AOL Sub-delegation

Requirements for AOL Project Template:

- template is a “batteries included” example that includes best practices examples of how to build an app.

JeffG

General end-user mental model issues:

- delete doesn’t delete
- other users can copy on read
- revoke just cuts user off to future changes

JeffG

Languages - app-oriented?

App-sharing as a ty pe of sharing, other types?

JeffG

MFA / added security?

I may want to specify that for certain permission within an app I want multiple auth factors to be triggered in order to grant it

Ryan Betts

Goals:

1. creating a spec that includes all fields necessary of info to provide to the user when they are
2. create a working experience shipped in the project template that provides developers with best practices for user messaging

JeffG

Fields

- who you’re sharing it to
- what you’re sharing (path(s))
- how long?

JeffG

Modes:

1. Access to a new device
2. access to another app
3. access to another person
4. (some combination of all three depending on context)

JeffG

Prior art:

- browser permissions (eg extension installation, Geolocation)
- iOS
  - granularity: ability to respond w/ more than just yes or no (eg. “just this once”)
  - composition: permissions are asked for 1 by 1.

JeffG

Permissions

- App manages it’s own data
- App accesses $x data from another app

JeffG

Request history

All requests for permissions are done async.
+ How do they get grouped over time to be responded to
+ how do they clobber each other
+ can I review all requests from all time and how does that look?

JeffG

Demo app notes:

- uploader app uploads to private filesystem
- publisher (ig clone) asks for permissions to read the uploaded photos, can publish them publicly
- when people view the photos from the public side, the publisher app tracks views as private side data
- the uploader app can request and represent the views data from the publisher app in it’s own UI
- The ***MVP*** for this use case is single-player - app-to-app linking.

JeffG

Biggest questions a user might have when sub-delegating:

- **scope** of the delegation: what permissions am I granting?
- **identity** of the thing I'm delegating to: who exactly am I granting them to
- **time** horizon of the delegation: how long are these permissions going to last?
- **permenance** of the decision: how final is this decision? (eg. process for revocation)

Ryan Betts

Composition

Balancing between asking for all the permissions at once upfront, versus one at a time as needed.

Both approaches should be supported with guidance on when it is appropriate to use one versus the other.

Ryan Betts

Granularity of response

Is there an ability to offer a response more granular than “yes or no”? eg. “just this once” in iOS, except that session based scope isn’t relevant in WNFS.

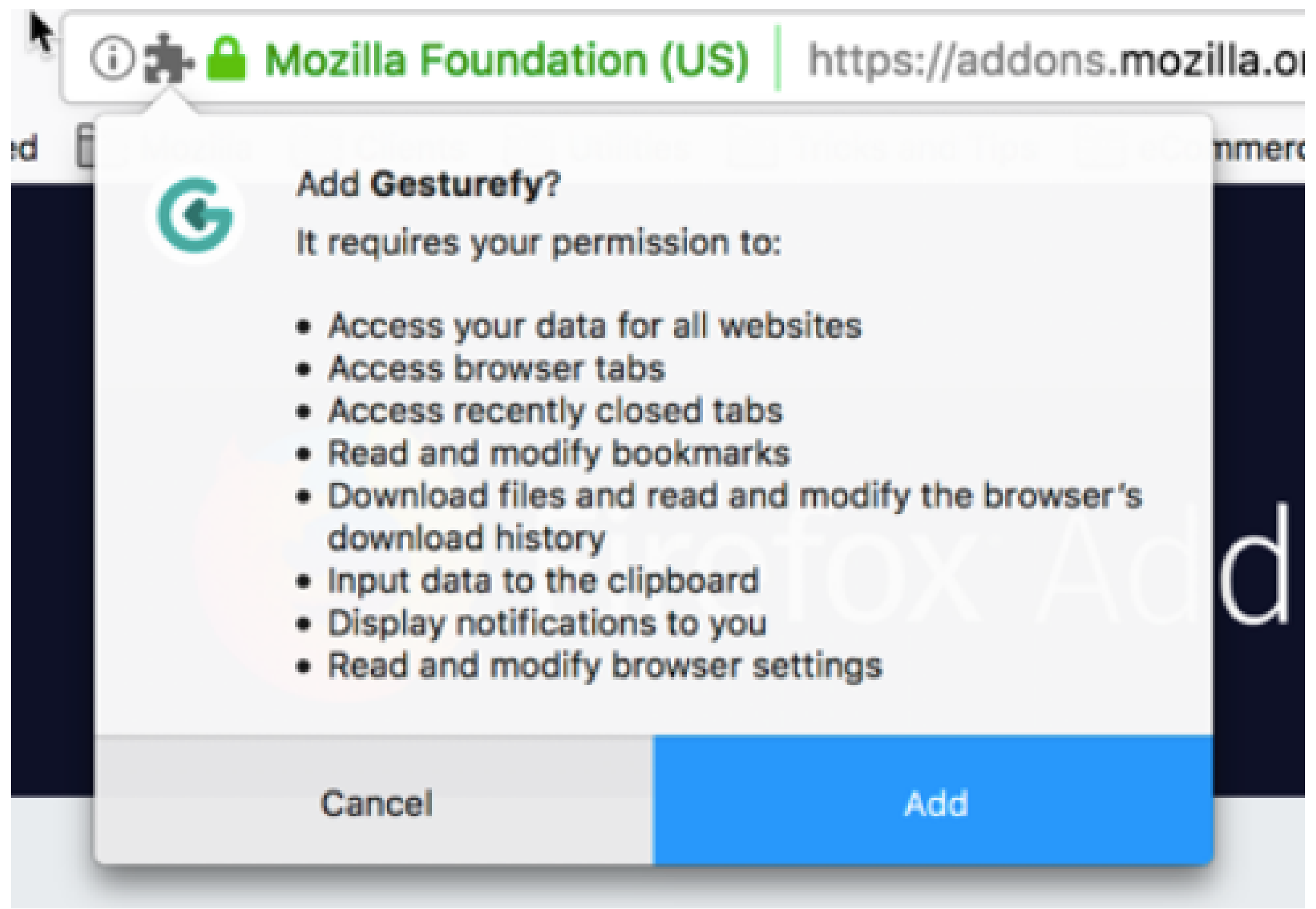
Ryan Betts

Role of trust

+ Two apps talking to each other might develop a “permission shorthand” so that the experience is smoother and more concise: “Do you want Apple Music to import your Spotify Playlists?” ← could happen, lol. The Granting App could be willing to present a specific Requesting App’s permission request differently.
+ How is requestor identity verified? Both the app context and the user issuing the thing: when there is no notion of official username?
+

Ryan Betts

Firefox installation permissions prompt



Add **Gesturefy**?

It requires your permission to:

- Access your data for all websites
- Access browser tabs
- Access recently closed tabs
- Read and modify bookmarks
- Download files and read and modify the browser’s download history
- Input data to the clipboard
- Display notifications to you
- Read and modify browser settings