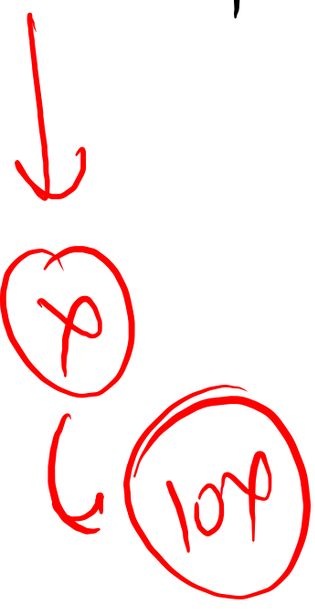
15th — 24th Dec

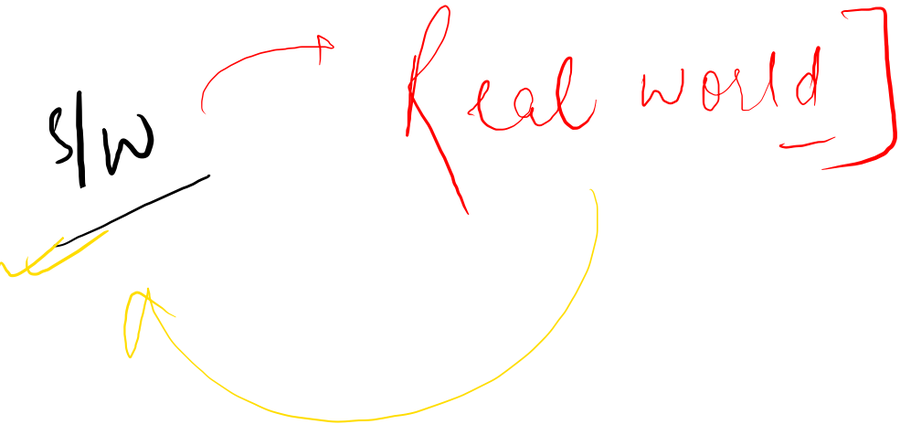
(10 days)

concepts

Problems/
Case studies/
interviews

System Design :-




S/W
Real world

→ Create a city

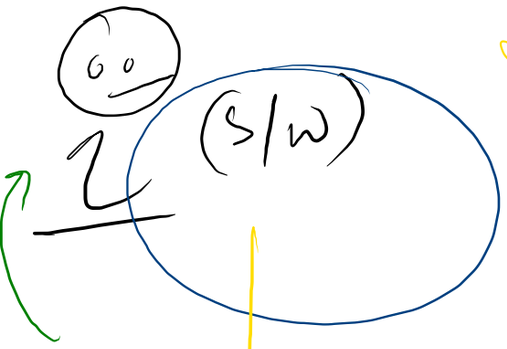
[System Design]

How level plan :-

1. school
2. House
3. Hospitals
4. Roads
5. Gym

House

Area
Rooms
Size
Materials

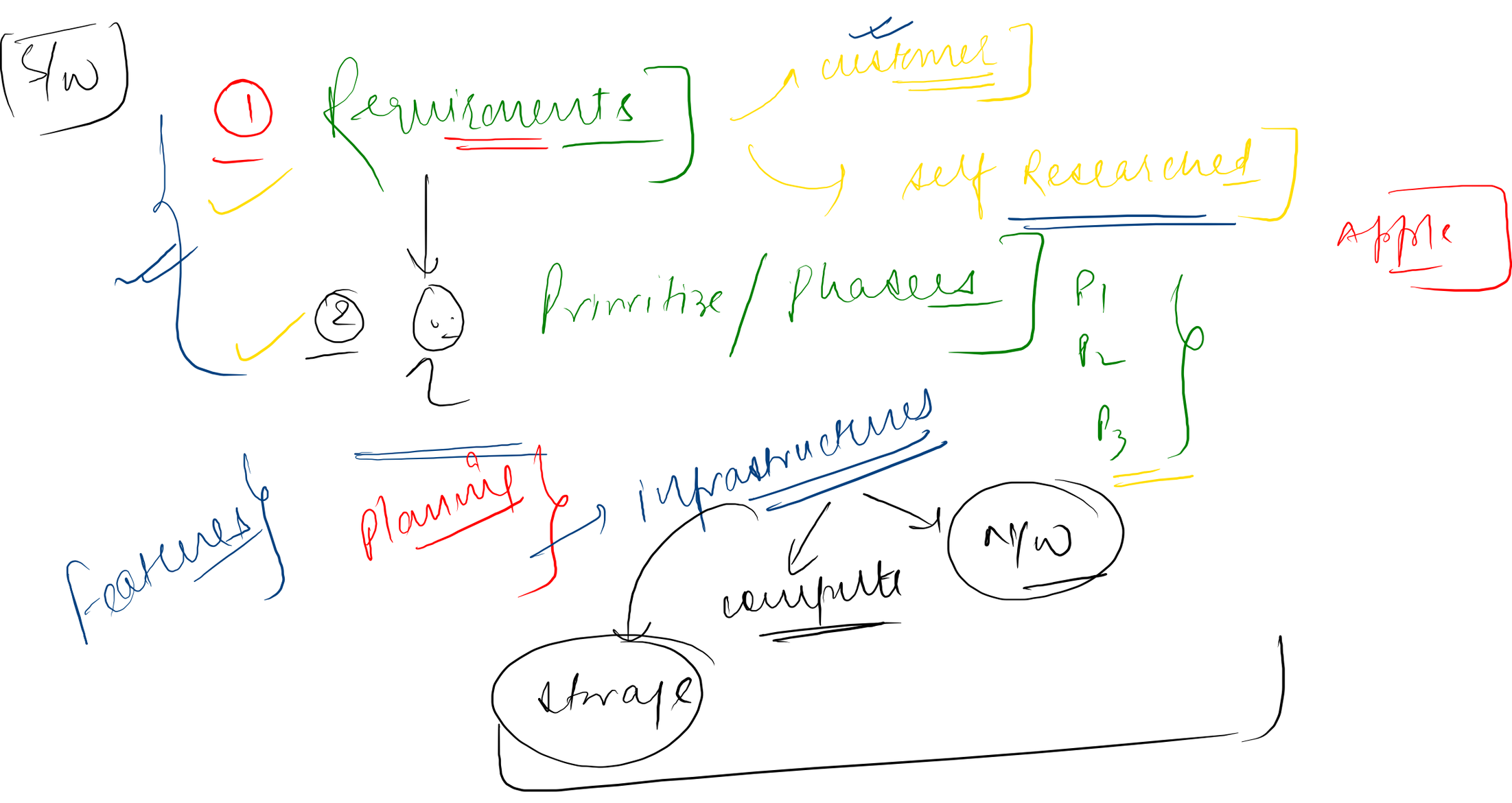


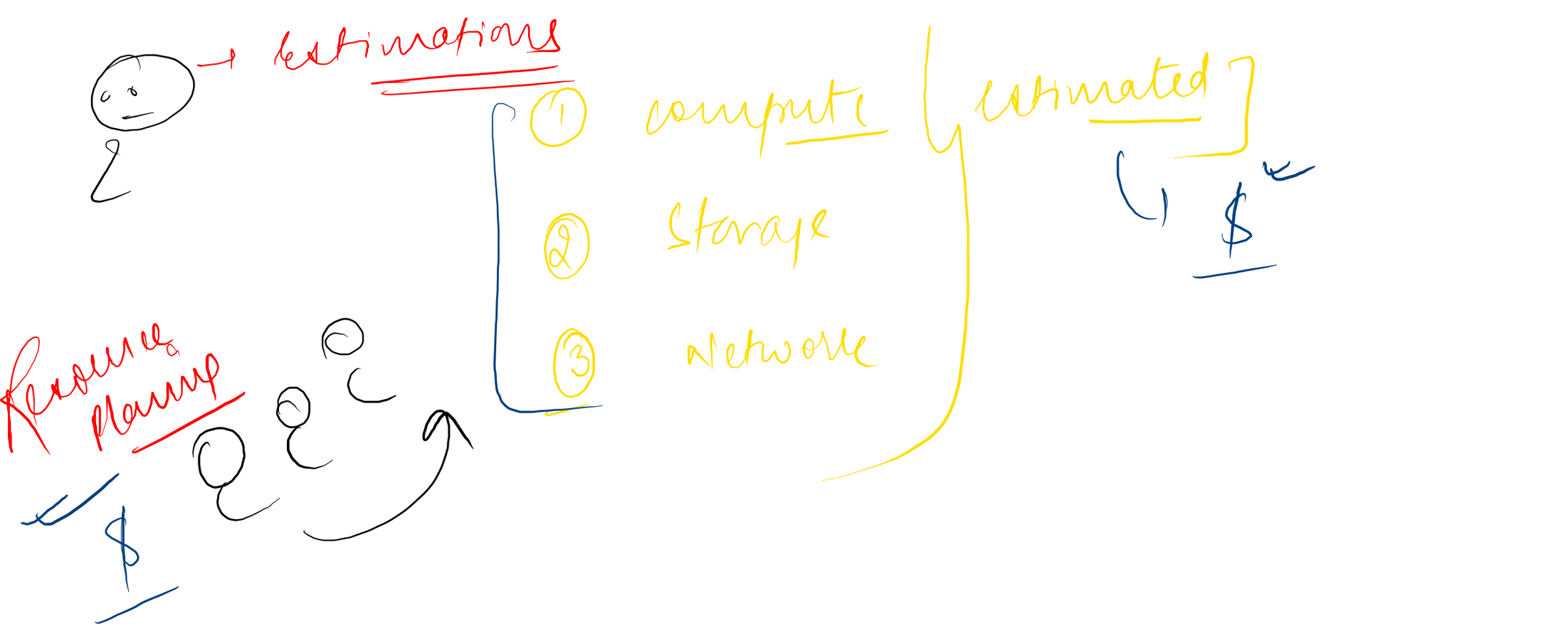
High level Planning

↳ Low/Detailed Planning

↳ Execution

System Design


S/w
1
Requirements
customer
self Researched
Apple
2
Prioritize / Phases
P1
P2
P3
Features
Planning
infrastructures
compute
n/w
storage


estimations
1 compute
2 Storage
3 Network
estimated
$
Resource planning
$

Estimation & Infrastructure

Components

Facebook Messenger

1. UI
2. Message Processing System
3. Notification System

destination → components

Interface

communicate among each other

communication

Messages

UI

Message Processor

Requirement

↳ Prioritization

↳ infrastructure like mate
component
communication

HLD (High level Design)

Requirements

Functional requirement

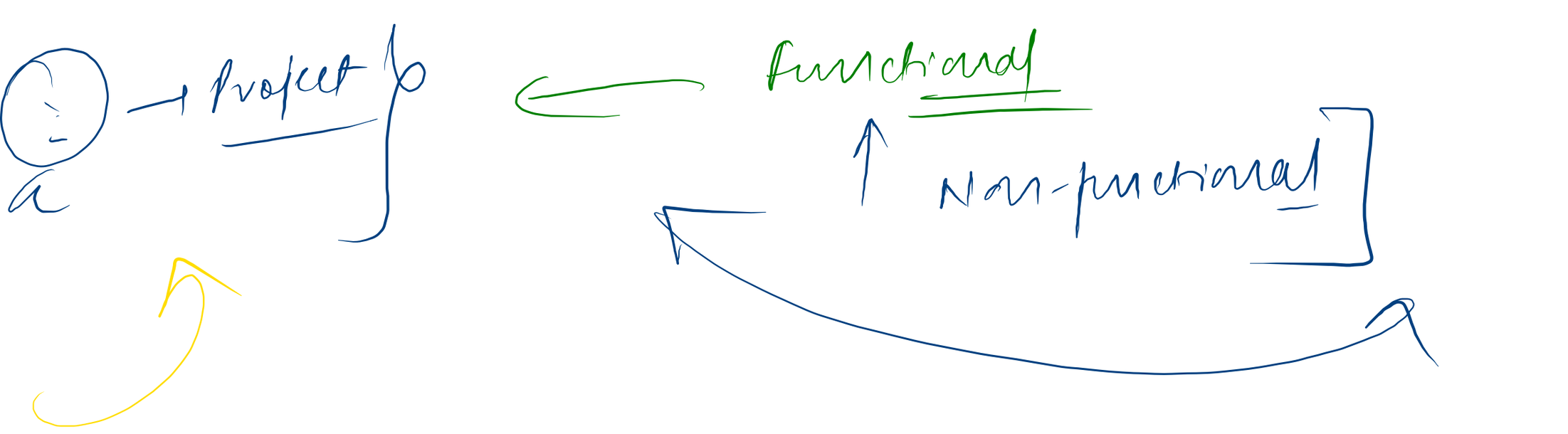
Messenger

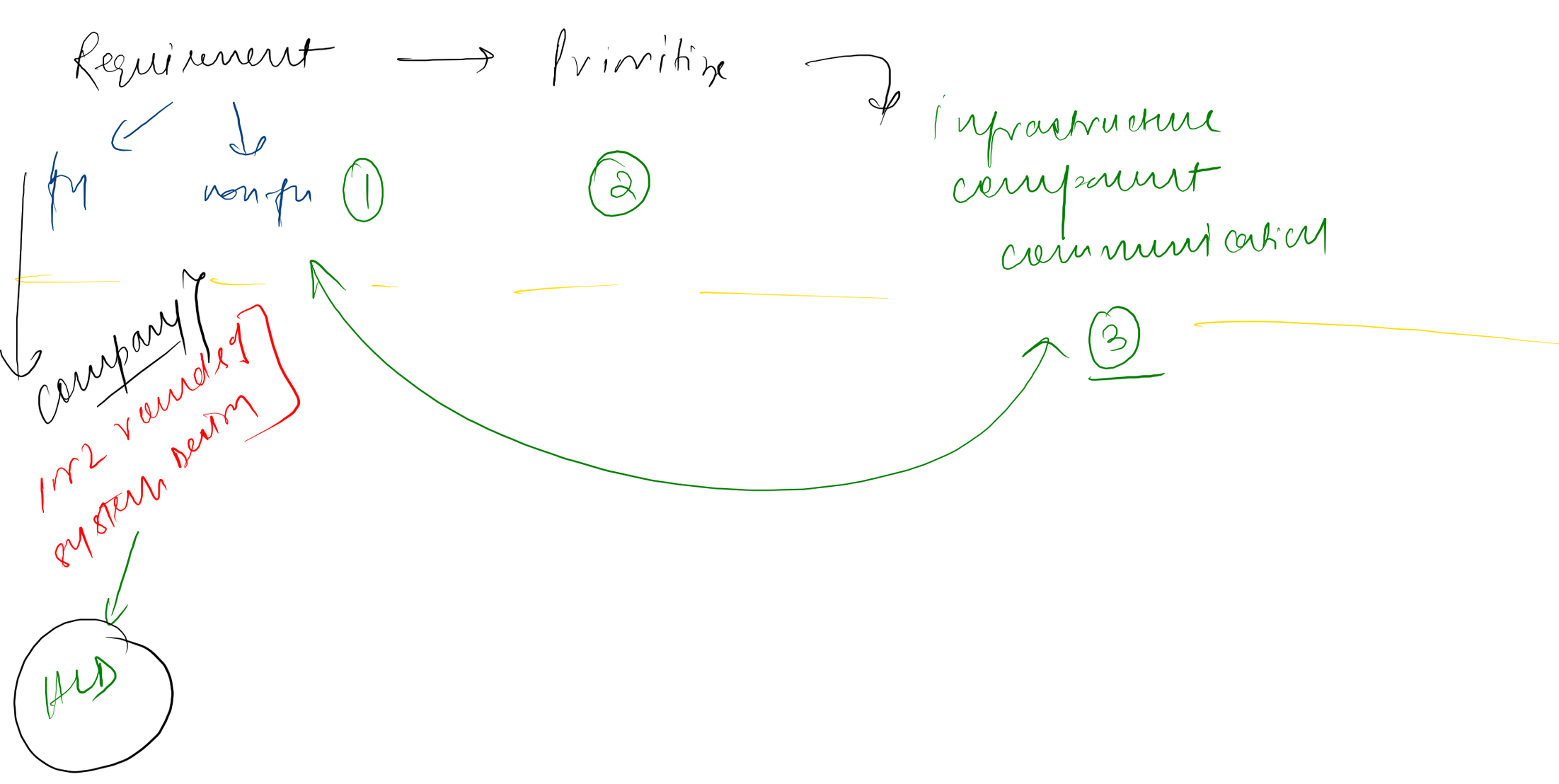
Non functional requirement

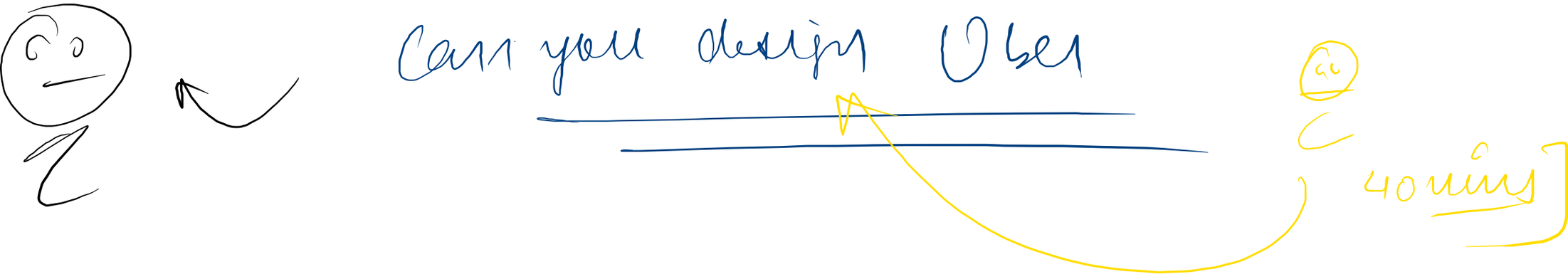
Qualities and Characteristics

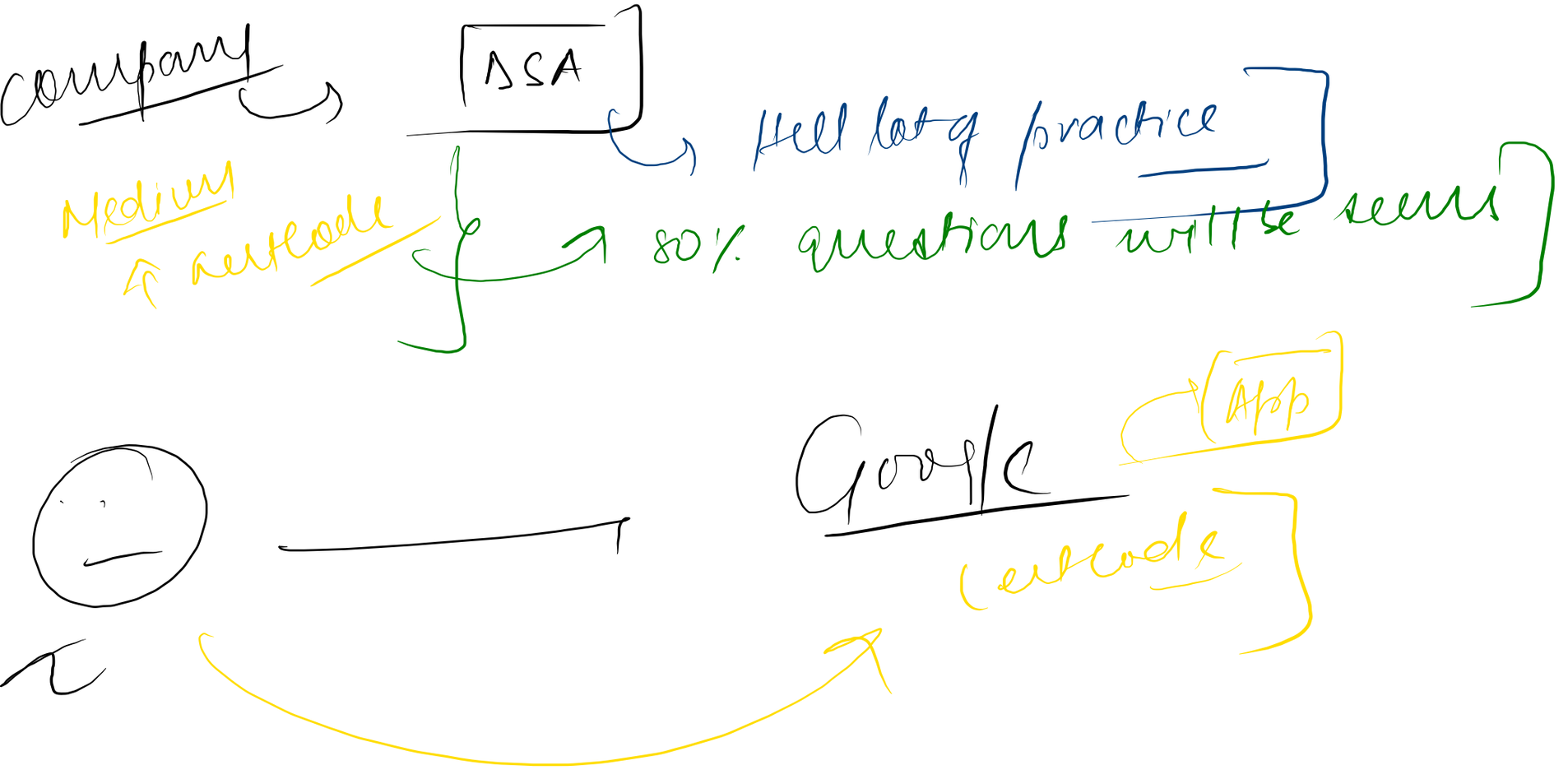
1. Performance
2. Latency
3. Security
4. Ease of use
5. Reliability
6. Scalability
7. Availability
8. Consistency

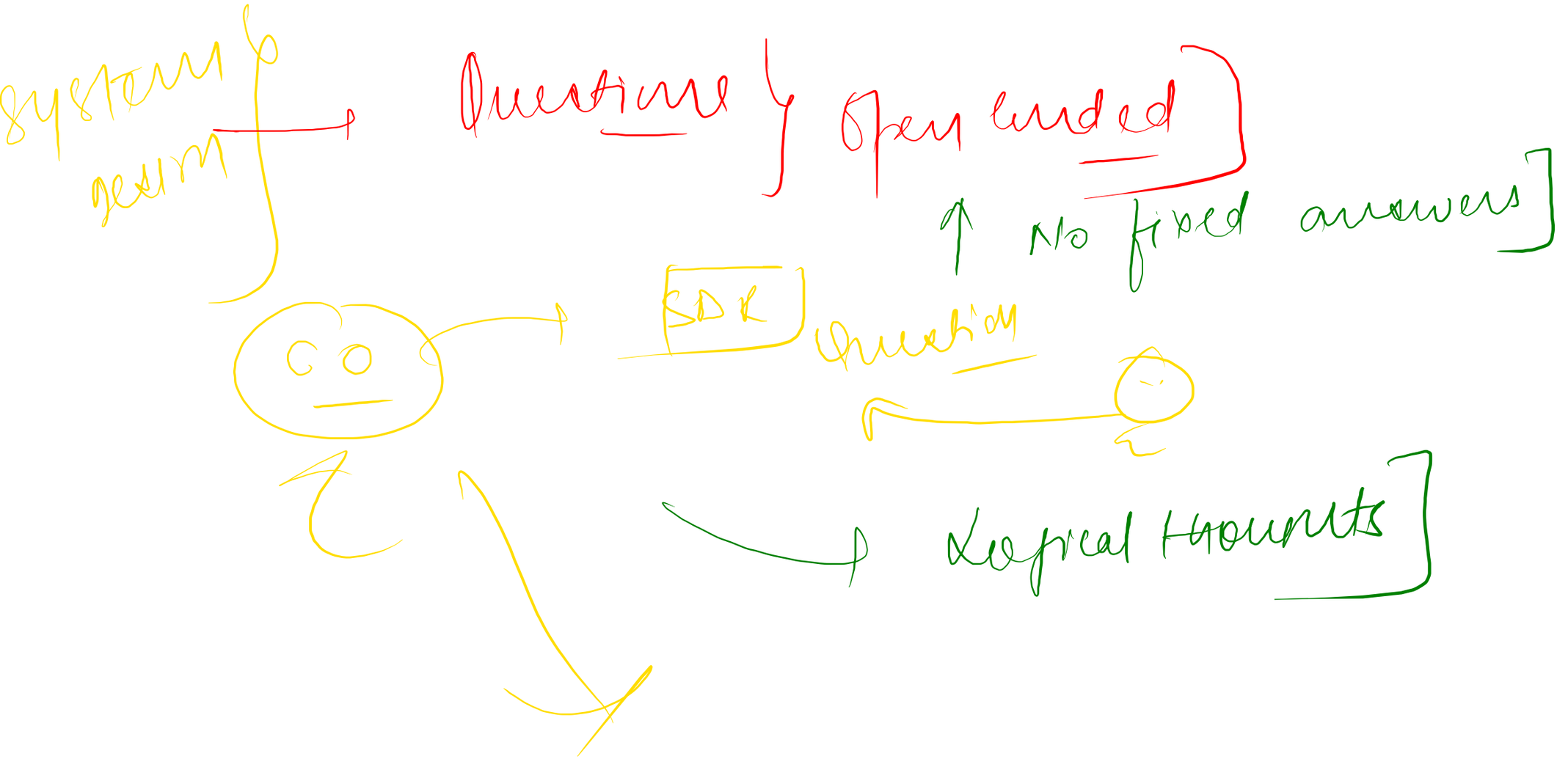
1. Needed by customers
2. Functions

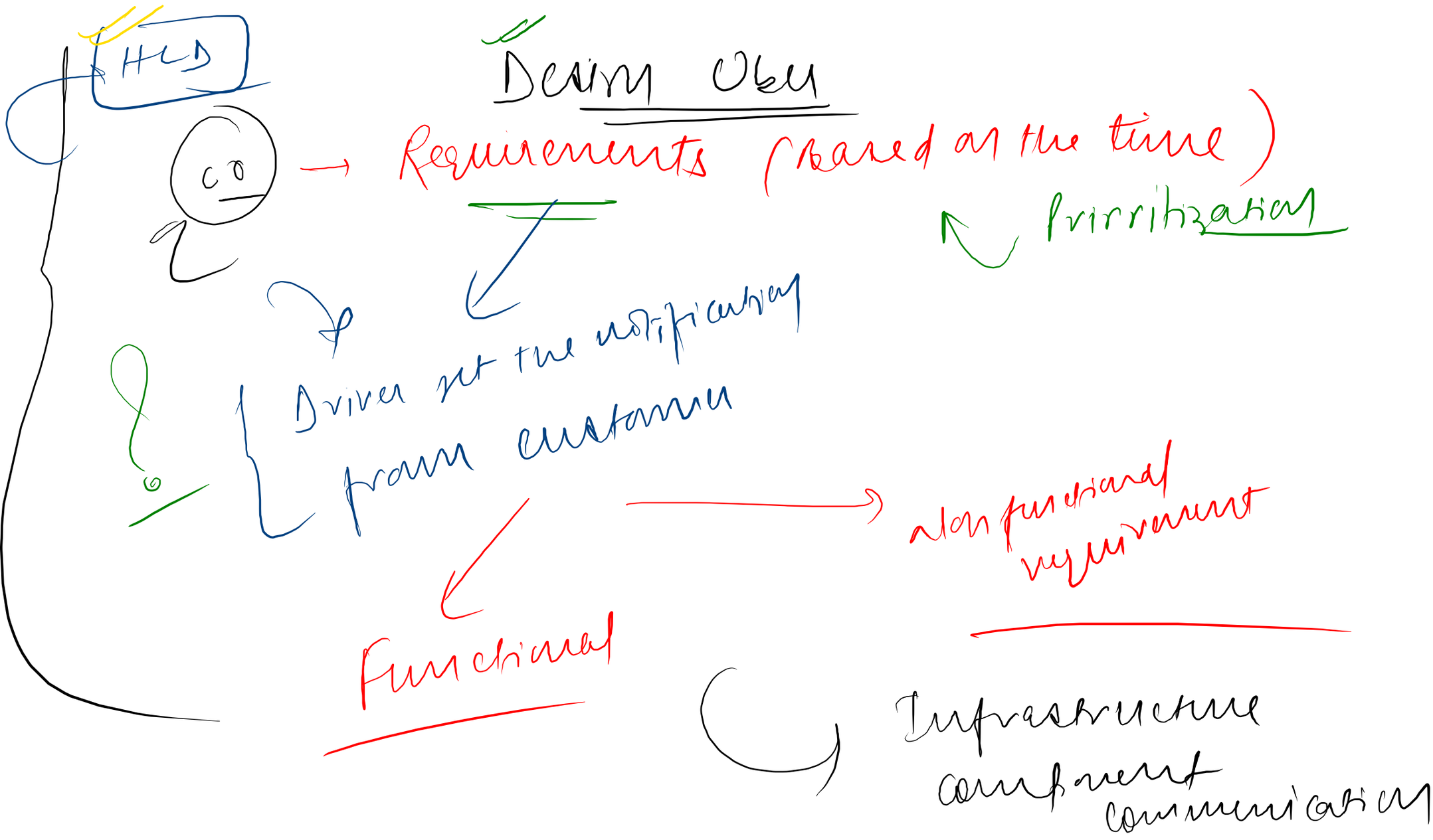

Project
functional
Non-functional


Requirement
Prioritize
infrastructure component communication
fr
non-fr
1
2
3
company
1 or 2 rounds of system design
HLD


Can you design Uber
40 мин


company
ASA
Hell lot of practice
Medium
↑ leetcode
80% questions will be same
Google
App
leetcode


system design
Questions
Open ended
No fixed answers
SDK
Question
Logical thoughts


HLD
Design Obj
Requirements (based on the time)
Prioritization
Driven at the notification from customer
Functional
Non functional requirement
Infrastructure
component
communication


Requirements → Prioritization → component Architecture communication
HLD
Messages
Low Level Design
Layers
Services
SOLID
Design Patterns
OOPs
Detailing of each component
Message Parsing Service
Class
relationships
Behaviours

System Design

① HLD

components
infrastructure
communication

Architects
Sr Eg
Product.

Experienced
+
Beginner

college

10 days

② LLD

Detailing of each component

Class Diagram

Design Principle / Patterns

DS

Developer
Sr developer

Pre skills Team

Frameworks

Problem

↳ Requirement phase

Fn

Non Fn

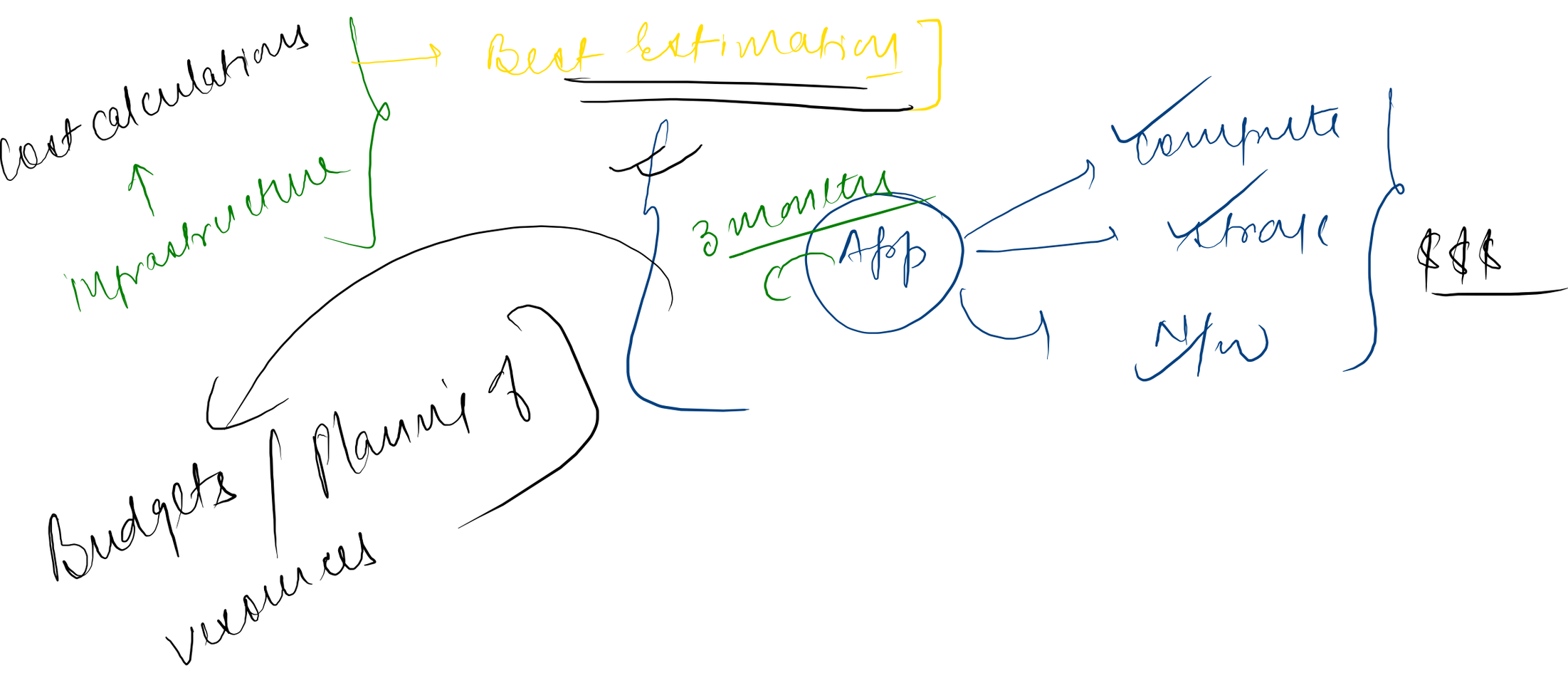
Infrastructure

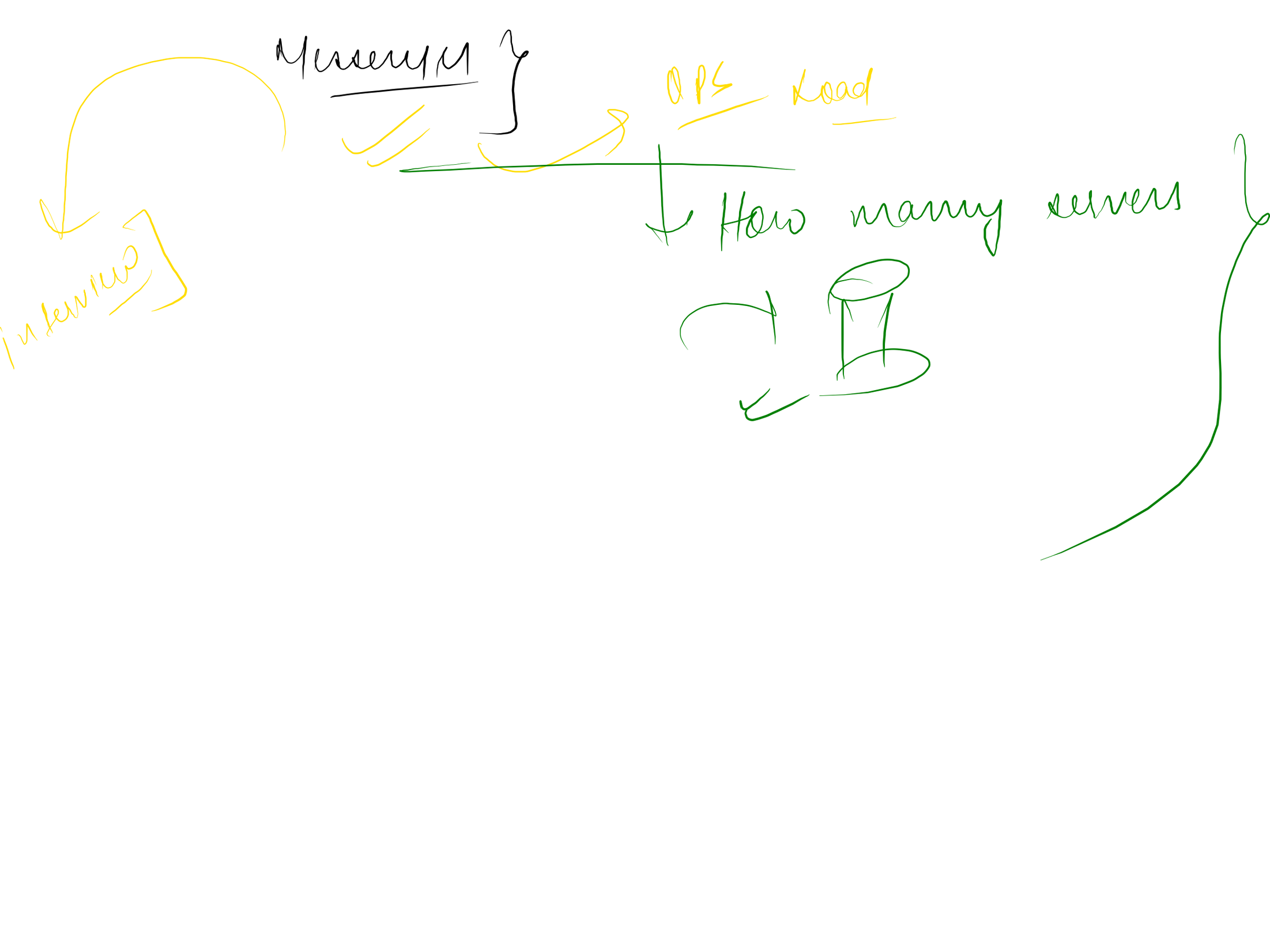
Component

Communication

Pwskills Tech

↳

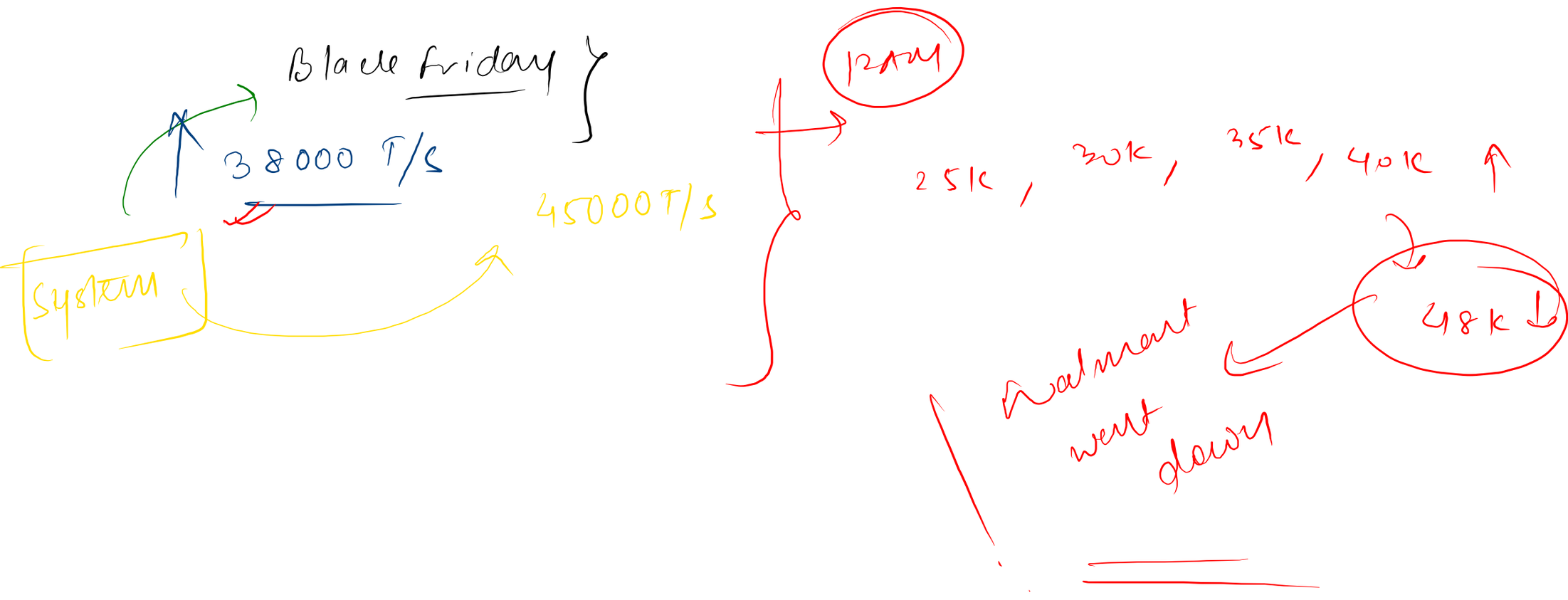

Cost calculations
Infrastructure
Best estimation
3 months
App
Compute
Storage
N/w
$$$
Budgets / Planning / resources


Memory
QPS
Load
How many servers

Back of the envelop calculations

↳ infrastructure estimations


Walmart
Development
Jan – May
June – Oct
Performance Test
Scalability
Target
Amazon
Nov
Thanks Giving !!!

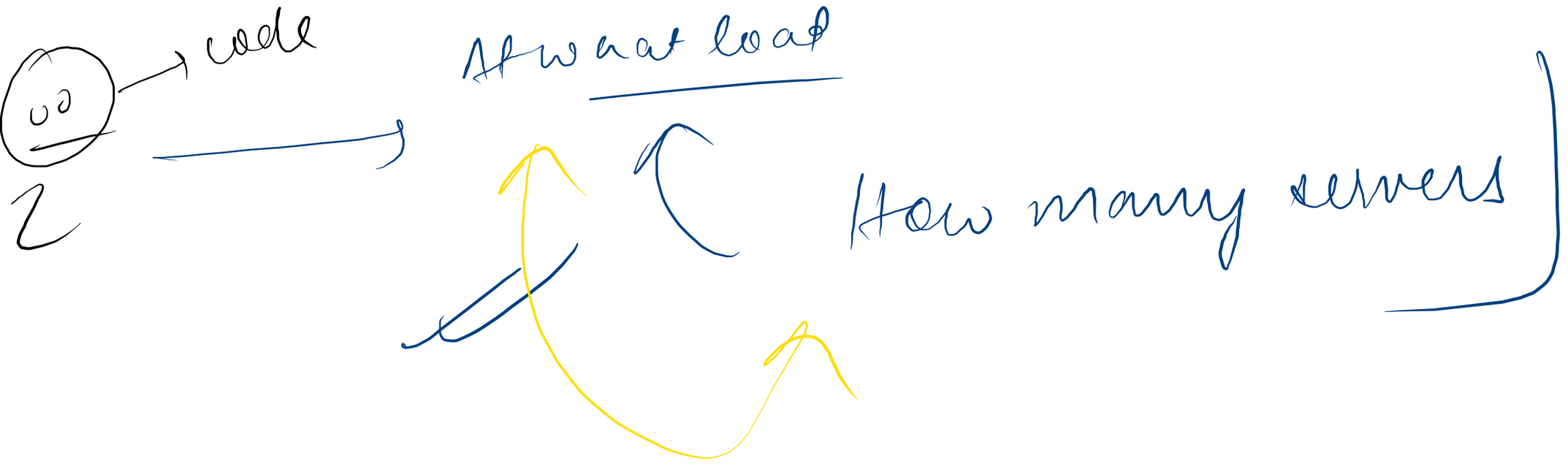

Black Friday
38000 T/s
45000 T/s
System
12AM
25K
30K
35K
40K
48K
Realment
went
down

No of transactions/second → 300-400

(×100 time

30,000 - 40000 / second

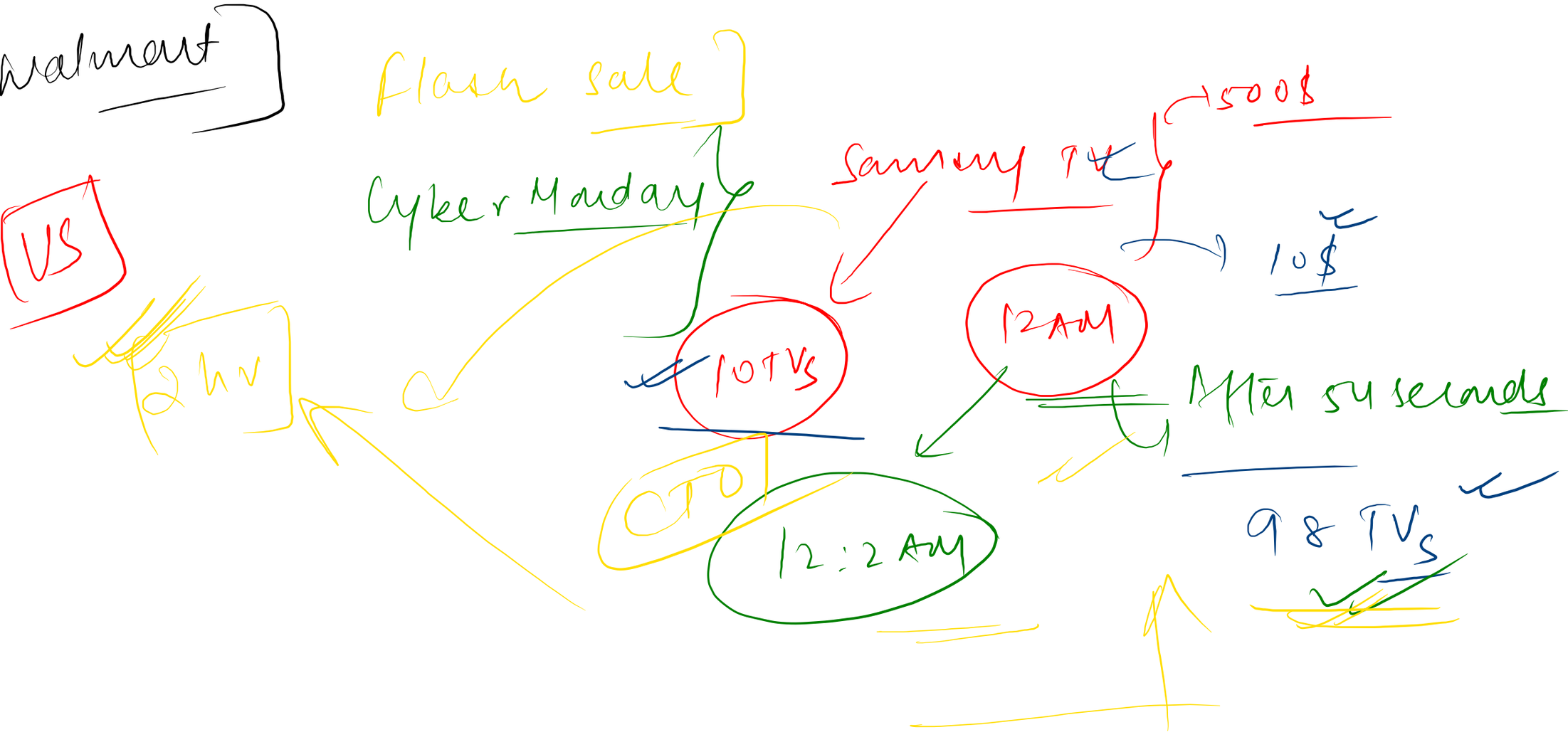
Holiday


code
At what load
How many servers

Private → Load Balancer

Load ↑ → server ↑

→ elastic scaling


Walmart
Flash sale
Cyber Monday
Samsung TV
500$
10$
VS
2 hr
10 TVs
12 AM
After 54 seconds
98 TVs
12:2 AM

7 - 9 PM

Test

5 mins

8:07

8:12 PM

Break time

Basic of the envelop Calculation :-

1 Byte = 8 bits

1 ASCII character → 1 Byte

ASCII : American standard code for Information Interchange

coding system

character → numerical code

A → 65

Twitter → 140 characters → 1 Byte

→ 140 Bytes

1 KB = 1024 Bytes $\approx 10^3$ Bytes.

1 MB = 10^6 Bytes = 10^3 KB → 1 Million Bytes

1 GB = 10^9 Bytes = 10^6 KB = 10^3 MB → 1 Billion Bytes

1 TB = 10^{12} Bytes = 10^9 KB = 10^6 MB = 10^3 GB

1 PB = 10^{15} Bytes = 10^{12} KB = 10^9 MB = 10^6 GB = 10^3 TB

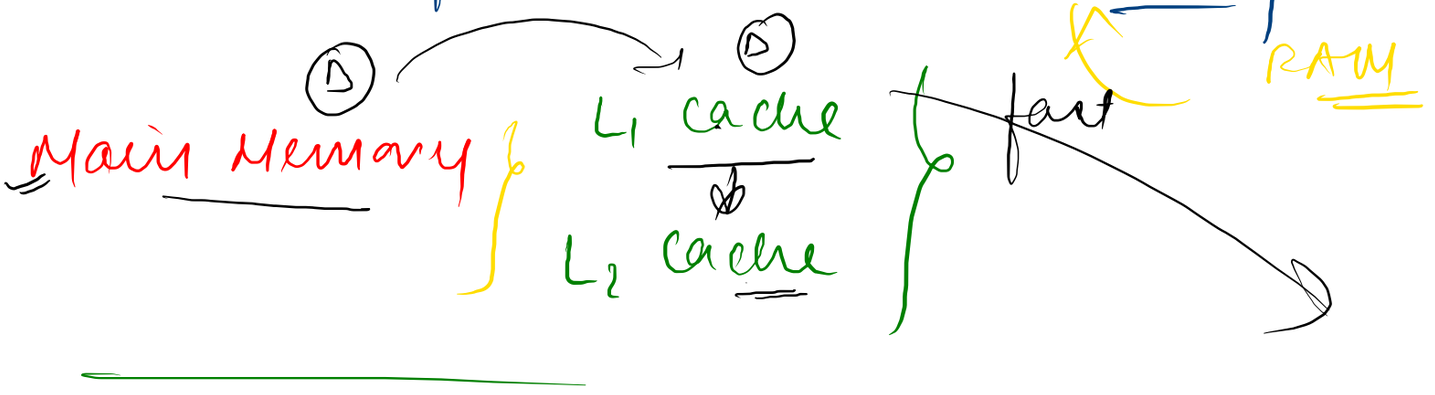
Units of time

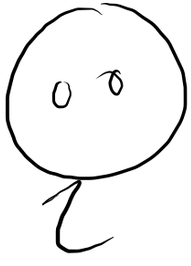
- seconds → s
- milliseconds → ms
- Microseconds → μs
- Nanoseconds → ns

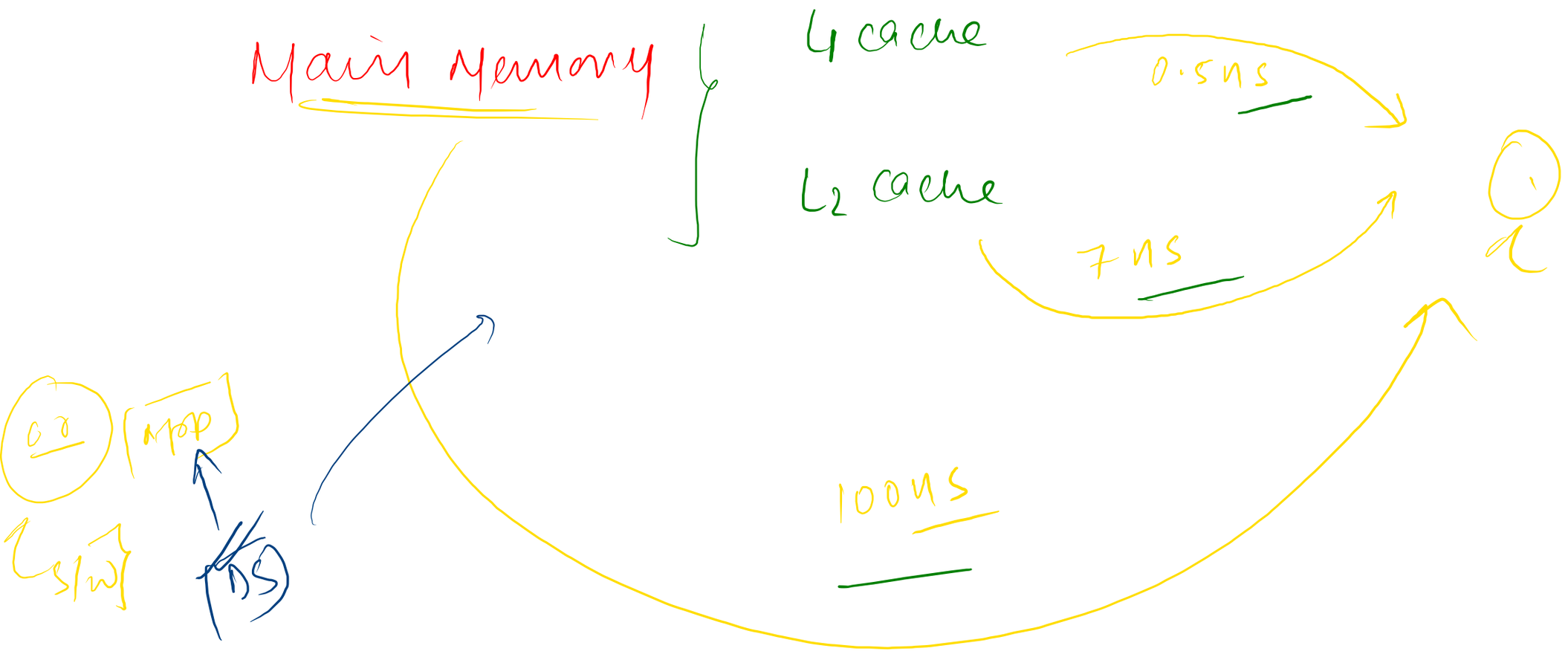
$$1 \text{ second} = 10^3 \text{ ms} = 10^6 \mu\text{s} = 10^9 \text{ ns}$$

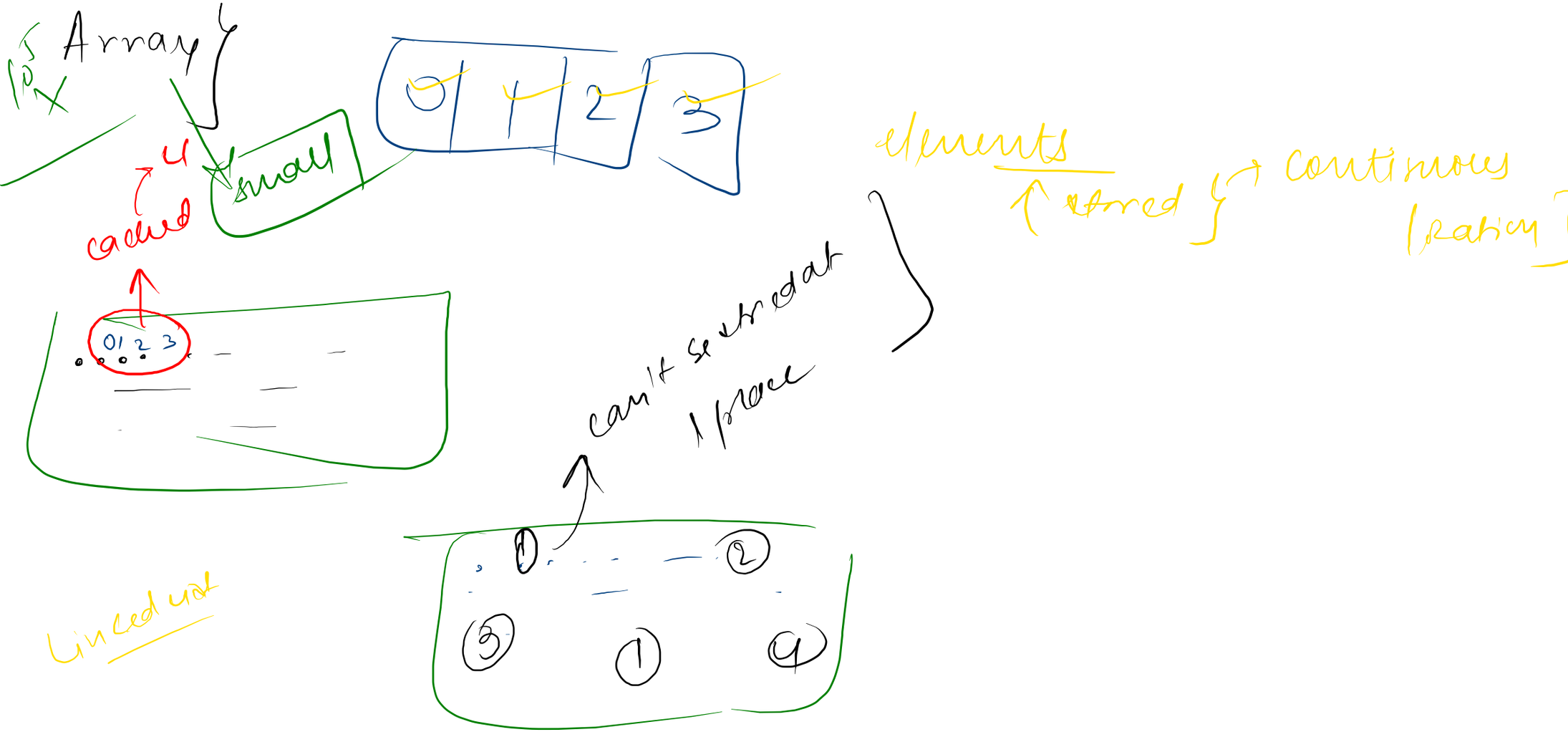
Common execution flow :-

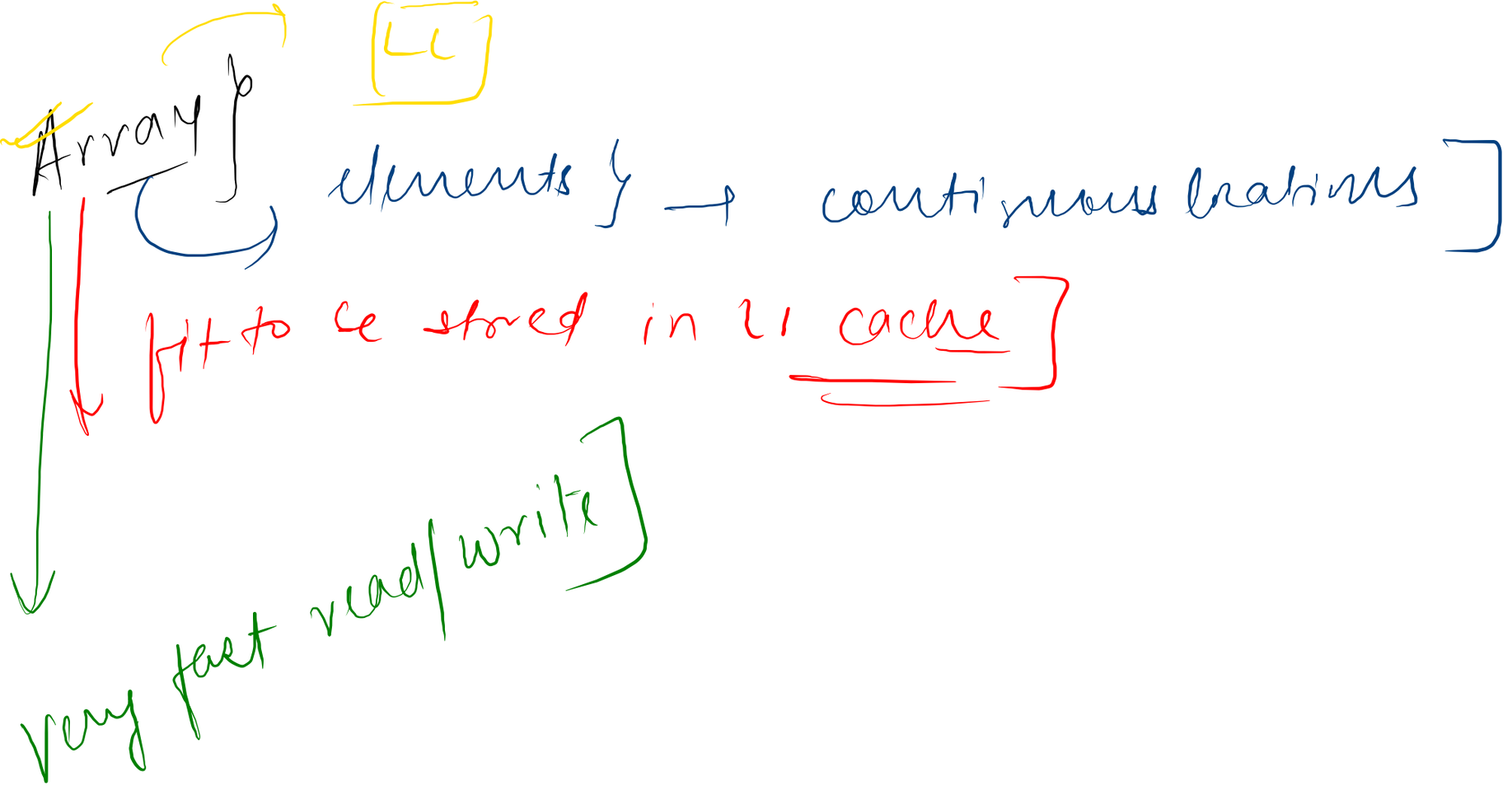
Read from main memory






Main Memory
L1 cache
0.5ns
L2 cache
7ns
100ns
mov
s/w
HS


Array
0 1 2 3
small
called
0 1 2 3
elements
stored
continuous
location
can't store data
1 place
1
2
3
1
4
Linked list

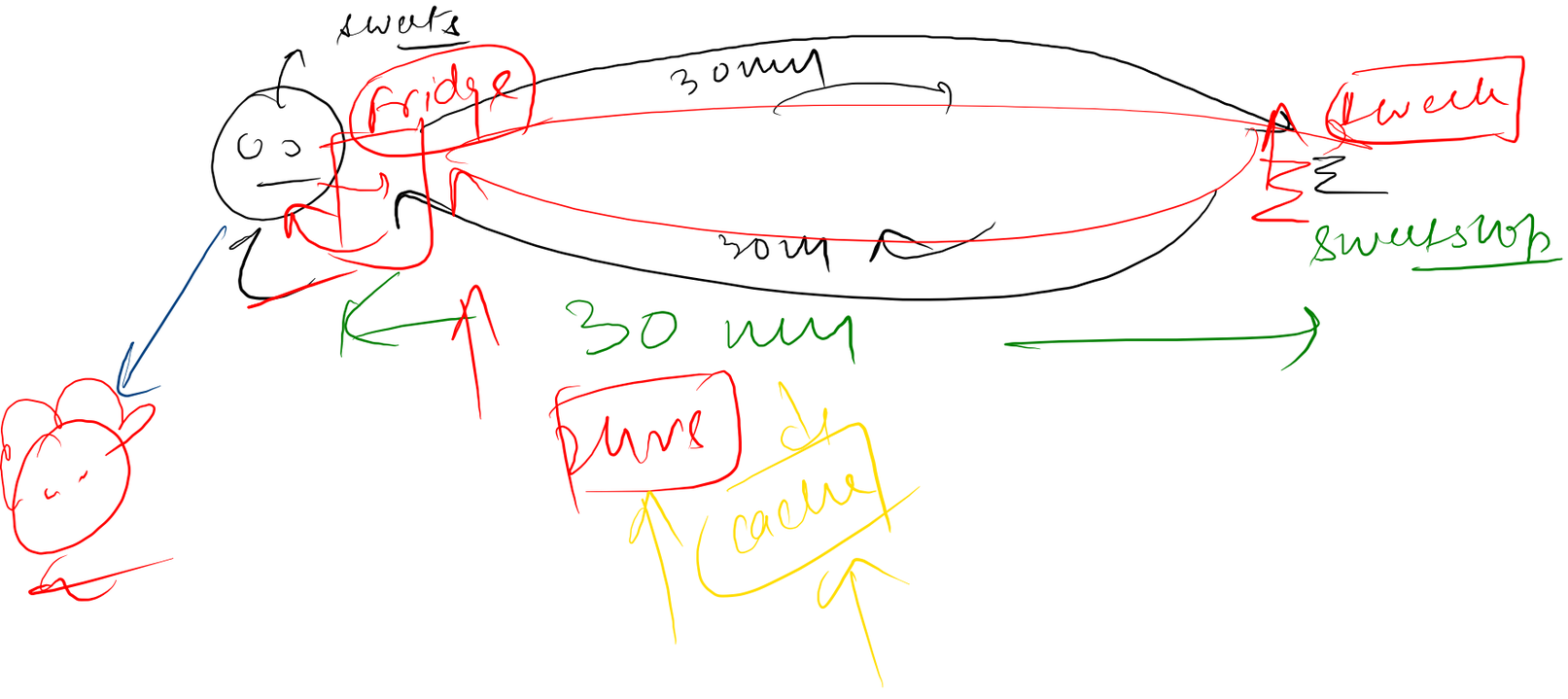

Array
elements → continuous locations
fit to be stored in L1 cache
very fast read/write

Cache ?

Repeated calculations

Time consuming

Cache


sweets
fridge
30 min
sweets
sweetshop
30 min
30 min
pluto
cactus

sweets

2 hrs

L_1 cache → 0.5 ns

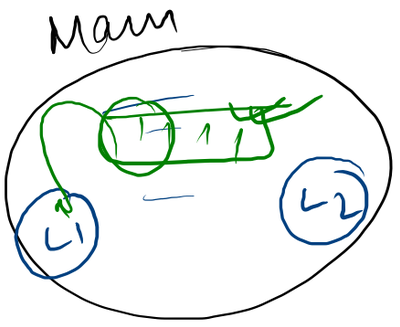
L_2 cache → 7 ns

Main Memory → 100 ns

Reading data

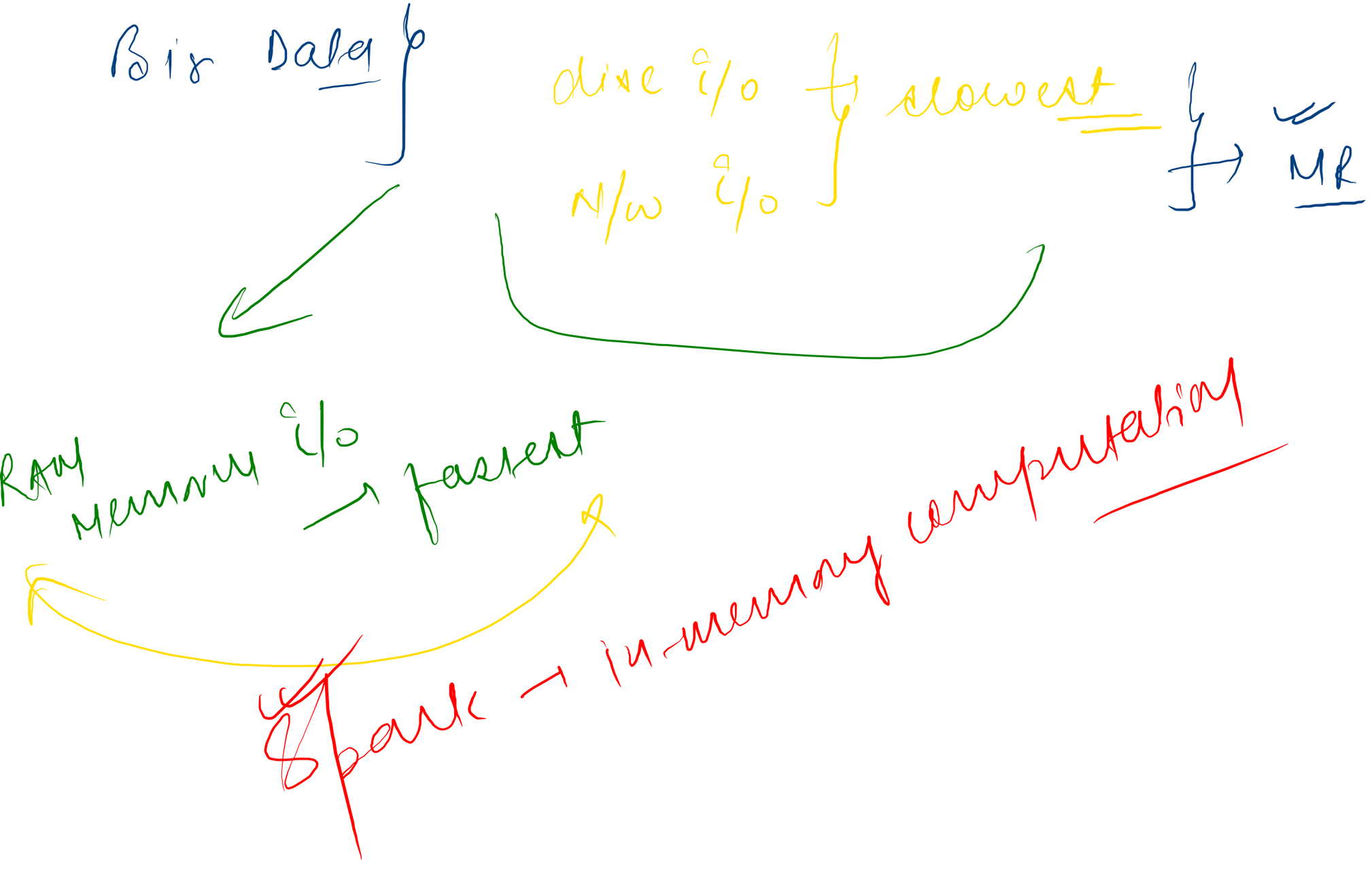
(1) ✓ (1MB) sequentially

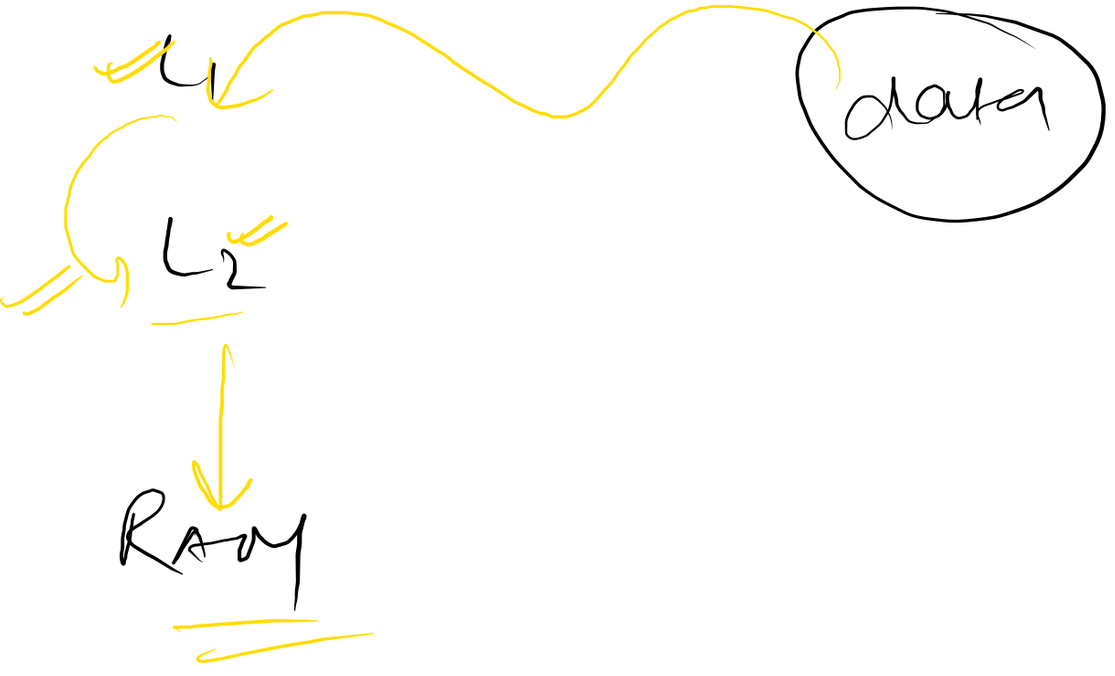
Main Memory → 250 µs | fastest



(2) ✓ Disc → 30 ms → slowest

(3) ✓ N/W → 10 ms → slower


Big Data
disc i/o
n/w i/o
slowest
MR
RAM memory i/o
fastest
Spark → in-memory computation


L1
data
L2
RAM

Availability → Measured → 9's

System

LinkedIn

99.9999

① 99 % Available ⟶ downtime 1% ⟶ years $\frac{1}{100} \times 365 = 3.65$ days

② 99.9% available

$= 0.365$ days

③ 99,99% available

$= 0.0365$ days

99, 9999 % available

$= 0.000365$ days

✓ 99% available

↳ ↓ 1% down

1% — 365

$\frac{1}{10} \times 365 = 3.65$ days.

99,9999% available

↳ 0.00001

company → Monitoring → App

↗ downtime

calculate

↑ total downtime

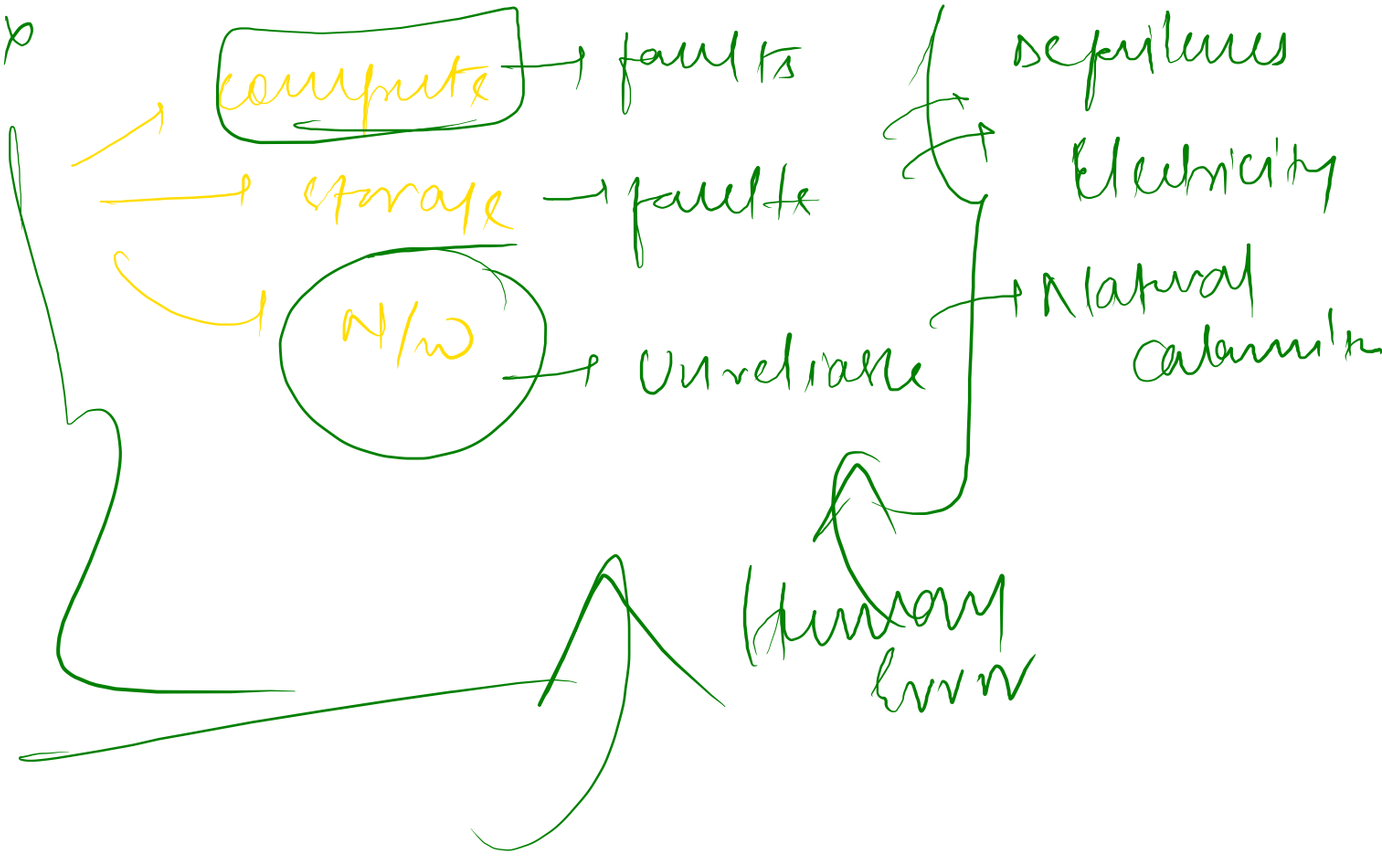
98%

99%

99.9

%

Companies } complex mix



Application

Twitter

Daily Active User

DAU = 150 Million

storage → 10 years

avg 2 tweets / day

10% have media content

QPS (Queries per seconds)

=

QPS = DAU × 2 per day

= 150 × 2 per day

= $\frac{150 \times 2}{24}$ per hr

= $\frac{150 \times 2}{24 \times 60 \times 60} \times 10^6$ /seconds

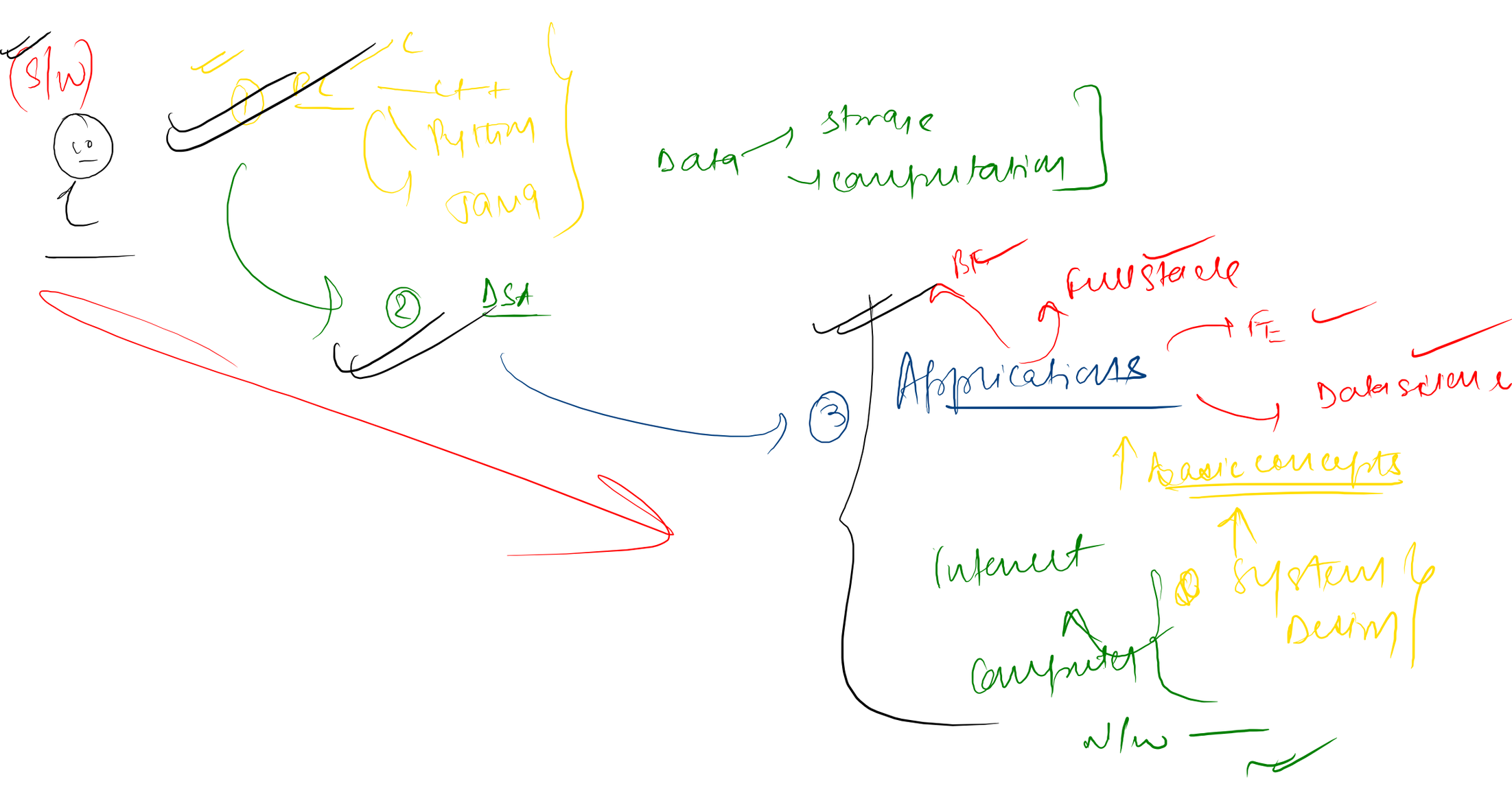
Approximate ≈ 8000 QPS

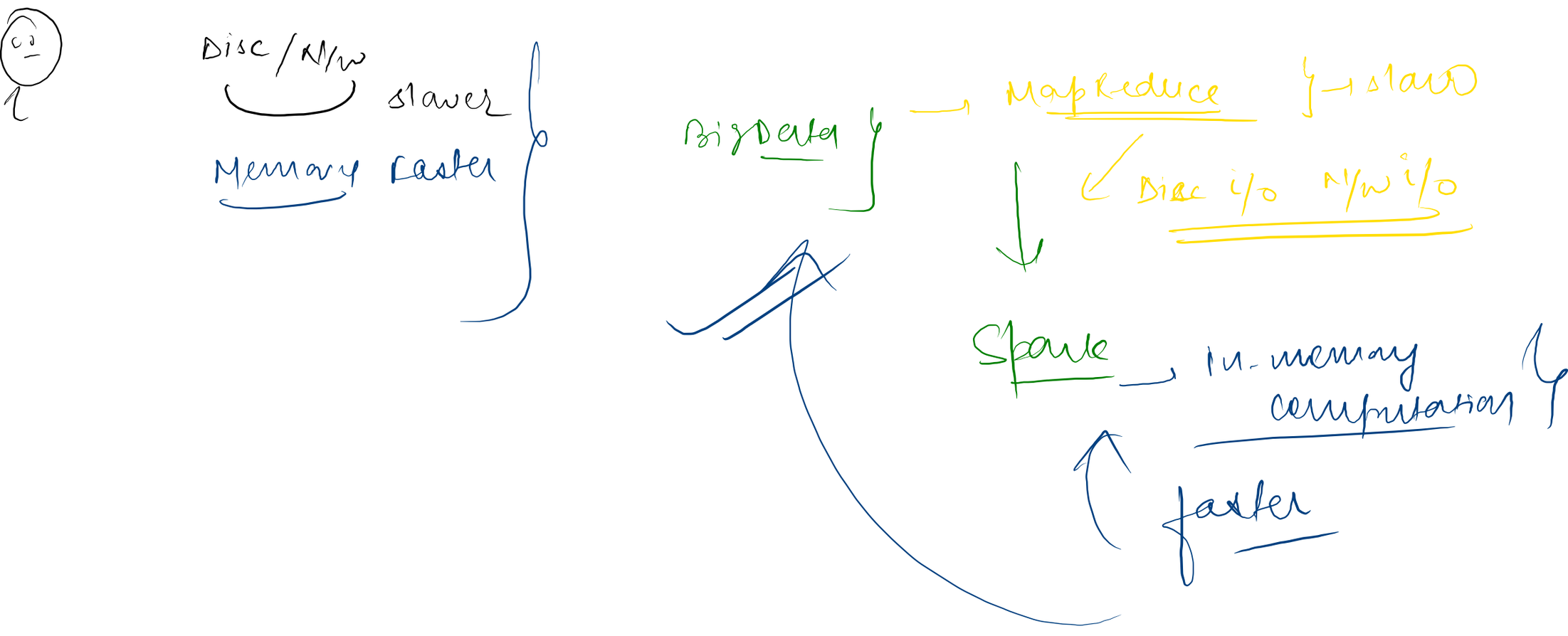
QPS → Queries / seconds

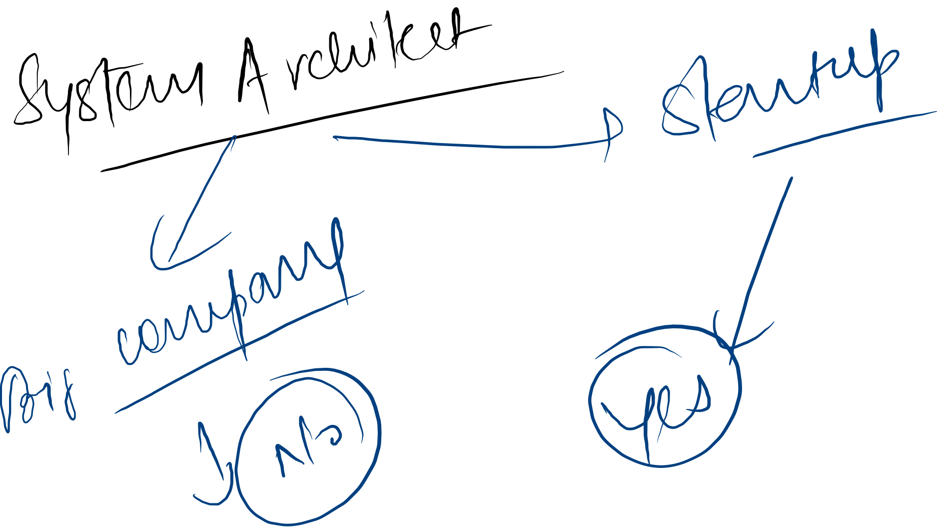
QPS

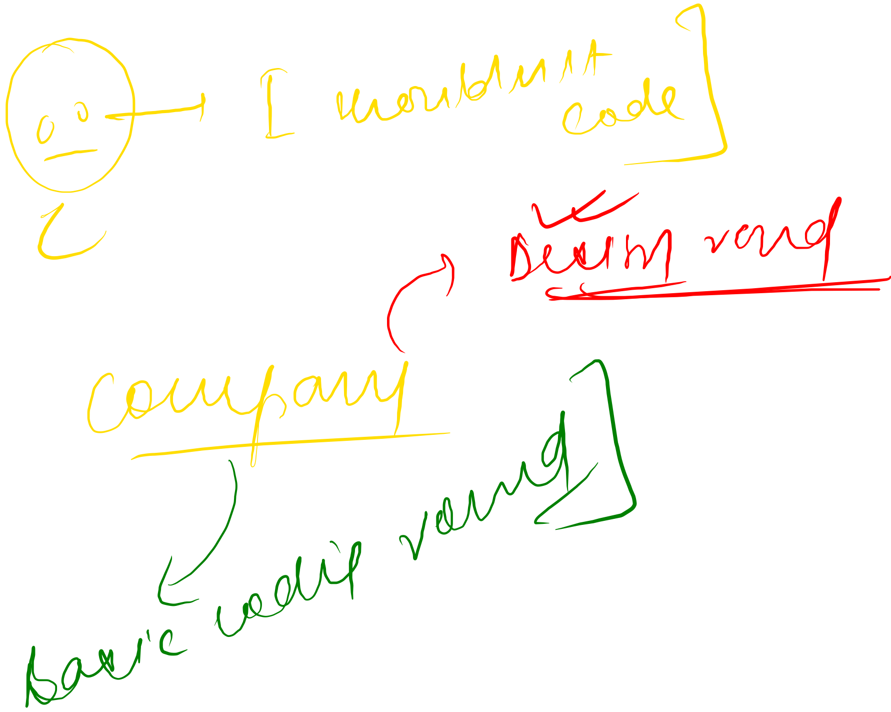
Peak QPS = 2 x QPS = 2 x 7000 = 14000

7 – 9PM


(S/W)
C
C++
Python
Java
Data
storage
computation
DSA
BE
Full stack
FE
Applications
Data science
Basic concepts
Internet
Computer
System
Design
n/w


Disc / N/w
slower
Memory
faster
BigData
MapReduce
slow
Disc i/o
N/w i/o
Spark
in-memory
computation
faster


System Architect
Big company
No
Startup
Yes


[maintain code]
Design round
company
basic coding round