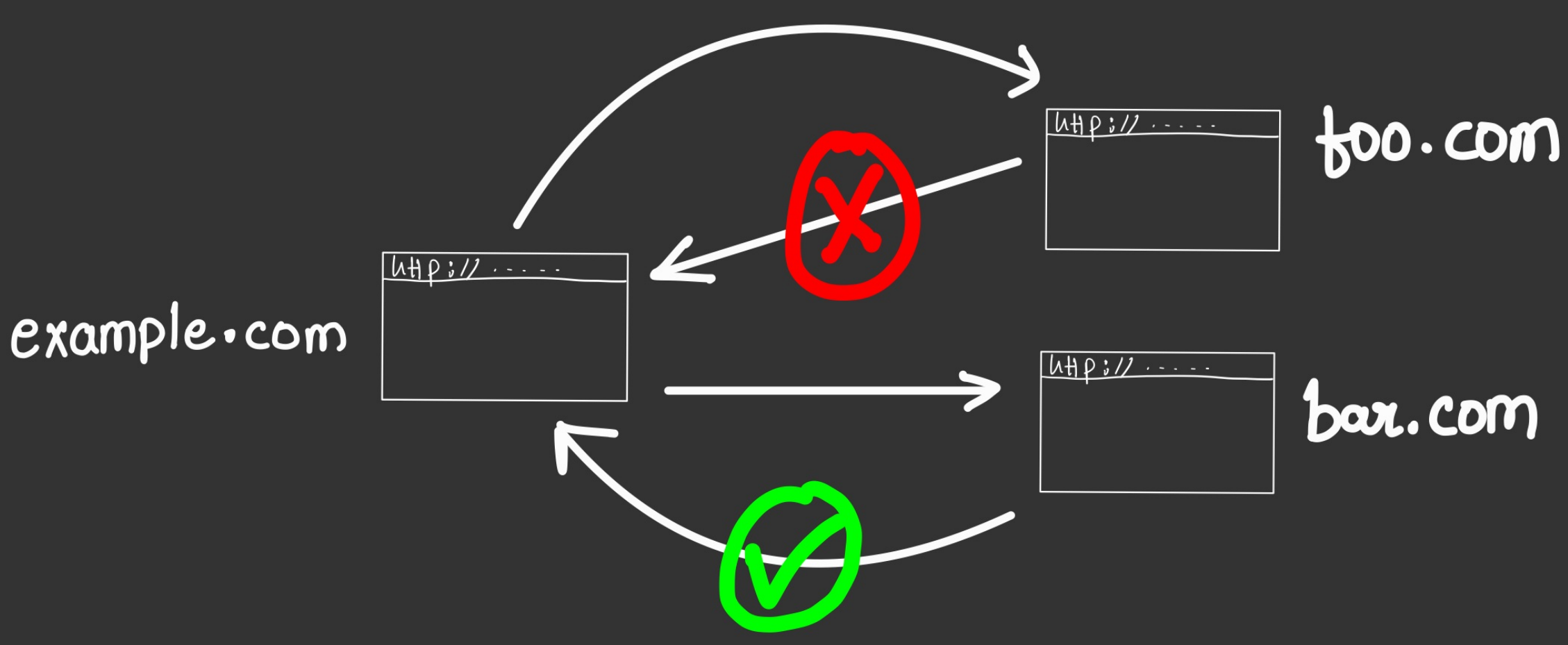
SOP

Same Origin Policy

Brower's security model



sec_r0

What is this ?

Hey !!

I am Rohit

Your Mentor

sec_r0

And today I want to tell you about

SOP

< Same Origin Policy >

* what is SOP ?
* why SOP ?
* By passing SOP
* CORS * Domain Lowering

What is SOP ?

The same-origin policy is a critical security mechanism that restricts how a document or script loaded from one origin can interact with a resource from another origin. It helps isolate potentially malicious documents, reducing possible attack vectors.

This is what mozzila doc says.

I don't understand it either....

Do you know Origins ?

Somewhat

Then let's talk about it first

sec_r0

ORIGINS

You have seen URI right ?

* Uniform resource identifier

like

http://example.com:80/a

- http → protocol
- example.com → domain
- 80 → Port
- a → Path

protocol + domain + Port → Origin

Check if the origins below matches the origin above :-

http://example.com/a

* Default port for http is 80

Yes

https://example.com/a

* protocol mismatch

No

http://example.com:8080/a

* port mismatch.

No

http://sub.example.com/

* Sub domain present

No

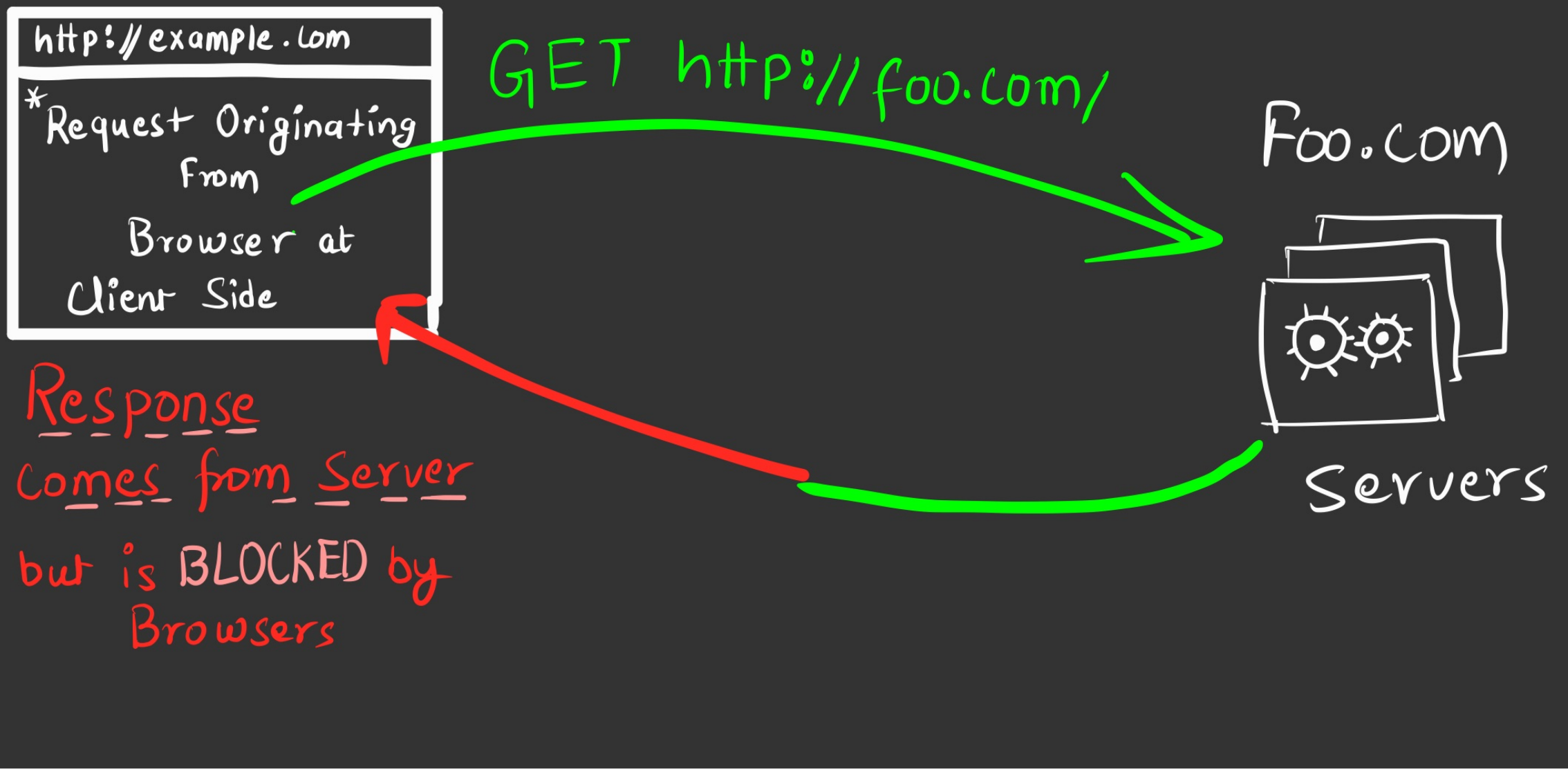
sec_r0

Back to SOP !

Same Origin Policy Instruct browser to not allow access to resources across different origins easily.

Resources can be
- HTML Page
- API response
- JSON Data, etc

Since this Control is at brower level, so response is not shared across domains



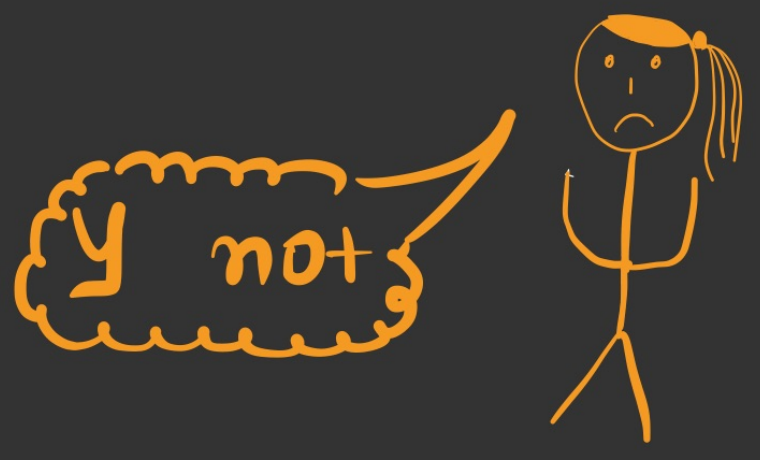
Need for SOP

* To understand the need for SOP, you need to know a couple of things first.

① Any Javascript code while making any Request to foo.com CAN → Script may or may not send credentials
Send Cookies of foo.com also.
→ and browser has to follow it
That's browser's nature

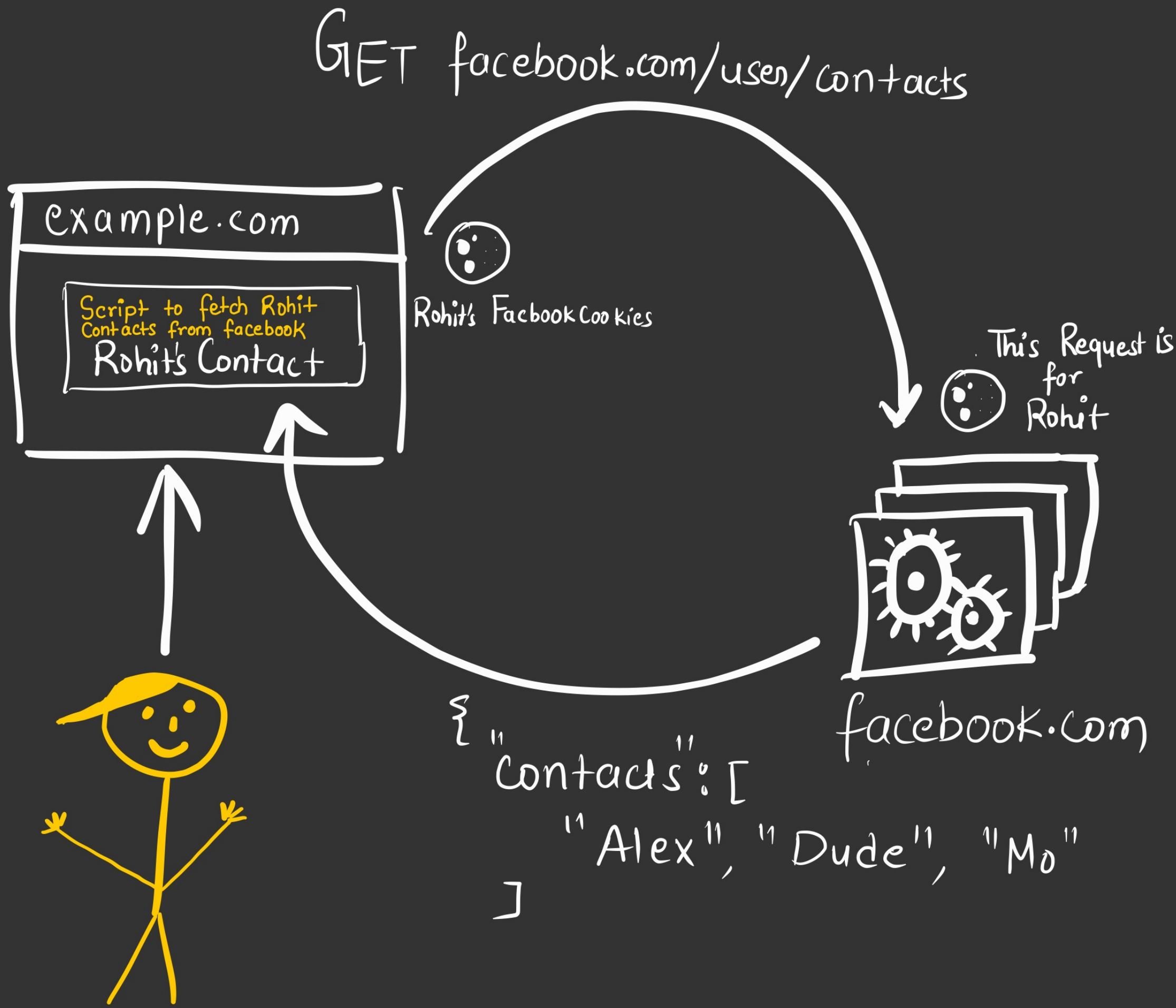
② Cookies Identifies Session directly or Indirectly

③ How the world be without same origin Policy ?



sec_r0

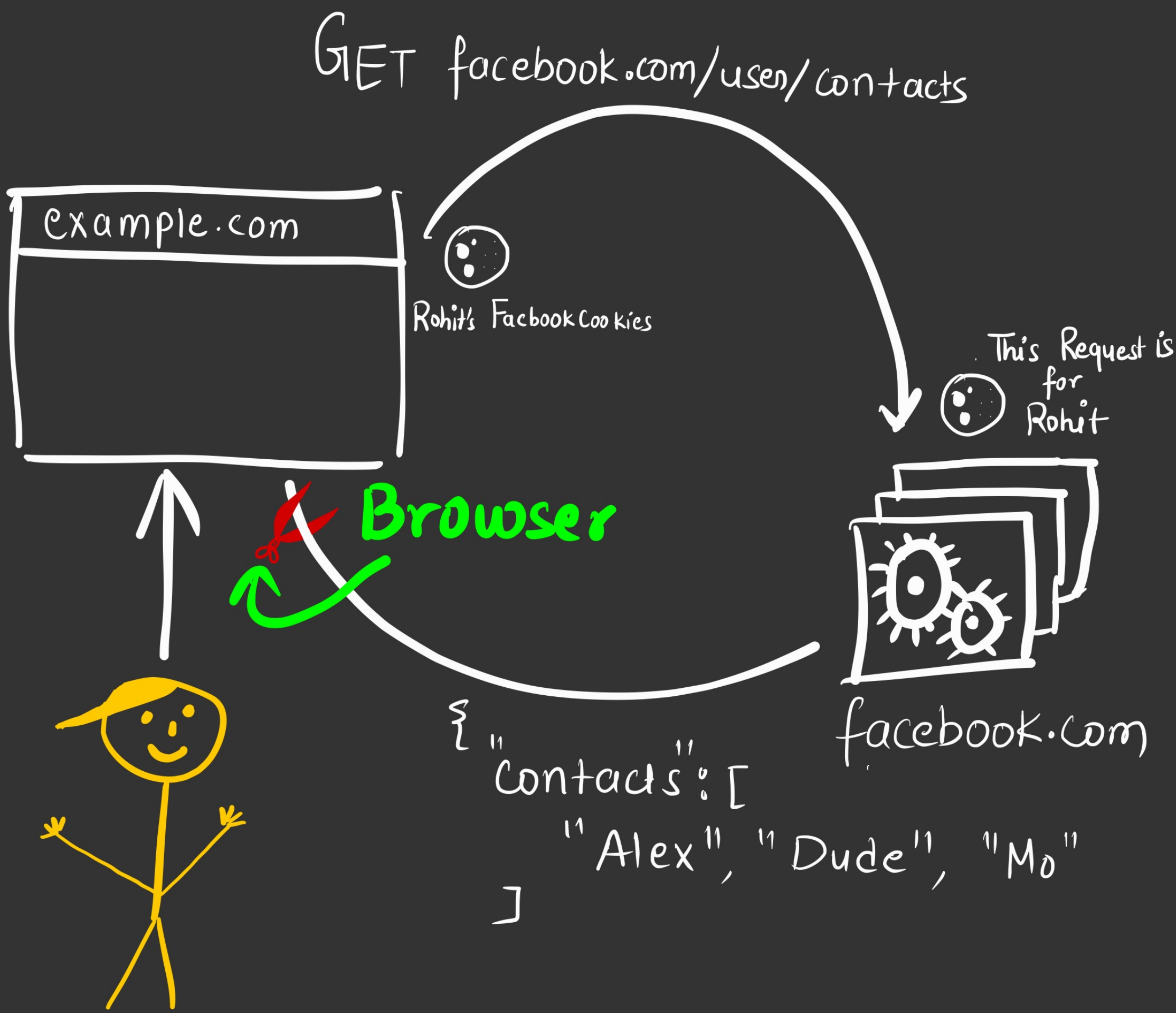
World without SOP ?



*Without SOP, example.com will be able to access Rohit's facebook's Contacts.

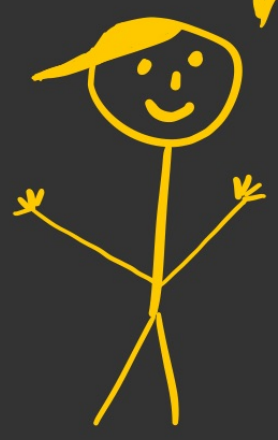
sec_r0

World without SOP ?



* With SOP, though the response will come but browser will make sure not to share it with example.com .

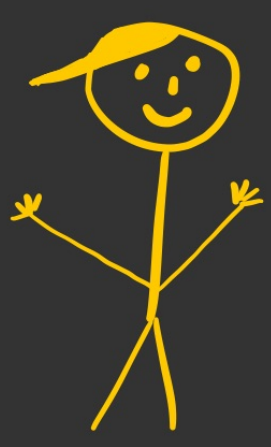
sec_r0



Some Important points

* There is no single SOP

* what I mean is: SOP is implemented differently for different DOM elements



Have you ever seen brower blocking

<script src="..." />

OR

<link src="..." />

?

Nope, never

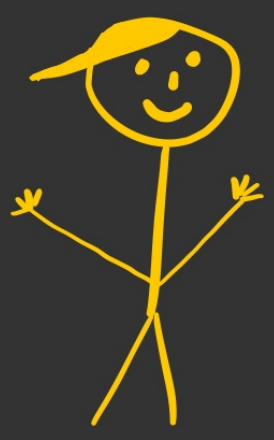
That's because for these two tags SOP is least restrictive

sec_r0

If that is the case then, I can get user facebook contacts using this method, right?

```
<script src="facebook.com/contacts"/>
```

And, browser will get contacts and I bypassed SOP

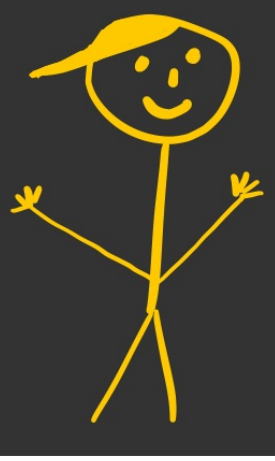


Not actually !!

You may bypass SOP like this but you can't access the response of this script !!

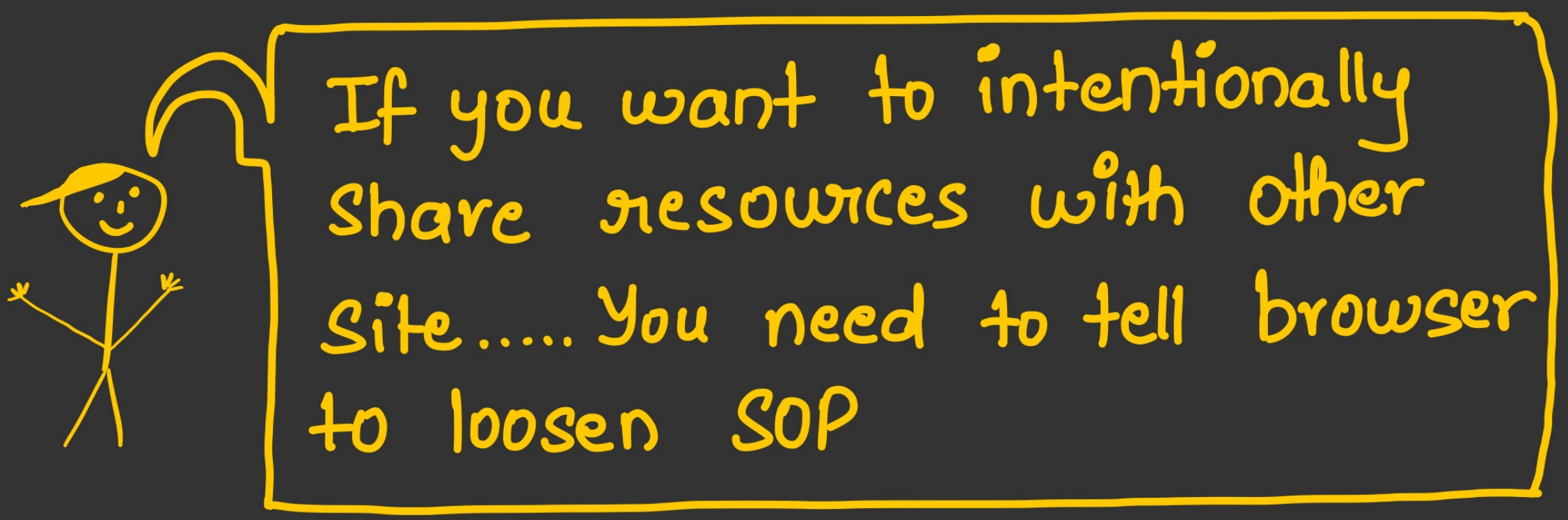
Trust me there is now way at all.

*Your hack worked, i.e SOP is bypassed but you still donot have user contacts.



are always a step ahead ....

sec_r0

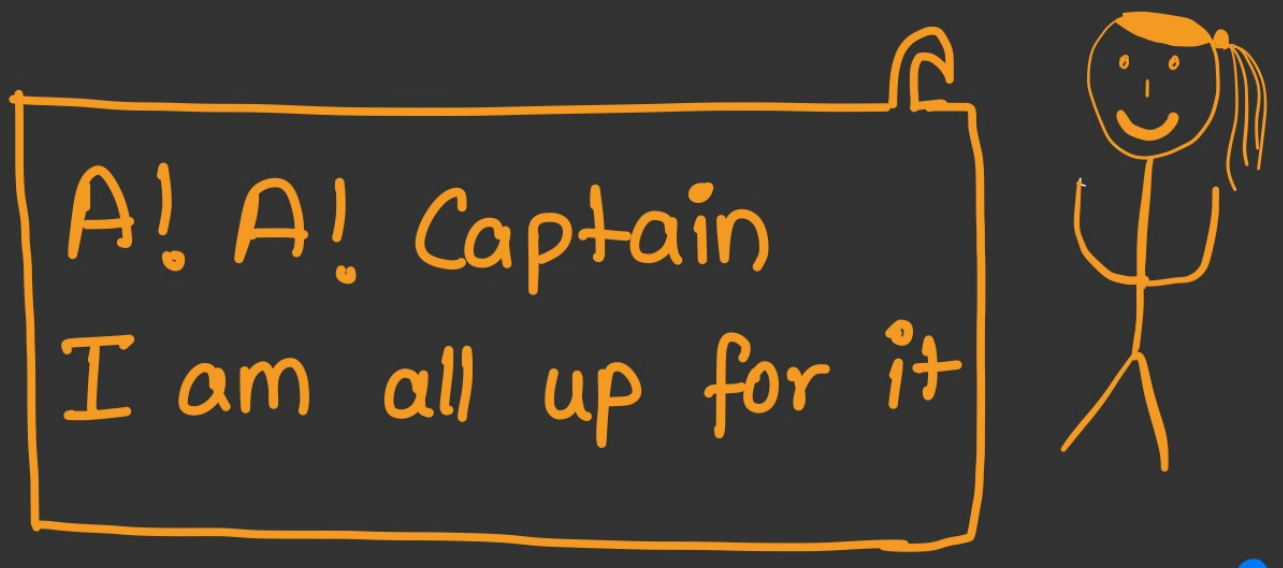


There are couple of ways of doing this.....

I know two of them, wanna learn?

Domain & lowering

CORS



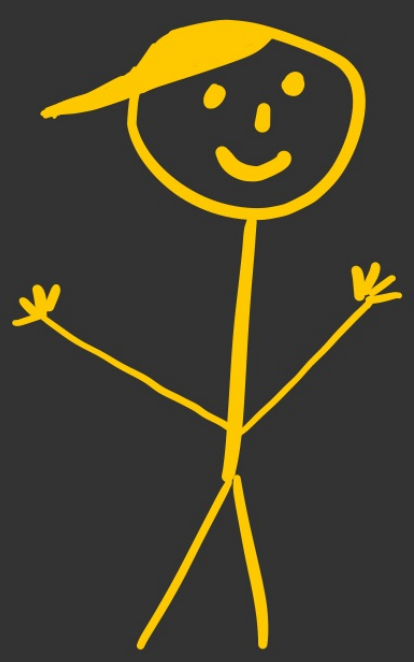
sec_r0

Bypassing SOP

* Bypassing SOP with CORS headrs is more secure way of allowing cross-site interaction.

Say with me....
when sharing resource use CORS

that's how messenger.com can access contacts from facebook.com ?



Yes !

CORS → Cross Origin Resource Sharing

headers

So now lets talk in detail about CORS headers

sec_r0

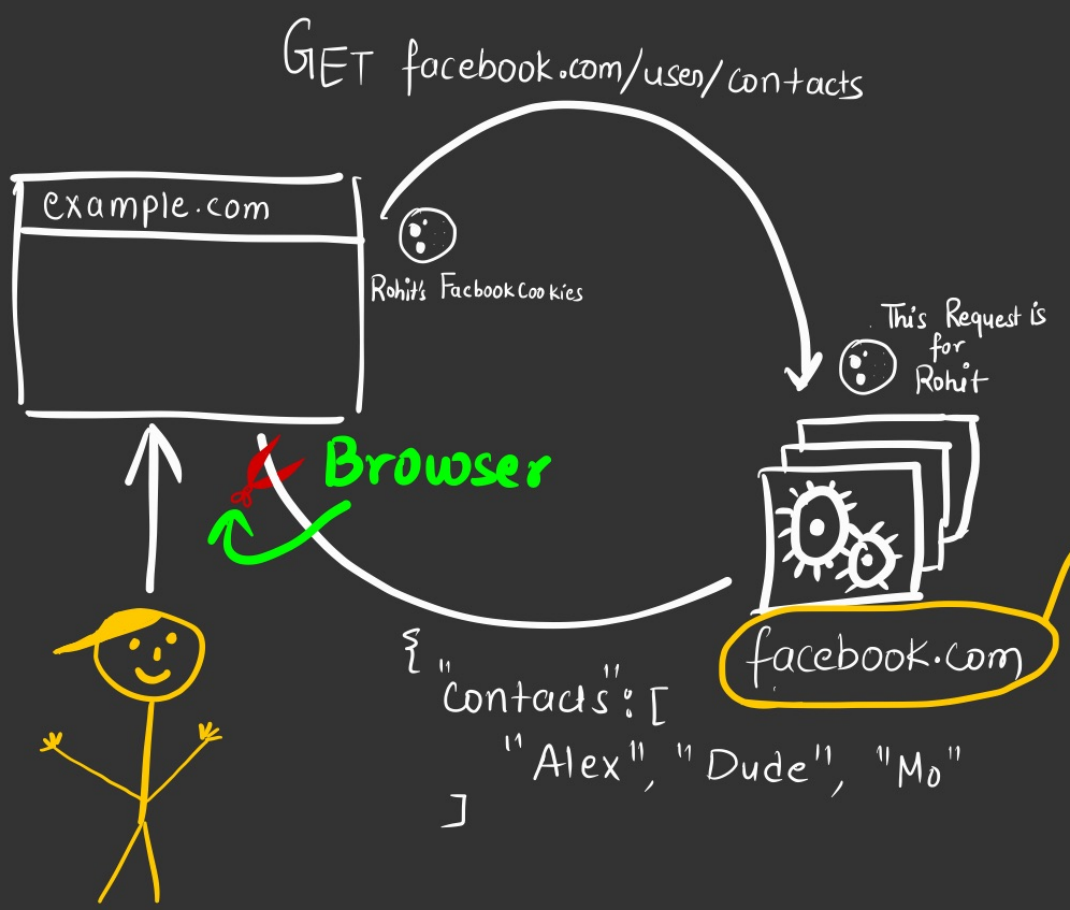
CORS Headers

* CORS headers are set by response server and are verified by browser before granting access.

* Various headers, most important

⇒ Access-Control-Allow-Origin

⇒ Access-Control-Allow-Crendentils



These response header has to be set by Facebook.com

Access-Control-Allow-Origin

This header can take single domain

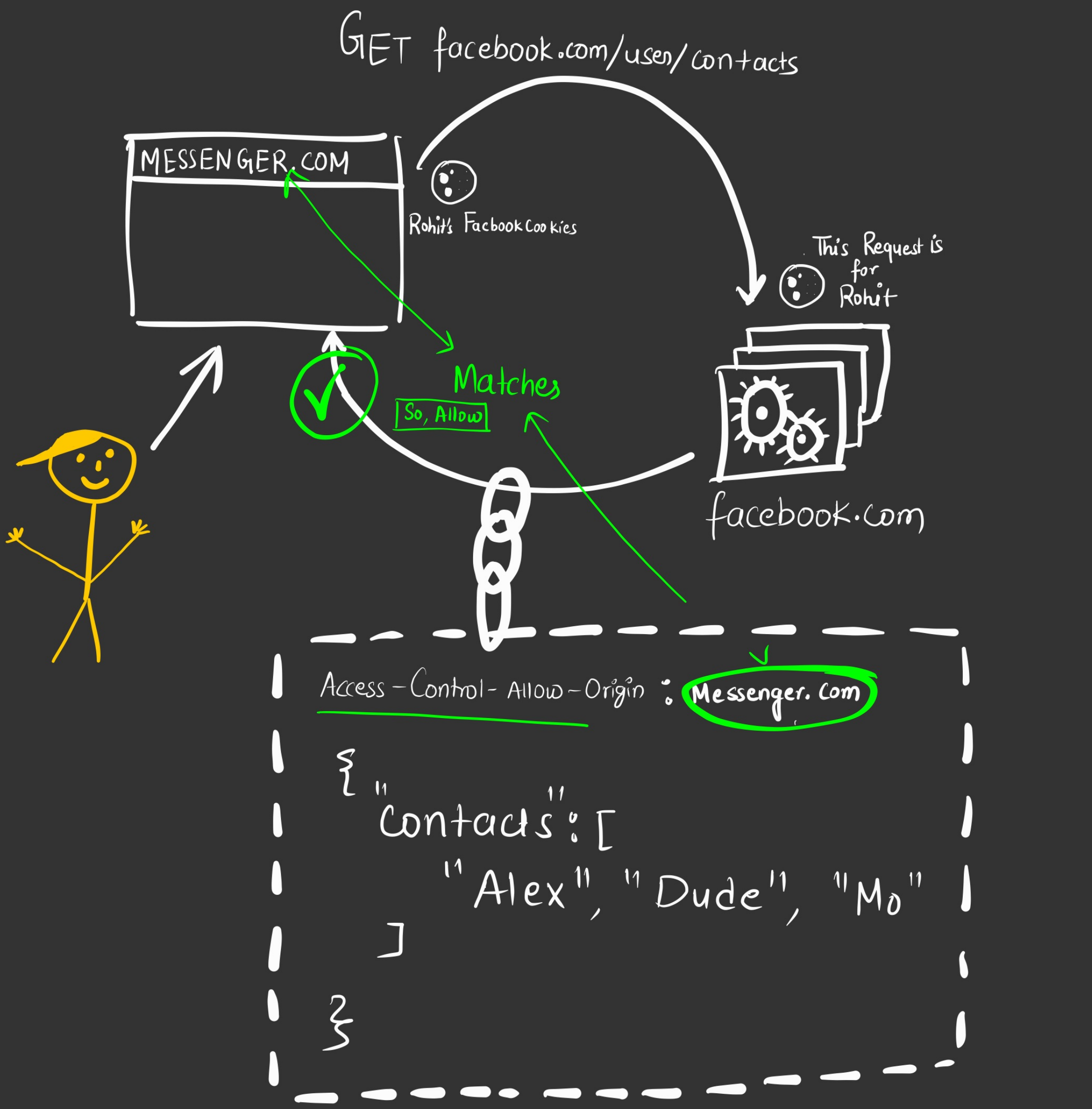
OR

wild card '*' for all/any domains.

OR

'null'

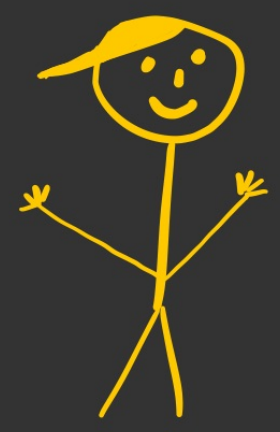
* So if facebook wants to share Contacts with messenger, it need to set that header.



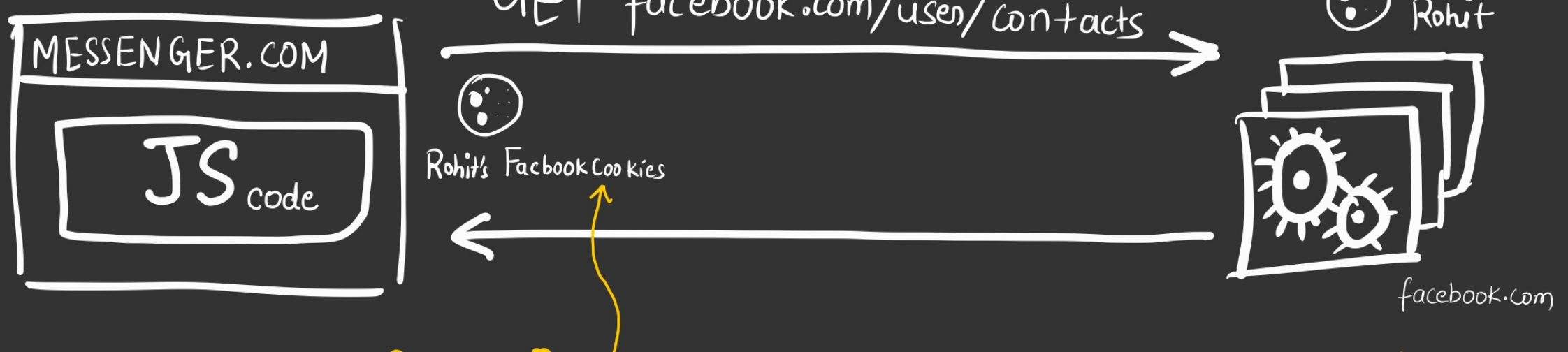
* If in response, brower see the this header it will allow messenger to access response from facebook.

Otherwise CORS error will come.

sec_r0



But there is a catch

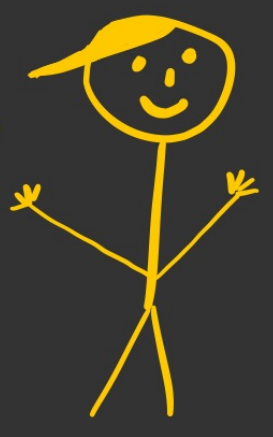


* JS running in messenger.com has to send Credentials as well to fb.com.
  - ↳ That Could be Cookie Tokens ets
  - ↳ For facebook to know from who the request is coming from.

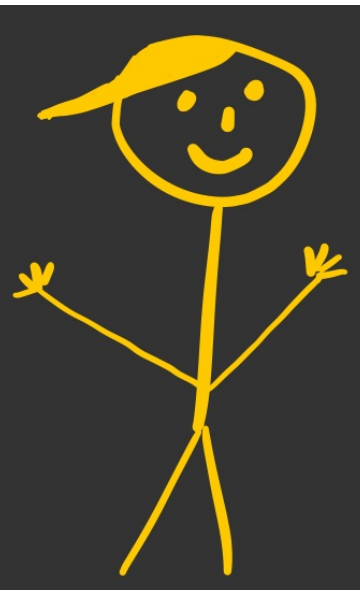
* This is done with setting Credentials = True, while making XHR request.
* And if server accepts the credentials and the response comes in as was coming in previous case; ↓
  Response again will be blocked by browser

But why !!

In short that is for server to make sure credentials are allowed in Cross-site-Request by originating domain



sec_r0



Let me explain that again

MESSENGER.COM

JS code
Credentials = true

GET facebook.com/user/contacts

Rohit's Facbook Cookies

This Request is for Rohit

facebook.com

Browser

Response comes in

blocks it, as browser wants the server fb.com to confirm that Credentials can be set from messenger.com site

fb.com has already confirmed that access is allowed for messenger.com

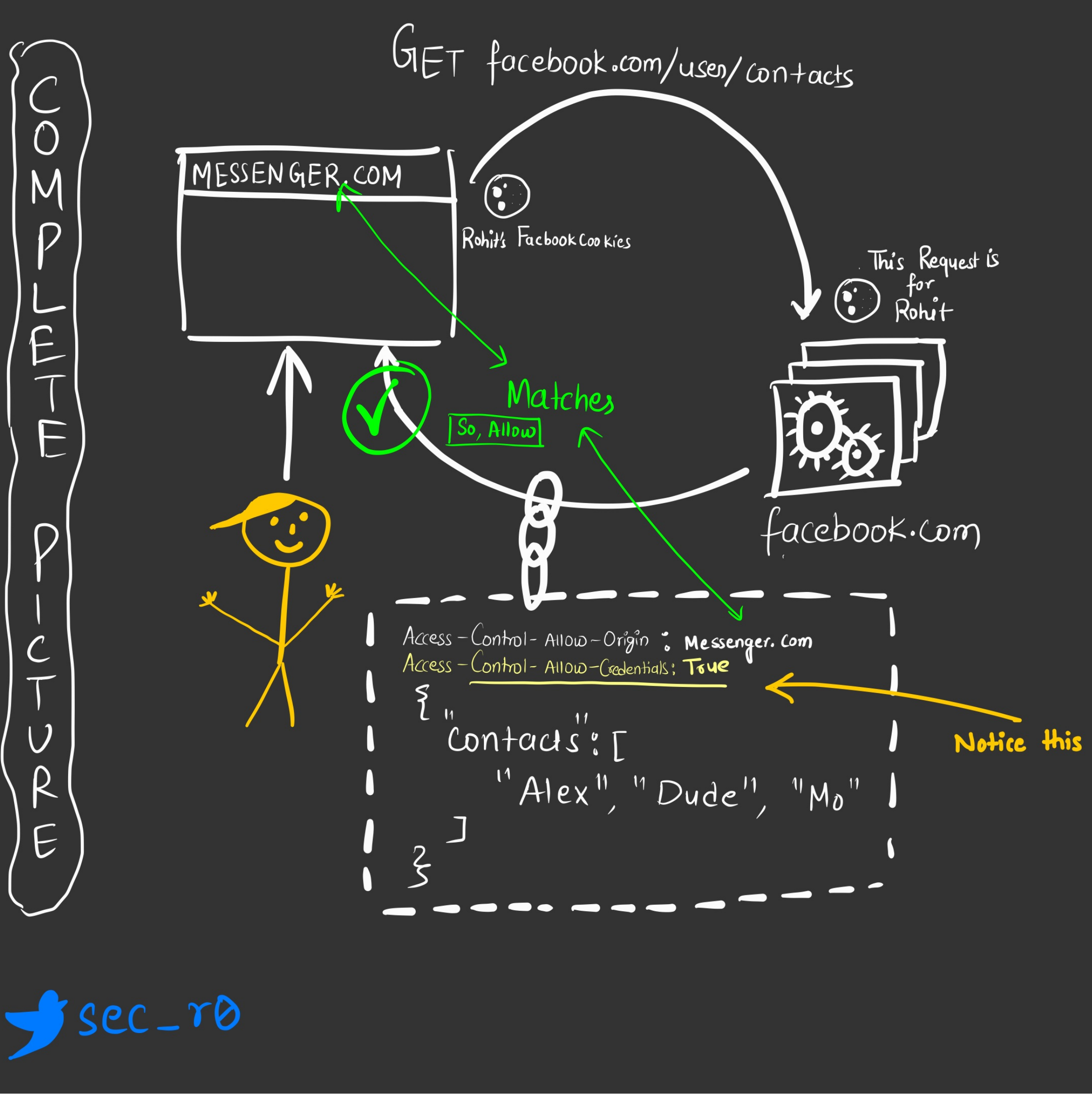
Yes, but fb.com also need to Confirm, messenger.com can pass credentials to fb.com via XHR

And, that Confirmation is made by fb.com by Setting Access-Control-Allow-Credentials

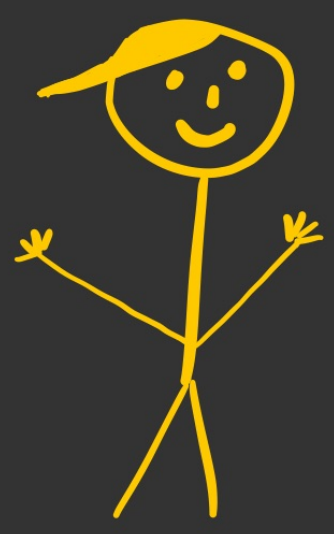
sec_r0

Access-Control-Allow-Credentials

* When browser sends authenticated req (as in previous case), & resource is protected, SOP will block it if response donot contain header.
* That's where this header comes in for rescue
* So, browser will allow this response only if server responds with this header in response.



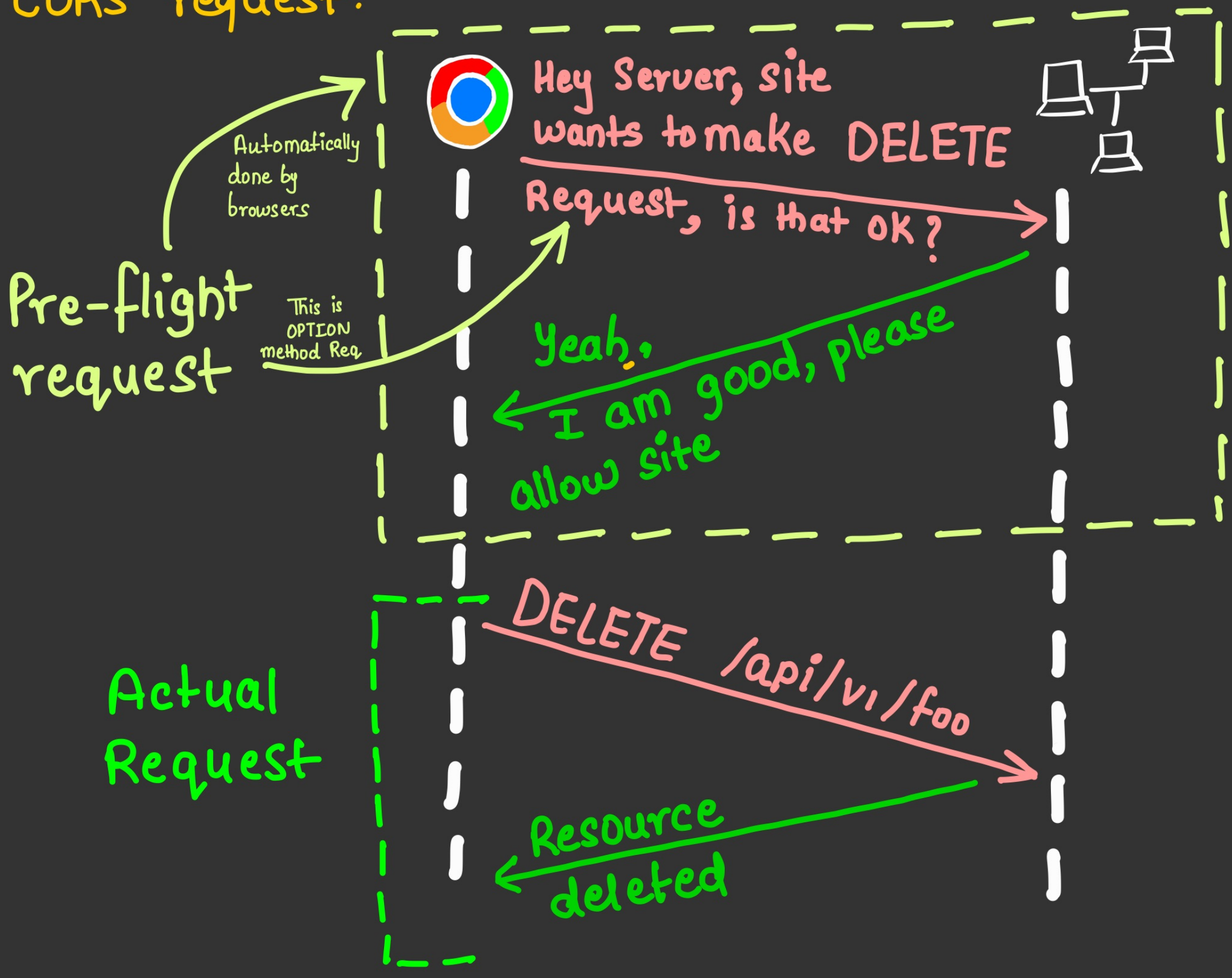
sec_r0



In SOP to Support backword Compatility there is something called as pre-flight Request

Nothing to do with Site Security

* Pre flight request is a sanity check, that is just used to check if server is capable of handling CORS request.



* Browser sends pre-flight requests only for non simple request methods i.e other than GET and POST methods.

sec_r0

Let me tell you another hack of bypassing SOP

DOMAIN LOWERING

This is not encouraged. As there may be security issue.

I want to know
Domain lowering as well
as the security issue
that might be there with it

Hold your seatbelts tight.....

sec_r0

Domain Lowering

* Javascript Running in a site can change its origin with Some limitations through document.domain property

→ can only change origin to super domain name of Current domain name

Super domains

Foo.bar.qux.example.com

Sub domains

Eg Subdomain of example.com can be

qux.example.com, bar.qux.example.com,

Foo.bar.qux.example.com

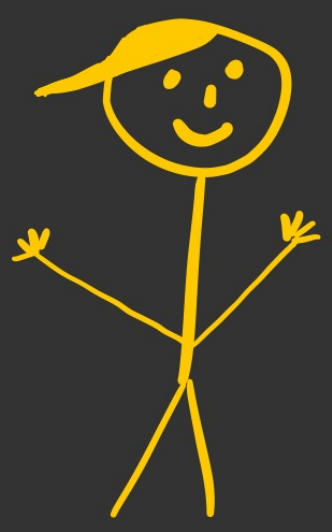
Like wise super domain of Foo.bar.qux.example.com can be

bar.qux.example.com, qux.example.com, example.com

* During domain lowering you can only change domain to super domains.

sec_r0

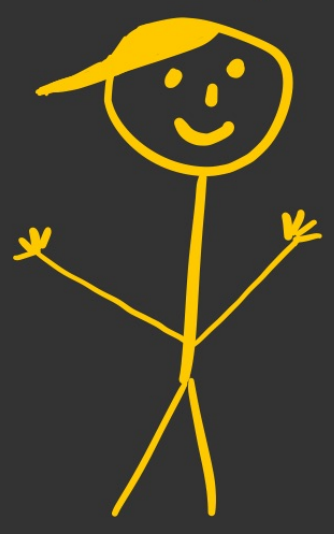
notice this technique is only for cross domain interaction in sub domain sites



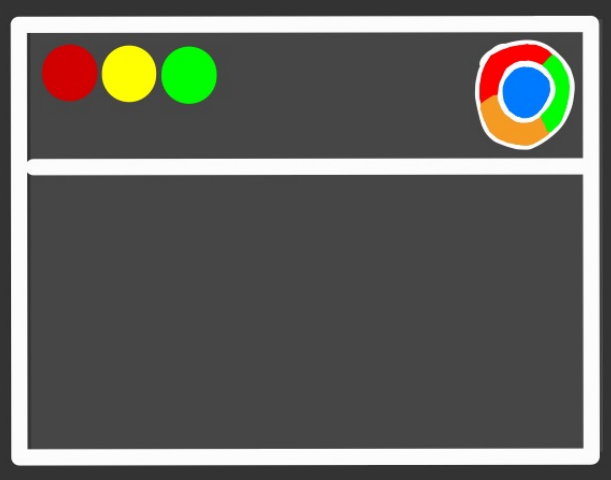
Exactly, because we can't change origin to anything....
Only superdomains

Sites on subdomains may or may not sit in same server

Now, the main question, How cross site interaction is done with domain lowering



1

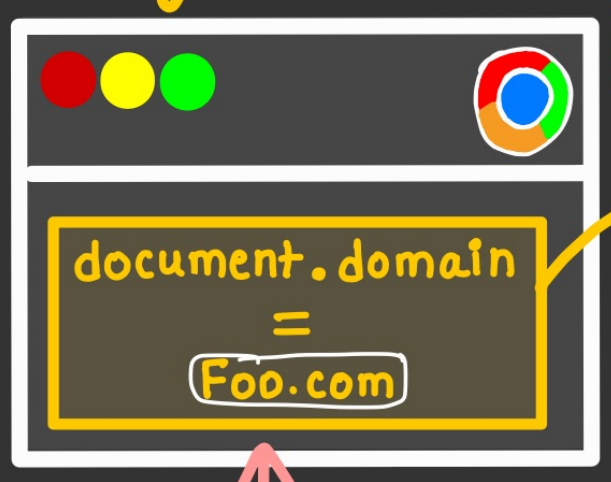


b.foo.com

wants to
talk to

a.foo.com

2 b.foo.com

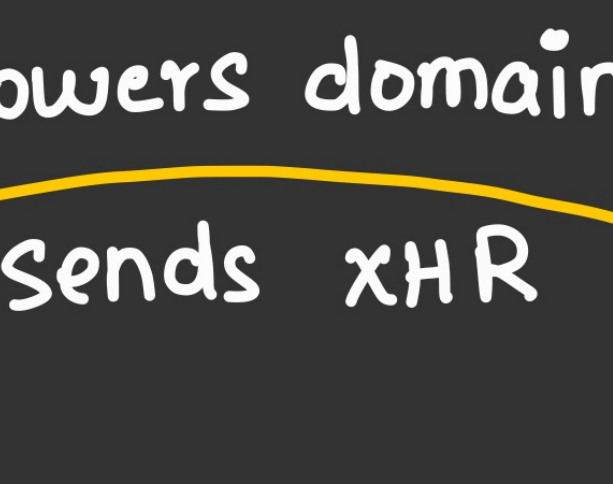


1 Lowers domain

2 Sends xHR

a.foo.com

3 Server Sends the response



4 JS in response lowers the domain

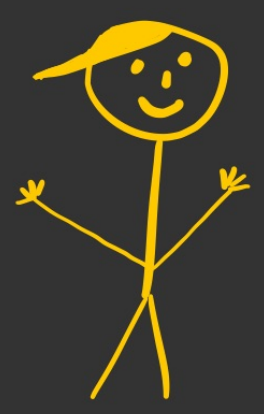
5 browser check origin or document.domain

6 Allows response if matches

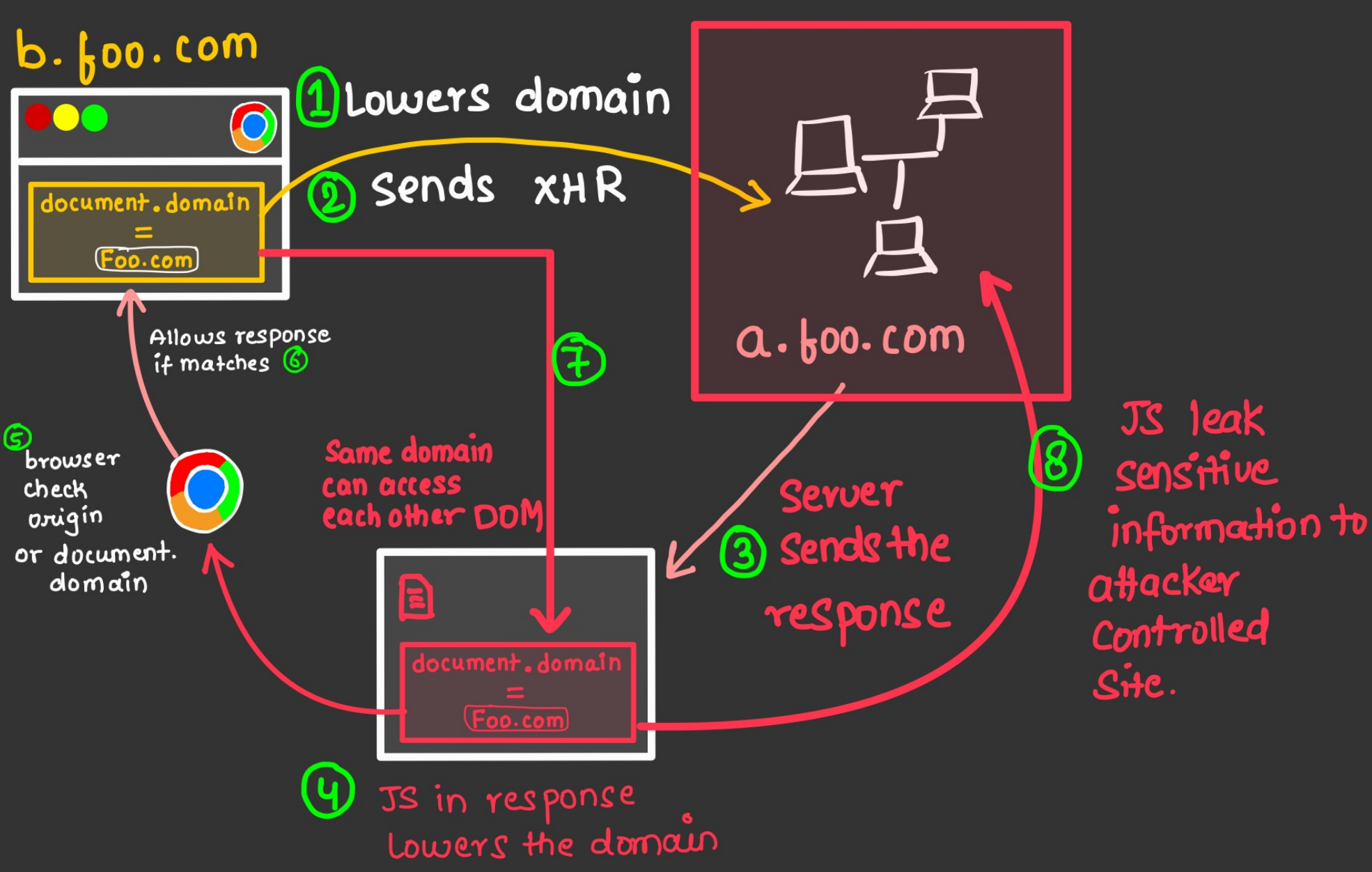
sec_r0

All good right?

No, there is a Security issue



The issue is if attacker compromised site a.foo.com



* Since a.foo.com and b.foo.com are now under same domain this makes b.foo.com compromised as well.

* Severity of issue depends on criticality of a.foo if a.foo.com handles login Auth for entire foo.com then with this attacker will be able to Control the login for all of foo.com subdomains.

sec_r0

That was a fun ride.

So that's all about SOP ??

For now yes !!

Thanks for Reading

sec_r0

Join for updates

Read More Zines
@
securityzines.com

Also, thankyou

@musana
@nirali0
@susam
@KhajaSubhani

Sneha Anand

for Review