Emerging Computing Architectures in Machine Learning

Daniel Holmberg 22.3.2020

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

TABLE OF CONTENTS

01
Introduction

Remote Computing Timeline

early 2000s

- Application service providers (ASPs)
- Rapid development of processing and storage technologies,
- More widely-spread and faster internet

2006

AWS EC2 release: enabled compute in the **cloud**, pay only for capacity that you actually use

2012

Fog computing introduced to facilitate the deployment of distributed, latency-aware services

2015

Edge computing for applications where real-time processing of data is required

2020s

5G + edge computing = <3
Low latency first hop to edge servers

Modern ML Computing Timeline

2011

DistBelief: ML on Google's internal computing clusters

2015

ML application-specific integrated circuits (ASICs)

2016

- Tensor processing unit (TPU) announced
- AlphaGo beats Lee Sedol at Go

2017

- TPUs available in the cloud
- TensorFlow 1.0: open source ML and dataflow library
- TensorFlow Lite with mobiles in mind

2019

- AWS Sagemaker Neo: neural net reduces model memory footprint for running on edge devices
- TensorFlow 2.0 released
- TensorFlow Lite Micro for microcontrollers

02
Cloud Computing

Cloud Computing Architecture

End User	
Application	Business Applications, Web Services, Multimedia
Platforms	Software Framework (Java/Python/.NET) Storage (DB/File)
Infrastructure	Computation (VM) Storage (block)
Hardware	CPU, Memory, Disk, Bandwidth

Cloud Computing Architecture

End User

Application
Business Applications, Web Services, Multimedia

Platforms
Software Framework (Java/Python/.NET)
Storage (DB/File)

Infrastructure
Computation (VM) Storage (block)

Hardware
CPU, Memory, Disk, Bandwidth

Cloud Computing Architecture

End User	
Application	Business Applications, Web Services, Multimedia
Platforms	Software Framework (Java/Python/.NET) Storage (DB/File)
Infrastructure	Computation (VM) Storage (block)
Hardware	CPU, Memory, Disk, Bandwidth

Cloud Computing Architecture

End User	
Application	Business Applications, Web Services, Multimedia
Platforms	Software Framework (Java/Python/.NET) Storage (DB/File)
Infrastructure	Computation (VM) Storage (block)
Hardware	CPU, Memory, Disk, Bandwidth

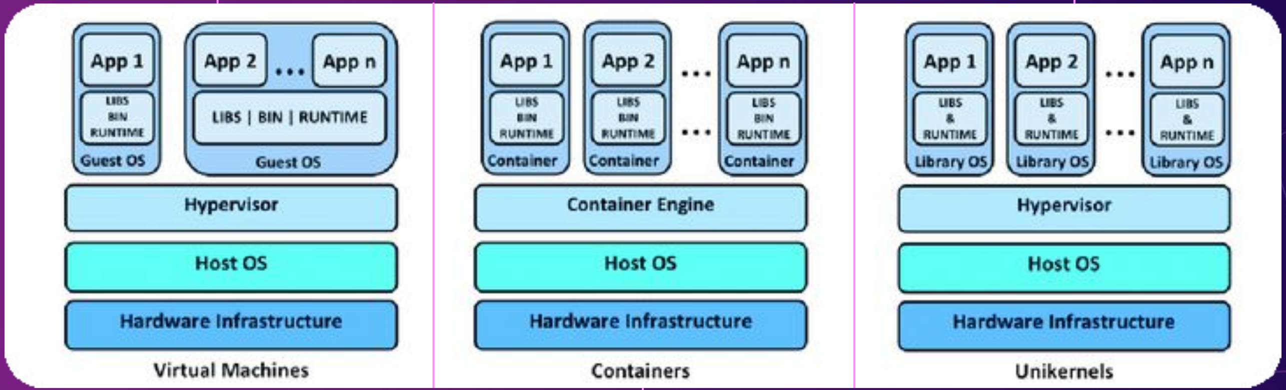
Cloud Computing Architecture

End User	
Application	Business Applications, Web Services, Multimedia
Platforms	Software Framework (Java/Python/.NET) Storage (DB/File)
Infrastructure	Computation (VM) Storage (block)
Hardware	CPU, Memory, Disk, Bandwidth

Virtualization

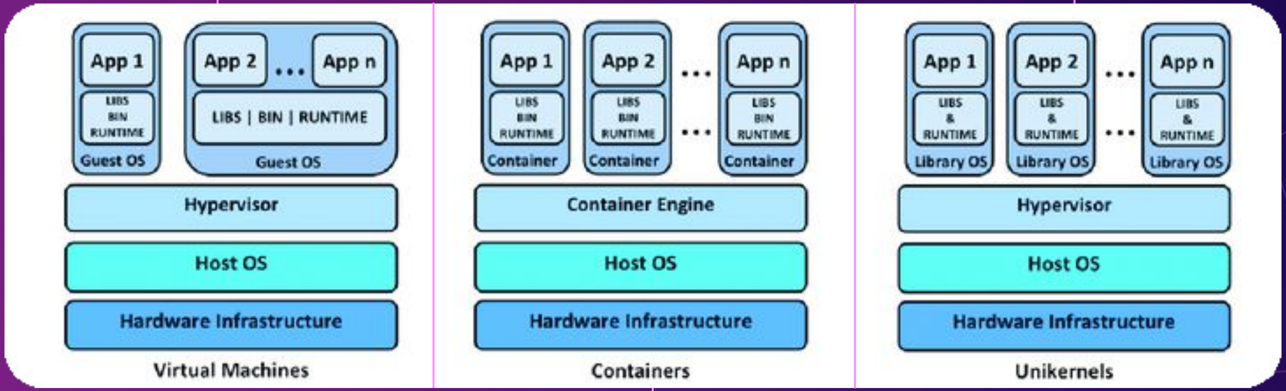
A full operating system is installed on top of the virtualized hardware.

Specialized into standalone kernels at compile time, modification is impossible after deployment.



Isolate processes at the OS level instead of virtualizing hardware and drivers.

Virtualization



Virtualization

	Instantation time	Image size	Programming language dependency	Hardware portability
VM	~5/10 s	~ 1000 Mb	No	High
Container	~800/1000 ms	~ 50 Mb	No	High
Unikernel	~< 50 ms	~< 5Mb	Yes	High

Architectural Service Models

IaaS

Control over operating systems, storage, and deployed applications, but not the underlying physical components.

PaaS

Customer provided with software development frameworks: programming languages, libraries and services + configuration of the application-hosting environment.

SaaS

Access to provider's application, but no modification of underlying code or infrastructure.
Only user-specific application configuration settings.

MLaaS

Cloud provider toolkits with user friendly interfaces and APIs with access to ML algorithms, analytics and visualisation capabilities.

Architectural Service Models

IaaS

Control over operating systems, storage, and deployed applications, but not the underlying physical components.

PaaS

Customer provided with software development frameworks: programming languages, libraries and services + configuration of the application-hosting environment.

SaaS

Access to provider's application, but no modification of underlying code or infrastructure.
Only user-specific application configuration settings.

MLaaS

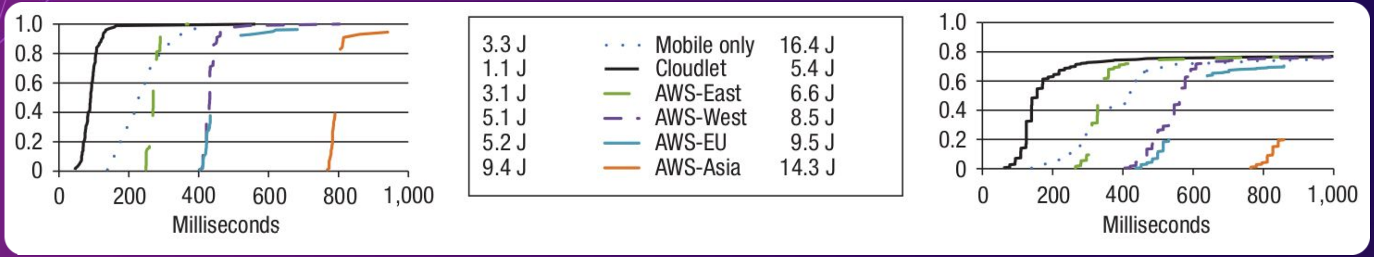
Cloud provider toolkits with user friendly interfaces and APIs with access to ML algorithms, analytics and visualisation capabilities.

03

Emergence of Edge

Edge Computing Need

(a) (b)



Mobile response time distribution and **per-operation energy cost** of an **(a) augmented reality** and **(b) face recognition** application.

An image from a mobile device in Pittsburgh is transmitted over Wi-Fi to a cloudlet or an AWS datacenter. The graphs demonstrate the need for low-latency offload services.

Advantages of Edge Computing

1. physical proximity **lowers end-to-end latency** and achieves **high bandwidth**

2. feed to cloudlets for **feature extracting** and send only that to the central server

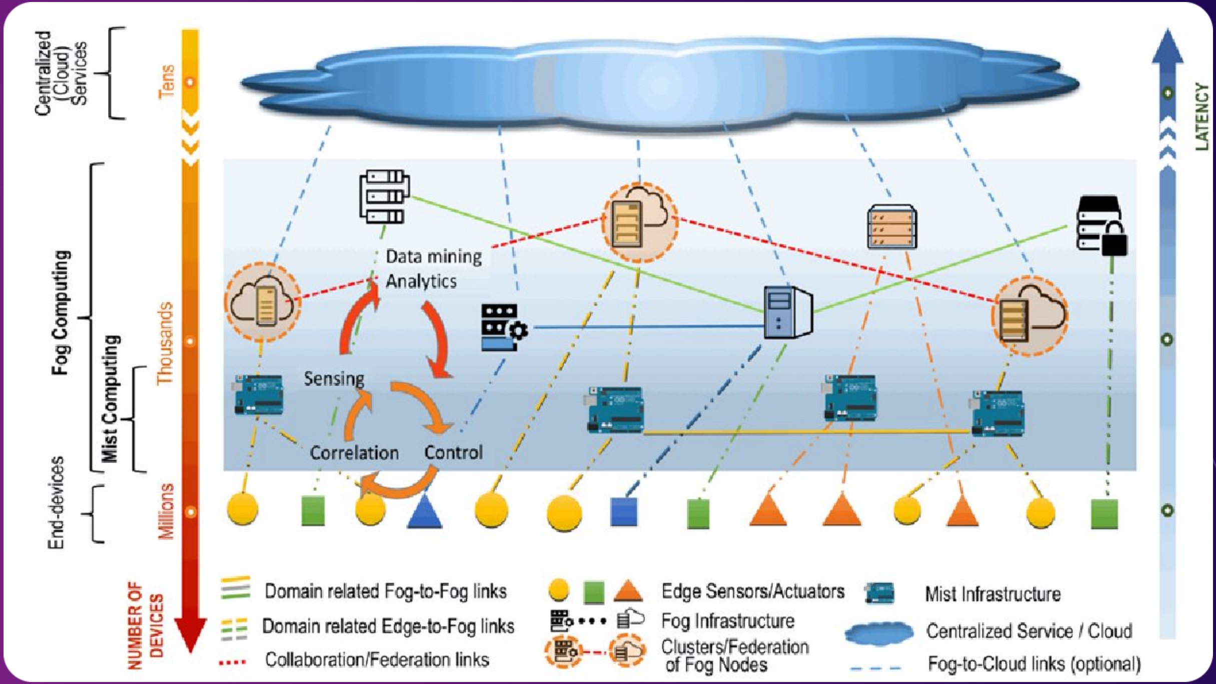
3. cloudlets can invoke **own privacy policies** prior to releasing IoT data prior further to the cloud provider

4. if a cloud service becomes unavailable, the cloudlet can conceal it temporarily with a **fallback service**

04

Fog Computing

Cloud-based Ecosystem



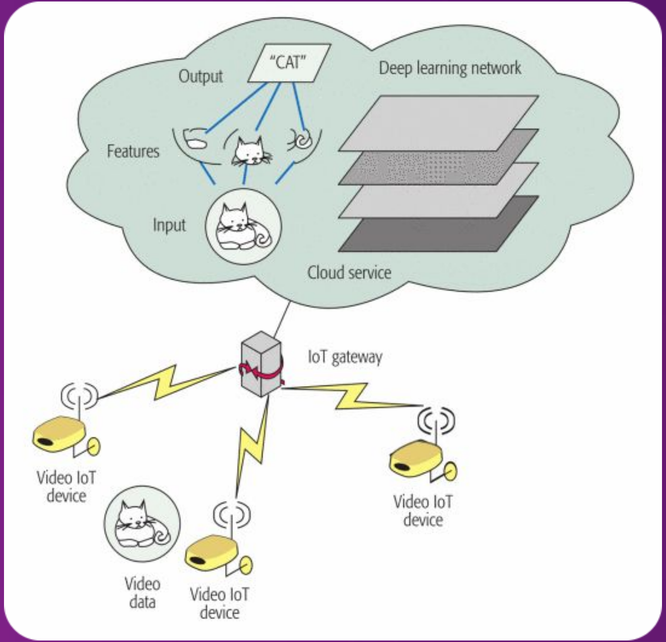
Terminology

	Components	Location	Use
Fog	Physical: gateways, servers, routers, ... Virtual: VMs, virtualized switches, cloudlets, ...	Multi-layer architecture between cloud datacenters and end devices	Computation, networking, storage, control and data-processing acceleration
Mist	Microcomputers and microcontrollers	Edge of the network fabric	Feed into fog computing nodes
Edge	Same as fog except limited to a small number of peripheral devices	IoT Network: layer with end-devices and users	Run specific applications in a fixed logic location and provide a direct transmission service

05

ML Implementation

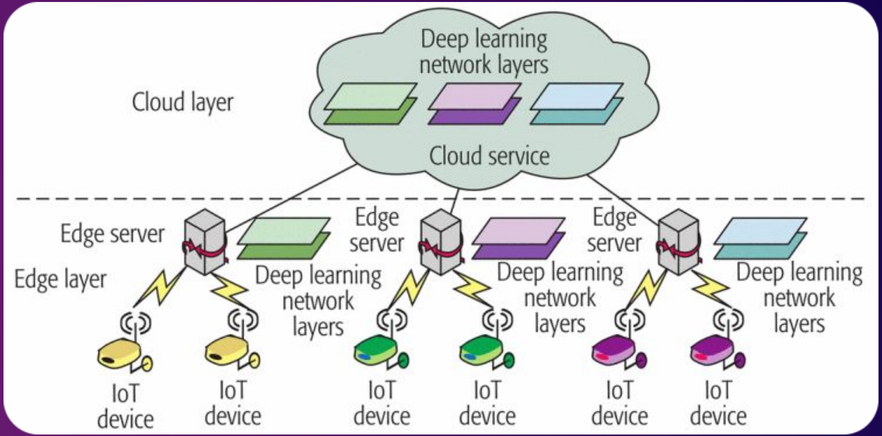
Deep Learning IoT Application



- Task: IoT video recognition
- Videofeed bit-rate: 3000 kb/s
- Enters neural net initial layer
- Output from previous layer to next layer
- Target features extracted from final layer

Deep Learning IoT Application

- Split neural net at suitable layer
- Feed reduced to 2300 kb/s in edge server
- Balancing:
  - Latency benefit from more preprocessed data to the cloud
  - Limited capacity and power of fog node processing

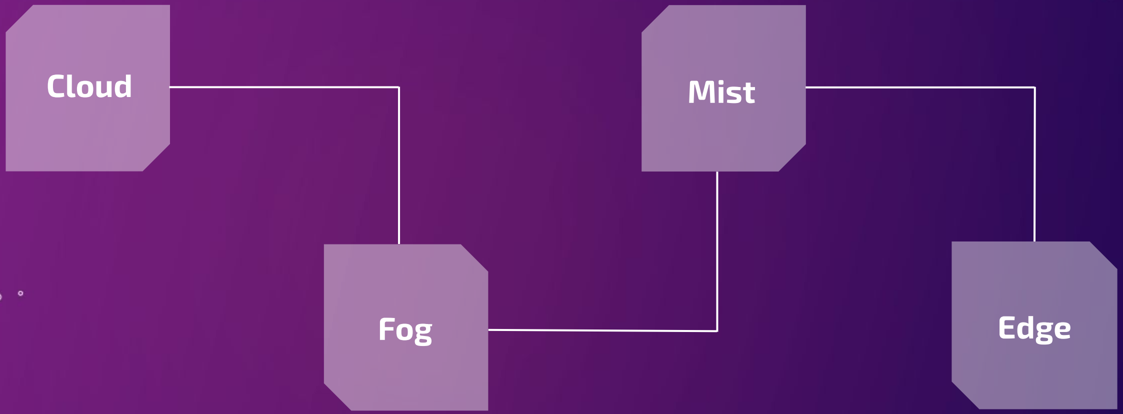


06
Conclusion

Recap

- On-demand datacenter resources
- SaaS, PaaS, Iaas and MLaaS

- Feed into fog
- Reside in network fabric
- Microdevices



- Multi-layer architecture
- IoT infrastructure scalability

- Few devices
- Mobile applications' interactive performance in mind

“Since the 1960s, computing has alternated between centralization and decentralization. The centralized approaches of batch processing and timesharing prevailed in the 1960s and 1970s. The 1980s and 1990s saw decentralization through the rise of personal computing.
By the mid-2000s, the centralized approach of cloud computing began its ascent to the preeminent position that it holds today. Edge computing represents the latest phase of this ongoing dialectic”
- Dr. M. Satyanarayanan

THANK YOU

daniel.holmberg@helsinki.fi

UNIVERSITY OF HELSINKI

Figure References

p. 12 & 13 Roberto Morabito, Vittorio Cozzolino, Aaron Yi Ding, Nicklas Beijar, and Jörg Ott. Consolidate iot edge computing with lightweight virtualization. IEEE Network, 2018.

p. 18 K. Ha et. al. The impact of mobile multimedia applications on data center consolidation. p. 166–176, 2013.

p. 21 National Institute of Standards and Technology. Fog Computing Conceptual Model: NIST SP 500-325 500-291. CreateSpace Independent Publishing Platform, 2018.

p. 24 & 25 He Li, Kaoru Ota, and Mianxiong Dong. Learning iot in edge: Deep learning for the internet of things with edge computing. IEEE Network, 32:96–101, 01 2018.