Hopin Resources for Speakers*

Before your session:

- Check your network connectivity and browser compatibility
- Sign up for a Hopin demo or contact us at blinkon@chromium.org with questions

Additional resources:

- Hopin Knowledge Center
- Quick Troubleshooting Reference Guide
- Sessions Tutorial - During the Event
- Tips for Speaking in and Moderating Sessions
  - Your session is an Open Session (anyone can participate on screen)

Please note that this slide is for speakers only. Please feel free to delete this slide before sharing.

Breakout Session Instructions*

To join your session:

- Head to the calendar invite or the Sessions tab in Hopin and find the Session you'll be hosting
- Click Share Audio and Video in the center of the Session screen
  - If you haven't allowed access to your camera or mic in the event, you'll be prompted to do so. Once you see yourself on screen, you're live to your audience.

To share your screen:

- Click the Screen sharing icon at the bottom of the page
- Switch to Chrome Tab on the popup window and choose the required browser tab
- Check the Share audio box
- Click Share to start sharing

Your session will be automatically recorded.

Please note that this slide is for speakers only. Please feel free to delete this slide before sharing.

Fenced Frames
progress

Dominic Farolino and Shivani Sharma

dom@chromium.org
shivanisha@chromium.org

Google

May 18, 2022

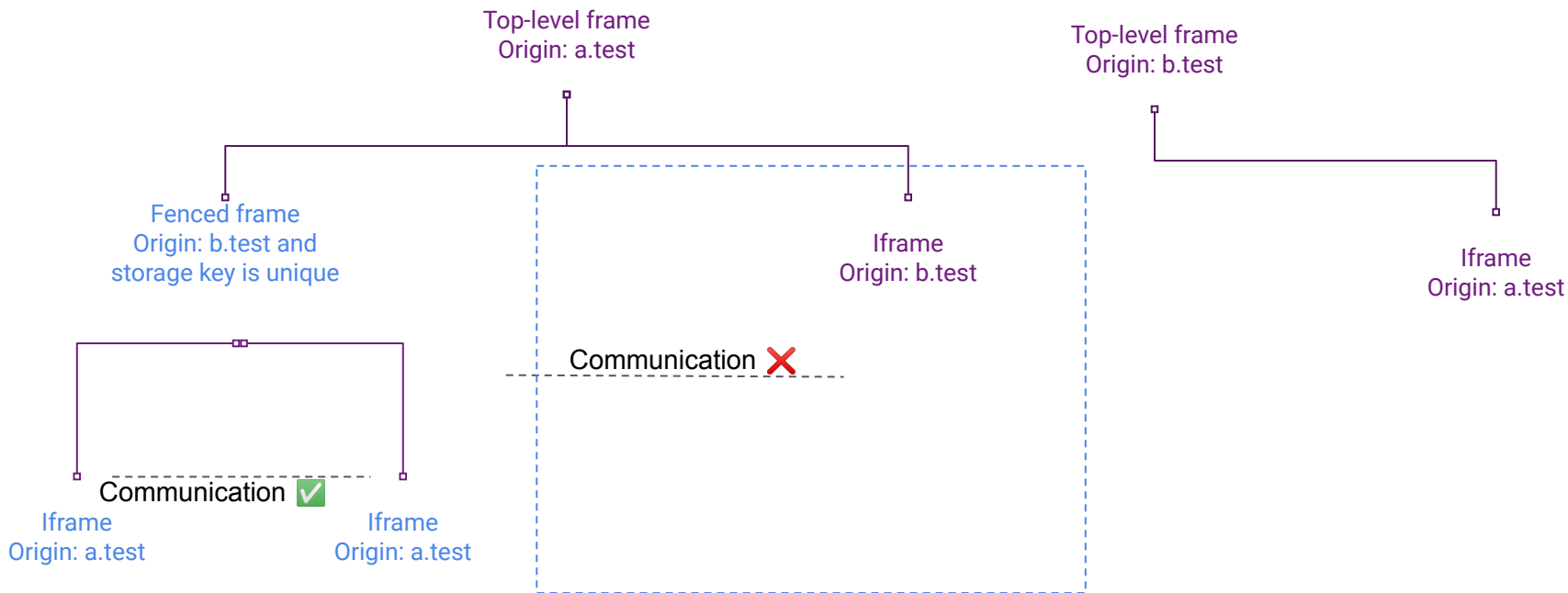https://github.com/WICG/fenced-frame/tree/master/explainer

Topics

- Origin/origin keyed storage/cookies

- Modes

- Web platform integration (a subset)

  - Security (CSP, Permissions, Sandbox)

  - Session history

  - Opt in header

  - _unfencedTop

  - Size constraints

- Specification

- Origin Trial

To recap from [BlinkOn 14](#)…

"In a web that has its cookies and storage partitioned by top-frame site, there are occasions (such as Interest group based advertising or Conversion Lift Measurements) when it would be useful to display content from different partitions in the same page. This can only be allowed if the documents that contain data from different partitions are isolated from each other such that they're visually composed on the page, but unable to communicate with each other."

Origin and origin keyed storage

Top-level frame
Origin: a.test

Top-level frame
Origin: b.test

Fenced frame
Origin: b.test and
storage key is unique

Iframe
Origin: b.test

Iframe
Origin: a.test

Communication ❌

Communication ✅

Iframe
Origin: a.test

Iframe
Origin: a.test

Communication here implies same-origin channels like broadcastChannel, service/shared workers, webLocks and accessing the same storage like local storage etc.

Modes

- https://github.com/WICG/fenced-frame/blob/master/explainer/modes.md
- In all modes, the fenced frame is isolated from the embedded context via any JS window references, script access, storage access, messaging APIs etc. The fenced frame does not know the embedding site's origin or etld+1.
- Implemented and in experimentation
  - Opaque-ads,
  - default
- Future
  - read-only,
  - unpartitioned-storage

Opaque-ads mode

```
navigator.runAdAuction(myAuctionConfig).then((auctionWinnerUrl) => {

  // auctionWinnerUrl value e.g.

  // urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6

  var adFrame = document.createElement('fencedframe');

  adFrame.mode = "opaque-ads";

  adFrame.src = auctionWinnerUrl;

});
```

Opaque-ads and default modes

- Information flow and privacy model:
  - Src: opaque to the embedding context and k-anonymous.
  - Size: limited to a set of popular ad sizes.
  - The fenced frame is allowed to create a popup (with noopener) or navigate the top-level page on user activation. (This is an ads requirement)
- Default mode
  - no unpartitioned/cross-site data available to the fenced frame, it is not able to do any cross-site data joining.

Permissions, CSP, Sandbox

- Principle: No communication channel, no privilege escalation

- No permission delegated features allowed in a fenced frame.

  - Future: Should fenced frames have static list of features always delegated - this would likely be mode based.

  - https://github.com/WICG/fenced-frame/blob/master/explainer/permission_document_policies.md

Permissions, CSP, Sandbox

- CSP:
  - No cspee (could be a communication channel)
    - Cannot create a fenced frame within an iframe with cspee
  - Csp of the top-level page applies to the fenced frame url.
  - For opaque urls only scheme source like https: apply.
  - https://github.com/WICG/fenced-frame/blob/master/explainer/interaction_with_content_security_policy.md

Permissions, CSP, Sandbox

- Sandbox:
    - Fenced frames are considered sandbox iframes with the following static set of flags:
        - allow-same-origin
        - allow-forms
        - allow-scripts
        - allow-popups
        - allow-popups-to-escape-sandbox
        - allow-top-navigation-by-user-activation
    - Design doc

Opt-in header

- Since fenced frames allow a document to have many constraints in place, an opt-in mechanism is a good way for the document to accept those restrictions.
- security: Due to privacy reasons, a fenced frame does not honor headers like frame-ancestors and x-frame-options all the way up to the primary top-level frame but only till the fenced frame root, thus opting in explicitly is important for all frames in a fenced frame tree.

```
Supports-Loading-Mode: fenced-frame
```

Session History

- Fenced frames can navigate but their history is not part of the browser back/forward list as that could be a communication channel from the fenced frame to the embedding page.

- Always have a replacement-only navigation

Navigating the outermost document

- Limited to `mode=opaque-ads`

- Introducing a new reserved frame name: "_unfencedTop"

  - Alongside "_top", "_parent"

  - window.open(*url*, ***target***);

  - <a target="***target***">

- Requires user activation (allow-top-navigation-by-user-activation)

User Activation

- Fenced!

- No propagation in either direction

Size restrictions

- Size is immutable after navigation

- Coerced to an acceptable size

- Element size isn't locked, just the internal frame view

Specification

Fenced frame

Draft Community Group Report, 3 May 2022

W3C

This version:
https://wicg.github.io/fenced-frame/

Editor:
Dominic Farolino (Google)

Participate:
GitHub WICG/fenced-frame (new issue, open issues)

Commits:
GitHub spec.bs commits

Copyright © 2022 the Contributors to the Fenced frame Specification, published by the Web Platform Incubator Community Group under the W3C Community Contributor License Agreement (CLA). A human-readable summary is available.

Abstract

The fenced frame enforces a boundary between the embedding page and the cross-site embedded document such that user data visible to the two sites is not able to be joined together.

Status of this document

This specification was published by the Web Platform Incubator Community Group. It is not a W3C Standard nor is it on the W3C Standards Track. Please note that under the W3C Community Contributor License Agreement (CLA) there is a limited opt-out and other conditions apply. Learn more about W3C Community and Business Groups.

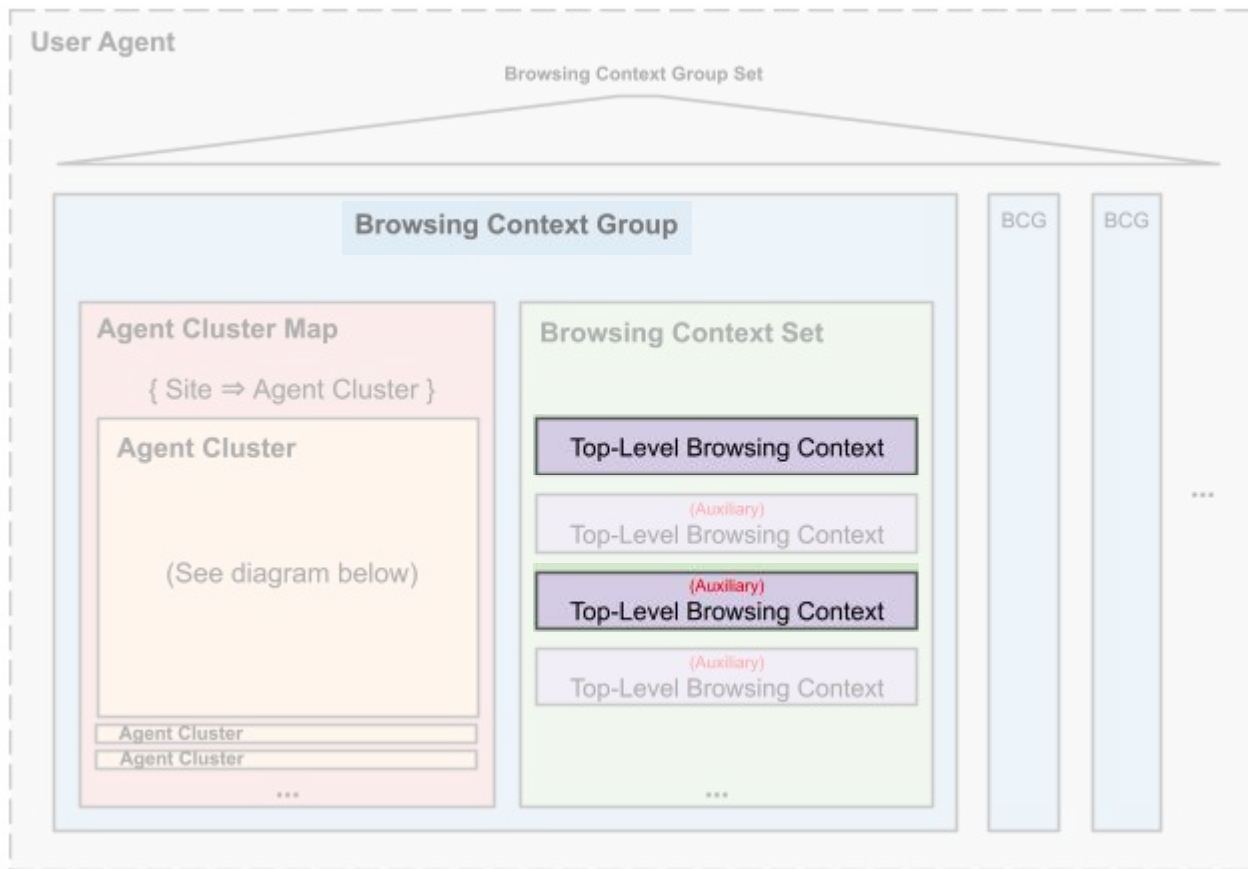§ 1. The *fencedframe* element

Categories:
Flow content.

Phrasing content.

Specification

Source: bit.ly/browsing-contexts

Specification

When an `iframe` element *element* is underlined into a document whose browsing context is non-null, the user agent must run these steps:

1. Create a new nested browsing context for *element*.

2. If *element* has a `sandbox` attribute, then parse the sandboxing directive given the attribute's value and *element*'s `iframe` sandboxing flag set.

3. Process the `iframe` attributes for *element*, with *initialInsertion* set to true.

Source: HTML Standard

Implications

- Contribute to frame tree:

  - Load events, window.parent, location.ancestorOrigins, …

- Accessible by name

- Inheritance of certain properties

https://github.com/whatwg/html/pull/6315

§ **7.3.1.2 Top-level traversables**

A **top-level traversable** is a traversable navigable with a null parent.

> **Note**
>
> *Currently, all traversable navigables are top-level traversables.* *Future proposals envision introducing non-top-level traversables.*

A user agent holds a **top-level traversable set** (a set of top-level traversables). These are typically presented to the user in the form

https://github.com/whatwg/html/pull/6315

Origin Trial: Setup

- Launched in M102

- Beta & Dev, along with Privacy Sandbox APIs

- 50/50 ShadowDOM/MPArch

- Goal: Phase out ShadowDOM by launch

Origin Trial: Metrics

- HTMLFencedFrameElement UseCounter

- Adding metrics:

  - Cannot be created (sandbox flags)

  - Cannot be created (different `mode` than parent)

  - Try to mutate `mode` attribute

  - Navigation to invalid URL

Origin Trial: Stability

- Stability Guardian Metrics

- Monitor crashes by variation group

References

[Explainer](#)

Feel free to create issues here: https://github.com/WICG/fenced-frame/issues

[Intent to experiment](#)