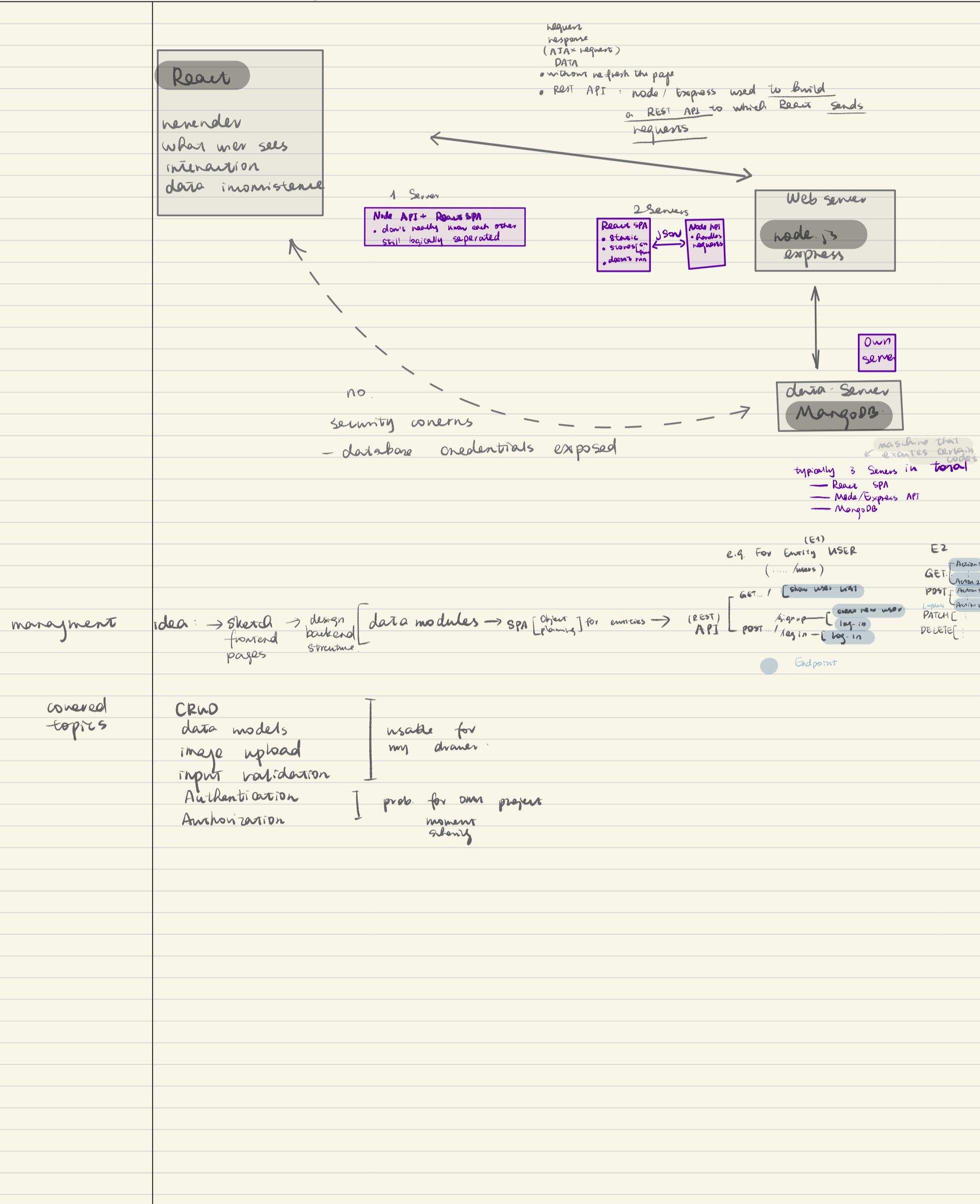


Overview



node + React

Structure

separate
(as if they
were two projects
because they are
logically separated)

frontend

since it shares everything in common and has extra stuff than server
take frontend as example to analyse structure

`package.json`: all dependencies
with in node (`npm install`) → `npm install` → `node`

`package-lock.json`: a sign that you installed the dependencies needed

`index.js`: entry point for frontend app

src

`App.js`:

JSX (HTML style, etc)
import Components from other files

- front end logic
- hook state-management

useEffect (fetch something
from server)

fetch (...5000/product)

→ is

done, what's

responded from

cd path
`npm start`
let code execute
then it would
run on `localhost:3000`

understand

on your

domain

Same as

domain.com

backend

do the same thing here

`npm start` → port 5000

`Server.js`
(backend logic)

responde (REST API: the combination thing, remember?)
`app.get('/product', (req, res) => {`

`res.status(200).json({product: haha})`

) → OK when in

a json that's why when

you open 5000,

you see this

`productS = []`

temporary solution for dummy

recall: Players Array

it's now only on node

however when you refresh the page

all data get lost (explains why you can't

enter a new game)

so later on: MongoDB

`app.listen(5000)`