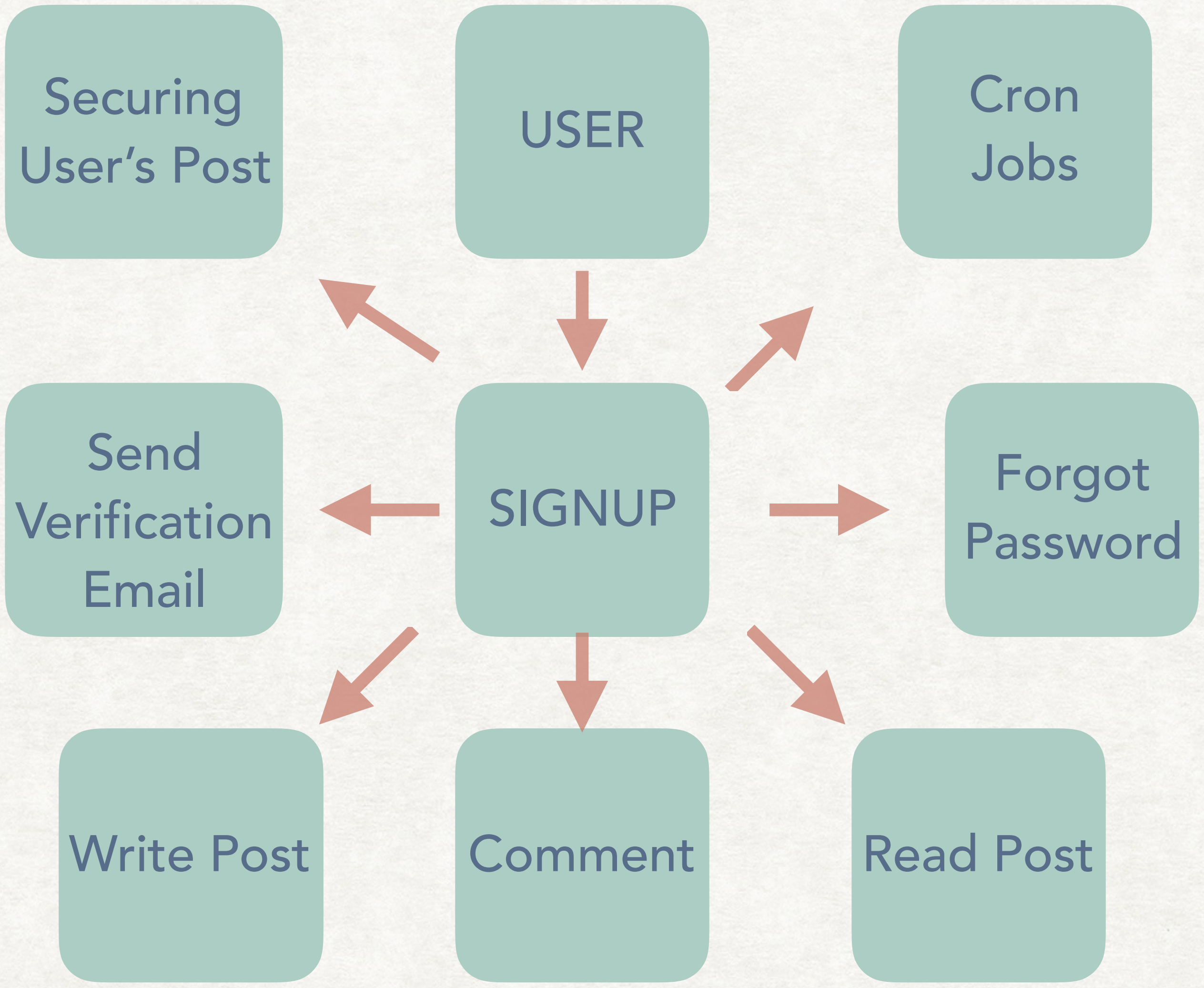
WHAT IS THIS COURSE ABOUT?

- ✓ Creating Rest Api with Node/Express/Mongo.
- ✓ Using Typescript with Node.js & Express.js.
- ✓ Learning About Dozens of Third Party modules/packages.
- ✓ Creating Rest Apis Based on a real Use Case from Scratch.
- ✓ Deploying your Project for production.

WHAT WILL YOU LEARN?

- ✓ How to Build Rest Apis.
- ✓ How to think about structure of api and database.
- ✓ How to Write Clean Code (Good Pattern of Coding).
- ✓ Working with local and remote DataBase.
- ✓ Handling Image Upload , Automating Server (Cron Jobs), Authentication and much more.
- ✓ We will Look Intro Web Scraping.
- ✓Load testing of Server
- ✓Automatic Scaling of Server
- ✓Mongodb indexing, replication

WHAT WE ARE GOING TO BUILD?



Securing User's Post

USER

Cron Jobs

Send Verification Email

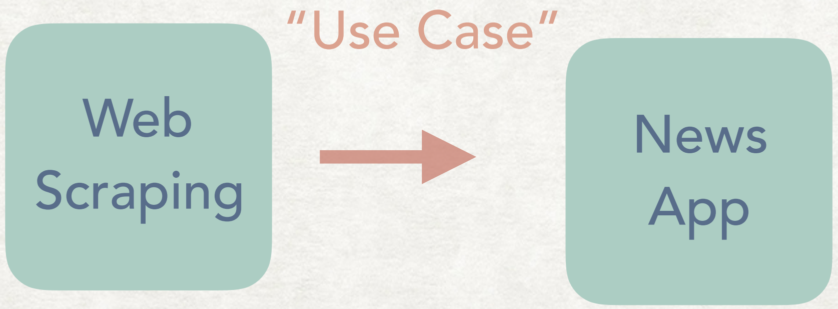
SIGNUP

Forgot Password

Write Post

Comment

Read Post



"Use Case"

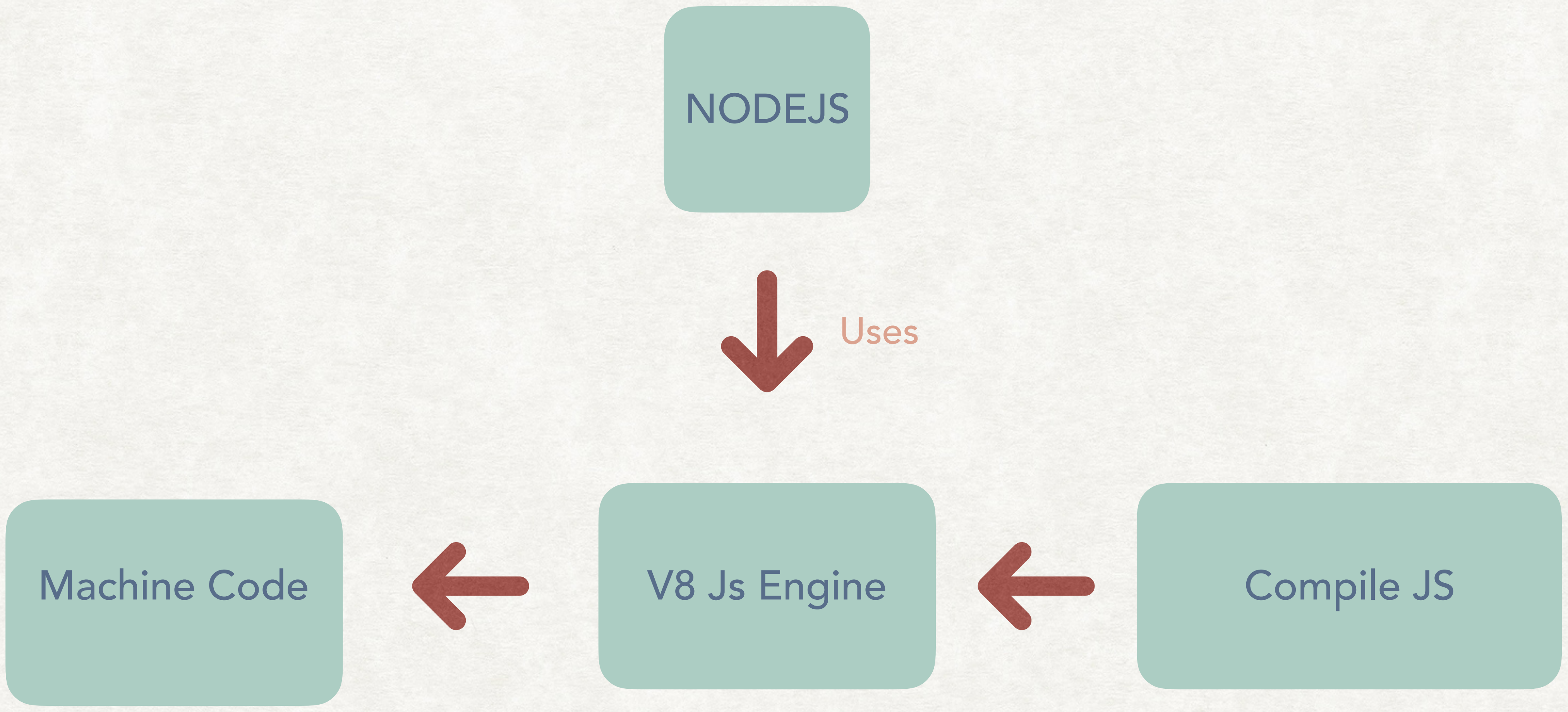
Web Scraping

News App

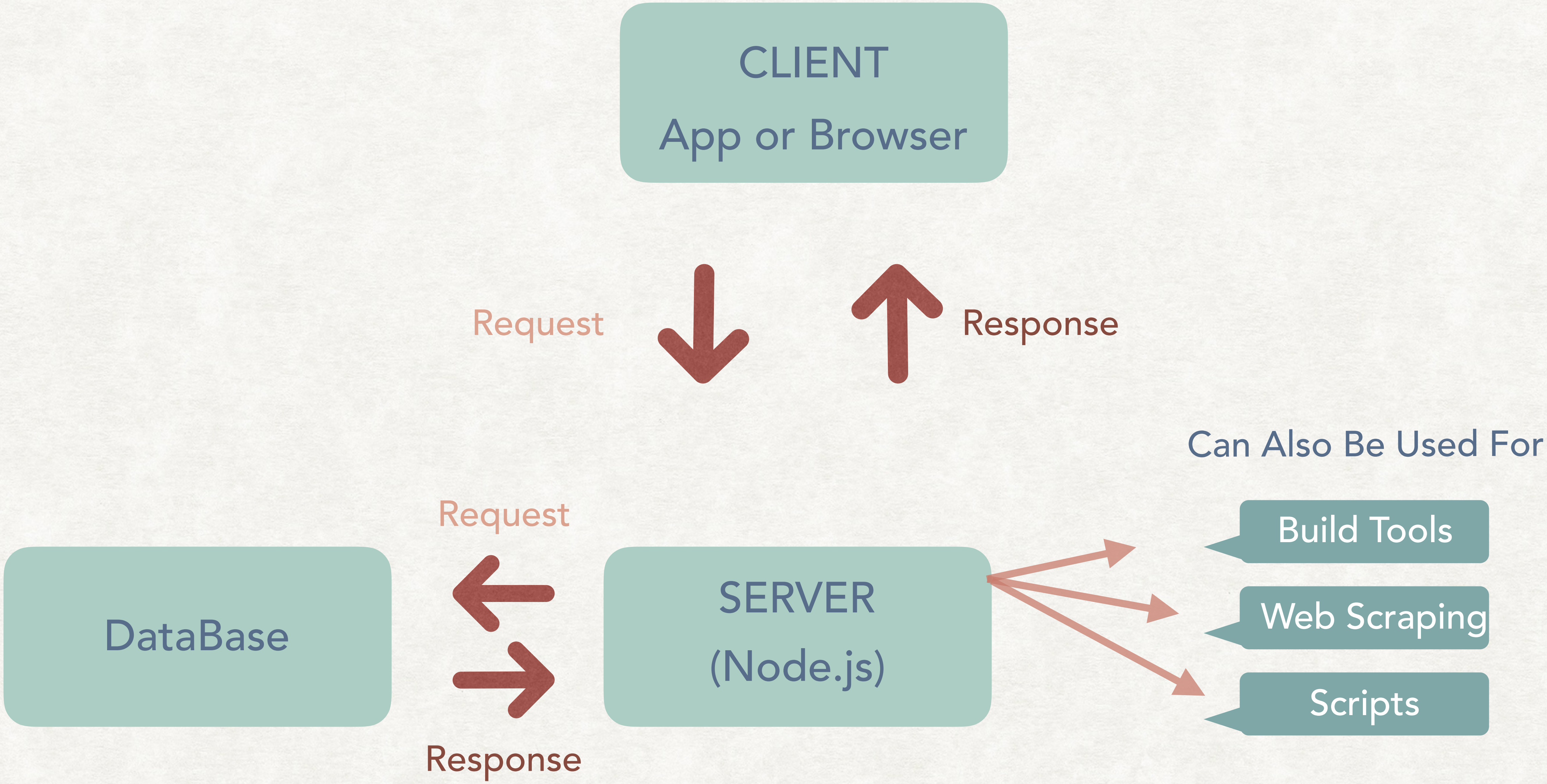
What is Node.js?

A Javascript Runtime

It is Javascript Running On Server

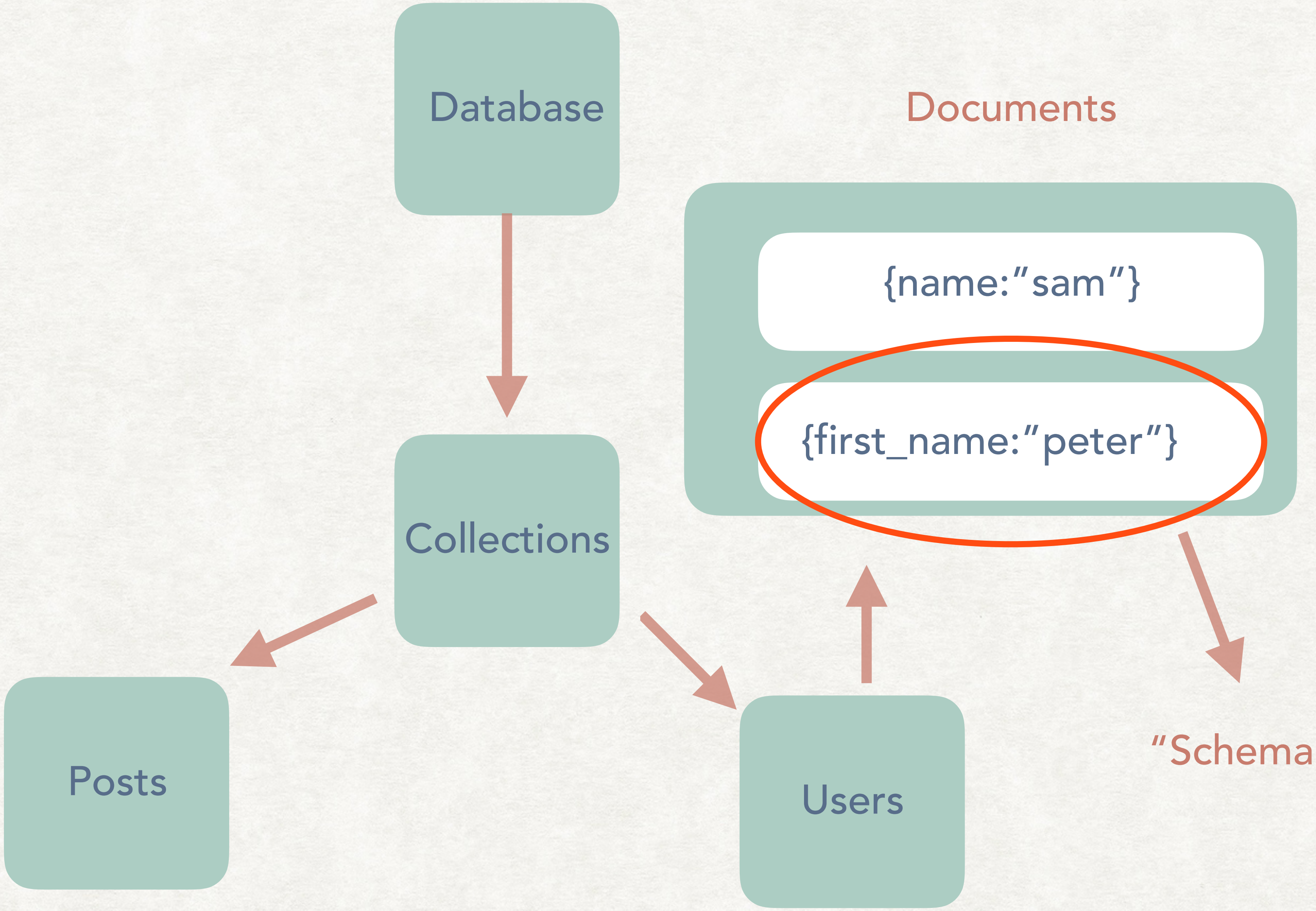


How Node/Express.js Works?



What is MongoDB?

"It Uses JSON (BSON) Format"



Database

Documents

{name:"sam"}

{first_name:"peter"}

Collections

Posts

Users

"Schema Less"

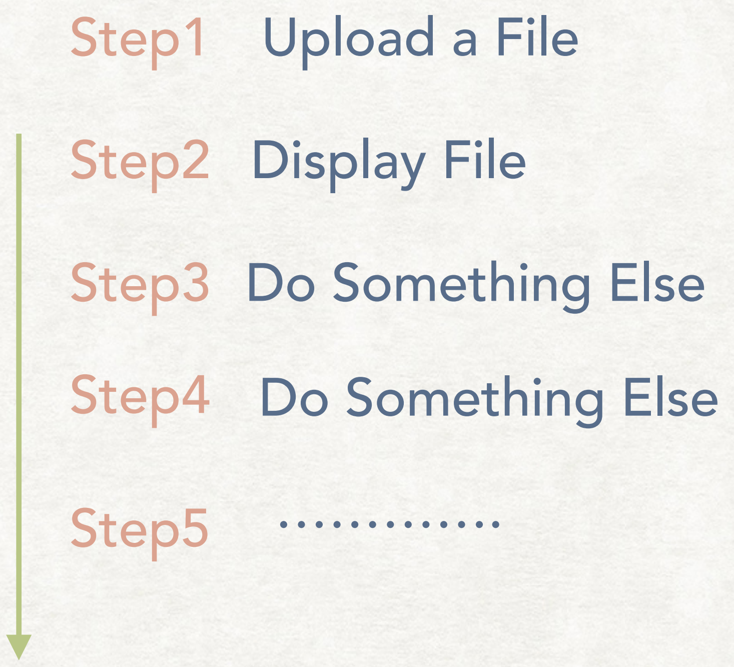
Blocking Vs Non Blocking Code?

Blocking

✓ Upload a File.

✓ Then, Display The file.

✓ Do Something else.



Step1 Upload a File

Step2 Display File

Step3 Do Something Else

Step4 Do Something Else

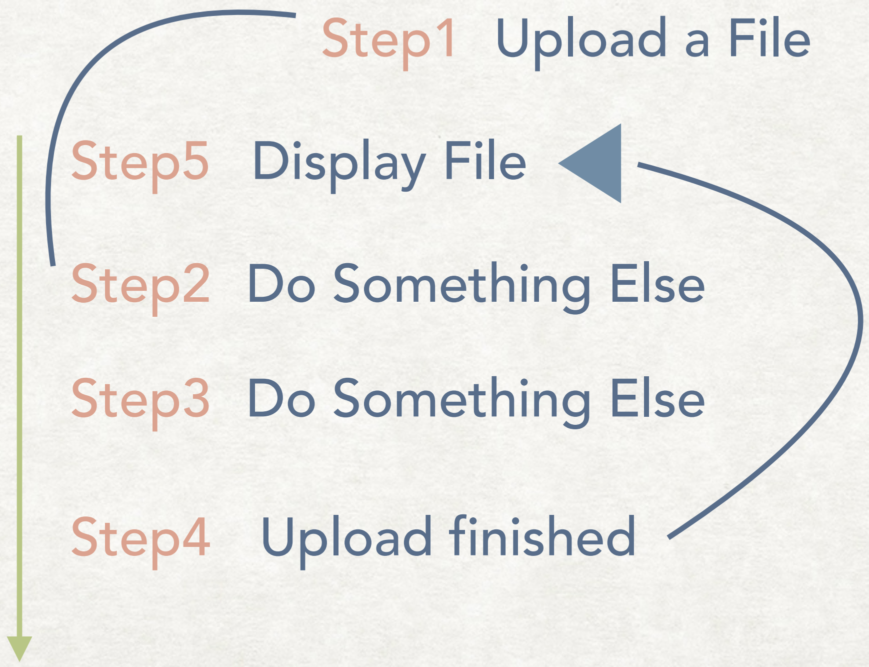
Step5 ....

Non-Blocking

✓ Upload a File.

✓ Do Something else.

✓ Display when Upload completes.



Step1 Upload a File

Step5 Display File

Step2 Do Something Else

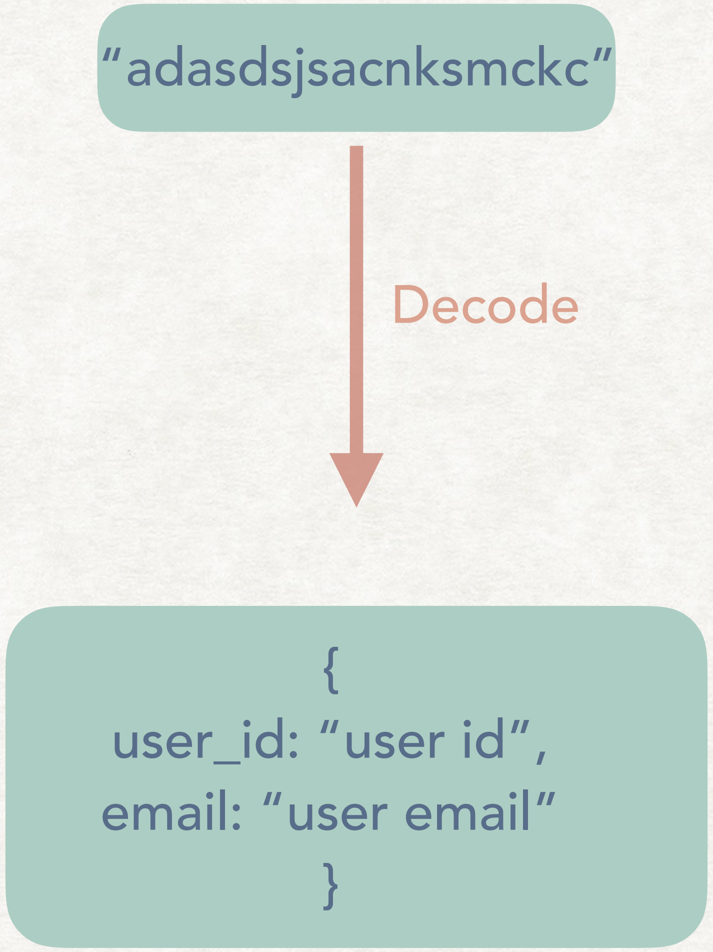
Step3 Do Something Else

Step4 Upload finished

WHAT IS JWT?

“It is JSON Web Token”

“It is a Long String”

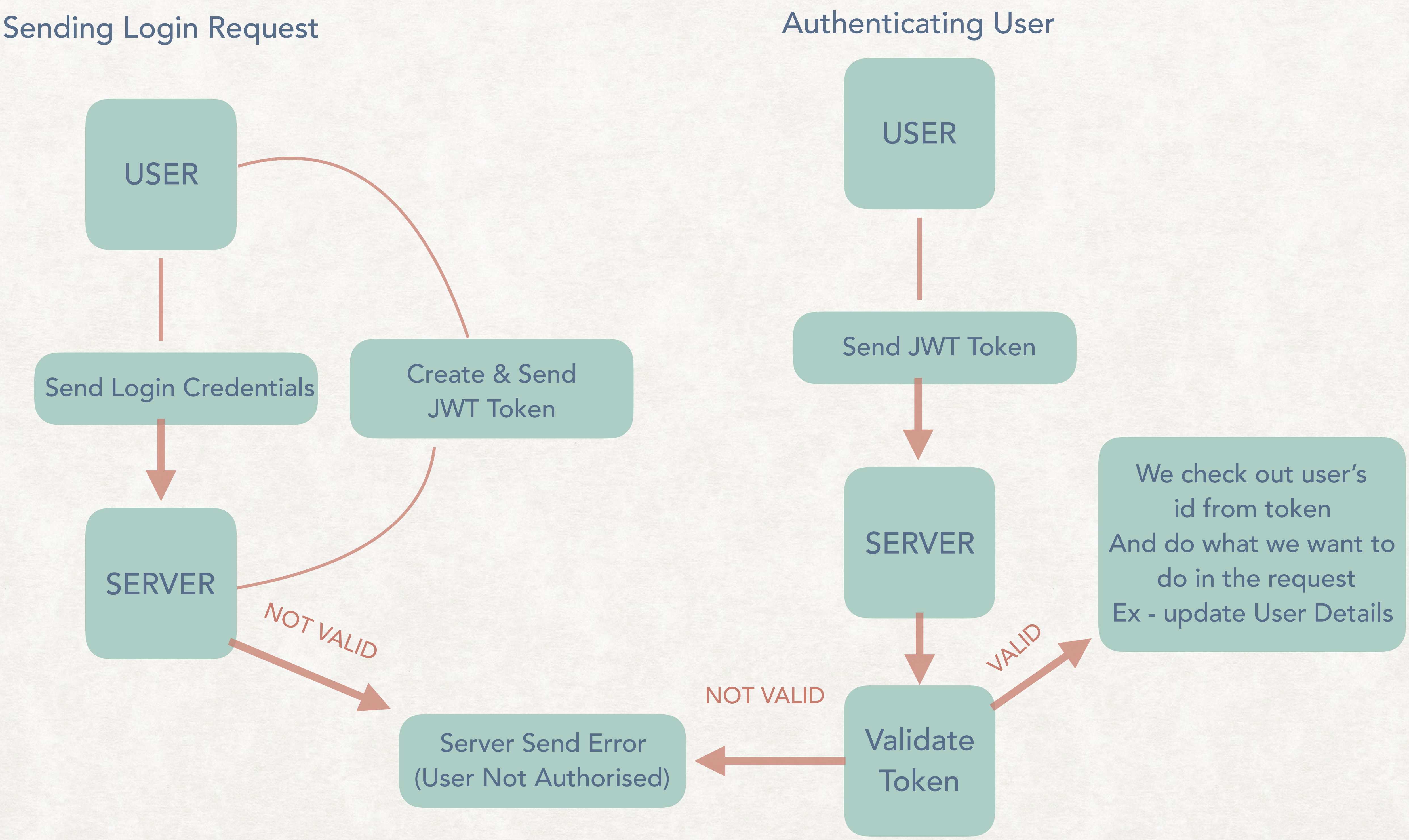


“adasdsjsacnksmckc”

Decode

{
user_id: “user id”,
email: “user email”
}

How Authentication Works?



Sending Login Request

USER

Send Login Credentials

SERVER

Create & Send JWT Token

NOT VALID

Server Send Error (User Not Authorised)

Authenticating User

USER

Send JWT Token

SERVER

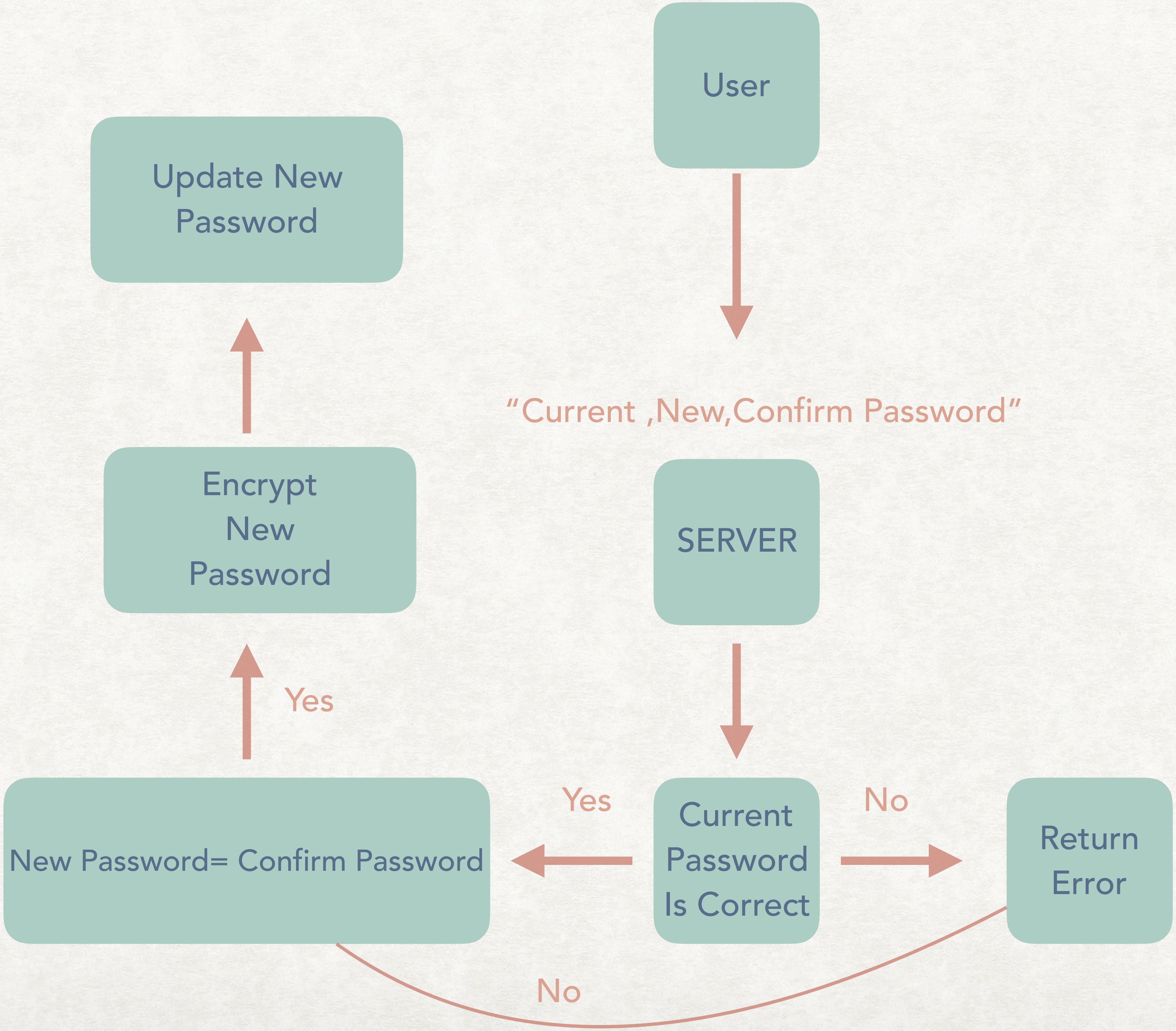
Validate Token

NOT VALID

VALID

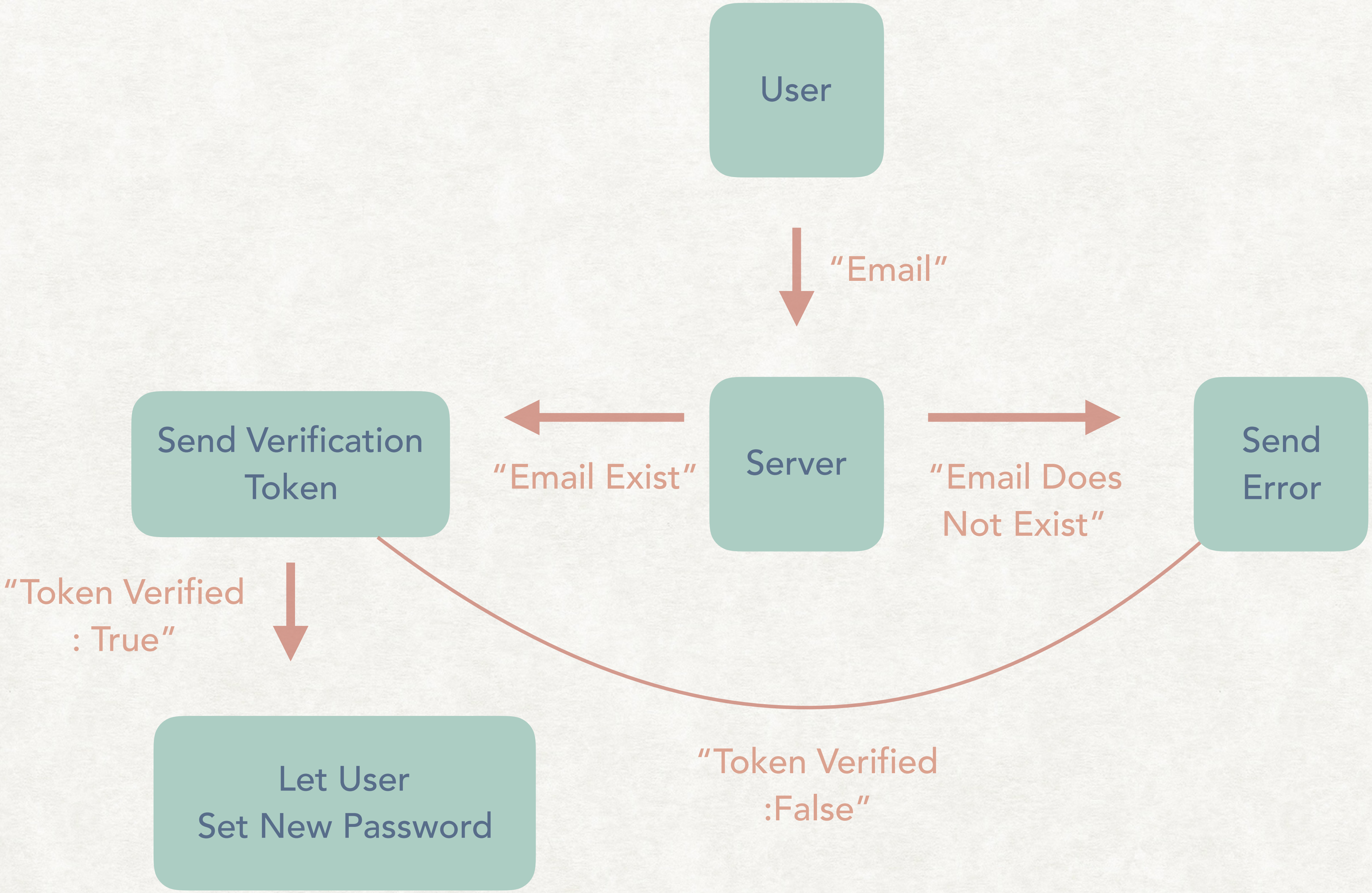
We check out user's id from token
And do what we want to do in the request
Ex - update User Details

Update Password Process



User
"Current ,New,Confirm Password"
SERVER
Current Password Is Correct
No
Return Error
Yes
New Password= Confirm Password
Yes
Encrypt New Password
Update New Password
No

Reset Password Process



User

"Email"

Server

"Email Exist"

Send Verification Token

"Email Does Not Exist"

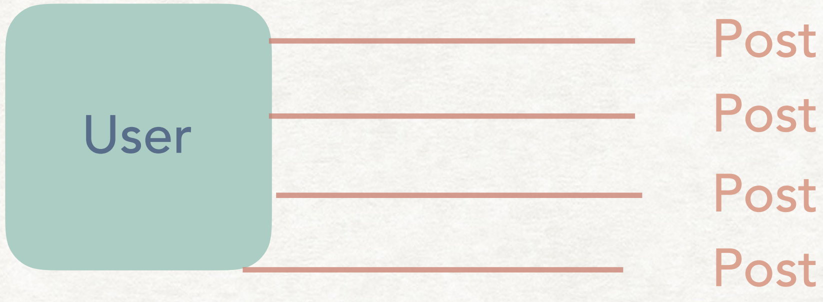
Send Error

"Token Verified : True"

Let User Set New Password

"Token Verified :False"

One To Many Relationship



Way1 (User Schema)

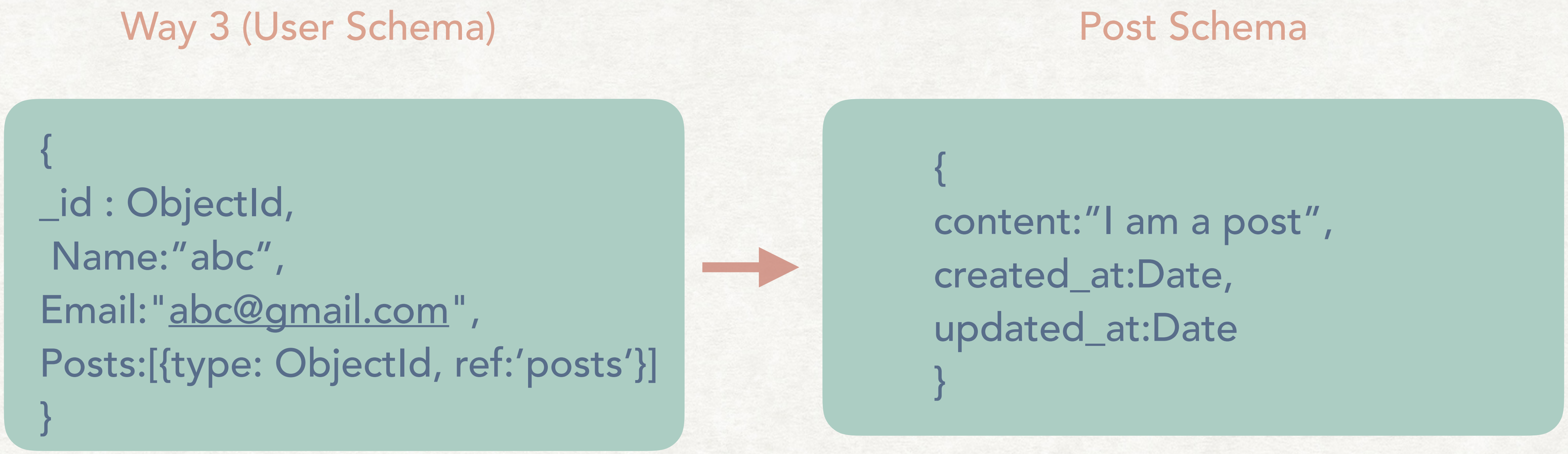
```
{
_id : ObjectId,
 Name:”abc”,
Email:"abc@gmail.com",
Posts:[{content:”I am a post”,created_at:Date,
updated_at:Date}]
}
```

Way2 (User Schema)

```
{
_id : ObjectId,
 Name:”abc”,
Email:"abc@gmail.com",
Posts:[postSchema]
}
```

Post Schema

```
Let postSchema = {
content:”I am a post”,created_at:Date,
updated_at:Date
}
```

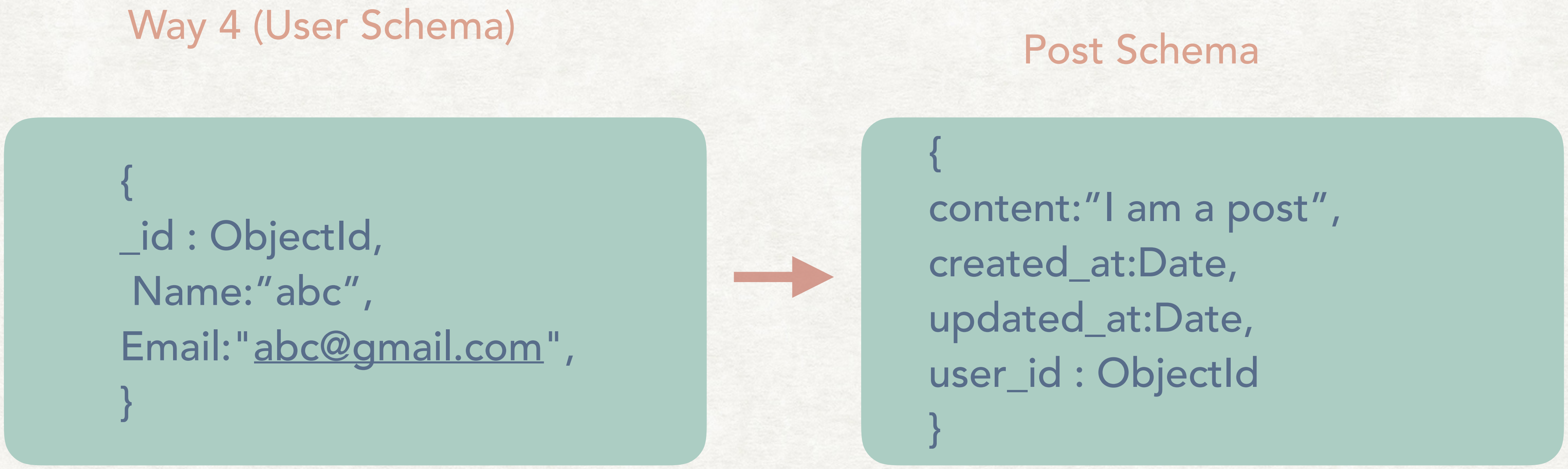


Way 3 (User Schema)

```
{
_id : ObjectId,
 Name:"abc",
Email:"abc@gmail.com",
Posts:[{type: ObjectId, ref:'posts'}]
}
```

Post Schema

```
{
content:"I am a post",
created_at:Date,
updated_at:Date
}
```



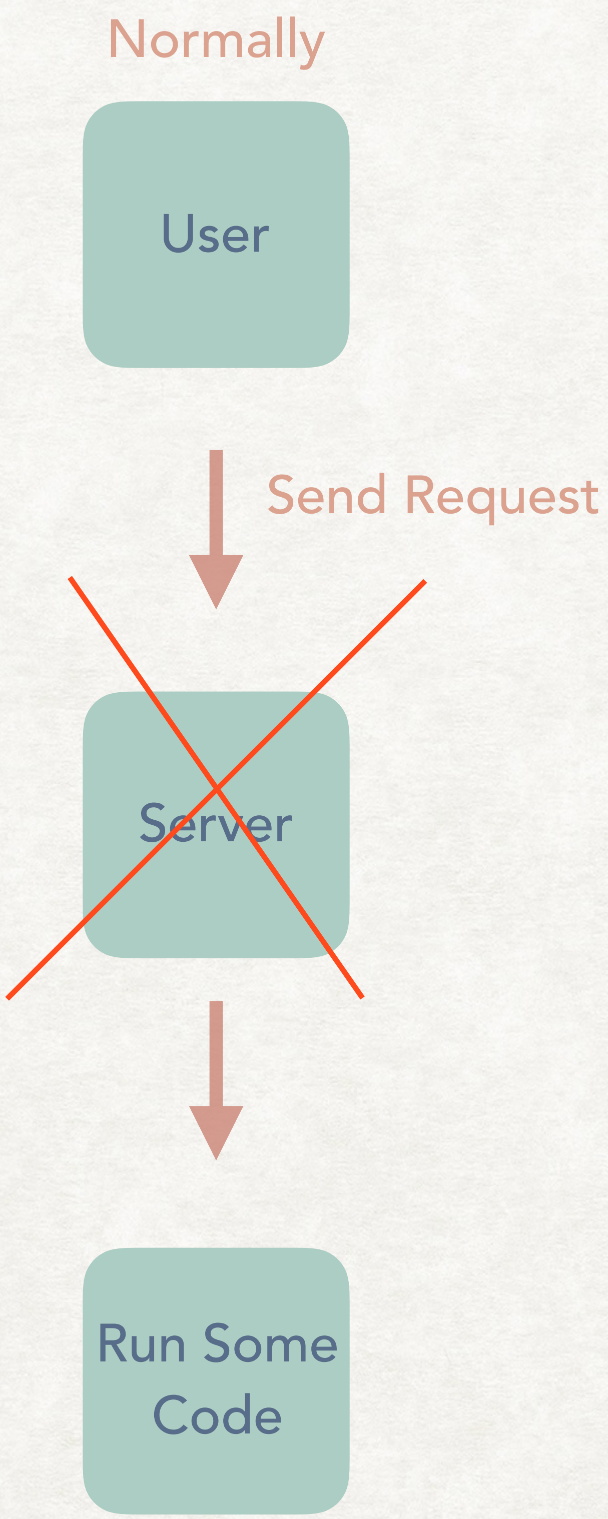
Way 4 (User Schema)

```
{
_id : ObjectId,
 Name:"abc",
Email:"abc@gmail.com",
}
```

Post Schema

```
{
content:"I am a post",
created_at:Date,
updated_at:Date,
user_id : ObjectId
}
```

WHAT IS A CRONJOB?



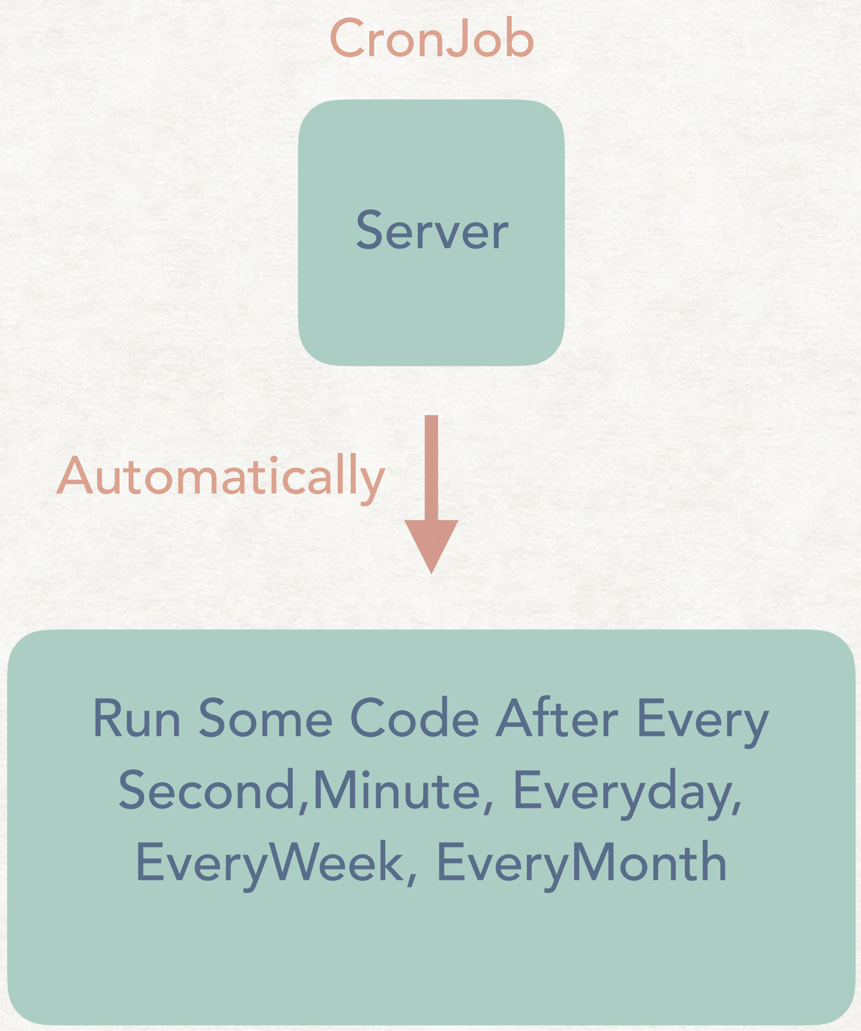
Normally

User

Send Request

~~Server~~

Run Some Code



CronJob

Server

Automatically

Run Some Code After Every Second,Minute, Everyday, EveryWeek, EveryMonth

Use Cases

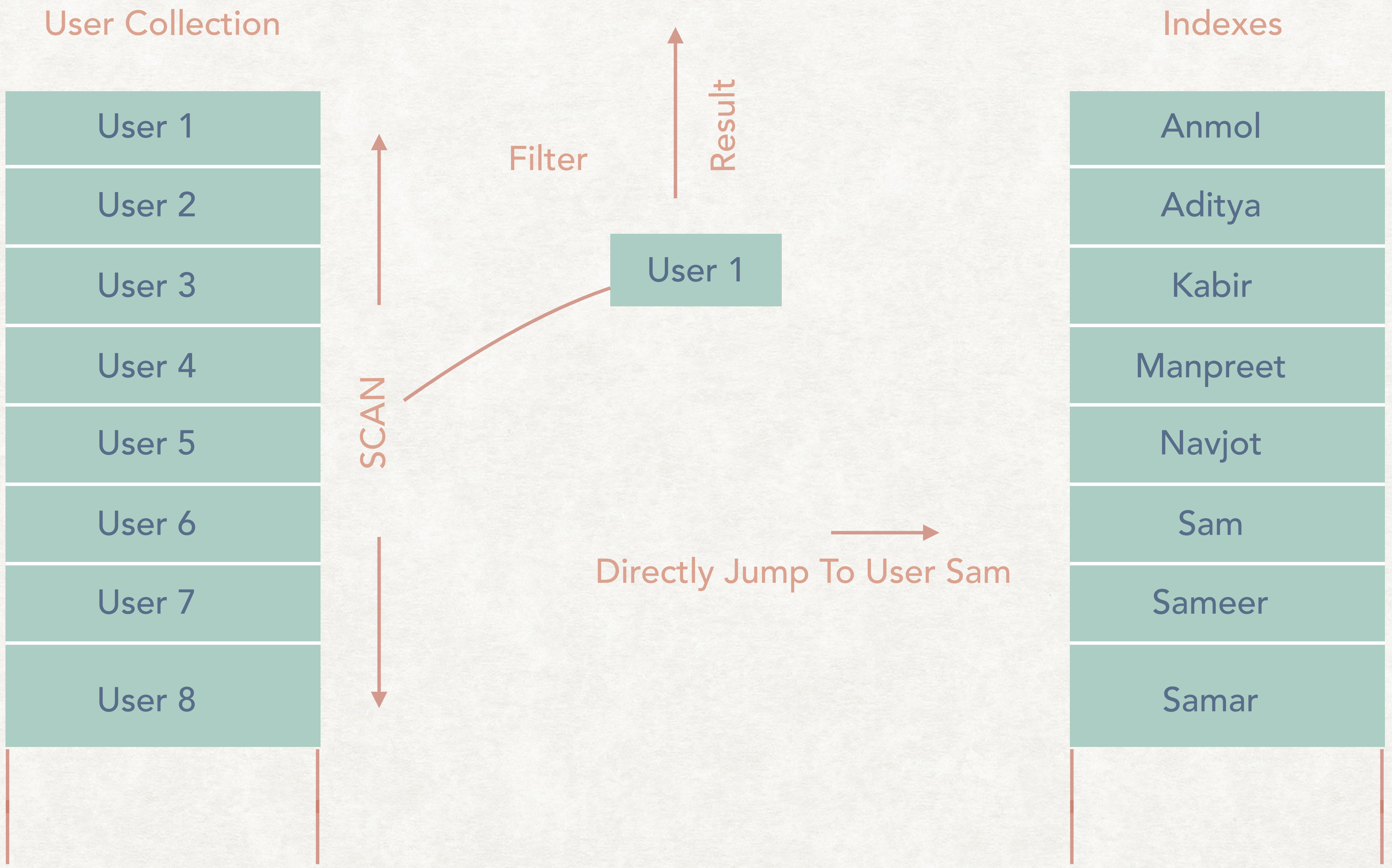
— Sending Emails.

— Updating Database

— Database Backup

WHAT IS MONGODB INDEXING?

```
User.findOne({name:”sam”})
```



User Collection

User 1
User 2
User 3
User 4
User 5
User 6
User 7
User 8

SCAN

Filter

Result

User 1

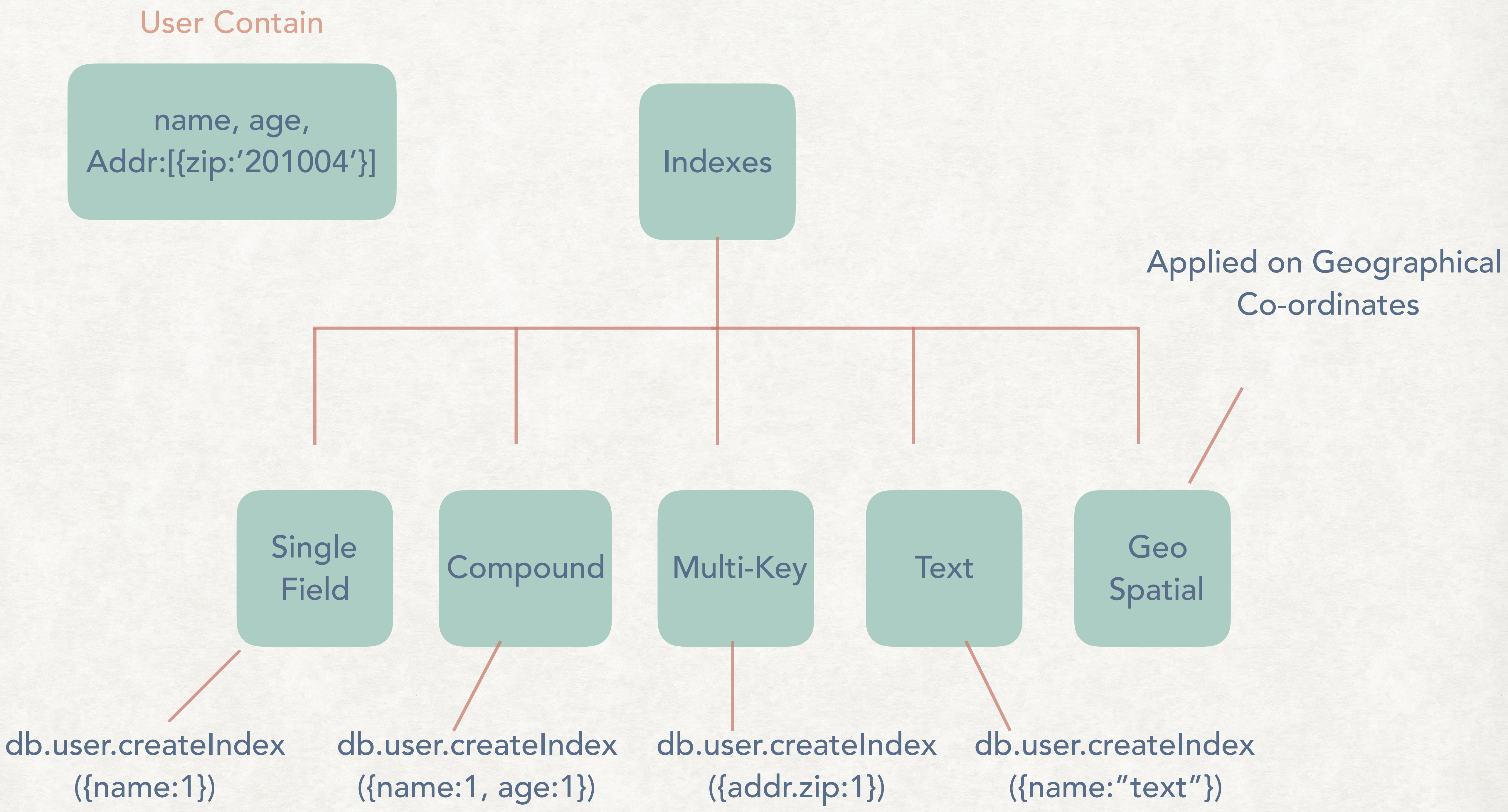
Directly Jump To User Sam

Indexes

Anmol
Aditya
Kabir
Manpreet
Navjot
Sam
Sameer
Samar

* All index are stored in RAM

TYPE OF INDEXES?



User Contain

name, age,
Addr:[{zip:'201004'}]

Indexes

Single Field

Compound

Multi-Key

Text

Geo Spatial

Applied on Geographical Co-ordinates

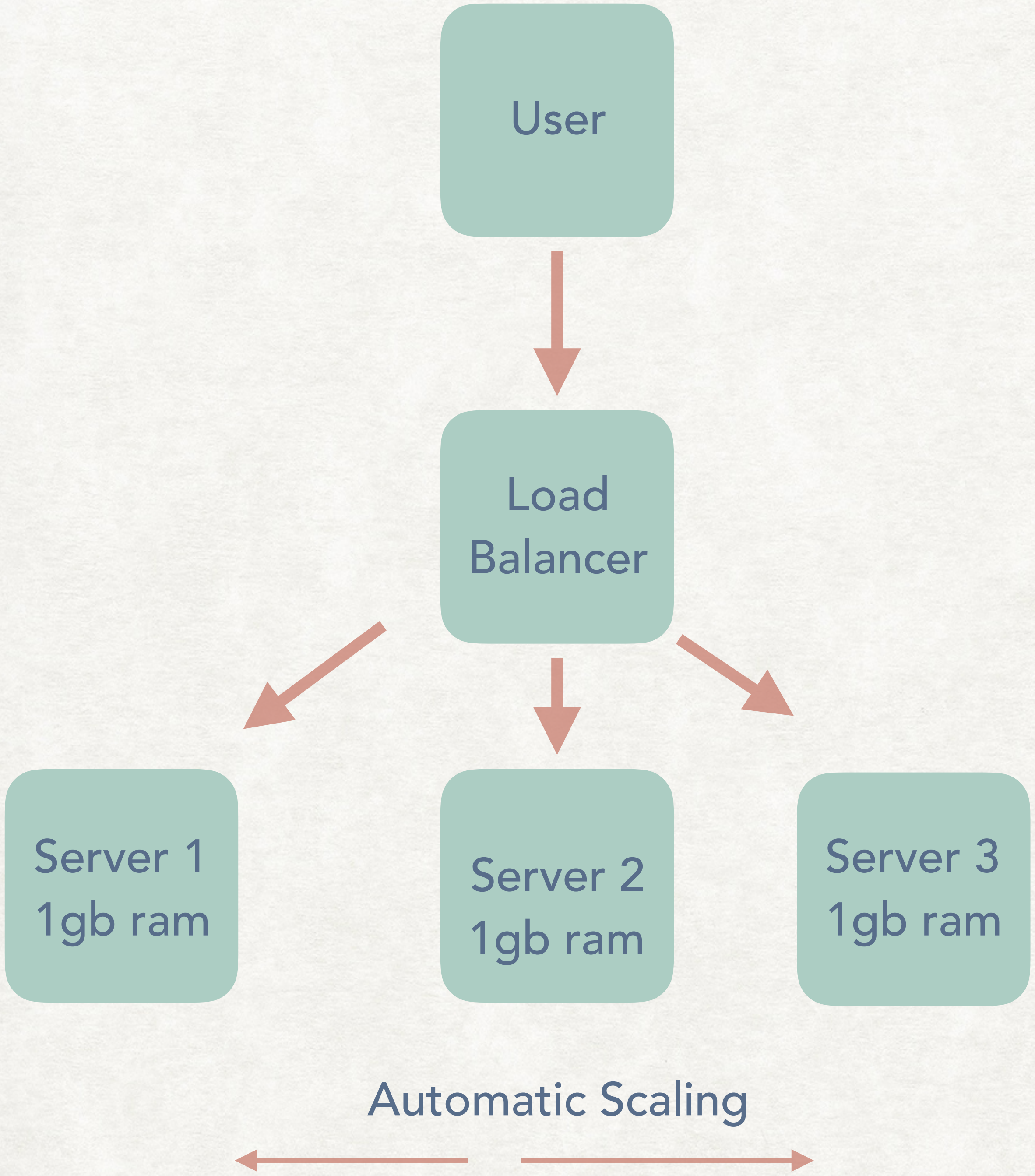
```
db.user.createIndex
({name:1})
```

```
db.user.createIndex
({name:1, age:1})
```

```
db.user.createIndex
({addr.zip:1})
```

```
db.user.createIndex
({name:"text"})
```

HORIZONTAL SCALING



User

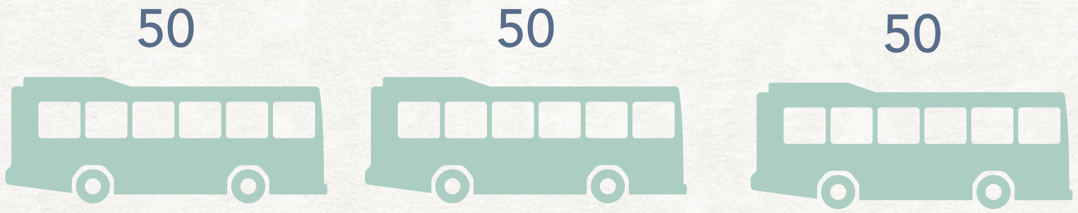
Load Balancer

Server 1 1gb ram

Server 2 1gb ram

Server 3 1gb ram

Automatic Scaling



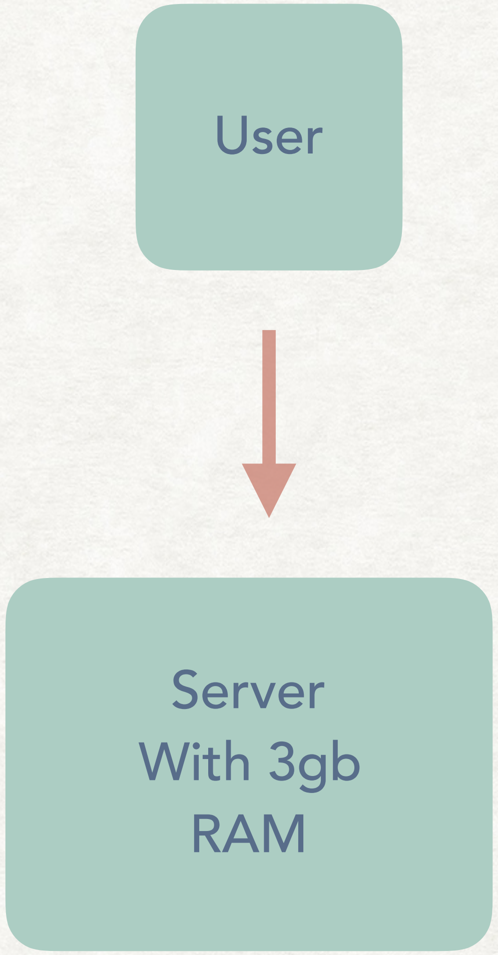
"Cost Effective"

"Automatic Scaling is Possible"

"Set No of Initial Servers Running All the Time"

"Add Additional Servers Based on Traffic"

VERTICAL SCALING



User

Server
With 3gb
RAM

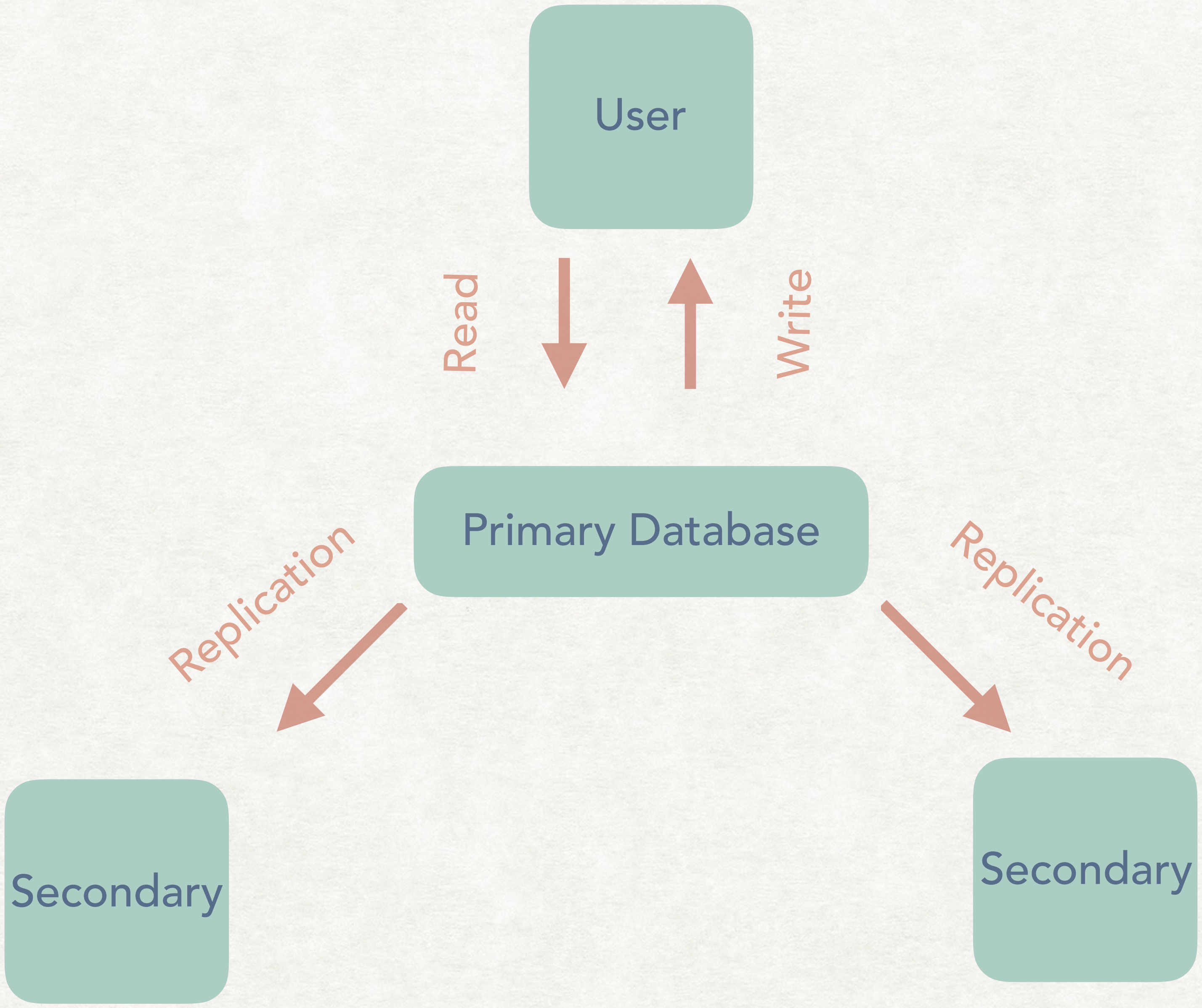
Change Server Configuration when Needed

More Costly

Automatic Scaling is not Possible

Chances of server Failure

MONGODB REPLICATION



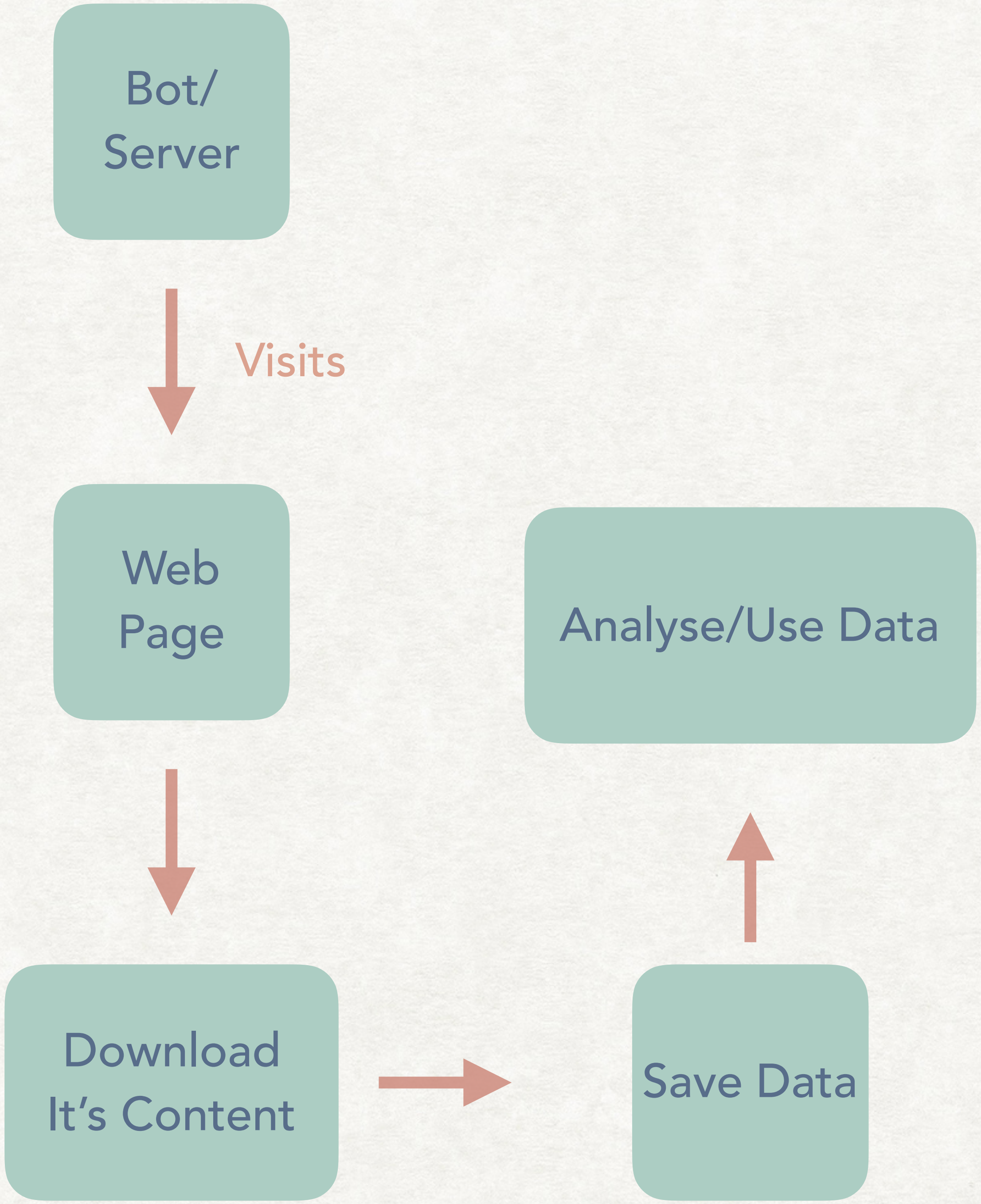
WHAT IS WEB SCRAPING?

Problems

— Websites changes Frequently.

— Ip Address May get blocked by the Owner

— Low on Speed.


Bot/ Server
Visits
Web Page
Download It's Content
Save Data
Analyse/Use Data