Web Cache Deception

Why caching?

- It's fast
- Reduces the load on the main server
- Caching static files or big values from main DB

How does it work?

Client

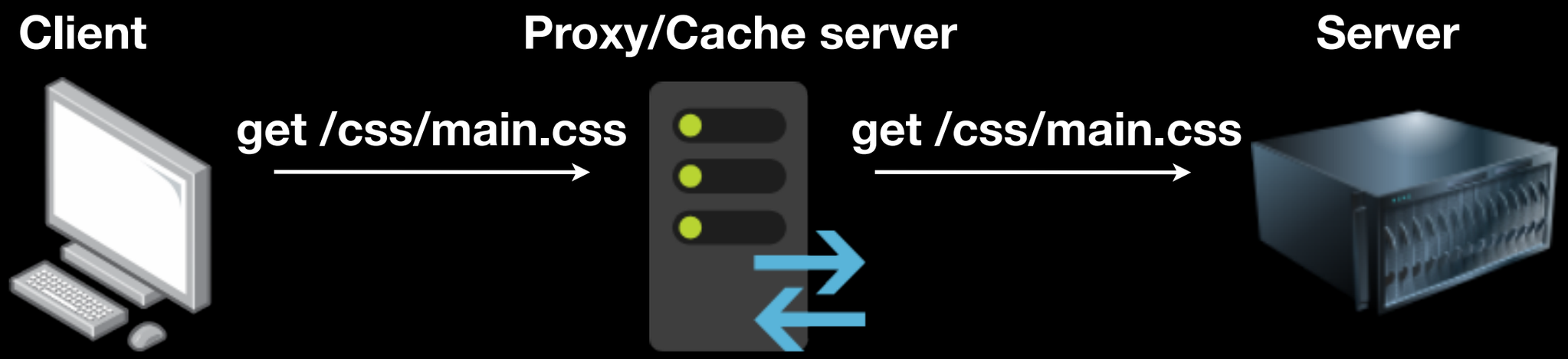
get /css/main.css

Proxy/Cache server

Server

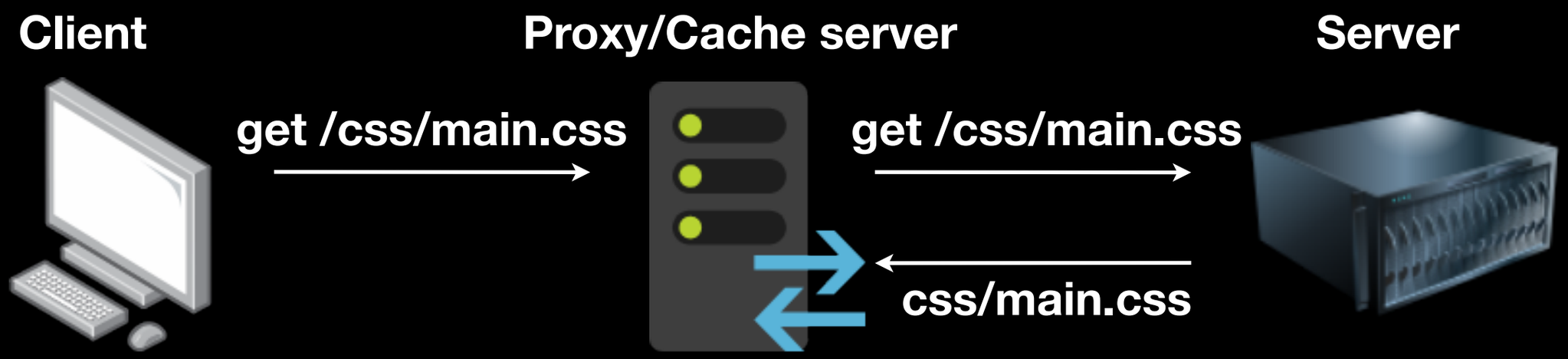
Cache storage

How does it work?



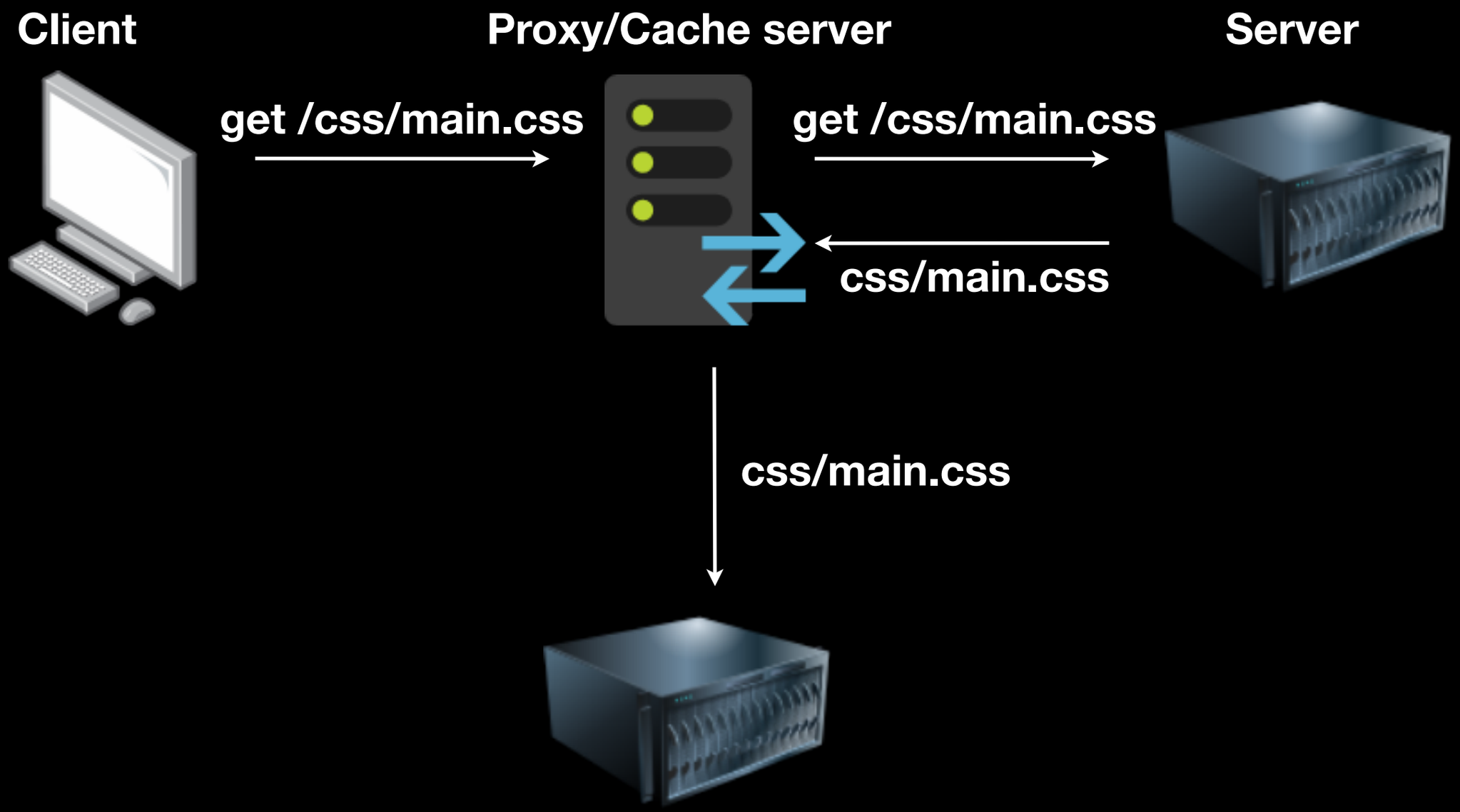
Cache storage

How does it work?

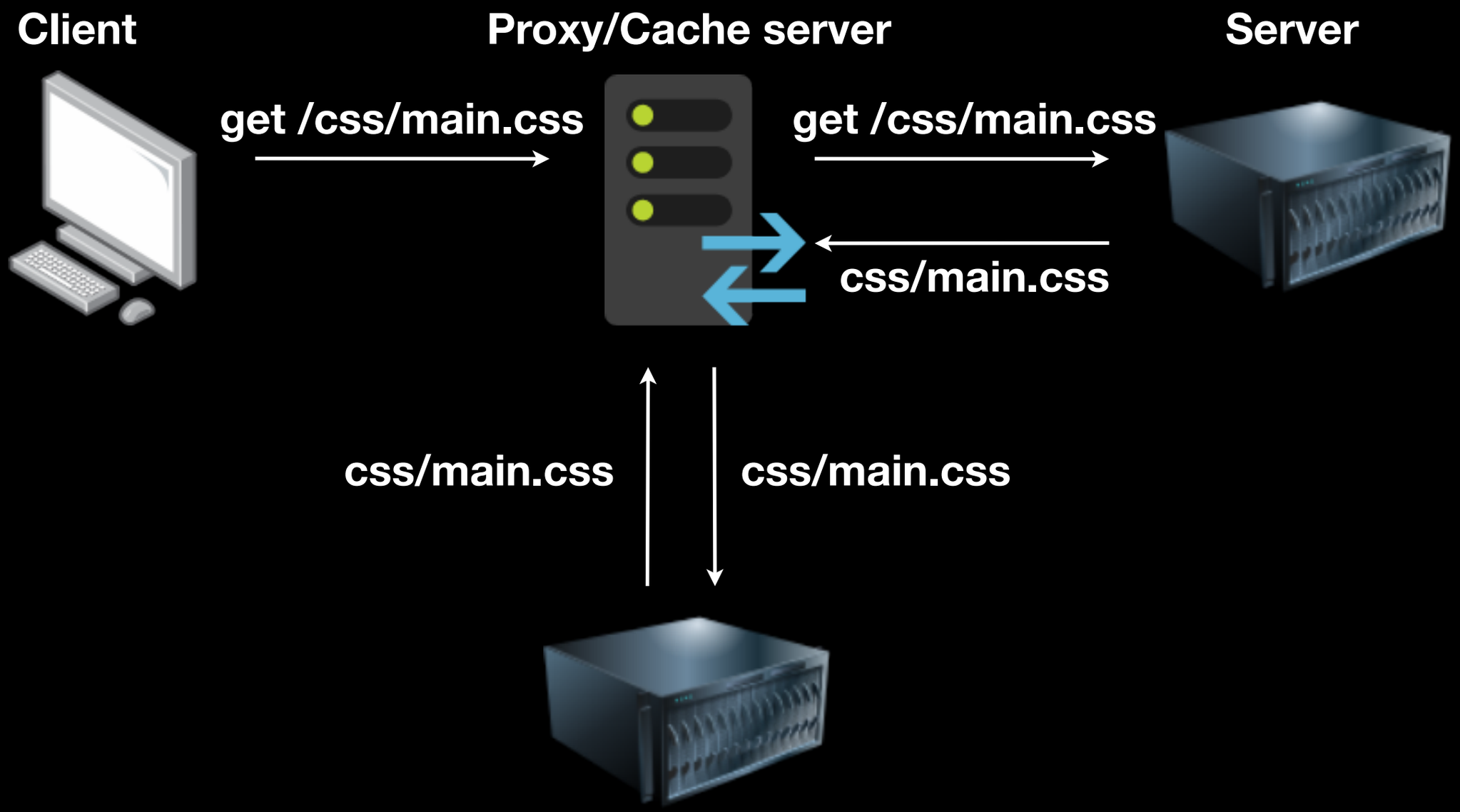


Cache storage

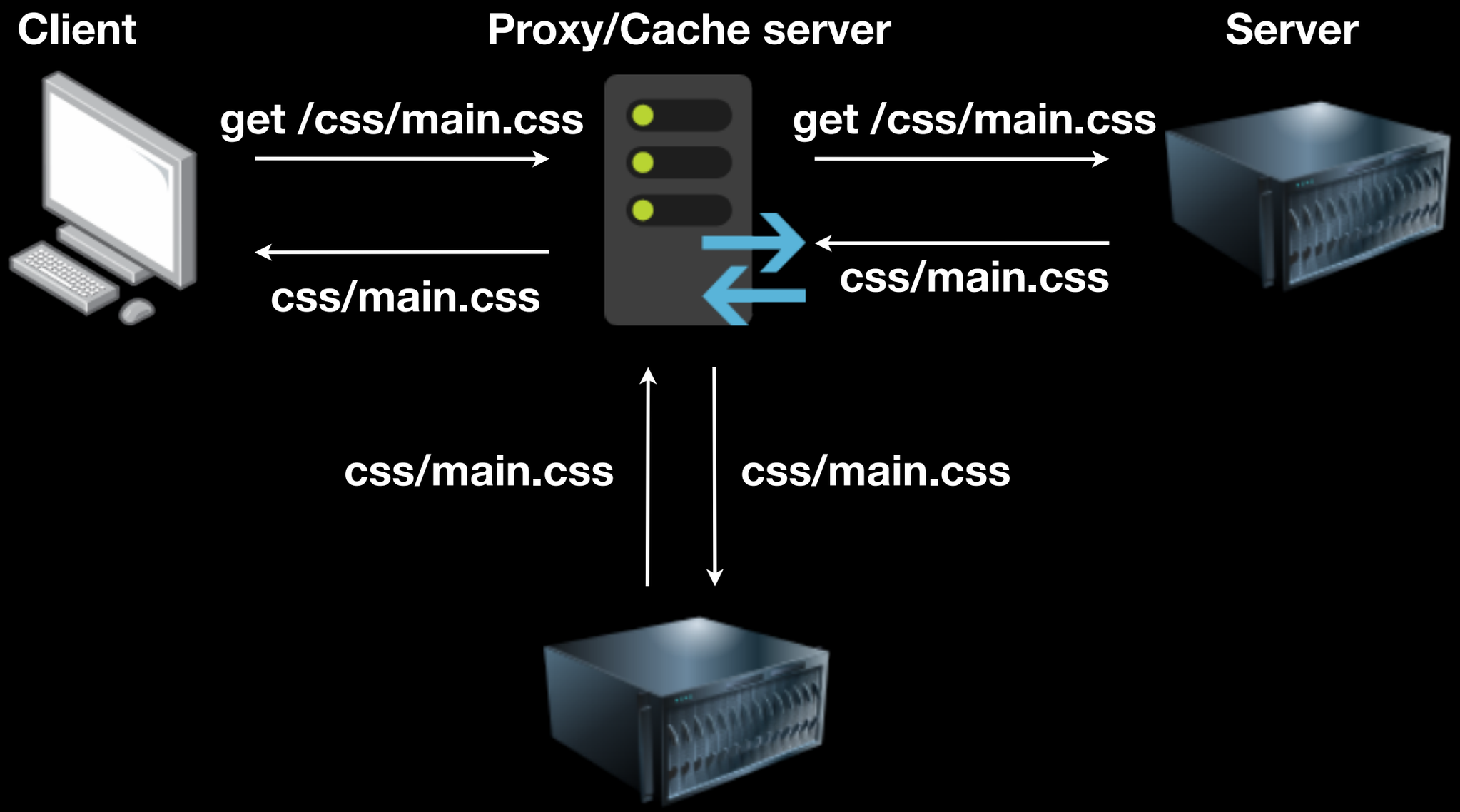
How does it work?



How does it work?



How does it work?



Web Cache Deception

Profit:

- Leak victim's data
- Cache poisoning
- Upgrade self-XSS to stored XSS!

Web Cache Deception

Example conditions:

- Application is caching anything with css/png/etc in the end of url
- Application returns profile html page without checking end of the url:

```
/profile == /profile/1.css == /profile/blabla/1.png
```

Web Cache Deception

Attack plan:

- Attacker gives victim a link to non-existing content, which will be cached because of .css in the end of url

http://app.com/profile/blabla.css

- Victim opens this link, get his profile page content and application saves it in cache

- Attacker opens this url, application returns content of victim’s profile page from cache

Attack scenario

Client

Proxy/Cache server

Server

Hey! check this out
{..}/account/1.css

H4ck3R

Cache storage

Attack scenario

Client

Proxy/Cache server

Server

Seems legit, ok

Hey! check this out
{..}/account/1.css

H4ck3R

Cache storage

Attack scenario

Client

Proxy/Cache server

Server

Seems legit, ok

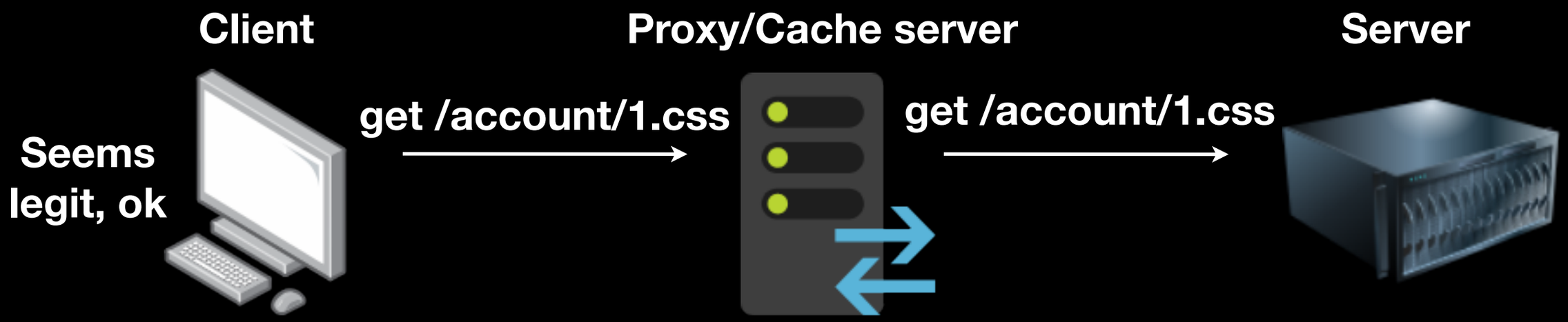
get /account/1.css

Hey! check this out
{..}/account/1.css

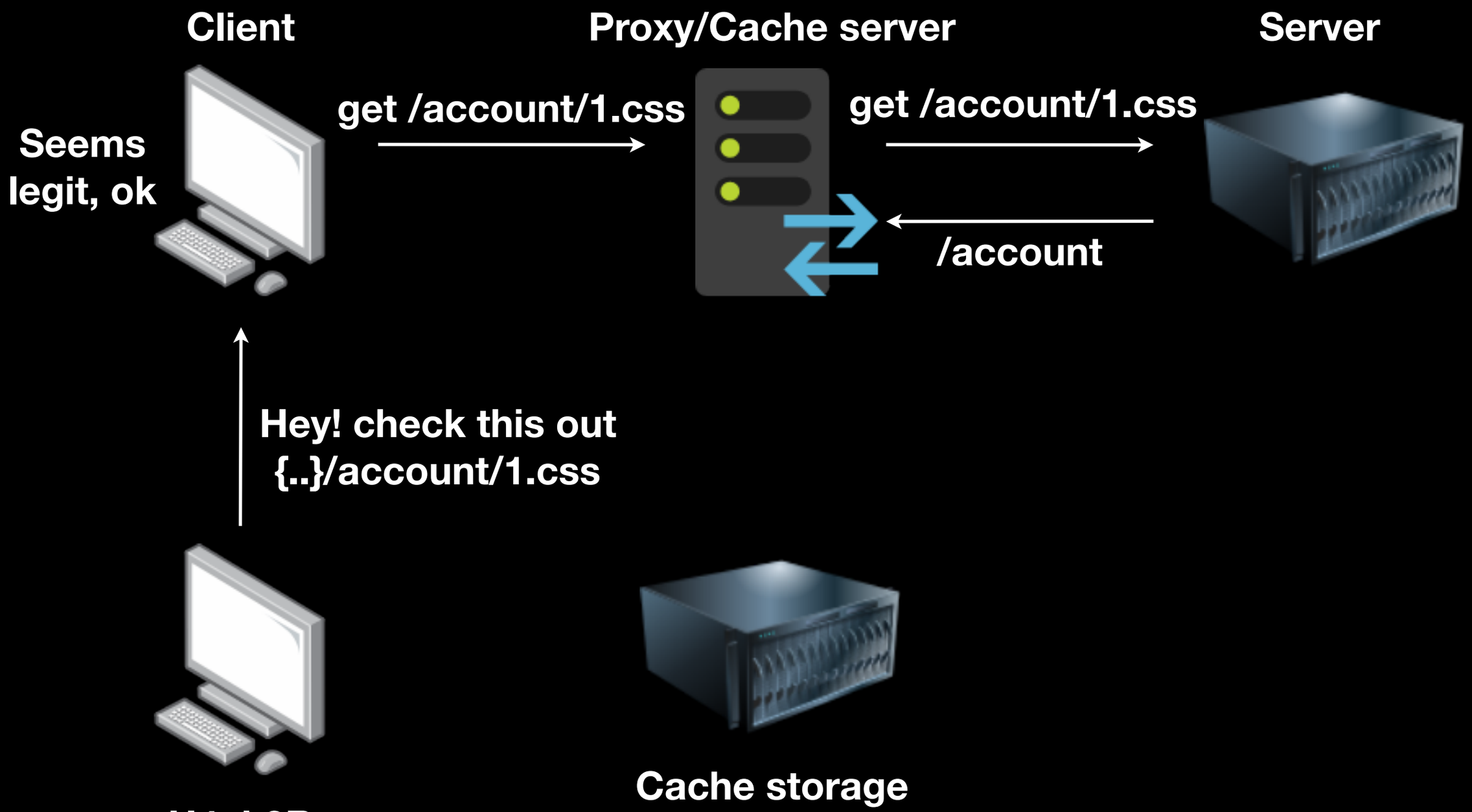
H4ck3R

Cache storage

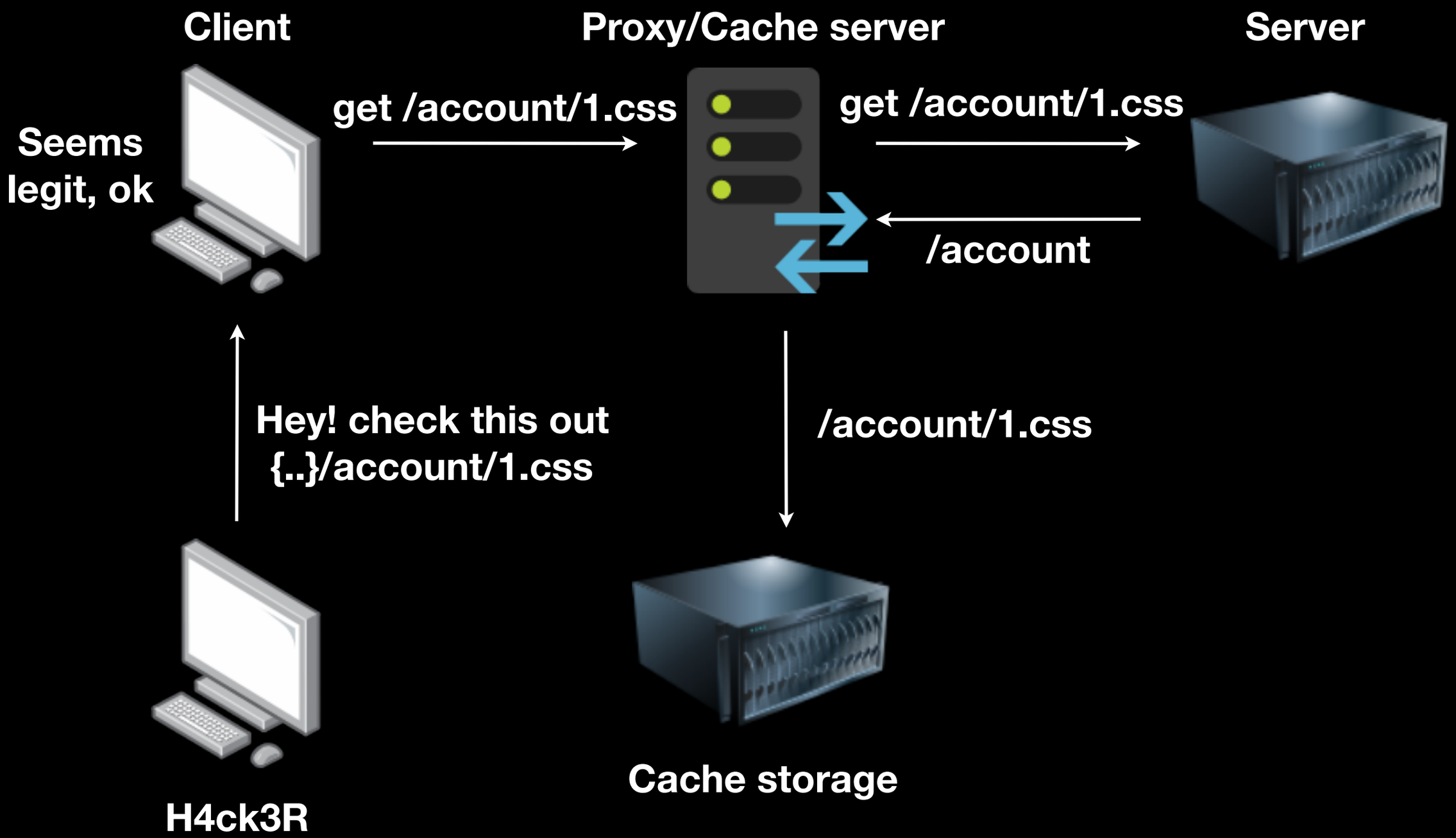
Attack scenario



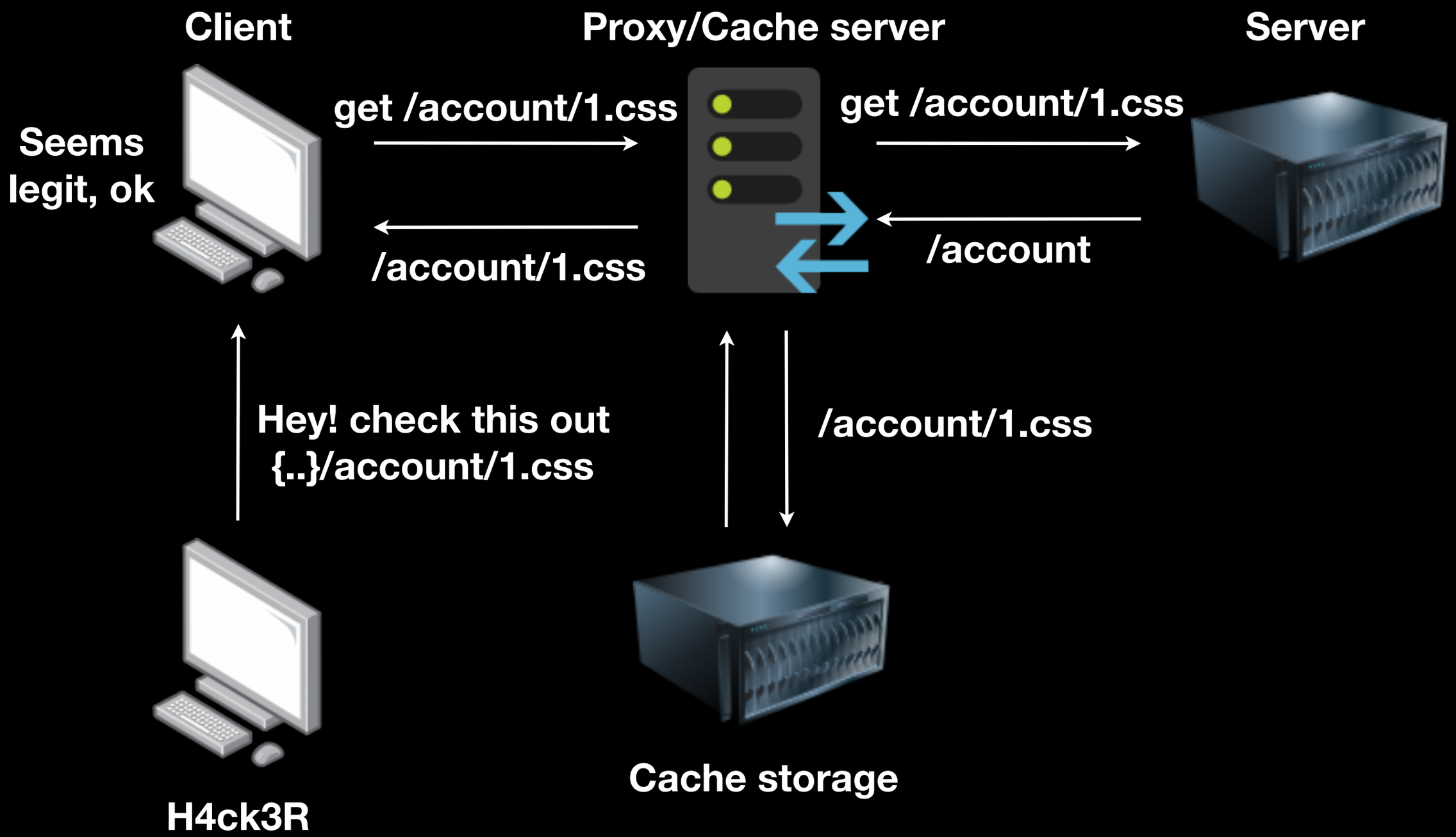
Attack scenario



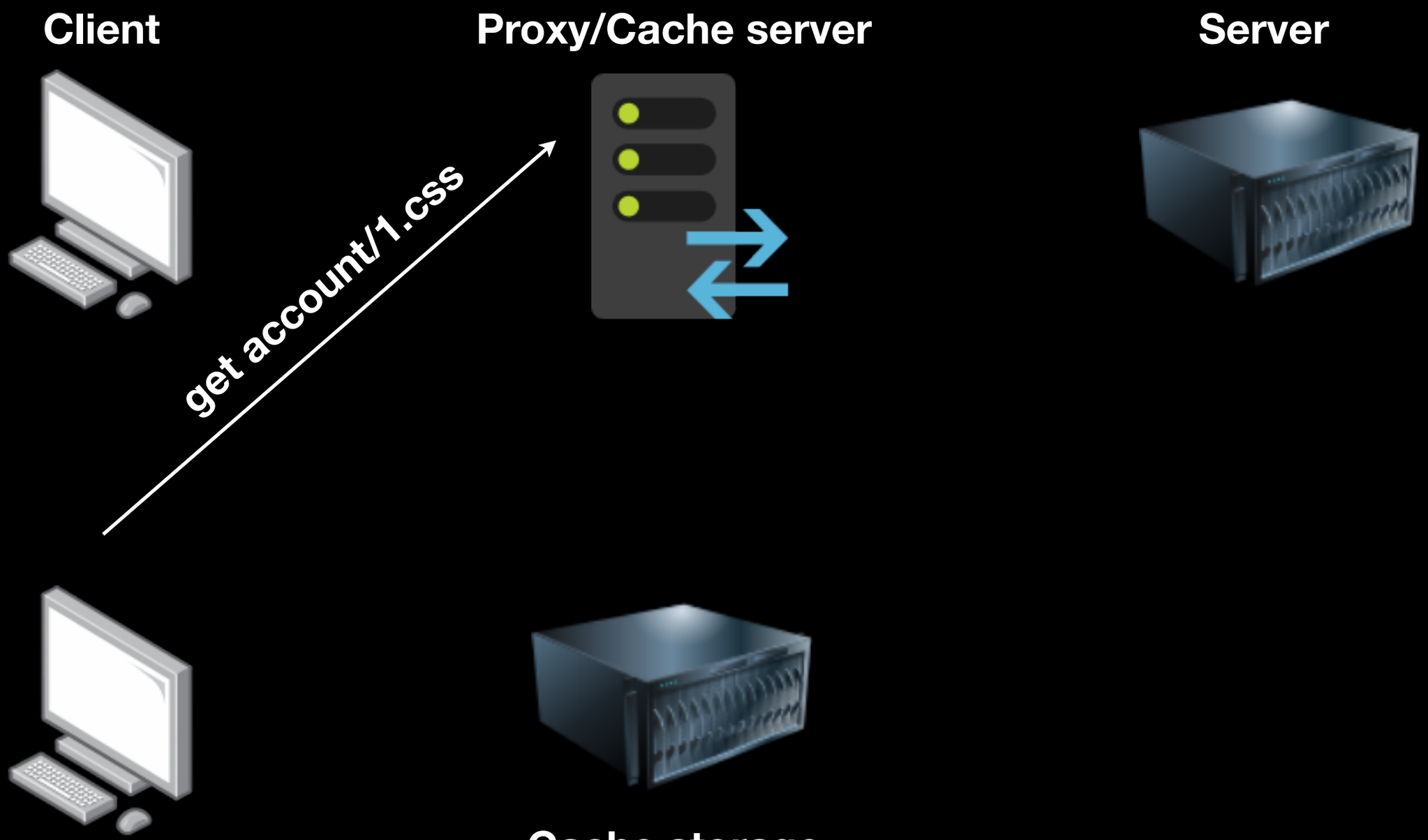
Attack scenario



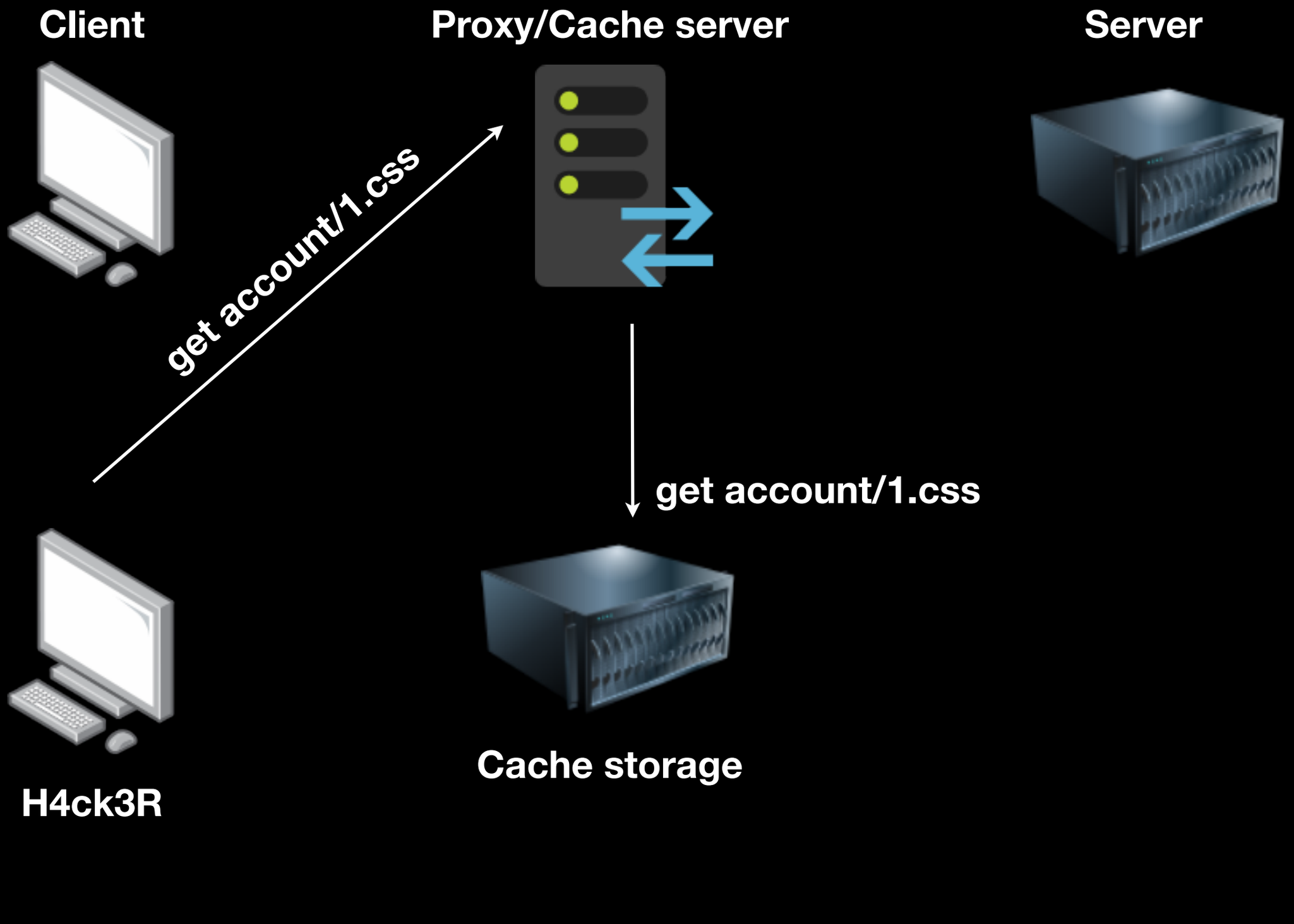
Attack scenario



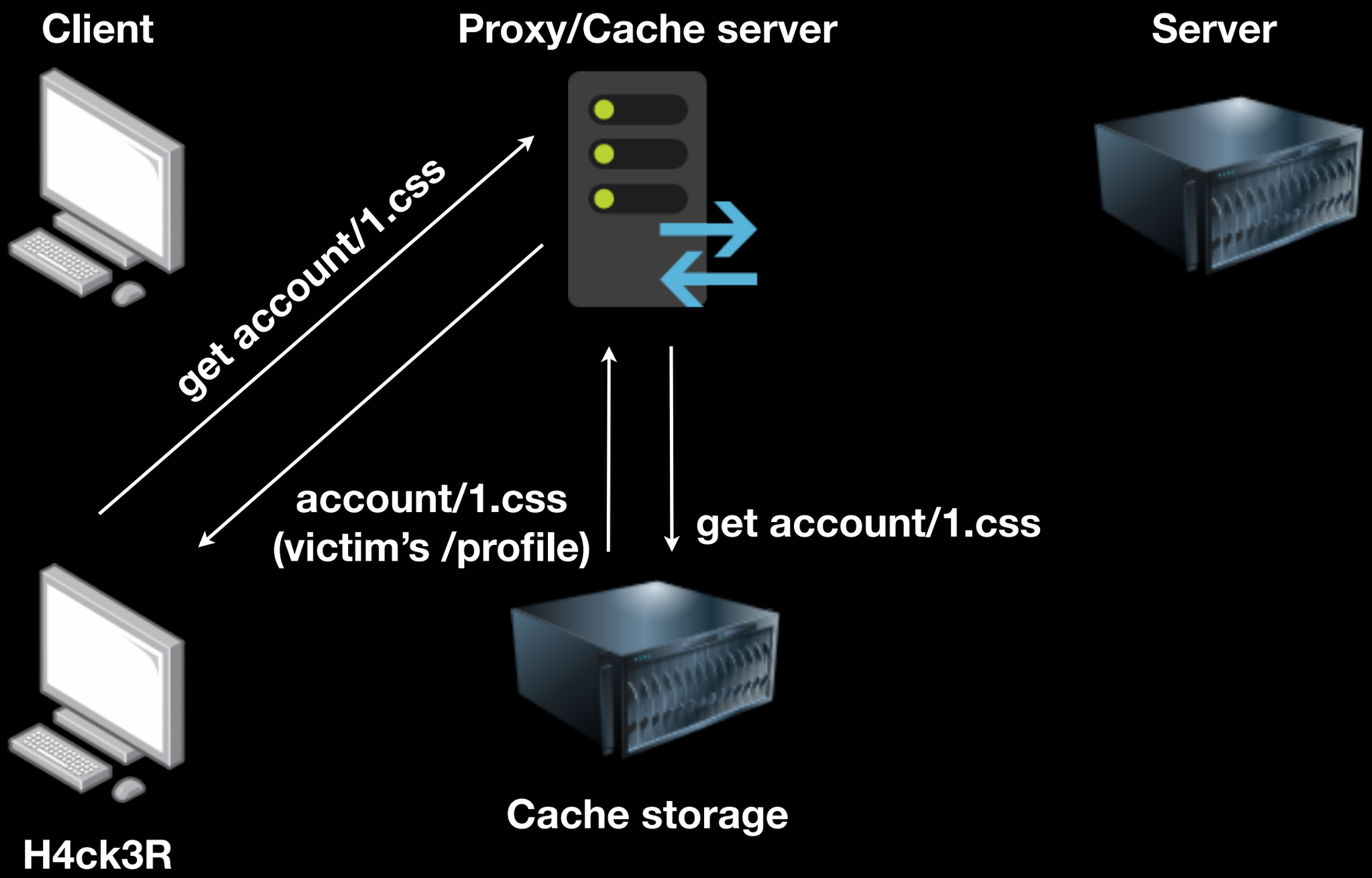
Attack scenario



Attack scenario



Attack scenario



Cache - poisoning

GET /en? HTTP/1.1
Host: www.target.com
X-Forwarded-Host:
server_url
…

GET /en?poison=1 HTTP/
1.1
Host: www.target.com
X-Forwarded-Host: **a.”**
><script>alert(1)</script>
…

...
<meta
property="og:image"
content="https://
server_url/
...

…
<meta property="og:image"
content="https://
a."><script>alert("pwned"
)</script>"/>
...