

Server.js

listen to Refs
& Dispatch

Response
↓

Service

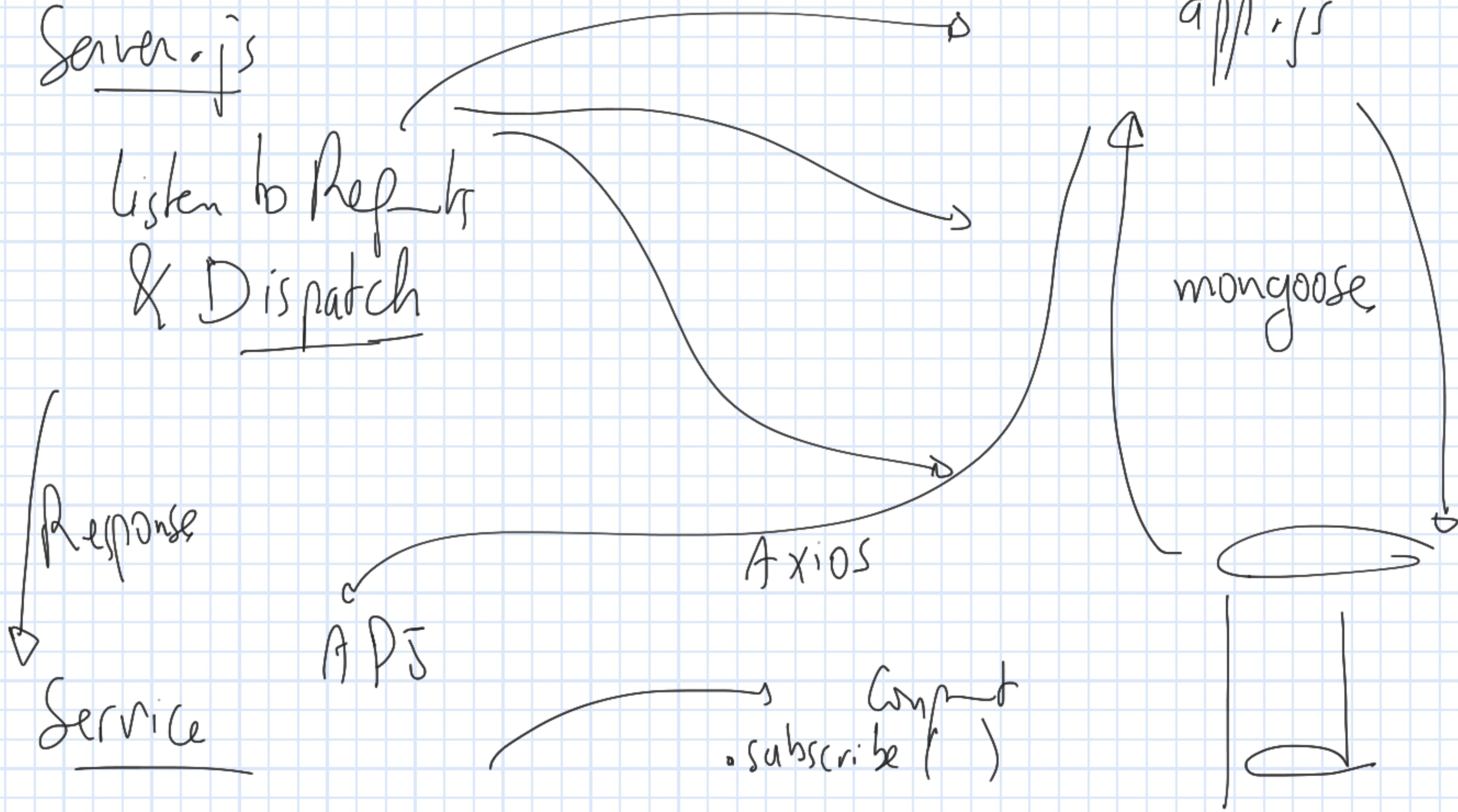
API

Axios

Component
• subscribe()

app.js

mongoose



- 7 methods mongoose

- connect

- model

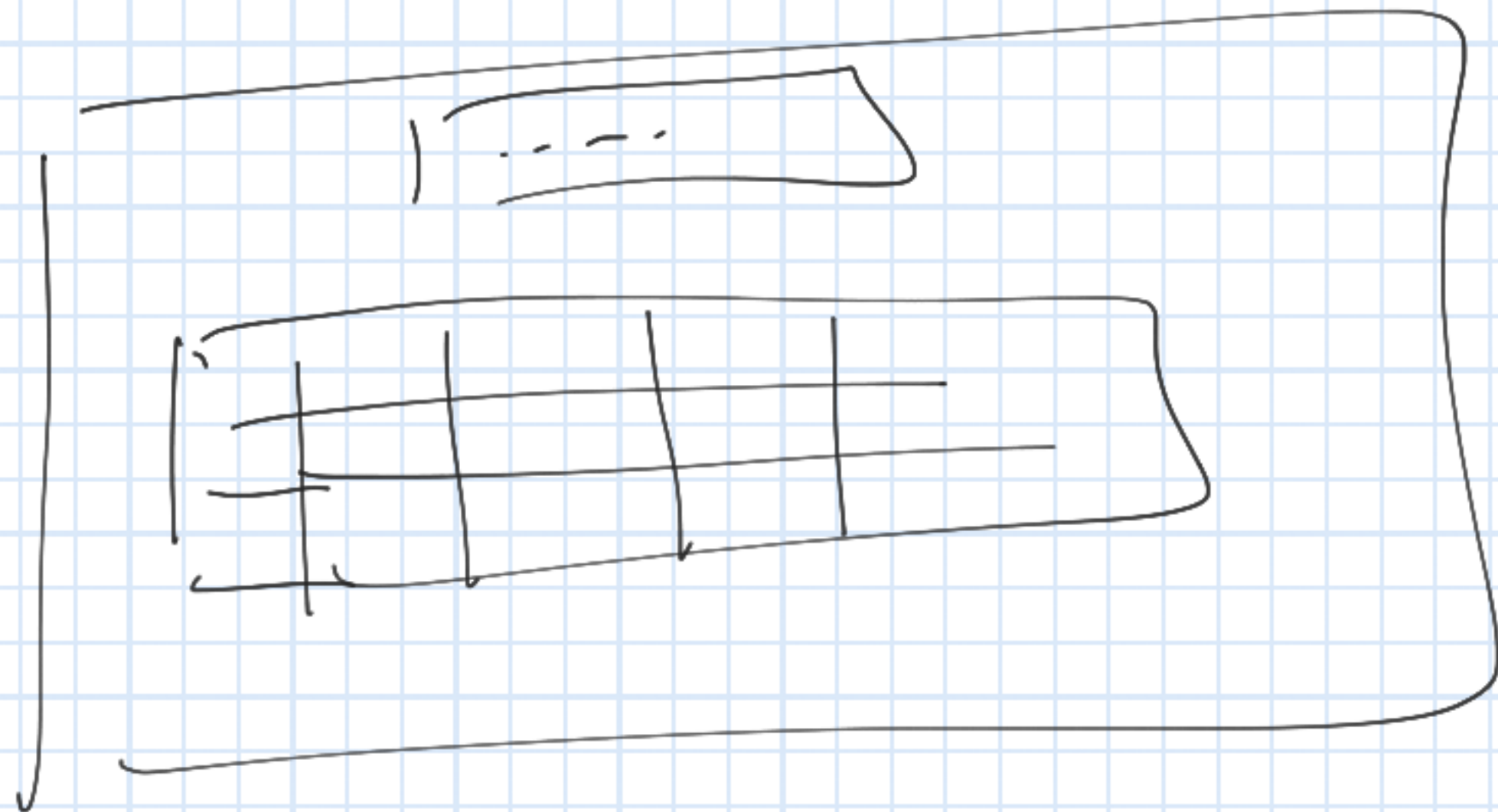
- Schema

- find

- update One

- delete Many

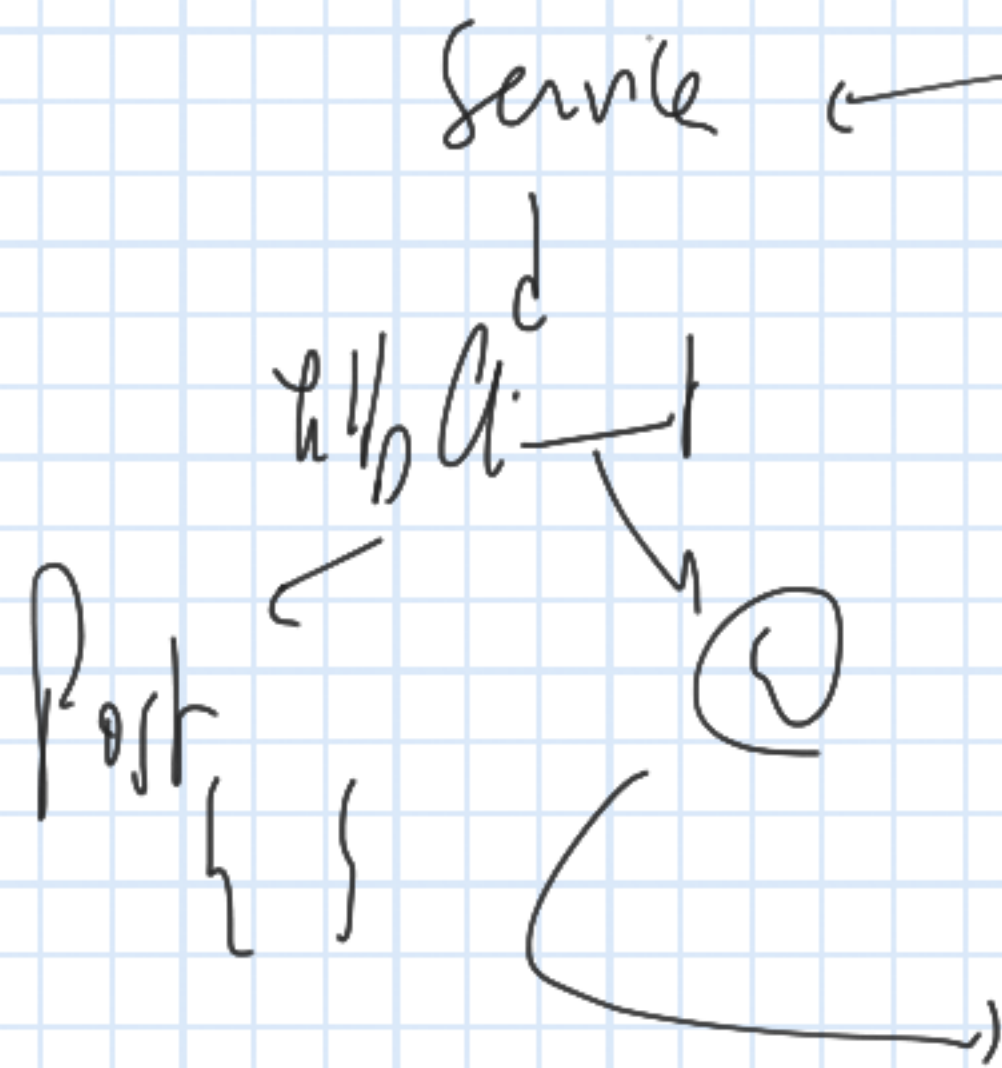
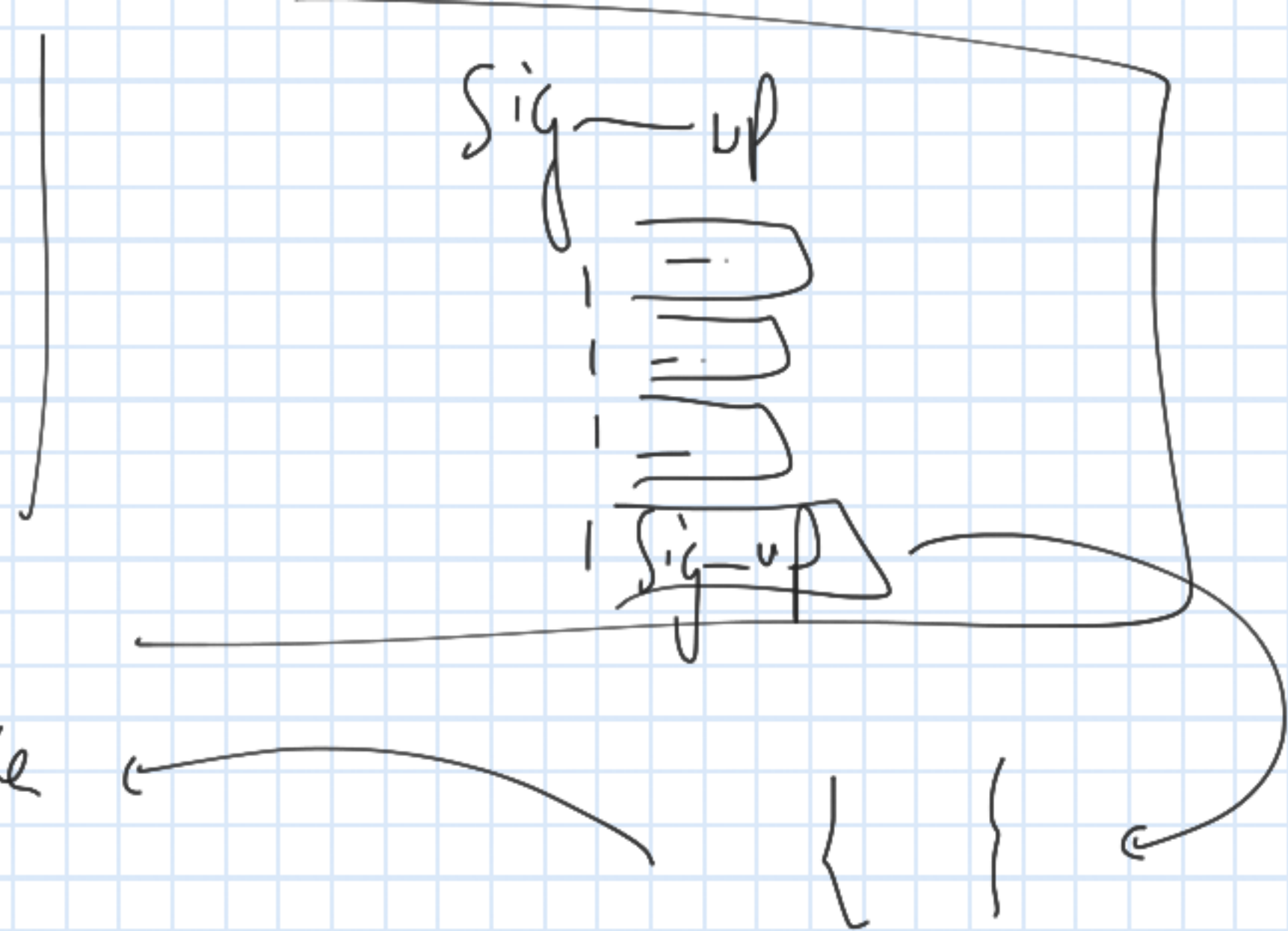
- save



[1,1,1] ← T.filter ((obj) => { Condition })

One
element ← T.find ()

⑦ →



app's B.L
copy paste Pwd
✓ verify Unique E-l

①

E - P
Pwd

login

~~get('R: ...')~~

1. Check E - P

Server

api.js

(2) Compare Pwd

(3) OK

fw -
lw -
id,

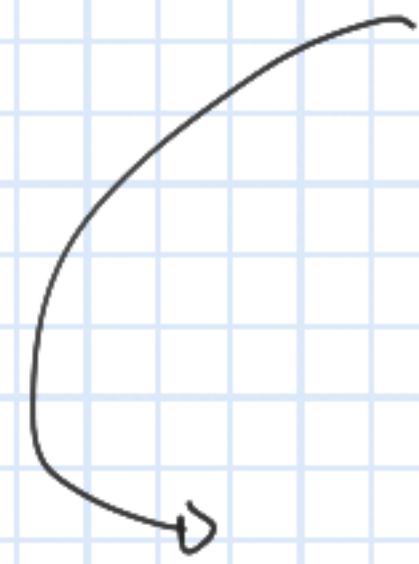
token: (given par
son who when)

Post

help
@

token : {

Payload
data



Service

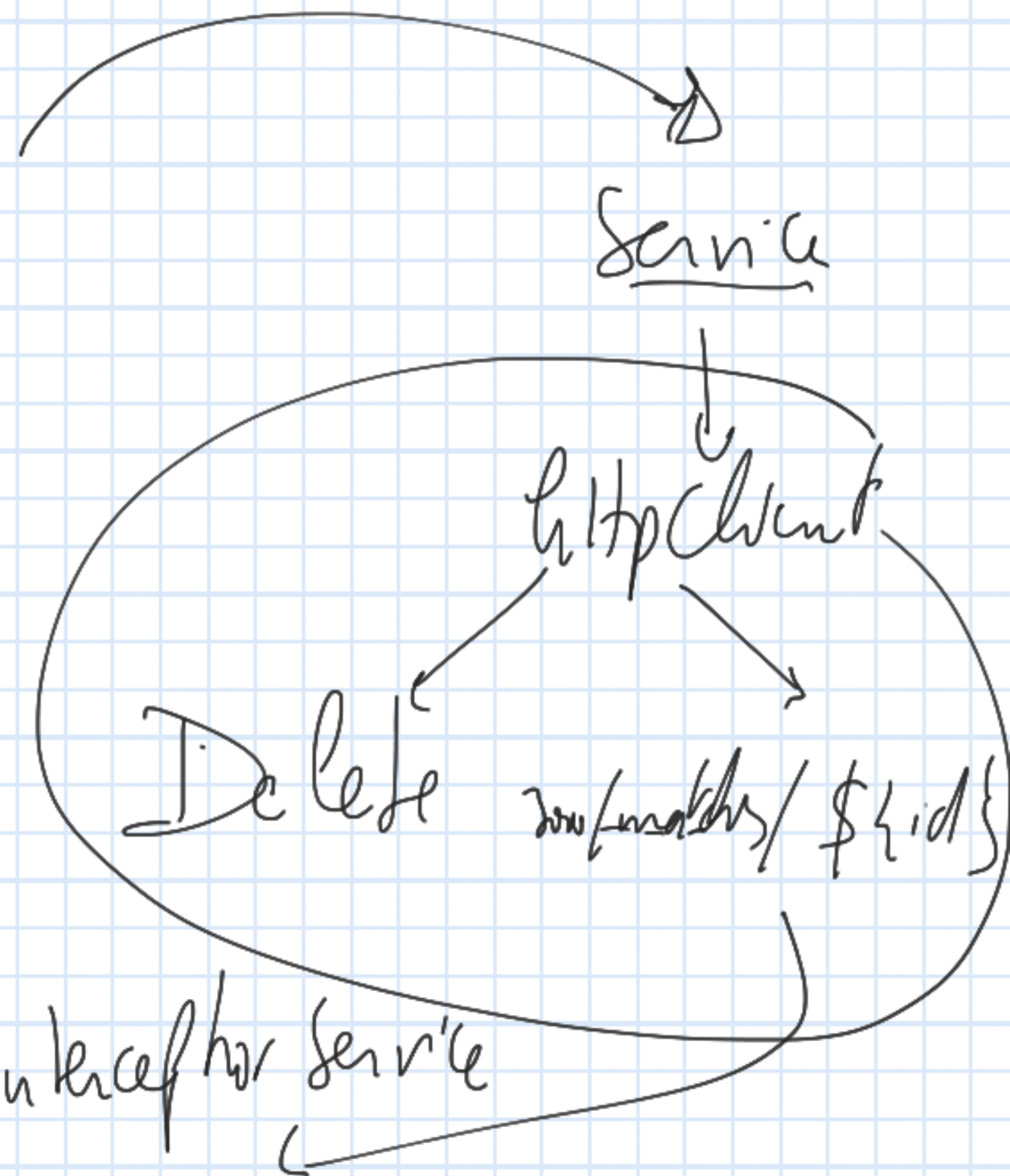
token → LS
fn → LS
LN → LS

Delete Team by Id

Headers

Ref	Authorization: Bearer <u>token</u>
-----	---------------------------------------

token
→ LS



app.js

app.delete('/matches/:id', authenticate
(req, res) => {
 ↓
 req.params.id

authenticate.js
req.header('authorization')

Verify

});

Sweet Alert

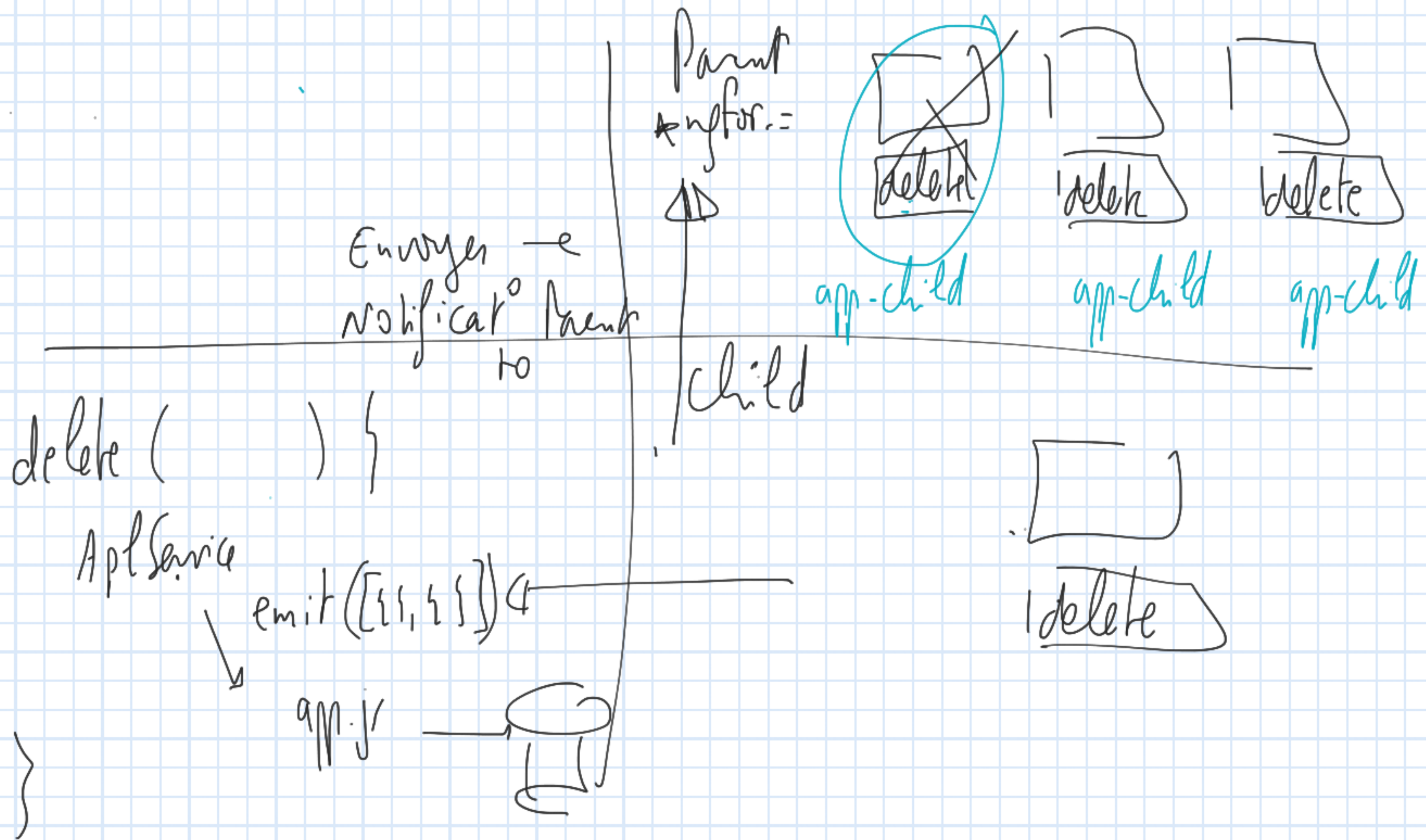
oks

G = r

selector
↳ ~~html~~
↳ ~~html~~

template :

Code
HTML



Validators

TDF:

React

name

|

stadium

|

|

...

↓
Required

|

|

name = "Model"

→ Affecter un id à l'input

required: Valider HTML

<BO

[ngClass] = ""

CSS

{
 'classN1' : w1, t1,
 'classN2' : Co2,
 :
}

TS

Content

</BF>

<BO [ngStyle] =

{
 "prop" : value
}