

Permission bundling

A way to combat permission fatigue?

The Permission Prompt Problem

This page would like to:

- Do popups
- Play sounds at random times
- Capture your microphone
- Capture your camera
- Know your position (for emergency services location purposes)
- Gather network information
- Avoid being suspended when it's in the background
- A dozen other permissions you don't even want to think about

It's a phone. How many clicks did you make, and how many did you understand?

Maximal Bundle Version

“I’m a (video) phone app. Can you allow me to do phone-like things?”

The Consent Model - the Parties

- The user (Good by definition)
 - Knows what he wants done, and what he's willing to risk.
 - Doesn't know how things work
- The platform (Good because all is lost if it isn't)
 - Knows how things work (mostly)
 - Has history with the user's actions in the past
- The application (Good, Bad or ugly)
 - Knows what it wants to do
 - Knows how things work for the specific cases it wants to do
 - Does not have an implicit trust relationship with the user
 - May have an explicit trust relationship (or may try to sneak in)

Ways to Mediate

- Standardize a number of roles, with associated capabilities - let app ask
 - Pro: Makes for understandable role set
 - Con: Extra caps needed for “extra value features” don’t fit with the role caps
- Let the app ask for a bunch of permission, browser guesses what user wants
 - Pro: A trusted party chooses what is granted
 - Con: The user has limited understanding of what the browser decides to permit
 - Con: Apps that don’t fit browser’s model may be badly served (autoplay!)
 - Con: Implies “asking up front”, which has been declared an antipattern
- App asks for permissions, user grades trust in page, browser implies perms
 - Pro: Simple for user
 - Con: Over-shares permissions (since trust is one-dimensional, perms are many)
 - Inspection of resulting behavior is a key feature in building trust (but may be too late)

Solutions?