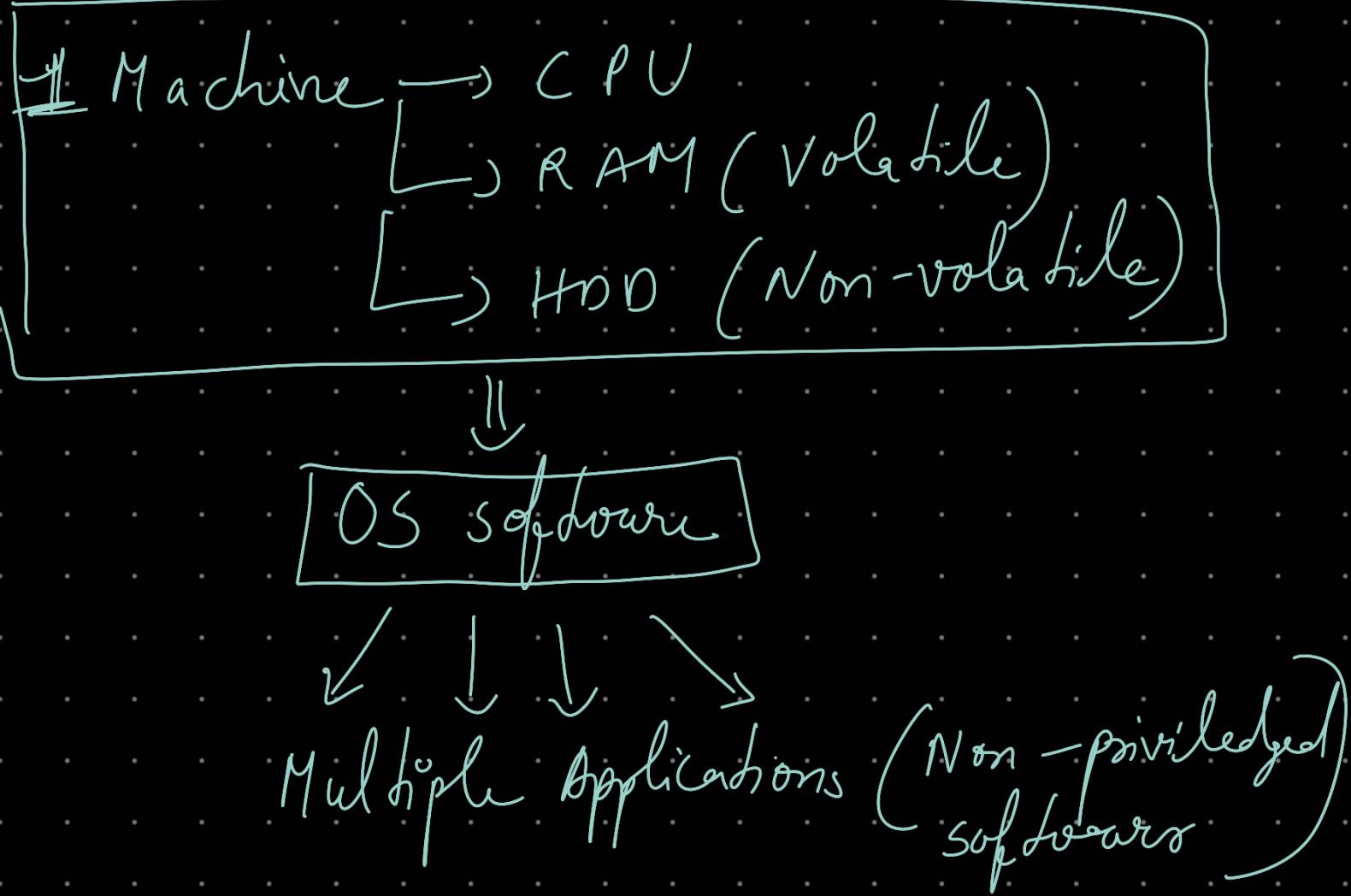


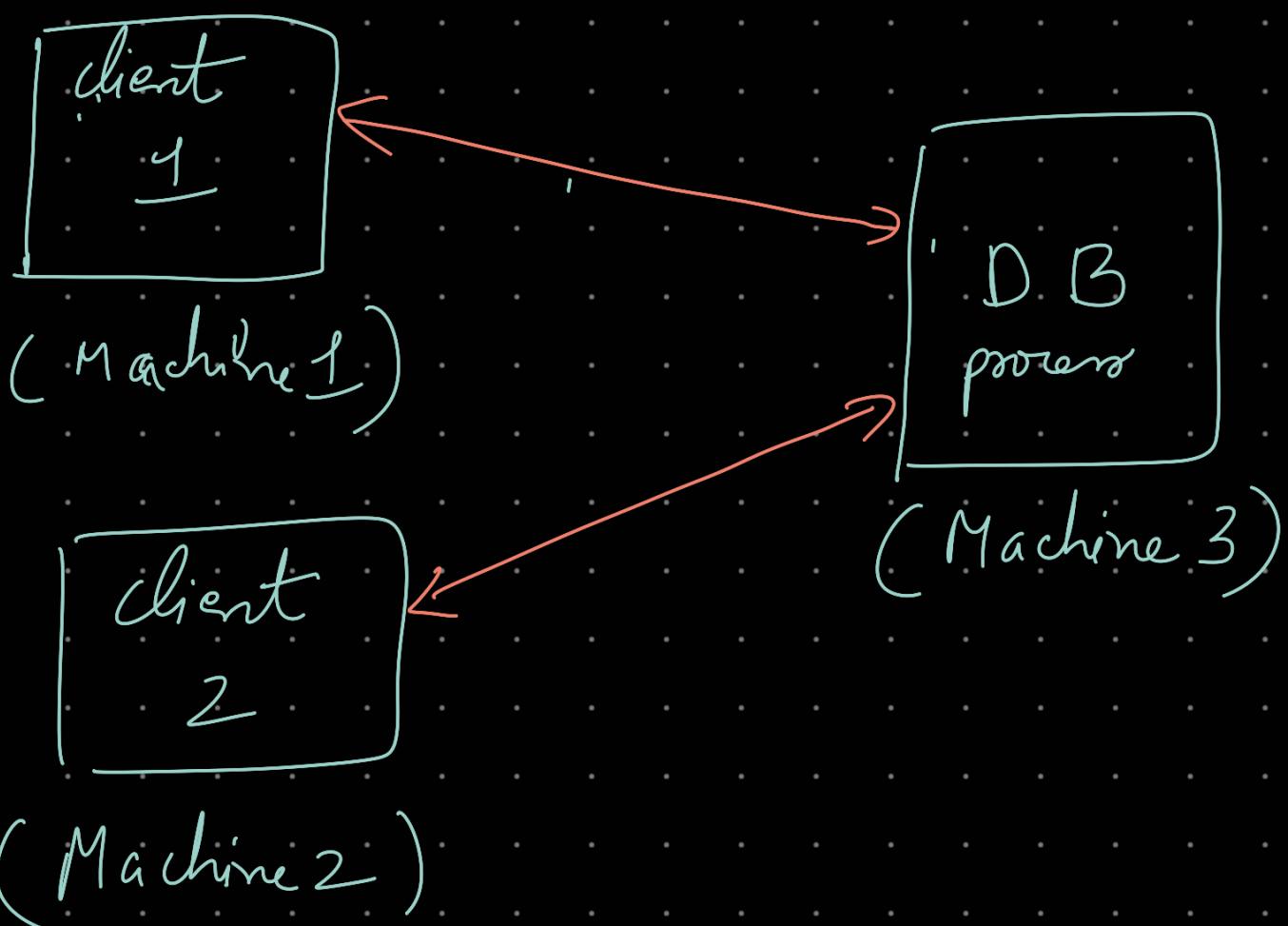
[Early days Computing and Two-tier Architecture]



- * Early days Internet used to be very small. E.g. → A University network



Two tier architecture



- * Client should have secrets of DB and all drivers to connect to DB
- * This was original client-server architecture
- * There is tight coupling b/w client and DB Machines.

Problems

① What if drivers need to change
↳ client/presentation layer
Code needs to be updated
In each client

② Coupling

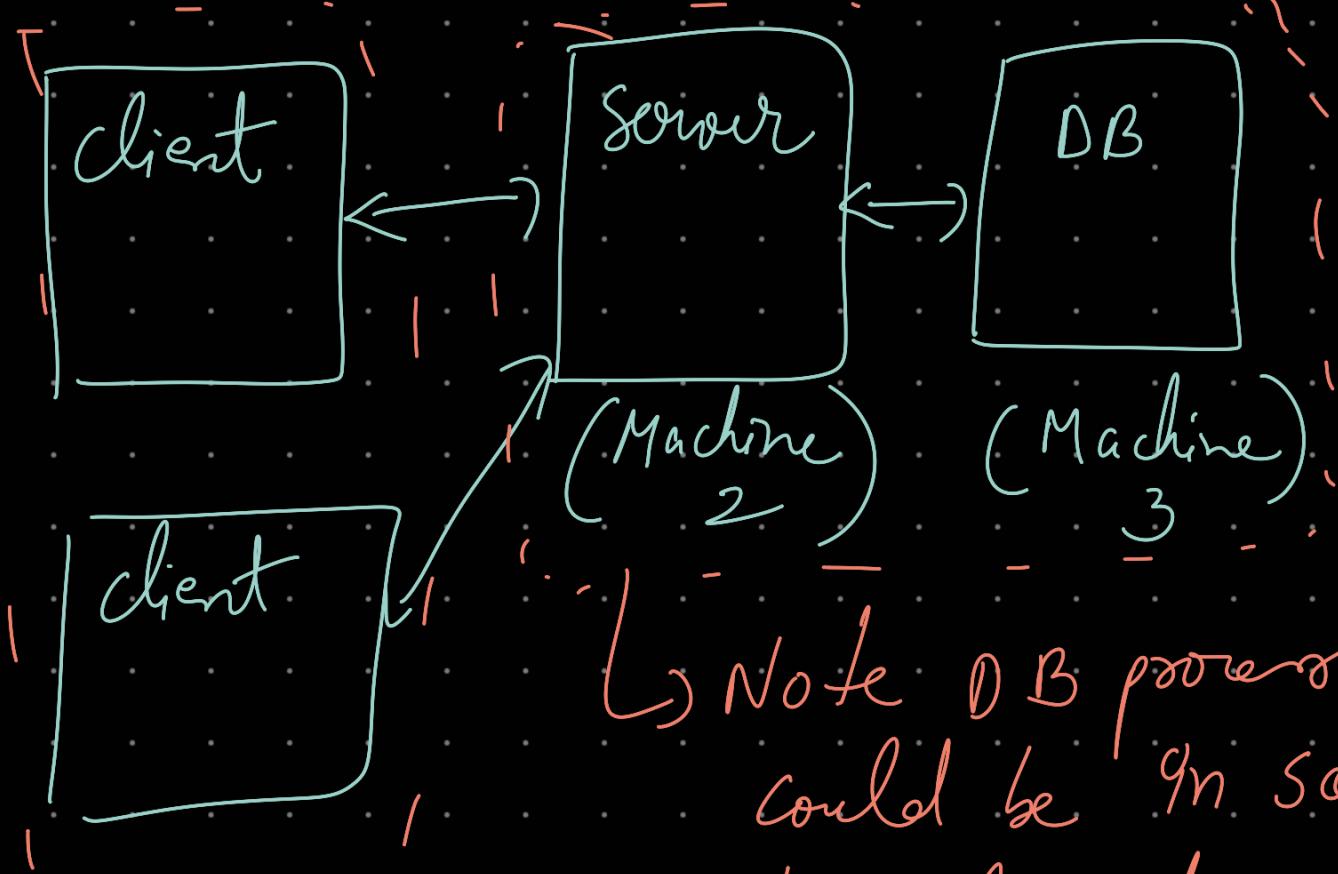
③ Scaling not possible



Solved by 3-tier architecture

* Need a layer b/w client and
Database



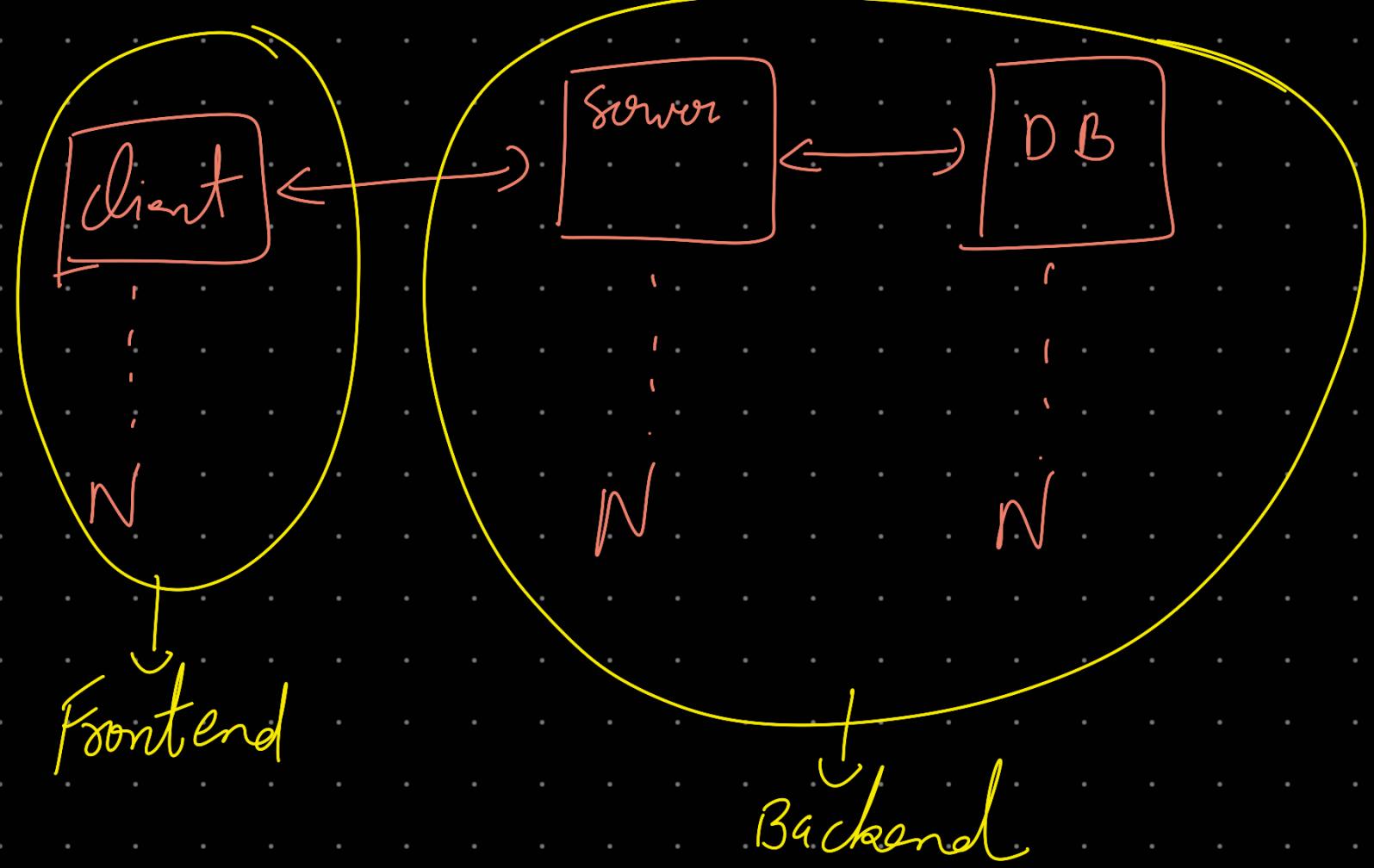


→ Note DB server
could be in some
physical Machine or
that of sever

→ This could be Desktop application,
Browser/Web app, Mobile Application.

Advantages

- ① Separation of concern
- ② Loose Coupling
- ③ Client is presentation layer only



- * What exactly is **server**?
- * It's a software running on Remote physical Machine serving requests from client
 - ↳ Generally use HTTP/HTTPS protocol.

Static Content

E.g. HTML file,
CSS file,
Image, video
etc

Dynamic Content

E.g. Data dynamic
e.g acc. to user
query DB
and give result

- * Desktop PC/Laptop have different kind of Hardware as compared to Server machine
- * Server OS are optimised for server operation as compared to Desktop OS generally contain GUI code as well