## Exemplar

Caching means rolling out htmlchanges (classes,etc) is slower If the cookie is large (due to many features or inefficient naming), more data has to be sent for every request and interpreting it in JS could take a long time

No flash of unstyled content or layout shift

PERFORMANCE EXPERIENCE critical rendering path performance

How much will render blocking hurt our time to first paint?

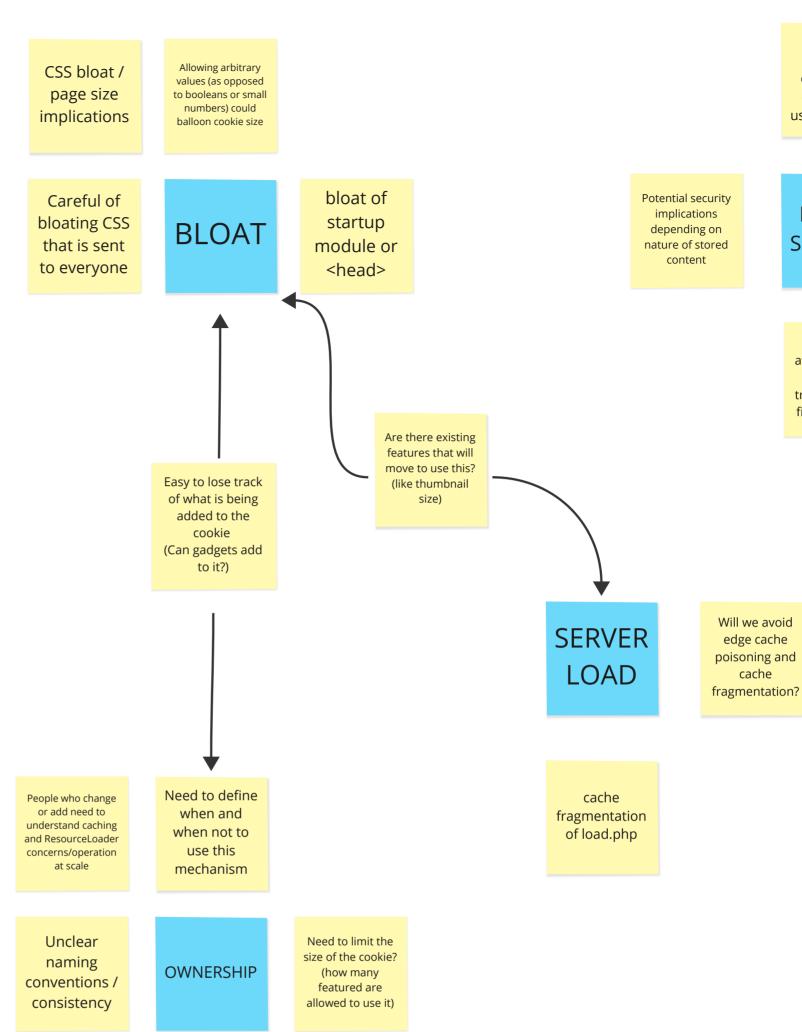
Would some preferences be expected to remain across logged-in experience?

Will this interfere with logged-in users' prefs?

Cookies can expire / be lost.
Do we need other storage mechanisms?

USER EXPERIENCE What does the UI for setting these prefs look like? Is there a single unified UI for all of them?

Different user experience on different devices may be unexpected Interaction with loggedin or temp users



Who decides

whether something can use this functionality? Additional legal

compliance

costs when

using a cookie

LEGAL /

**SECURITY** 

How do we

avoid this being

used for

tracking (option

fingerprinting)?

## **Solution requirements**

Avoids caching impact (does not fragment edge cache)

Extensible for things other than Vector

Should not break existing APIs/etc.

Avoids layout shift

Performant

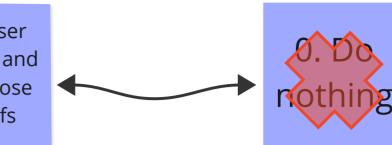
Targets logged-out users

Clear docs: what should / shouldn't this be used for

Supports booleans or limited groups of values Easy to use API for devs using this

## Possible solution approaches

force user creation and store those as prefs



allow new behaviour / Look&Feel only for logged-in

currently in prod

1. Read a
cookie & add
a class to the
HTML body





4. Read cookie & have startup module pick a different module



