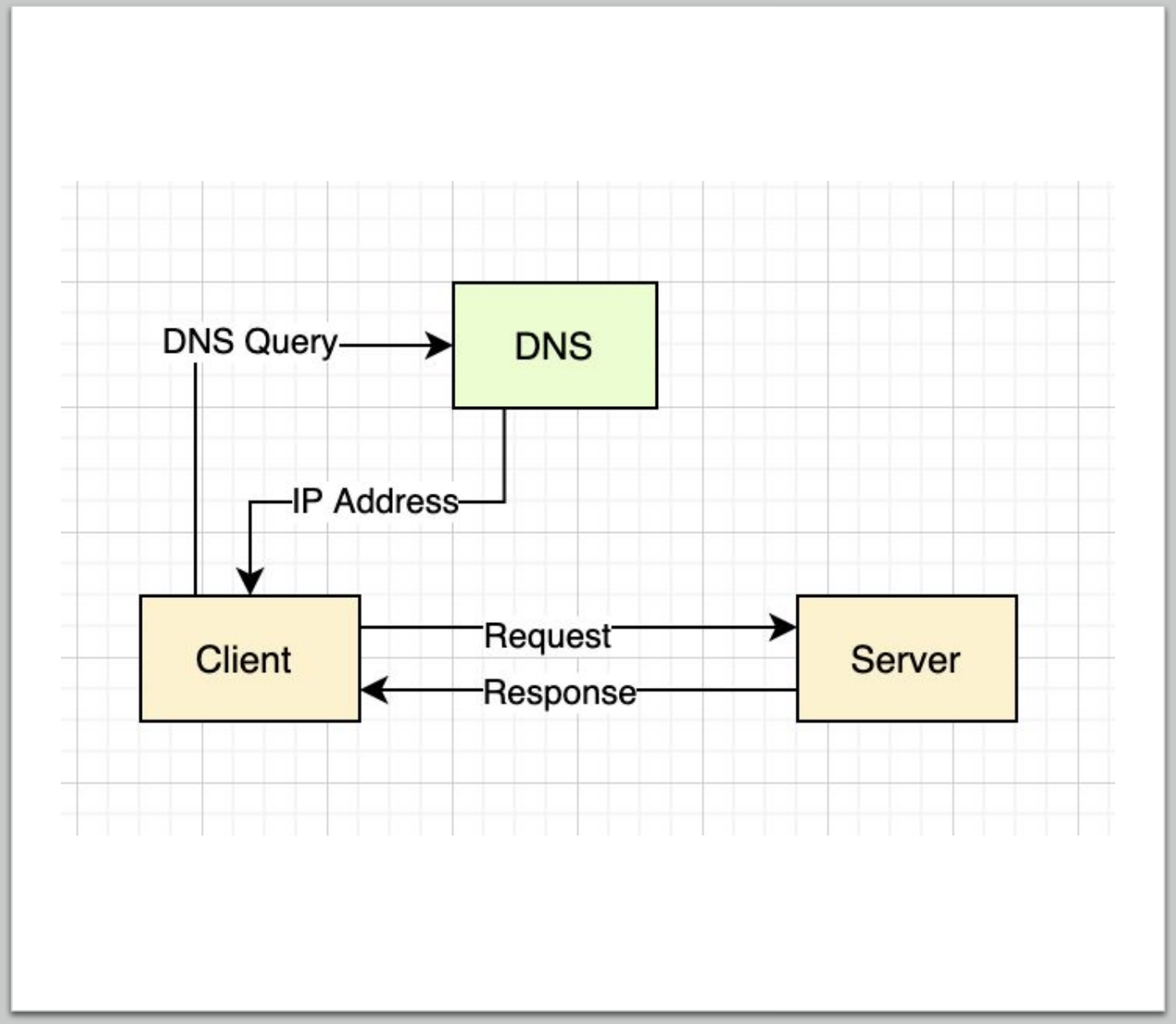
System Design Fundamentals

Anshu Sharma – CTO, Uolo

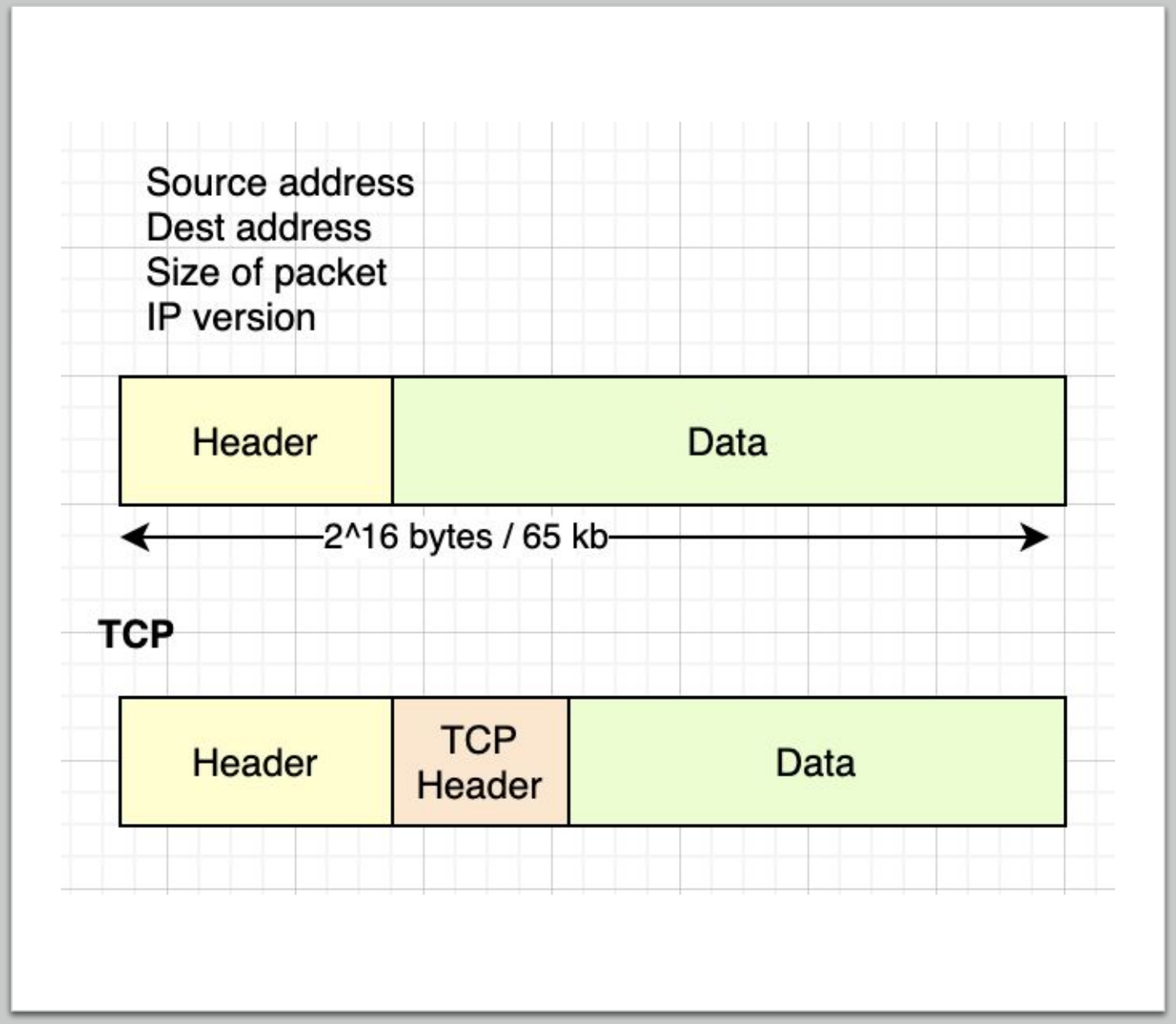
Client—Server Model

- Client
- Server
- IP Address
- Port
- DNS (domain name system)



Network Protocols

- Protocol
- IP
  - Rules for sending messages over Internet
- IP Packet
- TCP
  - Send IP packets in ordered, reliable way
  - Communication happens over TCP connection
- HTTP



Network Protocols (contd)

- Protocol
- IP
  - Rules for sending messages over Internet
- IP Packet
- TCP
  - Send IP packets in ordered, reliable way
  - Communication happens over TCP connection
- HTTP

HTTP Requests typically have the following schema:

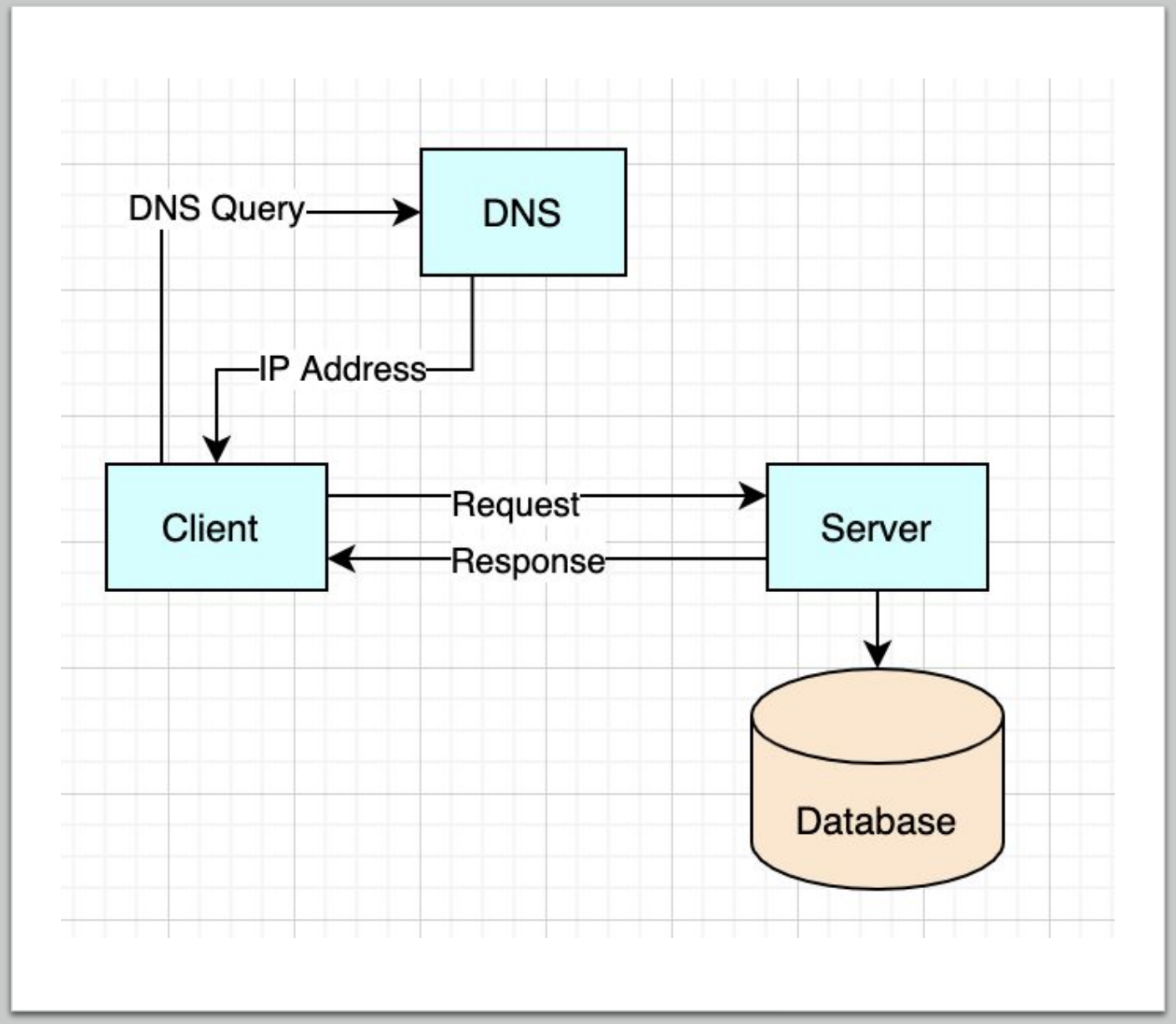
```
host: string (example: economictimes.com)
port: integer (example: 80 or 443)
method: string (example: GET, PUT, POST, DELETE, OPTIONS or PATCH)
headers:  pair list (example: "Content-Type" => "application/json")
body: opaque sequence of bytes
```

HTTP Responses typically have the following schema:

```
status code: integer (example: 200, 401)
headers:  pair list (example: "Content-Length" => 1238)
body: opaque sequence of bytes
```

Storage

- Databases
- Persistent Storage
  - HDD / SSD / Cloud
- Memory
  - RAM



Relational Databases

- **Relational Database / SQL database**
- **Non-Relational Database / NoSQL database**
- **Database Indexes**
- **ACID Transaction**
  - 1) Atomicity 2) Consistency 3) Isolation 4) Durability

ACID Properties

- **Atomicity**
  - All the changes are performed, or none of them are..
- **Consistency**
  - Data is in a consistent state when a transaction starts and when it ends.
- **Isolation**
  - Transactions that run concurrently appear to be serialized.
- **Durability**
  - Completed transactions are recorded in non-volatile memory

Latency And Throughput

- Latency
  - Time taken to complete certain operation
- Throughput
  - Number of operations system can handle per unit time (RPS or QPS)

Reading 1 MB from RAM: 250 µs (0.25 ms)

Reading 1 MB from SSD: 1,000 µs (1 ms)

Transfer 1 MB over Network: 10,000 µs (10 ms)

Reading 1MB from HDD: 20,000 µs (20 ms)

Inter-Continental Round Trip: 150,000 µs (150 ms)

Availability (Uptime)

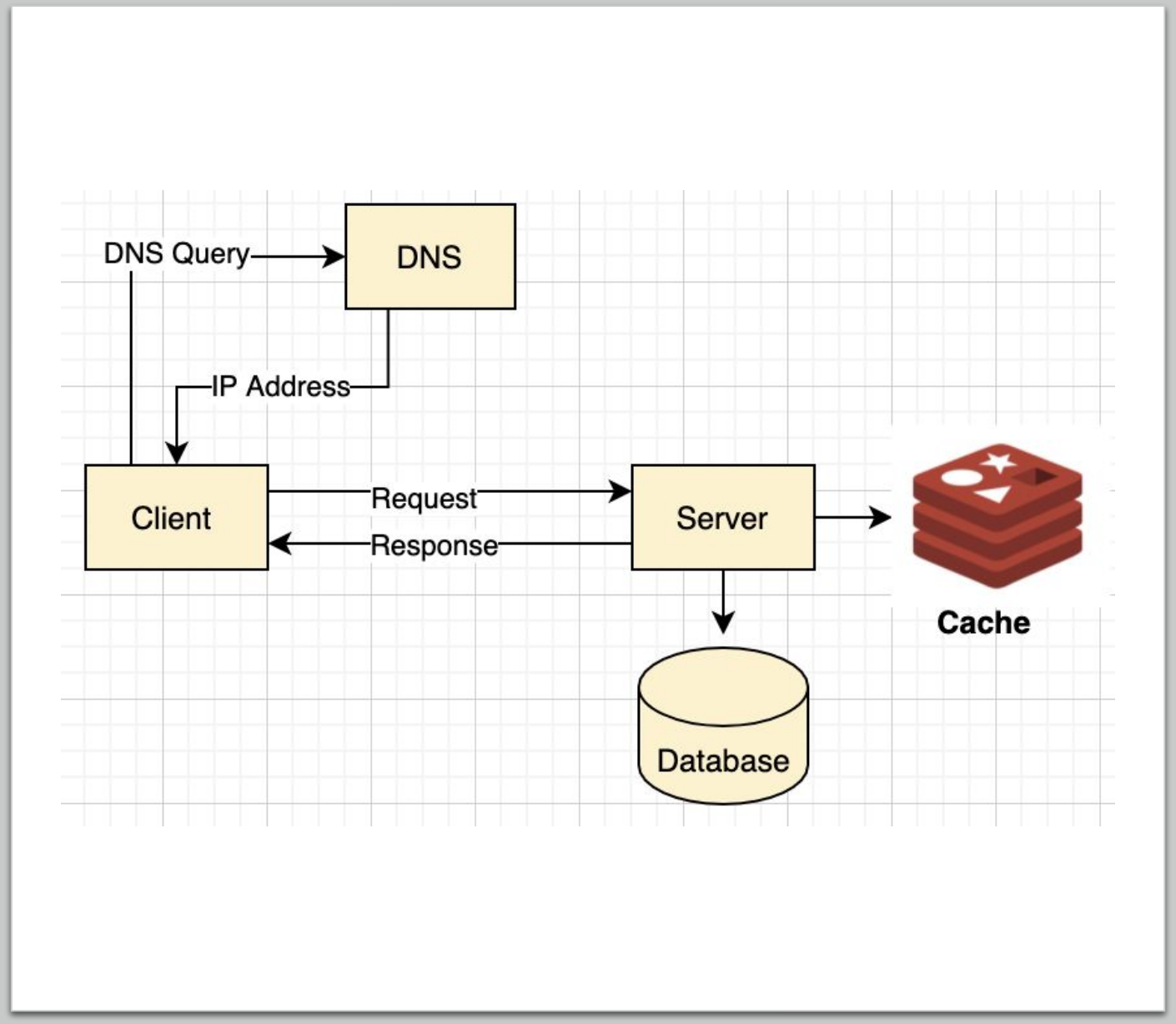
- Availability
- High Availability
- Nines
- Redundancy
- SLA
- SLO

Downtimes for different availabilities

- 99% (two 9s): 87.7 hours
- 99.9% (three 9s): 8.8 hours
- 99.99%: 52.6 minutes
- 99.999%: 5.3 minutes

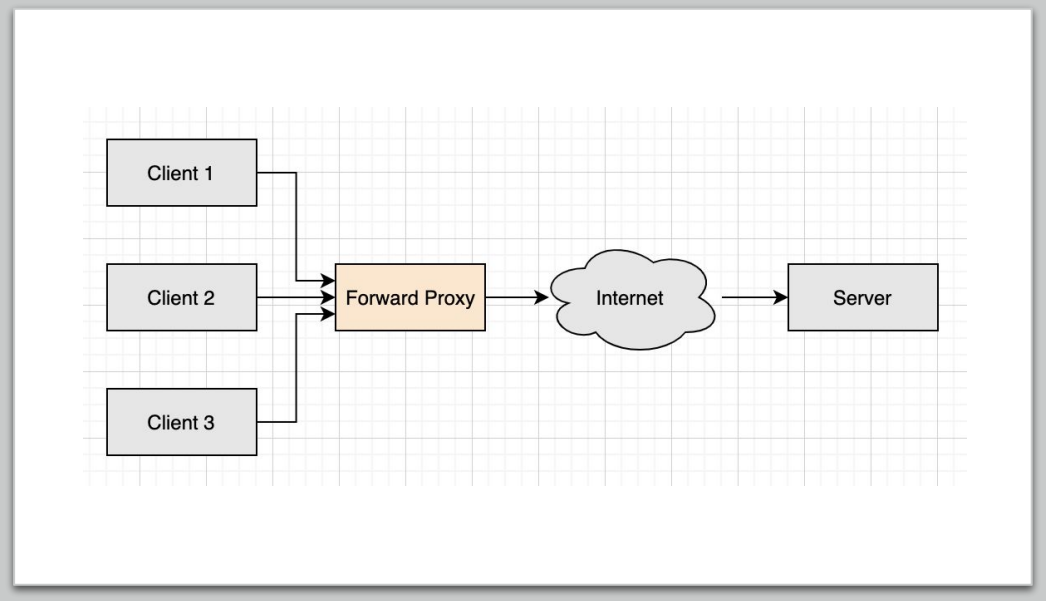
Caching

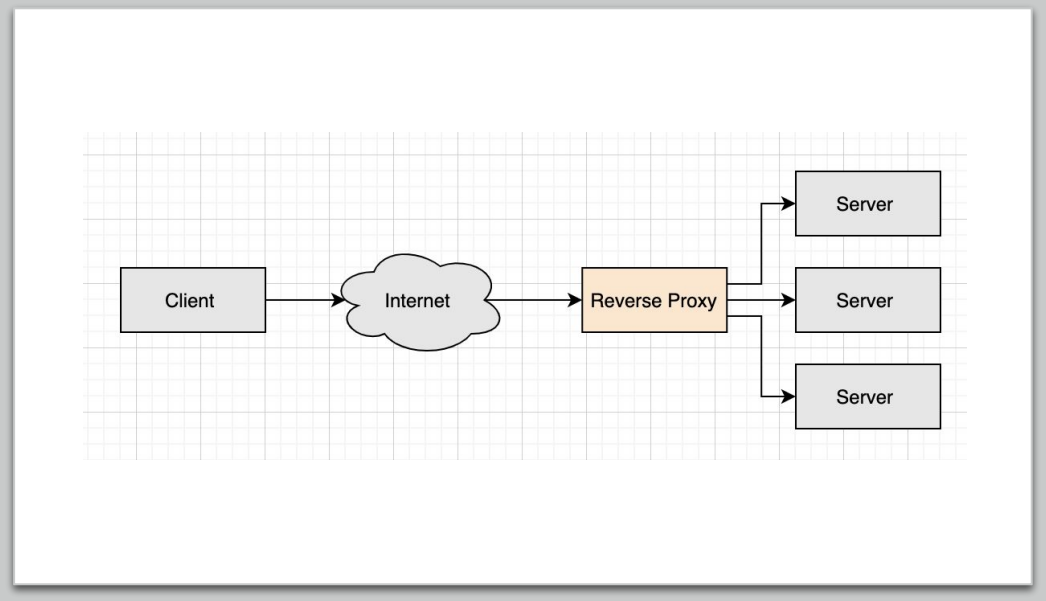
- Cache
- Cache Hit
- Cache Miss
- Cache Eviction Policy
- Content Delivery Network



Proxies

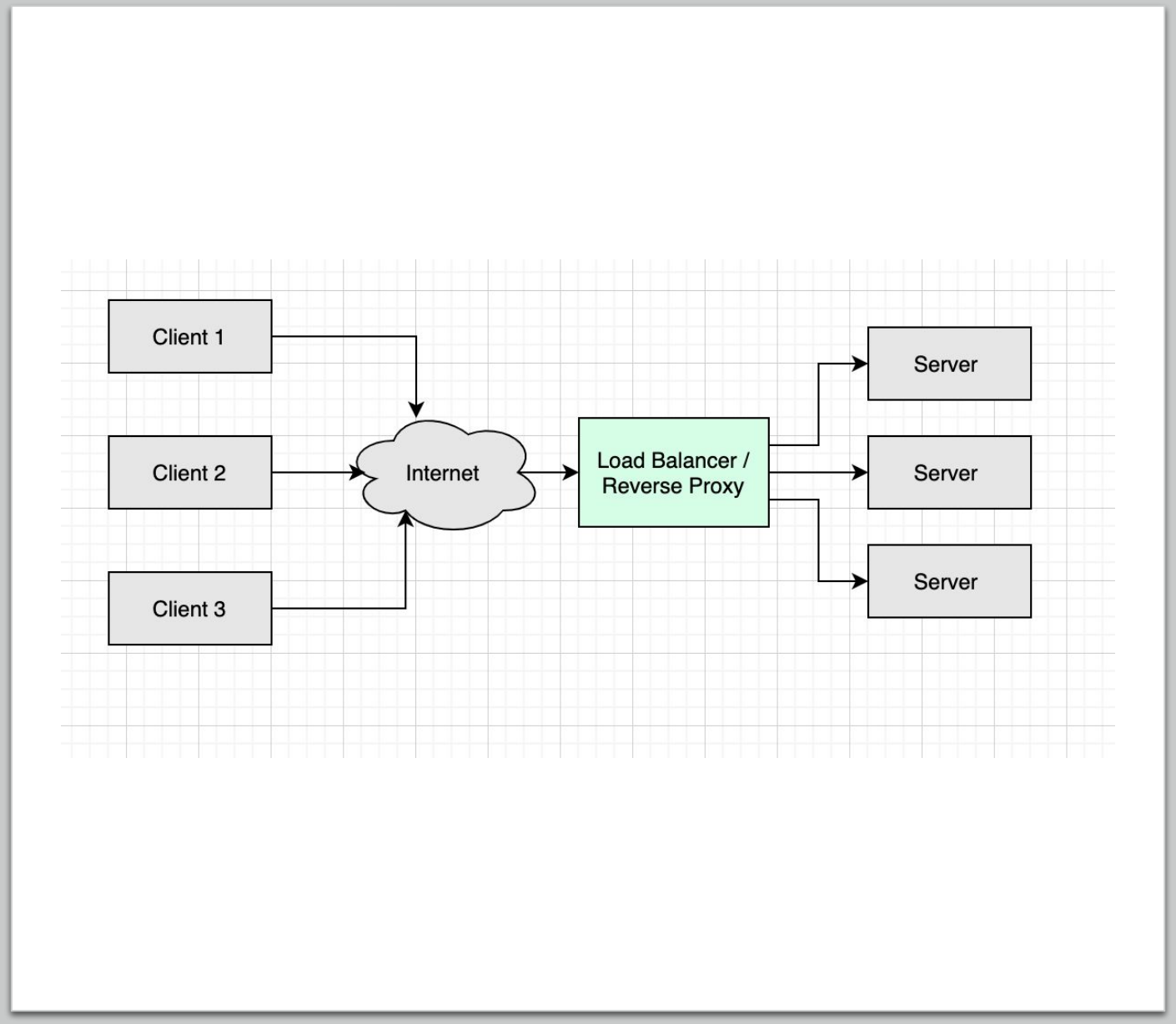
- Forward Proxy
- Reverse Proxy





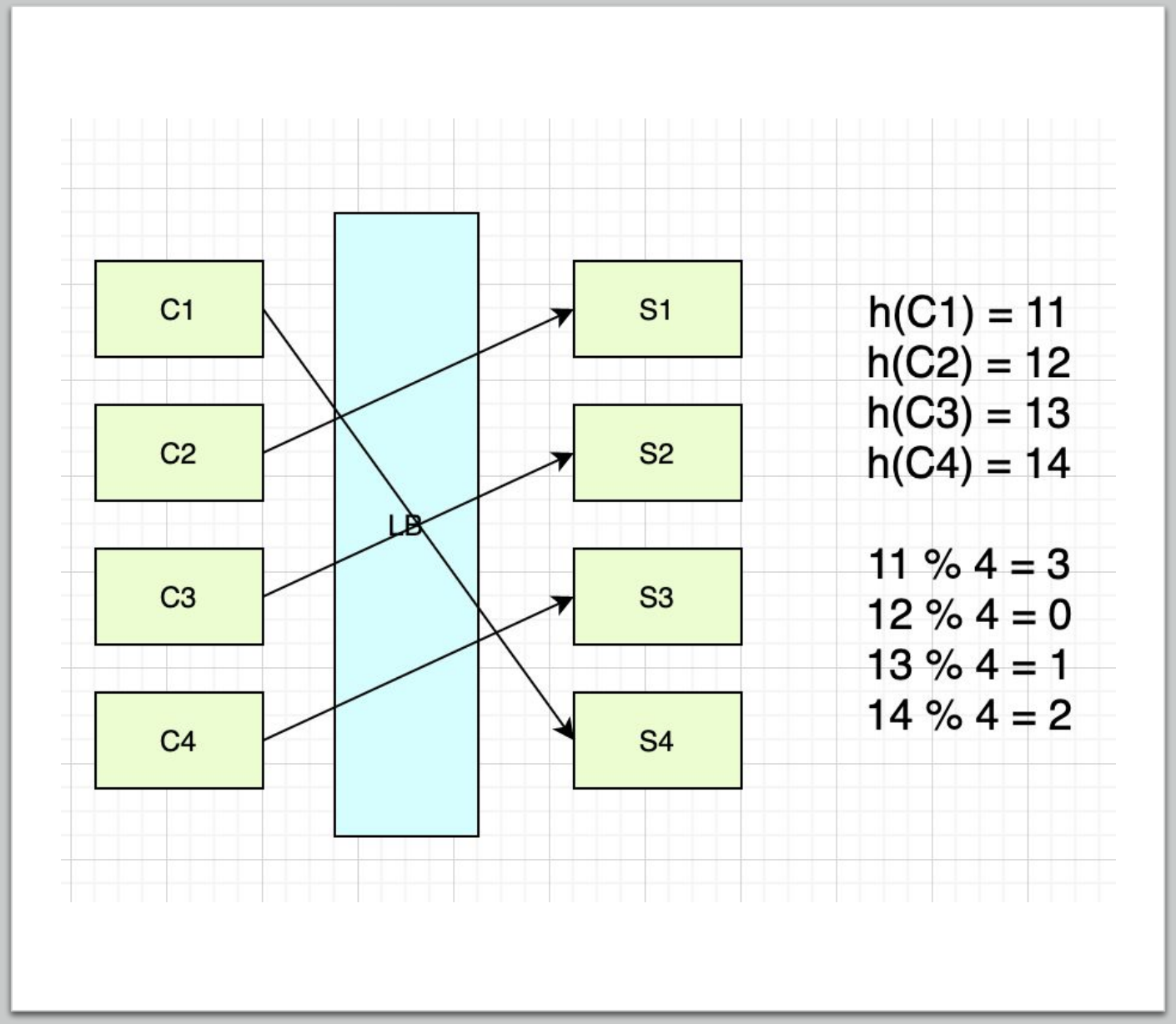
Load Balancers

- Vertical Scaling vs Horizontal Scaling
- Load Balancer
- Server-Selection Strategy
- Hot Spot
  - Requests(a-i) => Server 1
  - Requests(j-r) => Server 2
  - Requests(s-z) => Server 3



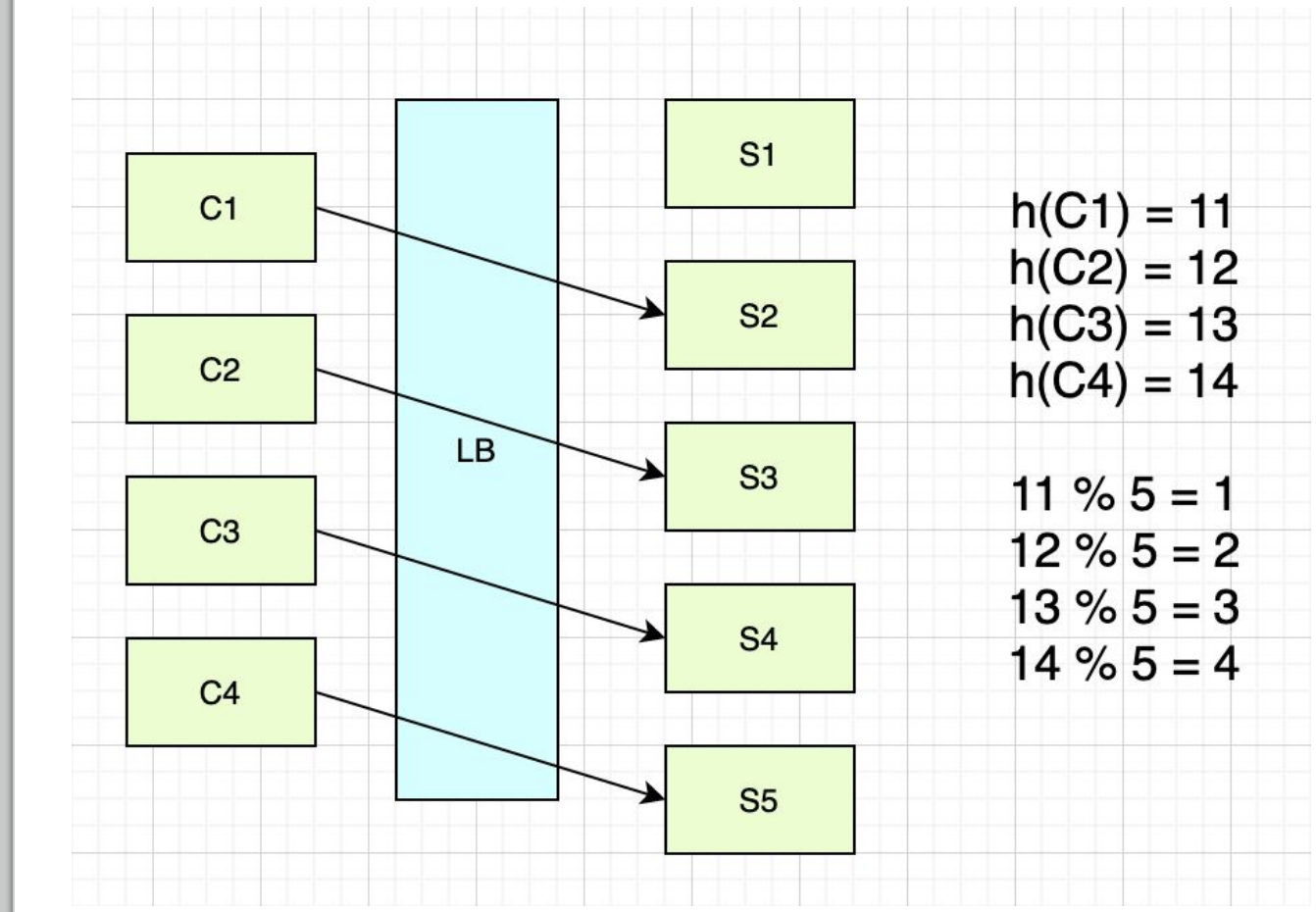
Data Sharding

- **Conventional Hashing**
  - hashF(key) = hash value
- Consistent Hashing
- Rendezvous Hashing



Hashing

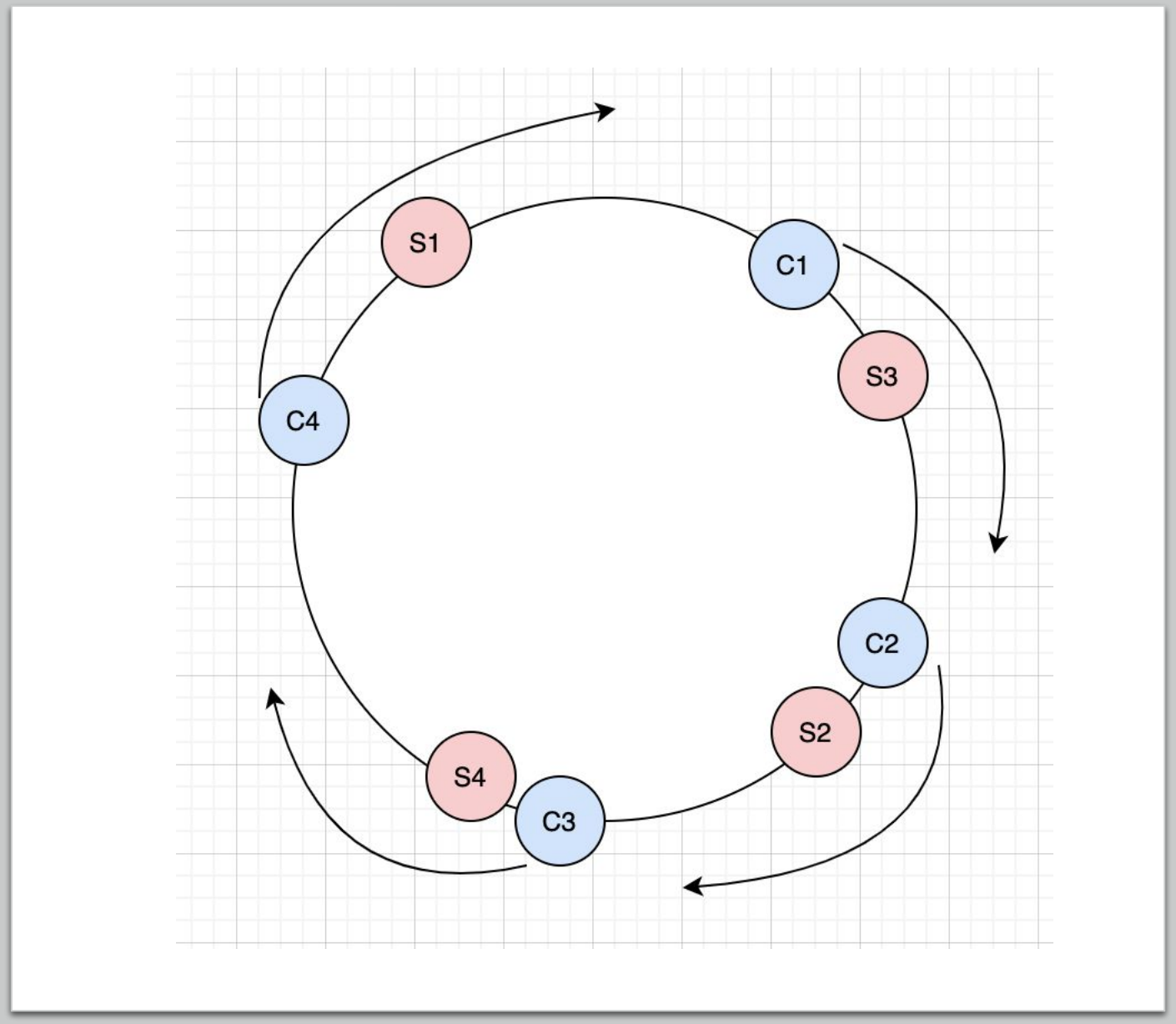
- **Conventional Hashing**
- Consistent Hashing
- Rendezvous Hashing



Hashing (Contd.)

- **Consistent Hashing**

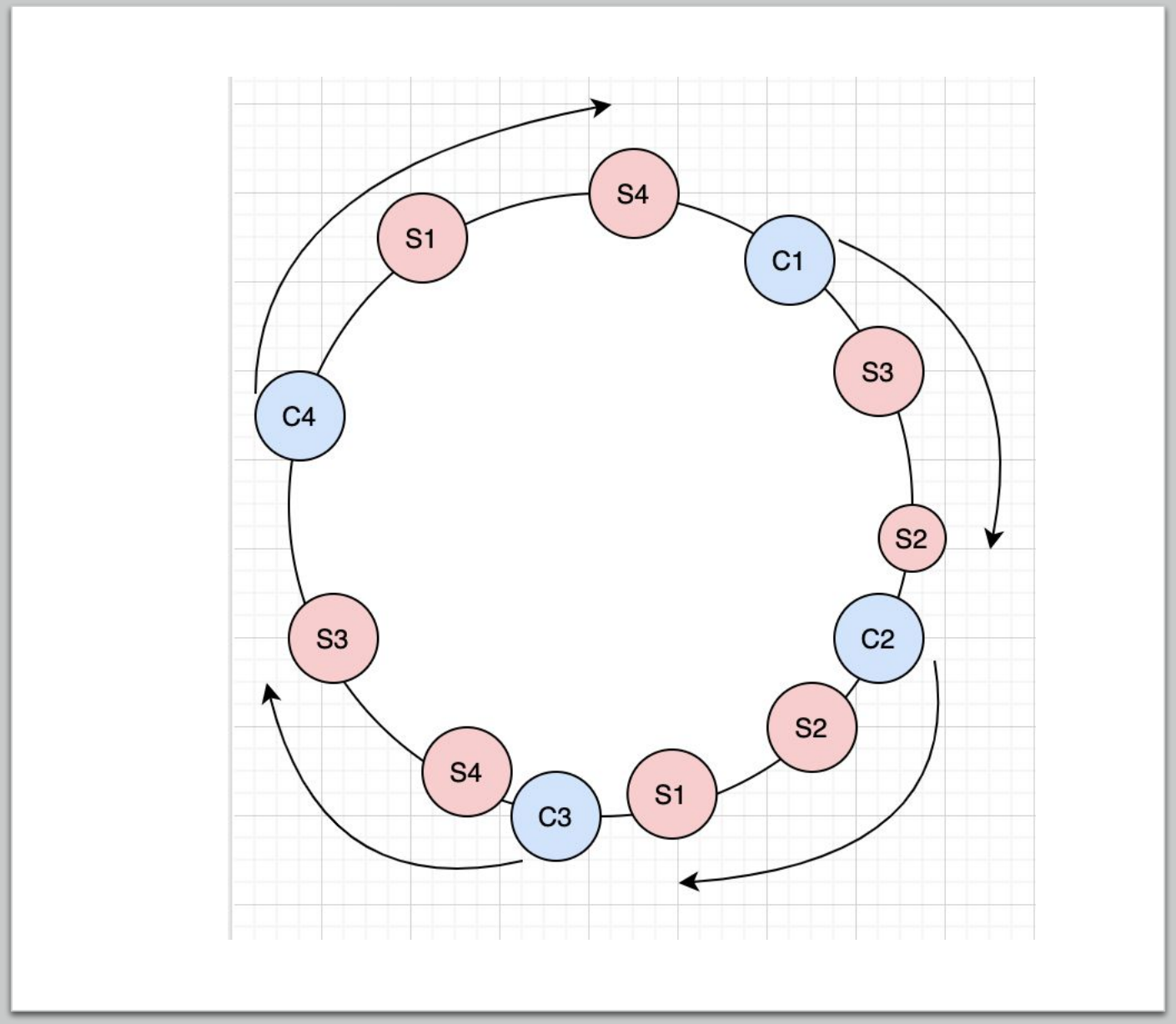
- Clients and Servers are mapped on a circle after hashing
- Addition/removal impacts only 1 or few servers
- Redundancy achieved by mapping servers through multiple hash functions



Hashing (Contd.)

- **Consistent Hashing**

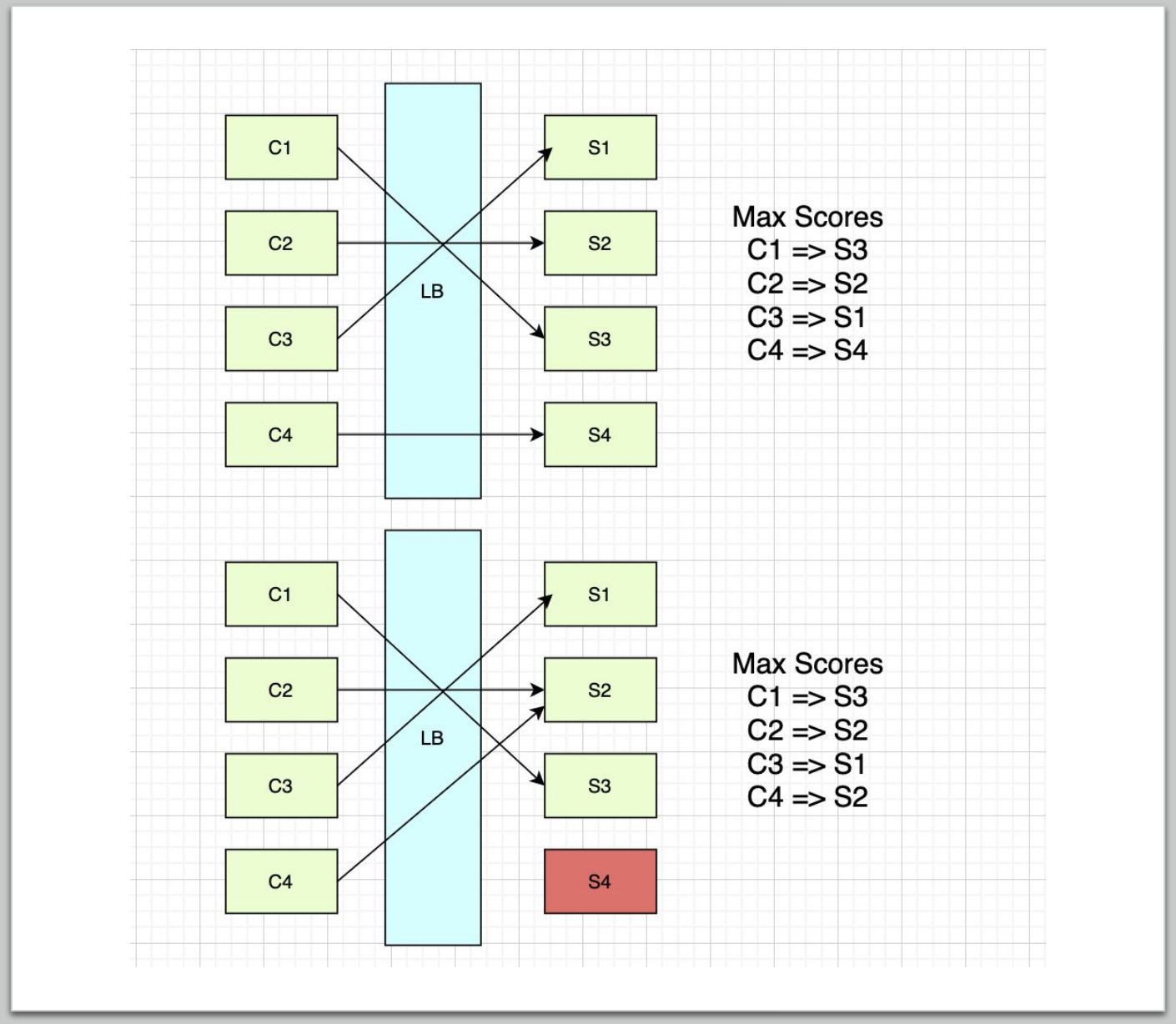
- Clients and Servers are mapped on a circle after hashing
- Addition/removal impacts only 1 or few servers
- Redundancy achieved by mapping servers through multiple hash functions



Hashing (Contd.)

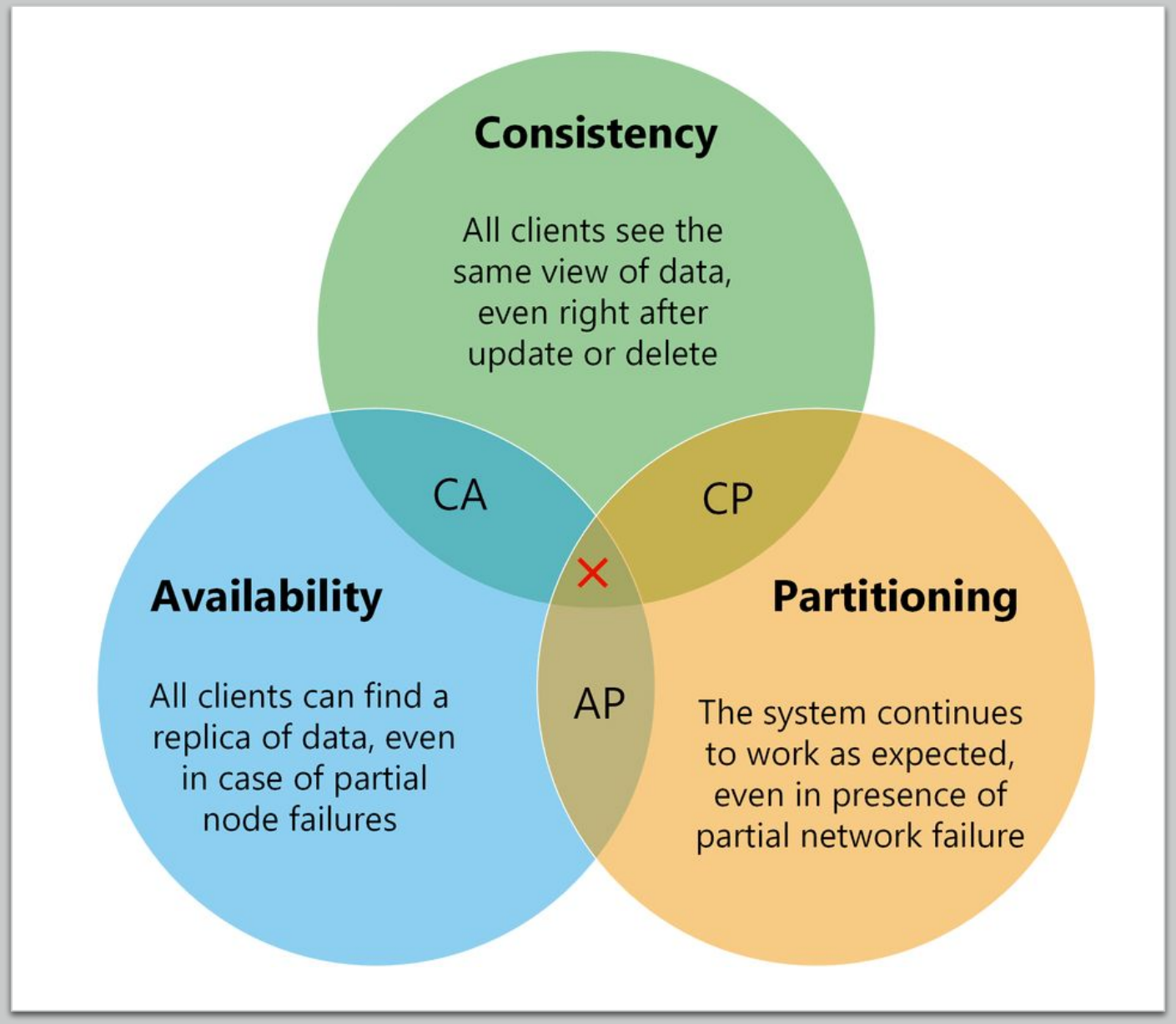
- **Rendezvous Hashing**

- Calculate server score for every client.
- Request is redirected to server with max score



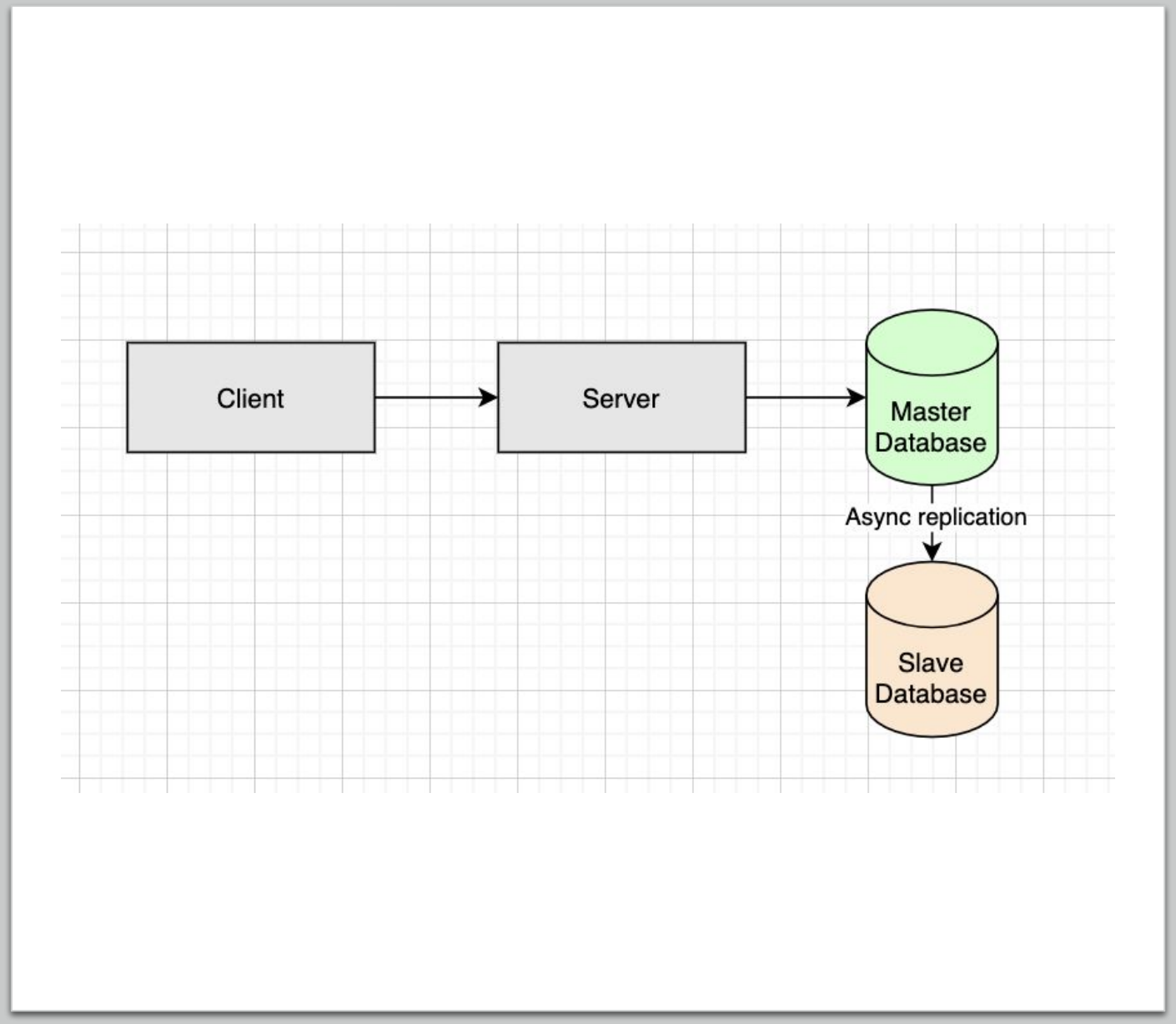
CAP Theorem

- Consistency
- Availability
- Partitioning Tolerance



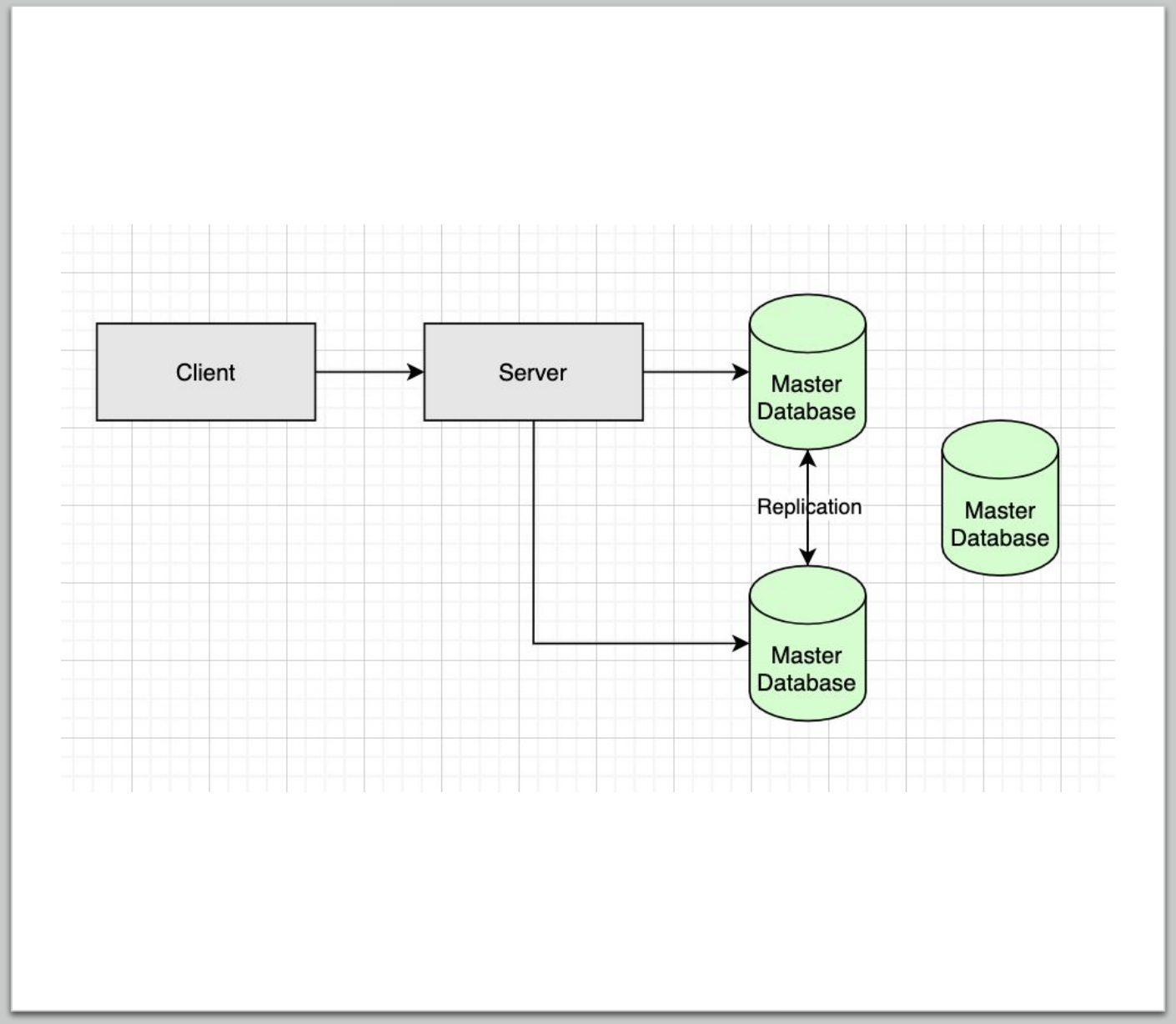
Data Replication

- Master – Slave
  - Application writes to master & data is replicated to slave



Data Replication

- Master – Master
  - Application can write to multiple masters
- Eventual consistency / Strict consistency



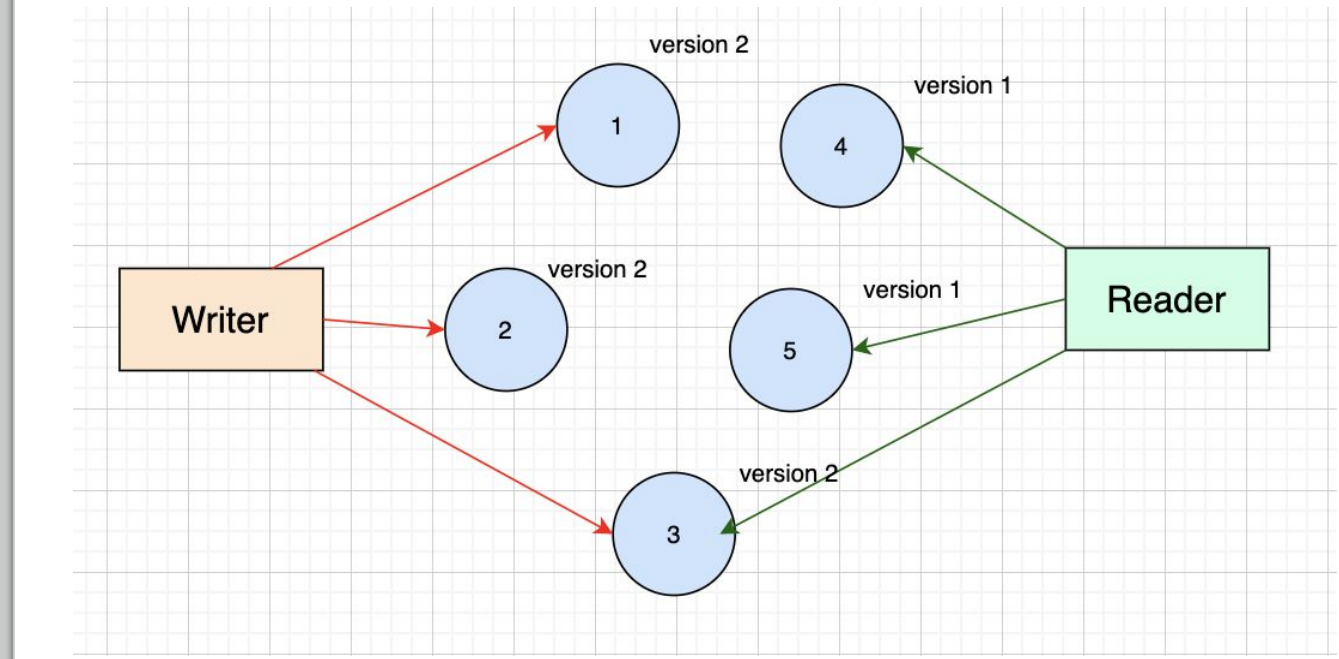
Quorum Consensus

- Allows for strict consistency without writing to all masters
- Keep version along with records

  Key : Value : version
  K1 : a : 1
  K2 : b. : 2

R + W > N

Select R & W to optimize for reads or writes



Thank You