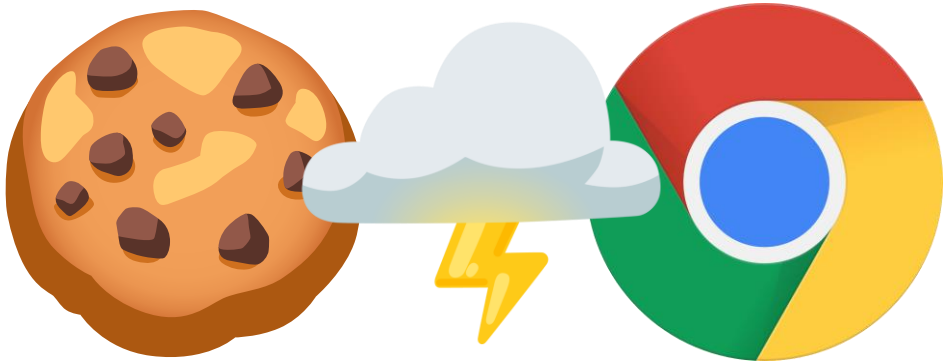
NILS GEHRKE

Google Chromes change of the Same-Site-Cookie behavior

And how this could affect an import process via EMREX



Content

About Cookies in Browsers

The Problem we had

Our Solution

Discussion

About Cookies in Browsers

About Cookies in Browsers

- Cookies are simple and small text files saved on the machine of the user
- They are widely used by a web application to identify a returning user
- Acess to cookies by a web application is restricted to the source domain by default

About Cookies in Browsers

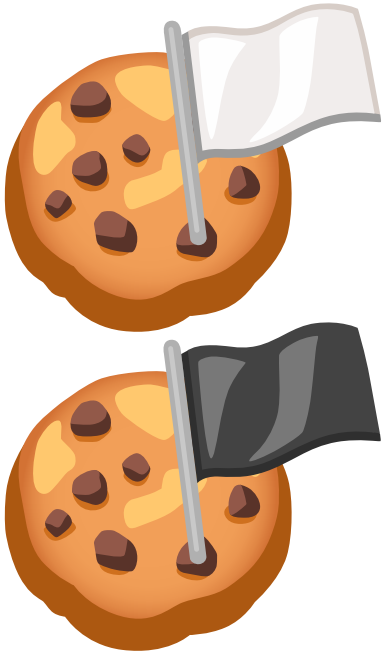
- Secure cookie access by only domain is not enough
- Cross-Site Request Forgery (CSRF) would be possible
- Example:
  <img src="https://example.com/index.php?action=delete&id=1">

About Cookies in Browsers

To prevent this, access is also restricted by site context of the user:

- Same site = Okay
- Different site = well, maybe okay …

About Cookies in Browsers



To give webdevelopers control they can flag cookies to use different level of access. The flag is called **SameSite**.

Three values are possible:

- NONE = Everything can read cookie anytime.
- STRICT = Only on same site context.
- **LAX** = Only on same site context and if user **navigates** to the site from a different site

Default was NONE in most browsers, but this changed recently to LAX.

The Problem we had

The Problem we had

- To import data via EMREX the user need to visit a different website
- The user comes back via POST request by submitting a form
- So, the user **navigates** back, the session cookie should be treated as **LAX**. No problem, or is it?

The Problem we had

Google and Mozilla have a different definition of »user ***navigates***«:

= clicking on a link (GET) **or** submitting a form (POST)

= **only** clicking on a link (GET)

The Problem we had

But stop!

Treat all cookies to LAX only for GET Request would break nearly all Single-Sign-On (SSO) Implementations out there!

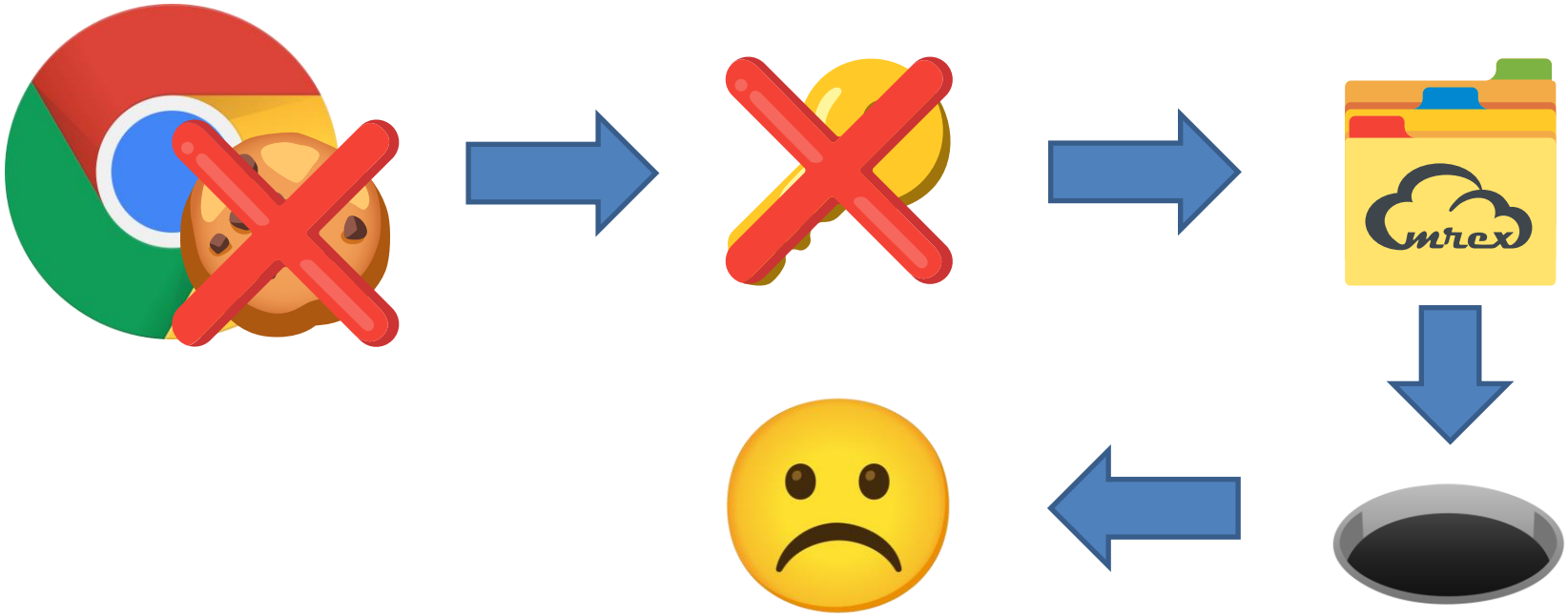
The Problem we had

Yes …

Because of that chromium got a 2-minute time window implemented, where also POST is allowed.

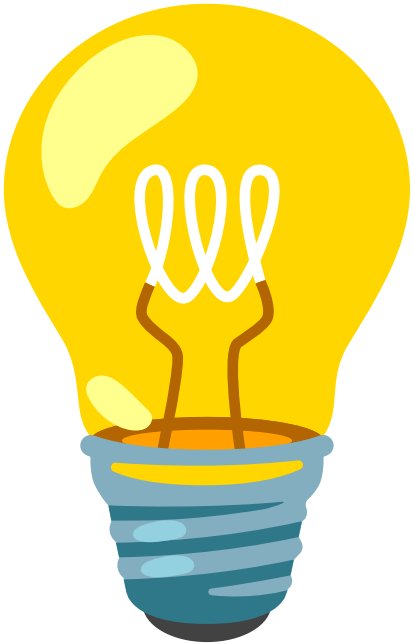
That’s why we took time to discover what is going wrong.

The Problem we had

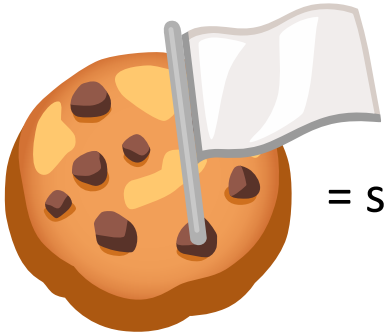


The Problem we had

Our Solution



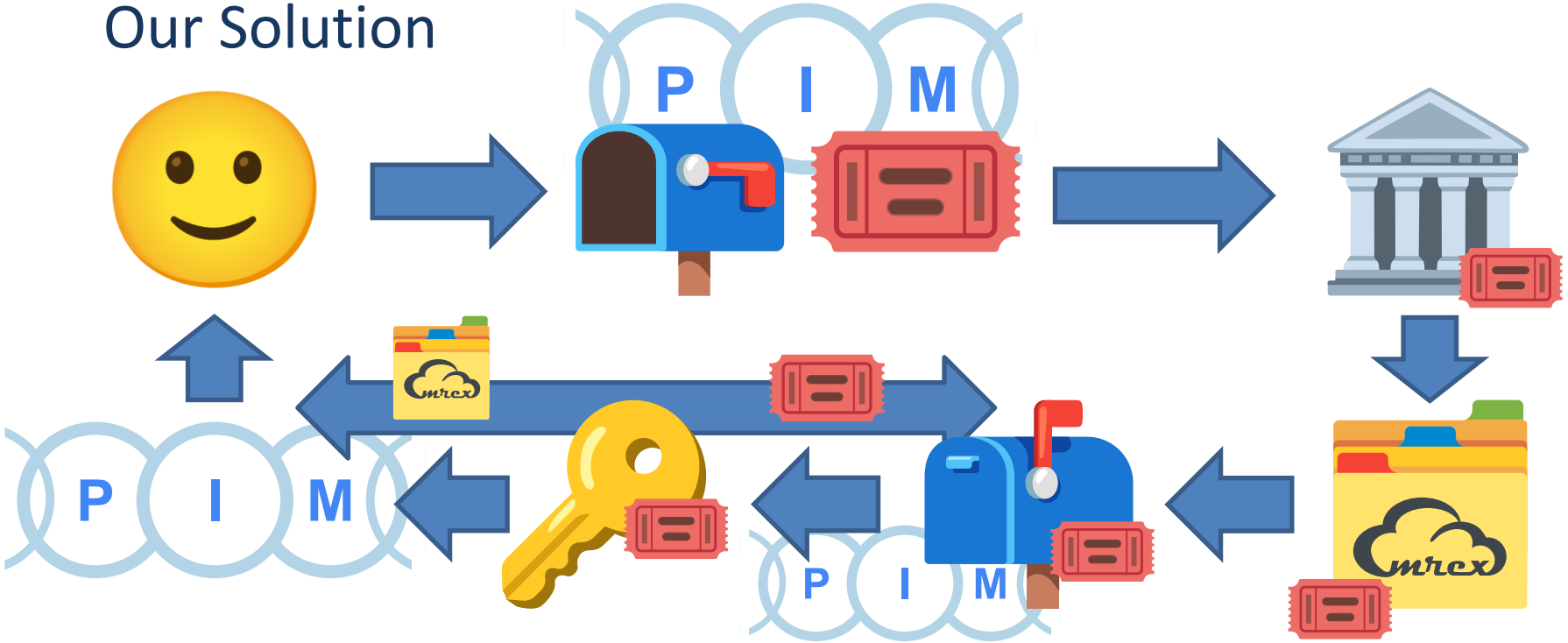
Our Solution



= set the cookie flag to NONE (insecure)

= don’t use the session to identify EMREX requests

Our Solution



Discussion

Read more

web.dev/samesite-cookies-explained/

www.chromium.org/updates/same-site

developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Set-Cookie/SameSite

developer.mozilla.org/en-US/docs/Glossary/CSRF

Further questions?

nils.gehrke@uni-goettingen.de

THANK YOU FOR LISTENING!