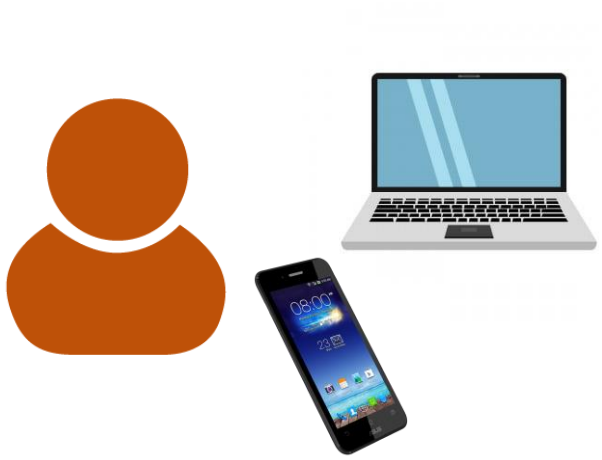
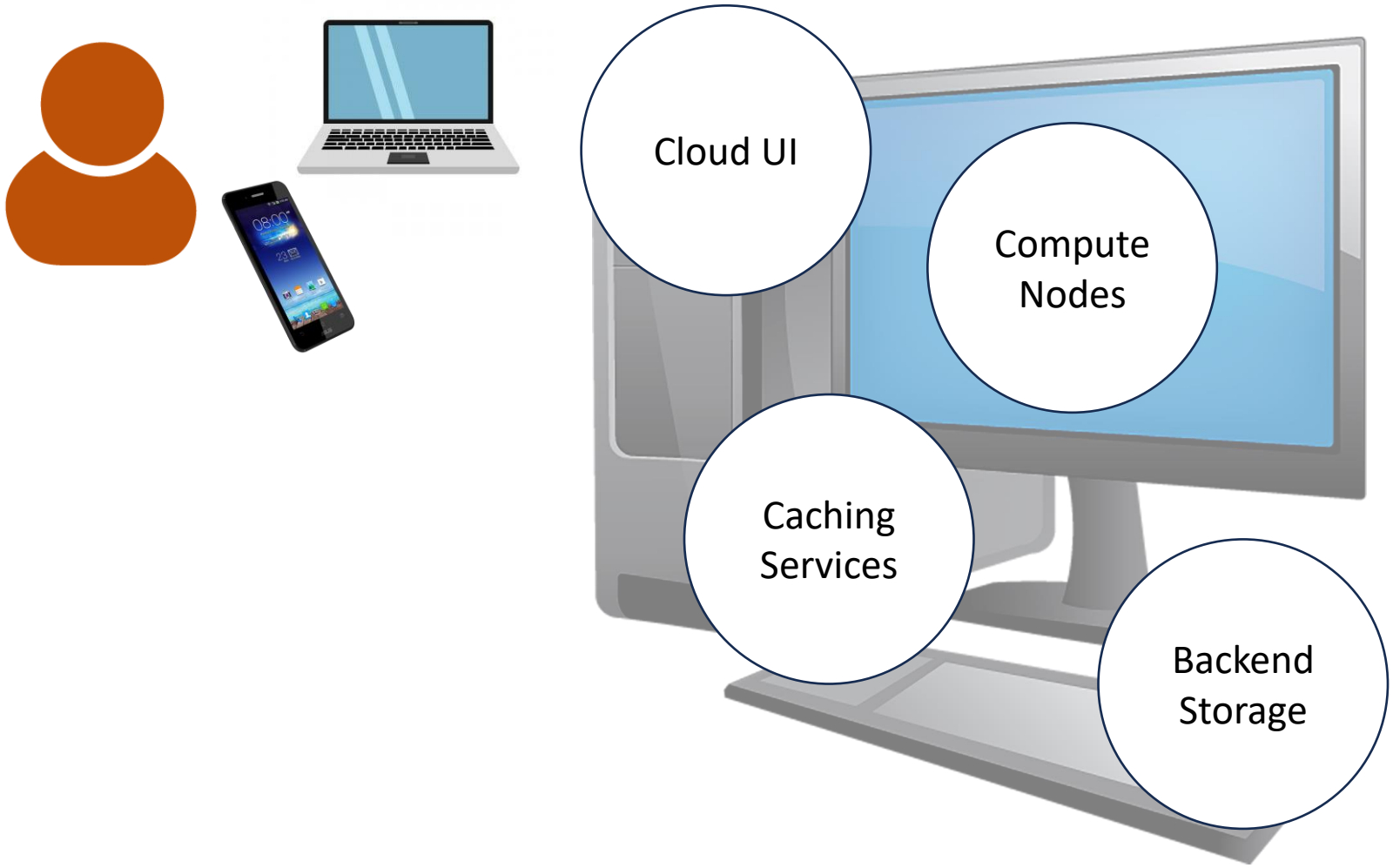
Cloud Computing

ENGR 689 (Sprint)

A Computer in the Cloud



A Computer in the Cloud



Why Cloud Computing?

- **Cost Efficiency (Economy of Scale):**
  - Reduce facility, management, power, innovation cost
- **Multi-tenancy:**
  - Accommodating multiple users in one infrastructure
- **Elasticity:**
  - Adaptive resource allocation for customers' need
  - Pay-per-use
- **Ease of management:**
  - Variety of cloud services and utilities
  - Fault/crash tolerance and disaster recovery

Why Cloud Computing?

- **Cost Efficiency (Economy of Scale):**
  - Reduce facility, management, power, innovation cost
- **Multi-tenancy:**
  - Accommodating multiple users in one infrastructure
- **Elasticity:**
  - Adaptive resource allocation for customers' need
  - Pay-per-use
- **Ease of management:**
  - Variety of cloud services and utilities
  - Fault/crash tolerance and disaster recovery

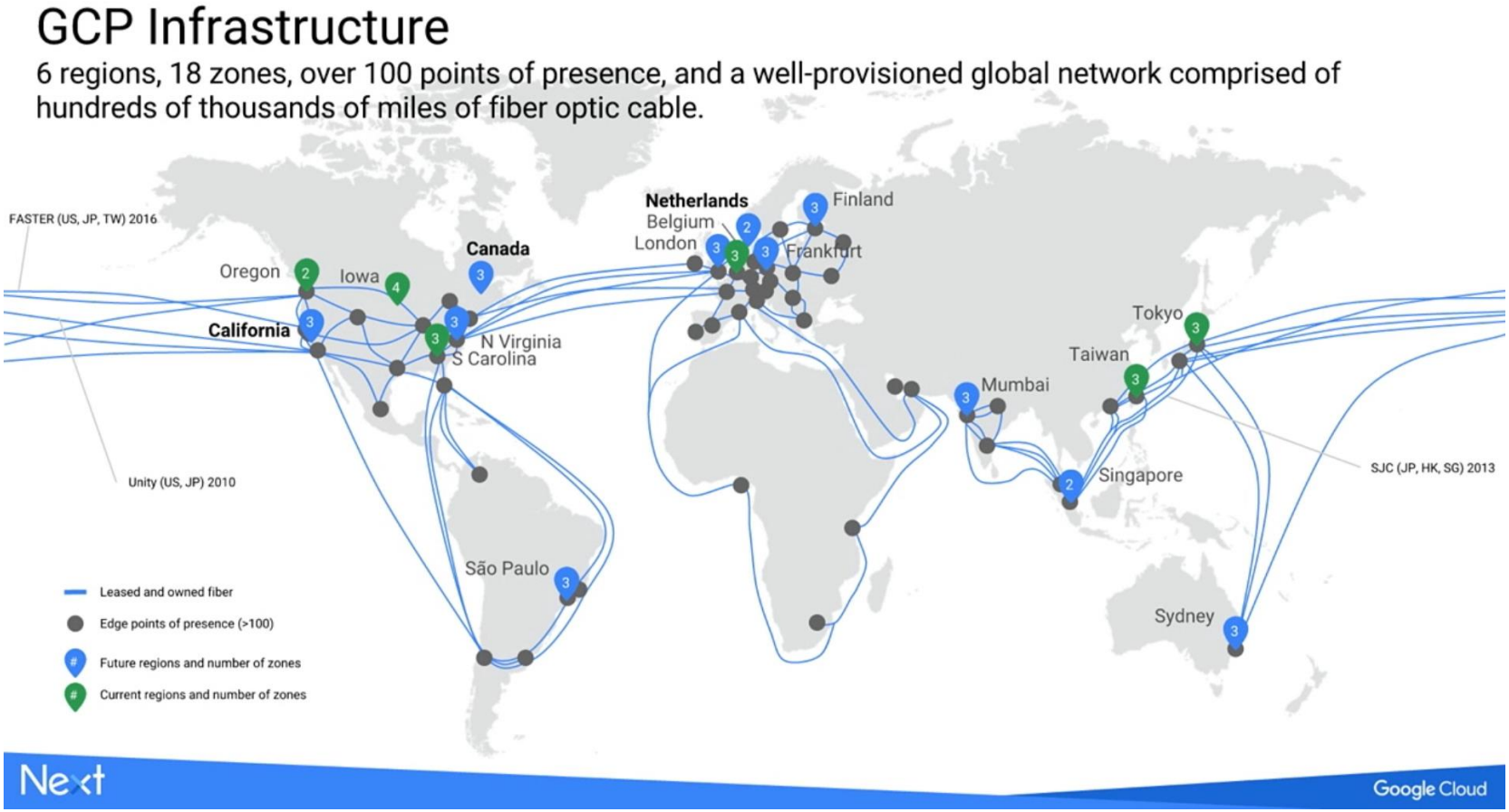
Datacenters with 100,000+ Servers

Google Datacenter

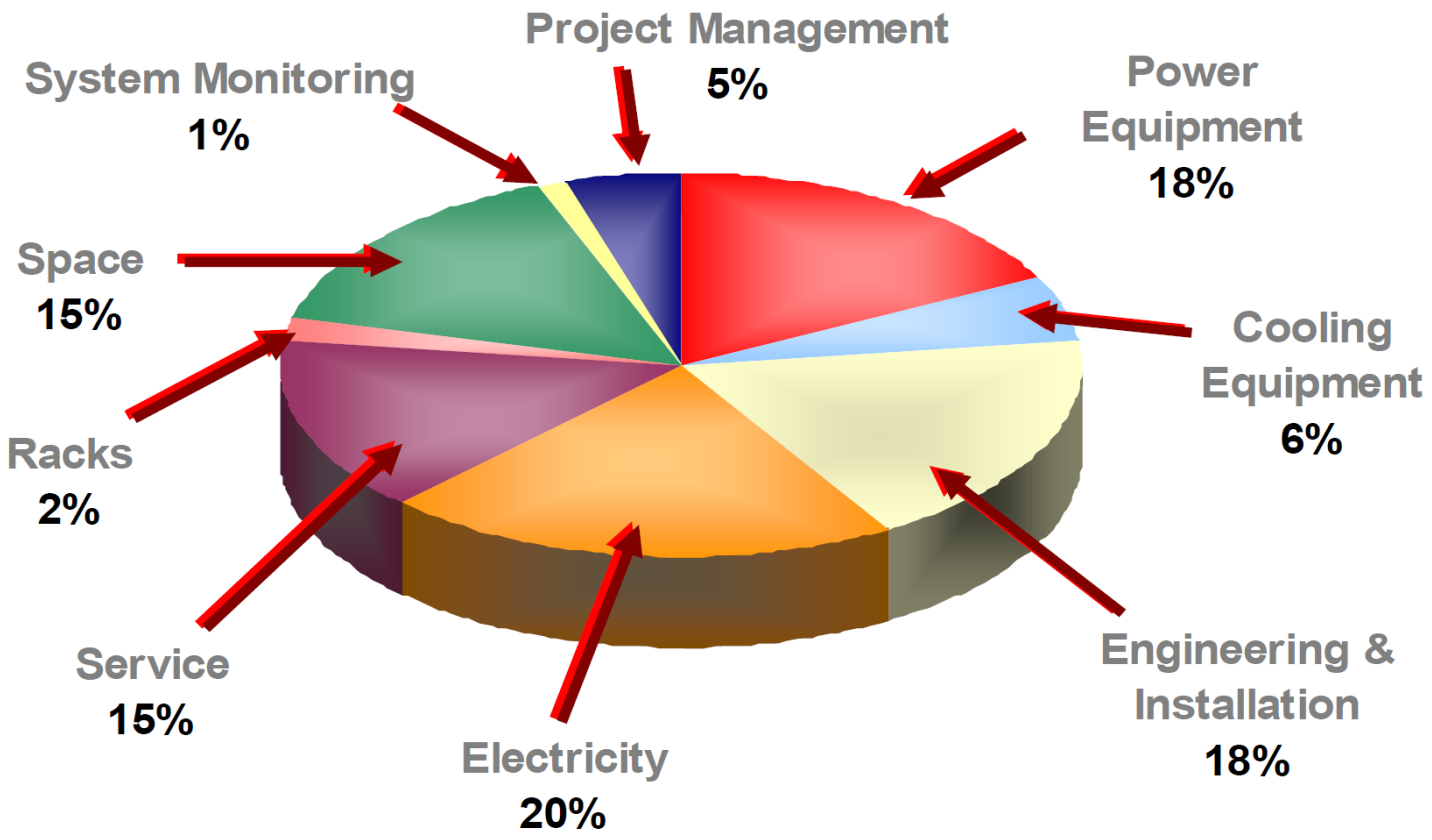
Microsoft Datacenter

Facebook Datacenter

Datacenters Across the Globe



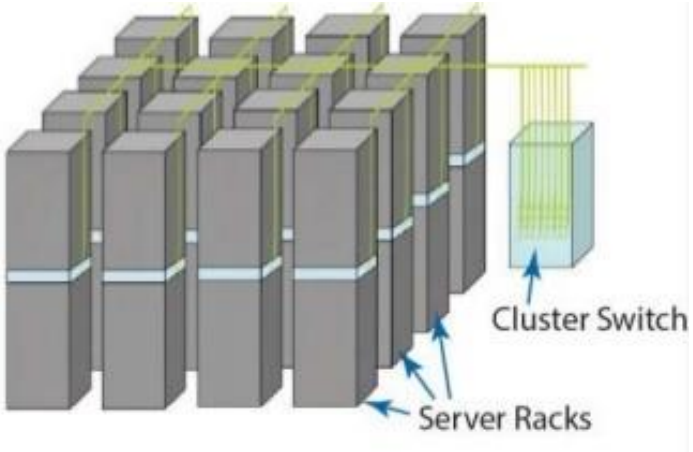
Cost of Maintaining a Datacenter



Internal of a Datacenter

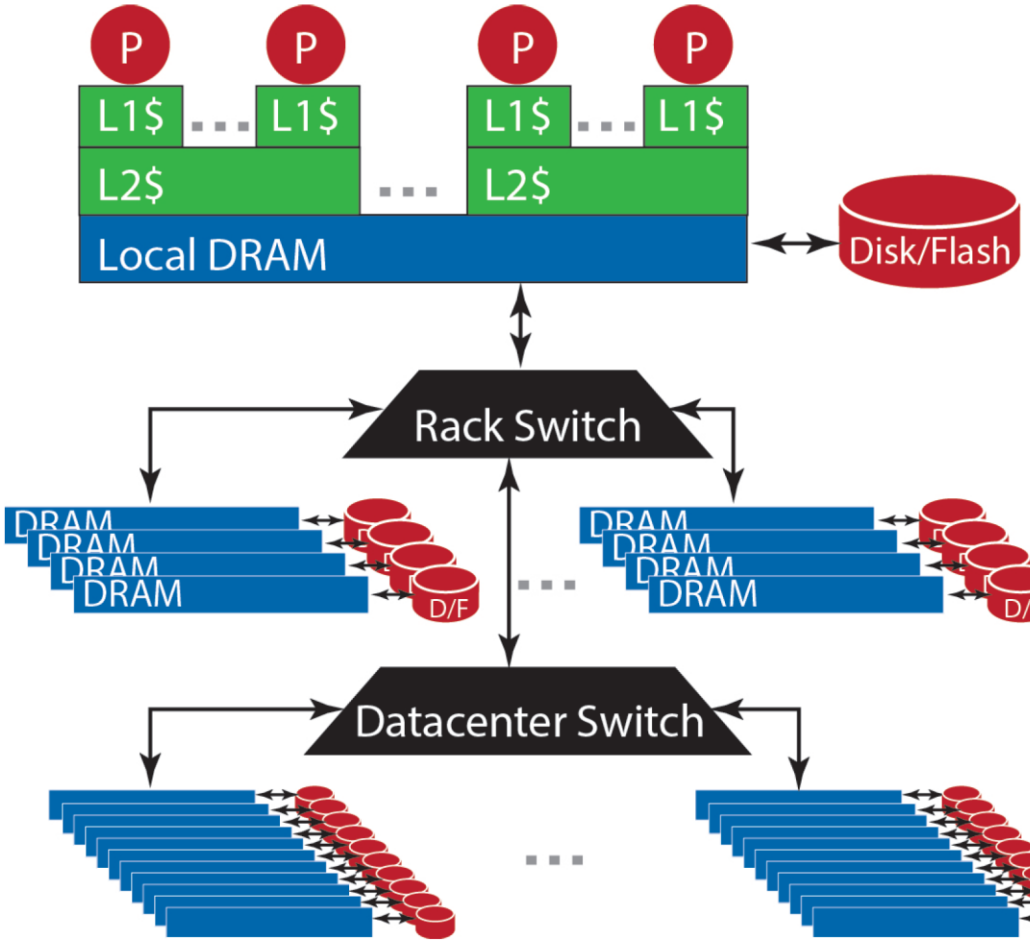
Server

Rack



Cluster

Aggregating Resources

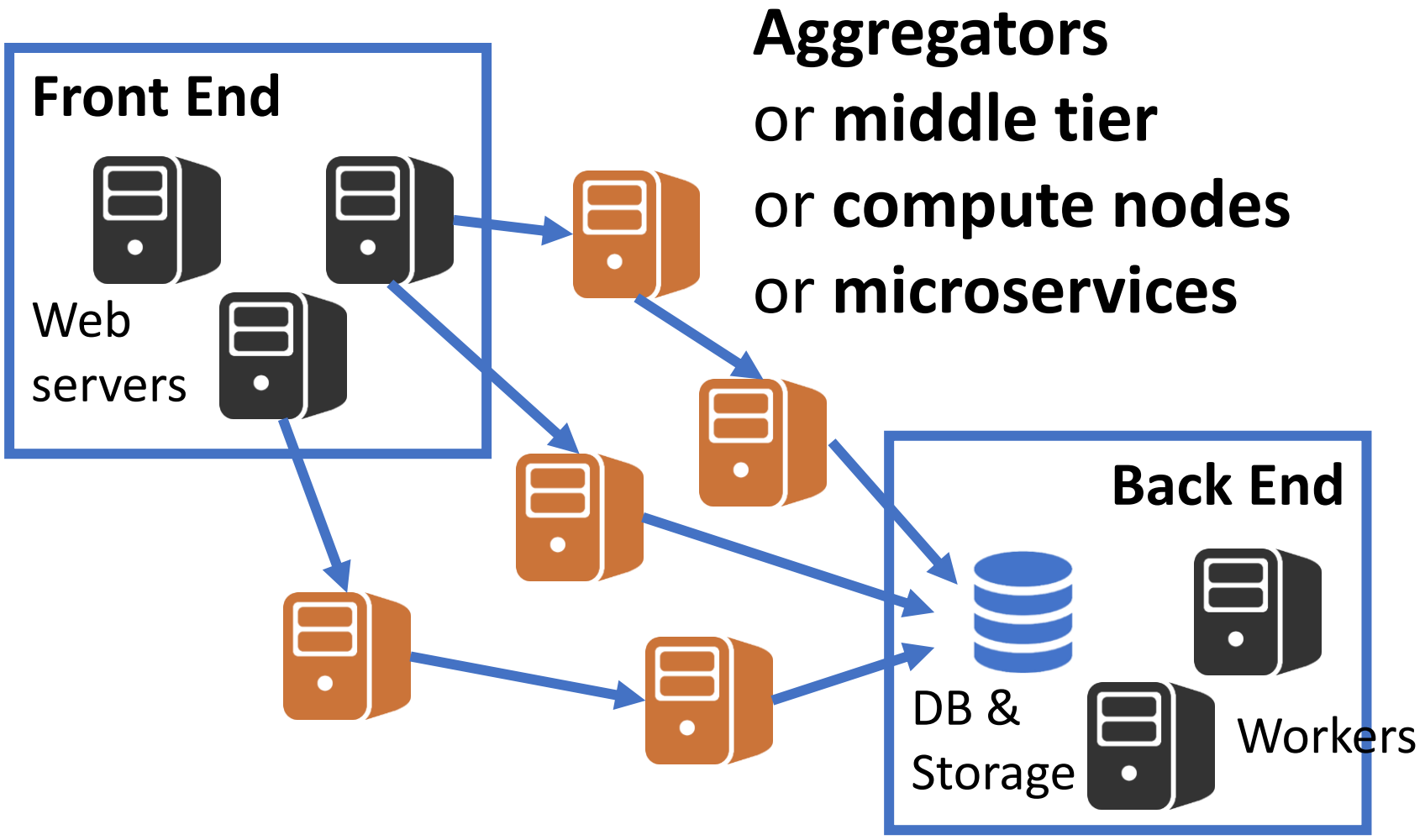


One Server
DRAM: 16 GB, 100 ns, 20 GB/s
Disk: 2T B, 10 ms, 200 MB/s
Flash: 128 GB, 100 us, 1 GB/s

Local Rack (80 servers)
DRAM: 1 TB, 300 us, 100 MB/s
Disk: 160 TB, 11 ms, 100 MB/s
Flash: 20 TB, 400 us, 100 MB/s

Cluster (30 racks)
DRAM: 30 TB, 500 us, 10 MB/s
Disk: 4.80 PB, 12 ms, 10 MB/s
Flash: 600 TB, 600 us, 10 MB/s

Multi-Tier Data Centers



Front-End Services

- Directly deal with data from/to users
- Latency matters a lot
- Oftentimes dealing with mobile apps, streaming service, telecommunication, etc

Middle-Tier Services

- Each node has a specific task
- Installed on commodity computers ➔ Highly elastic and easy to scale up
- Example:
  - Data processing: Hadoop, Spark
  - Caching: Redis, Memcached
  - Application servers: JBOSS, Tomcat

Back-end Services

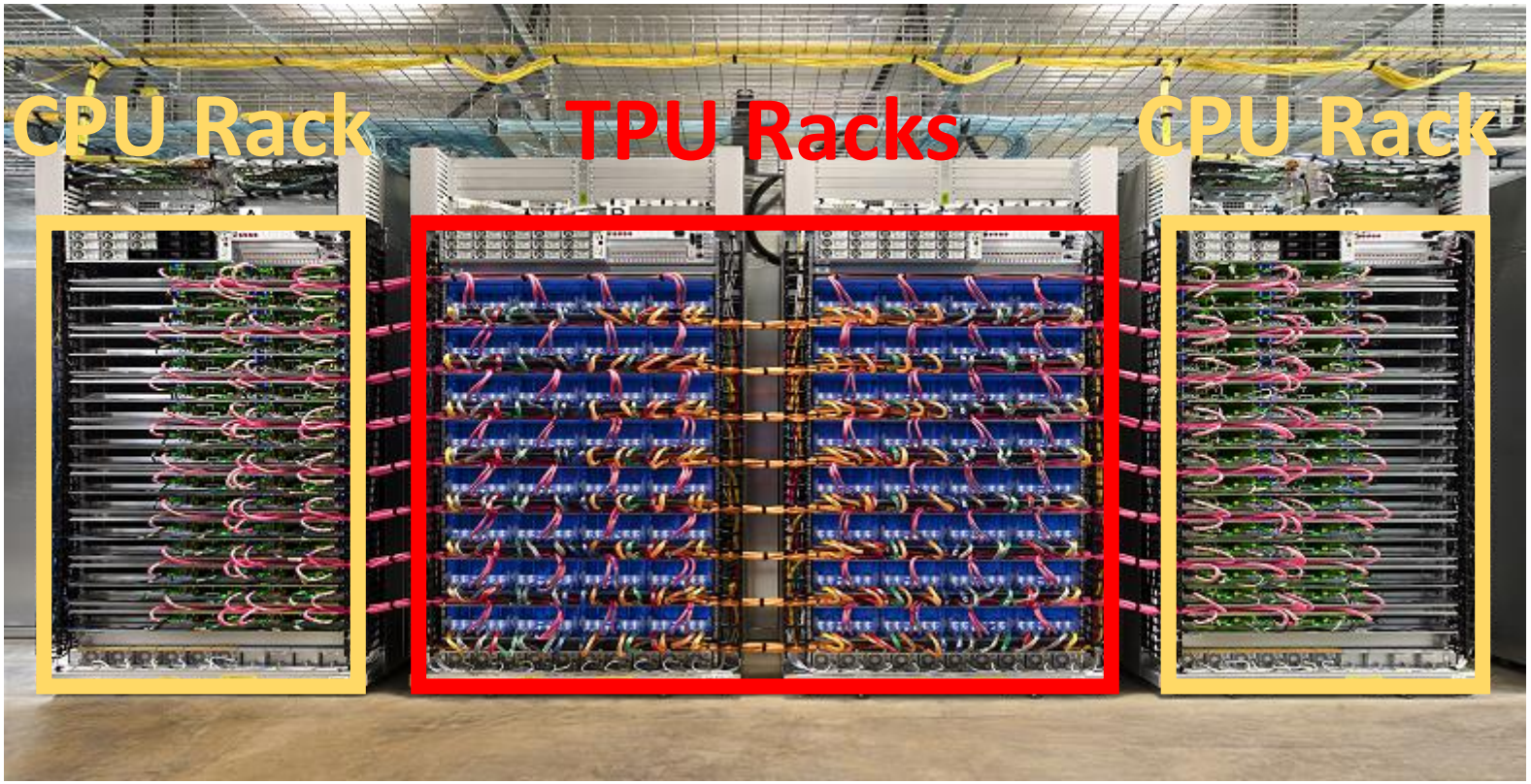
- Storage/DB or batch processing
- Sometimes are separately rented and connected to other cloud ➔ Ex: AWS S3 or EBS
- Example:
  - Sorting / filtering / indexing
  - Federated learning
  - Databases / key-value stores

Specialized Racks/Clusters

Cloud allocates homogeneous hardware in scale, but can still customize for racks/clusters.

- CPU racks/clusters (Compute Nodes)
- RAM racks/clusters
- Storage racks/clusters
- Other specialized hardware racks/clusters

Google TensorFlow Processing Units (TPUs)



1 TPU Pod = 64 TPUv2

= 11.52 Petaflops (11.52 x 10^{15} float-point ops per sec)

Why Datacenters?

- **Cost Efficiency (Economy of Scale):**
  - Reduce facility, management, power, innovation cost
- **Multi-tenancy:**
  - Accommodating multiple users in one infrastructure
- **Elasticity:**
  - Adaptive resource allocation for customers' need
  - Pay-per-use
- **Ease of management:**
  - Variety of cloud services and utilities
  - Fault/crash tolerance and disaster recovery

Types of Cloud Tenancy

- **Private Cloud:**
  - Datacenters run by organizations (e.g., banks, DoD)
  - Strong isolation but expensive to build
- **Public Cloud:**
  - AWS, Microsoft Azure, GCP, etc
  - Rented by public
- **Hybrid Cloud:**
  - Private cloud using public cloud as backend or backup
  - Private cloud hosted by public cloud

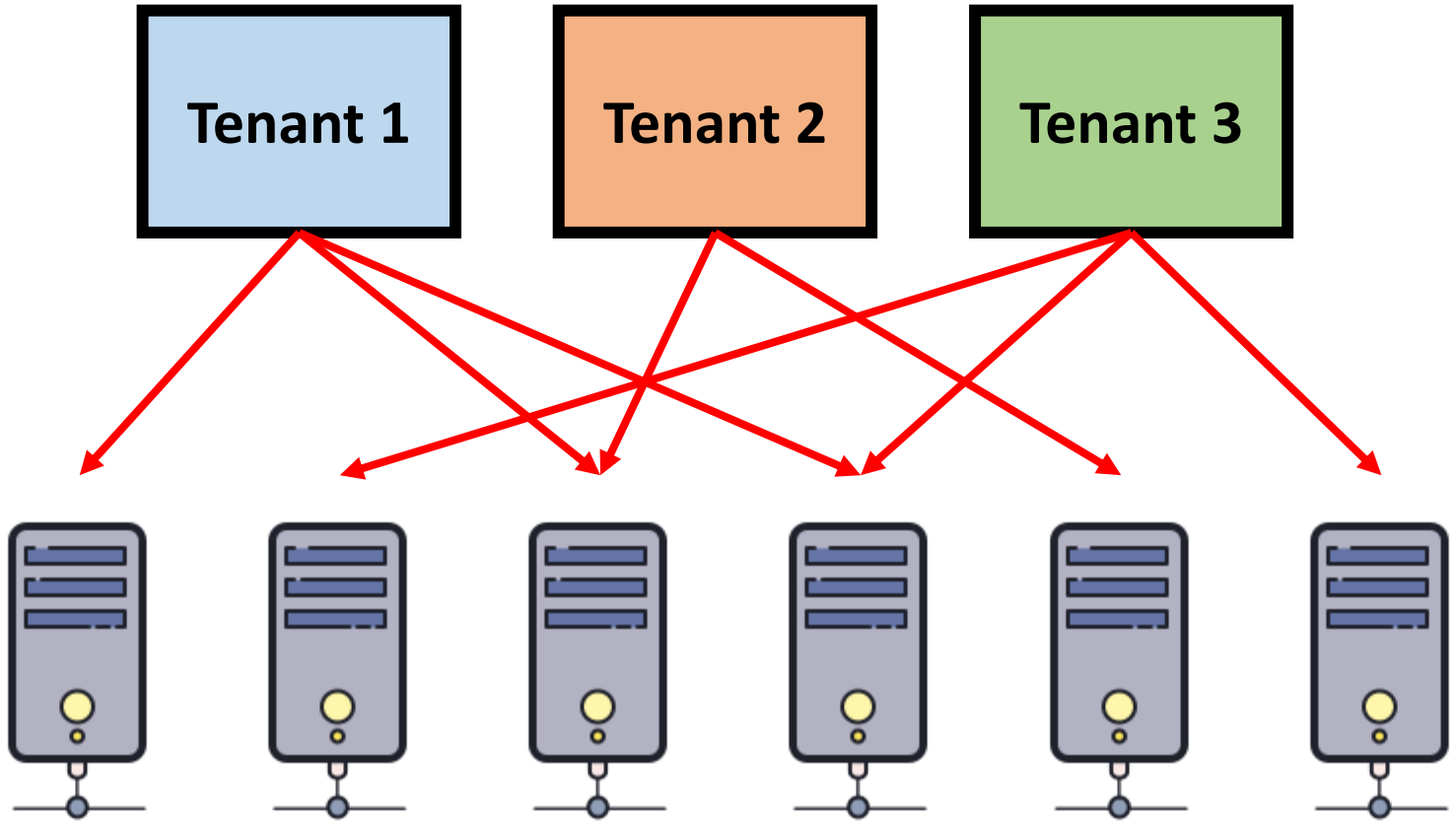
Public Cloud Platforms

Google Cloud Platform

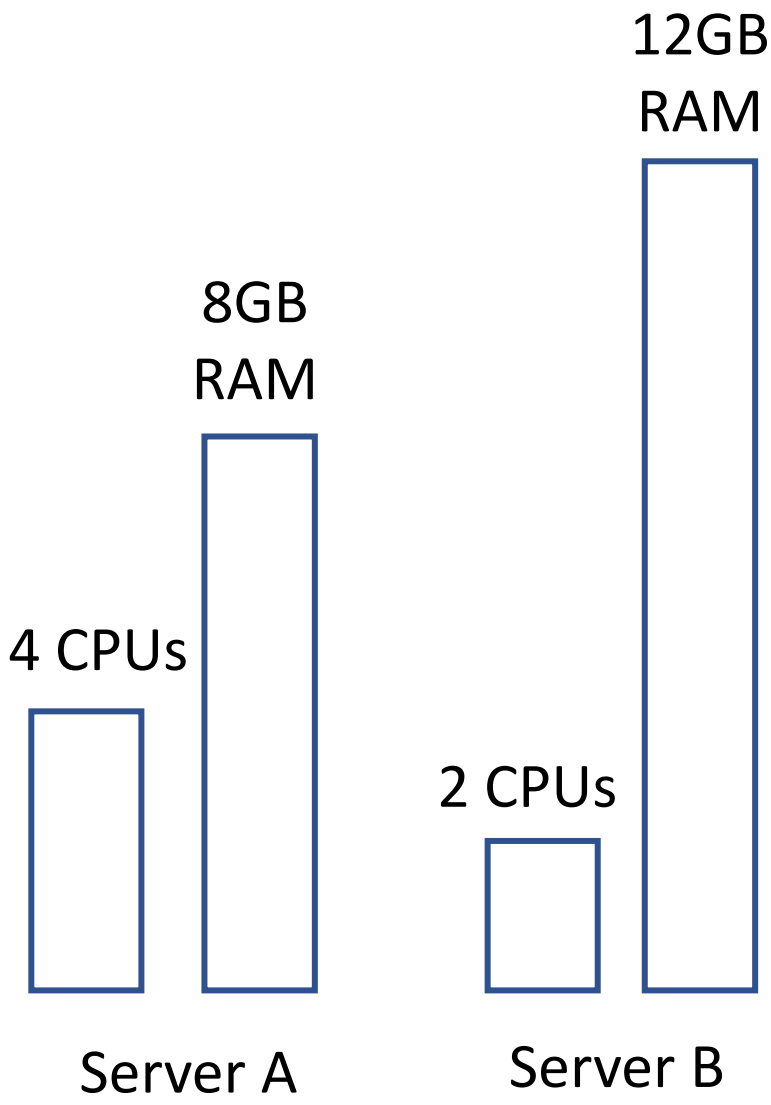
IBM Cloud

Alibaba Cloud

Multi-Tenant Service



Resource Allocation

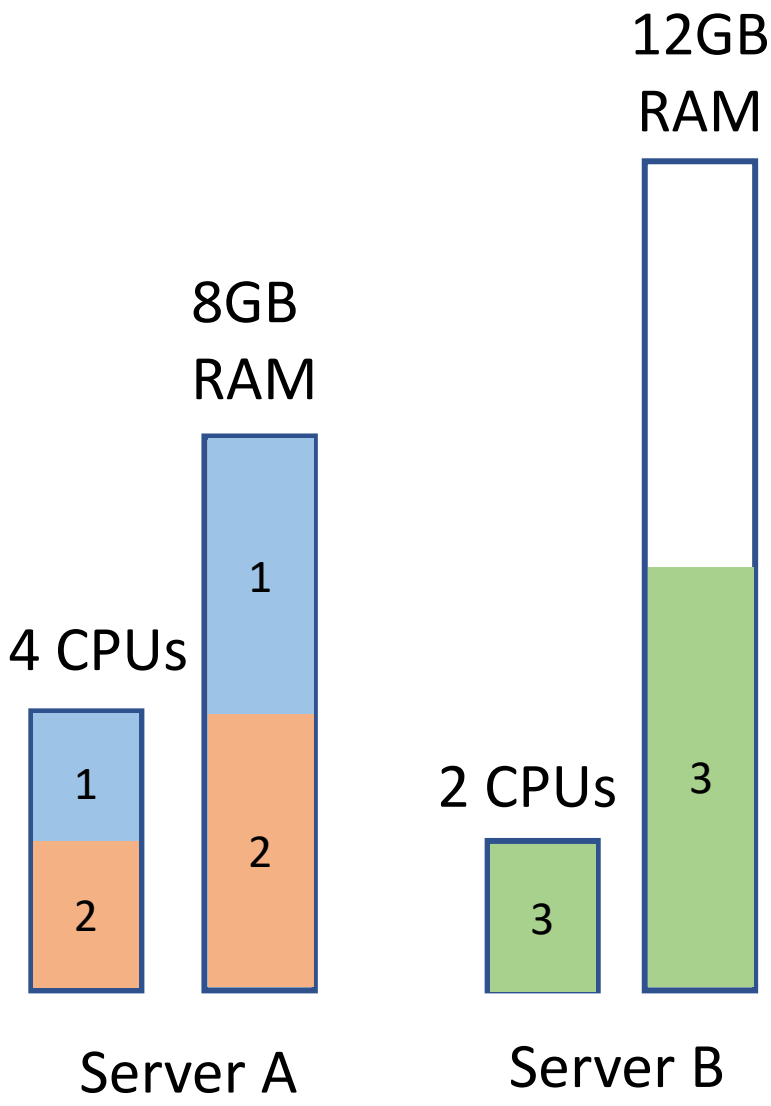


Tenant 1:
2 CPUs, 4GB RAM

Tenant 2:
2 CPUs, 4GB RAM

Tenant 3:
2 CPUs, 6GB RAM

Resource Allocation

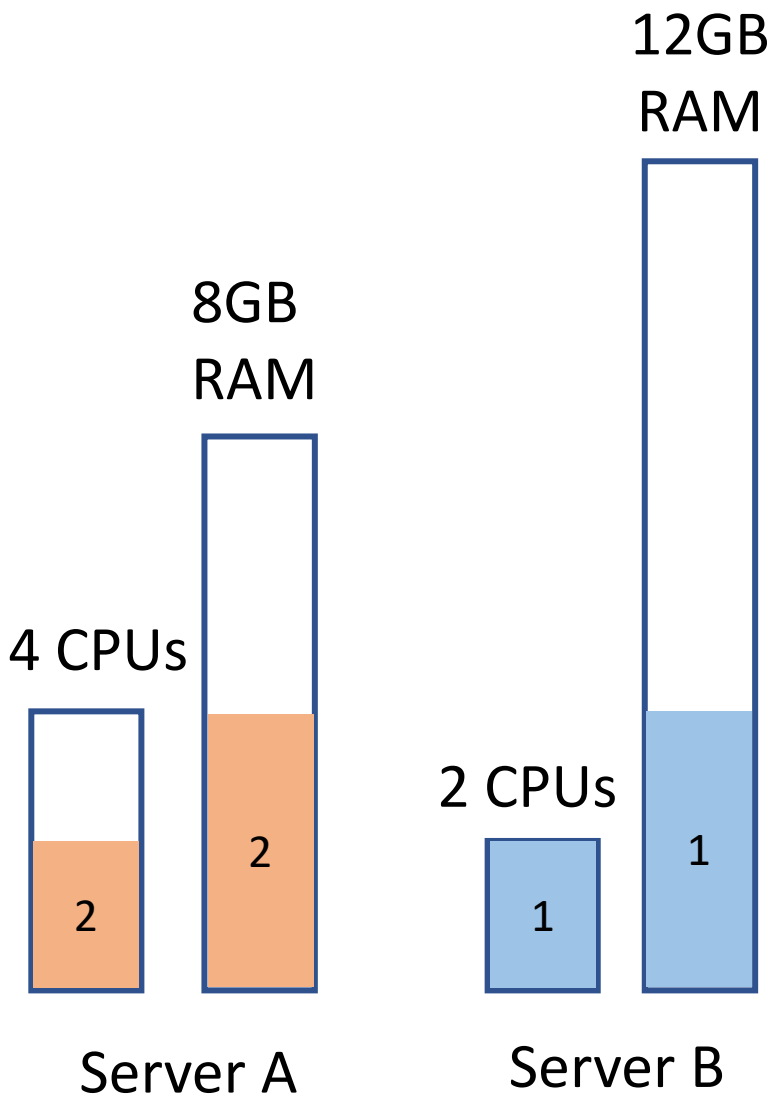


Tenant 1:
2 CPUs, 4GB RAM

Tenant 2:
2 CPUs, 4GB RAM

Tenant 3:
2 CPUs, 6GB RAM

Resource Allocation Failure



Tenant 1:
2 CPUs, 4GB RAM

Tenant 2:
2 CPUs, 4GB RAM

Tenant 3:
2 CPUs, 6GB RAM

Enough spared resources, but not on one server.

Service Level Agreement (SLA)

- A contract about expectation and responsibility of cloud providers and customers
- Usually comes with financial penalties

SLA, SLI, and SLO

- Defining SLA requires precise specification
  - Service Availability
  - Defect Rates
  - Technical Quality
  - Security
- **Service Level Indicators (SLIs):** Metrics that can be used to evaluate service levels
- **Service Level Objectives (SLOs):** Goals to maintain target service levels for a period of time

Examples of SLA

Availability %	Downtime per year	Downtime per month	Downtime per week
90% ("one nine")	36.5 days	72 hours	16.8 hours
95%	18.25 days	36 hours	8.4 hours
97%	10.96 days	21.6 hours	5.04 hours
98%	7.30 days	14.4 hours	3.36 hours
99% ("two nines")	3.65 days	7.20 hours	1.68 hours
99.50%	1.83 days	3.60 hours	50.4 minutes
99.80%	17.52 hours	86.23 minutes	20.16 minutes
99.9% ("three nines")	8.76 hours	43.8 minutes	10.1 minutes
99.95%	4.38 hours	21.56 minutes	5.04 minutes
99.99% ("four nines")	52.56 minutes	4.32 minutes	1.01 minutes
100.00%	26.28 minutes	2.16 minutes	30.24 seconds
99.999% ("five nines")	5.26 minutes	25.9 seconds	6.05 seconds
99.9999% ("six nines")	31.5 seconds	2.59 seconds	0.605 seconds
99.99999% ("seven nines")	3.15 seconds	0.259 seconds	0.0605 second

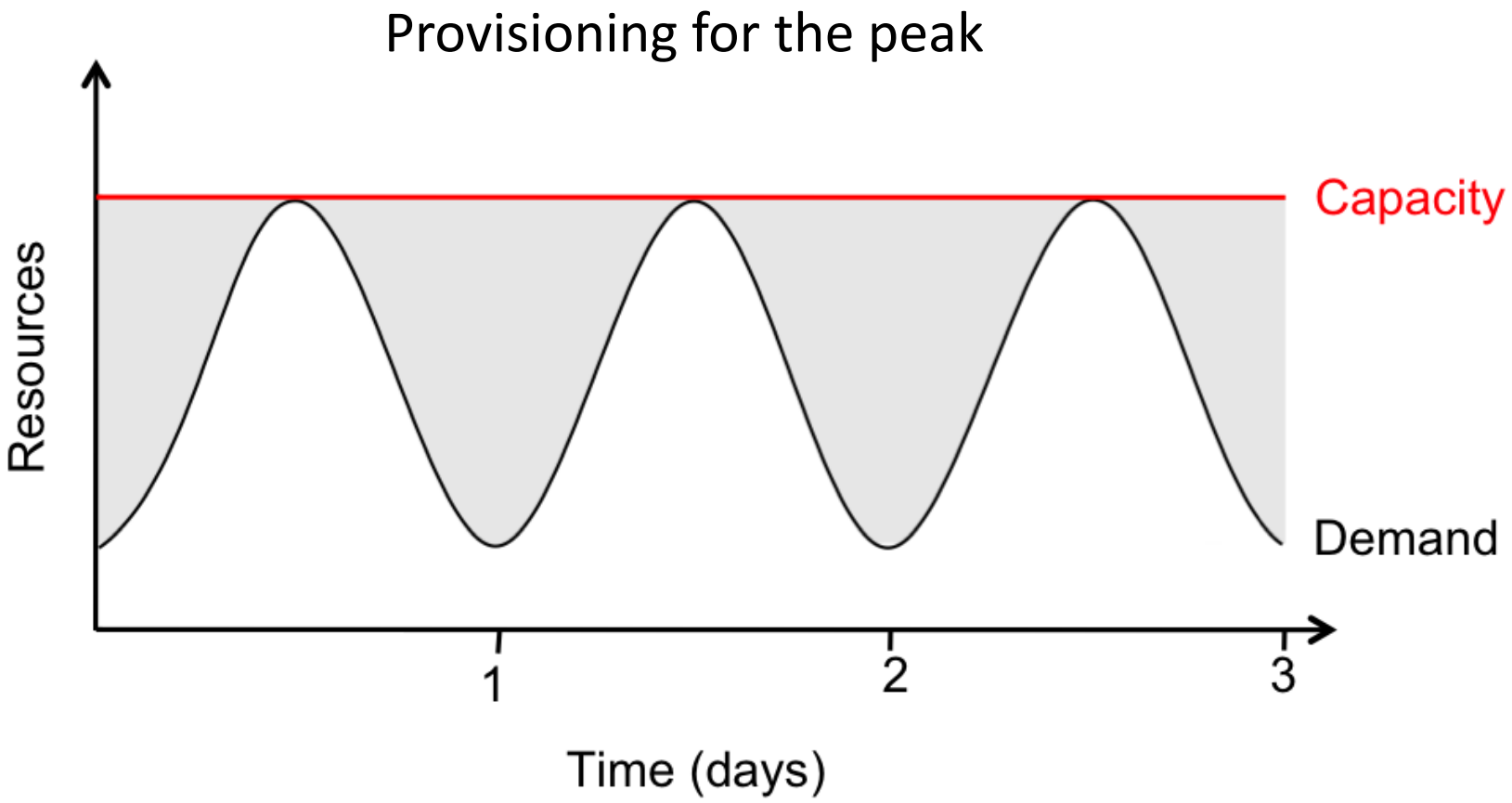
SLA Decides Everything

- Cloud providers make decisions based on SLA
  - Resource Allocation
  - Disaster & failure recovery
  - … and a lot more

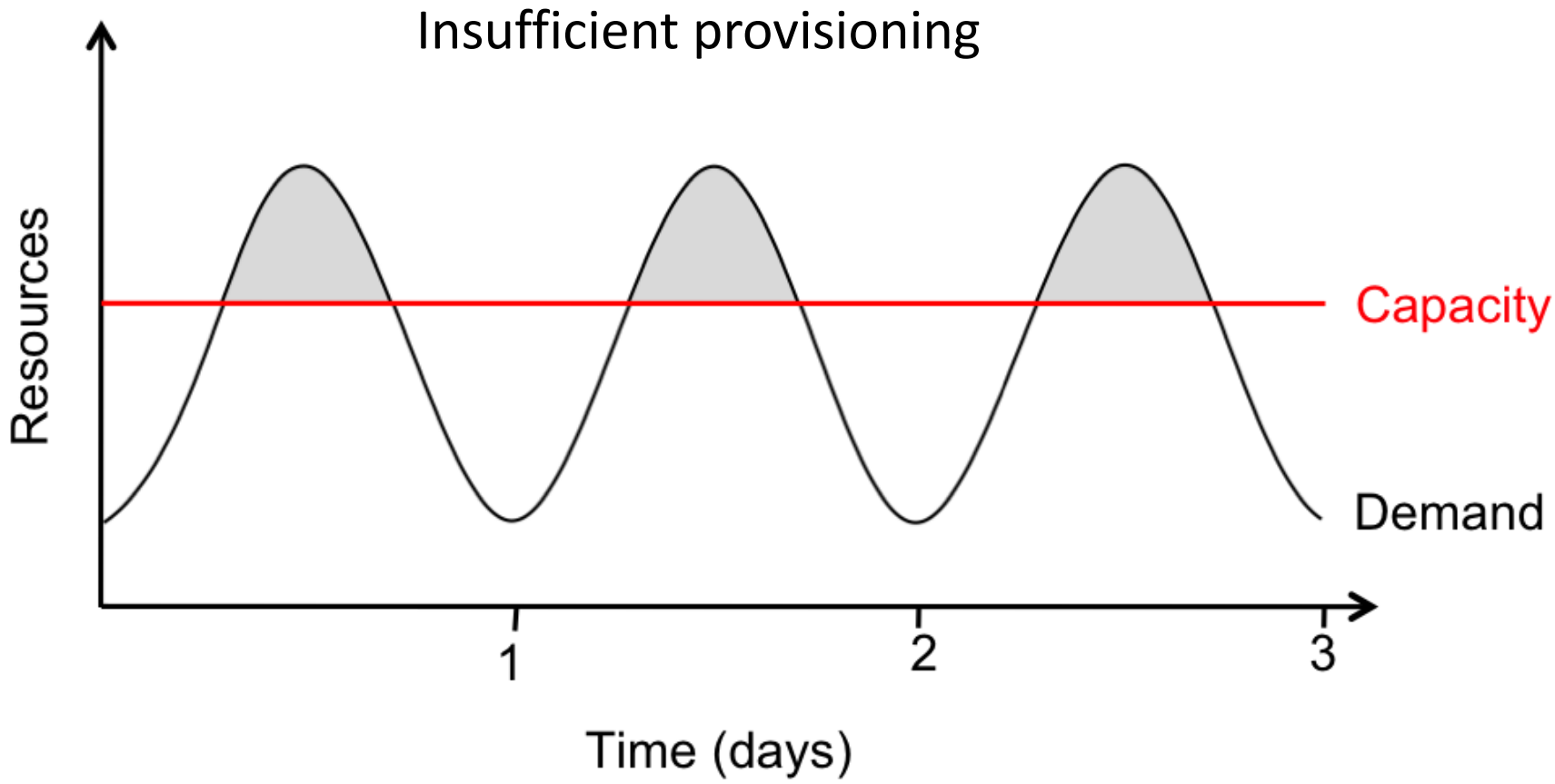
Why Datacenters?

- **Cost Efficiency (Economy of Scale):**
  - Reduce facility, management, power, innovation cost
- **Multi-tenancy:**
  - Accommodating multiple users in one infrastructure
- **Elasticity:**
  - Adaptive resource allocation for customers' need
  - Pay-per-use
- **Ease of management:**
  - Variety of cloud services and utilities
  - Fault/crash tolerance and disaster recovery

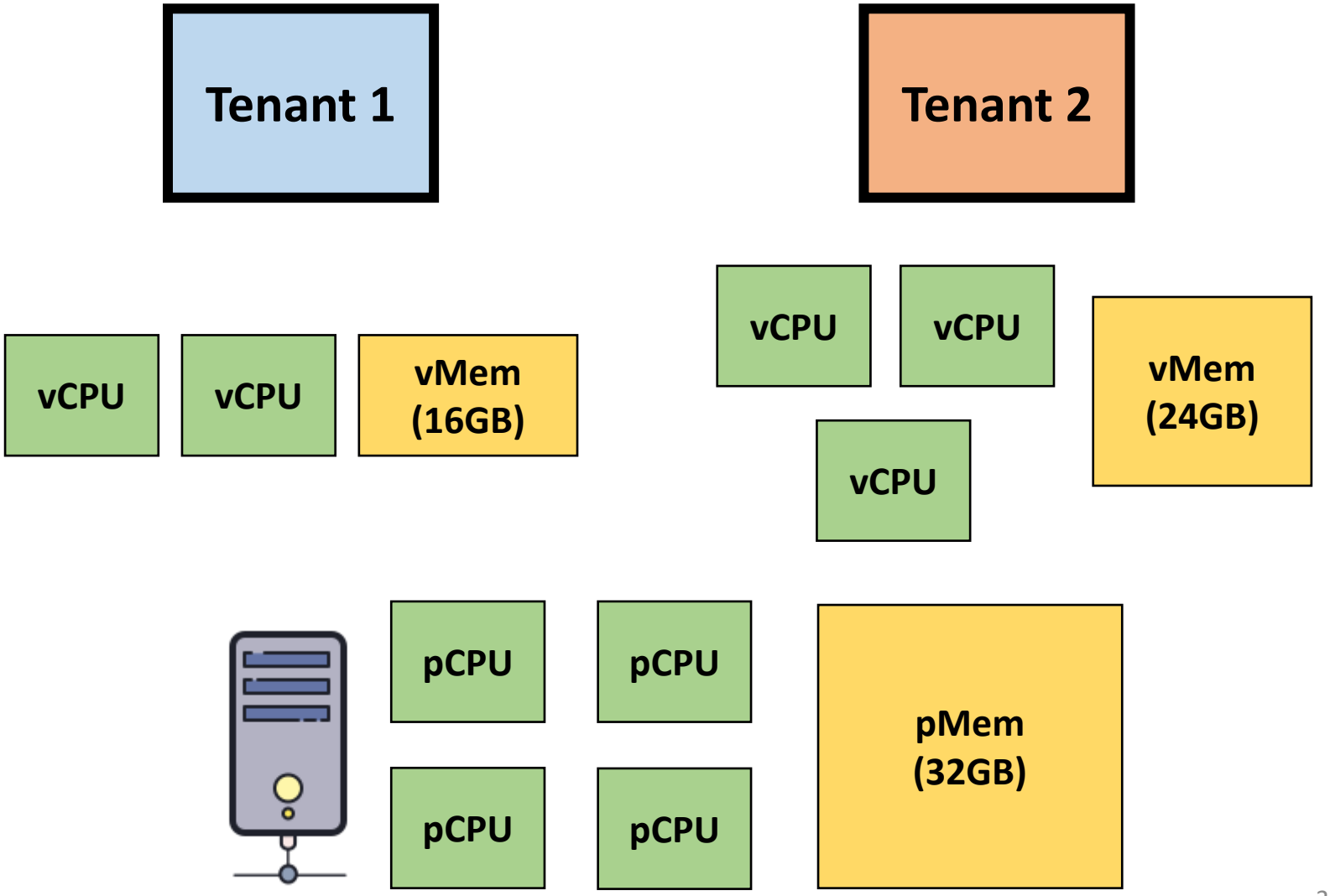
Challenges of Provisioning



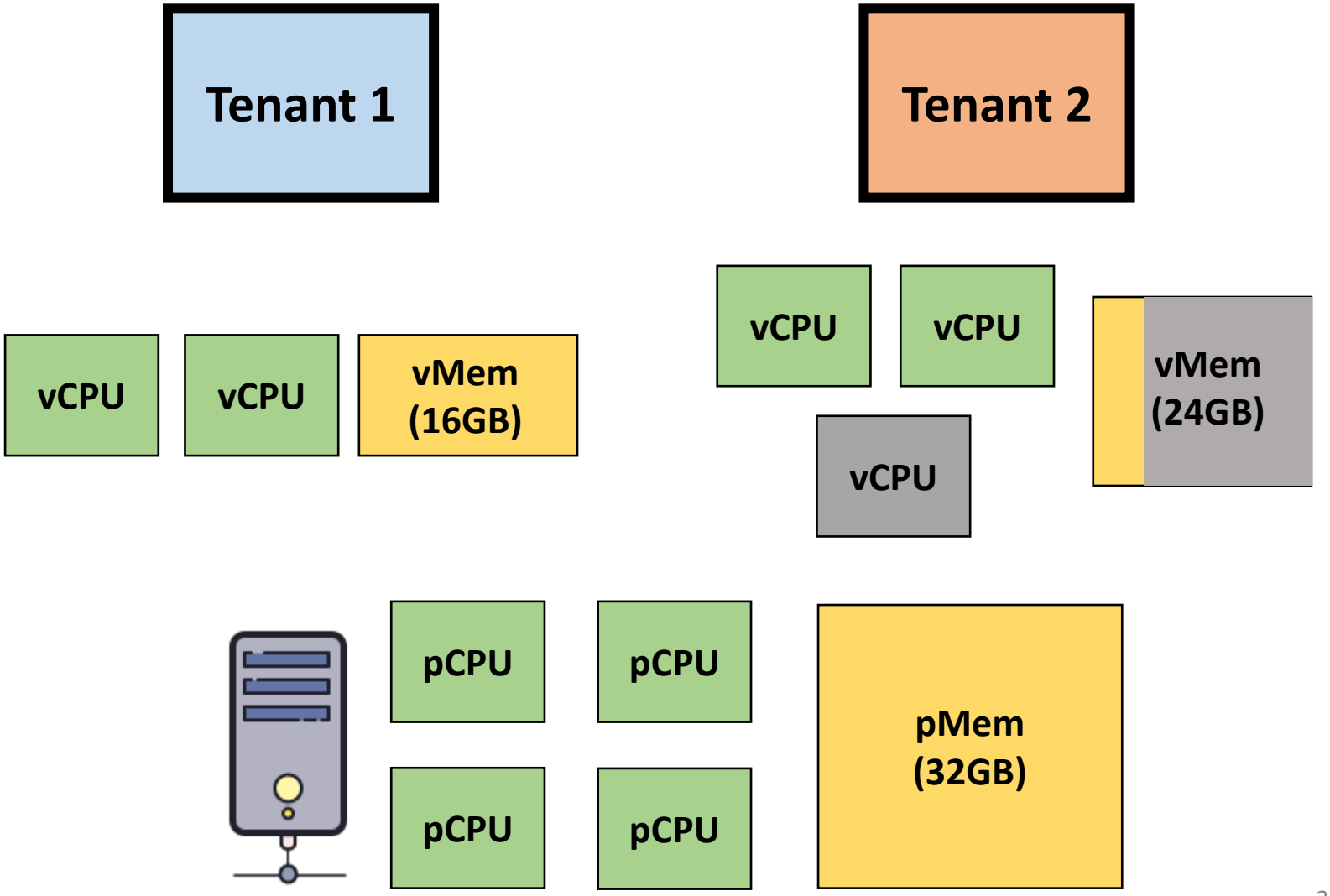
Challenges of Provisioning



Physical vs Virtual Resources



Overprovisioning via Virtualization



Overprovisioning vs Reserved

- Virtualization Technology allows overprovisioning
  - Offering more resources than you have
  - Example: 8GB RAM shared by 4 tenants who want 4GB
  - Assuming they won't use 4GB simultaneously
- Problem: cannot guarantee availability and performance isolation
  - Vendors are worried about violating SLA
  - Out of memory will cause thrashing
  - **Generally, vendors just reserve resources for customers**

Resource Disaggregation

- Break down a workload into smaller ones to make allocation easier
- 2 CPU, 6GB RAM → (1 CPU, 3GB RAM) * 2
- Occupation time also matters
  - Short occupation → easier for time-sharing
- Example: Function-as-a-Service
  - Proportional DRAM (128/192/…/3008 MBs) and CPUs
  - Time limit (10 mins)

Why Datacenters?

- **Cost Efficiency (Economy of Scale):**
  - Reduce facility, management, power, innovation cost
- **Multi-tenancy:**
  - Accommodating multiple users in one infrastructure
- **Elasticity:**
  - Adaptive resource allocation for customers' need
  - Pay-per-use
- **Ease of management:**
  - Variety of cloud services and utilities
  - Fault/crash tolerance and disaster recovery

Types of Cloud Services

- Software as a Service (SaaS)
  - Vendors sell licensed cloud software to customers
  - Naturally multi-tenant and scalable
  - Example: Google Doc, Office 365, Saleforces.com
- Platform as a service (PaaS)
  - Vendors/Providers provide development platforms
  - Rapid deployment
  - Example: Google AppEngine

Types of Cloud Services

- Infrastructure as a Service (IaaS)
  - Providers provision computing resources
    - vCPU, memory, network, disks, etc
  - Customers is provided with virtual machine instances
    - a1.large, t3.medium, t2.micro, etc
  - Customers have control over OS, storage, system configuration, etc
  - Example: AWS EC2

Other New Service Models (I)

- Function as a Service (FaaS)
  - Developers deploy functions, not VMs
  - Event-driven, pay-per-use
  - Spread out execution to 5000+ cores in < 1 sec
- Backend as a Service (BaaS)
  - Connect web/mobile apps with APIs or SDKs
  - Example: connect cloud services (DB, LDAP, etc) to a unified REST API for mobiles

FaaS + BaaS = Serverless Computing

Other New Service Models (II)

- Database as a Service (DBaaS)
  - Provide users access to a relational or NoSQL database
- Big Data as a Service (BDaaS)
  - Programmable analytic platform for users (i.e., PaaS)
- Security as a Service (SECaaS)
  - Security primitives (single sign-on, antivirus, intrusion detection, etc) for corporates

Large Disasters in Datacenters

Apr 20, 2014
Samsung SDS datacenter burning during a fire, causing disruption in global data services

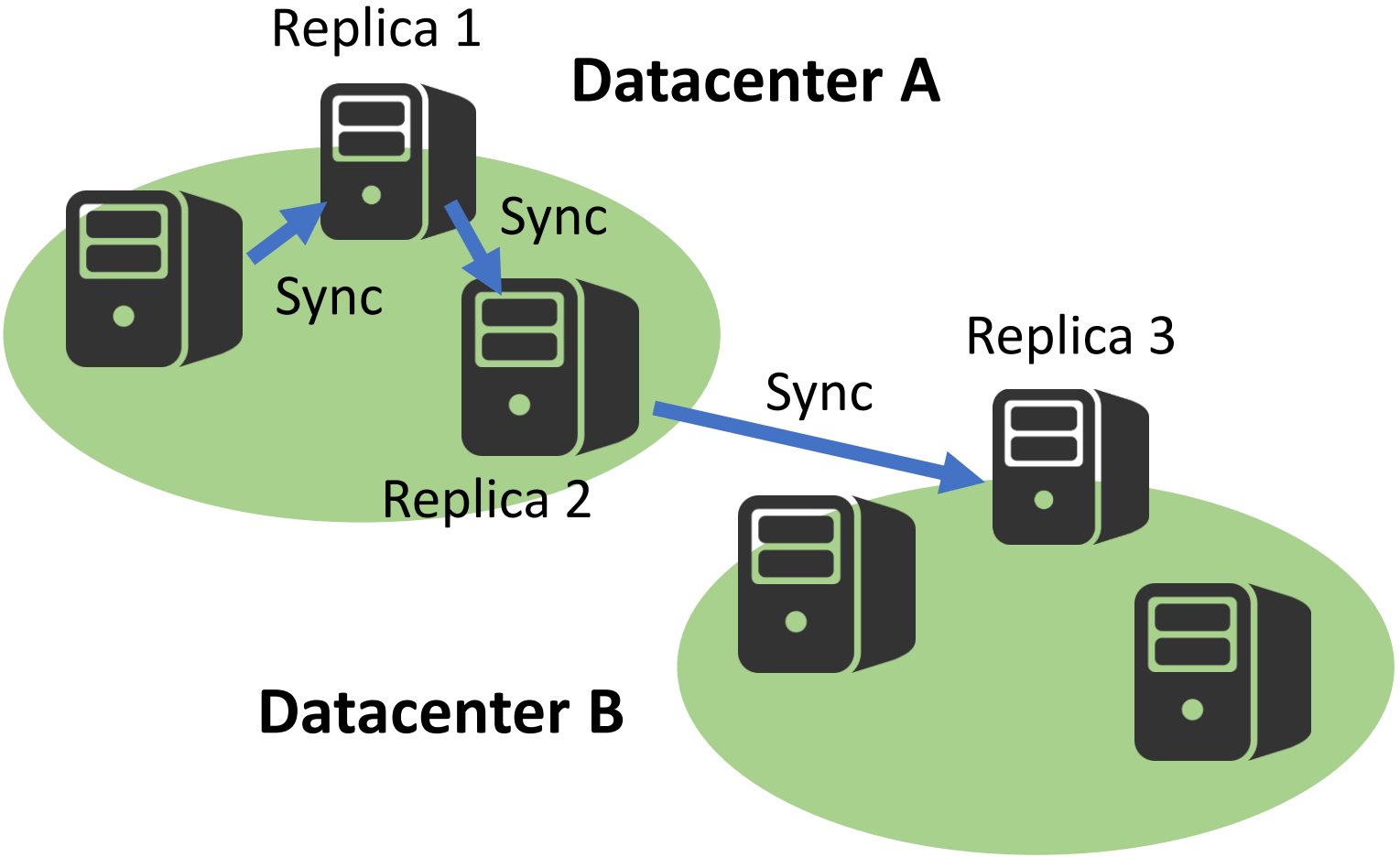
Dealing with Failures

- Even a small server room has failures
- A warehouse-size datacenter faces failures daily; **cannot be prevented at hardware level.**
- Can only handle at **software** & **management** level

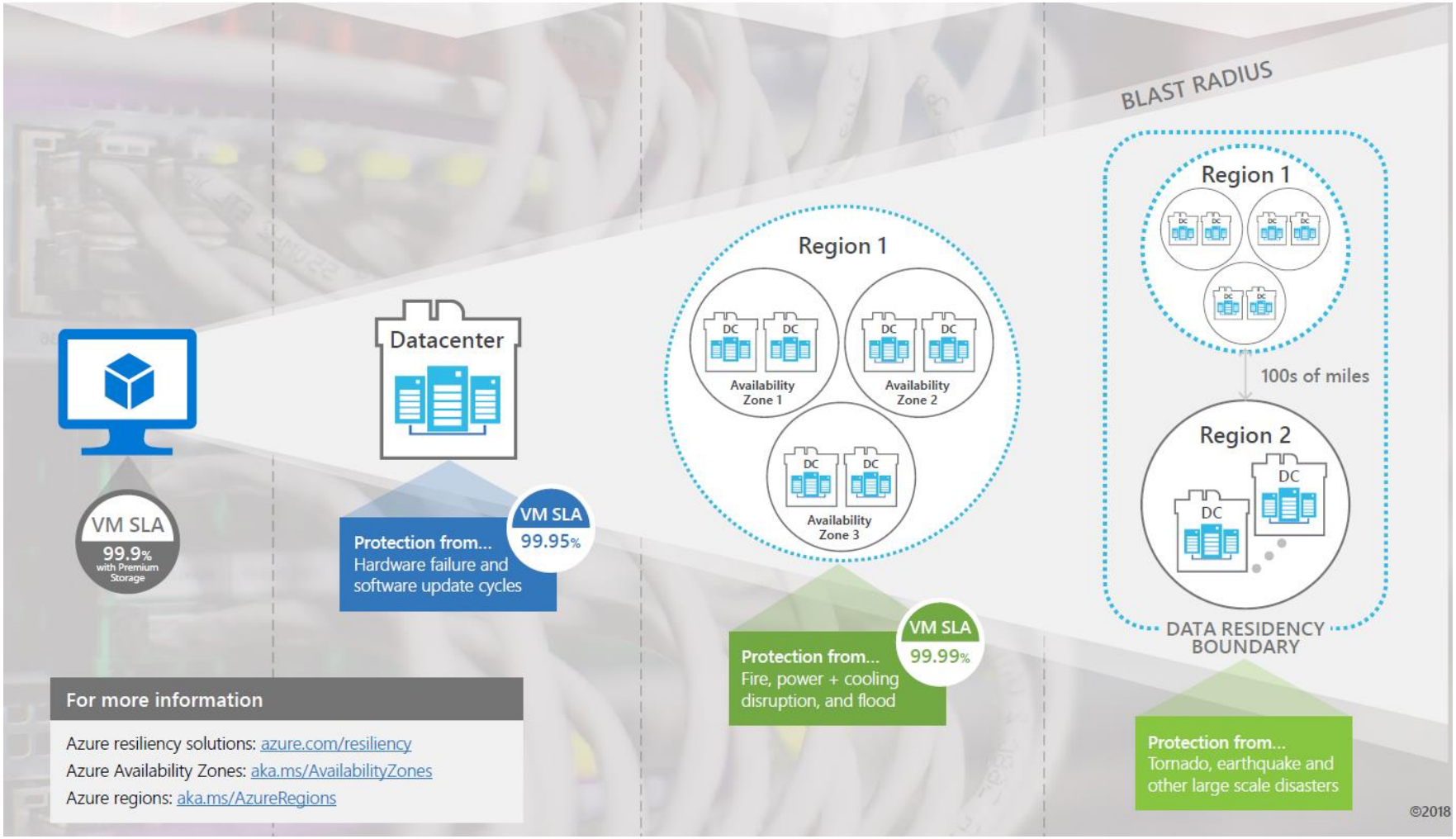
Failure Rates of Datacenters

- According to Google fellow Jeff Dean (2008), in the first year of a datacenter:
  - Typically 1,000 machine failures
  - 500-1,000 machines are down for ~6 hours
  - 20 racks with 40-80 machines vanish from network
  - 5 racks got half of the packets missing
  - 50% chance of overheating in the whole cluster, taking down all servers in 5 mins and taking 1-2 days to recover

Replication



Availability Zones



References

- “The Datacenter as a Computer – An Introduction to the Design of Warehouse-Scale Machines”, 2nd Edition, by Barroso, Clidaras, and Hölzle
- Course material: “Datacenter Fundamentals: The Datacenter as a Computer”, by George Porter, UCSD
- “What is Resiliency in Azure?” (https://azure.microsoft.com/mediahandler/files/resourcefiles/azure-resiliency-infographic/Azure_resiliency_infographic.pdf)