CLOUD COMPUTING
Cloud Models

Jannatun Noor
BRAC University
Jannatun.noor@bracu.ac.bd

Cloud Computing :
Two different Model

- Cloud computing refers to ***applications and services*** that run on a ***distributed network*** using ***virtualized resources*** and accessed by common Internet protocols and networking standards. It is distinguished by the notion that *resources are* ***virtual*** *and limitless* and that *details of the physical systems on which software runs are* ***abstracted*** from the user.

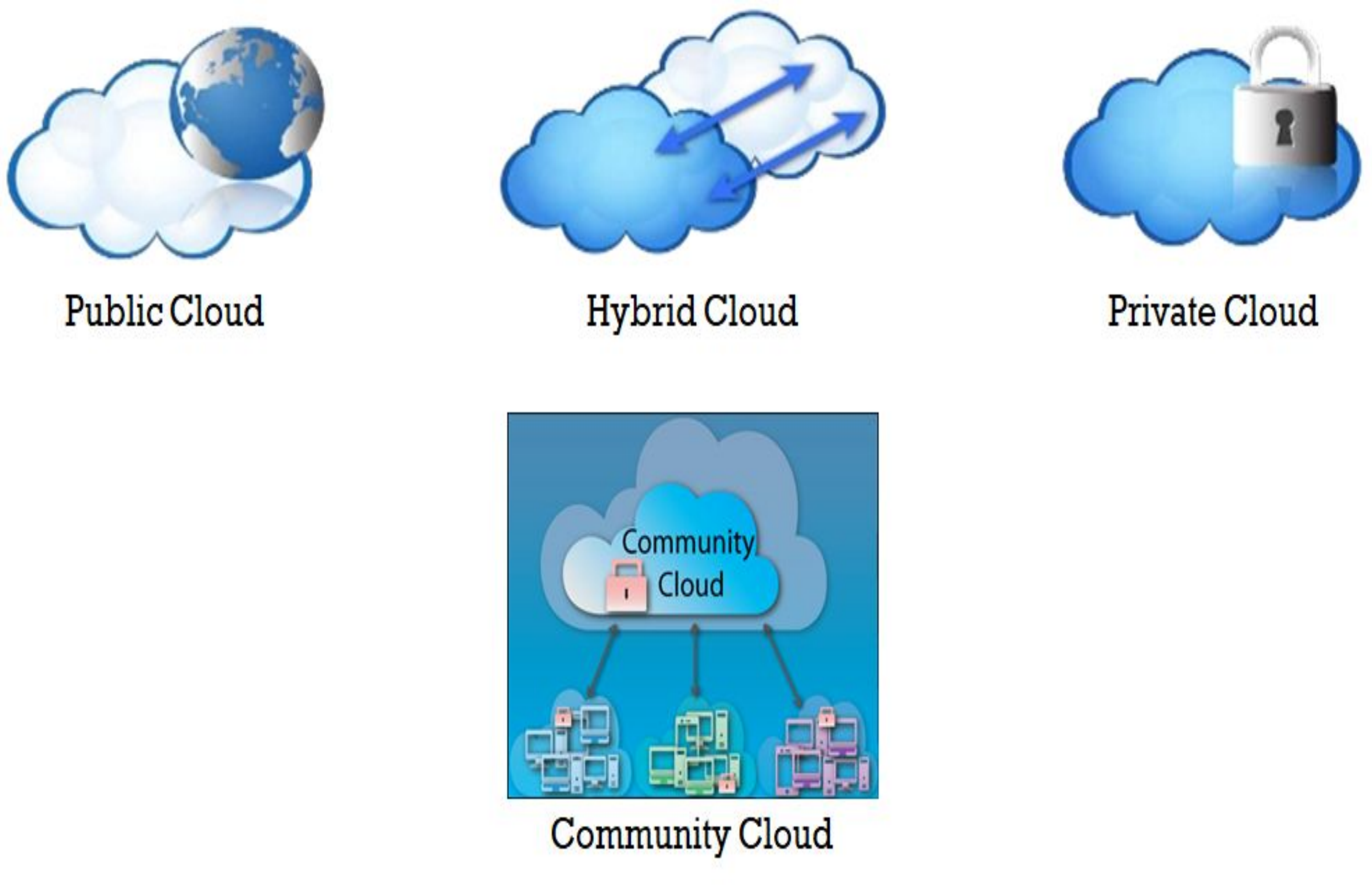
Cloud Computing : Two different Model (cont.)

- Two Different classes of clouds:
  - ***Deployment model*** tells where the cloud is located and for what purpose. Public, private, community, and hybrid clouds are deployment models.
  - ***Service model*** describes the type of service that the service provider is offering. Software as a Service, Platform as a Service, and Infrastructure as a Service—**the SPI model**.

Cloud Computing : Two different Model (cont.)

- Cloud computing makes the long-held dream of utility computing possible with a pay-as-you-go, infinitely scalable, universally available system. With cloud computing, ***you can start very small and become big very fast.***

Deployment Models

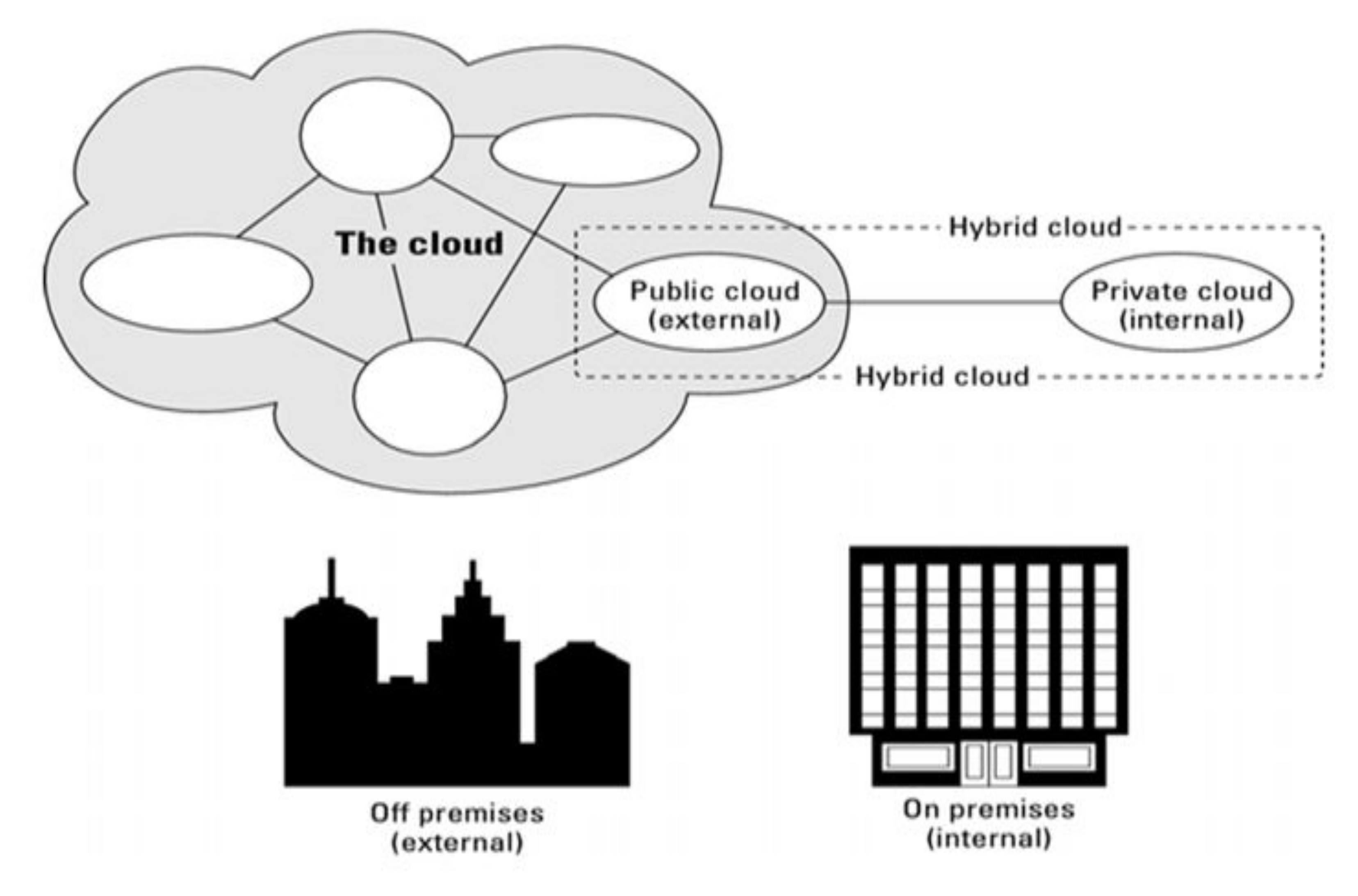


Cloud Computing :

Deployment Models

- Public clouds provide service to any paying customer:
  - Amazon S3: store arbitrary datasets, pay per GB-month stored
  - Amazon EC2: upload and run arbitrary OS images, pay per CPU hour used
  - Google App Engine / Compute Engine: develop applications within their App Engine framework, upload data that will be imported into their format, and run

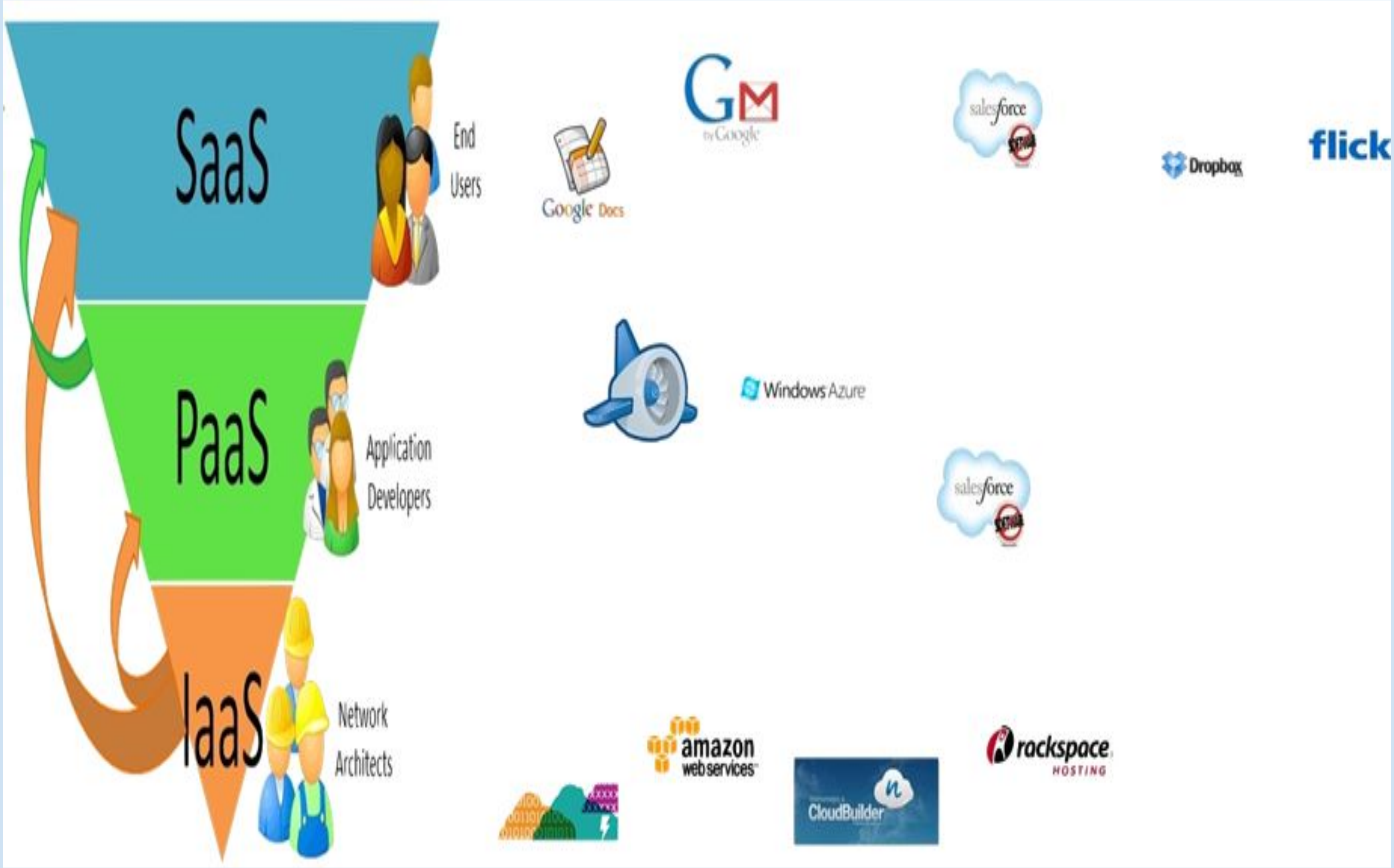
Deployment Models



Community Cloud

A **community cloud** in computing is a collaborative effort in which infrastructure is shared between several organizations from a specific **community** with common concerns (security, compliance, jurisdiction, etc.), whether managed internally or by a third-party and hosted internally or externally.

Service Models



Service Models: IAAS

- **Infrastructure as a Service**: IaaS provides virtual machines, virtual storage, virtual infrastructure, and other hardware assets as resources that clients can provision.
  - The IaaS service provider manages all the infrastructure, while the client is responsible for all other aspects of deployment such as operating system, applications, and user interactions with the system.

Service Models: PAAS

- **Platform as a Service**: PaaS provides virtual machines, operating systems, applications, services, development frameworks, transactions, and control structures.
    - The service provider manages the cloud infrastructure, the operating systems, and the enabling software. The client can deploy its applications on the cloud infrastructure or use applications that were programmed using languages and tools supported by PaaS service provider. Thus, the client is mere responsible for installing and managing the deployed application.

Service Models: SAAS

- **Software as a Service**: SaaS is a complete operating environment with applications, management, and the user interface.
  - In SaaS model, the application is provided to the client through a thin client interface (a browser, usually), and the customer's responsibility begins and ends with ***entering and managing its data and user interaction***. Everything from the application down to the infrastructure is the vendor's responsibility.

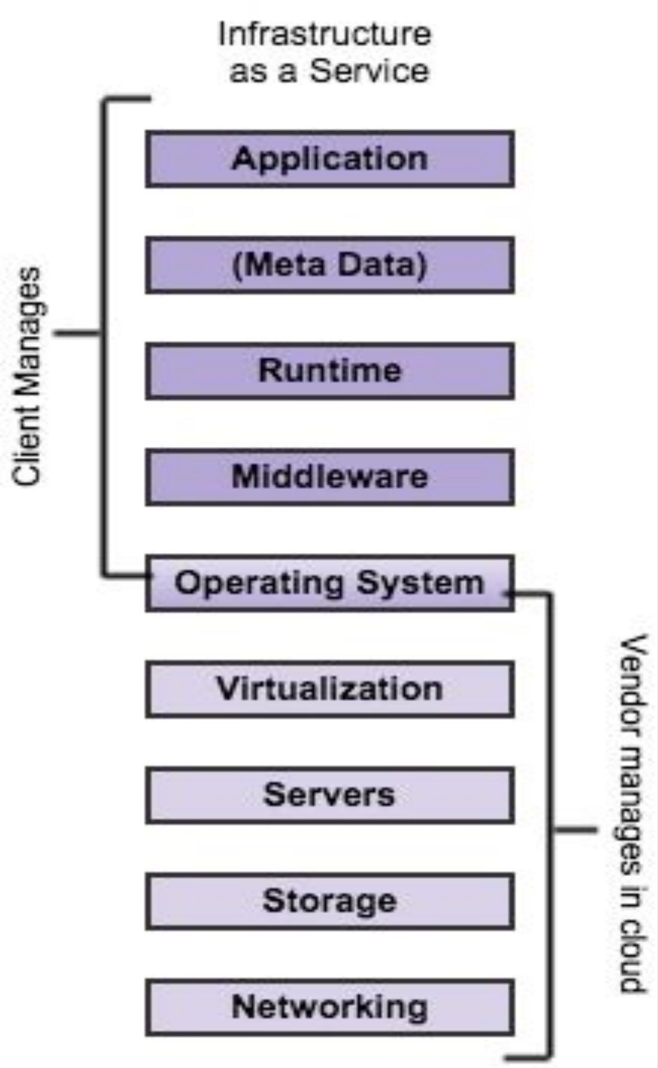
Service Models

- The three different service models taken together have come to be known as the **SPI model of cloud computing**. Many other service models have been mentioned: StaaS, Storage as a Service; IdaaS, Identity as a Service; CmaaS, Compliance as a Service; and so forth. **However, the SPI services encompass all the other possibilities.**

Cloud Computing :

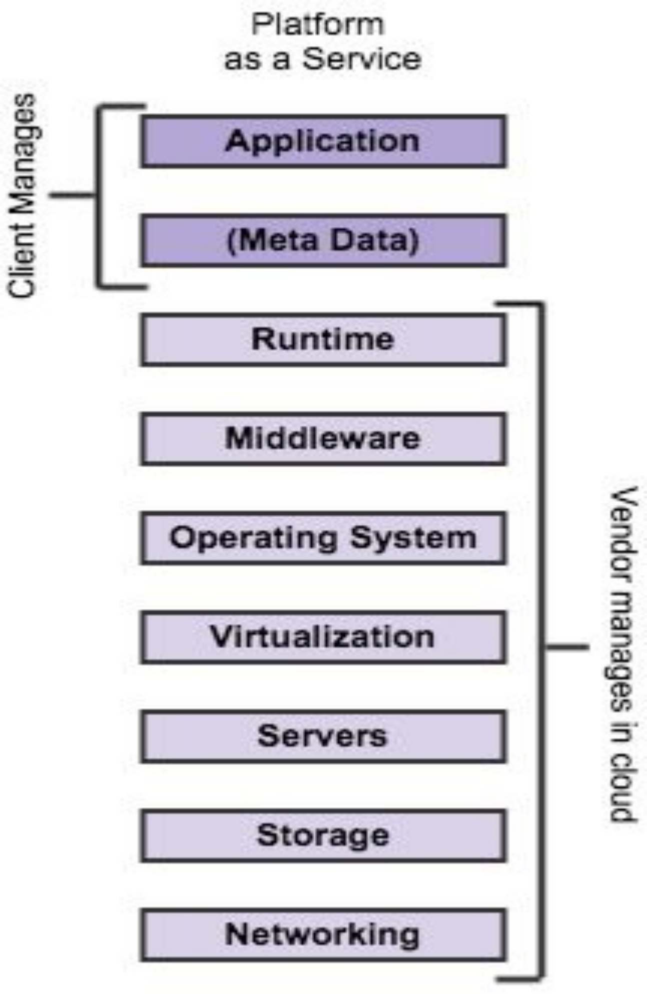
Service Models

- **Infrastructure as a Service (IaaS)** solution should include vendor-managed network, storage, servers, and virtualization layers for a client to run their application and data on.
    - **Examples:** Amazon Web Services (AWS), Microsoft Azure, Google Compute Engine (GCE), Joyent
    - **Use-Case:** Extends current data center infrastructure for temporary workloads (e.g. increased Christmas holiday site traffic)



Cloud Computing :

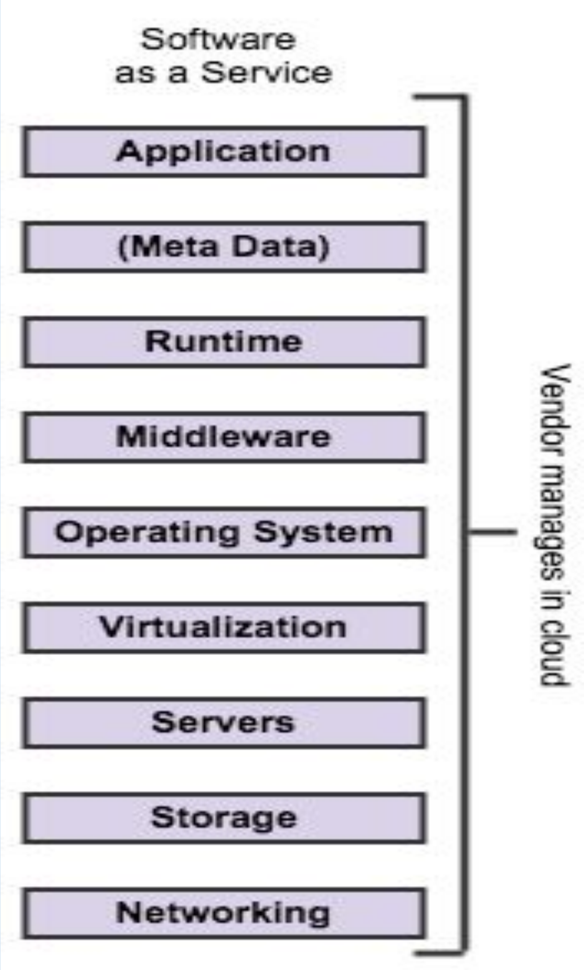
Service Models



- **Platform as a Service (PaaS)** builds on top of infrastructure as a service adding vendor-managed middleware such as web, application, and database software.
  - **Examples:** Apprenda
  - **Use-Case:** Increases developer productivity and utilization rates while also decreasing an application’s time-to-market

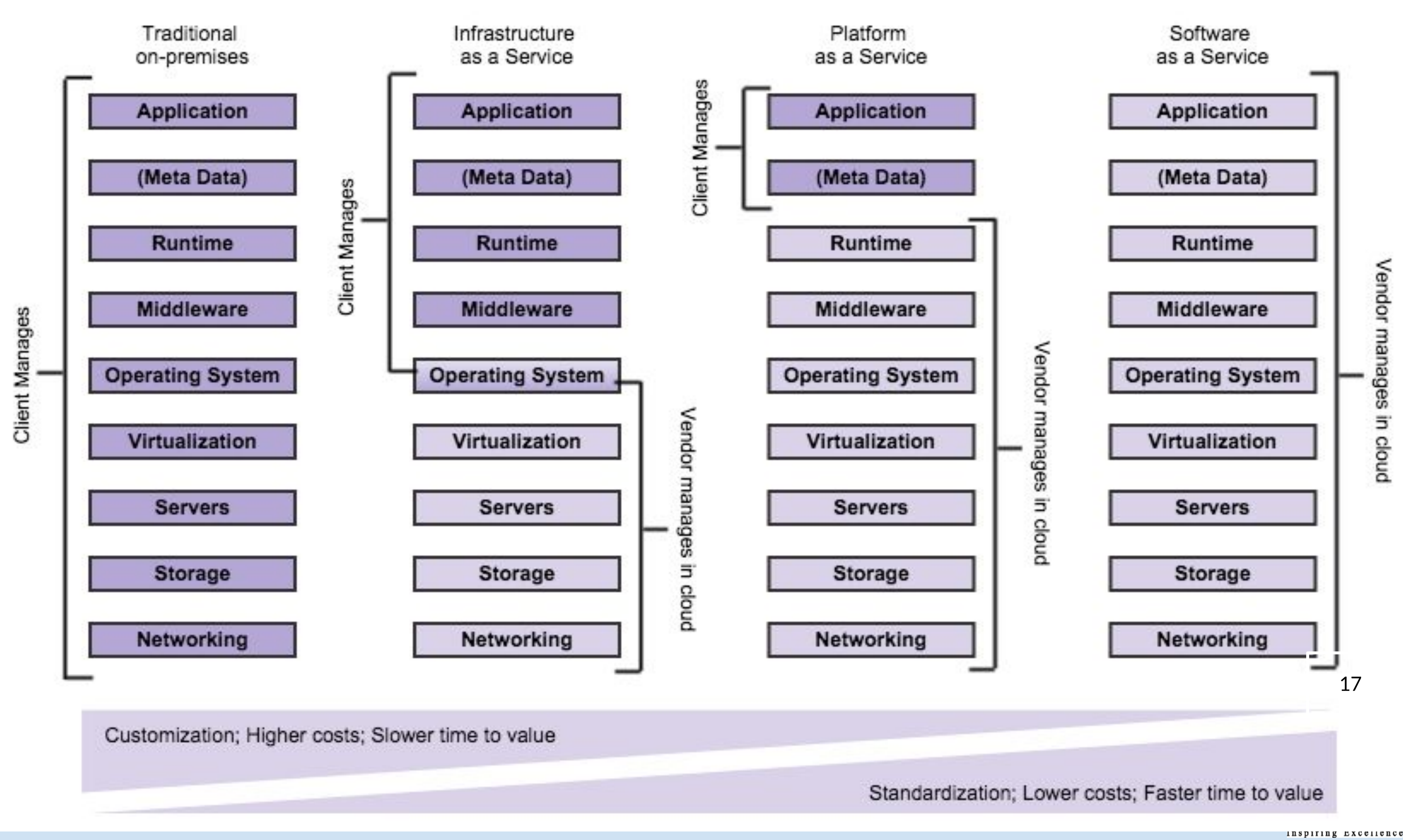
Cloud Computing :
Service Models contd.

- **Software as a Service (SaaS)** again builds on top of that, most of the time adding applications that implement specific user functionality such as email, CRM, or HRM.
    - **Examples:** Google Apps, Salesforce, Workday, Concur, Citrix GoToMeeting, Cisco WebEx
    - **Use-Case:** Replaces traditional on-device software



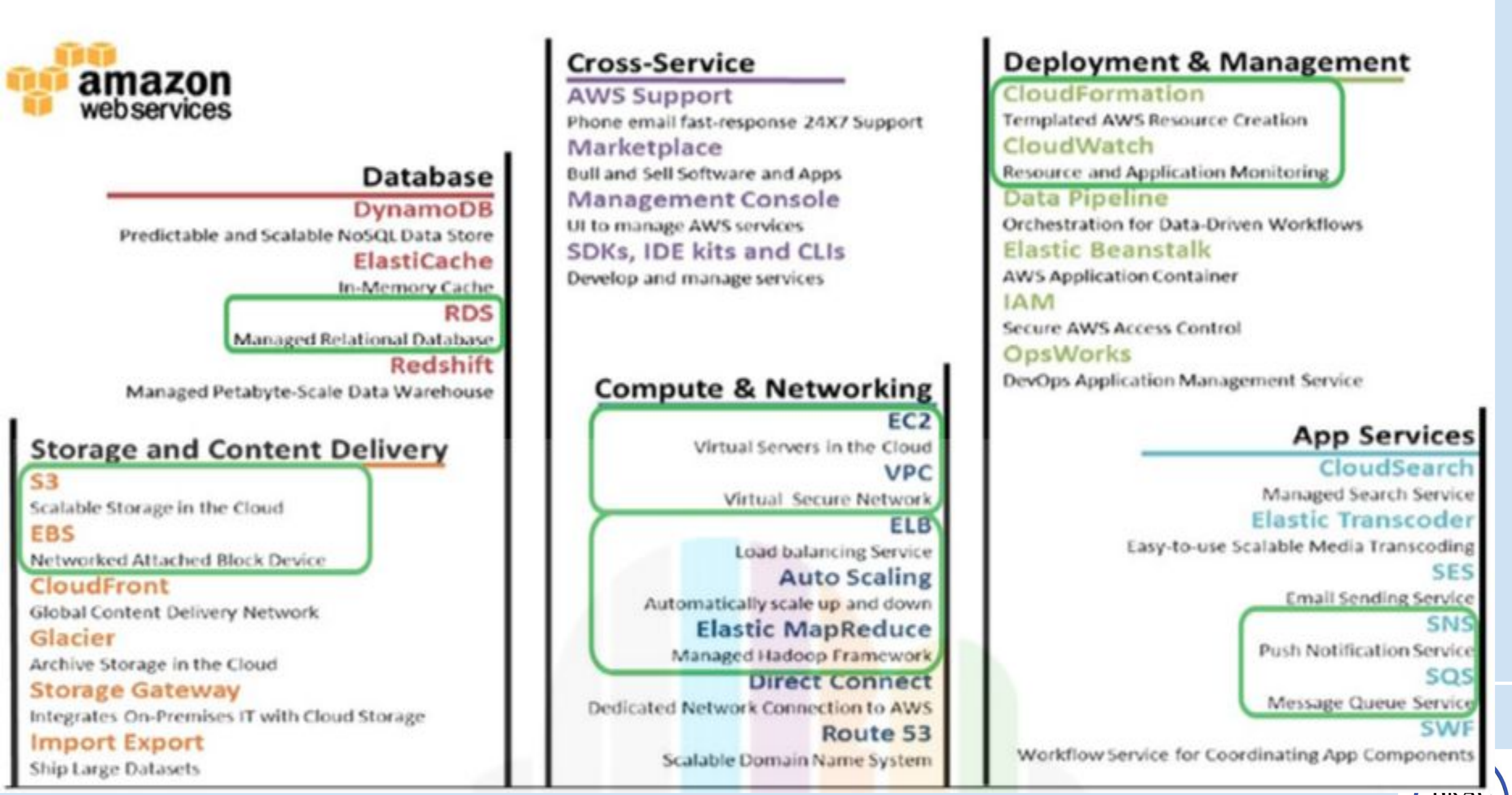
Cloud Computing :

Service Models contd.



Case Study: Cloud Provider

Amazon



amazon webservices

Database

DynamoDB
Predictable and Scalable NoSQL Data Store

ElastiCache
In-Memory Cache

RDS
Managed Relational Database

Redshift
Managed Petabyte-Scale Data Warehouse

Storage and Content Delivery

S3
Scalable Storage in the Cloud

EBS
Networked Attached Block Device

CloudFront
Global Content Delivery Network

Glacier
Archive Storage in the Cloud

Storage Gateway
Integrates On-Premises IT with Cloud Storage

Import Export
Ship Large Datasets

Cross-Service

AWS Support
Phone email fast-response 24X7 Support

Marketplace
Buil and Sell Software and Apps

Management Console
UI to manage AWS services

SDKs, IDE kits and CLIs
Develop and manage services

Compute & Networking

EC2
Virtual Servers in the Cloud

VPC
Virtual Secure Network

ELB
Load balancing Service

Auto Scaling
Automatically scale up and down

Elastic MapReduce
Managed Hadoop Framework

Direct Connect
Dedicated Network Connection to AWS

Route 53
Scalable Domain Name System

Deployment & Management

CloudFormation
Templated AWS Resource Creation

CloudWatch
Resource and Application Monitoring

Data Pipeline
Orchestration for Data-Driven Workflows

Elastic Beanstalk
AWS Application Container

IAM
Secure AWS Access Control

OpsWorks
DevOps Application Management Service

App Services

CloudSearch
Managed Search Service

Elastic Transcoder
Easy-to-use Scalable Media Transcoding

SES
Email Sending Service

SNS
Push Notification Service

SQS
Message Queue Service

SWF
Workflow Service for Coordinating App Components

Some Amazon S3 Use Cases

- Content storage and distribution
  - Serve static files or whole websites from s3 directly
- Better scalability for webserver tier.
  - Reduce ‘data gravity’, low I/O in the server, all HTTP
- Fine-grained access control to buckets.
- Backup, archiving and disaster recovery.
  - Even if Amazon Glacier is a cheaper option.
- … but it’s not a Content Distribution Network.
  - Does not optimize routing for lowest latency.
  - Is not optimized for content streaming.
  - That’s why Amazon Cloudfront exists.

The cost of Amazon S3

- Main reason to use S3: price

- Example: **1 TB** stored, modified 100 GB per month
  - Storage cost: $85 / month
  - Data transfer (upload): $0
  - Data transfer (download): $12 ($0.12 / GB)

- A cheaper option: reduced redundancy (99.99 % instead of 99.99999999 %)
  - Storage cost: $68

- Even Cheaper, but just for backups (very limited functionalities): Glacier
  - Storage cos: %10

Case Study: Cloud Consumer Viber

- Launched in 2011
- Acquired by Japanese e-commerce platform Rakuten in February, 2014 for a tidy sum of $900 million
- Have over 300 million users, growing at 1 million a day, according to statistics of 2014

- Have gone already through three generations :
  - Initially –
    - In 2011, had only few thousand users
    - Used in-memory in-house database
  - 2nd Generation –
    - 1 MongoDB cluster with 150 servers (master and two slaves)
    - 3 Redis cache clusters with a total of around 150 servers.
    - Redis had no support for sharding, so they wrote their own frontend which dealt with sharding.
    - Moved all of its application servers to Amazon Web Services
    - Things that went badly:
      - MongoDB could only managed tens of thousands of operations per second, they needed hundreds of thousands. Large datasets cause MongoDB problems.
      - MongoDB does not scale well with many application servers (it handles each connection in a separate thread, which ends up being very wasteful of CPU and memory).
      - The in-house sharding frontend for Redis wasn't easily scalable (it could only handle increasing the number of servers by factors of two).

Case Study: Cloud Consumer Viber cont.

- Current Architecture (3^{rd} Gen):
  - Have opted for small nodes (about 60GB RAM per node).
  - Have separate nodes for different types of operations (reads vs. sets vs. appends).
  - 400 application server with AWS.
  - 7 Couchbase clusters, each with up to 60 nodes
  - 0-2 replicas, XDCR and external backups
  - Used Amazon S3

They have a total of 150 Couchbase servers (less than half the number needed with MongoDB and Redis).

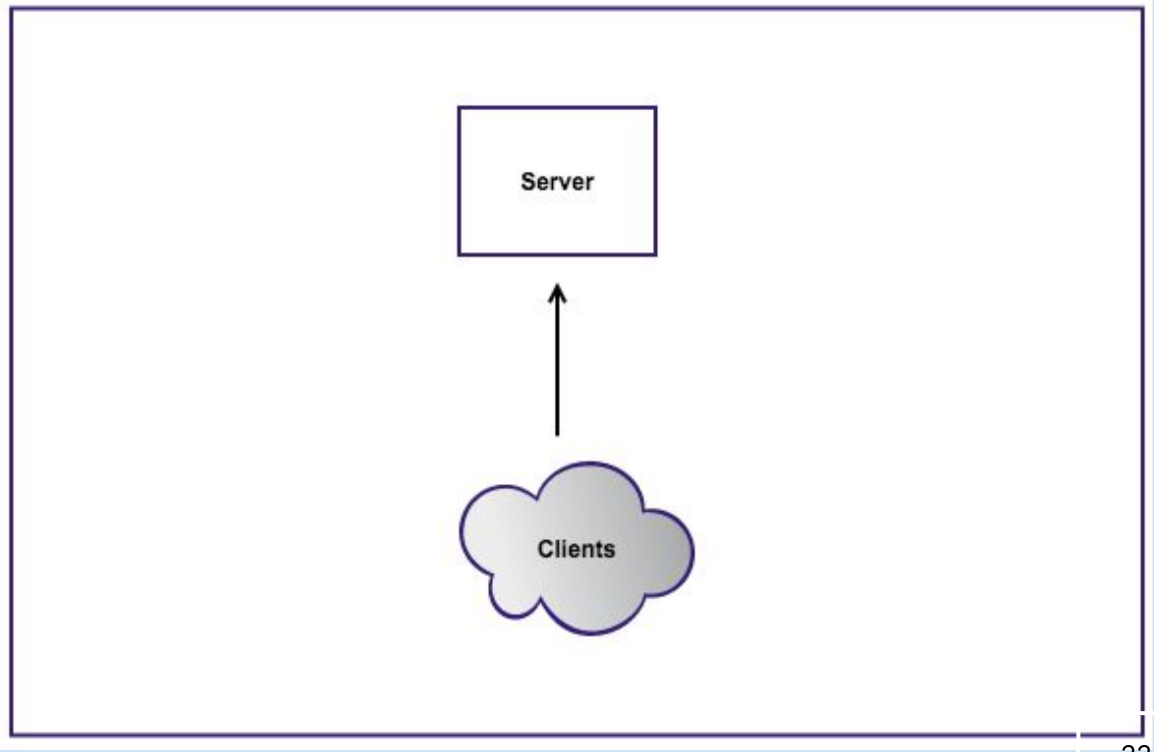
Source : here and here

Case Study: Cloud Consumer

Dropbox

Mid 2007

Almost 0 users

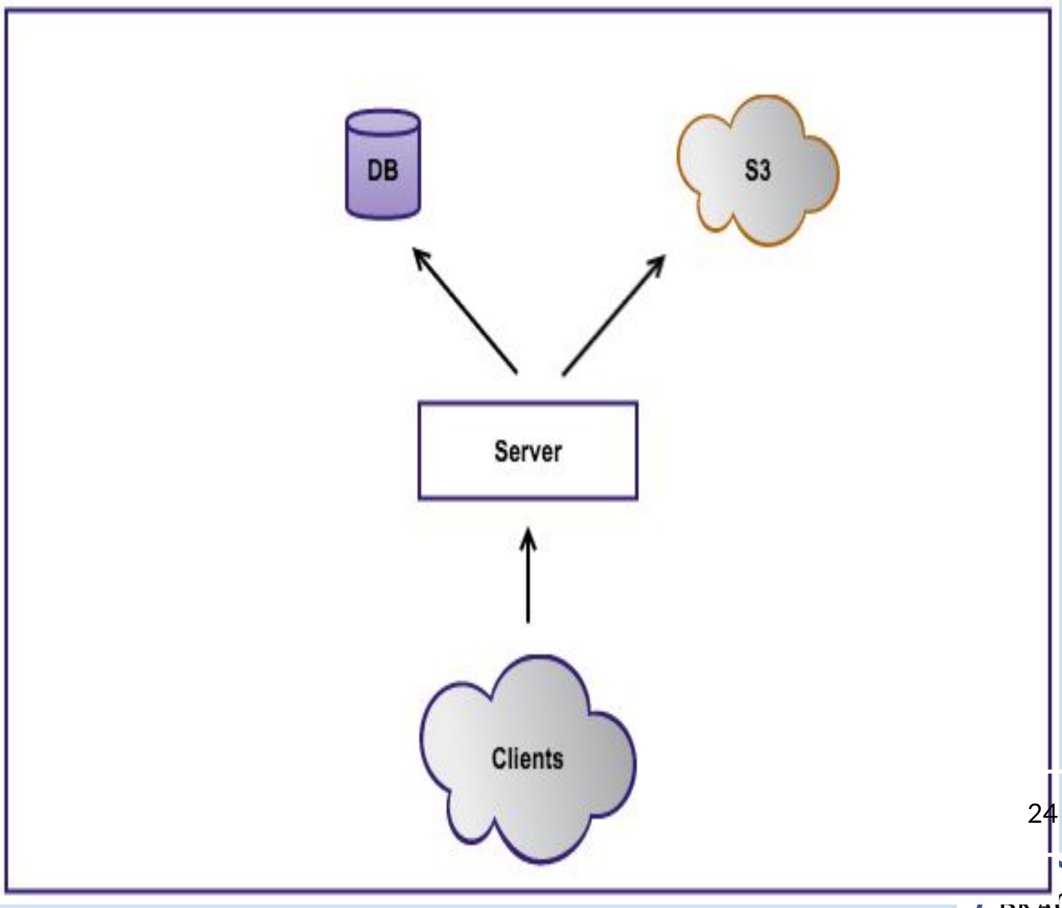


Case Study: Cloud Consumer

Dropbox cont.

Late 2007

- Fast Changing environment, resourcing are growing at the same time with the demand
- The server runs out of space and becomes overloaded
- Put the data on S3
- Use MySQL



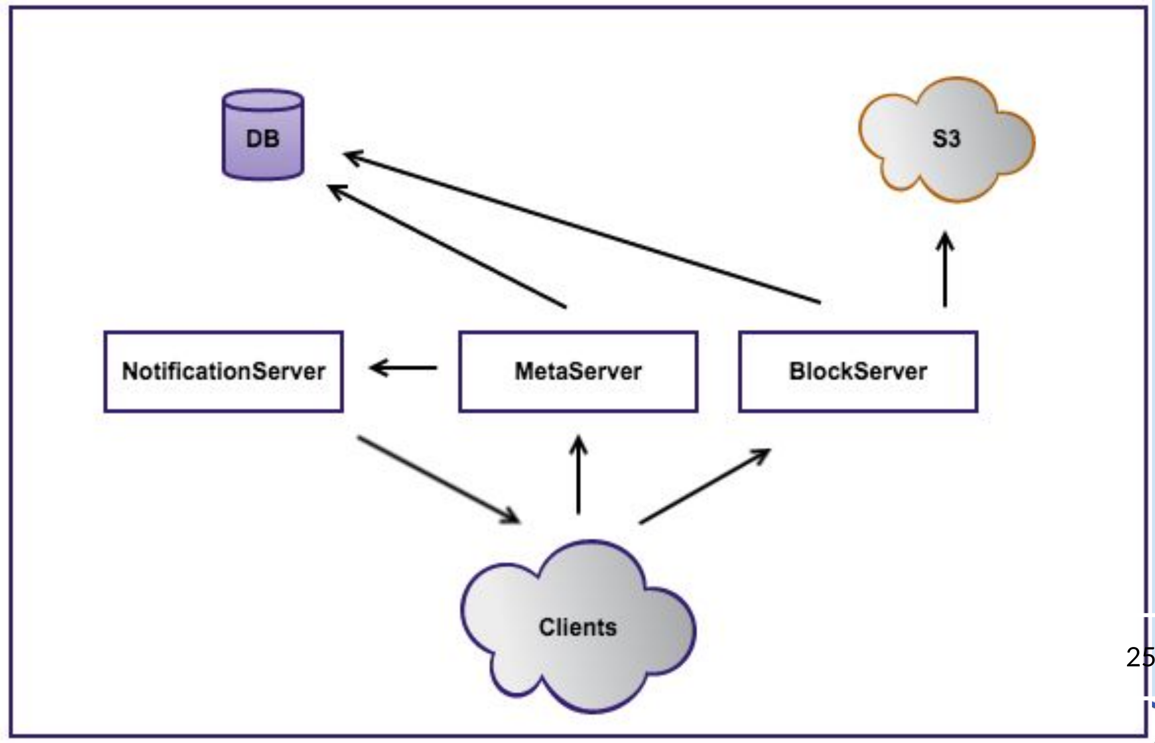
Case Study: Cloud Consumer

Dropbox cont.

Early 2008
50k users

The server is split into 2 servers:

- Metaserver: doing all the metadata calls
- Blockserver: hosting all the files' contents



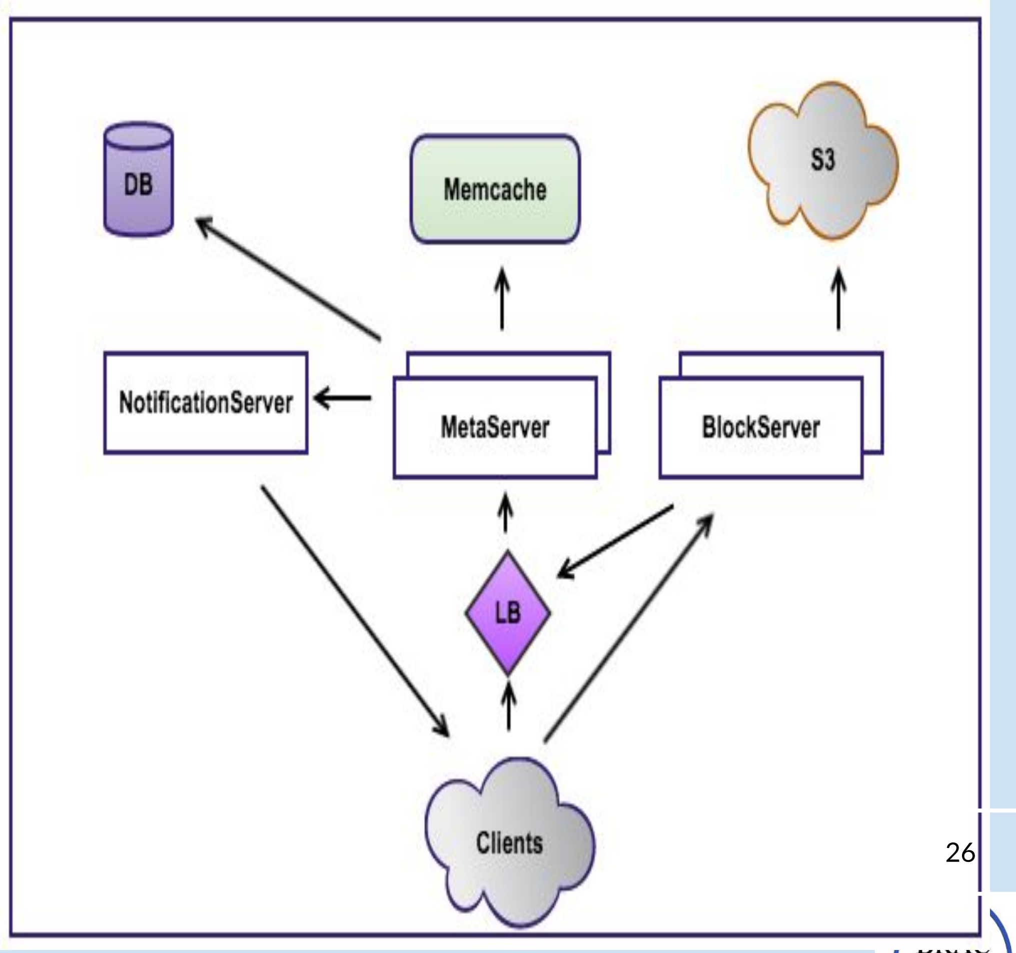
Case Study: Cloud Consumer

Dropbox cont.

Late 2008

100k users

- Need of adding more meta and block servers
- Only one DB -> Add Memcache to avoid DB scaling issues
- Load Balancing methods to optimize resources, minimize response time, avoid overload of resources

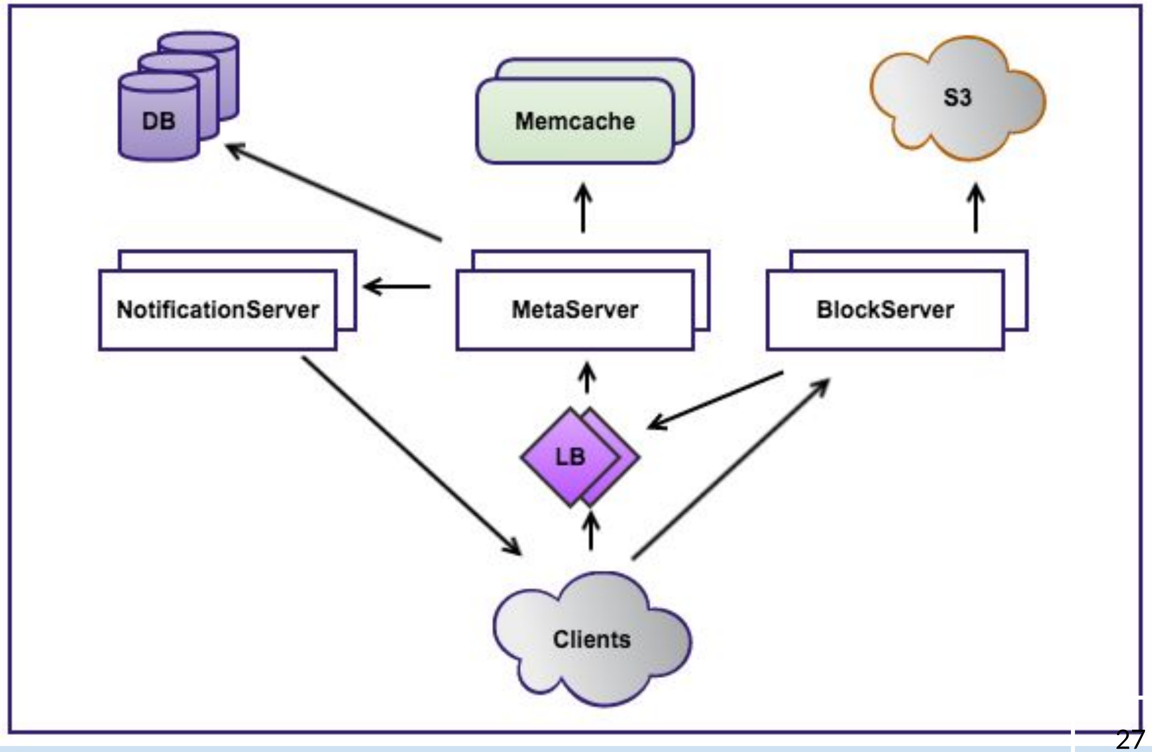


Case Study: Cloud Consumer

Dropbox cont.

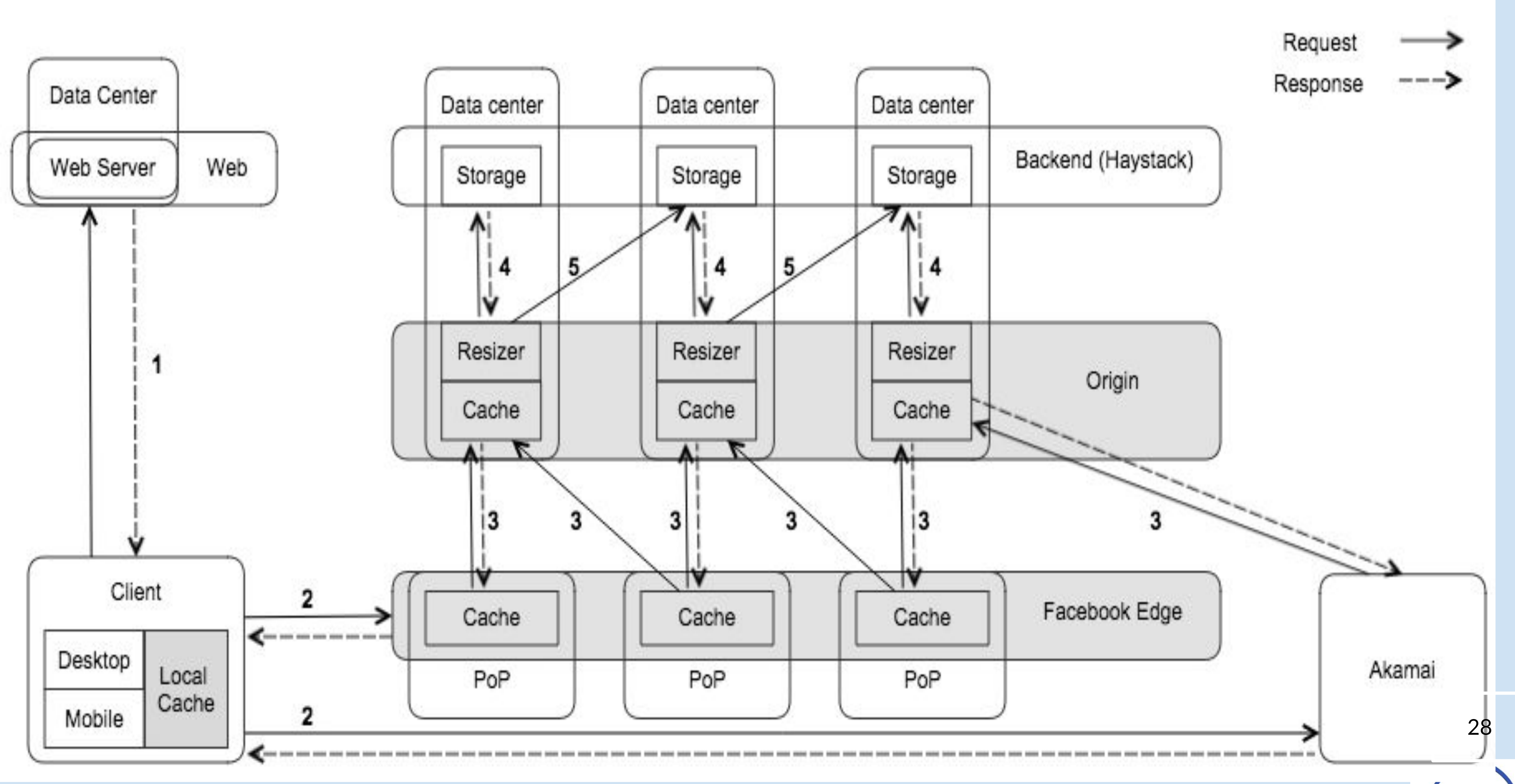
Early 2012
50M users

- Use of several DBs
- Large use of Load Balancing



Case Study:

Facebook



Activity 3: Everydays Cloud as a Service

1. Find a partner.
2. Lets search some different kind of cloud based on deployment model within 1 km area of you (clue: inside or related bracu) Write down in which category they fall? Don't forget about the cloud natures related to deployment models.
3. Now your task is to write >=5 cloud services you use in everydays tasks. Which service models they use according to your concept? Justify your answers.

Both Partner need to submit the answers, you can discuss both but the answer will not be the same, for example if partner 1 will write one example then partner 2 will write another one.

If I found copy then you will not get any marks from today's activity.

There will not be any Fixed answer!!

Be Creative whatever you answer :)