WEB TECHNOLOGY

Application Software :-

1. Standalone application | Desktop application
   Application stored at local machine
   Offline App.
   eg. Notepad, Microsoft word, Google chrome, Adobe photoshop
   → Upgradation done on client | Patch standalone machine

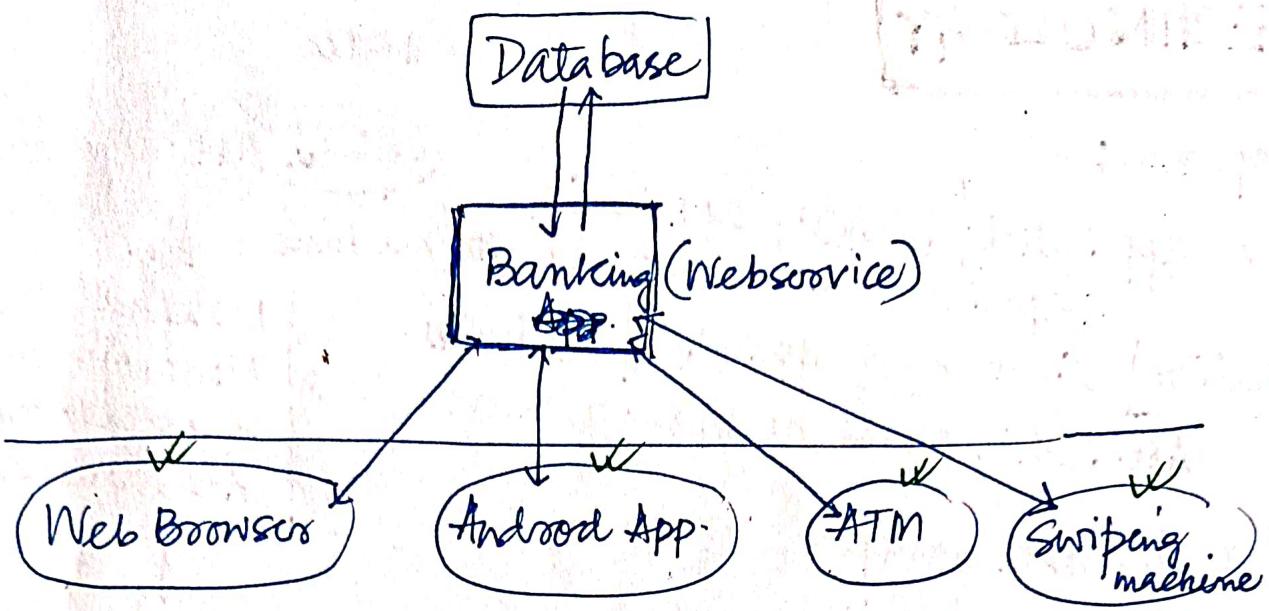
2. Web application | Client-Server application | Online App.
   Cloud solution
   Application stored at remote server
   → Internet for data transfer
   → Web browser
   → accessed from anywhere
   → platform independent
   → Has security risk
   → Does not need to be installed on local machine.
   → Can be used by multiple persons.
   → 2-tier or 3-tier application.
   → Upgradation done globally
   → Sharing S/W & H/W as service which may be expensive for a single user.
   → Better recovery management

Both versions of same software
eg. Whatsapp web
- Web version through browser
- Standalone version

→ Standalone app. can be portable

Standalone application
→ 1-tier application s/w.

→ Web apps are getting popular compared to standalone app.


Database
Banking (Webservice)
App
Web Browser
Android App
ATM
Swiping machine
DATABASE
SERVER
CLIEN
CLIENT

WEB APP DESIGN

Webapplication development phases

1. Project is defined
2. Contracts are written, reviewed & signed.
3. Project is designed.
   (eg. Website storyboard
   Website wireframe
   mockups etc.)
4. Project is developed & tested | Coding
5. Project is uploaded | Publication
6. Project ownership is transfered for maintainance.

Information architecture (IA)

~~How the content is organized~~.

- How the pages are displayed.
- Webgraph.

All-in-one model

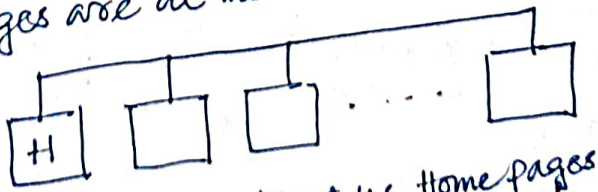
Include everything on home page

This works for small site but does not work for complex site.

Homepage | Gateway page

Flat model

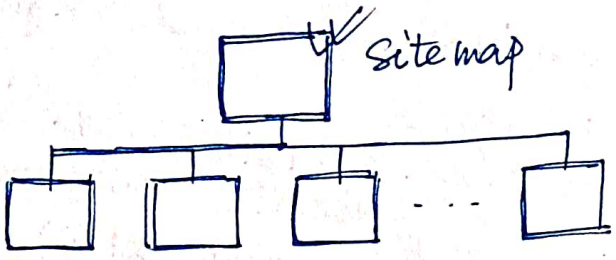
Every pages is accessible from others ones.
All pages are at the same level of structure



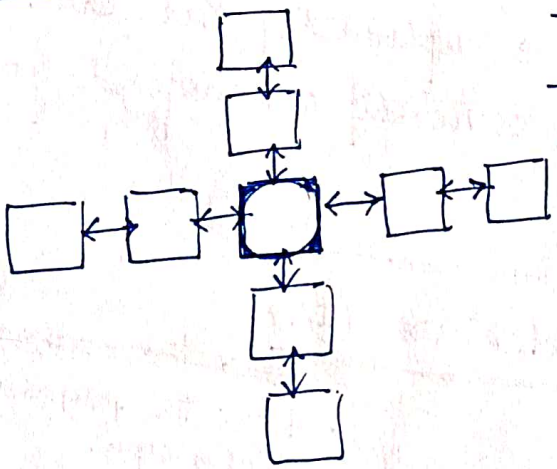
(eg. Contact us, About us, Home pages are accessible from each other of small business sites.)

Index

A spacial page has table of ~~content~~ contents | site map.
It is used for linking among the pages.

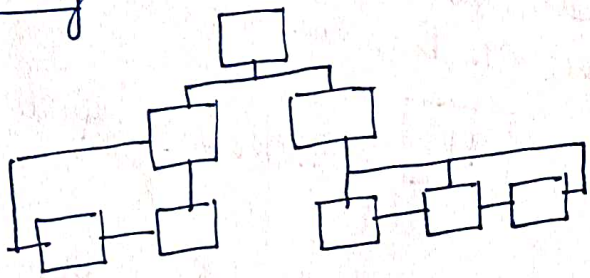


Hub-and-Spoke



This model is applicable for distinct users or groups.

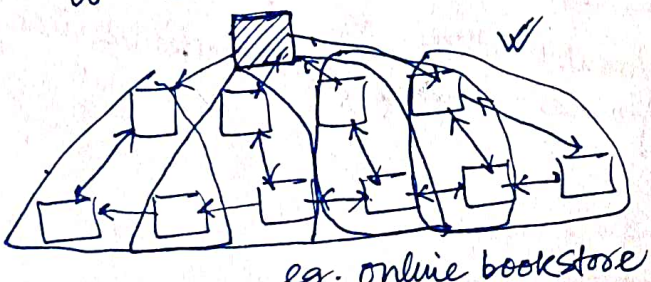
Strict hierarchy



This model is used for global business.

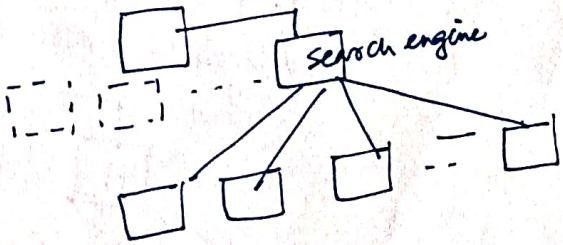
Multidimentional hierarchy

Access content without returning to the home page.



eg. online bookstore

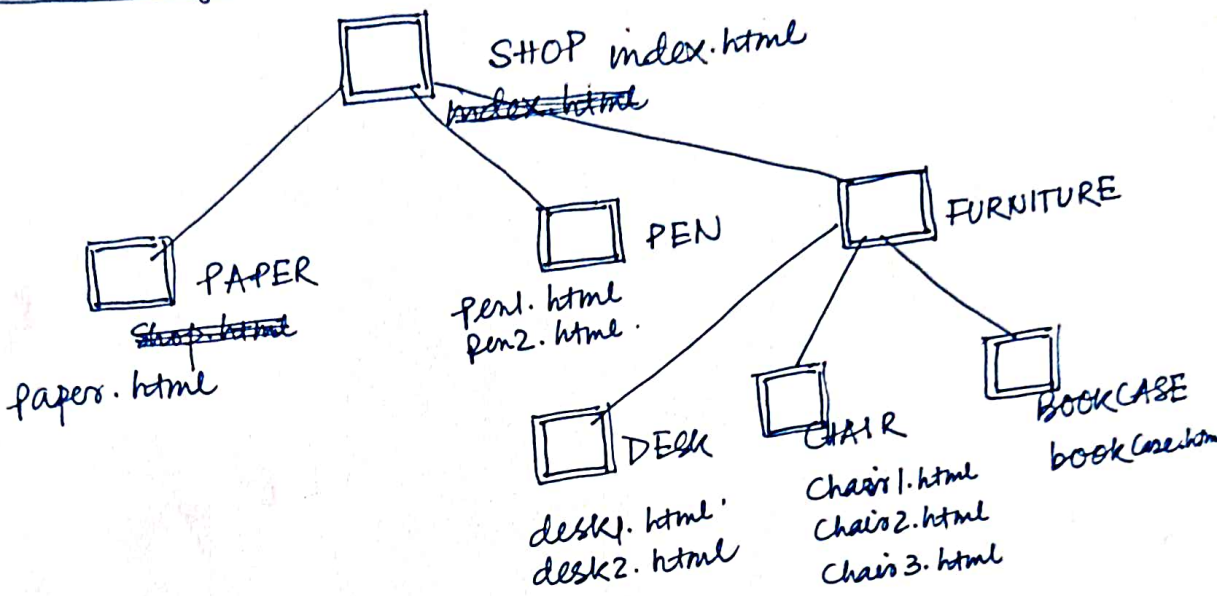
Hierarchial structure with search engine



Content organization | Site-Structure

How information is stored

1. A tree diagram



2. Tabular structure

Type of content →

Category	Folder Name	Filename
Papers	SHOP/PAPER/	papers.html
Pen	SHOP/PEN/	pen1.html pen2.html
Furniture > Desk	SHOP/FURNITURE/DESK/	desk1.html desk2.html
Furniture > Chair	SHOP/FURNITURE/CHAIR/	chair1.html chair2.html chair3.html
Furniture > bookcase	SHOP/FURNITURE/BOOKCASE/	bookcase.html

DESIGN issues

Website wireframe

Blue print of a web pages (Page layout)
Visual guide that represents skeletal framework
Conceptual structure
Rapid prototype of pages

Elements of wireframe
- Information
- Navigation
- Interface (User's interface)

Navigation

Link to other webpage/content

Website storyboard

Blue print for the web project/web application

Design Mockups

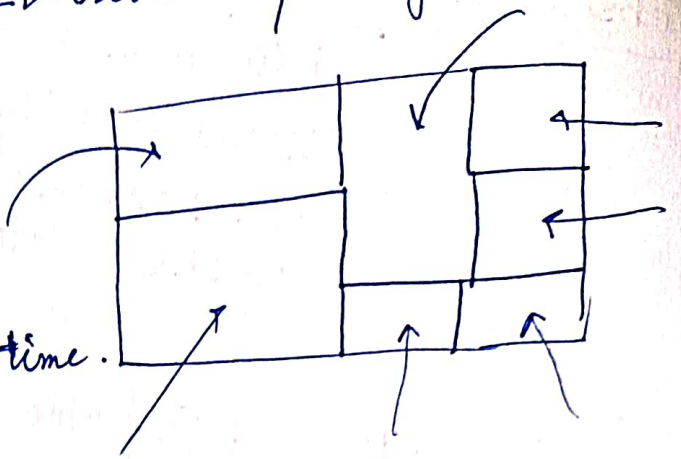
Static design of webpage/website
Looks like final but not functional
(users can not interact)

Slicing design

Process of dividing 2D user's interface layout into multiple files.

In each slide contents is loaded separately

This approach reduces page-weight and loading time.



~~Stock media~~ ~~image/ collection of clip/ video footage that can be used for other films.~~ ~~eg.~~

Stock media

Collection of image/ audio/ video footage, etc. that can be used for other films

Clip art
Production music
Stock footage

WORK-OUT

1. Analyze navigation of webpage

Book - Webdesign
Page - (88-89)

2. Discuss anatomy of a website.

Book - HTML
Page - 4

CLOUD COMPUTING

On demand delivery of resources over internet

Pay-to-use charge | Free of cost

Application software, Servers, storage etc.

Types of cloud

1. **Public**
   - A cloud computing environment is owned by a third party service providers and delivered over the network.
   - Netflix, Google Drive,

2. **Private**
   - A cloud computing environment is owned by an organization.
   - eg. Private cloud providers — Dell, IBM, Oracle
   - Belongs to a business organization.
   - Maintained by private network

3. **Hybrid**
   - Combination of public & private cloud.

Low cost
No maintainance
Limited control

High cost
Maintainance
Full control

Services

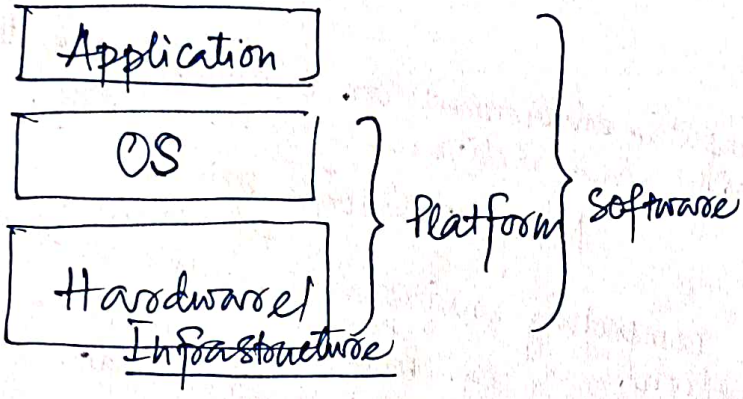
- Hardware
- Software
- Platform

(11) ✓

Basic cloud operations ✓ P-5.

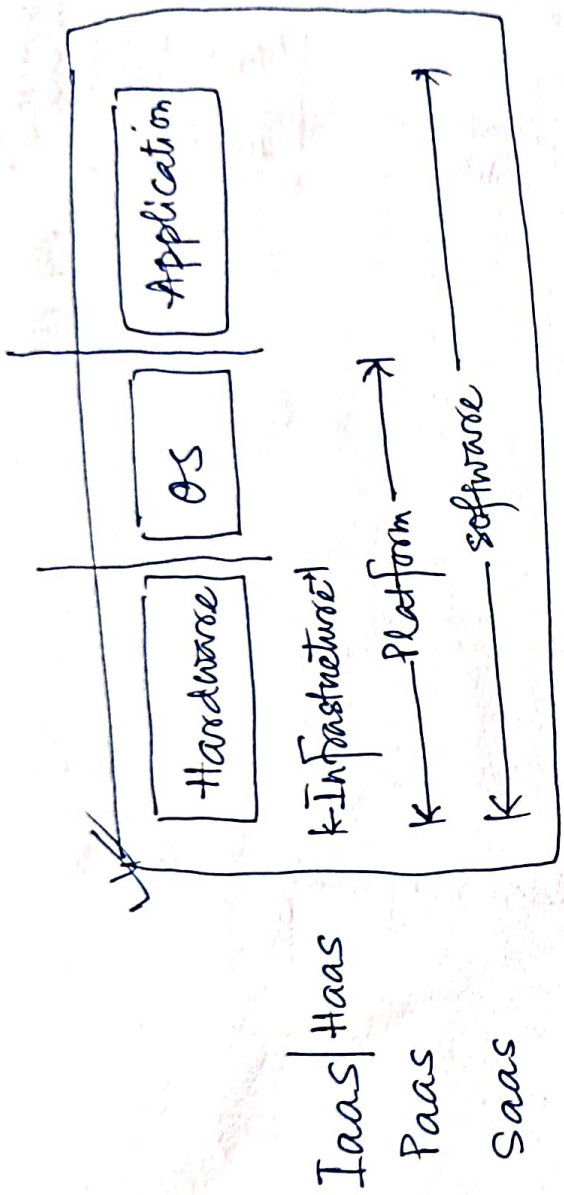
Cloud components P-6.

Client
Data center
Distributed Server



CLOUD SERVICES

Google
Amazon
Microsoft



Grid Computing

A computing infrastructure that combines computer resources spread over different geographical locations to achieve same goal.

Full virtualization

Complete installation of a machine (BIOS, drive, etc.) on another.

Paravirtualization

Multiple OS on single hardware.

Hypervisor application

Multiple virtual servers on one physical server (eg. VMware)

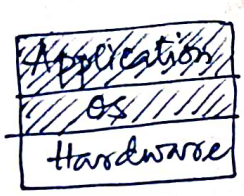
Hardware as a service Haas

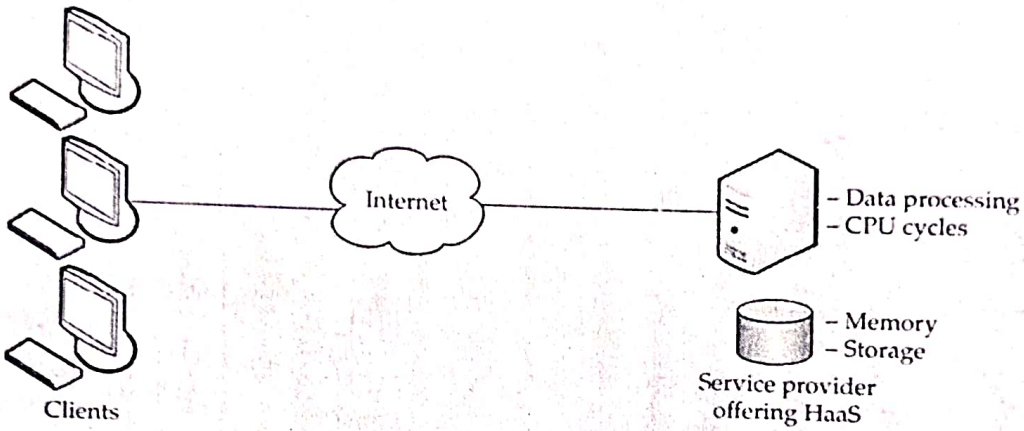
or Infrastructure as a service Iaas

Storage H/W | Memory
Networking H/W
Servers (H/W) Space
Virtualization S/W
H/W

Computing H/W

Amazon web service





HaaS allows service providers to rent hardware resources.

Benefits

Infrastructure can be scaled up & down very easily

Challenges

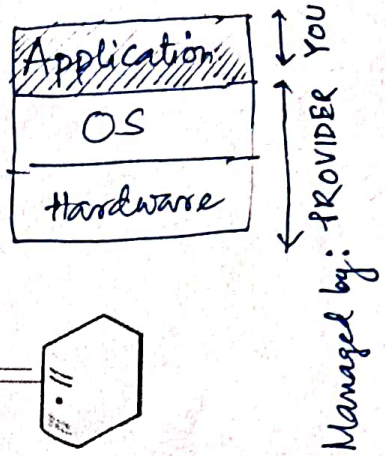
Data security

OUTSOURCING

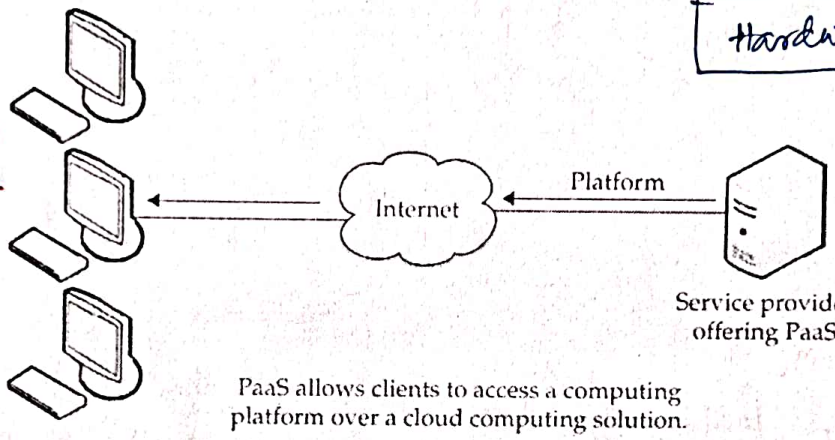
Platform as a service — Paas

Cloud platforms allow user to write applications that run in cloud. It also allows user to access data, application, storage etc.

→ Used as compute cloud.



Google App. Engine



PaaS allows clients to access a computing platform over a cloud computing solution.

Benefits

Cost effective & simple development

No maintainance of software

Integrates web service & database

Challenge

Data resides in third party → Data security

Vendor lock-in problem (not having provision to switch Paas)

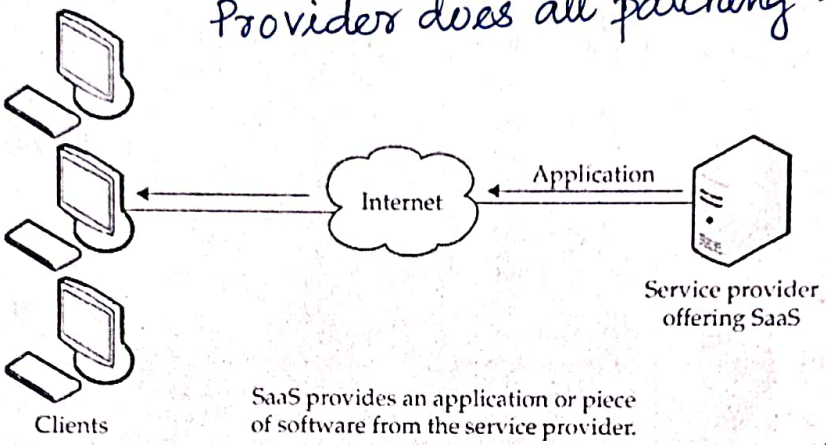
Alternative to buy software & install on local machines

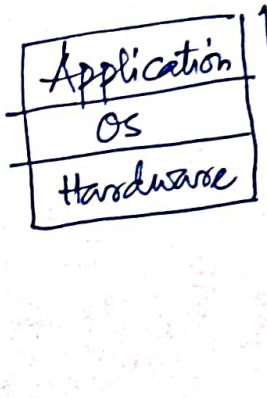
Software as a service SaaS

When software is available at the cloud, customers does not have to maintain or support it.

Google doc
Gmail

Provider does all patching & upgrades.





Customer has to pay for the service to the provider.
Some such applications are Video Conferencing, Accounting etc

Benefits

Does not need complex configuration
Better marketing with the application
Better reliability with web service
Bandwidth is proportional to quality of service.

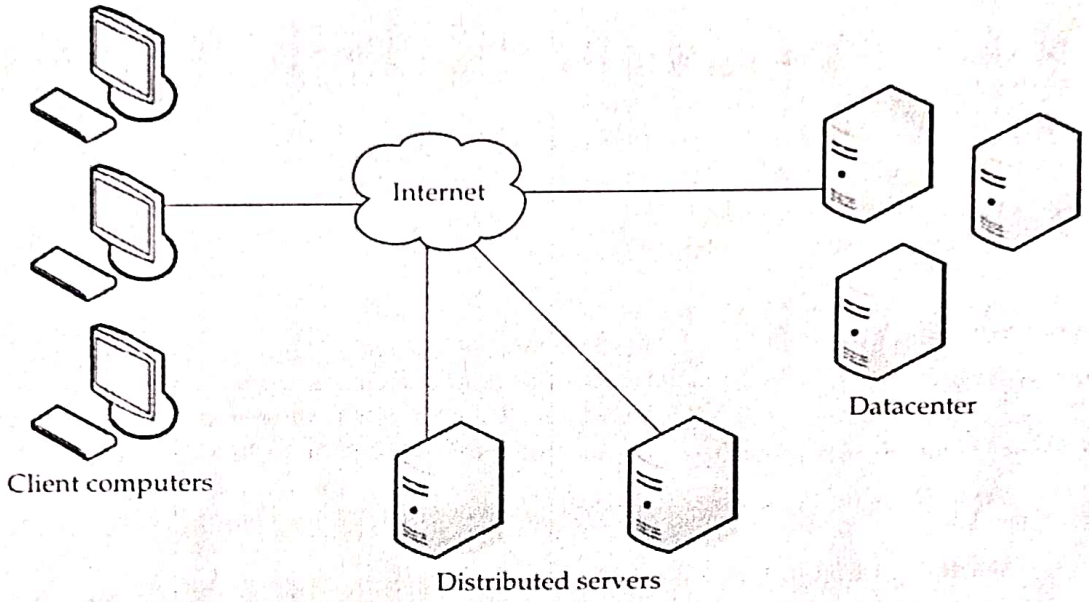
Challenge

Network issue
Cost of the service

CLOUD COMPONENTS

Cloud Components

In a simple, topological sense, a cloud computing solution is made up of several elements: clients, the datacenter, and distributed servers. These components make up the three parts of a cloud computing solution. Each element has a purpose and plays a specific role in delivering a functional cloud-based application, so let's take a closer look.



Distributed Servers

But the servers don't all have to be housed in the same location. Often, servers are in geographically disparate locations. But to you, the cloud subscriber, these servers act as if they're humming away right next to each other. This gives the service provider more flexibility in options and security. For instance, Amazon has their cloud solution in servers all over the world. If something were to happen at one site, causing a failure, the service would still be accessed through another site. Also, if the cloud needs more hardware, they need not throw more servers in the safe room—they can add them at another site and simply make it part of the cloud.

Clients

Clients are, in a cloud computing architecture, the exact same things that they are in a plain, old, everyday local area network (LAN). They are, typically, the computers that just sit on your desk. But they might also be laptops, tablet computers, mobile phones, or PDAs—all big drivers for cloud computing because of their mobility.

Anyway, clients are the devices that the end users interact with to manage their information on the cloud. Clients generally fall into three categories:

- **Mobile** Mobile devices include PDAs or smartphones, like a Blackberry, Windows Mobile Smartphone, or an iPhone.
- **Thin** Clients are computers that do not have internal hard drives, but rather let the server do all the work, but then display the information.
- **Thick** This type of client is a regular computer, using a web browser like Firefox or Internet Explorer to connect to the cloud.

Thin clients are becoming an increasingly popular solution, because of their price and effect on the environment. Some benefits to using thin clients include

- **Lower hardware costs** Thin clients are cheaper than thick clients because they do not contain as much hardware. They also last longer before they need to be upgraded or become obsolete.
- **Lower IT costs** Thin clients are managed at the server and there are fewer points of failure.
- **Security** Since the processing takes place on the server and there is no hard drive, there's less chance of malware invading the device. Also, since thin clients don't work without a server, there's less chance of them being physically stolen.
- **Data security** Since data is stored on the server, there's less chance for data to be lost if the client computer crashes or is stolen.
- **Less power consumption** Thin clients consume less power than thick clients. This means you'll pay less to power them, and you'll also pay less to air-condition the office.
- **Ease of repair or replacement** If a thin client dies, it's easy to replace. The box is simply swapped out and the user's desktop returns exactly as it was before the failure.
- **Less noise** Without a spinning hard drive, less heat is generated and quieter fans can be used on the thin client.

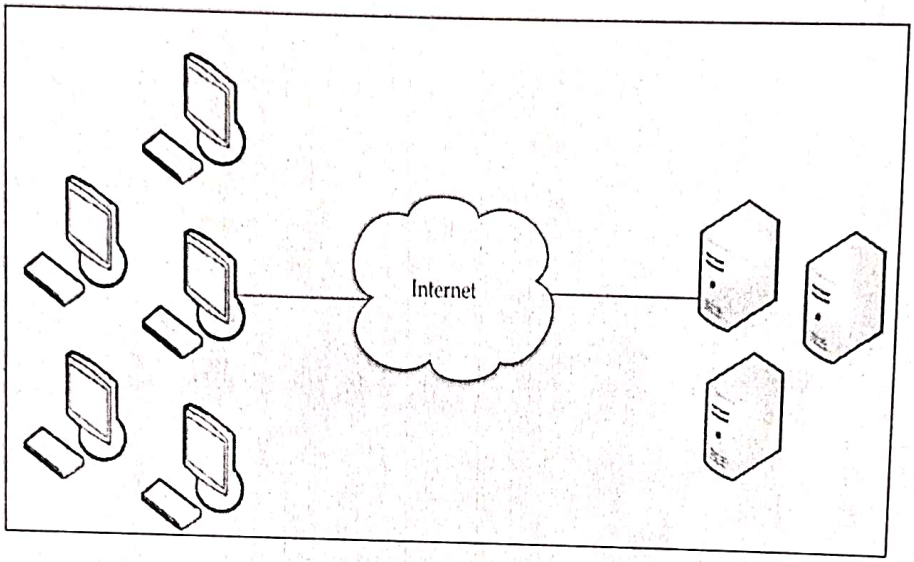
Datacenter

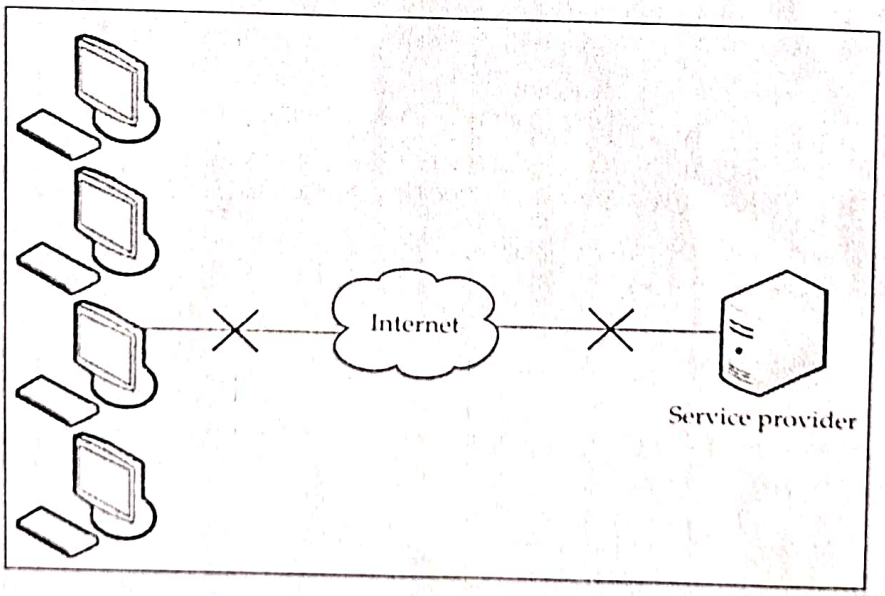
The *datacenter* is the collection of servers where the application to which you subscribe is housed. It could be a large room in the basement of your building or a room full of servers on the other side of the world that you access via the Internet.

A growing trend in the IT world is virtualizing servers. That is, software can be installed allowing multiple instances of virtual servers to be used. In this way, you can have half a dozen virtual servers running on one physical server.

NOTE *The number of virtual servers that can exist on a physical server depends on the size and speed of the physical server and what applications will be running on the virtual server.*

BASIC CLOUD OPERATIONS





Types of attack | Weakness.

Input validation → Blacklist validation → Block bad
Blacklist → Validation → block
Does not contain 'known bad'
~~Block invalid input.~~

→ Whitelist validation → Allow good
~~Accept good only~~
Whitelist → validation → allow.

Attack surface ~~detection~~ reduction

Classifying & Prioritizing Threats.

"Not exposed to internet
More difficult to attack"
→ Set application's availability

Authentication — IDENTITY OF USER — 54

Access Control

Web-App Authentication — 56

Sign-on authentication

Securing athentication (P-84)

Confidentiality
Integrity *
Authentication *
Non-repudiation

Password based.

Authorization — 94
ACCESS RIGHT

Authorization layers → (103)

Borrowers ~~secur~~ security principles — Same origin policy.

Data base & file security principles

Web application security | WebAppSec

Practice of protecting websites, applications, ~~api~~ API from attack.

Types of attacks | weakness

1. **Injection** | code injection

Attacker finds an way to ~~xxxx~~ inject & run code on web server.

Examples –

- SQL injection (SQLi)
  Attacker tries to manipulate SQL ~~queries~~ query.
- Cross-site scripting (XSS)
  Attacker injects own script code into a webpage.
- Command injection
  Attacker inserts command into an operating system.
- CCS injection
  (Change cipher spec)
  Attacker injects code at the time of communication between client & server, to seize encryption key.

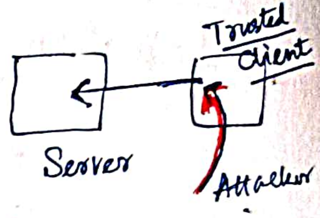
2. **Broken authentication & session management**

Attacker can capture/bypass authentication methods used by a web application.

3. Insecure direct object references

Attacker can access resources in a system directly.
This is a type of weakness in application as it provides direct access to object based on user's input.

4. Cross site request forgery



CSRF

Attacker create mallicious link.
Let webserver receive mallicious request from a trusted browser.

5. Security misconfiguration

E.g. User fails to enable antivirus, firewall etc.
Security controls are inaccurately configured that data are put to risk.

6. Insecure cryptographic storage

Sensitive data (eg. password) are stored in plaintext on the server.

7. Failure to restrict URL Access

Allow users to use certain webpages selectively.
Allow authorized users only.

8. Insufficient transport layer Protection

Not taking any measures to protect network traffic.

9. Unvalidated redirects & forwards

Web application accepts untrusted input that cause web application to redirect.

Input validation

→ Analyzing input
→ Allow or disallow by checking against standard defined

Blacklist validation

Blocklisting

Prepare Blacklist → Validation → Block ✓

Whitelist validation

Whitelisting

Prepare Whitelist → validation → Allow ✓

Purpose

Data/Input does not contain "bad" content.

Done at server side

list-out invalid inputs.

Standard

Analogy

No T-shirts
No Jeans

P_1 P_2 P_3

P_1, P_2 → Blacklisted → Blocked.

P_3 → Allowed

list-out valid inputs

Standard

Analogy

Full sleeve shirt
Formal trouser

P_7 P_8 P_9

P_7 → Not allowed.

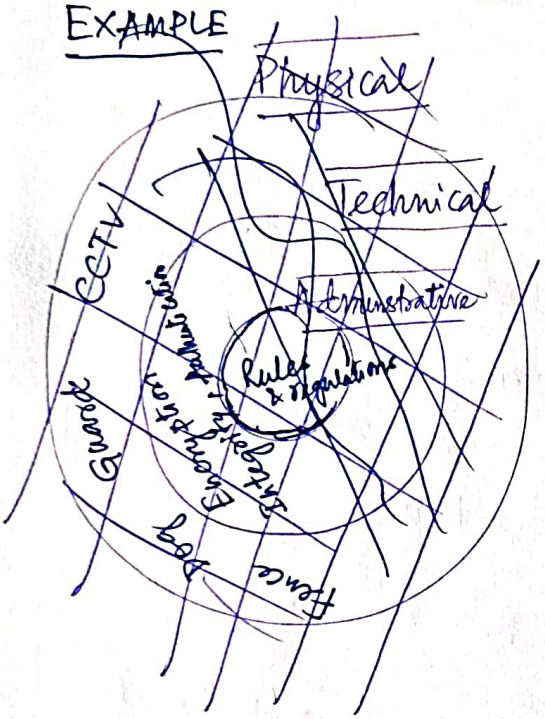
P_8, P_9 → Whitelisted → Allowed.

Defense in depth approach

Castle approach

Multiple layers of security control
Failure of at any point does not make the system completely vulnerable

EXAMPLE



Example

Technical

- Data — Database security / Message level security
- Application — Authentication, Authorization, Audit.
- OS — Vulnerability management
- Hardware — Network security (e.g. Firewall)
- Physical — Fence, Dog, Guard, CCTV
- Administrative — Rule & regulations, Policies. etc.

Advisable to use defense-in-~~depth~~ depth in combination with whitelist validation.

Attack surface detection

Leave attacker with fewer ways to perform attack.

Attack surface ← Places where attacker could compromize device or network of an organization.

Attack surface reduction rule (ASR rule) →

eg. ✓ 80/20 rule
reduce the amount of running code
80% users do not need service more than 20%

principle of Least exposure ✓
" not exposed to internet
more difficult to attack "
— set application's availability

ASR rules report:

Information about ASR rules.

eg. → Threat detection ✓
→ Configuration of ASR rule ✓
→ Configure exclusion ✓
~~Stop-AttackSurface~~.
Files excluded from ASR

~~principle of least privilege~~

or

principle of least privilege

Allow user what he needs to do only.
Minimize permission grant

Classifying & Prioritizing threat

All security vulnerabilities are not equally serious.

STRIDE threat model

(S) **S**poofing : Act of disguising so that it appears to be authorized.

(T) **T**ampering : Unauthorized modification of system/data

(R) **R**epudiation : Denial of responsibility

(I) **I**nformation disclosure (information leakage) : System reveals sensitive information to potential attackers.

(D) **D**enial of service (DoS) : Making system inaccessible to its intended users → Flood servers with traffic.

(E) **E**levation of privilege : System grants right to attackers. (privilege)

IIMF

(I) Interception : Allow unauthorized users to access data

(I) Interruption : Cause assets to become usable or unusable.

(M) Modification : (Tampering)

(F) Fabrication : Attacker inserts fake object.

CIA

(C) Confidentiality — Illegal access to system | data | application | ... etc.

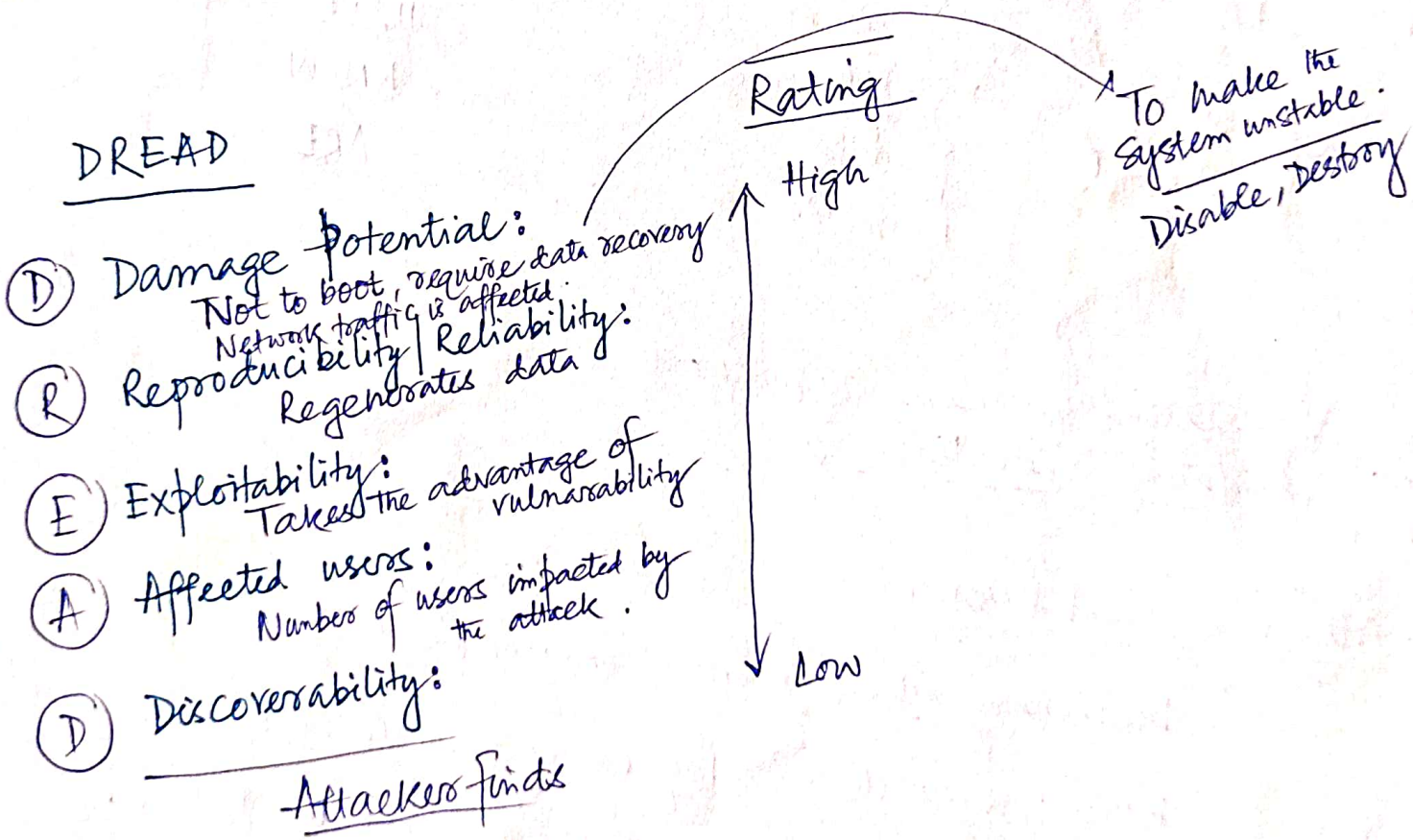
(I) Integrity — Attempt to corrupt data

(A) Availability — Use multiple devices to make system inaccessible.

(Distributed DoS)
DDoS.

— Use multiple computers | device to flood a targeted resource.

DREAD



(D) Damage potential:
Not to boot, require data recovery

(R) Reproducibility | Reliability:
Network traffic is affected.
Regenerates data

(E) Exploitability:
Takes the advantage of vulnerability

(A) Affected users:
Number of users impacted by the attack.

(D) Discoverability:
Attackers finds

AUTHENTICATION & AUTHORIZATION

Identity of user

Access right of user

1. Proving identity

key, password, smart card, security token

Two factor and three factor authentication KNOW + HAVE + ARE

2FA → ATM card + PIN

ATM card — user has HAVE

PIN — user knows. KNOW

3FA → ATM card + PIN + Biometric

ATM card — HAVE

PIN — KNOW

Biometric — ARE

2. Validating username & password

Validating credentials ✓

Attack against password

Dictionary attack → Guess common words & simple variation

Brute force attack → Try every possible combinations of symbols

Rubber hose attack → Extracting password by torture.

Web App Security

Authentication & Authorization.

Authentication mechanism

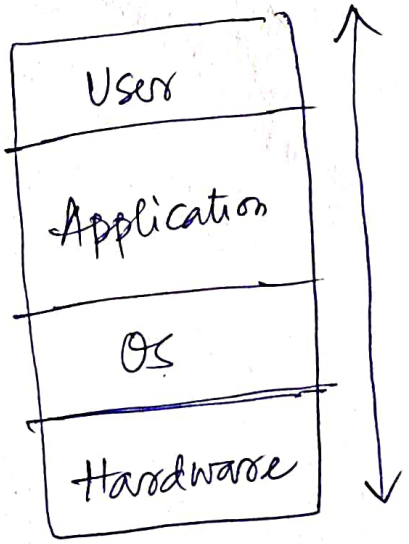
- Secure the transmission by using encrypted channel (SSL/TLS)
- Account lockout works after certain number of unsuccessful login attempt.
- Disable account when user leaves the system, for long time. (Not using the system)
- Use strong credentials
- Avoid 'remember me' option or stay sign-in option saving password in local machine

Authorization layers

Authorization happens at many points forming layers

Horizontal layers

Vertical layers



Types of permissions — Read, Write, Execute