

(public) Notifications - Sharing current problems & issues - TPAC 2023 Breakout

Minutes notes for [13 September 2023 | Notifications - Sharing current problems & issues | Calendar | W3C](#)

Status: Meeting complete, public edit access revoked.

Presenters

Penelope McLachlan

Dan Murphy

Present:

- Penelope McLachlan (Google Chrome)
- Dan Appelquist (TAG)
- Sangwhan Moon (TAG)
- Dan Murphy (Google Chrome)
- Kagami Rosylight (Mozilla)
- Erik Anderson (Microsoft Edge)
- Howard Wolosky (Microsoft Edge)
- Marian Harbach (Google Chrome)
- Vincent Scheib (Google Chrome)
- Austin Sullivan (Google Chrome)
- Lucas Haraped (Google Chrome)
- Alex Christensen (Apple WebKit)
- Martin Thomson (Mozilla)
- Masakazu Kitahara
- Toshiaki Koike
- Reilly Grant (Google Chrome)

Goal

Share issues with current notification spec & ecosystem, possibly connect to problem solve in the future.

Scribes

- Dan Murphy

Slides

📄 (public) Notifications discussion - TPAC 2023

Notes

Notification problems & possible mitigations

- Penny:
 - Spam & abuse problem
 - Users have learned to not trust notifications
 - Poisoning the ecosystem
 - Considering rate limiting & other countermeasures
 - What opportunities are there out there for standards revisions that would help with this
 - Trivial for a site to do origin rotation to avoid countermeasures
- Erik:
 - All sounds true, would love to hear more about interventions
- Penny:
 - Training basic models that run on the device that predicts whether user is ever likely to accept a notification request
 - Impacts visual priority of the request. Strong signal might overlay web content area. Low signal might show a chip over the lock icon in the omnibox Less obtrusive
- Dan Appelquist: Does the engagement with the site feed into this? One of the things I've seen early on is that the website asks right away for push notifications, which the site thought was good, but the user didn't want to see
 - Penny: YEs, it's a hybrid model. There is crowdsourced data, based on data. But that is limited to users that have opted in & have safebrowsing. As we share origin signals back to train the origin model on the server side
 - If they have opted out, then they need a offline-only user-specific model trained as the user uses
 - Those are prompt spam mitigations, then there is message spam too. Hard to mitigate message spam, as it might require sampling message content. Due to encryption.
- Marian: We have written a paper about message spam, and will be publishing it soon
- Erik: Are you able to observe the origin of the messages coming through

- Penny: We can take an intervention for a specific origin. When we do metrics about origins sending, there are power-law effects, but clear that there are some actors that send massive amounts.
 - The site that the user signed up might not be the origin the messages are coming from, not always clear. But certainly there are origins that look pretty shady.
 - We are thinking about how we can do rate limiting for senders that are sending > 10-20 qps.
 - But, that brings us back to origin rotation as a workaround
 - Maybe for higher volumes, we require registration scheme.
- Martin: That was the original idea for VAPID scheme, the idea was that if you are operating a service & seeing high volume from a particular endpoint, you can contact them & set up more infra.
 - But difficult, as you don't see the origin at the push service, which is challenging
 - The browsers can do something about this, but IDK if we have the feedback channel for the push surface
 - Penny: Are there interventions you have considered?
 - Martin: We have thought about not showing notifications once the origin abuses that. But that still has drained the battery. But we haven't gotten that far.
- Penny: Energy drain on notifications / serviceworker is another subject. Apple has put together an explainer for exactly this. Completely aligned on the problems that it seeks to solve - real problems.
- Martin: The privacy aspect "is this site bothering you" is one we are more able to deal with on the client. The energy drain issue we should talk about. There are two ways to think about this
 - We expect the push service to do something for us
 - Or maybe the client should be doing something for us - after battery drain occurs, it should tell the push service.
 - The push service sees end points for the senders, so the client can turn off each one of them individually
 - Penny: Yes, the client is in a good position there. Ideas for possible future
 - If we see no engagement with the user with notification or the origin sending notification, likely we can intervene & cancel that subscription
 - But that might not be what they want. Example: Sports scores. The'll never interact with notification or origin, but they want them. Hard to account for corner cases
 - Vince: Idea - you could check-in with the user, asking if the user finds these notifications useful.
 - Apple does this for some high-priority notifications
 - Martin: We do kind of this, but there isn't UX that is loud, user has to go to settings
 - Penny: We also want to make it easier from the message itself for the user to unsubscribe.
 - Dan Appelquist: Doesn't every user agent have a different...

- This solution doesn't tell the website, doesn't give feedback that the user has unsubscribed, and they should have less spammy notifications
 - Product managers at that company would find this info useful to argue to have less notifications, for good actors
 - Reilly: Maybe the answer is an aggregated feedback signal to prevent abuse (per-user is too granular, allows bad actors to force users to re-add notifications)
 - Given we have an agreement with the push service anyways, maybe an aggregation data endpoint to report to the service would be helpful
 - Penny: We do something like this for permissions - permission report, launched, made a splash, fizzled out.
 - Having that public data, at least for sites to benchmark & compare with each other, could be persuasive
- Erik: There is no way for the user to see what subscriptions are currently stored... then the user sees random bad notifications. Hard to view.
 - Vince: it depends on platform - on Android you can go to settings, notifications, and go to chrome you can see all of the notifications. On iOS, there is a similar thing w/ notification center.
 - Erik: that is after they have shown notifications? Or enabled? Maybe this is push permission, which is visible.
 - The goal is to find all sites that can show notifications
 - In FF, there is a notification settings, all sites that have ever asked for permission. And turn them all off
 - Reilly: There is a feature on Android for notification history. Some kind of history UI can be helpful for users to understand what sites have sent before, recognize what site & have a bit of accountability.
- Dan Appelquist: It's painful to me when people always disable notifications due to bad behavior, and there are legitimate use cases & good players.... There is more to be done on the platform to ensure notifications are used responsibly. Somehow we need to think more about this.
 - Martin: it's a great feature & we want to make it better
 - Penny: Ray of hope - the top abusers are sending so much volume, if we can somehow make it too costly for them or block them, then we can reduce global volume by maybe 25%. They will take countermeasures to what we take, but the potential for improvements in a short period of time is there.

"important" notifications (VOIP)

- MSFT already working on this:
- [Intent to Prototype on blink-dev](#), [explainer](#), [initial implementation](#)
- Example hero use-case - empower Skype on the Web.

- On Android this would require a permission (FOREGROUND_SERVICE_PHONE_CALL) that we probably don't want to issue to Chrome itself. Limiting this functionality to installed experiences seems good in either case, as otherwise there would be complicated incentives.

Notification sounds (ringing?)

- Related to important notifications, some apps might want a ringing sound
- Somewhat limited by OS support of custom notification sounds, however, may be possible to create a "Safe" set of standard notification sounds for developer to choose from e.g. "Chime" "Ringing" "Alarm", and perhaps others
- This was part of a Microsoft proposal that we provided feedback on [Explainer](#)
 - Long story short, a (admittedly brief) joint investigation showed that very few sounds would actually be consistent across operating systems, which is hard to explain to developers. It's important that these are OS-consistent: ideally an e-mail in Apple Mail and one in Gmail alert Apple users in exactly the same way.
 - For Chromium to support arbitrary sounds, the media team advised that the best way to achieve that would be to load a (background) tab with a element in which we play the developer-provided URL. That's a significant system health cost.
 - On Android such sounds are tied to the notification channel as opposed to individual notifications, and actually cannot be set for individual notifications anymore.
- Penny: Media team has said we might need a background tab
- Matt: This is also needed for the ringing case, different experience. How do we do "answer" and "hang up". Do we want a dedicated ringing API
 - Erik: There are things MSFT is proposing, different
 - Dan Appelquist: Both Android & iOS have dedicated system level APIs for integrating with the call system. Seems like this would need to plug into that - seems like it needs that, otherwise it would be a very confusing user experience. Desktop OS is a different store
 - Penny: Two - for the notification system, we should plug into that, but for the web developer, it could just be part of the notification API.
- Marian: If it is a specific use-case for calling, then it might be an opportunity to present better permissions UX to the user - "do you want this app to do phone calls"
 - Penny: Could we use two different permissions for this, notifications & phone calls?
- Matt: Do we have use-cases of important notifications that aren't incoming phone calls? Are there two proposals here?
 - For phone call, that seems it is beyond notification, it's an ongoing connection for the API for the phone call, not just the first part.

- Maybe the phone calling business is a whole separate thing.
- Penny:
 - Use-cases that aren't call: Alarm clock, special access for loved ones to send emergency messages. Maybe more
- Erik: Main concern is about rate of abuse. Phone calls are easier message to a user
- Martin: We might also not give sites the ability to ask, just say to user "this is a ringing notification, would you like to see it as a ringing notification?"
- Penny: interestingly, the Apple team has gated notifications behind install. Probably part of that is mitigating abuse, but concern now is that we have moved problem to installing things that users don't want to install
 - Dan Appelquist: I asked this from Apple, I find this incredibly interesting. What is the future of web apps, and what role does the install step play in that?
 - Penny: There's been a lot of research here w.r.t. User mental model & installed web apps
 - Users do form different expectations from installed apps vs things in tabs
 - Those things are technically untrue, sometimes wildly wrong
 - They believe an installed app has more power, and is a lot safer, than sites in a tab
 - They derive trust from stores quite a lot.
 - Marian: How do you come to trust a piece of software? Trust in the source came out quite high - "how am I getting it"?
 - Installation is not just about the install ceremony, it's more about 'how did I get here'. Where do they go to find the app, where did they install it from. And then what does it look like after installation, where does it sit on the device, how is it integrated?
- Permission element talk later - goal of that is to move from developer push to user pull, and that might reduce some of the concerns around prompt spam.
 - Dev puts black box on page, that browser controls, for sensitive stuff like requesting notification permission.

Notification channels.

- Allow sites to register and send to specific channels, similar to capabilities offered on Android and iOS platforms
 - Can be a useful way for users to reduce unwanted notification volume, for example on a messaging app like IG the user could choose to allow DMs but not receive notifications about new posts in the feed
- We might be able to mimic support in Windows through grouping.
- Nuances to consider on Android: channels are declared once and are then permanent, i.e. updating the importance, sound, vibration etc. of a channel should not happen after it's been created, even when the channel's name changes.

- Erik: I don't know how it works on Android - on iOS - if the user changes their choice, does that get given abc to the site?
- Martin: There are different things to consider
 - Time-sensitive notifications (we have affordances for that)
 - This is about the ability to sort the notifications (moving notifications settings in app to client)
 - Question: How much of this needs to move into the declarative stuff, because this could be a useful thing to have if you are presenting the notifications through that medium as well
 - Declarative doesn't hit SW ([new proposal](#)). Anything we do in this form needs to be able to flow through that. Except perhaps the ringing thing
 - Alex Christensen: Not necessarily - you could start ringing, but only start SW if the user answers
 - Martin: The primary reason for declarative is to reduce energy use - if you're turning the screen on & making noise, then you may have well wake up the SW.
 - Erik: There could also be considerations to not ring on 6 devices at the same time. That might need service worker logic.
 - Reilly: There is also latency concern too for declarative proposal, so
 - Penny: I think SW latency is 200ms, so it's not a ton
 - Alex: If I receive a notification that rings the phone, and I hang up, then no need to start the SW.
 - Martin: My concern is that there is a bunch more complexity around sounds & other presentation things, solving that in the declarative sense might be more difficult. Not necessarily a bad idea, just harder.

Reducing notification energy use (currently need to spin up JS)

- The current web notifications spec was designed with maximum flexibility as the objective. With the current spec, notifications require a Service Worker and when a notification is received, the ServiceWorker wakes and can then take some action.
- The solution is an amendment to the notification specification that optimizes for the most common notification usage pattern of simply displaying a notification on the user's device. This change will improve performance and energy usage of notification delivery.
- There has been some resistance from partners to support this as they need the SW / analytics logic to run for monetization.
- Android would be able to support high priority notifications in a declarative path. This is important because the framework deprioritises notifications for an entire app (i.e. the browser) based on show and interaction rates, which means that currently a single misbehaving site can negatively impact the holistic browser notification experience.
- Big +1 to doing this if we can get broader ecosystem adoption.

- Penny: If partners don't adopt, then things are dead in water (declarative)
 - Perhaps we can tie these features to requiring the new declarative method to carrot developers to the new model
 - Erik: Reason I asked about how channels worked on native platforms - if we are going to improve performance, it would be nice if the service knew that the user disabled that channel.... Infinite complexity, I could have an endpoint on web site that registered for the push channel to 'disable' a channel for a client, so the push service can just not send those notifications
 - Dan Appelquist: Concerned that adding more analytics can make it more attractive as an advertiser platform / possible abuse
 - Penny: Current model, most things can be tracked because SW is woken up, and it can report whatever it wants back to the origin. So what is the minimal thing that declarative needs to support for 3rd parties to adopt it. A huge portion of adoption goes through 3rd parties, as people don't want to adopt the own implementation
 - Thirdparty = push vendors. Integrated into customer management systems. E.g. telling customer their order has shipped.
- Martin: There are ways for the push vendor to tell the service "this is a low priority thing". It would be interesting to see whether that signal matches what is encrypted to the browser. That might give us an opportunity to provide feedback. E.g. sender says 'high priority', but ends up being a marketing message, perhaps we could "rate" each channel
 - Penny: How do we incentivize good behavior & disincentivize abuse.
 - Martin: We avoid knowing which sites the user is visiting. So hard to collect "this origin sent notification" - so we would need some sort of privacy-preserving mechanism
 - We are building some of those already, but maybe not a big deal. But raises bar for price & complexity
- Martin: Question - what are people's thoughts about how to proceed? There are a lot. Other things I have on the list, e.g. cryptography
 - Penny: We / Chrome does not have a roadmap here. I am interested in this space & want to see improvements. I'll have to build advocacy & willpower to take on this effort
 - But much easier to do this with spec & agreed path forward with alignment. Which is why we made this session
 - Should we meet more regularly to get spec alignment?
 - Martin: Yes
 - Dan Appelquist: Perhaps this can be a task group - charter...
 - Martin: IDK if we turned this into a recommendation
 - Dan Appelquist: From a vendor perspective....
 - Martin: Reservation - declarative stuff involves changing to messaging format & protocols, and we did that in IETF. Revisions to RFCs.

- Dan: Mostly interesting in talking about the venue to continue this discussion, and how TAG can plug in.
- Vince: We have push API in webapps working group. That group has the potential for more activity than we have seen. Meeting more, etc. If there is consensus by multiple browser vendors to commit work to it, then we need to actually start working more regularly.
- Dan: It sounds like Apple is more interested in this, and I see Mozilla here, interested
 - Talk to webapps chairs to set up separate call / work stream.
- Vince: there was a call for more meeting, and I think we should.