Egholm Pedersen
<jeep@cphbusiness.dk>

Agenda

- Why virtualisation
- Virtualisation techniques
- The cloud
  - X as a service
- Docker
- Serverless computing

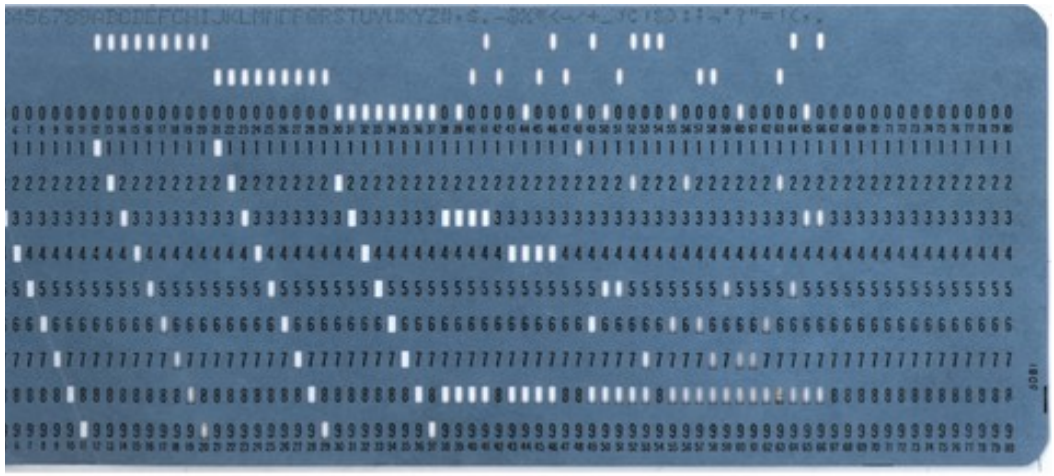
History of computing

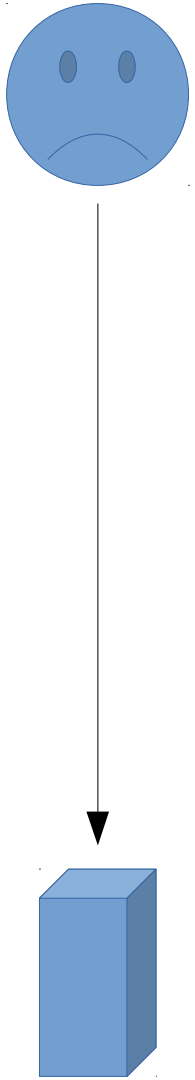
- Manual computing
- Mainframes
- Racks
- Virtualisation

See also: Evolution of computing at CERN

Early mainframes

- One huge computer
- Humans wrote machine code
  - Punchcards and later consoles

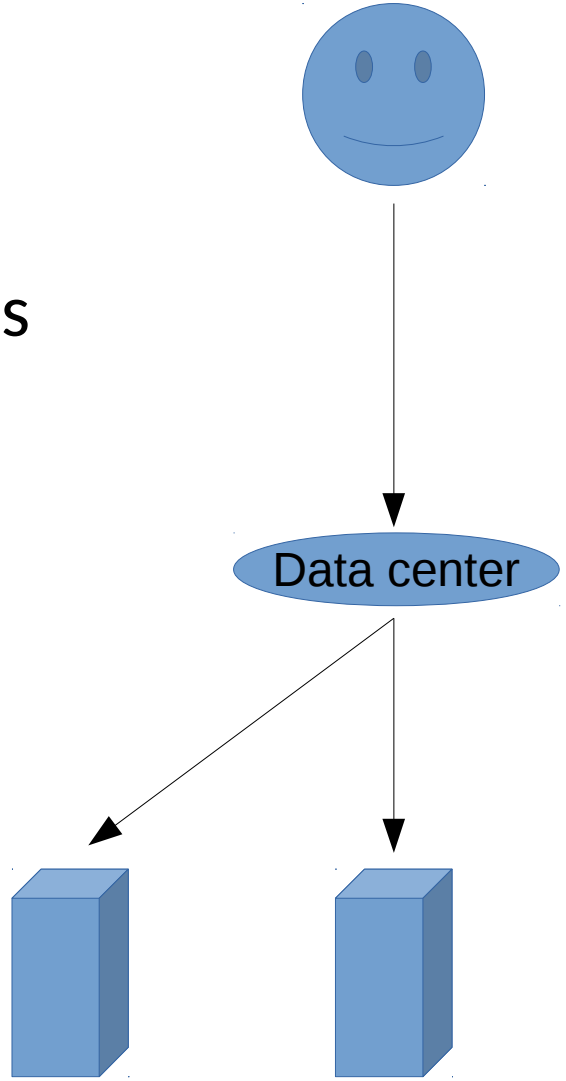




Racks

- Many computers in one place
- Stored in a data center
- Few jobs distributed to many machines

- How to interact with the racks?

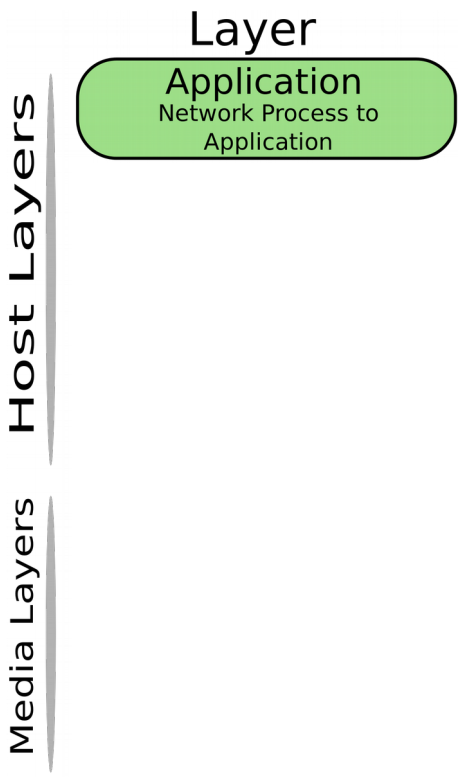


cphbusiness

OSI

- Layer interacts up and down
- We use the application layer
- We (rarely) care about others

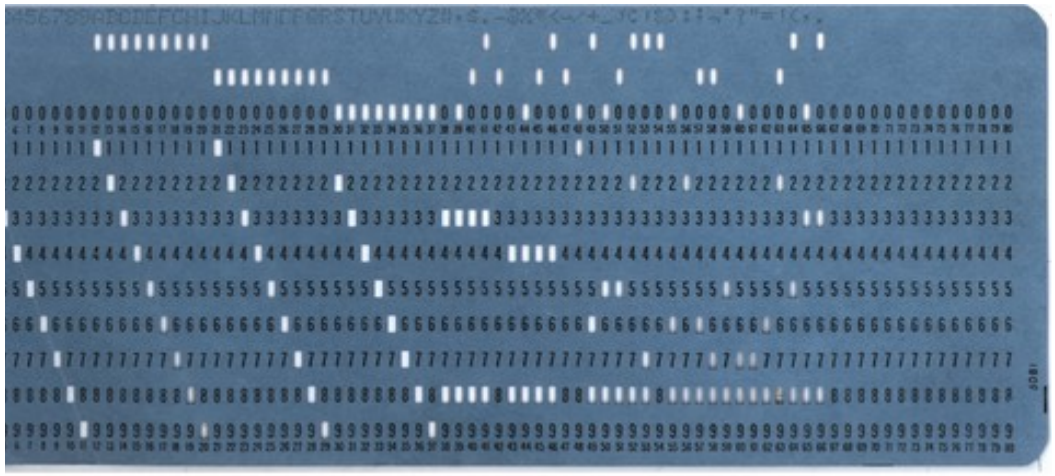
- The application layer has to communicate with the OS

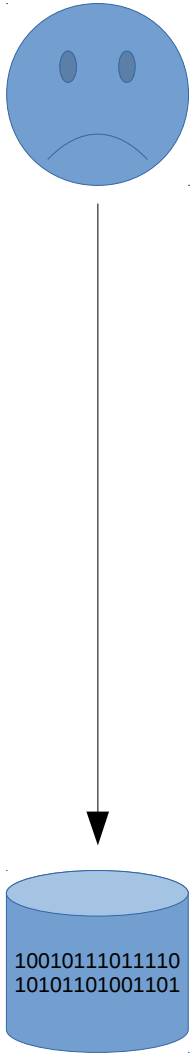


See also: OSI model on Wikipedia

Early operating systems

- Human inputs machine code
  - Punchcards

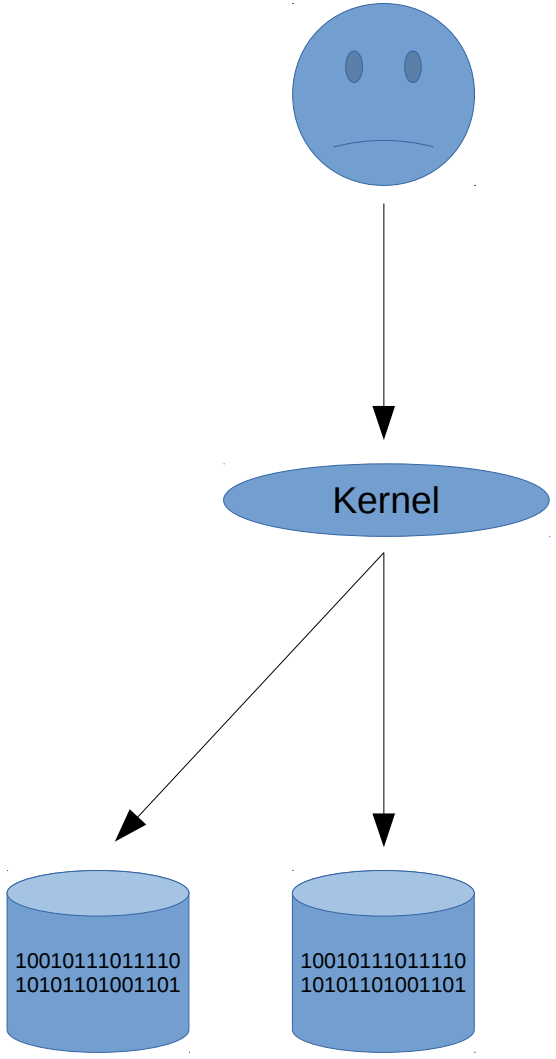




Later operating systems

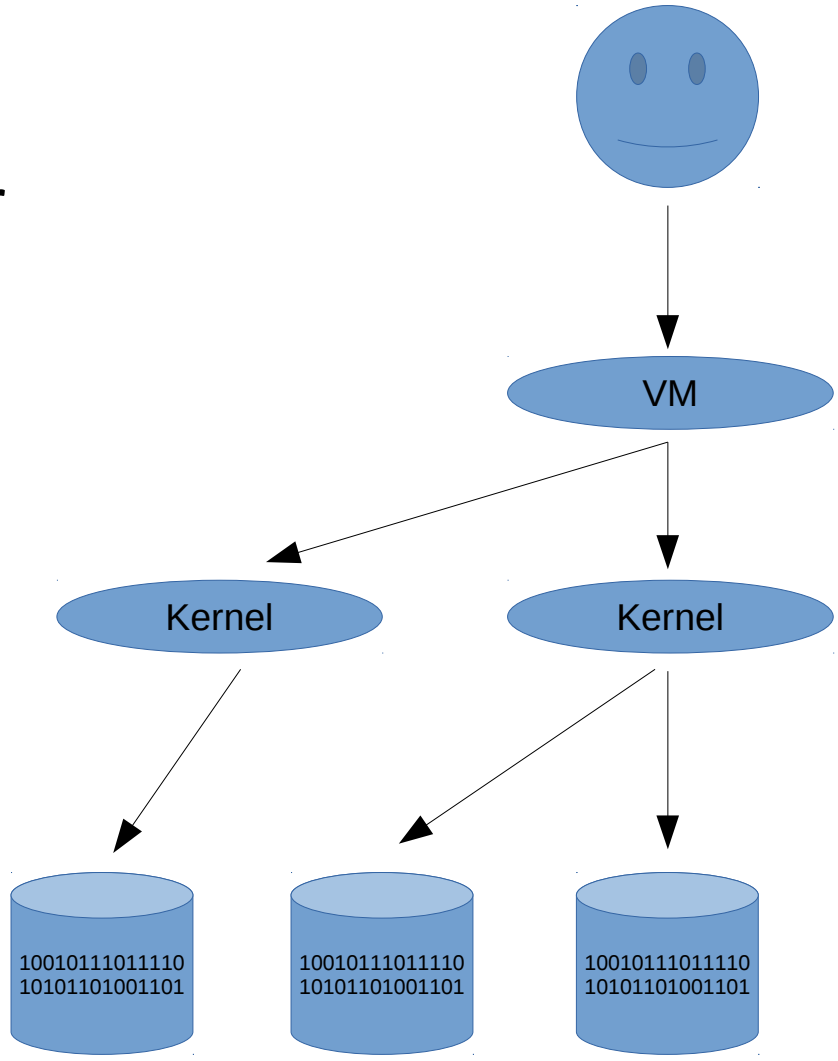
- Multiple resources, cores, users
- How to manage them?

- Operating system kernel
  - User space
  - Kernel space



Virtual machine

- Emulated operating systems
- Single entrypoint for the user
  - Not OS specific



See also: Virtual machine on Wikipedia

Virtual machine benefits

- Write once, run everywhere
- Share resources
  - Exploit economies of scale
- Scaling
  - Elasticity
  - Resources provided on-demand

See also: Docker

Virtual machine disadvantages

- Less efficient than bare-metal
  - *All problems in computer science can be solved by another level of indirection*
  - *No performance problem cannot be solved by removing a level of indirection*
- Centralisation of data = security risk
- Putting data in someone else's computer

Virtualisation

- Abstraction of resources
- Implementation decides how and where
- Typically managed by a Hypervisor
  - Creates and runs virtual machines

- Similar to the OSI model of abstraction
  - When you interact with a layer, you don't care about the implementation

See also: Virtualisation on Wikipedia

The ‘cloud’

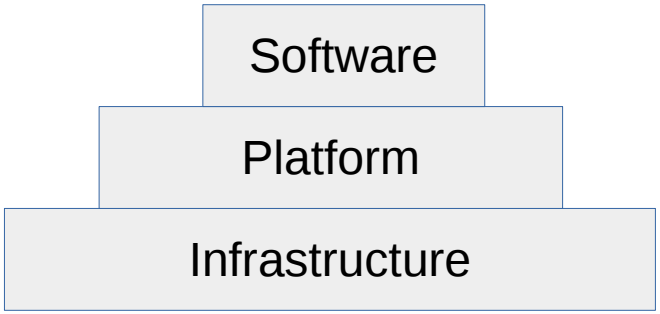
- Huge data centers running virtualised environments
- Examples: Amazon Web Services (AWS)
  - Distributed across the entire world (fast!)
- 2011: Netflix migrated to AWS

- Remember
  - Always someone else’s computer

See also: Cloud computing on Wikipedia

X as a Service

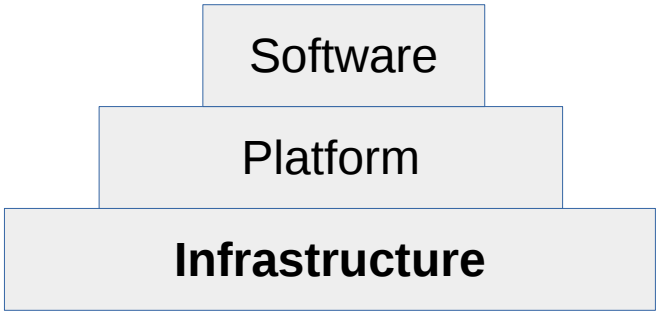
- Three variants of the cloud:
  - Infrastructure
  - Platform
  - Software



Infrastructure as a Service

- Write software for VM, scale dynamically
- Examples
  - Digital Ocean, Amazon Web Service

- Advantages
  - Pay as you go
  - Low vendor lock-in
- Disadvantages
  - Still need to interact with OS / VM
  - Critical infrastructure out of your hands

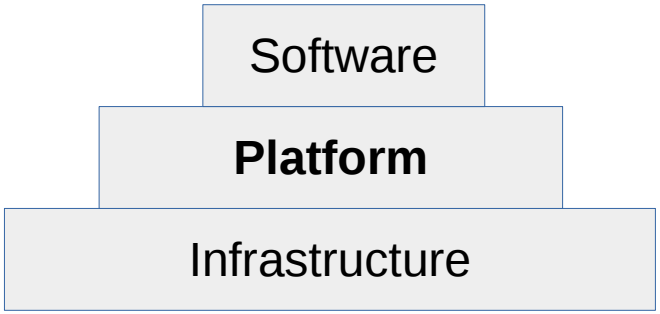


cphbusiness

Platform as a Service

- Full development environment
  - Typically: Database, web-server, execution environment
- Examples
  - Openshift, Heroku

- Advantages
  - Pay as you go
  - No need to know the VM environment
  - Built-in scalability and availability
- Disadvantages
  - Medium vendor lock-in

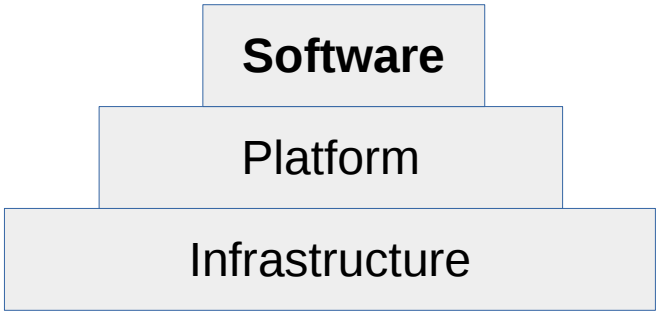


cphbusiness

Software as a Service

- Buy access to application or data
- Examples
  - Gmail, Facebook, Dropbox

- Advantages
  - Pay as you go
  - Easy to use
- Disadvantages
  - High vendor lock-in
  - No control over security



X as a Service

- As Momondo, which X would you choose?

Local	IaaS	PaaS	SaaS
Data	Data	Data	Data
Runtime	Runtime	Runtime	Runtime
Virtualisation	Virtualisation	Virtualisation	Virtualisation
Hardware	Hardware	Hardware	Hardware

Private hosting

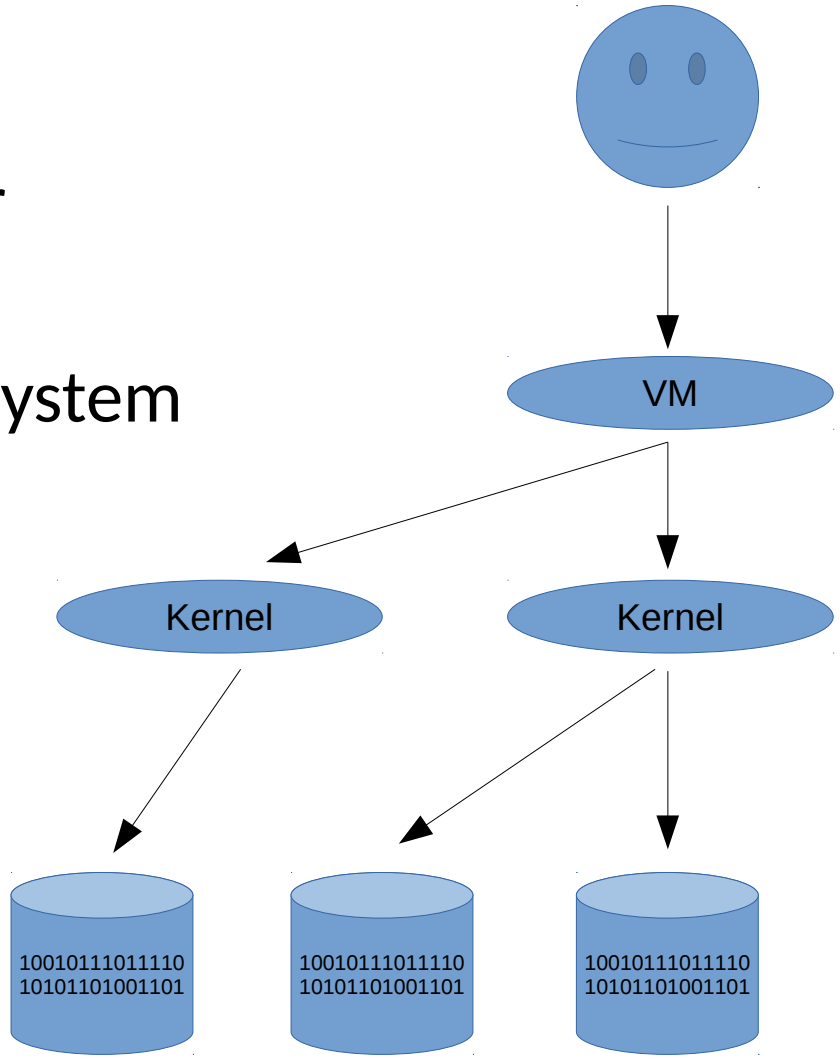
- Running software on local datacenter

- Advantages
  - Full control over hardware, OS, software and data
  - Easier to secure
  - Cheap in the long run
- Disadvantages
  - Expensive in the beginning
  - Unused hardware
  - Bad scaling
    - Slow and expensive
  - Worse infrastructure

cphbusiness

Virtual machine

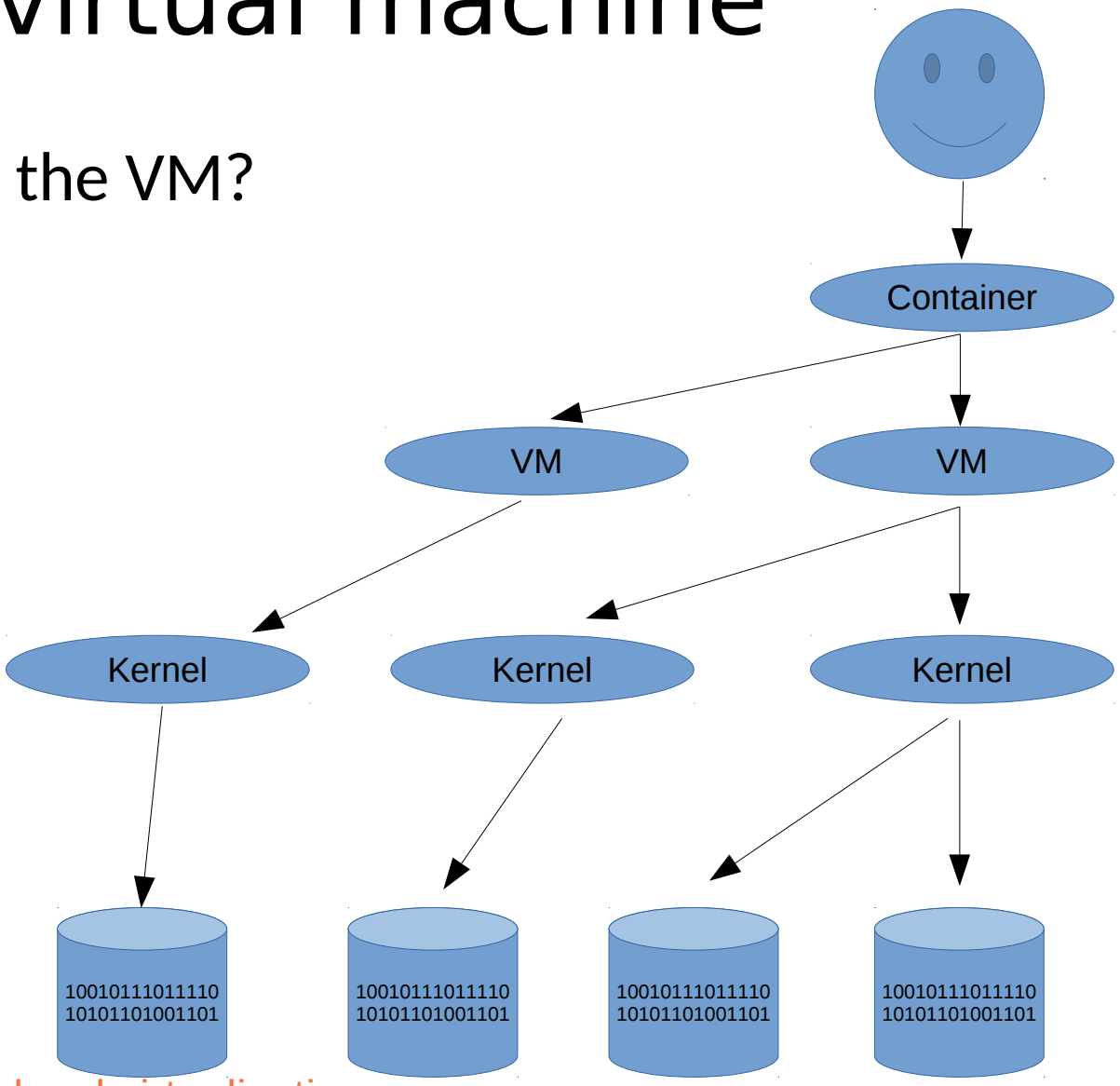
- Emulated operating systems
- Single entrypoint for the user
  - Not OS specific
- …. But it is still an operating system
  - “It works on my machine”



See also: Virtual machine on Wikipedia

Virtualised virtual machine

- Can we virtualise the VM?
  - Yes we can!



See also: Operating-system-level virtualisation

Containerisation

- One version: Docker
  - Basically just translates kernel calls
- Similar to a shipping analogy

See also: Operating-system-level virtualisation

Recap

- Virtualisation
- ‘Cloud’ computing (= NSA)
- XaaS
- Containerisation
  - Containers that can run anywhere

Lambda services

- We reduced our applications to containers
- Why not boil it further down?
  - What does an application typically consist of?
  - Functions (lambdas): some input and some output
- Now we don’t even need the idea of a machine
  - We can just provide a single function, that can be executed

Serverless cloud computing

- A new technology
- … Nope!
- There is no such thing as no server!
  - There is only someone else's PC