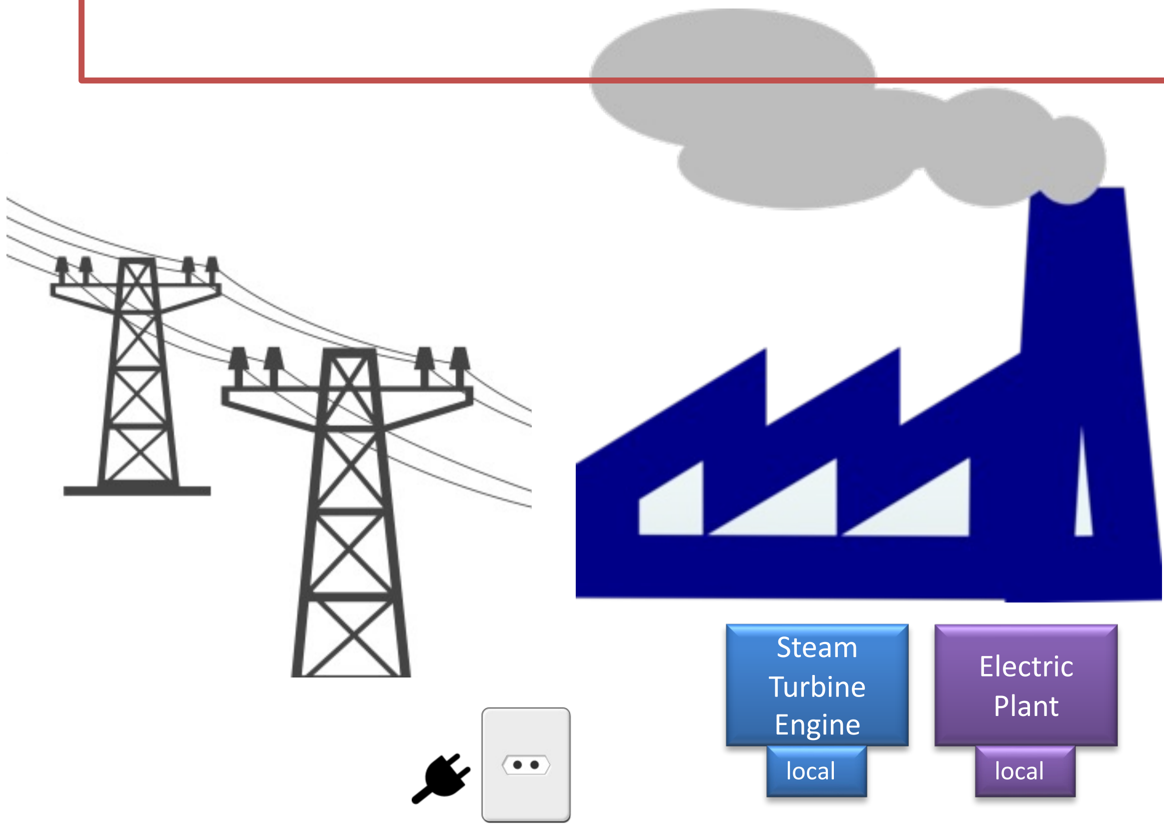
CS 436
Cloud Computing

22.02.2024

29.02.2024

Understanding Cloud Computing

Utility Computing (The Electricity Analogy)



Opportunities & Challenges

- Scalability
- Flexibility
  - Sometimes limited control over infrastructure
- Physical Area
  - Storage area for compute and network resources
  - Strict A/C
- Maintenance
  - Administrative
  - Software Updates
  - Component failures
- Data reliability (redundancy)
- Security
- Cost model
  - “pay as you go” vs “on going cost”
  - Can be complicated
- Vendor lock-in problem

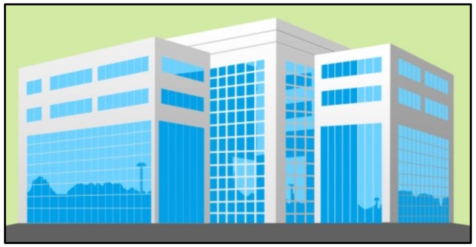
Kullanım Modelleri

(Deployment Models)

- Public Cloud
- Private Cloud
- Hybrid Cloud
- Community Cloud

Şirketler için Bulut Hesaplama

(What Cloud Computing Means for Businesses)



Startup

SMEs of various sizes

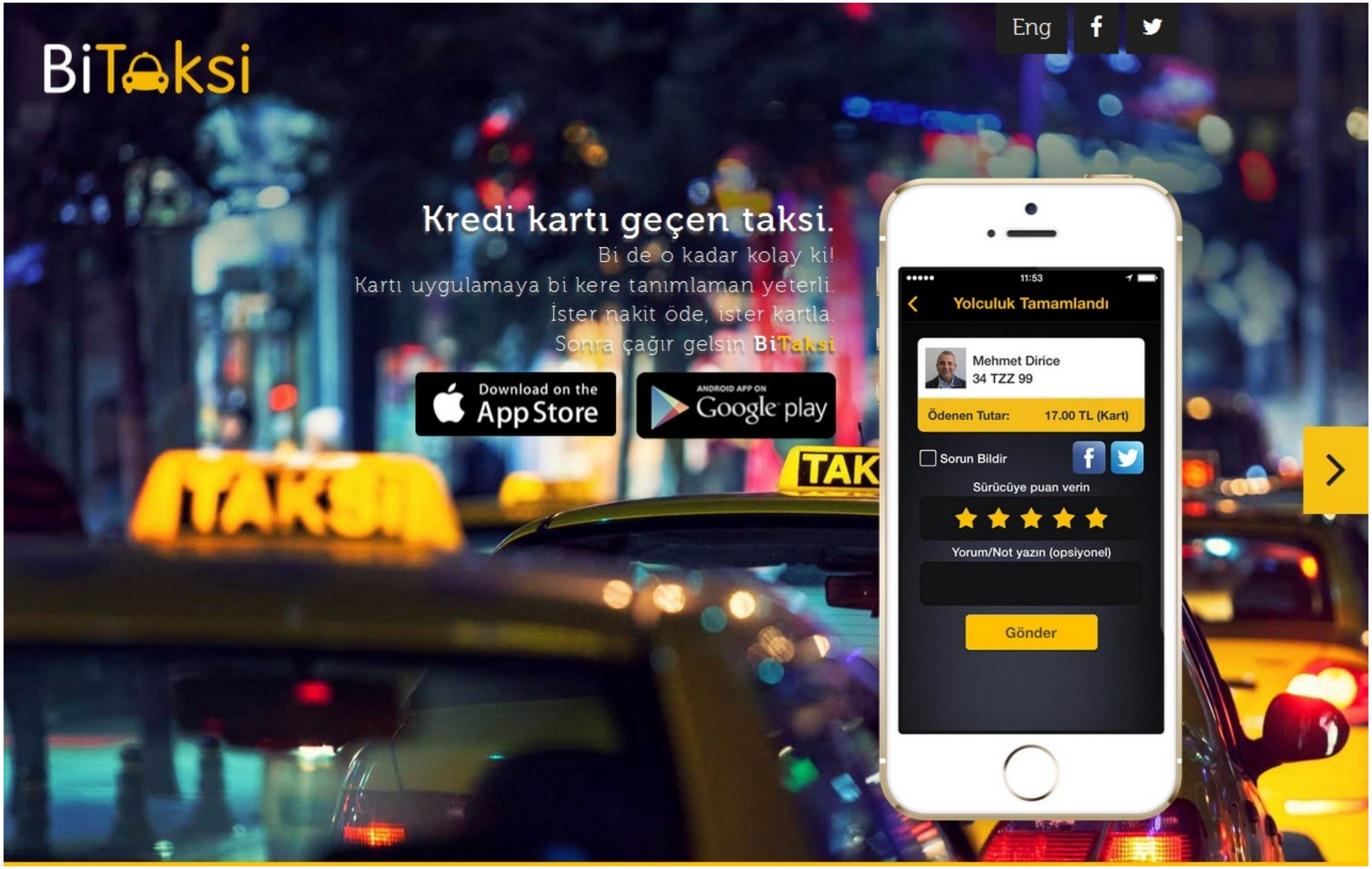
Sector Giant

COST (Maliyet)

(IT Know-How)

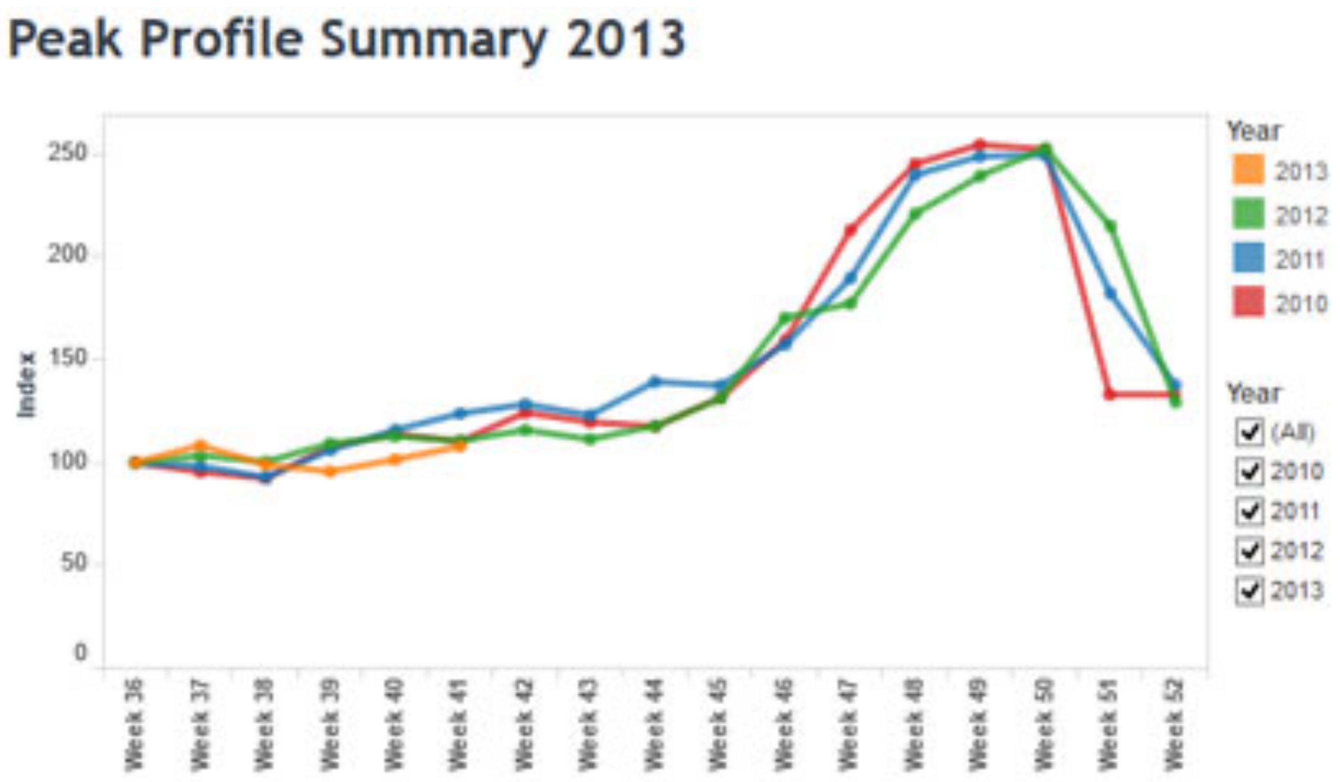
(IT Infrastructure)

Start-up Altyapısı



Talep Dengesizliği

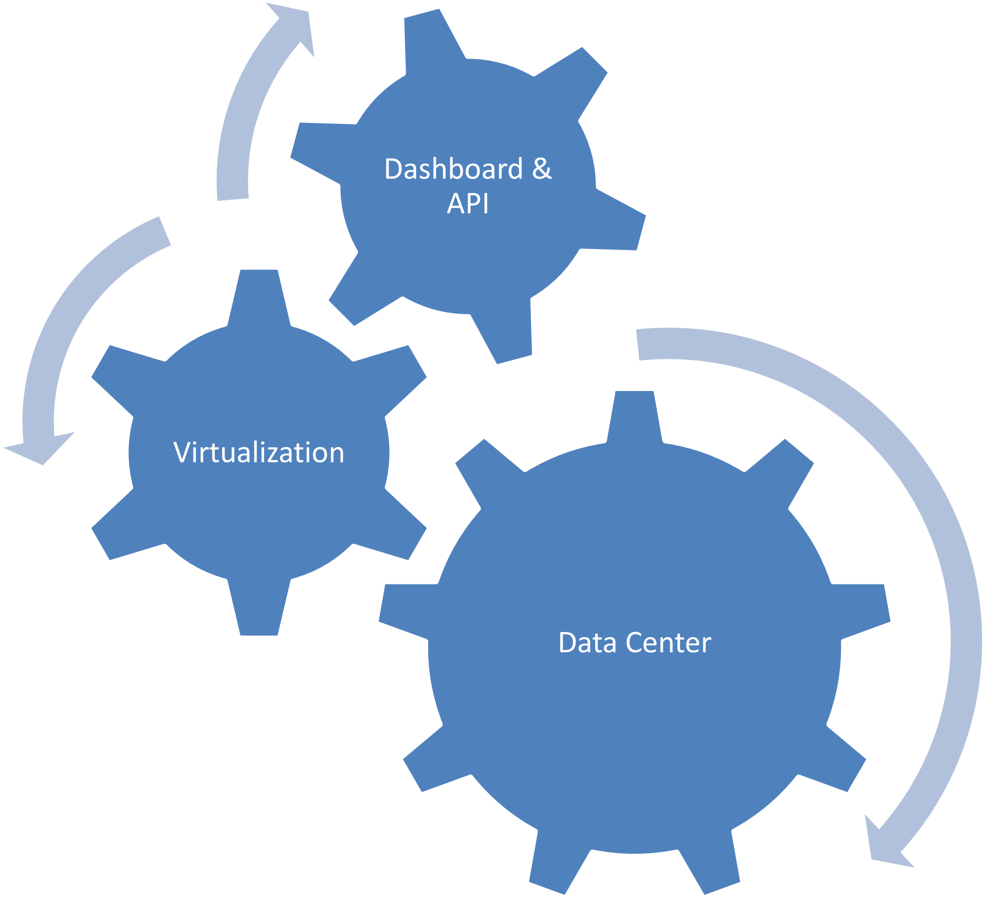
(Demand inconsistency)



Ref: Metapack

“Bulut”un Teknik Bileşenleri

(Technical Components of the Cloud Computing)



Vision Inspection of Bottle Caps in Drink Factories Using Convolutional Neural Networks

learning helps to develop classification approaches more accurately. The effect of classification methods on machine vision algorithms for food processing applications has been explored in [12]. Also in [13], the authors proposed a deep learning framework with application in the food and drink packaging industry. The suggested model is based on CNN and classification to perform quality control vision inspection in order to check that the right package is used for the right product and that package labeling is visually correct.

In this work, we present a novel approach based on deep learning classifiers to recognize *good* and *not good* bottles on the product line. To tackle this problem, we define three different classes named: *Normal cap*, *Unfixed cap* and *No cap*. The last two classes represent not good products. These classes are depicted in Figure 1.

The rest of our paper is organized as follows. Section II deals with the methodology of the proposed approach. The experiments and results are reported in Section III. Finally, in Section IV, we conclude the paper.

II. Methodology

Classification found on CNN and deep learning methods have led to strong approaches [14]. Several pre-trained classification networks have been proposed and evaluated as feature extractors for object detection. These networks include several types such as VGG-16 and VGG-19 networks [15], the ResNet50 network [16], Mobilenet [17], and Xception [18].

The VGG network was introduced by Simonyan and Zisserman in [15]. VGG is a well-known model; the authors evaluated very deep convolutional networks consisting of up to 19 weight layers for classification of large images. This network uses multiple 33 convolutional layers one after another in increasing depth. The 16 and 19 represents the number of weight layers in the network [15]. Our work employs this model given the good results it can produce and its simplicity.

A. Dataset

A major assumption in many machine learning algorithms is that the training and test data must be in the same feature space and have the same distribution. However, in many real-world applications, the training of deep convolutional neural networks is a difficult procedure because it requires a vast amount of labeled data for training, it is time-consuming and requires a great deal of expertise [19]. In such cases, successful

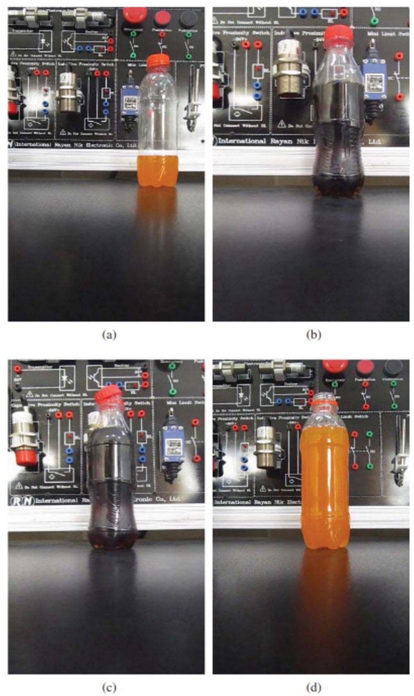


Fig. 1: Some dataset samples: (a, b) Normal Cap, (c) Unfixed cap, and (d) No Cap

B. Transfer Learning and Fine Tuning for VGG-19

Transfer learning is a machine learning technique where an existing model that was developed for a particular task is deployed as the starting point for a model on a new challenge

Fig. 2: Our proposed model for bottle cap inspection: an End-to-End deep learning model

copies the weights of a pre-trained network into the new network [19]. The authors in [21] investigated how fine-tuning a pre-trained CNN improves the classification performance with less training time.

In this work, achieving optimized values for hyperparameters such as epochs, learning rate, L1 and L2 (regularization parameters), and also considering the best optimizer method is targeted to get the highest amount of accuracy for the validation set. In addition, the last two layers in the original VGG-19 are replaced by two fully connected layers in our implementation. The last layer called the output layer should be matched with the number of classes in the dataset, which in our case is three. However, the number of neurons in the layer (N) just before is considered as an adjustable parameter in our deep learning architecture. Figure 2 depicts our proposed model for visual inspection of bottle caps based on an end-to-end deep learning structure.

III. Experiments and Results

In our study, the overall implementations and simulations of the VGG-19 pre-trained network were done using the Python programming language. We also used the Tensorflow library to show the performance of the proposed model. Tensorflow is an open-source software developed by Google, which provides a deep learning framework to model machine learning problems and applications [22].

such as L1 and L2 regularizations. According to Equations 1 to 6, both L1 and L2 regularization methods add a penalty to the cost function (CF) based on the model complexity. In fact, instead of calculating the CF by simply using a loss function (LF), it is much more effective to have an additional element called a regularization term (Equations 2 and 3) that is added in order to penalize complex models.

$$LF = \sum_{i=0}^{N} (y_i - \sum_{j=0}^{M} x_{ij} \times W_j)^2 \tag{1}$$

$$L1\_Regularization = L1 \sum_{j=0}^{M} |W_j| \tag{2}$$

$$L2\_Regularization = L2 \sum_{j=0}^{M} W_j^2 \tag{3}$$

$$CF(L1) = LF + L1\_Regularization \tag{4}$$

$$CF(L2) = LF + L2\_Regularization \tag{5}$$

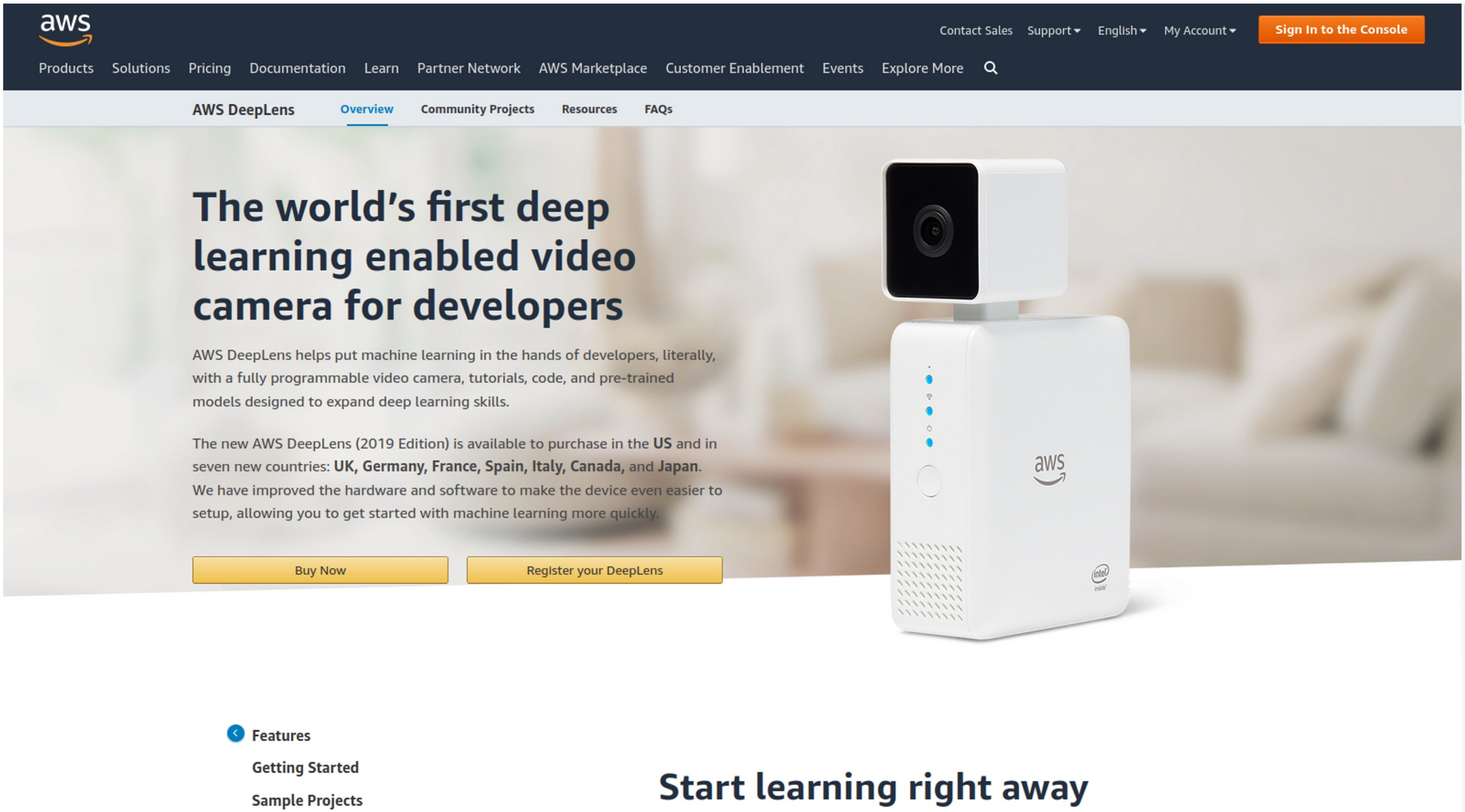
$$CF(L1, L2) = CF(L1) + L2\_Regularization \tag{6}$$

These hyperparameters are capable to update the general cost function. In Figure 3, the results of applying various amounts of L1 and L2 (L1=L2) from $L1 = 10$ to $L1 = 0$ are illustrated. The results indicate that for values both greater and smaller than 0.05 we have underfitting and overfitting, respectively. As a result, $L1 = L2 = 0.05$ is the best point for these hyperparameters.

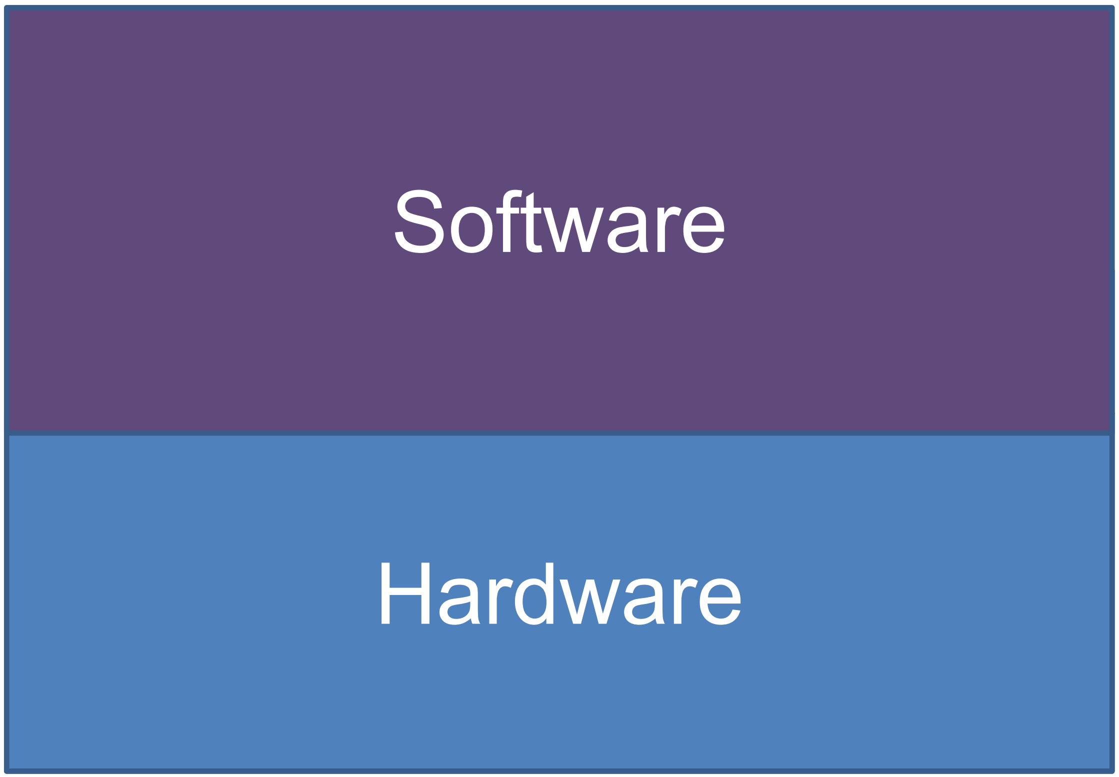
Ücretlendirme !

https://aws.amazon.com/tr/sagemaker/pricing/

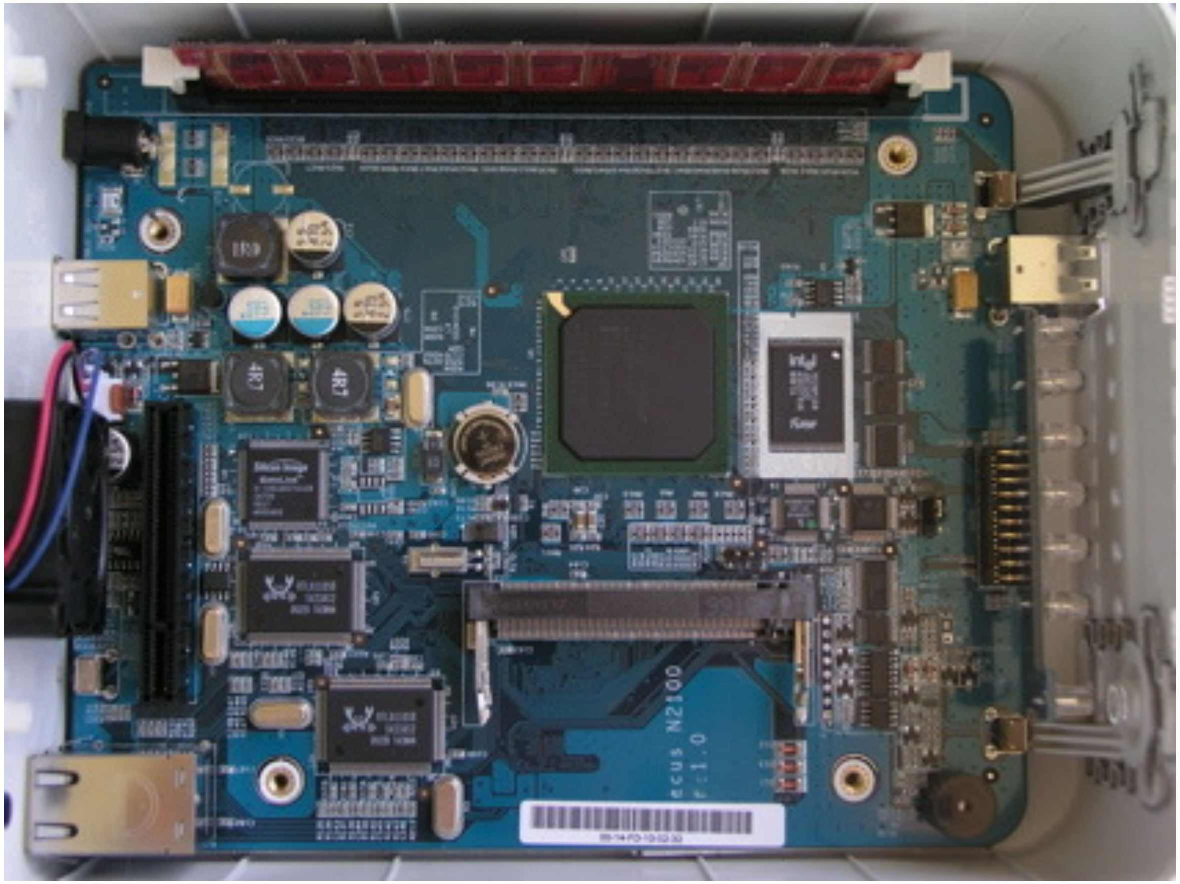
DeepLens



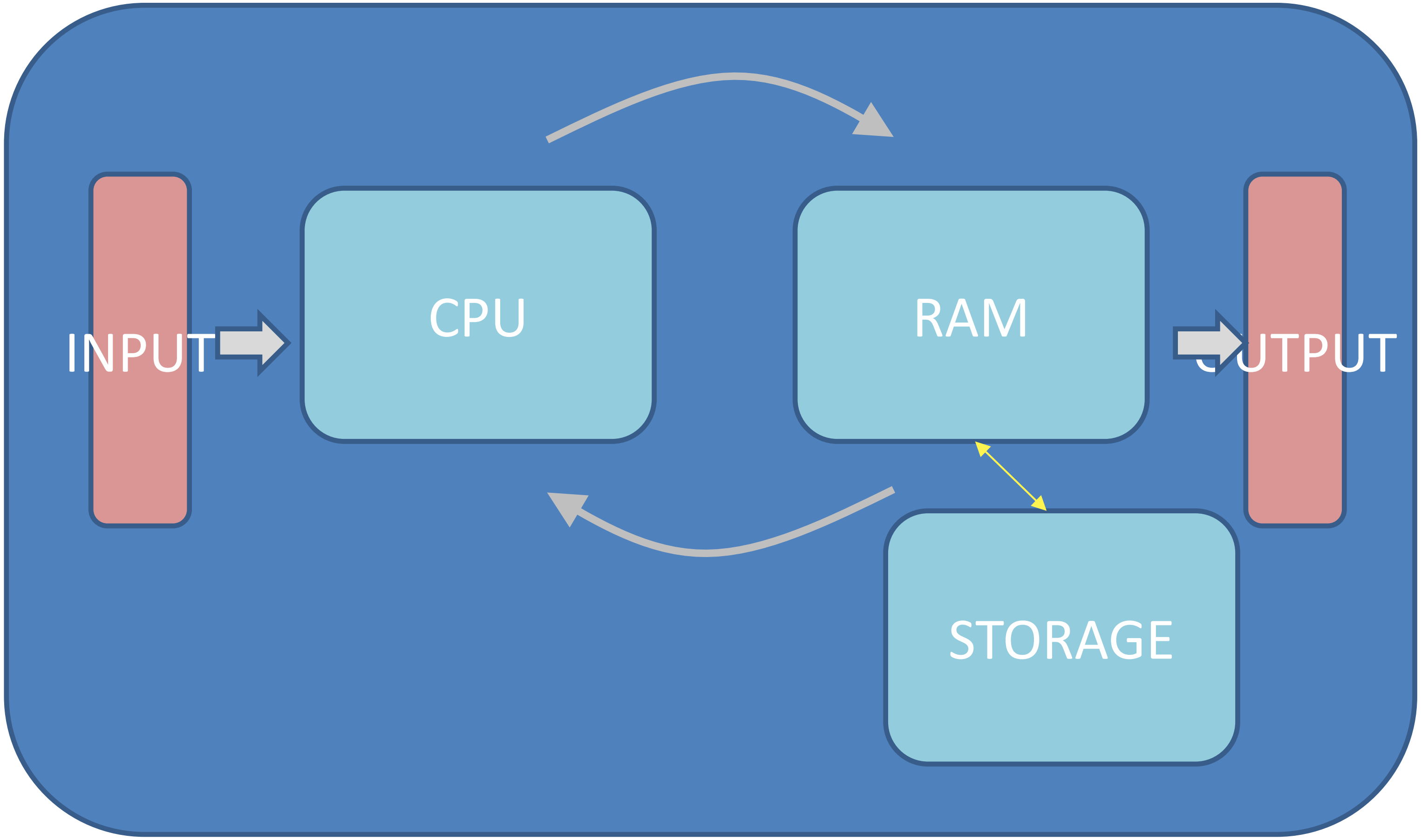
aws
Contact Sales
Support
English
My Account
Sign In to the Console
Products
Solutions
Pricing
Documentation
Learn
Partner Network
AWS Marketplace
Customer Enablement
Events
Explore More
AWS DeepLens
Overview
Community Projects
Resources
FAQs
The world's first deep learning enabled video camera for developers
AWS DeepLens helps put machine learning in the hands of developers, literally, with a fully programmable video camera, tutorials, code, and pre-trained models designed to expand deep learning skills.
The new AWS DeepLens (2019 Edition) is available to purchase in the US and in seven new countries: UK, Germany, France, Spain, Italy, Canada, and Japan. We have improved the hardware and software to make the device even easier to setup, allowing you to get started with machine learning more quickly.
Buy Now
Register your DeepLens
aws
Features
Getting Started
Sample Projects
Start learning right away

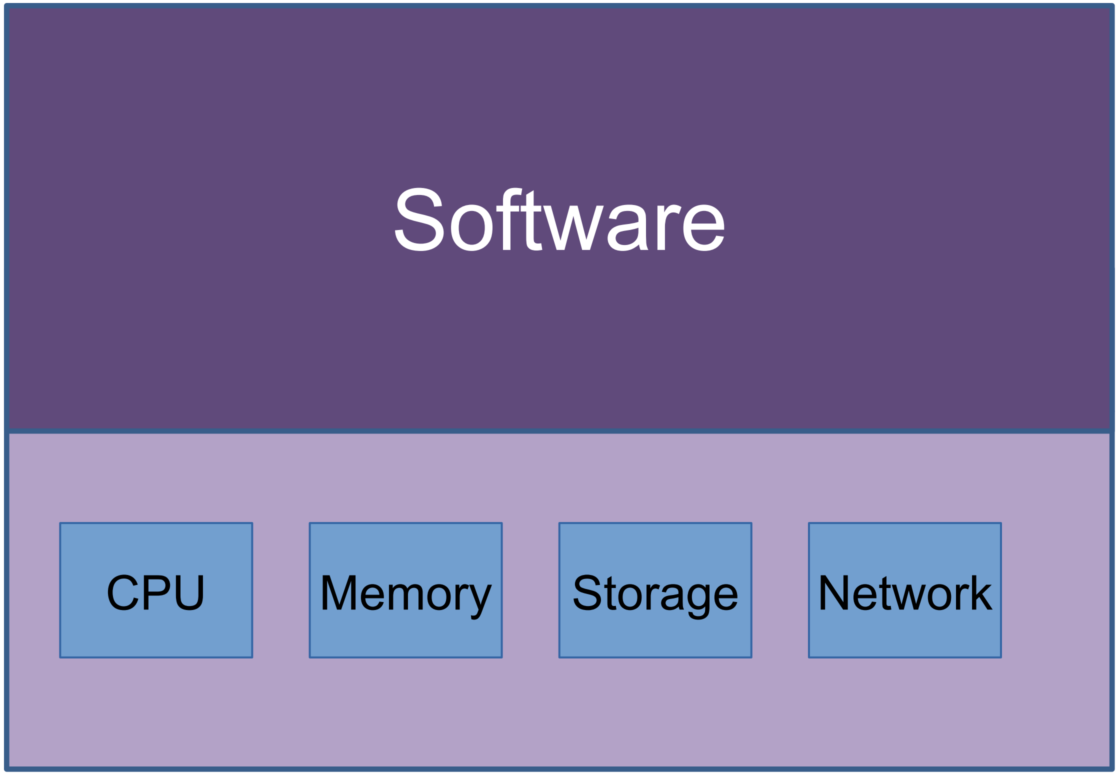

Software
Hardware

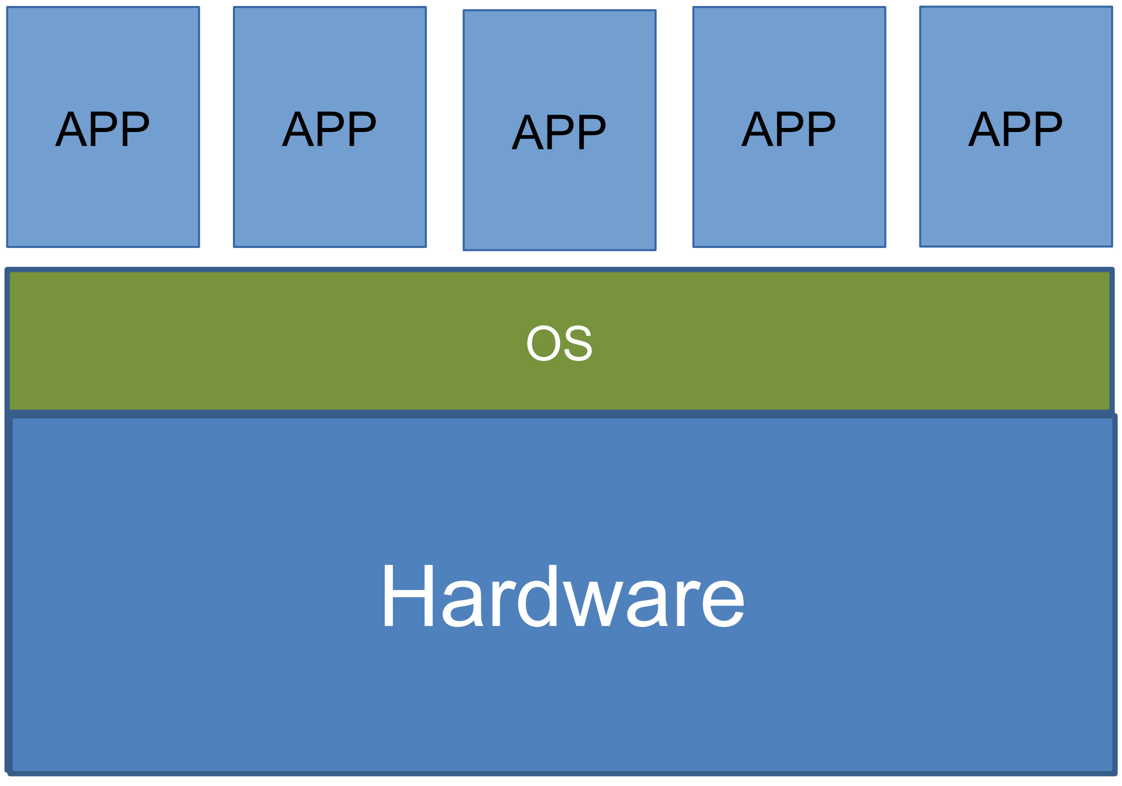
Hardware

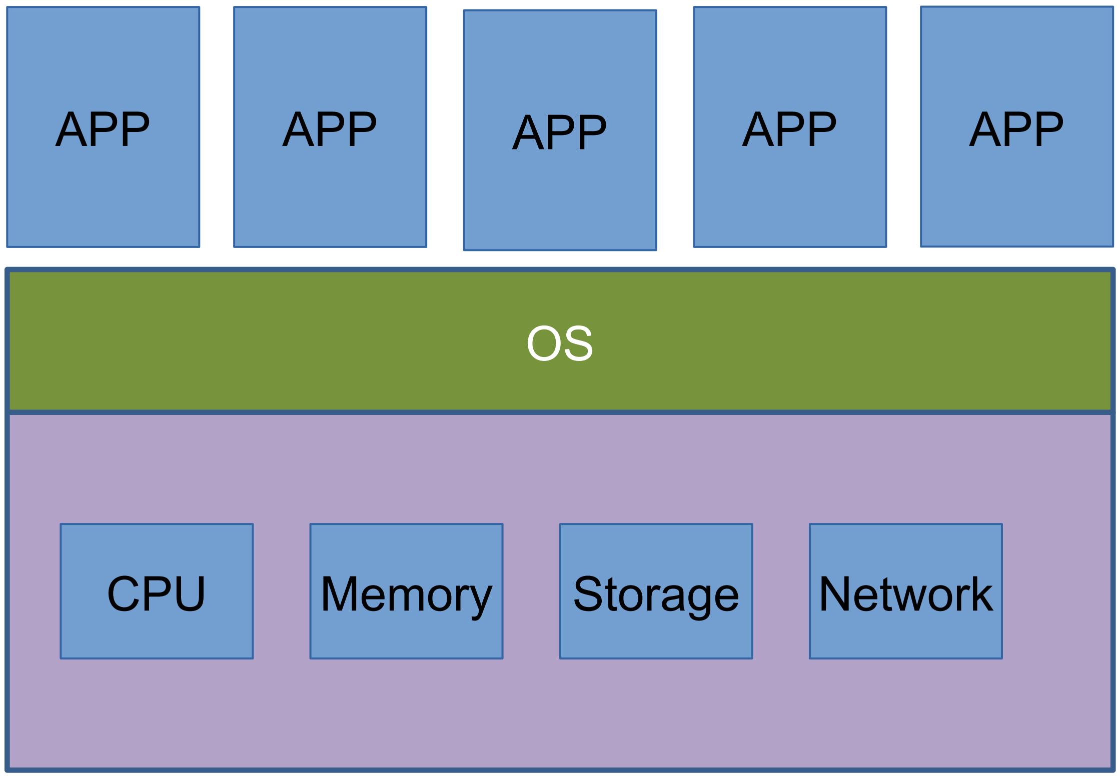


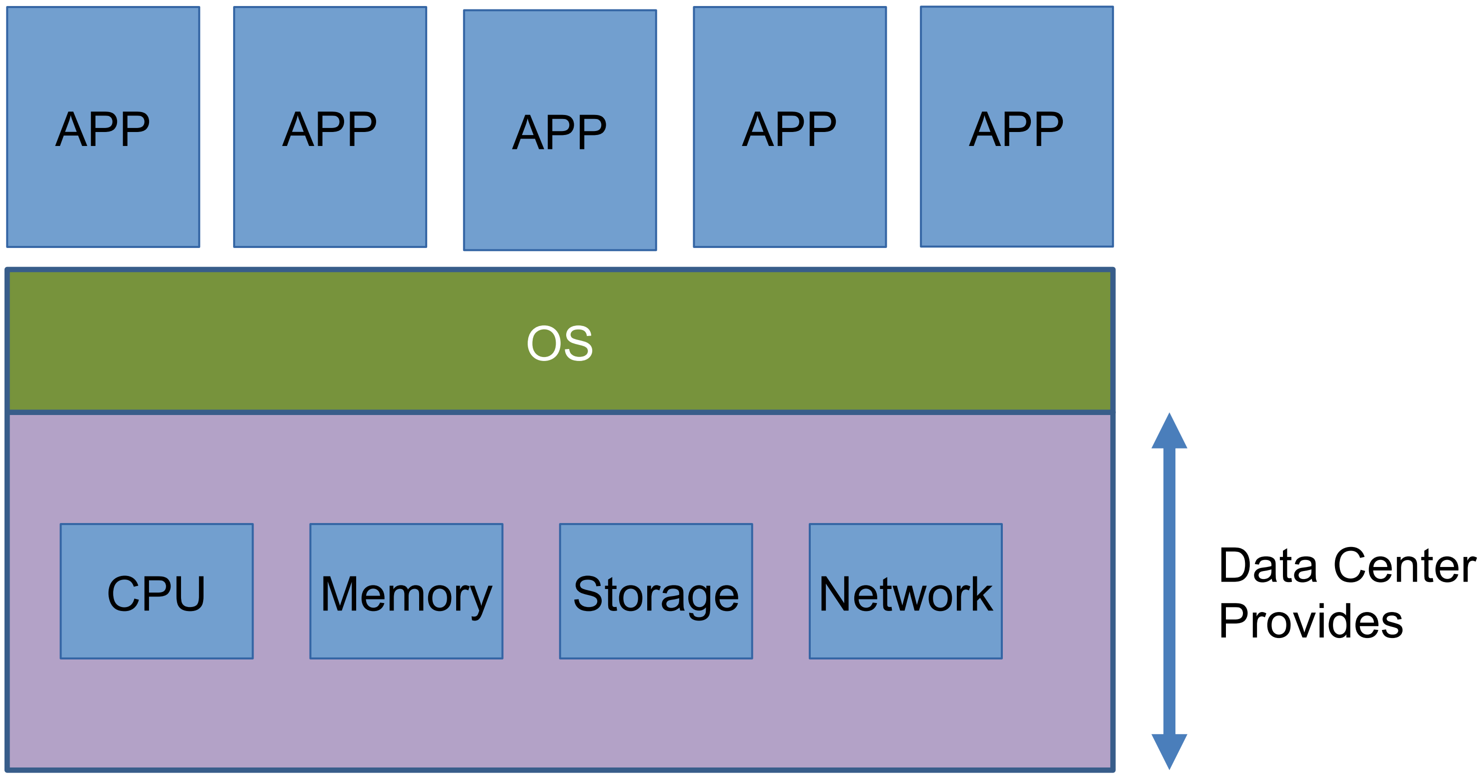
Computation Hardware



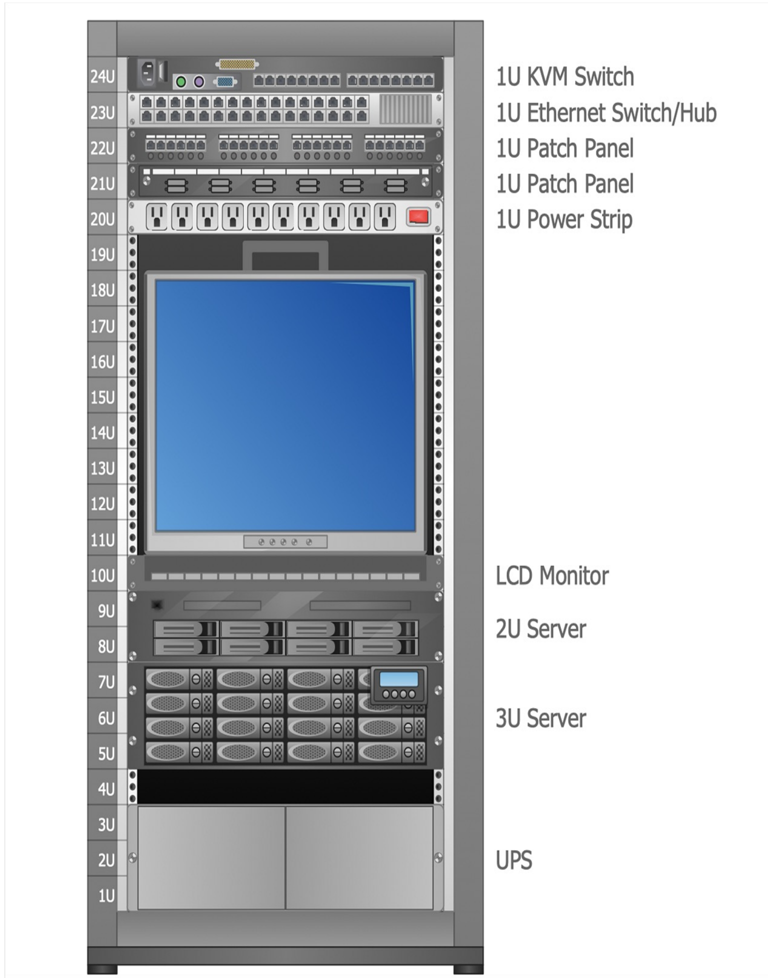

Software
CPU
Memory
Storage
Network


APP
APP
APP
APP
APP
OS
Hardware


APP
APP
APP
APP
APP
OS
CPU
Memory
Storage
Network

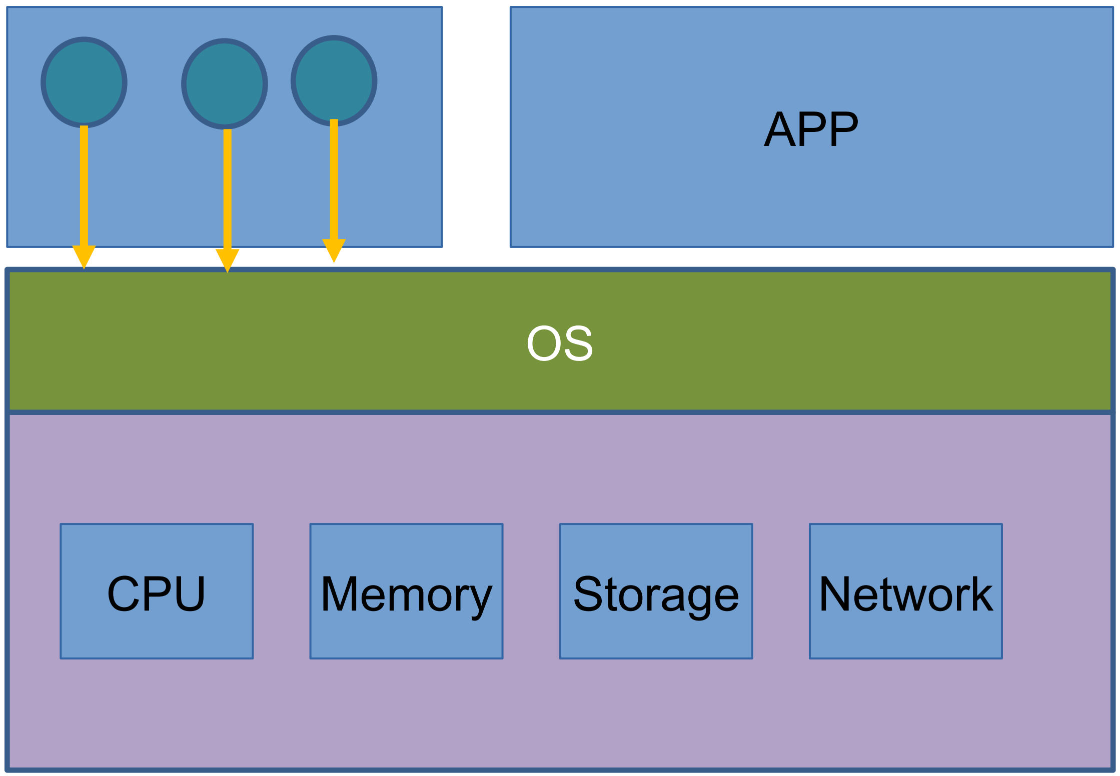

APP
APP
APP
APP
APP
OS
CPU
Memory
Storage
Network
Data Center
Provides

Rack Server

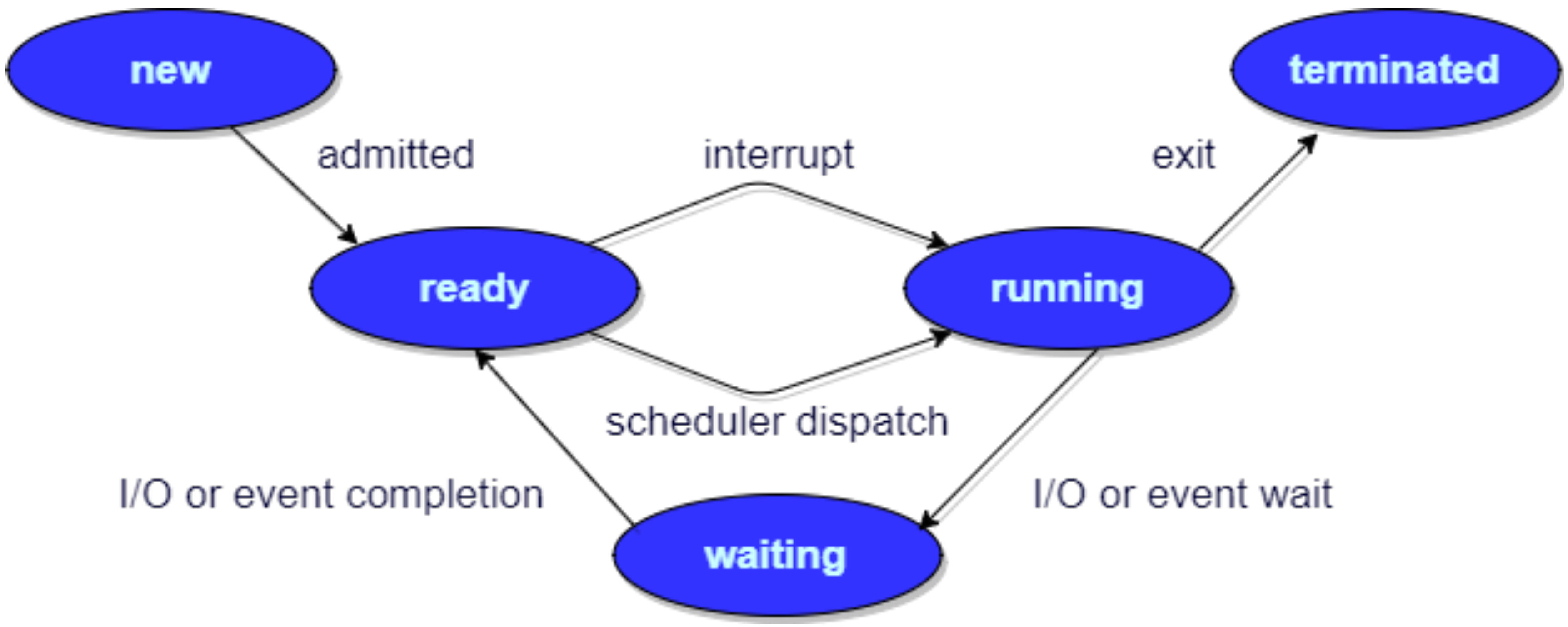


24U
23U
22U
21U
20U
19U
18U
17U
16U
15U
14U
13U
12U
11U
10U
9U
8U
7U
6U
5U
4U
3U
2U
1U
1U KVM Switch
1U Ethernet Switch/Hub
1U Patch Panel
1U Patch Panel
1U Power Strip
LCD Monitor
2U Server
3U Server
UPS

Software runs as a process on OS !

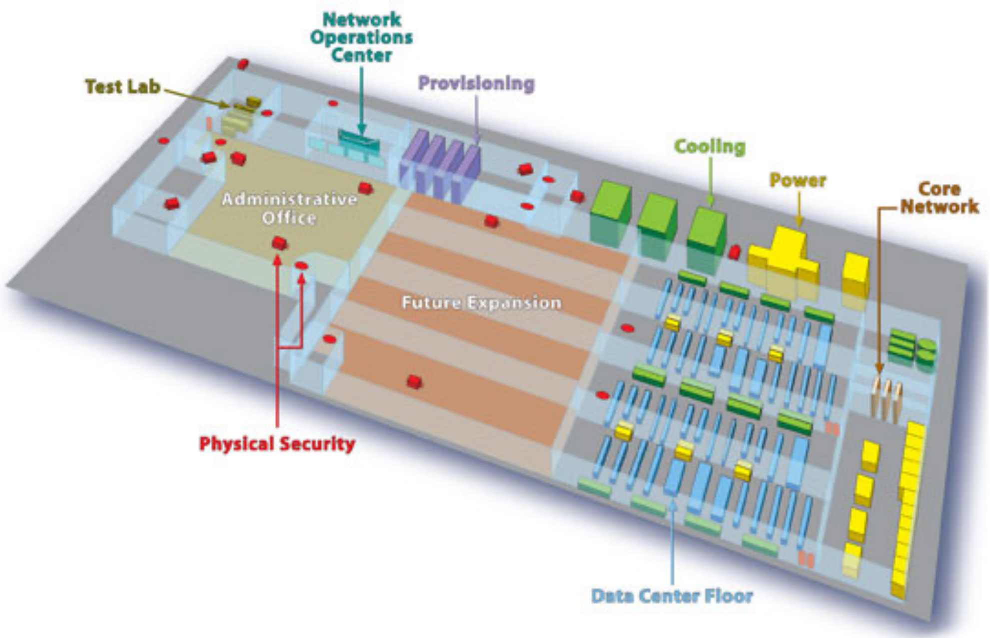


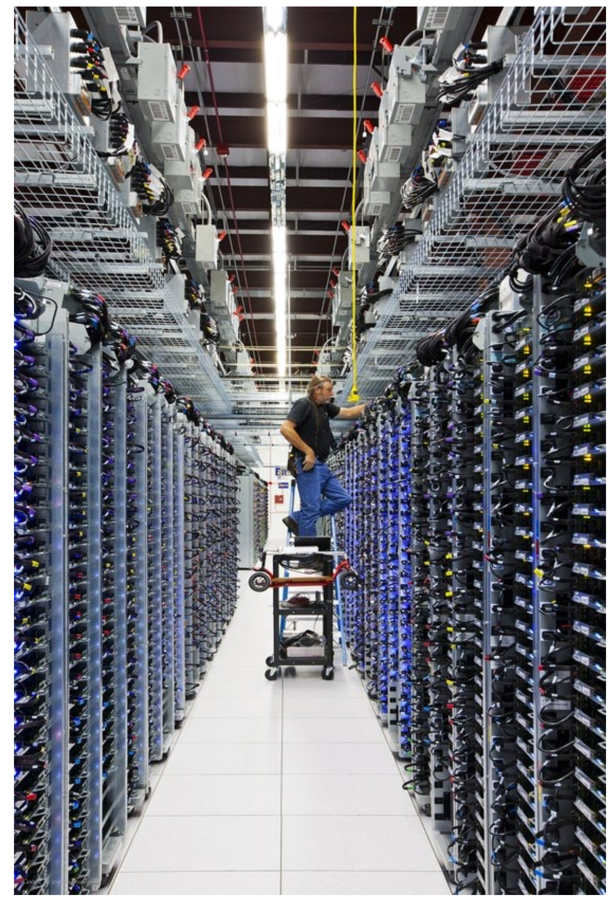
Process Lifecycle



Veri Merkezi

(Data Center)





Ref: tistime.com

Veri Merkezi

(Data Center)

- Resembles huge plants, needs good planning
- CapEx: Realestate, building, computational infrastructure, Energy infrastructure, strict A/C.
- OpEx: Energy consumption, staff, internet access, maintanenence.
- Big data centers can contain 50.000 to 100.000 server H/W.
- Data center construction cost: 10 Milyon USD / MegaWatt. (slightly old figure)

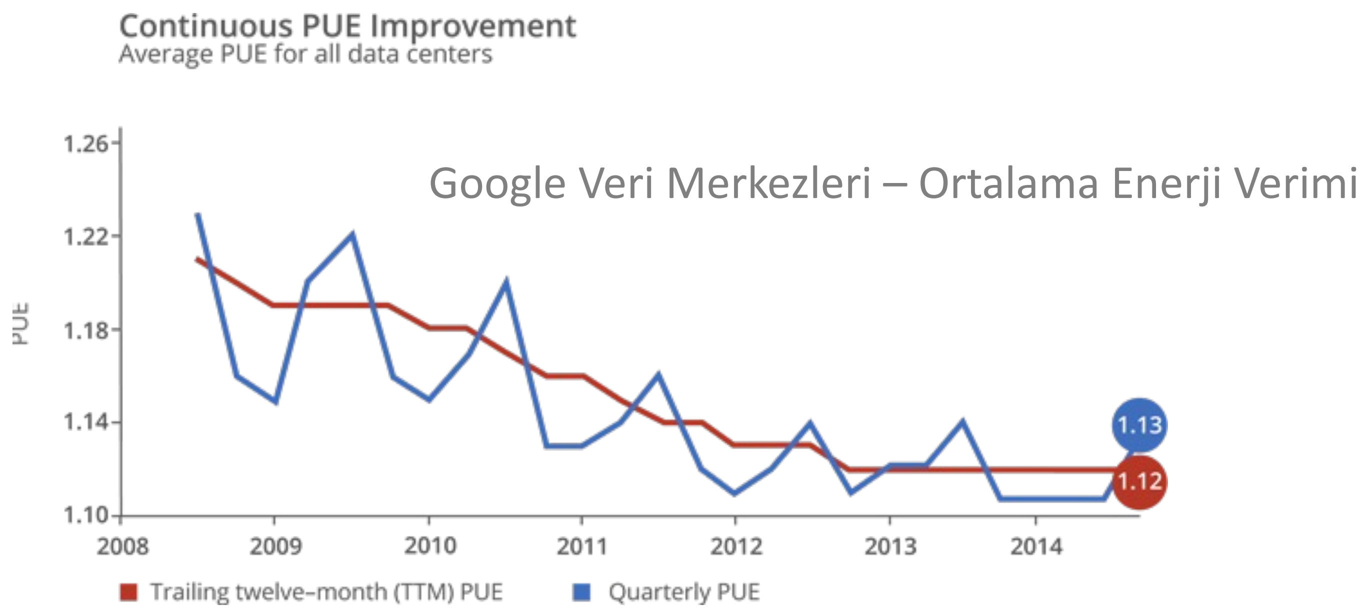
100.000 Sunuculuk Veri Merkezi

(Data Center with 100.000 Servers)

- Raw calculation: 250 watt / server (200Watt + 50Watt common usage)
- 100.000 server = 25 MegaWatt = 250 Million USD (Just the plant)
- 100.000 server x 1000USD = 100 Milyon USD
- Total CapEx: 350 Million USD
- OpEx = *f(Regional electricity rates, cooling costs)*

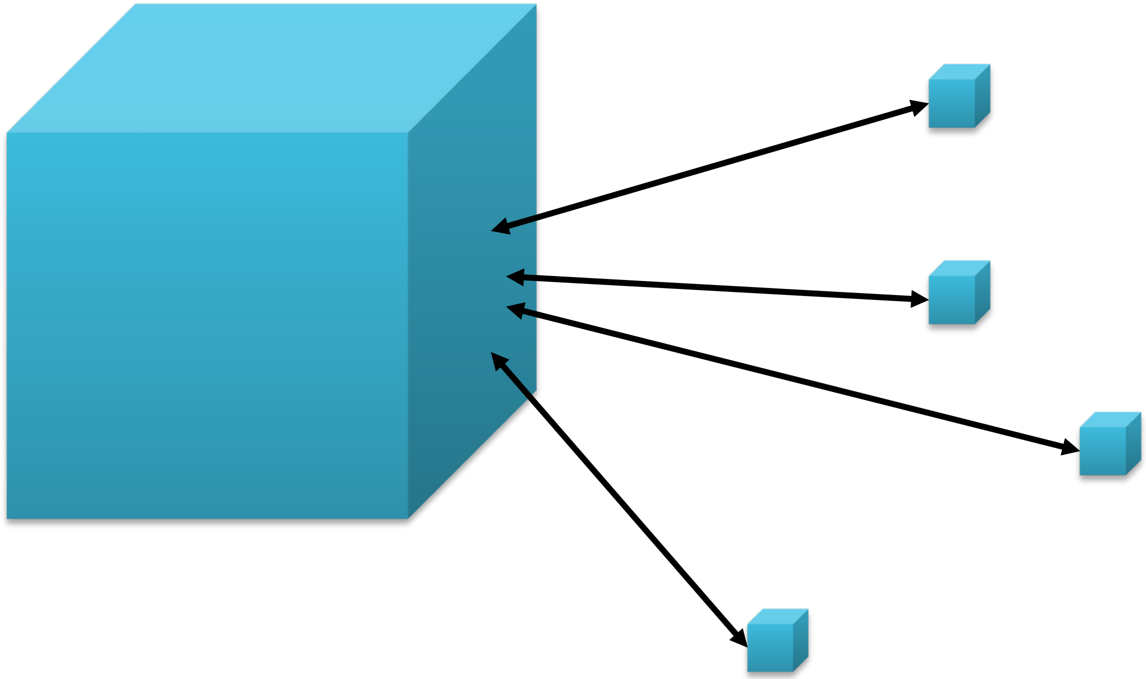
Enerji Verimi

(Power Usage Efficiency)



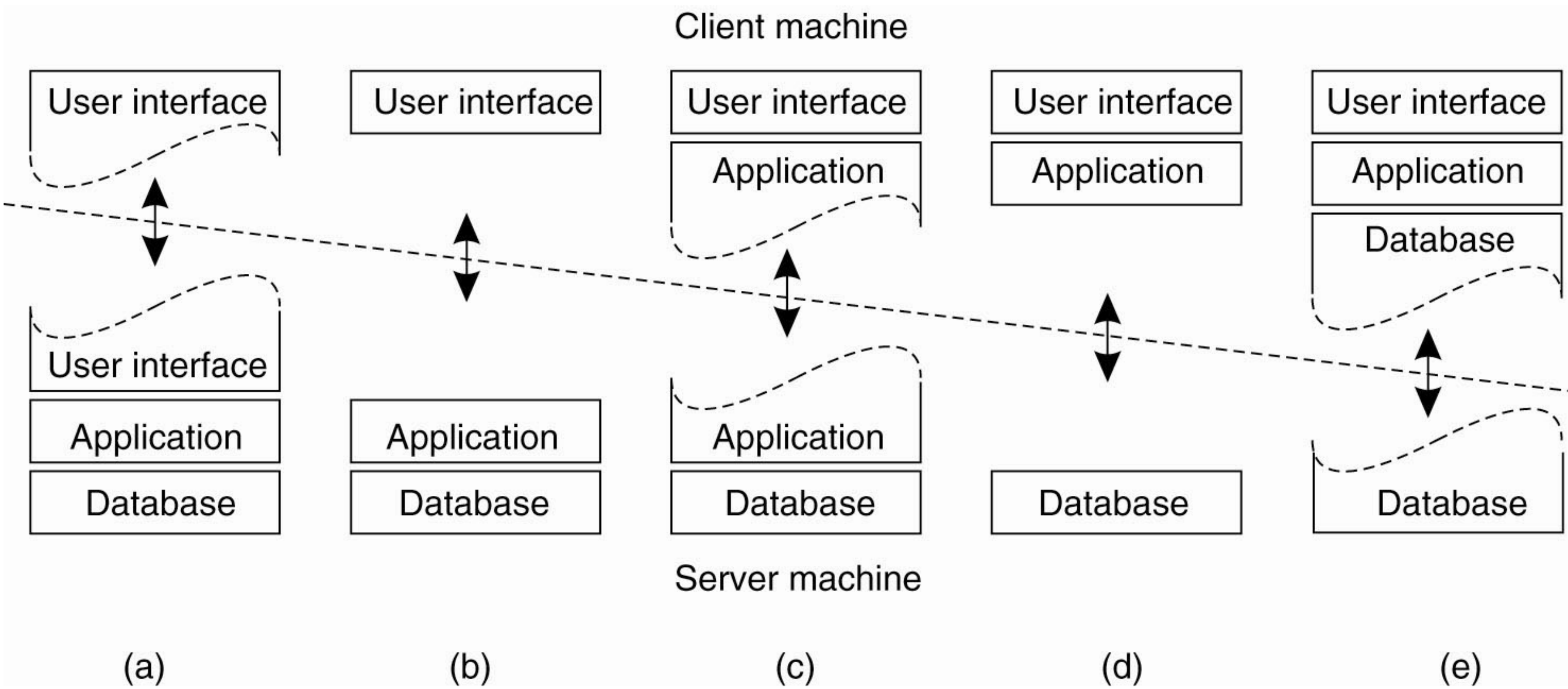
$$PUE = \frac{Total\ Facility\ Energy}{IT\ Equipment\ Energy}$$

Client-Server Architecture

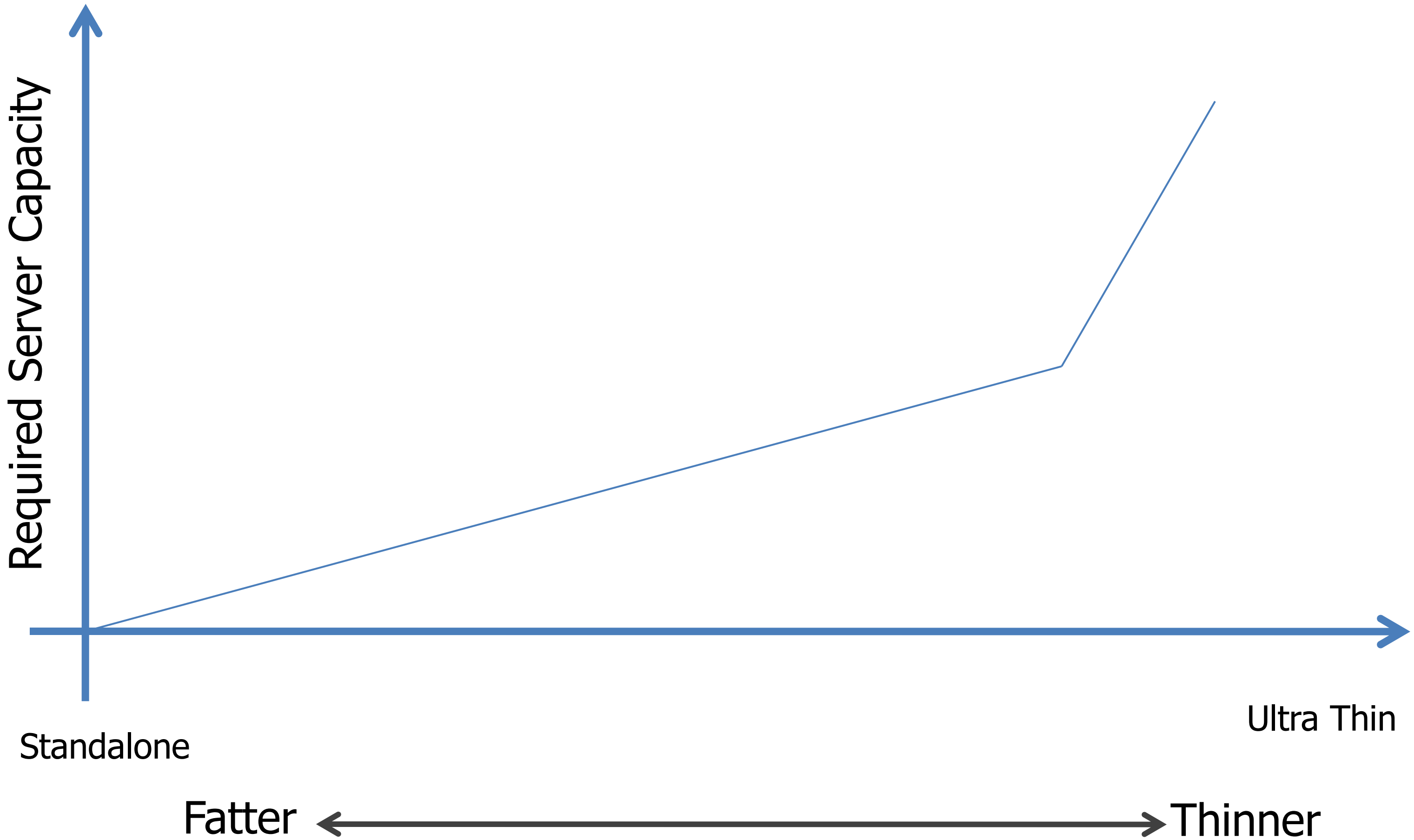


Computation Load is at the center (Thin client use case)

Functionality Assignment in C/S Architecture



Thin vs. Thick Clients



Diskless Mode

DELL Wyse Z Class Thin Client

Windows Embdedded/Modified Linux , 1.5 GHz single core AMD, 2GB RAM, 4GB Flash

Dimensions: 200 mm x 47 mm x 225 mm
Weight: 1.1. kg