Will we avoid edge cache poisoning and cache fragmentation?

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 JavaScript could take a long time.

No flash of unstyled content or layout shift.

Caching means rolling out HTML changes (classes, etc.) is slower.

Critical rendering path performance.

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

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 JavaScript could take a long time.

Easy to lose track of what is being added to the cookie (Can gadgets add to it?)

Do we need other storage mechanisms?

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

Different user experience on different devices may be unexpected.

Interaction with logged-in or temp users.

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

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

People who change or add need to understand caching and ResourceLoader concerns/operation at scale.

Need to define when and when not to use this mechanism.

Unclear naming conventions / consistency.

Who decides whether something can use this functionality?

Need to limit the size of the cookie? (how many featured are allowed to use it)

Allowing arbitrary values (as opposed to bookmars or small numbers) could balloon cookie size.

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

Additional legal compliance costs when using a cookie.

How do we avoid this being used for tracking (option fingerprinting)?

Are there existing features that will move to use this? (like thumbnail size)

Allowing arbitrary values (as opposed to booleans or small numbers) could balloon cookie size.

Careful of bloating CSS that is sent to everyone.

BLOAT

bloat of startup module or <head>

Easy to lose track of what is being added to the cookie (Can gadgets add to it?)

Are there existing features that will move to use this? (like thumbnail size)

Potential security implications depending on nature of stored content

Will we avoid edge cache poisoning and cache fragmentation?

SERVER LOAD

cache fragmentation of load.php

LEGAL / SECURITY

Potential security implications depending on nature of stored content

Will we avoid edge cache poisoning and cache fragmentation?
**Solution requirements**

- Extensible for things other than Vector
- Should not break existing APIs/etc.
- Targets logged-out users
- Performant
- Avoids layout shift
- 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**

1. Read a cookie & add a class to the HTML body
2. Read a cookie & add extra <script> tag (new code)
3. Read a cookie & add a <style> tag (new CSS)
4. Read cookie & have startup module pick a different module
5. Read cookie and set GET param
6. Server-side ESI?

Currently in prod

*force user creation and store those as prefs*
*allow new behaviour / Look&Feel only for logged-in*