Fenced Frames TPAC 2023 Notes

Speaker Queue

Attendees: please sign yourself in!

1. Shivani Sharma (Google Chrome)
2. Dominic Farolino (Google Chrome)
3. Michael Kleber (Google Chrome)
4. Stephen McGruer (Google Chrome)
5. Achim Schlosser (European netID Foundation)
6. Alexandru Mihai (eyeo)
7. Chris Fredrickson (Google Chrome)
8. Alex Cone (Google Chrome)
9. Erik Taubeneck (Meta)
10. Sven May (Google Privacy Sandbox)
11. Nicola Tommasi (Google Chrome)
12. Sarah Nogueira (Criteo)
13. Fabian Höring (Criteo)
14. Lionel Basdevant (Criteo)
15. Elias Selman (Criteo)
16. David Dabbs (Epsilon)
17. Lei Zhao(China Mobile)
18. Theodore Olsauskas-Warren (Google Chrome)
19.

(Stephen McGruer: I count 24 people in the room. Michael Kleber: plus another 10ish on Zoom.)

Note taker: Dominic Farolino

Agenda

[Shivani Sharma] Presentation of Fenced Frames variant in design phase, for personalized payment buttons

Slides:
[Public]Fenced frames for personalized payment buttons @ TPAC

Notes

- First slide intro, new use cases / updates from last year: payments use case
- Currently in design and exploration stage; we'd like feedback from what we have right now
- Last year's presentation: some use cases rely on cross-site data on same "page". Third-party site may require its own user data *while embedded* within a first-party site with *its* own data. Fenced frames solve this use case by acting like a top-level browsing context
- Fenced frames are in general availability in Chrome, with a specification on WICG. Aimed at the ads use-case (links in Overview slide). Protected Audience and Shared Storage are in GA; we are designing other use cases at the moment. Reusing all of the fenced frames infra for the payments use case, but information flow in and out of the frame is what differs the most
- Visual representation of the payments use case. In the payment button, you can see personalized information for the paying user, even though they are a 3p. Big impact on the user engaging with this flow, conversion rate etc.
- Vision for payments: Today there's a cross-origin merchant iframe w/ 3pc to identify user
- Proposal: Support this use case with fenced frames. The difference in information flows resulted in different design decisions for fenced frames aimed at this use-case. We're making it generic enough to solve other similar use cases in the future.
- Vision for FF as a product: **non-advertising use case**, which is important for adoption of fenced frames. TAG and others have positive/neutral support for FF, but in the past it has been tied to more complicated/controversial use-cases. TAG et al wanted fenced frames to be non-ads specific
- Ecosystem impact: merchant site and payment providers. Public support from Shopify
- Erik Taubeneck (Meta) question: Understanding of fenced frames is limited to PA. Piece of info in the frame shouldn't leak out. Not quite payments-related, where border ID and payment info goes in. Where does FF fit into that very different information flow.
- Shivani: No information needs to go *out* of the fenced frame. Once top-level page knows about FF click, then the top-level context can spin up a new top-level context and complete the transaction
- Achim Schlosser (netID): Co-chair of something. FedCM spec contains a personalized login button component in iframe. RP/IDP pair tied to (user?) activation. Button can be personalized when a user has gone through FedCM flow on that IDP in the past.
- Achim: Relies on RP/IDP mediation, so RP has already received enough information.. Returning user gets a quicker experience. Browser is aware of the (private?) use on the same device.
- Michael Kleber (Google): Clear difference in use cases. Shivani's example works even if the case that the user doesn't want to have relationship with surrounding page. They'd be happy to *not* give x-site ID to credit-card-provider page.
- <Presentation>: Shopify engagement metrics with personalized buttons
- <Information flow slide>: The change is that the payment provider decides what information from the user needs to be reused in the future *inside* the personalized

payment button. That gets stored in the browser that future buttons will pull the data from. We'll discuss specific storage mechanism later. This is all usable via window.sharedStorage.set(...)

- Same as today: Embedding page creates a personalized button with the fenced frame constructor. Fenced frame config is returned from a consumer API (PA or Shared Storage's select URL), **OR** the new constructor which this use-case introduces, where web script can create a new config with a non-opaque, arbitrary URL. Information flows inside the fenced frame but it has no unpartitioned data (at this point it does not have the personalized "last 4 digits" data from the user's credit card etc.).
- To get access to the data, it'll need to enter stage 2. The frame will need to be locked down such that it cannot exfiltrate any data. Do this via window.fence.disableUntrustedNetwork(). This lets the fenced frame contact shared storage with a known key, and retrieve the personalized information. In stage 2, fenced frame can access data from embedding site *and* its unpartitioned personalized data. No channels exist to exfiltrate data back to embedder.
- David Dabbs (Epsilon): It disables "untrusted" network, but is there a "trusted" network? Why not disable *all* network.
- Shivani: we still allow Private Aggregation reports which does not contain any joint data which is considered "untrusted".
- David: The egress of data from shared storage; it wasn't just reaching into the storage jar, instead it was getting an *opaque* URL. What are the privacy implications of this new mode *not* using opaque URL.
- Shivani: (Describing current shared storage opaque URL flow). In the new mode, the fenced frame is in a locked down mode where it can't exfiltrate any joint or unpartitioned data.
- Fabian Höring (Criteo): What happens when user goes through the whole flow ? How does the embedder page know if the user clicked ?
- Shivani: Outside of the fenced frame there is a embedder script listening to the click of the fenced frame payment button. The script is now responsible for opening the payment handler context or popup, which has access to the user's information independent of the FF. They complete the transaction. This happens today as well; even the iframe with the personalized [...]. Does that help?
- Fabian Höring: Yes.
- Shivani: Right now, the click listener must have an overlay on the FF, to capture the overlay click. This leaks click coordinates to the embedder, which we'd like to mitigate this side channel by communicating only the fact that there was a click
- Fabian Höring: Click is not enough, if the payment fails.
- Stephen McGruer (Google Chrome): Top-level frame (fenced frame embedder) communicates with the payment provider directly, not related to the fenced frame at all. Purpose of FF is to simply not leak the user's data (last 4 digits of credit card #). Could have it where FF triggers something *itself* on click handling, but what happens next is payment-provider-dependent. We don't want to bake all of that per-provider complexity into the FF. So the top-level is responsible for taking.

- Someone: Fair to say there're maybe certain flaws where the FF shows digits but then opens an iframe.
- SM: Could open an iframe but it'd be useless. No unpartitioned storage.
- Fabian Höring: Use case where you enter credit card number in a fenced frame?
- SM: No.
- <More technical design slides>: Privacy-safe == not exfiltrating from SS
- Why Shared Storage? For API simplicity. It already has "get()". Shared storage is already unpartitioned by definition, it's a good fit (no need for many partitions, just unpartitioned is what we want). No. Discussing trade-offs with other storage mechanisms (local storage, cookies, etc.)
- David Dabbs (Epsilon): Do you expect this to land prior to final 3PCD? Are all of the outgoing requests uncredentialed.
- Shivani: Working to figure out the final timelines, will update the repository via explainer.
- Achim: What happens when FF renders but user is not signed into the login provider. No shared storage, what's the downgrade scenario?
- Shivani: Provider can choose if there is no local data, they can choose an unpersonalized button. Fallback is per-provider.
- Shivani <User experience> slide: Disabled setting is introduced for users to disable functionality. We're talking through this in detail currently; many details need to worked out — how generic to make it (for other use case). Pros and cons of specific vs generic interface, but we're still thinking through the design
- <Next steps>: Continue designing & engaging with community (issues)
- Lionel Basdevant (Criteo): Wondering about FFs in the context of advertising… FF complex stuff… Fenced frame frames lose network access but gain shared storage access.
- Shivani: In context of Protected Audience. Nothing changes — still have network access until further notice unless the FF decides to invoke disableUntrustedNetwork.
- Michael Klebber: We should spend some time thinking about how an ad chosen via PA might make use of this fenced frame functionality. Initial thinking: ad would divide itself into two pieces. One piece still has ability to use functionality like event reporting… a different part of the creating inside nested fenced could have ability to change its appearance. Both of those things are important, and they can't yet mix together because fenced frames have a different mode for those behaviors. I can't figure out how this'd work in ads use case RIGHT NOW, but let's explore together in future.
- Lionel: We're talking about a change in the state of the fenced frame after accessing shared storage. Same for nested fenced frames?
- Shivani: Yes but have to think through the privacy model there to see if there're any differences, but overall answer is yes.
- David Dabbs: Shivani mentioned potential UX enhancement (afford user control separately from basic shared storage). In today's Privacy sandbox UX, there're 3 toggles: Topics, "Ad measurement" "Spam & fraud reduction". SS falls under ad measurement. Would we be adding extra control *there* for users? (If you turn "Ad measurement" off, that'd turn off shared storage).

- Shivani: UX studies are on the roadmap to figure out where is best to put additional controls with this mode.
- David Dabbs: Content of frame requests to be locked down? (Right). If it still has in-flight network requests, what happens to those? Are they summarily canceled?
- Shivani: Any pending network request would not be dependent on the shared storage data, so letting it request complete shouldn't be bad, as long as there's no redirect that can carry new data (which likely doesn’t happen anyways).
- Michael Klebber: Canceling request could signal side-channel, so maybe we shouldn't let them all go.
- Erik Taubeneck: Is this concern limited to network requests? What about async fns that a FF triggers in its embedder? Answer: not allowed in the first place.

Speaker Queue

- Name / Affiliation
-
-
-