

# Cloud Computing



**There is no cloud**  
it's just someone else's computer

| The entire history of software engineering is that of the rise in levels of abstraction.

-- Grady Booch

Cloud computing is 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 be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models.

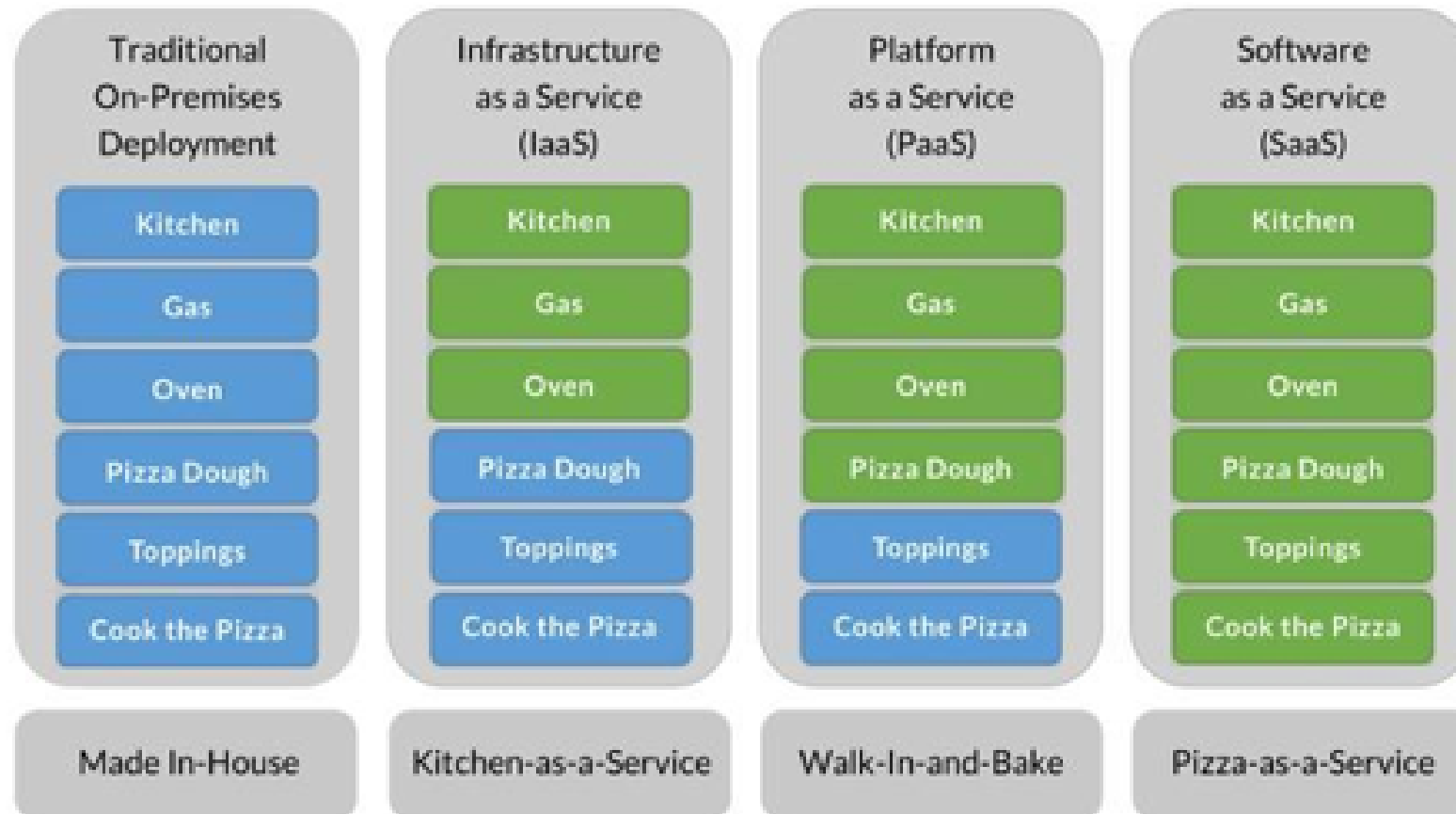
<https://csrc.nist.gov/pubs/sp/800/145/final>

## Essential Characteristics:

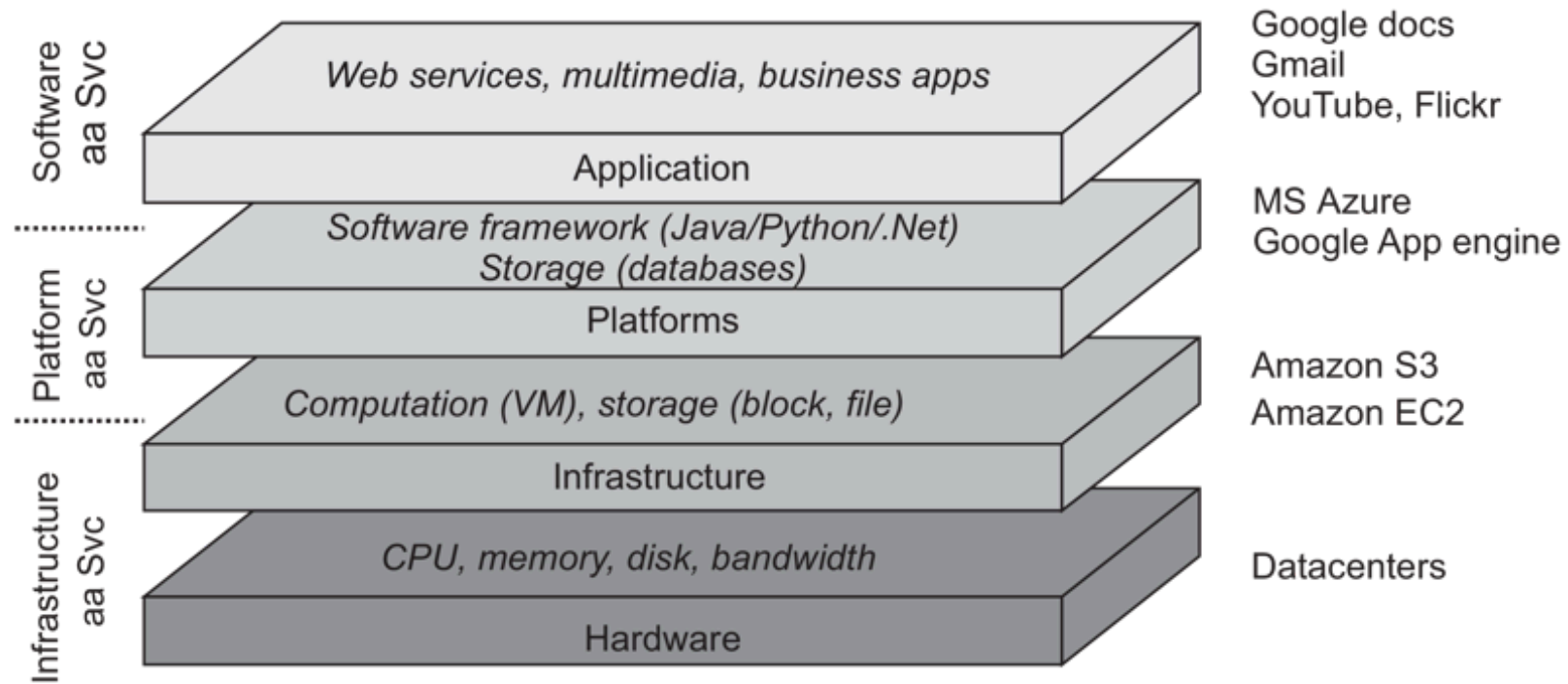
- On-demand self-service
- Broad network access
- Resource pooling
- Rapid elasticity
- Measured service

# Service Models

## New Pizza as a Service

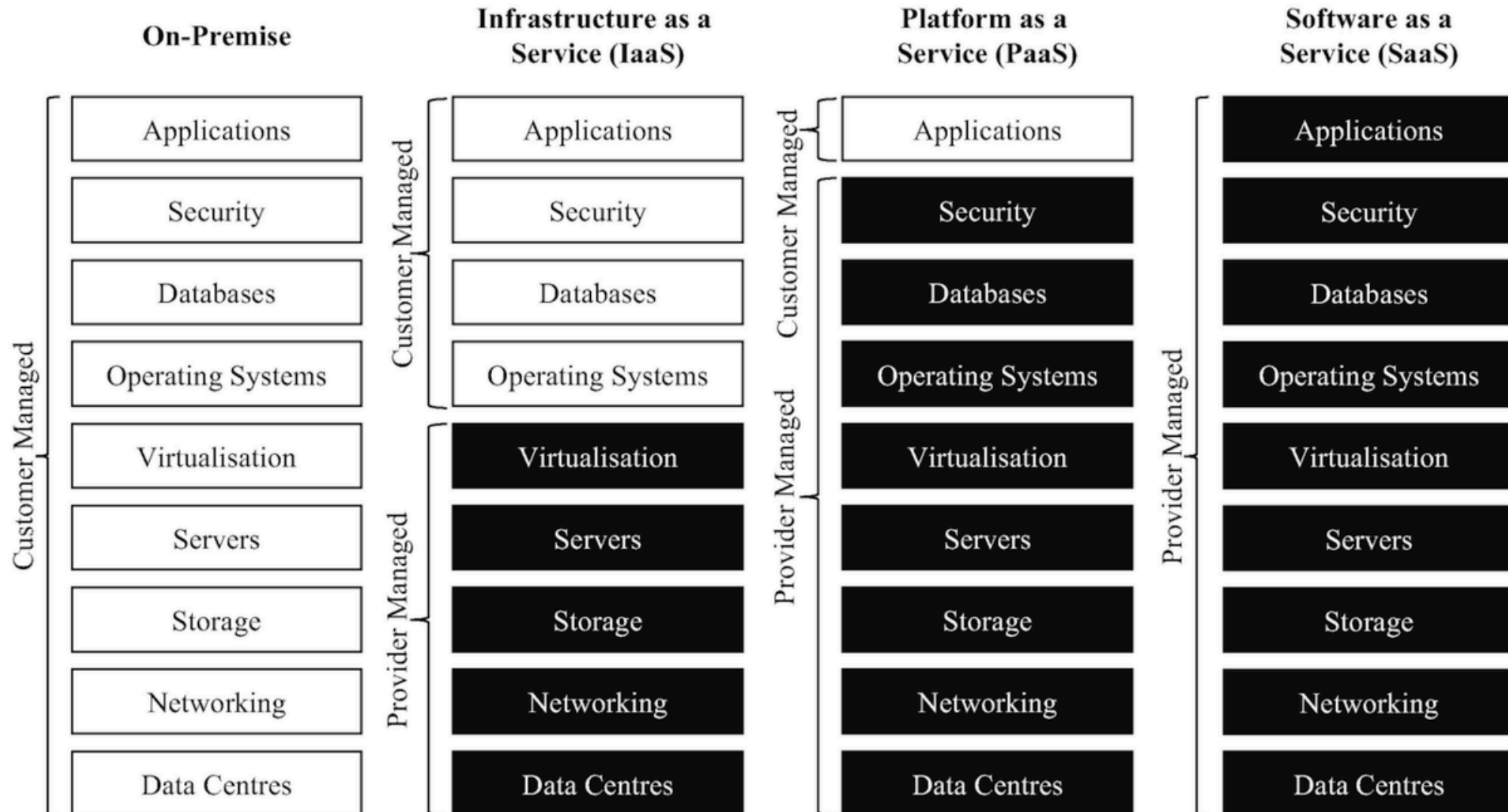


- Software as a Service (SaaS)
- Platform as a Service (PaaS)
- Infrastructure as a Service (IaaS)



(VanSteen, 2017, S. 30)

# XaaS



Low Cost

High Cost

FARM  
BUILD

WHOLESALE  
CO-LOCATION

GROCERY  
HETZNER

RESTAURANT  
AWS

DOORDASH  
VERCEL

<https://news.ycombinator.com/item?id=45614922#45616049>

# Deployment Models

- Private cloud
- Community cloud
- Public cloud
- Hybrid cloud



# Cloud Native Applications

- A "cloud native" application has adapted and evolved to be maximally efficient in its environment: the cloud.
- In the cloud, an application becomes **distributed**.
- It is forced to be **resilient to** hardware/network unpredictability and **unreliability**.

(vgl. <https://www.reactiveprinciples.org/cloud-native/index.html>)

- Ensuring responsiveness and reliability in this environment is difficult.
- The applications we build after embracing this environment better match how the real world actually works.
- This provides **better experiences** for our users, whether humans or software.

(vgl. <https://www.reactiveprinciples.org/cloud-native/index.html>)

The constraints of the cloud environment include:

- All inter-service communication takes place over unreliable **networks**.
- You must operate under the assumption that the underlying **hardware can fail** or be restarted or moved at any time.
- The services need to be able to **detect and manage failure** of their peers—including partial failures.
- Strong consistency and transactions are expensive. Because of the coordination required, it is difficult to make services that manage data available, performant, and scalable.

(vgl. <https://www.reactiveprinciples.org/cloud-native/index.html>)

Therefore, a Cloud Native application is designed to leverage the cloud operating model.

It is predictable, decoupled from the infrastructure, right-sized for capacity, and enables **tight collaboration between development and operations**.

It can be decomposed into loosely-coupled, independently-operating services that are resilient from failures, driven by data, and operate intelligently across geographic regions.

# Chancen

# Risiken und Herausforderungen

**When the cloud leaves the datacenter**



at Amazon Web Services (AWS) in the US-East-1 region. These problems are impacting multiple services that depend on AWS infrastructure. We're n



Snapchat



Reddit



Ring



Amazon Web Services



Chime



Amazon

