

Introduction

Big Data Systems

Dr. Rubi Boim

מהשפעות התקלה ב-AWS: מיטות חכמות התחממו, קפאו ונתקעו בזווית שלא מאפשרת לישון

הגאדג'טים הכי מגניבים למיטה שלכם עוררו (סליחה) סערה השבוע, כשהתקלה של אמזון הובילה להרבה צרות אצל המשתמשים

{ נאדניטים } • עידן בן טובים • 24.10.25



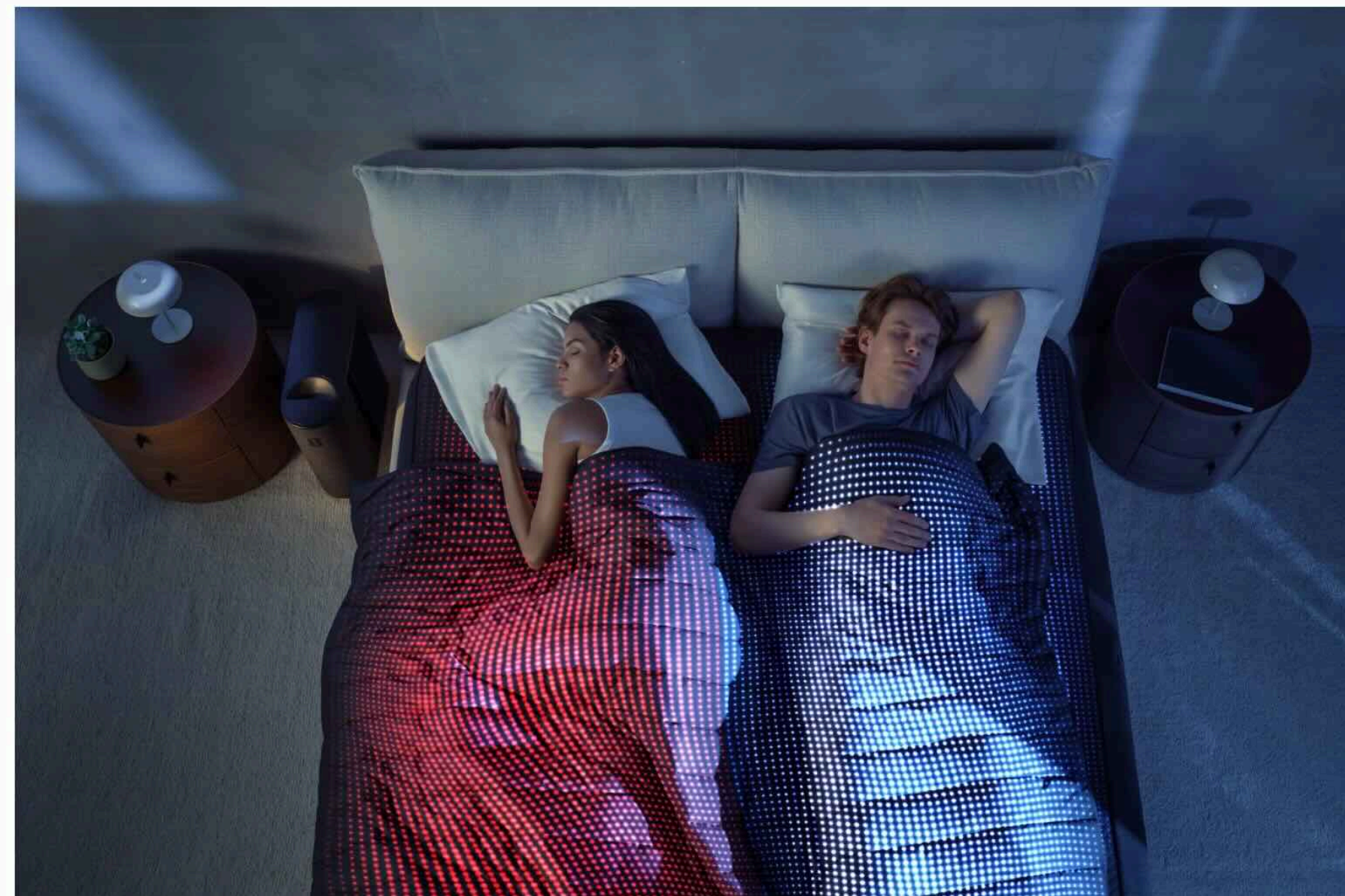
Powered by **Trinity Audio**

למה לקרוא כשאפשר להאזין?

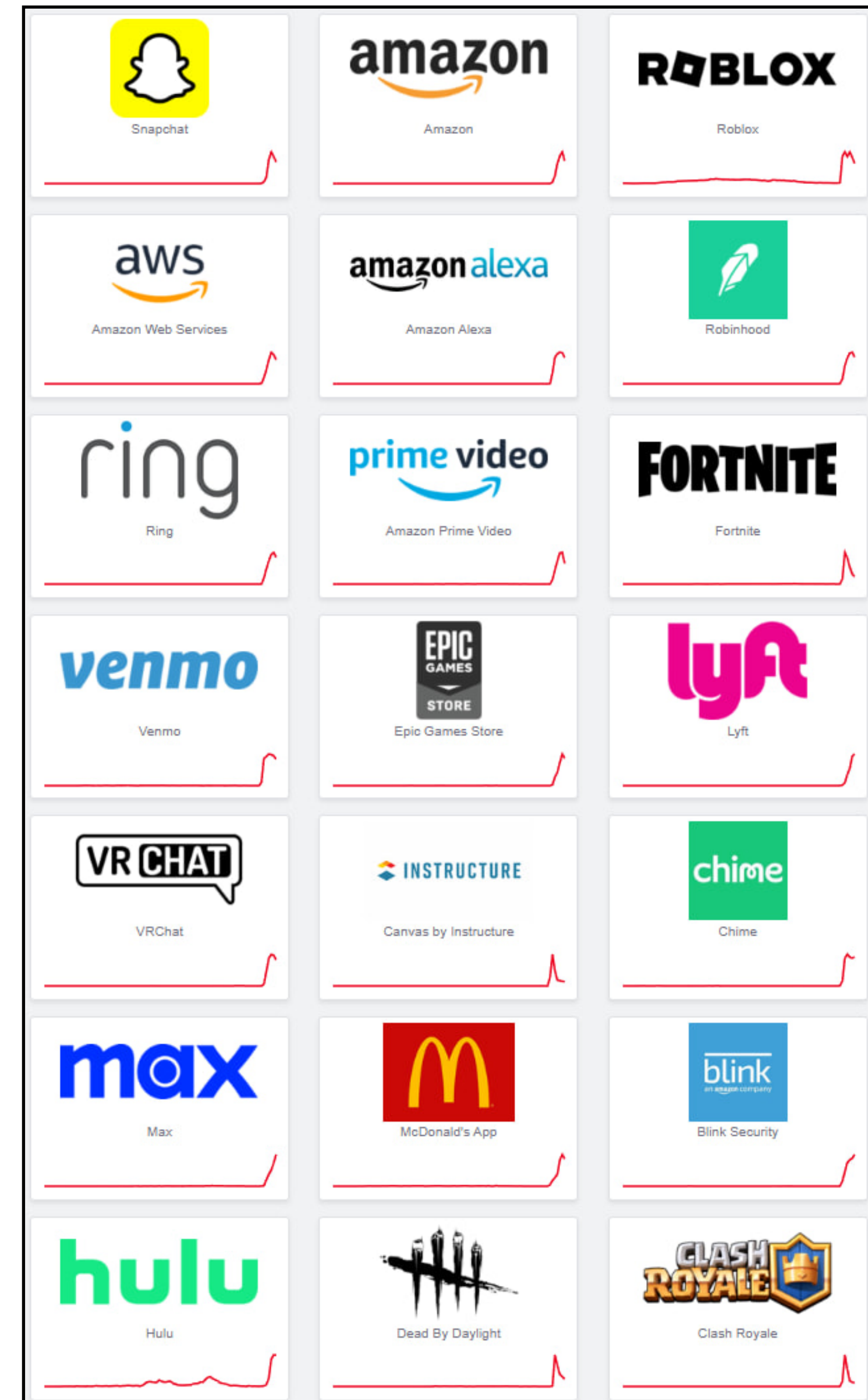
1.0x

00:00

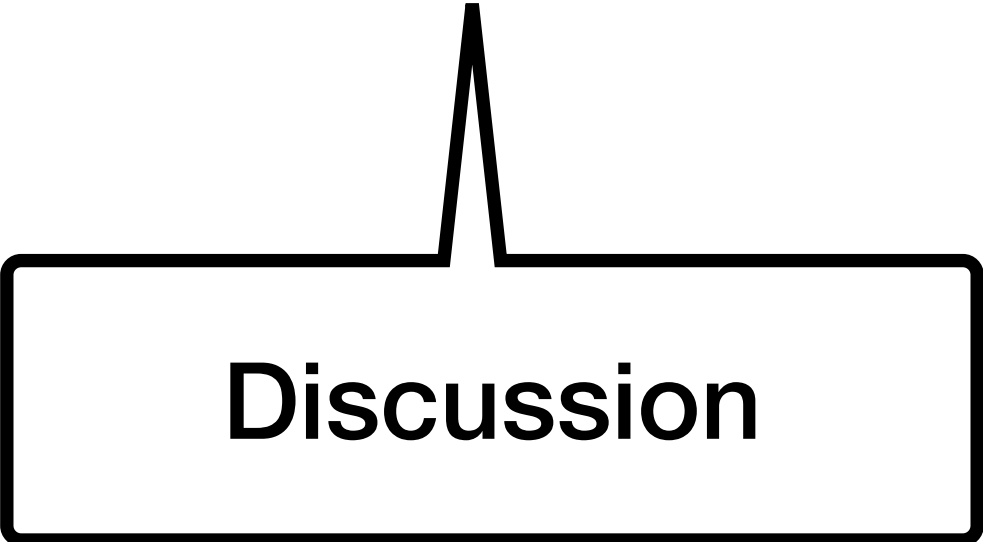
04:25



תמונה: Eight Sleep



When data is Big Data?



Discussion

Big data

ערך שיחה

Big Data ("ביג דאָטָה", לפי החלטת האקדמיה ללשון העברית: **נְתוּנֵי עֵתֶק**^[1]) הוא מונח המתייחס ל**מאגר מידע** הכולל **נתונים מבוזרים**, שאינם מאורגנים לפי שיטה כלשהי, שמגיעים ממקורות רבים, בכמויות גדולות, בפורמטים מגוונים, וב**איכויות** שונות.

Big data

Article [Talk](#)

From Wikipedia, the free encyclopedia

This article is about large collections of data. For the band, see [Big Data](#) and consumer data, see [Surveillance capitalism](#).

Big data primarily refers to [data sets](#) that are too large or complex to be dealt with by traditional [data-processing software](#). Data with many entries (rows) offer greater [statistical power](#), while data with higher complexity (more attributes or columns) may lead to a higher [false discovery rate](#).^[2] Though used sometimes loosely partly due to a lack of formal definition, the best interpretation is that it is a large body of information that cannot be comprehended when used in small amounts only.^[3]

Data Lake

▪ Big data

History

Early Foundations and
Precursors (Pre-2000)

Emergence in the Digital
Age (2000–2010)

Expansion and Mainstream
Adoption (2011–Present)

Definition and Characteristics

Core Definition

The "Vs" Framework

Fact-checked by Grok yesterday

Big data

Big data denotes the extensive assemblages of data arising from networked digital systems, sensors, and human activities, which exceed the processing capacities of conventional tools and demand specialized technologies for effective management and analysis.^[1] These datasets are primarily defined by three core attributes—**volume** (immense scale), **velocity** (rapid generation and flow), and **variety** (diversity of formats, from structured records to unstructured text and multimedia)—often extended to include **veracity** (reliability amid noise) and **value** (potential for meaningful extraction).^[2] Originating in the late 1990s amid advances in computing and storage,

Less than 24 hours form launch

Agenda for today

- **5 V's of Big Data**
- Cloud computing
- Highly available / highly Scalable
- Managed vs Unmanaged services

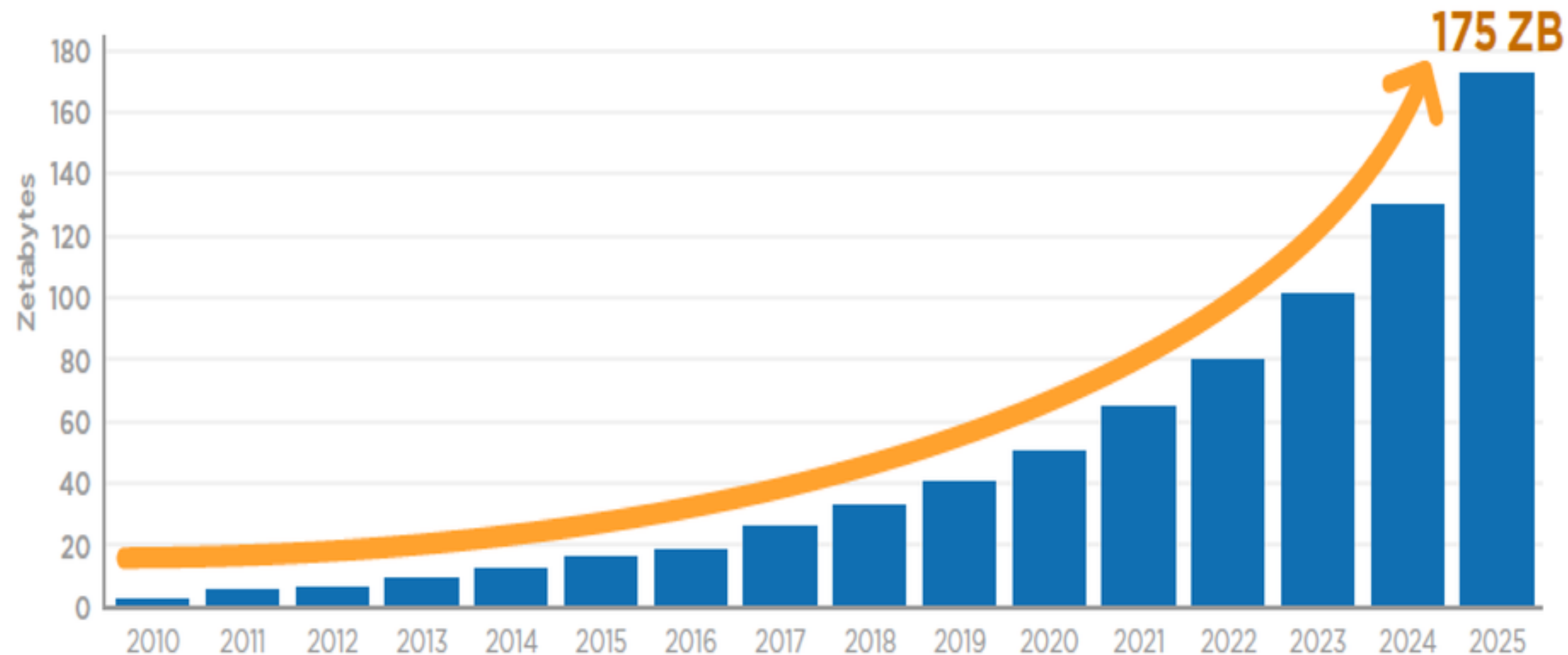
5 V's of Big Data

- Volume
- Velocity
- Variety
- Veracity
- Value



Volume

- Data is rapidly increasing
(due to cloud computing, mobile and more)



Source: Data Age 2025, sponsored by Seagate with data from IDC Global DataSphere, Nov 2018

Value	Metric
1000	kB kilobyte
1000 ²	MB megabyte
1000 ³	GB gigabyte
1000 ⁴	TB terabyte
1000 ⁵	PB petabyte
1000 ⁶	EB exabyte
1000 ⁷	ZB zettabyte
1000 ⁸	YB yottabyte

Volume

- Data is rapidly increasing
(due to cloud computing, mobile and more)

As of 2025, WhatsApp users send over 150 billion messages each day

Volume

- Data is rapidly increasing
(due to cloud computing, mobile and more)

As of 2025, WhatsApp users send over 150 billion messages each day

~1.74 million messages per second

Velocity

The speed at which data is generated

- Frequency of data generation (write)
everything is measured
- Frequency of data processing (read)
real time experience

Variety

- **Structured data**
info, transactions...
- **Semi structured data**
logs, sensor data...
- **Unstructured data**
images, video, audio...

Veracity

The truthfulness or reliability of the data

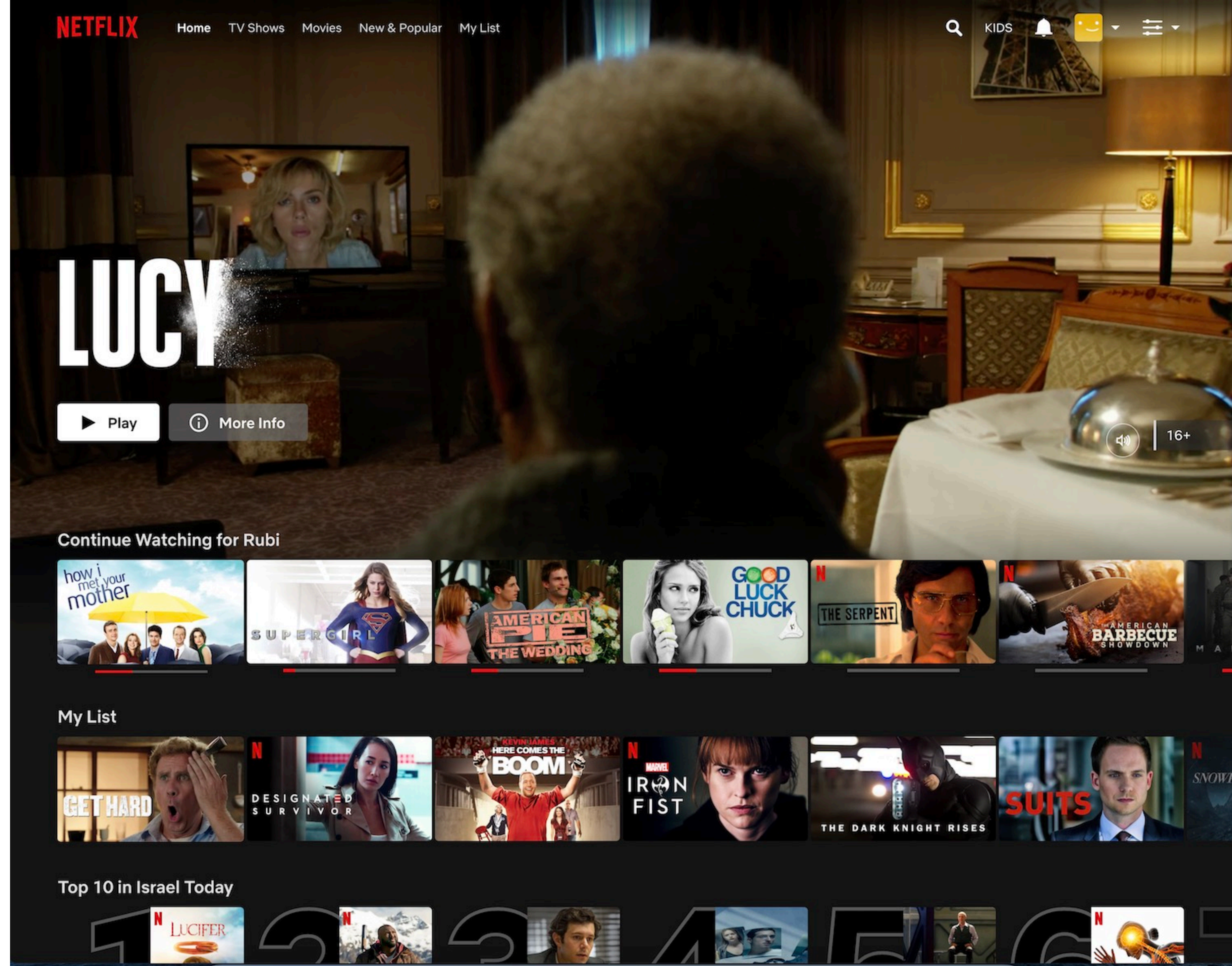
- data quality of captured data can vary greatly
 - bias
 - abnormalities
 - inconsistencies
 - duplication

Value

The final result.

- which questions were answered
- hidden insights (machine learning)
- collecting data without use is, well, useless

- Volume
- Velocity
- Variety
- Veracity
- Value



Agenda for today

- 5 V's of Big Data
- **Cloud computing**
- Highly available / highly Scalable
- Managed vs Unmanaged services



Cloud computing

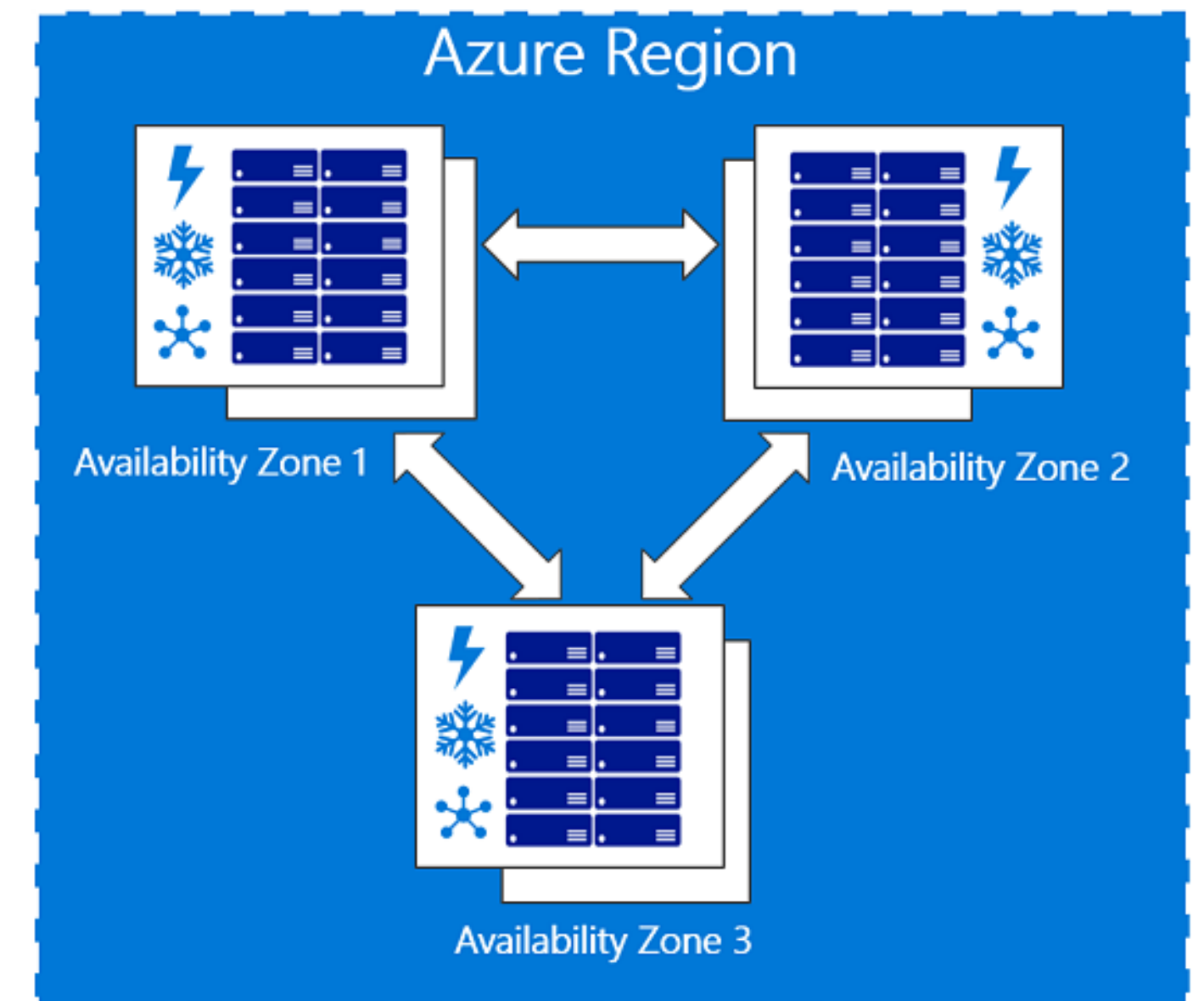


Cloud computing

Not so relevant for the “regular Database course”, but for Big Data it is crucial

Region / AZ / EL

- **Region**
Cluster of data centers in a physical location
- **Availability Zone**
a discrete data center with redundant power, networking, and connectivity in a Region
- **Edge Location**
access to the network with limited services (usually CDN)
- (Names may vary between cloud providers)



AWS Global Infrastructure Map

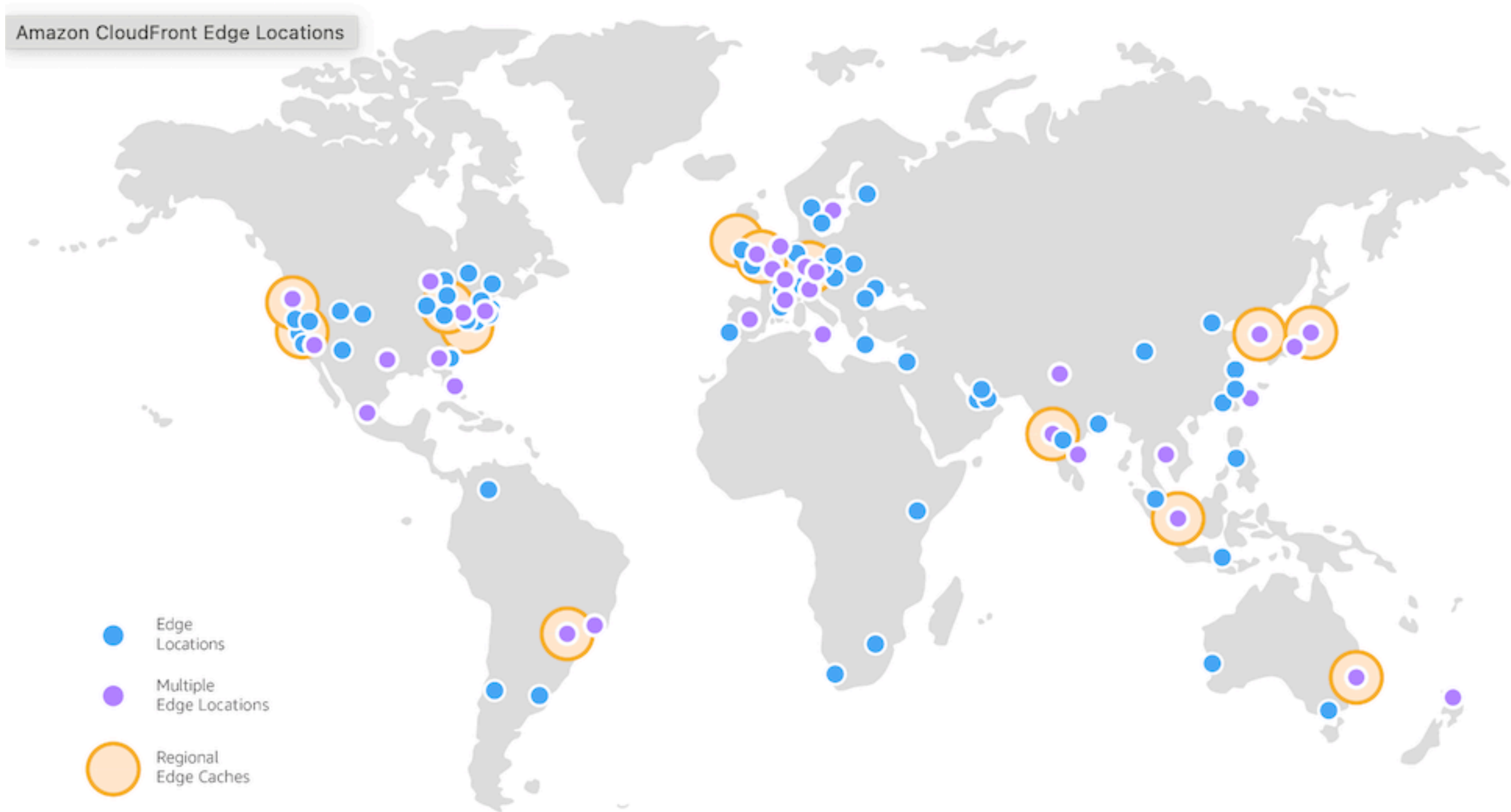
The AWS Cloud spans 105 Availability Zones within 33 geographic regions around the world, with announced plans for 12 more Availability Zones and 4 more AWS Regions in Germany, Malaysia, New Zealand, and Thailand.



AWS regions (January 2024)

 List view

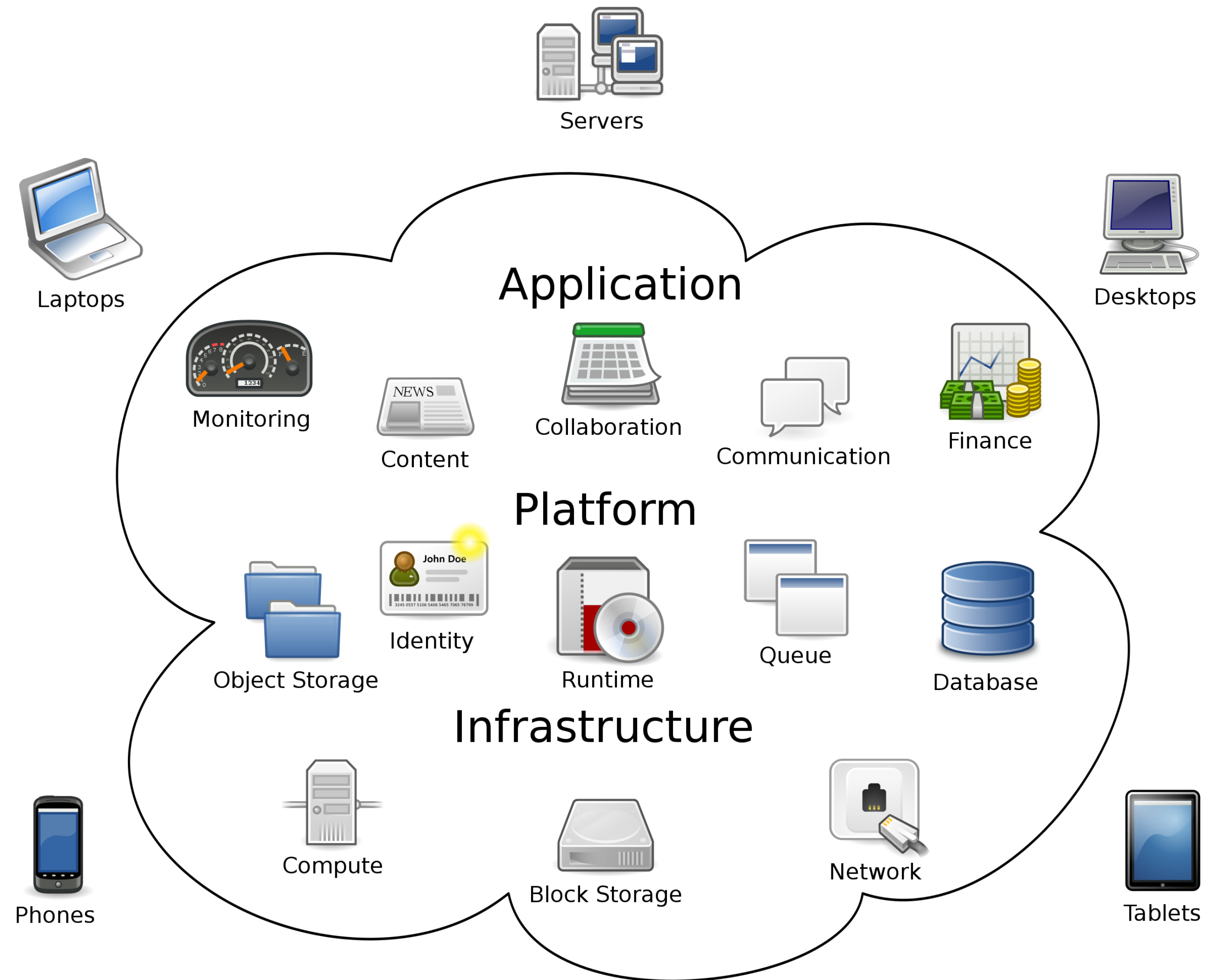
 Regions  Coming soon



AWS edge locations (January 2024)

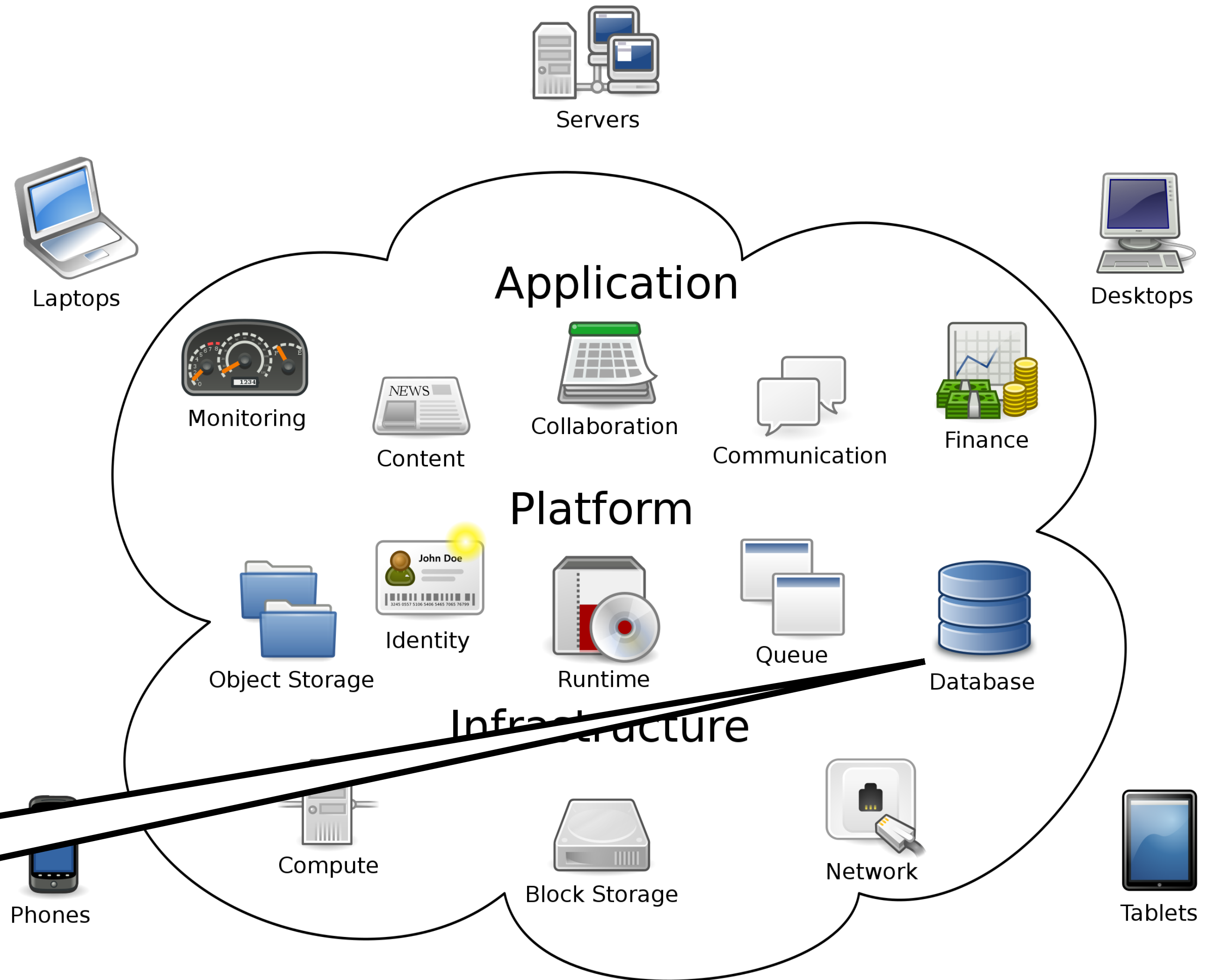
Cloud computing

- **SaaS**
software as a service
- **PaaS**
platform as a service
- **IaaS**
infrastructure as a service



Cloud computing

- SaaS
software as a service
- PaaS
platform as a service
- IaaS
infrastructure as a service



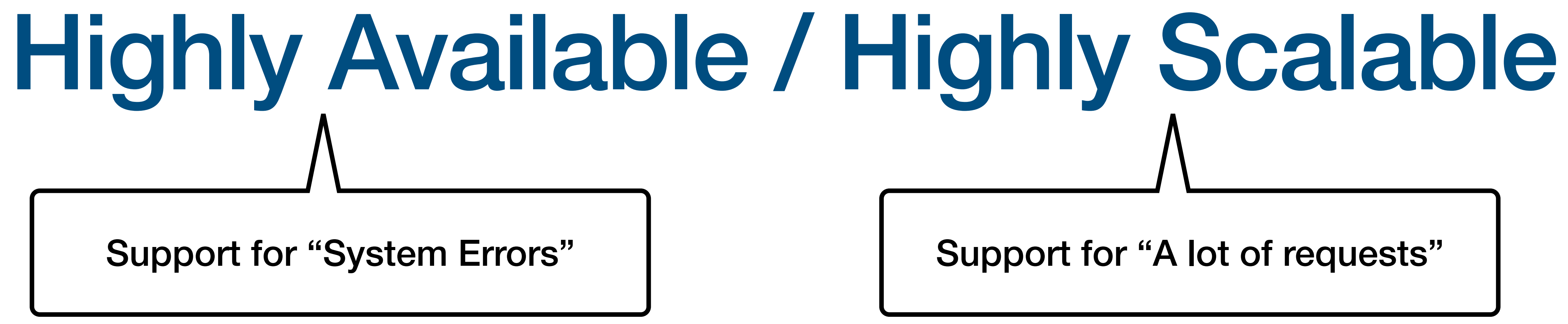
DynamoDB was down →
“Half the internet was down”

Agenda for today

- 5 V's of Big Data
- Cloud computing
- **Highly available / highly Scalable**
- Managed vs Unmanaged services

Highly Available / Highly Scalable

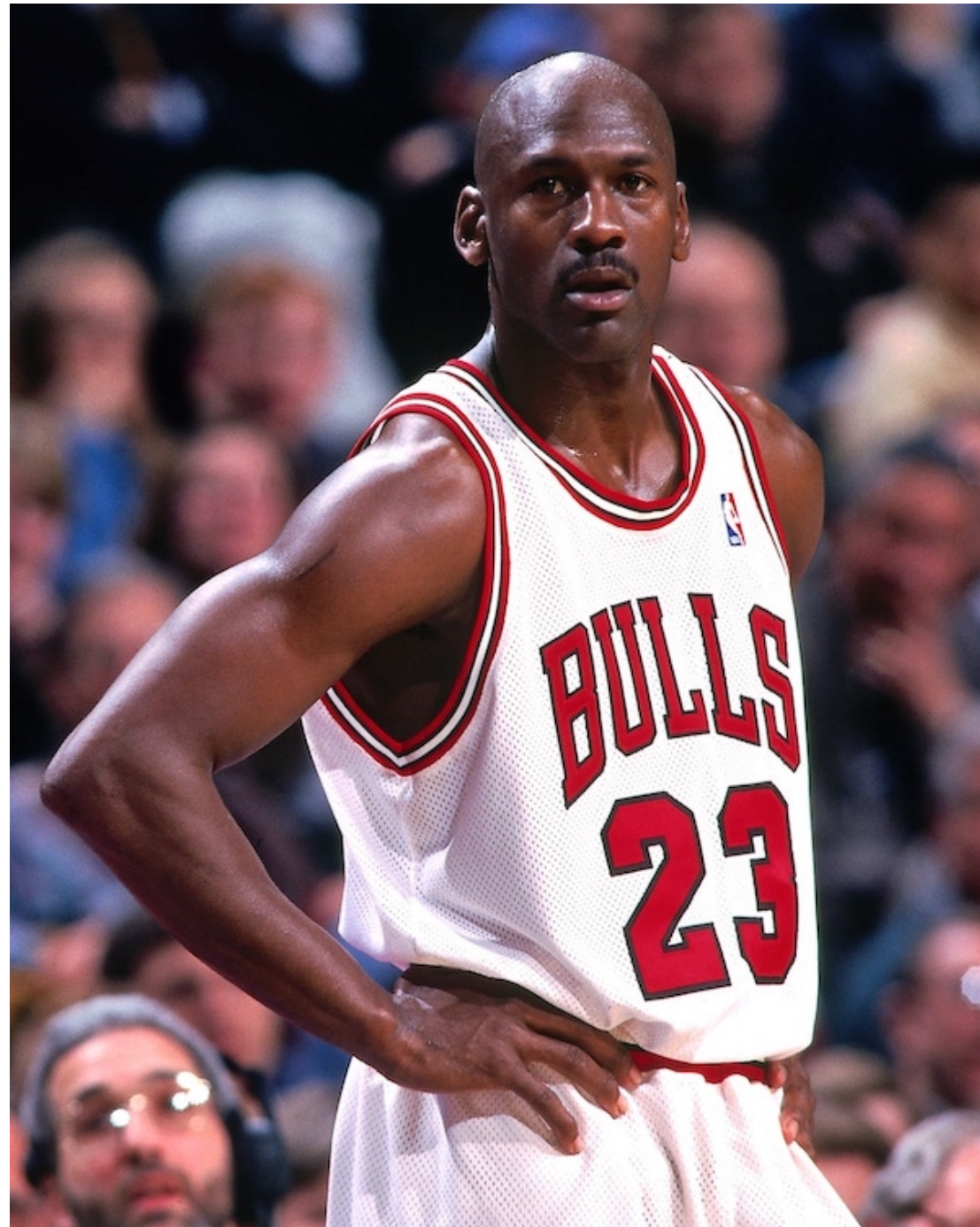
Highly Available / Highly Scalable



Support for “System Errors”

Support for “A lot of requests”

Mike orders a a basketball



Once clicked “order”

- Create order
- Check inventory
- Process payment
- Approve order
- Send to warehouse
- ...



System errors

fire / flood / electricity /
hardware malfunction /
software update...

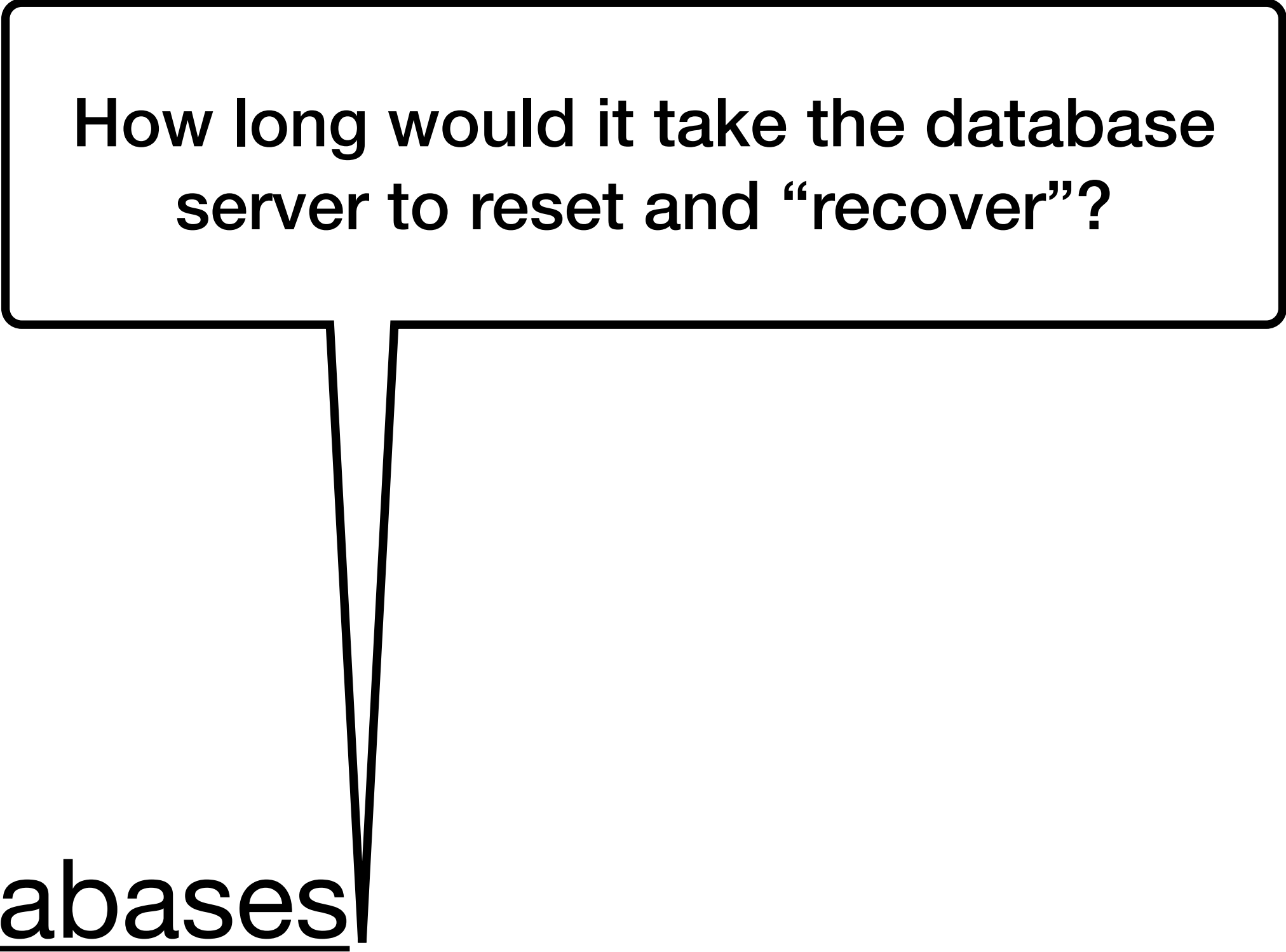
Availability problem

Possible outcomes

- Service disruption
- Data loss
- Data consistency
- Money lost (direct / reputation)
- A hard problem to solve for Databases
disaster recovery:
RTO (Recovery time) / RPO (Recovery point object)

Possible outcomes

- Service disruption
- Data loss
- Data consistency
- Money lost (direct / reputation)
- A hard problem to solve for Databases
disaster recovery:
RTO (Recovery time) / RPO (Recovery point object)



How long would it take the database server to reset and “recover”?

High availability

- “Nines”

Availability	Downtime per day	Downtime per year
90%	2.40 hours	36.53 days
95%	1.20 hours	18.26 days
99%	14.40 minutes	3.65 days
99.9%	1.44 minutes	8.77 hours
99.99%	8.64 seconds	52.60 minutes
99.999%	864.00 milliseconds	5.26 minutes
99.9999%	86.40 milliseconds	31.56 seconds

How would you solve Availability?

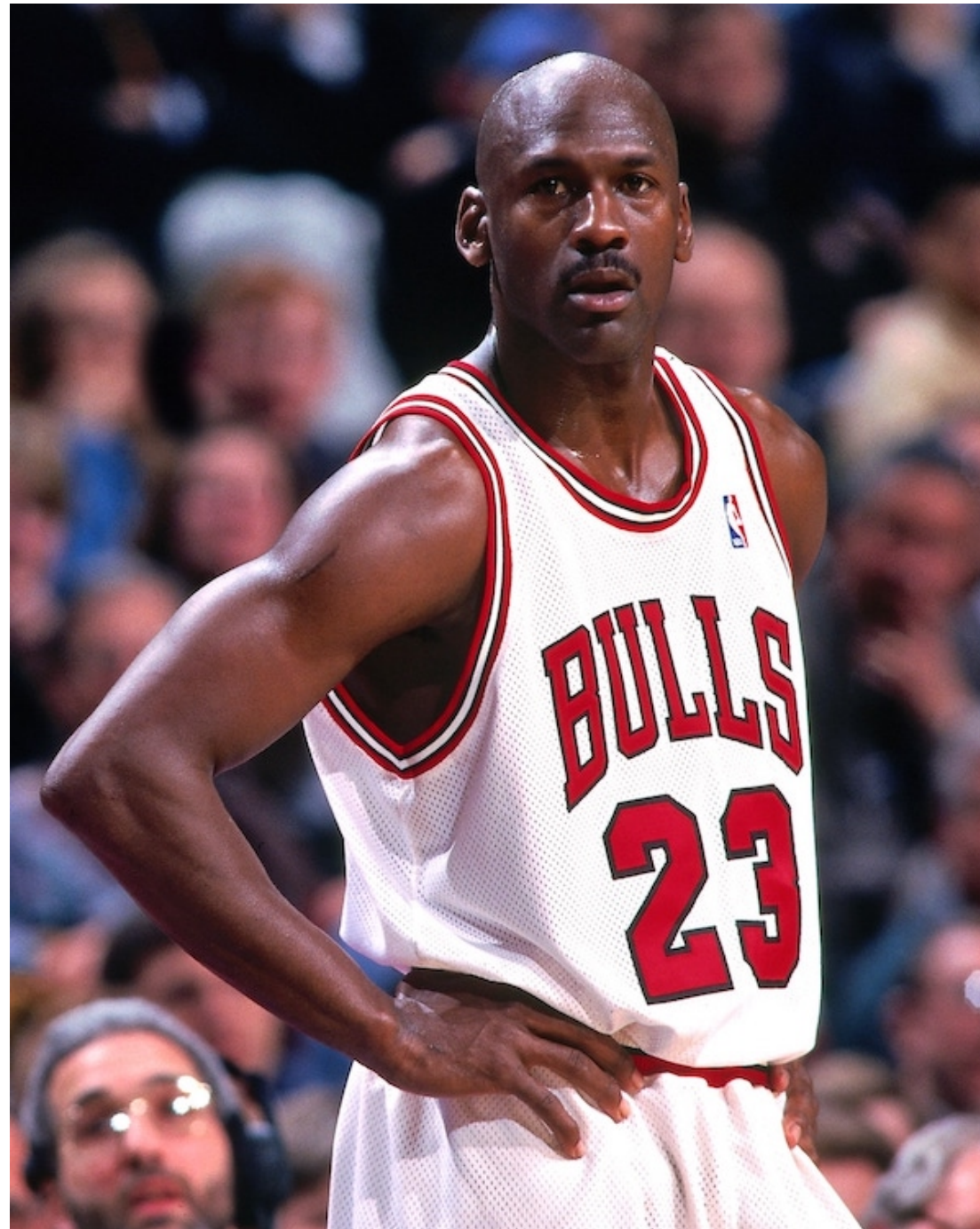
How would you solve Availability?

- Have a backup server on standby
- Have two servers operate simultaneously

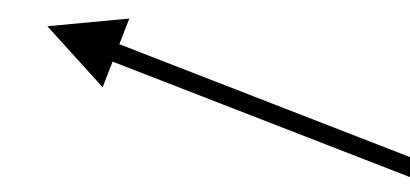
How would you solve Availability?

- Have a backup server on standby
 - How long would it take it to resume?
 - How to keep the backup server's data up to date?
 - ...
- Have two servers operate simultaneously
 - How to decide which server accept which request?
 - How to sync their data?
 - ...

Mike tweets about a basketball he bought



- Reach millions of users
- Millions of users try to buy the same basketball at the same time



System error

Too many requests

Scalability problem

How can you support millions of users?

How can you support millions of users?

- “Easy” - lets “use” much more servers
- But how would they “work” together?
- Do we need all servers 24h?

High scalability - key properties

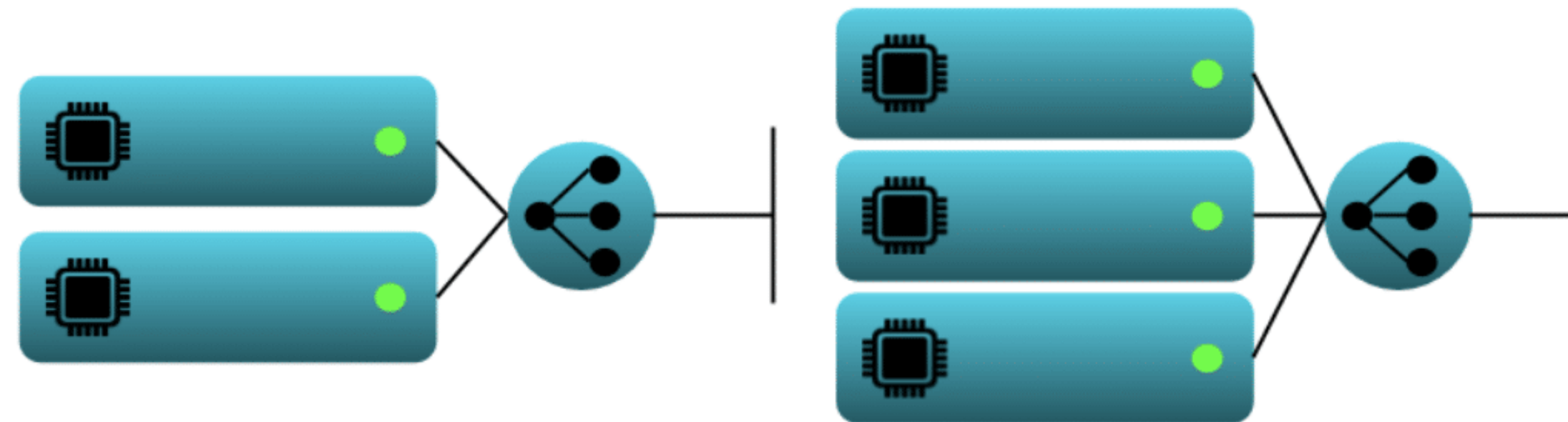
- Scale up vs scale out
commodity computing
- **Microservices**
- **Stateless**
amazon's shopping cart is stateless?
- **Data Sharding**

Scale up vs Scale out

- Commodity hardware



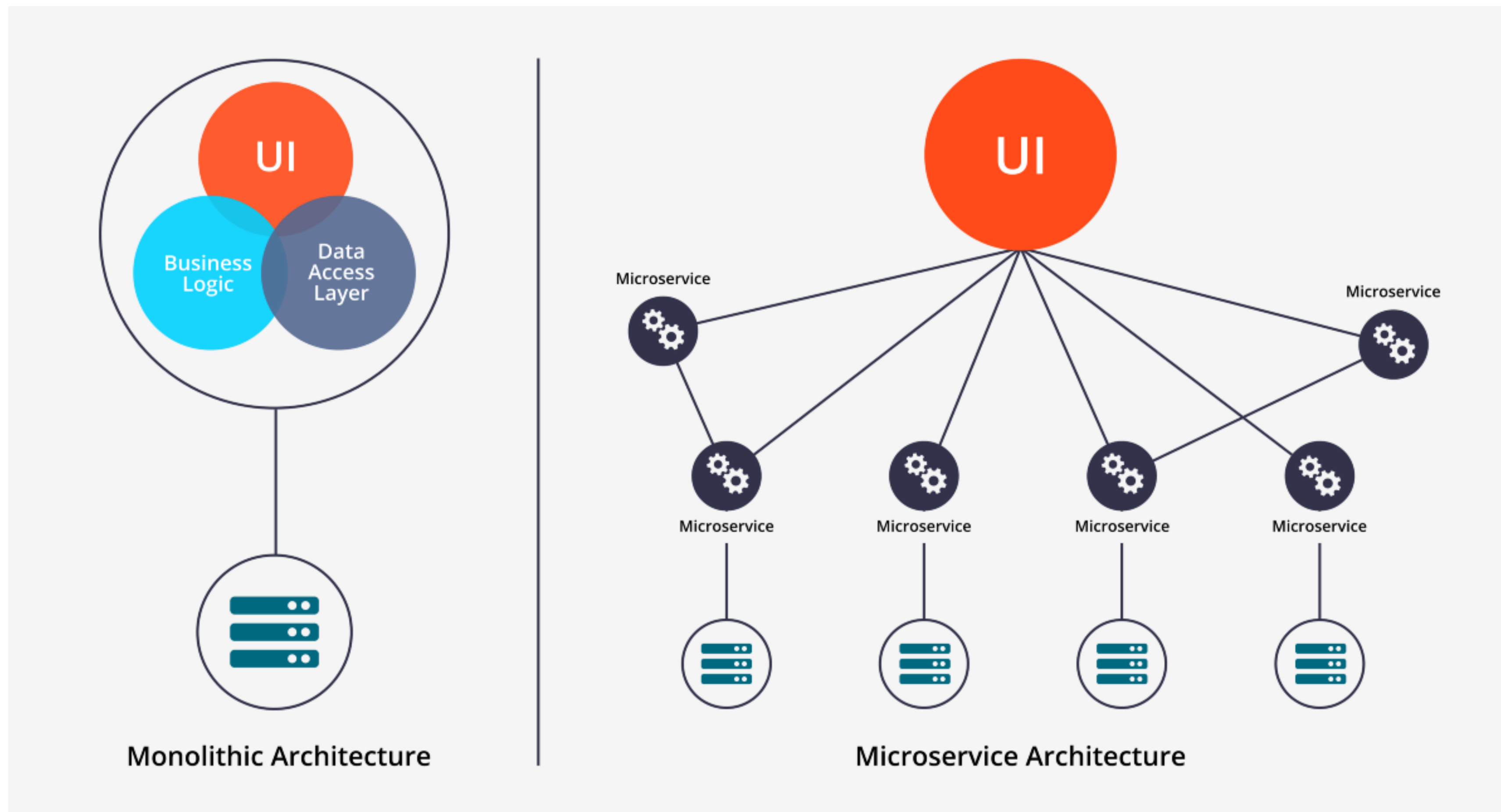
Scaling up from two to three CPUs



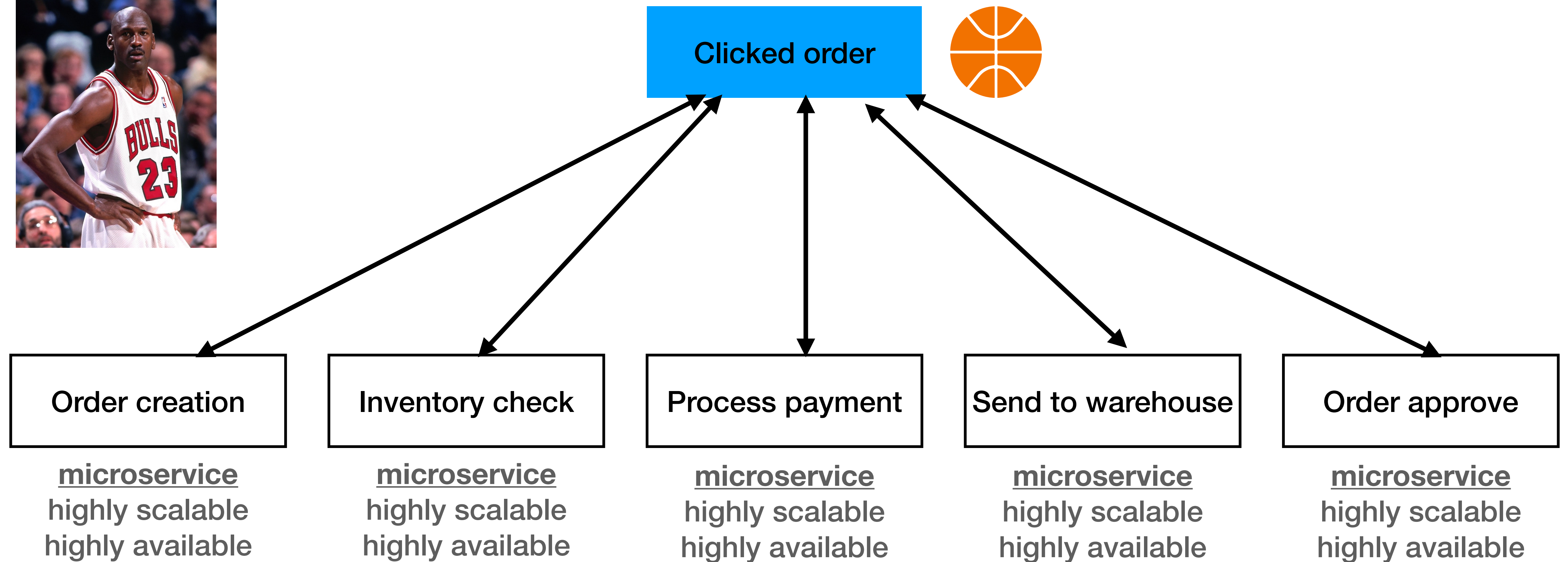
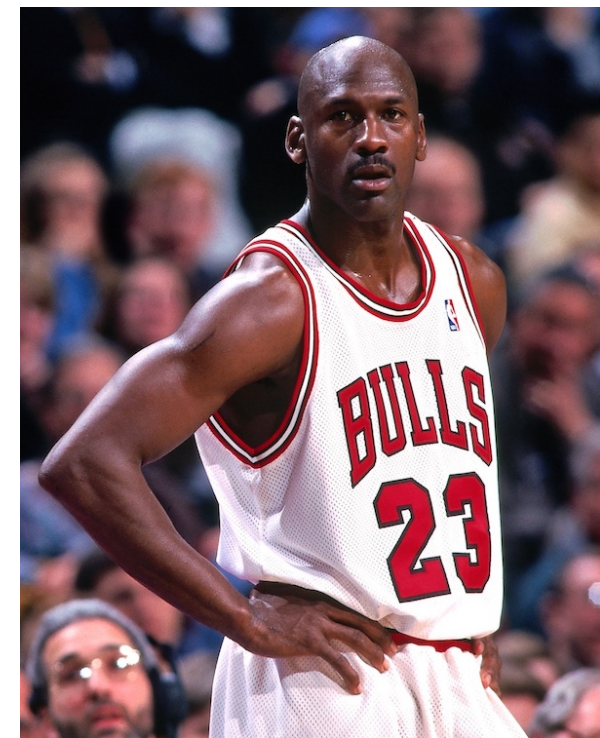
Scaling out from two to three CPUs

Microservices

- Split a big task into **loosely coupled** services



Example



Example - inventory check

- Ok so we have a smaller task - how to support millions of users?

Example - inventory check

- Ok so we have a smaller task - how to support millions of users?

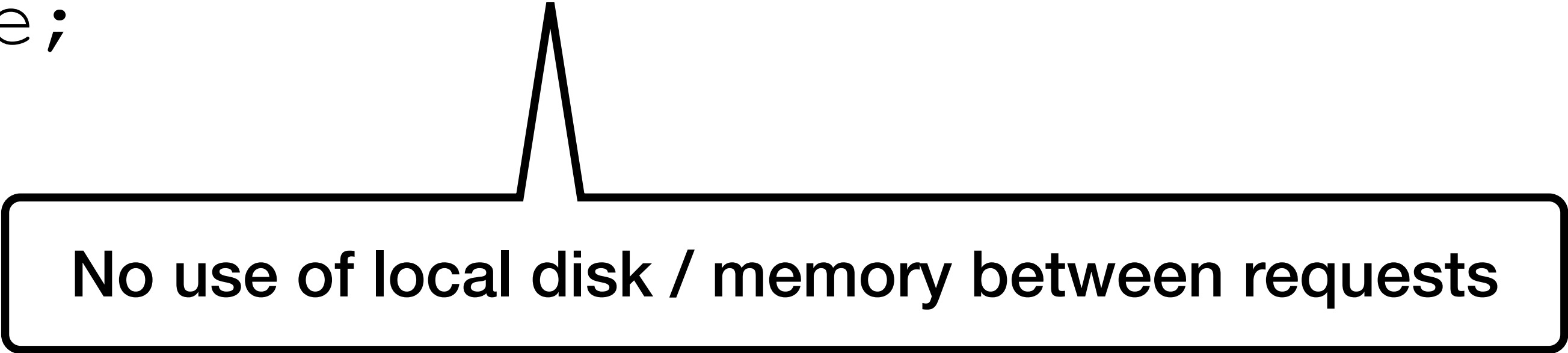
—> Stateless logic + load balancer + auto scaling

Statless

- Similar to “Model of computation” course
- Intuition:
 - **Do not store anything on local disk or memory**
 - Any server can handle any request

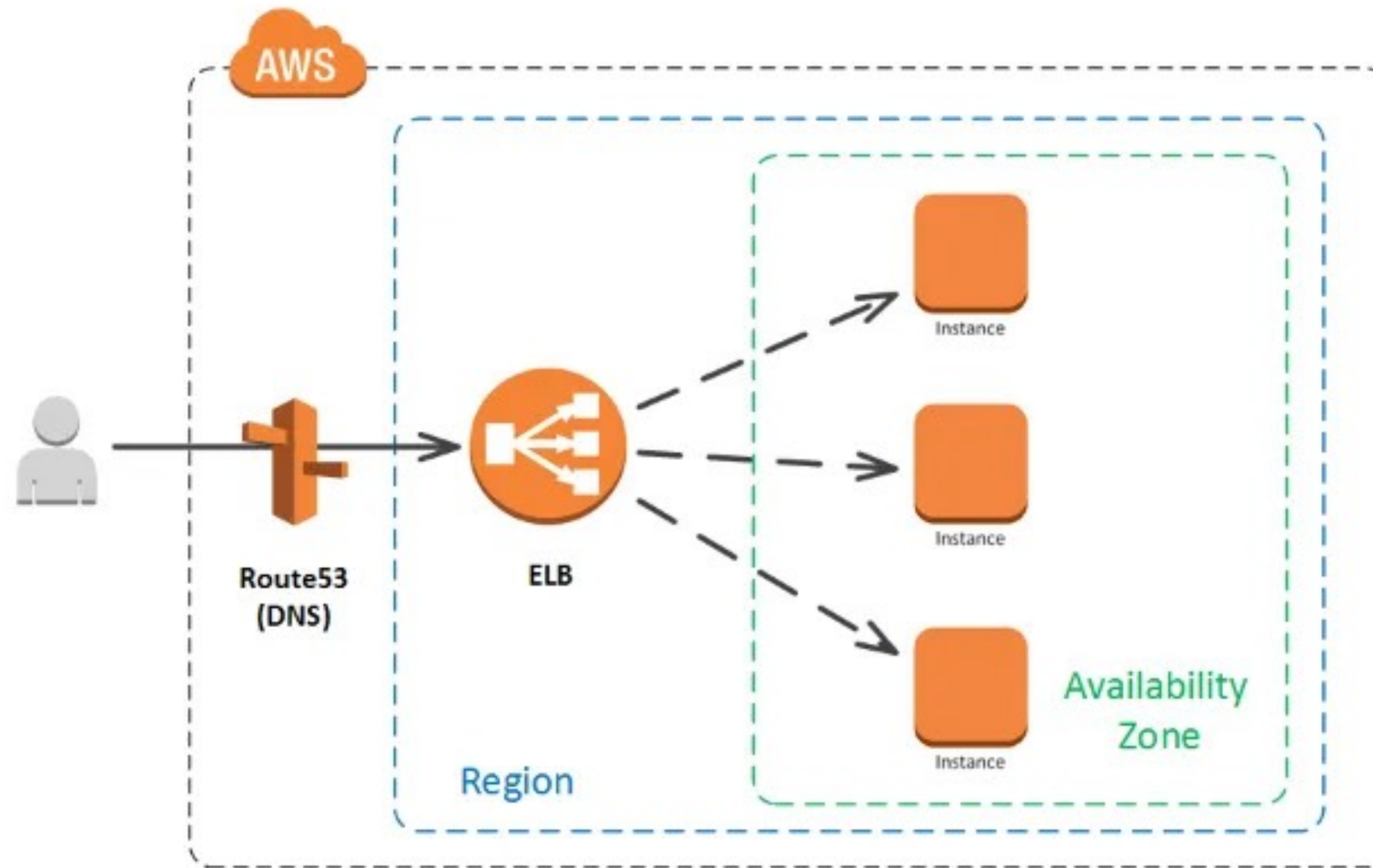
Example - Route Planning - stateless

```
private Route getDirections(Location src, Location dest) {  
    // do Dijkstra or something like that..  
    ...  
    return route;  
}
```

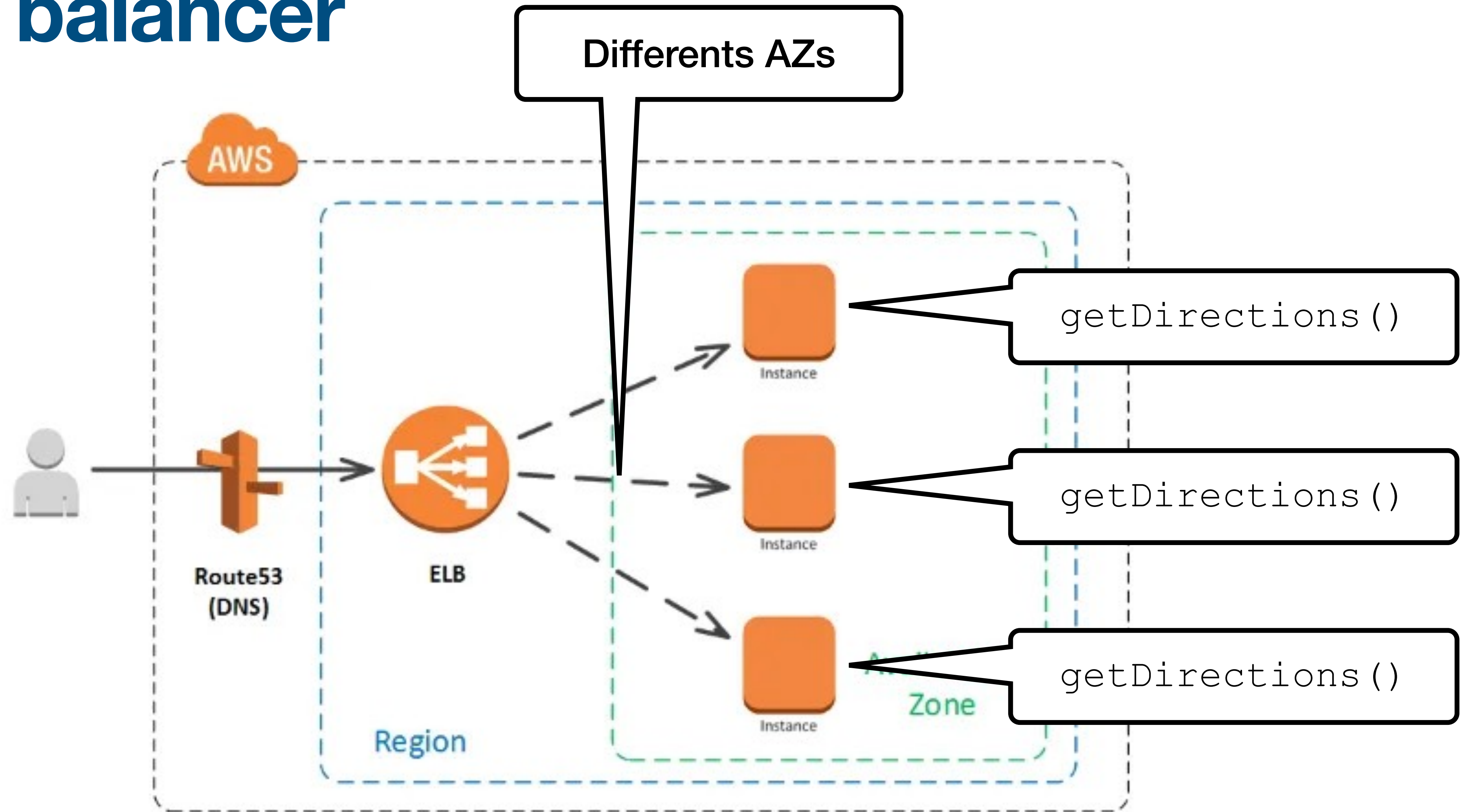


No use of local disk / memory between requests

Load balancer

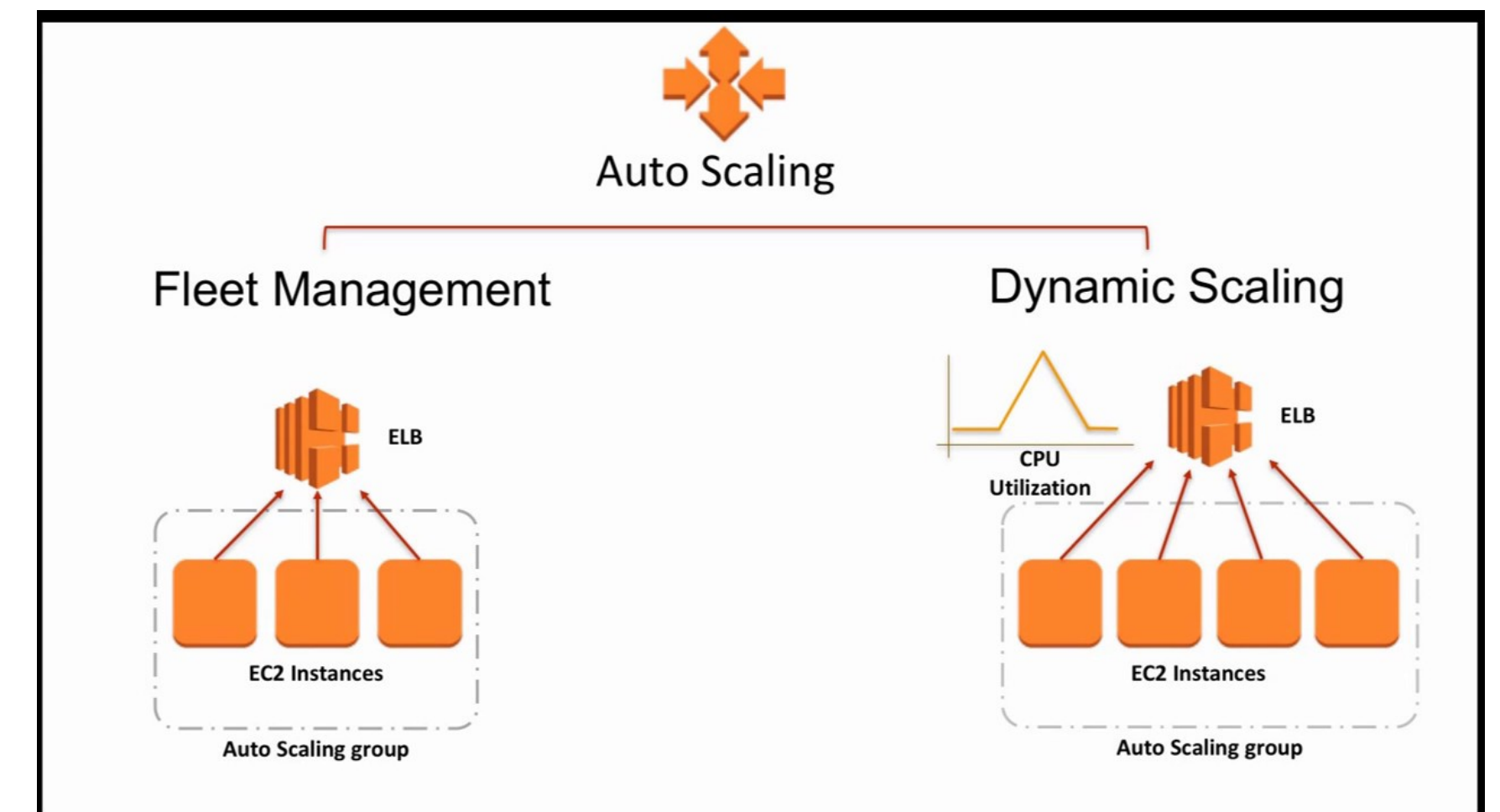


Load balancer



Auto scaling

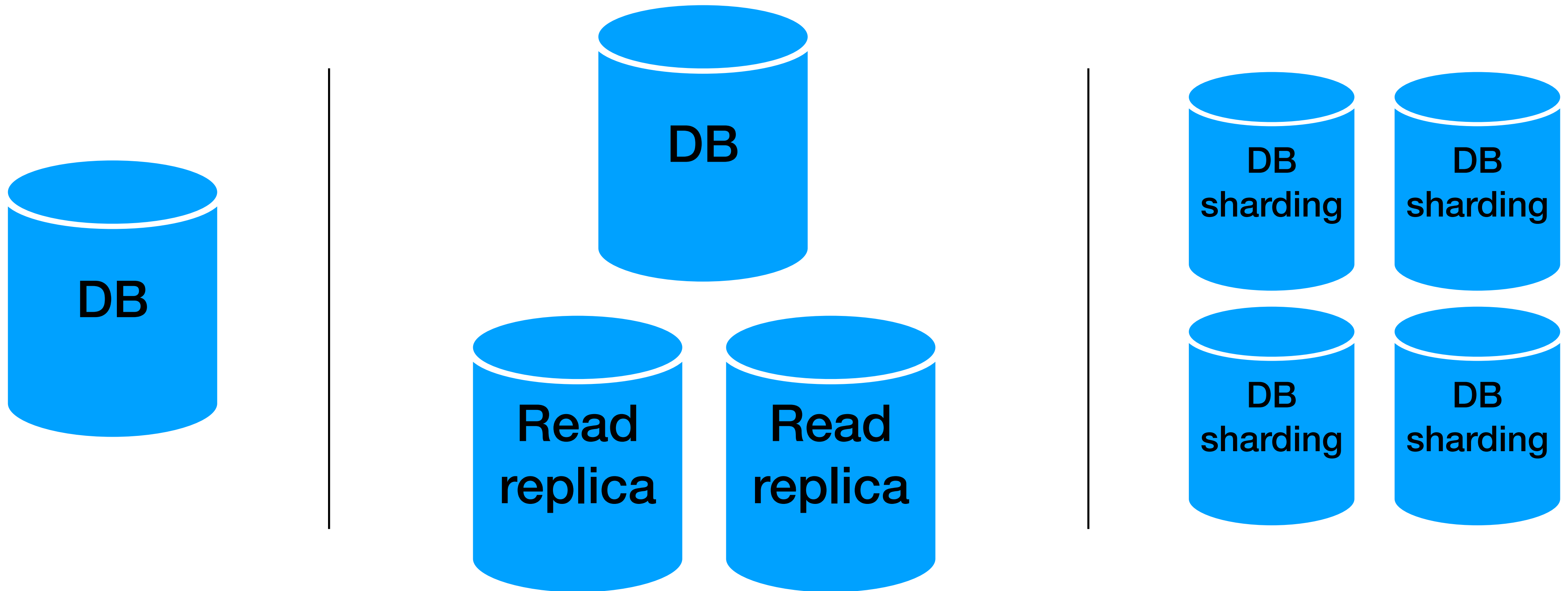
- When threshold occurs (hits / traffic / CPU...), create a new instance with the same logic and add it to the load balancer
- When threshold drops, remove the instance from the load balancer and terminate it
- Usually requires stateless logic
can Cassandra work with auto scale?



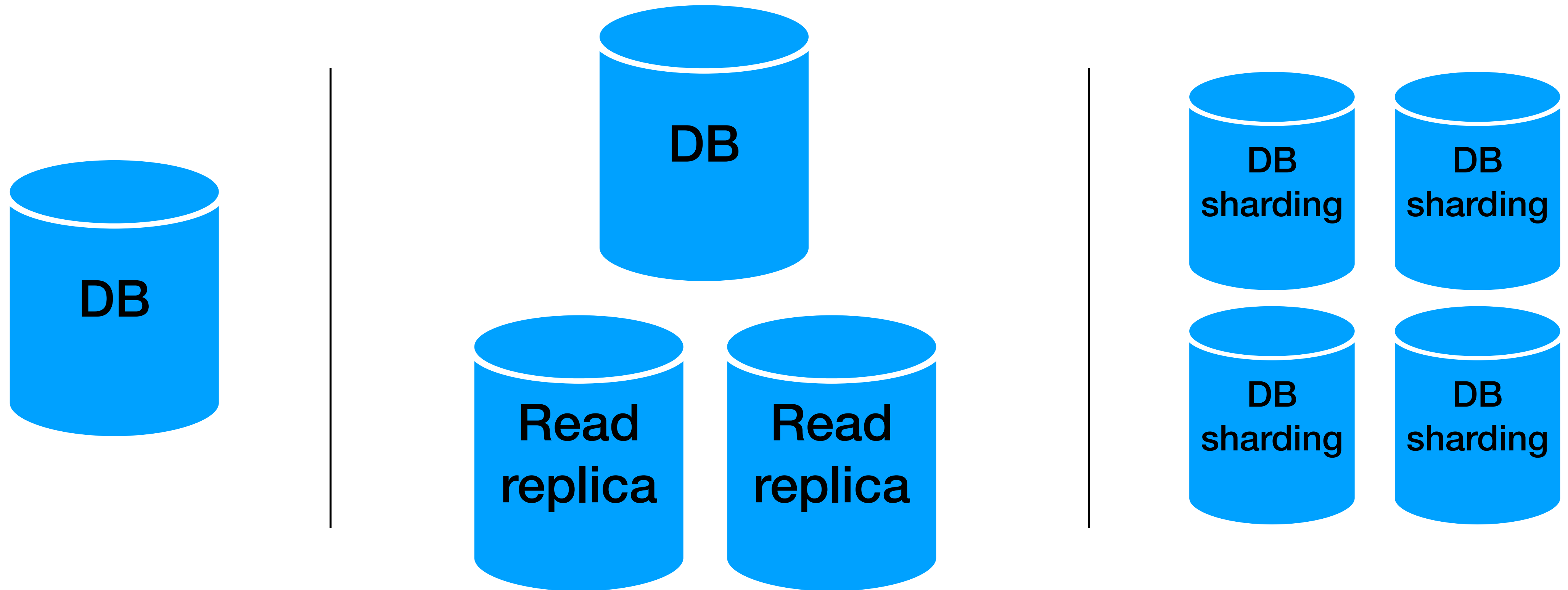
Auto scaling - compute + storage?

- Some applications use both compute and storage (for example the db service in the example)
- Stateless?
- What happens when we scale down?

Scaling databases



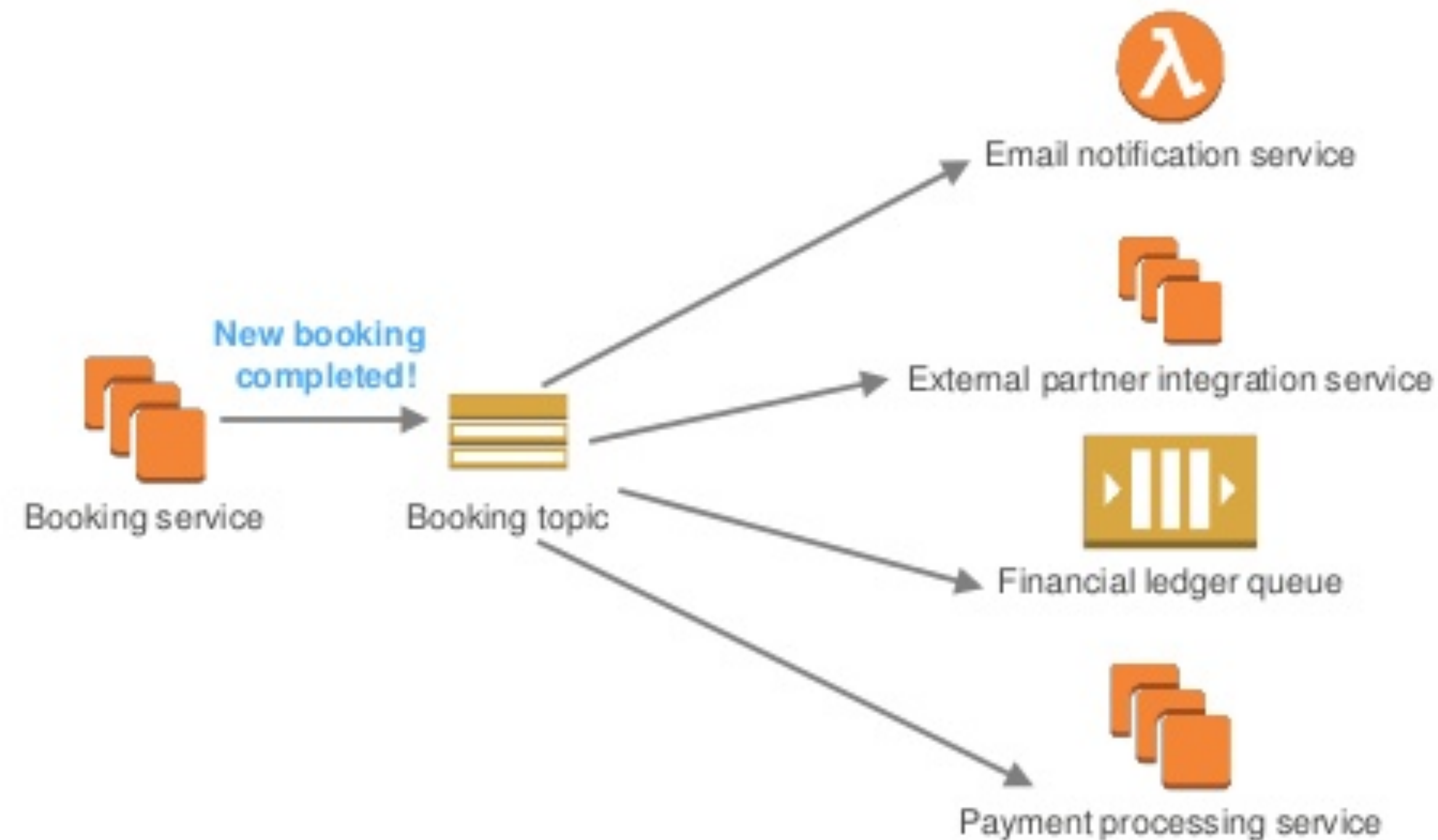
Scaling databases



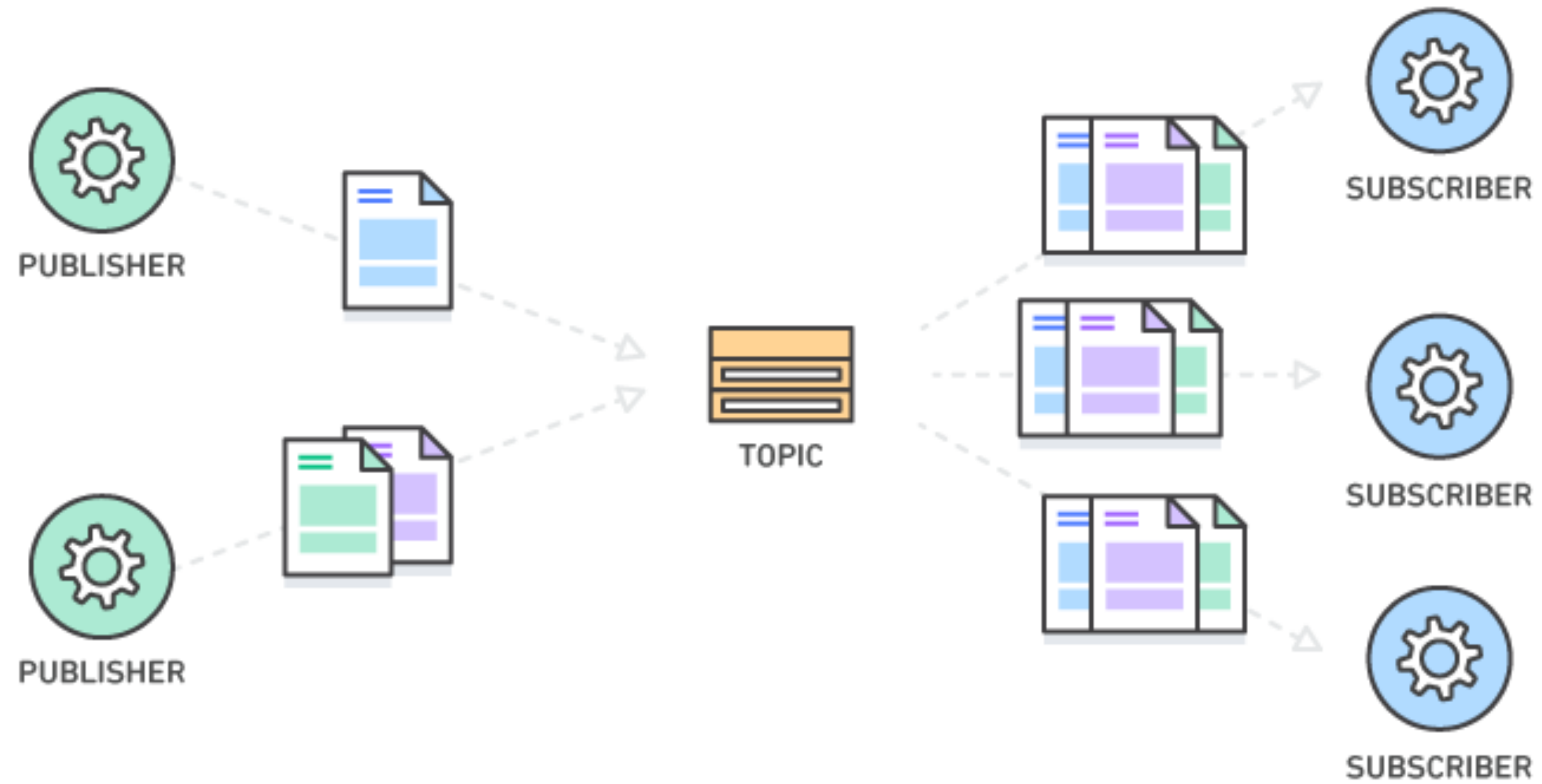
Warning - we will talk about this a lot :)

Decoupling + event based services

- autonomous and unaware of each other services



Pub sub



Agenda for today

- 5 V's of Big Data
- Cloud computing
- Highly available / highly Scalable
- **Managed vs Unmanaged services**

Unmanaged service

You are responsible for everything!

- Choosing CPUs, storage, network...
- Installing OS, Java, core software, dependencies...
- Patches, updates
- Security
- Backup
- Monitoring
- Availability

Unmanaged service (2)

Requires different skills

- System
- DevOps
- ...

Managed service

- All the stuff we talked about before are managed for you out of the box
- Hardware utilization
- Focus on stuff that really matters for you
- Cost?

Managed service cons

- **Cloud locked in**
- Slightly limited functionality
- Works only in the cloud
- **Cost?**
(cheaper to go unmanaged on large scale, but a lot of headaches)

In practice

- Some will be managed and some not
 - VMs
 - load balancers
 - network stuff
 - ...
- **To go managed or unmanaged with databases is a good question**

Managed vs Unmanaged Databases

Fully managed services on AWS

Spend time innovating & building new apps, not managing infrastructure

Self managed

You

Schema design
Query construction
Query optimization
Automatic failover
Backup & recovery
Isolation & security
Industry compliance
Push-button scaling
Automated patching
Advanced monitoring
Routine maintenance
Built-in best practices

Fully managed

You

AWS

מהשפעות התקלה ב-AWS: מיטות חכמות התחממו, קפאו ונתקעו בזווית שלא מאפשרת לישון

הגאדג'טים הכי מגניבים למיטה שלכם עוררו (סליחה) סערה השבוע, כשהתקלה של אמזון הובילה להרבה צרות אצל המשתמשים

{ נאדניטים } • עידן בן טובים • 24.10.25



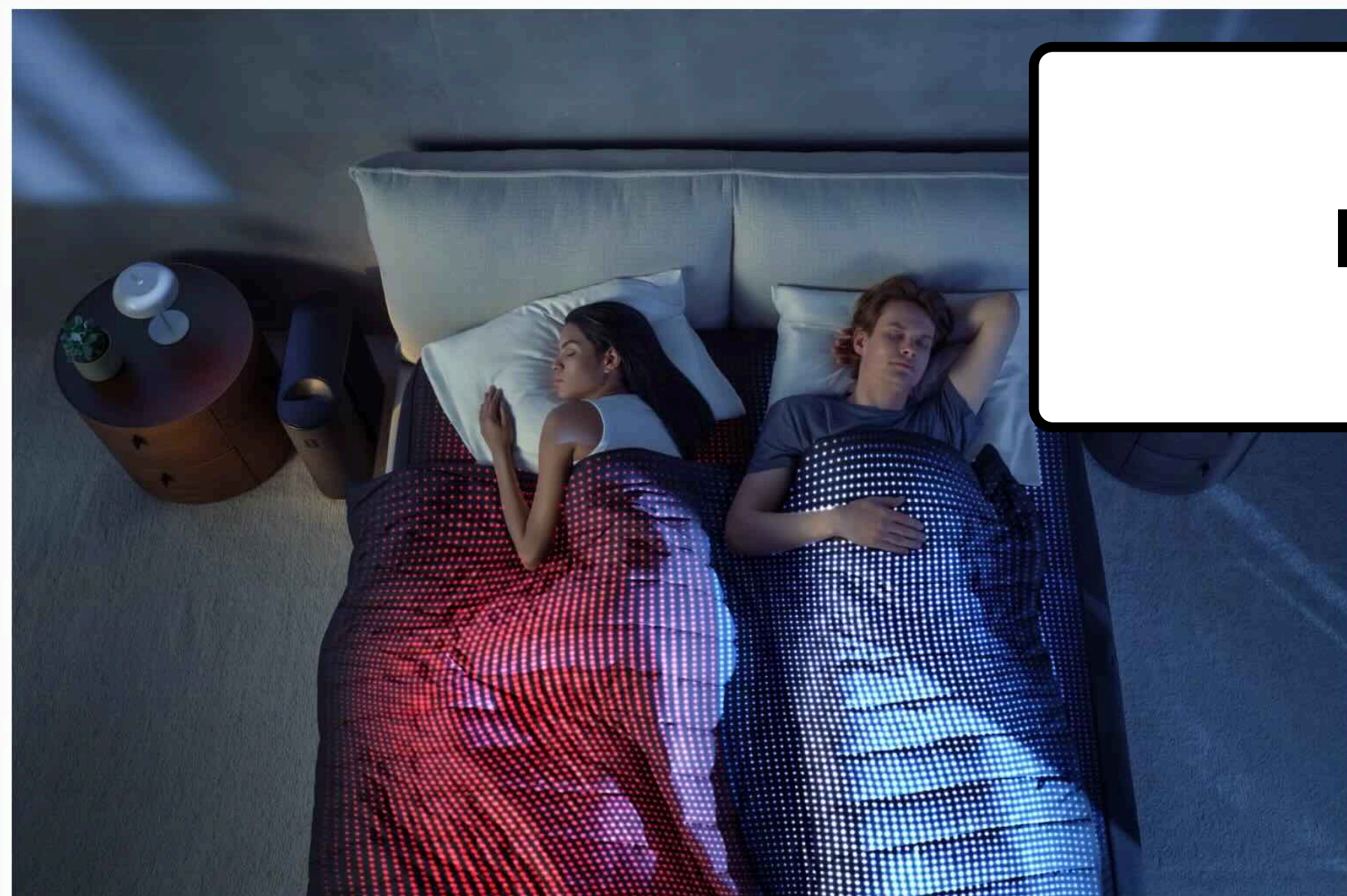
Powered by **Trinity Audio**

00:00

למה לקרוא כשאפשר להאזין?

A 文 10 10 1.0x

04:25



תמונה: Eight Sleep

In which case the

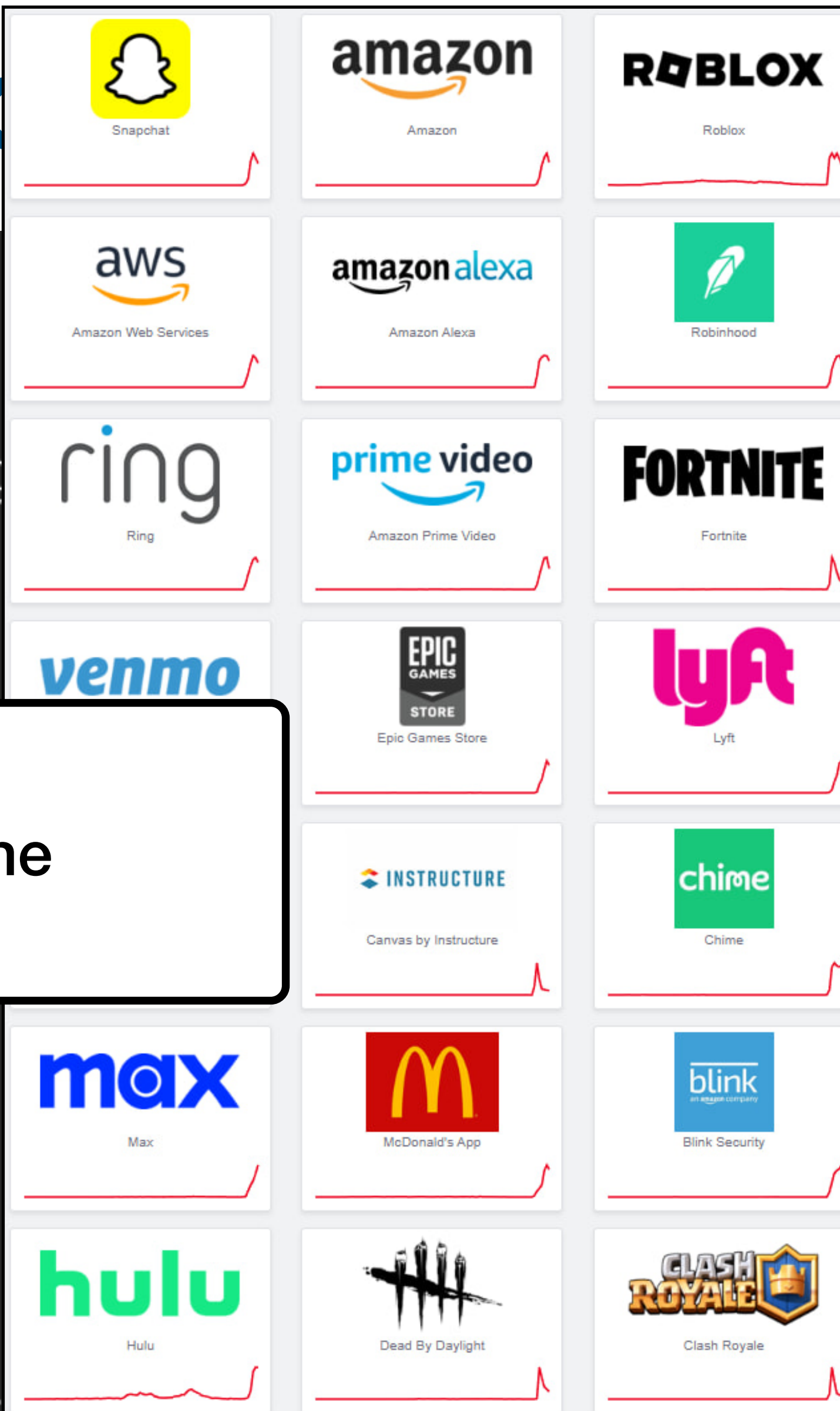
managed

s on AWS

w apps, not managed

chema design

ation
security
ustry compliance
h-button scaling
omated patching
anced monitoring
tine maintenance
-in best practices



But how managed service work?

- It is just someone else's software...
- Do we need to understand how it works behind the scenes?

For databases, YES!

Big Data databases

- Managed big data databases are built on, well, big data databases
- **Data modeling is crucial.**
(with bad modeling, nothing will work)

**To model data correctly,
we need to understand the technology**
(it is not just reading the API docs)