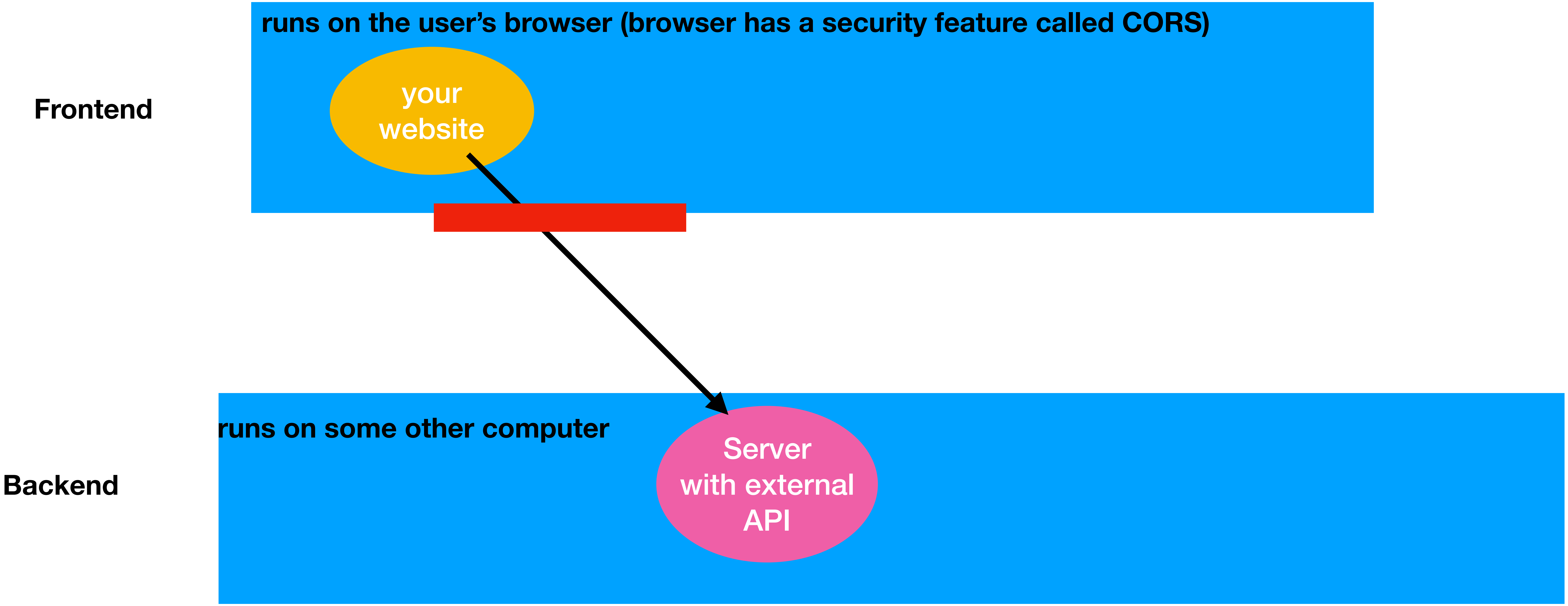
CROSS ORIGIN RESOURCE SHARING (CORS)

CROSS ORIGIN RESOURCE SHARING (CORS)



Frontend

runs on the user's browser (browser has a security feature called CORS)

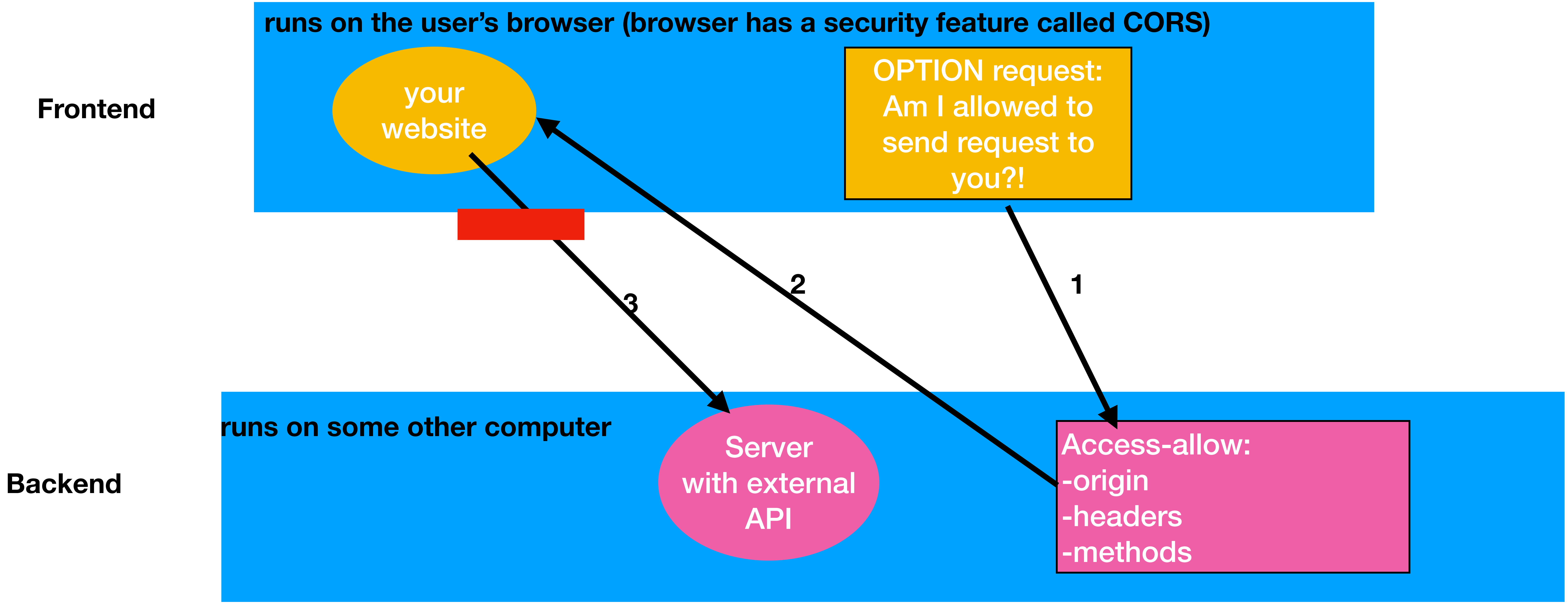
your website

Backend

runs on some other computer

Server with external API

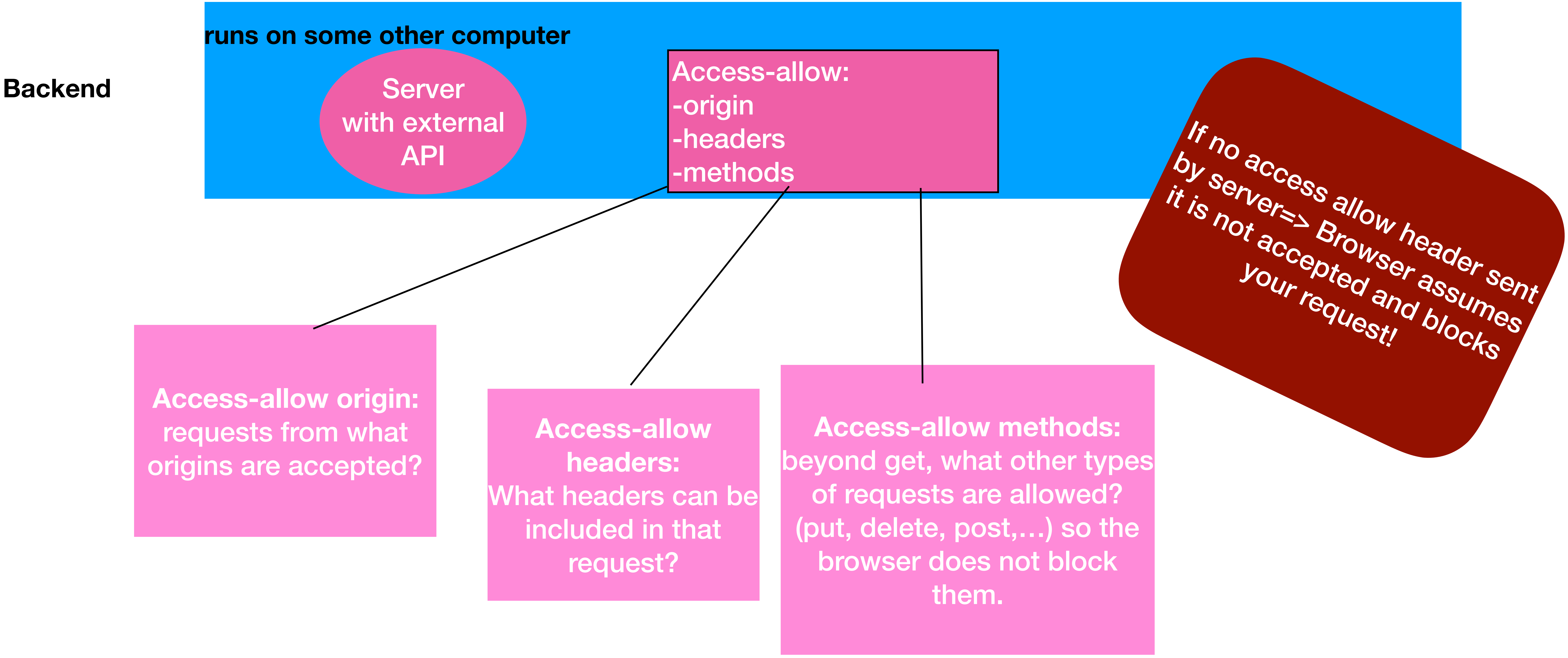
CROSS ORIGIN RESOURCE SHARING (CORS)



Frontend
runs on the user’s browser (browser has a security feature called CORS)
your website
OPTION request: Am I allowed to send request to you?!
1
2
3
Backend
runs on some other computer
Server with external API
Access-allow:
-origin
-headers
-methods

- The minute your Browser sees you are making a request to some other url other than the host you’re from, it first block your request:
    - WAIT! You are **Requesting Requests** from **another Origin!**
    - I let you do that only if that origin says that it’s ok.
    - So let’s send an **OPTIONS request** first to see if *I’m allowed to make this request?*
    - It asks for a handful of different headers (access-allow…)

CROSS ORIGIN RESOURCE SHARING (CORS)



Backend

runs on some other computer

Server
with external
API

Access-allow:
-origin
-headers
-methods

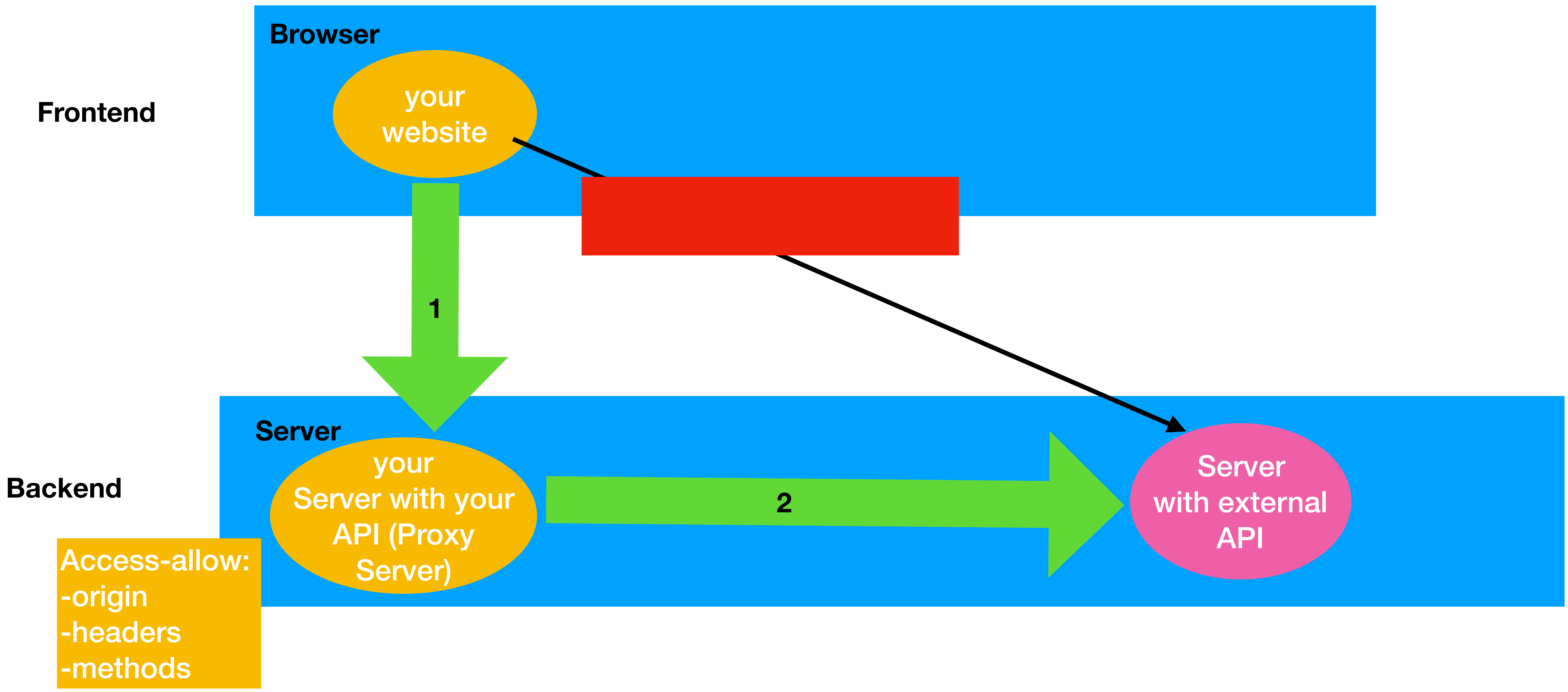
Access-allow origin: requests from what origins are accepted?

Access-allow headers: What headers can be included in that request?

Access-allow methods: beyond get, what other types of requests are allowed? (put, delete, post,...) so the browser does not block them.

If no access allow header sent by server=> Browser assumes it is not accepted and blocks your request!

What if no request is allowed?



Frontend

Browser

your website

1

Backend

Server

your Server with your API (Proxy Server)

2

Server with external API

Access-allow:
-origin
-headers
-methods