Native messaging for strictly confined browsers

Olivier Tilloy – Ubuntu desktop team

CANONICAL ubuntu

Benefits of firefox packaged as a snap

- Strictly confined: added security layer
- Endorsed and published by Mozilla
- Timely updates
- Runs on old releases (16.04)

Shortcomings of firefox packaged as a snap

- Browsers interact in many different and creative ways with the system
- Native messaging was broken

What is native messaging anyway?

- ~~Integration with your favourite instant messaging application~~
- ~~Sending/receiving SMS~~
- IPC between browser extensions and native applications
- Opens up a whole lot of possibilities
- Not necessarily great from a security perspective
- Use cases:
  - GNOME Shell extensions management
  - Password managers (KeepassXC, 1Password, …)

Problem statement

- A strictly confined browser cannot enumerate and spawn arbitrary executables on the host system
- How do we address that?

XDG WebExtensions desktop portal

- WebExtensions portal proposed and implemented by James Henstridge
- D-Bus API for browsers to request launching a native host connector
- The portal
    - Accepts an app name and a unique extension ID as parameters
    - Looks up the corresponding manifest
    - Checks the extension is allowed to run the application
    - Prompts the user
    - Spawns the native connector
    - Passes open FDs to the browser

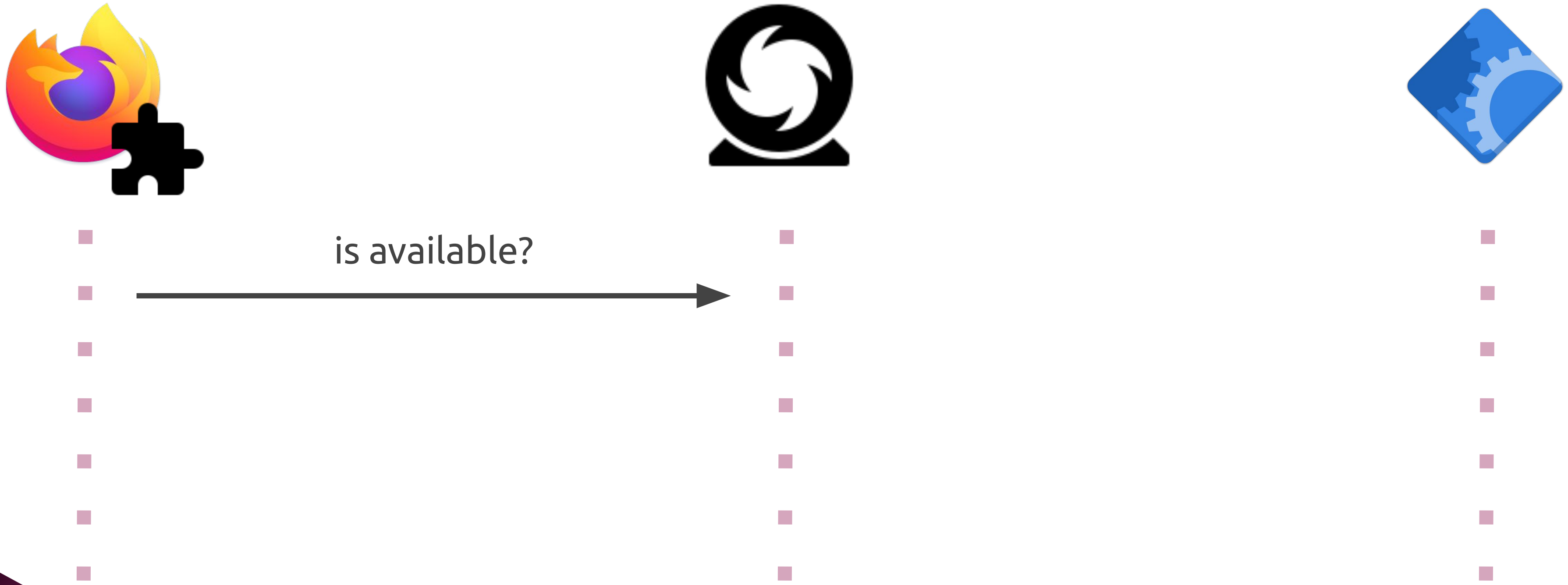
Advantages of the portal

- Not specific to one given browser
- Works across confinement technologies (snap, flatpak, …)
- Available in Ubuntu since 22.04

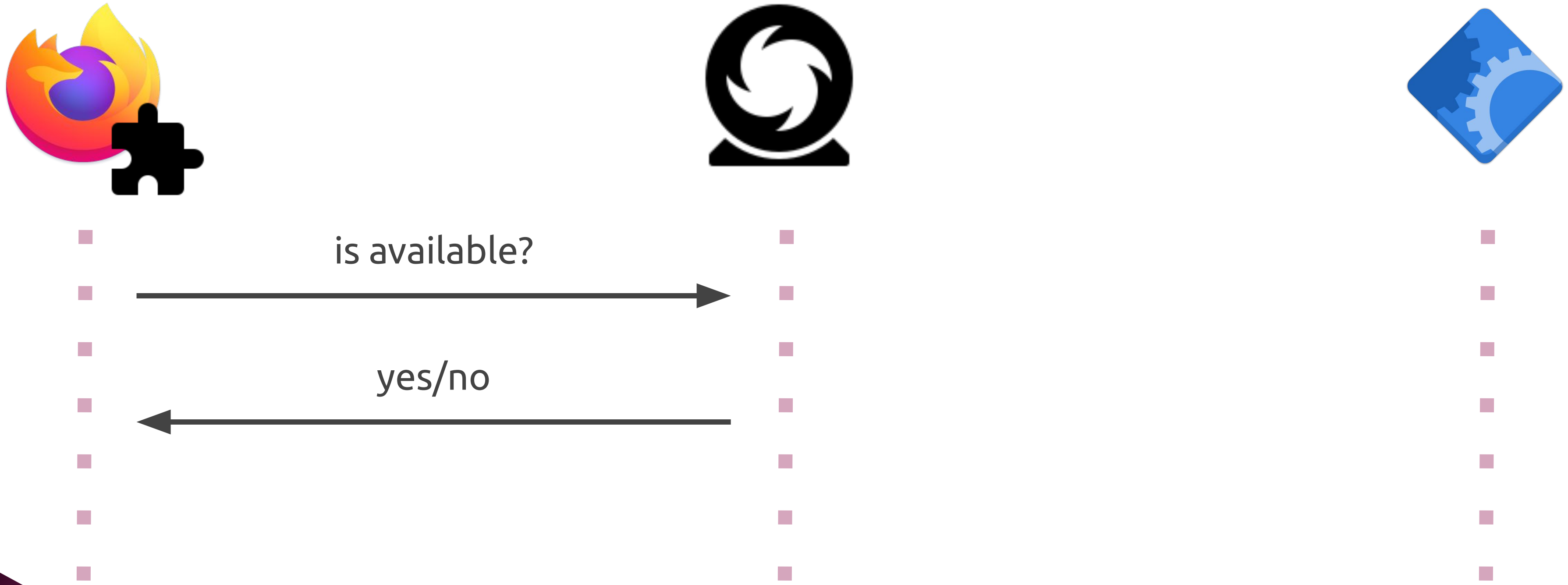
Making firefox use the portal

- Two code paths:
  - Legacy (unconfined)
  - Portal (confined, where the portal is available)
- Worked closely with Mozilla engineers
- Coding style and quality standards (tests)
- Pretty large patch (~1800 lines diff)
- Not merged upstream yet
- Cherry-picked in the stable snap last week

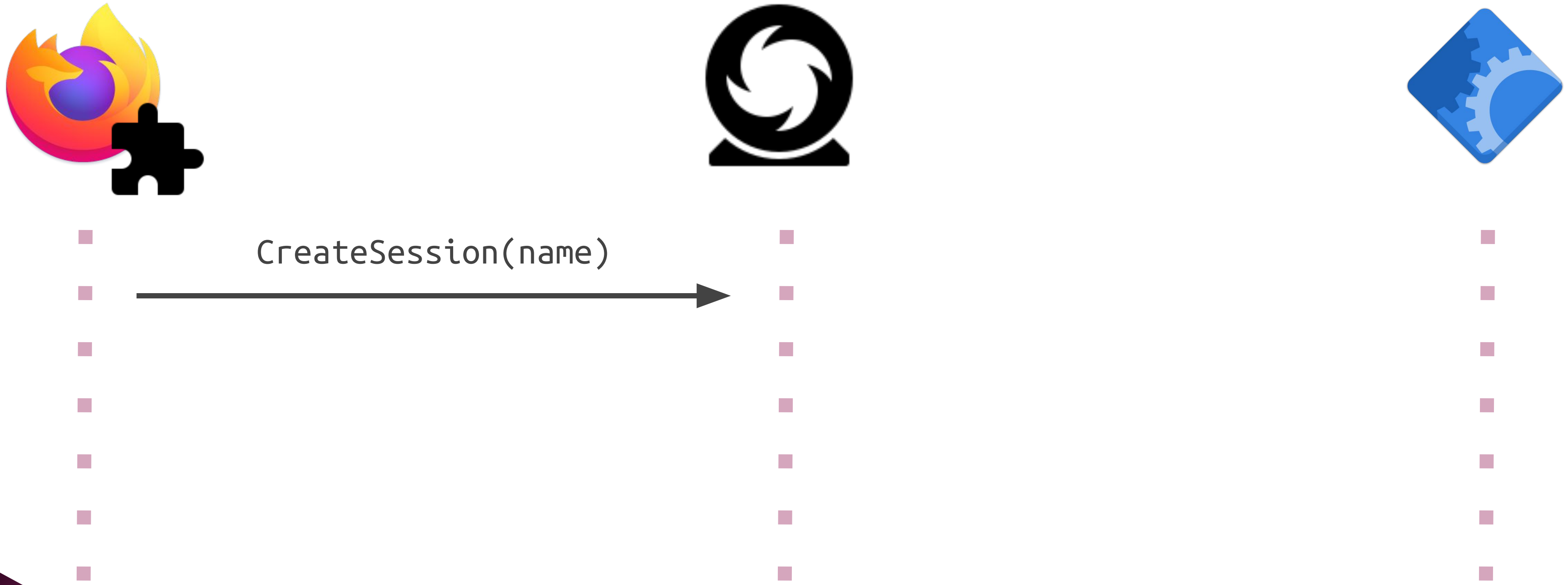
How does this work?



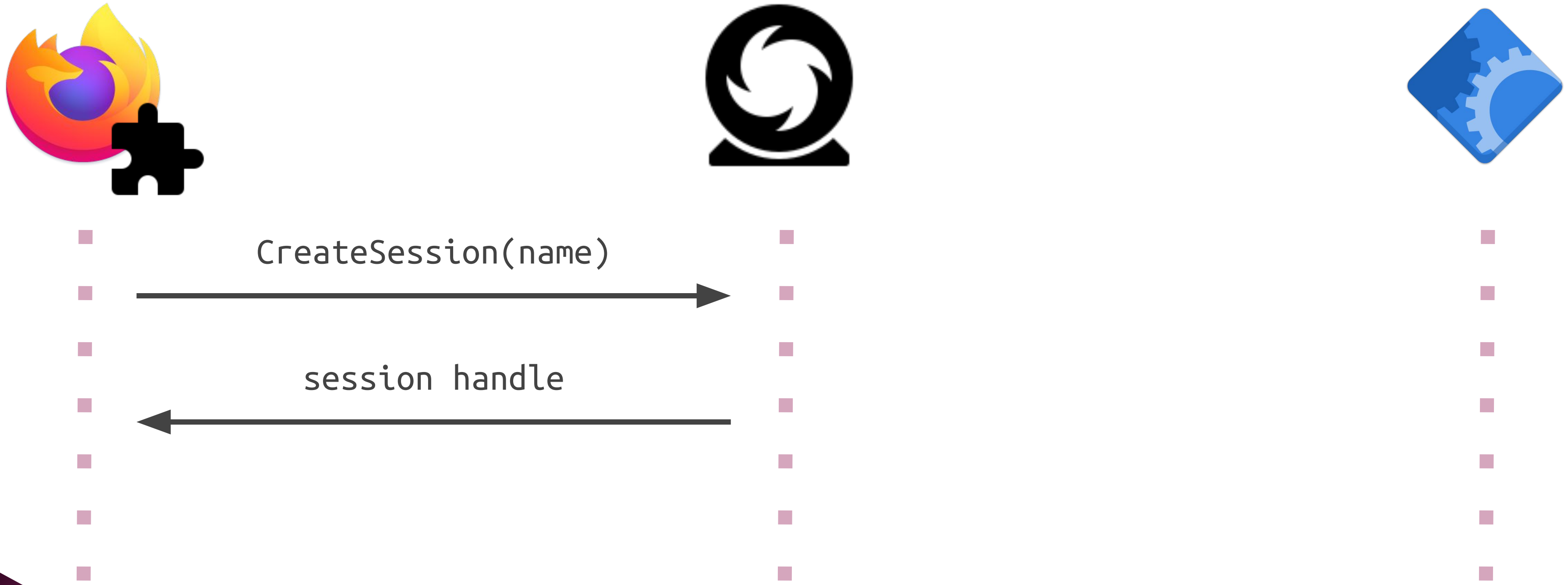
How does this work?



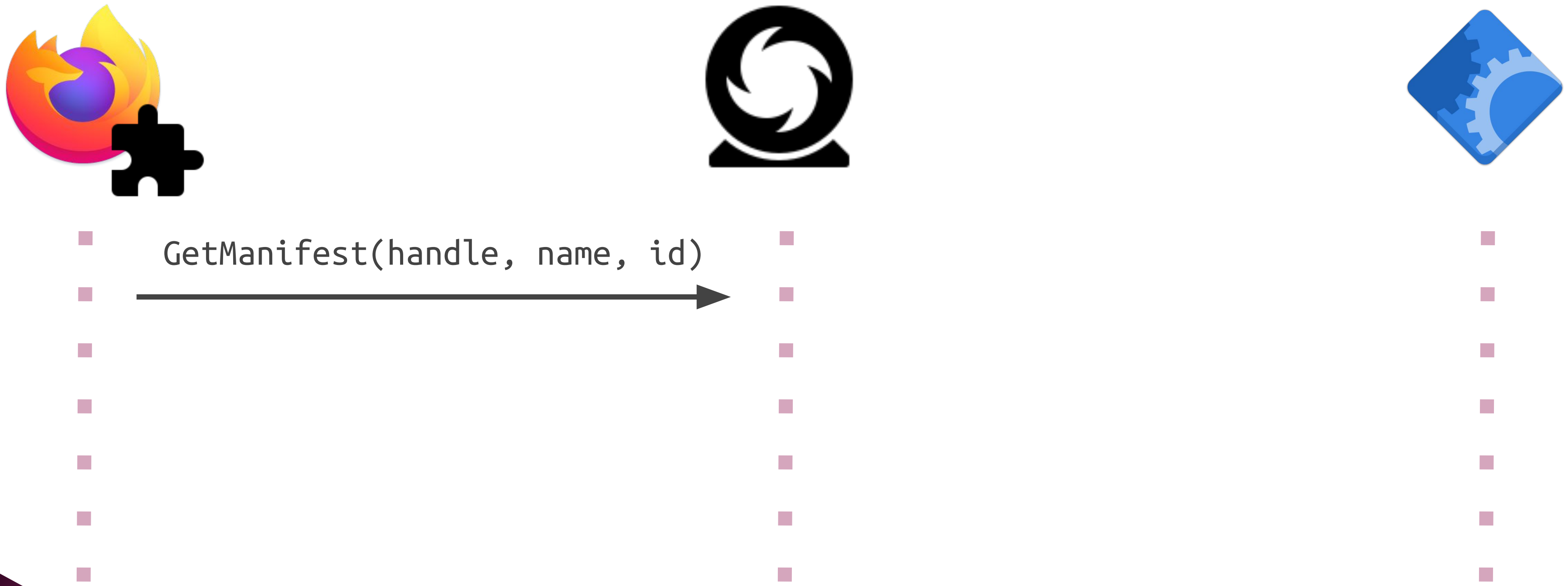
How does this work?



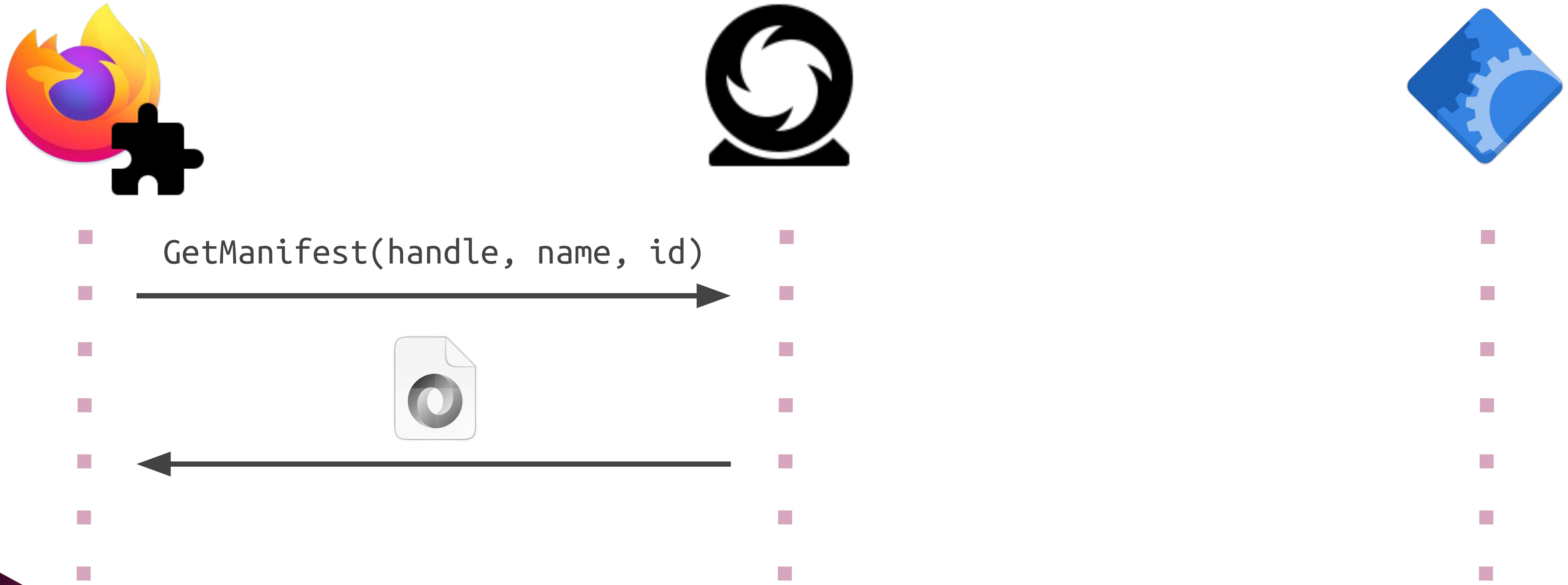
How does this work?



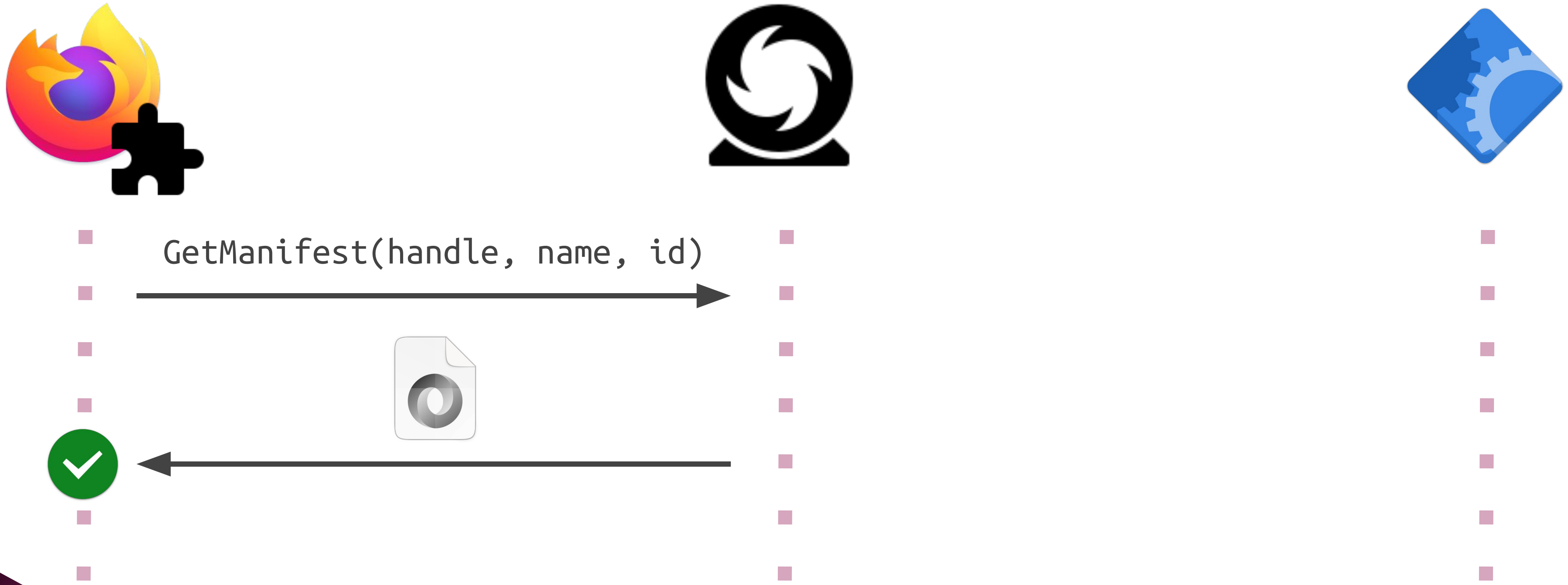
How does this work?



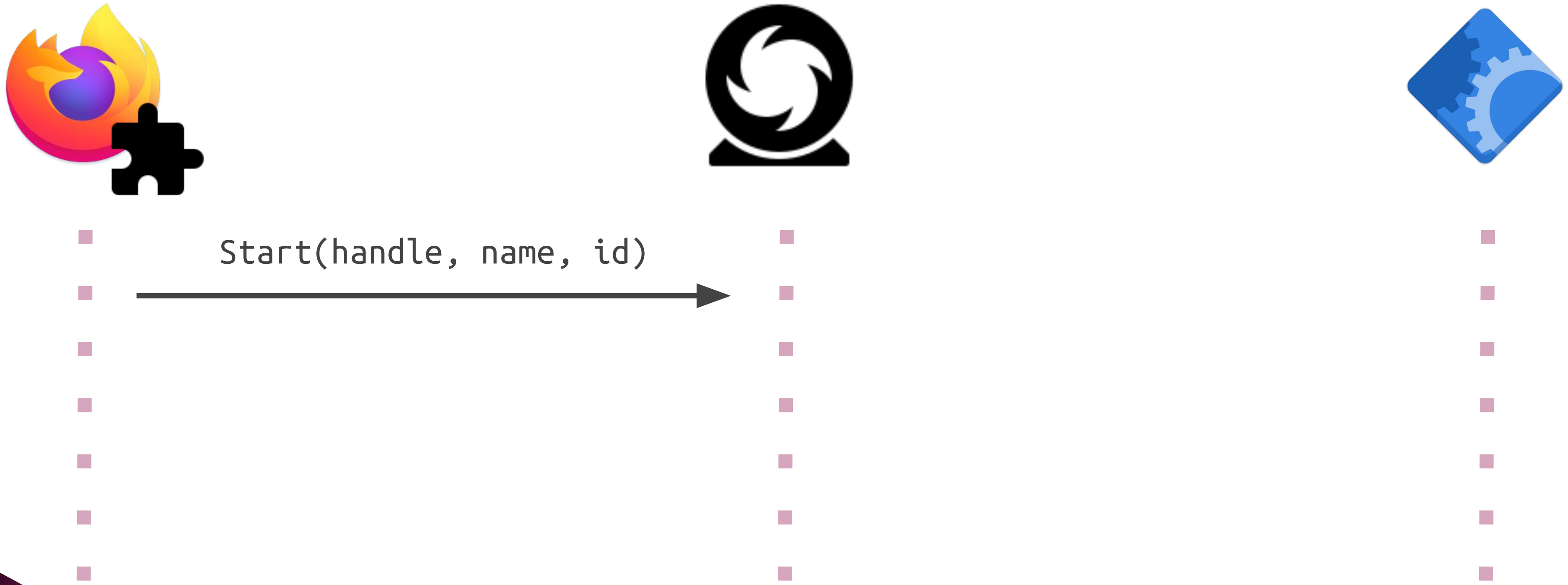
How does this work?



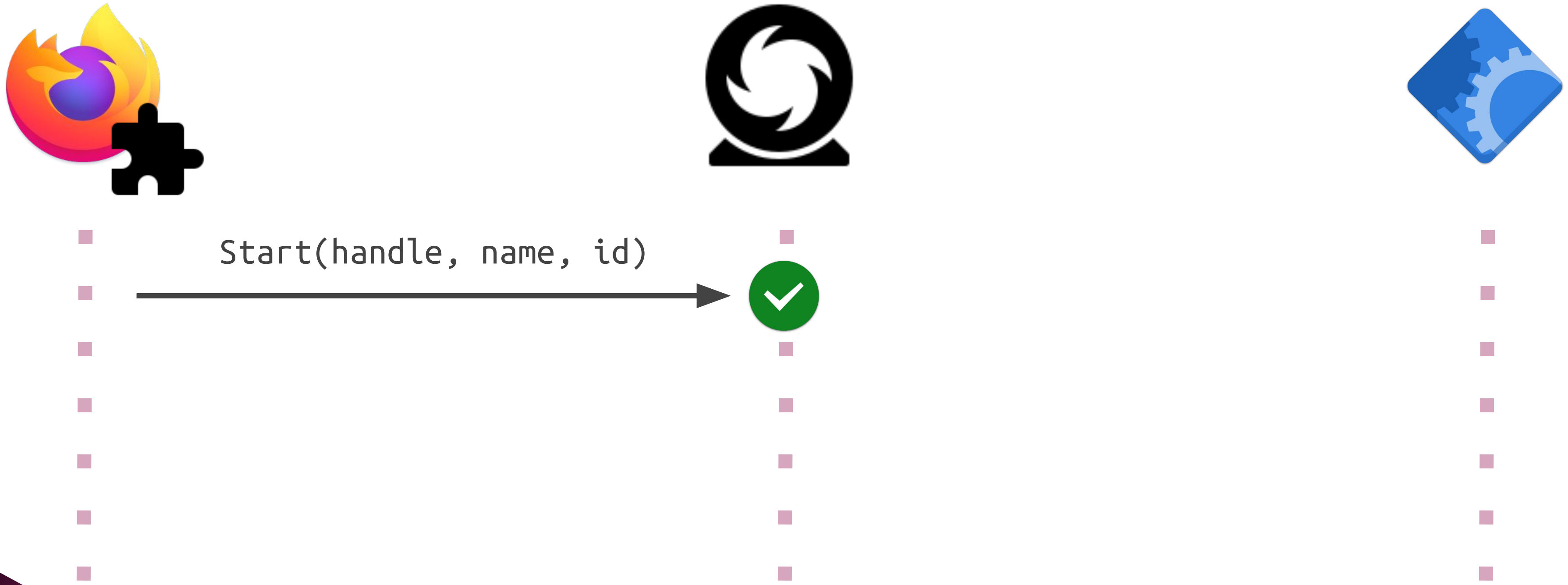
How does this work?



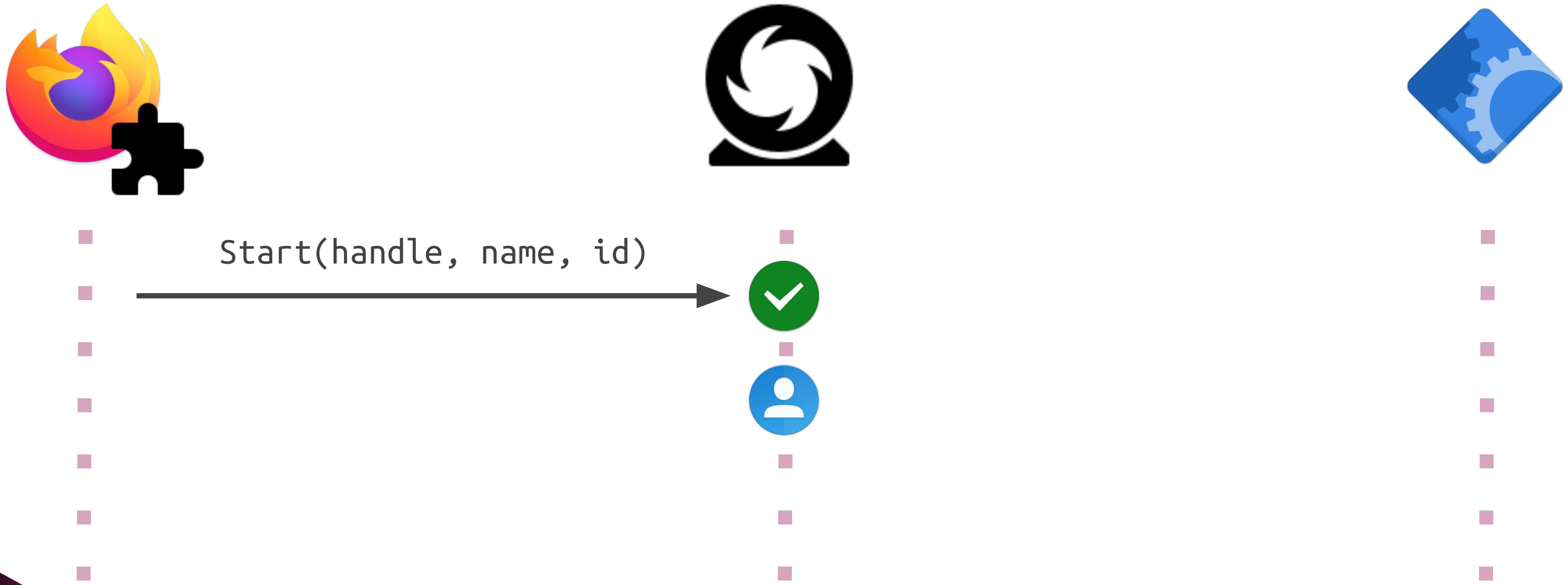
How does this work?



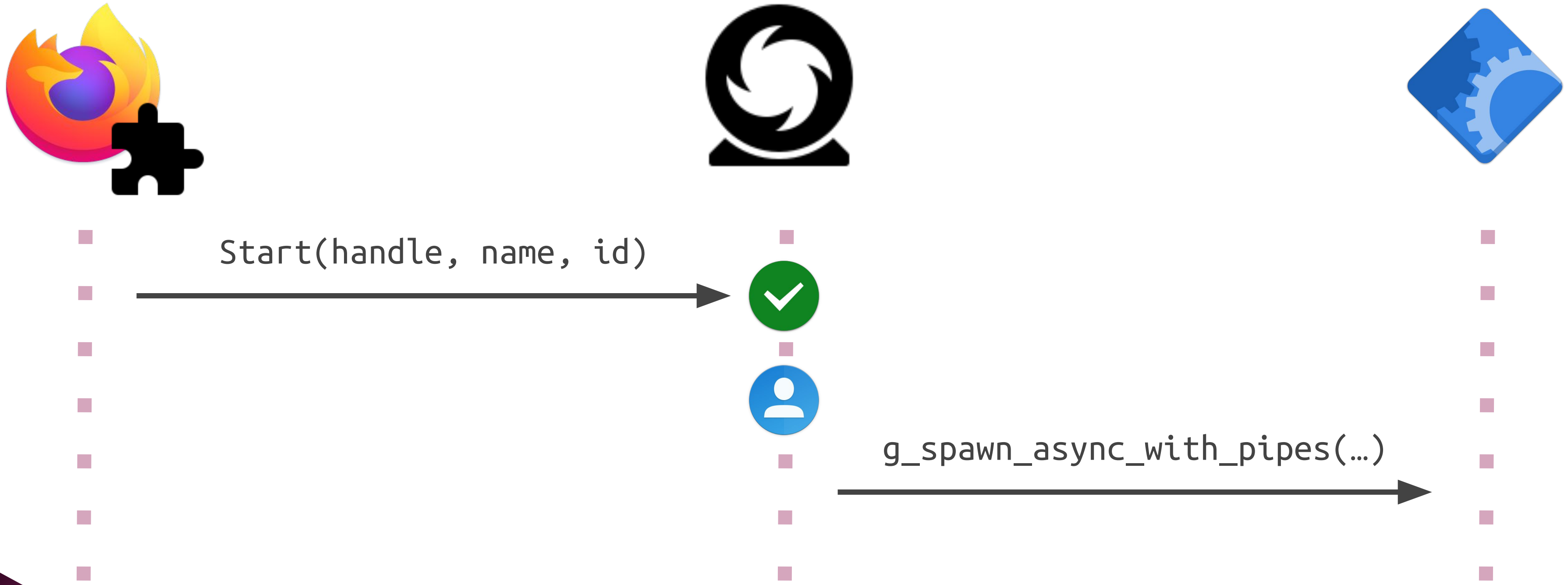
How does this work?



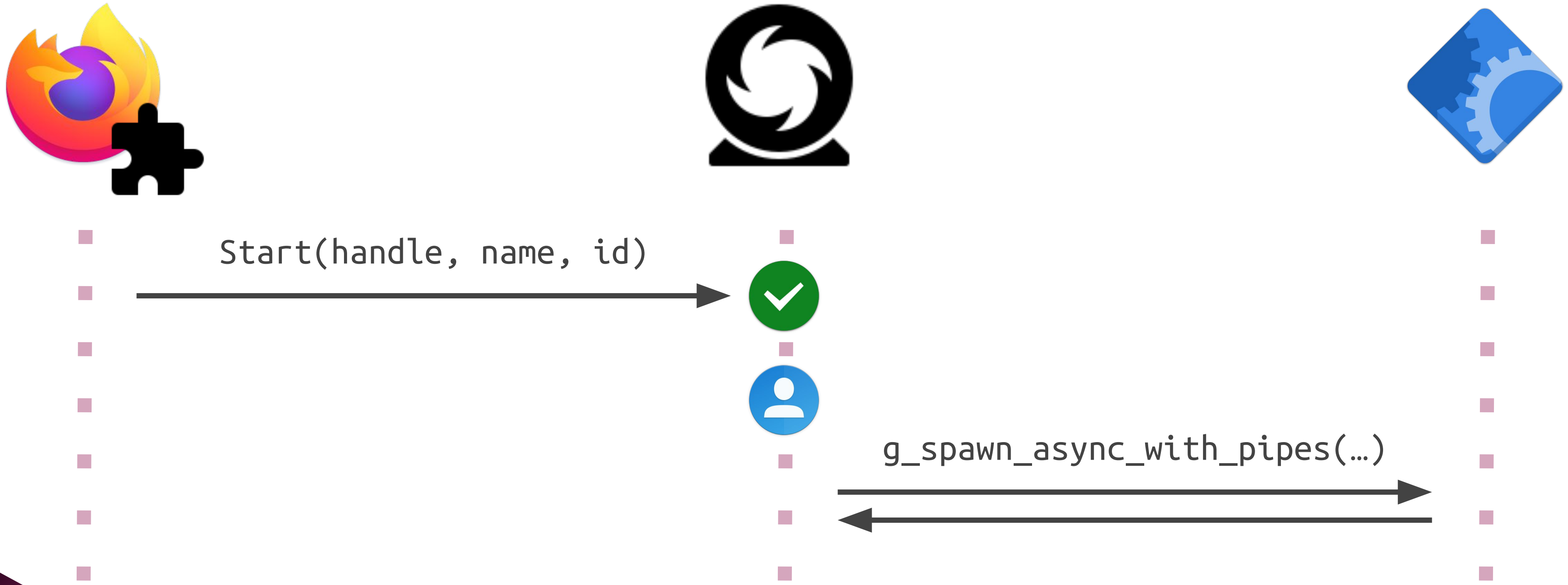
How does this work?



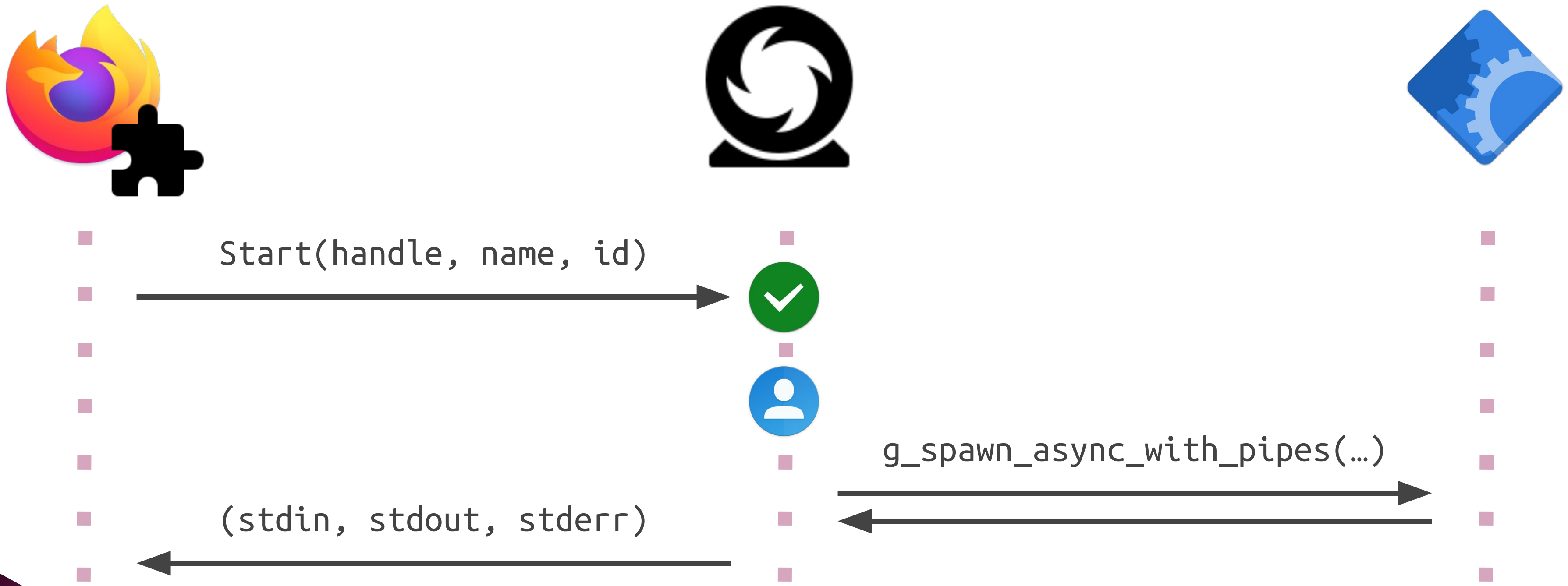
How does this work?



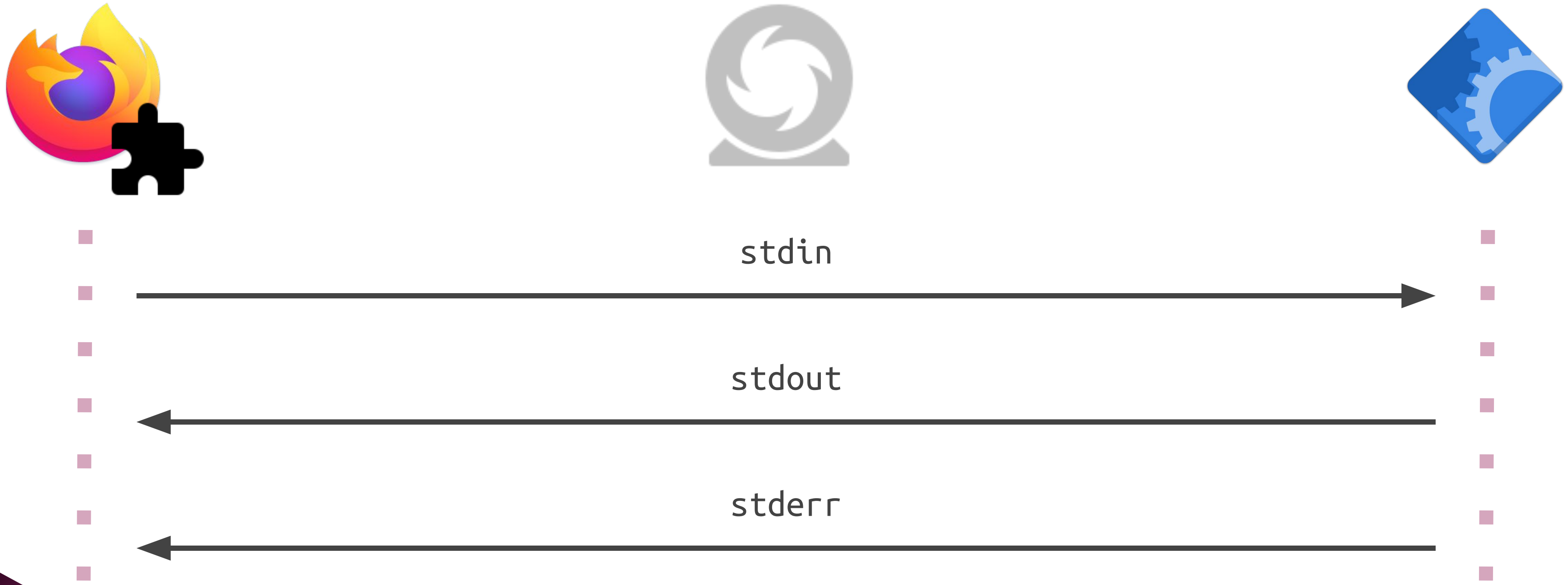
How does this work?



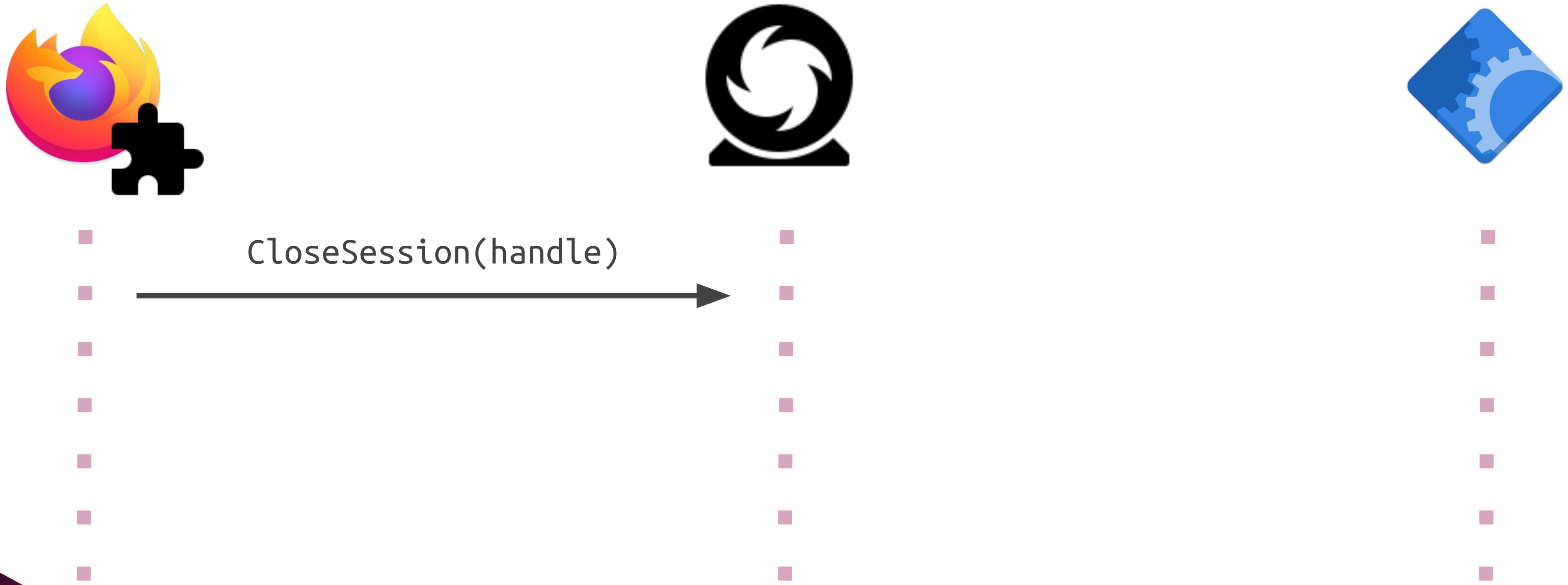
How does this work?



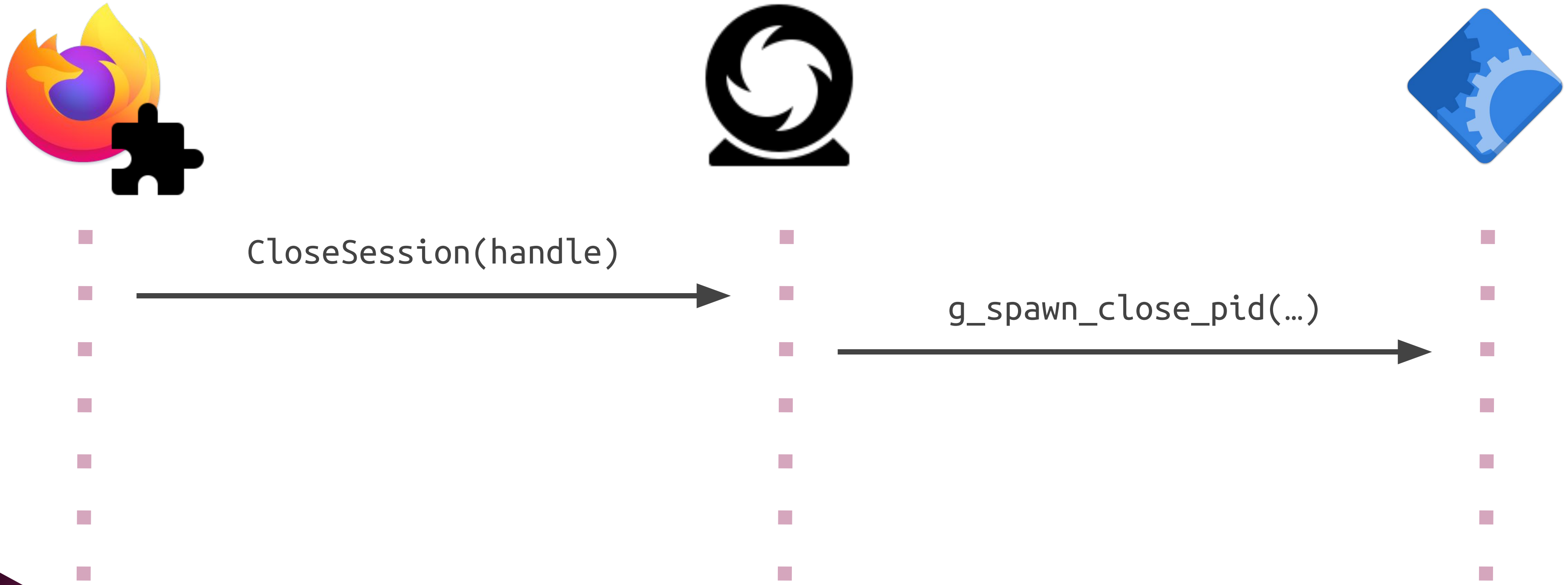
How does this work?



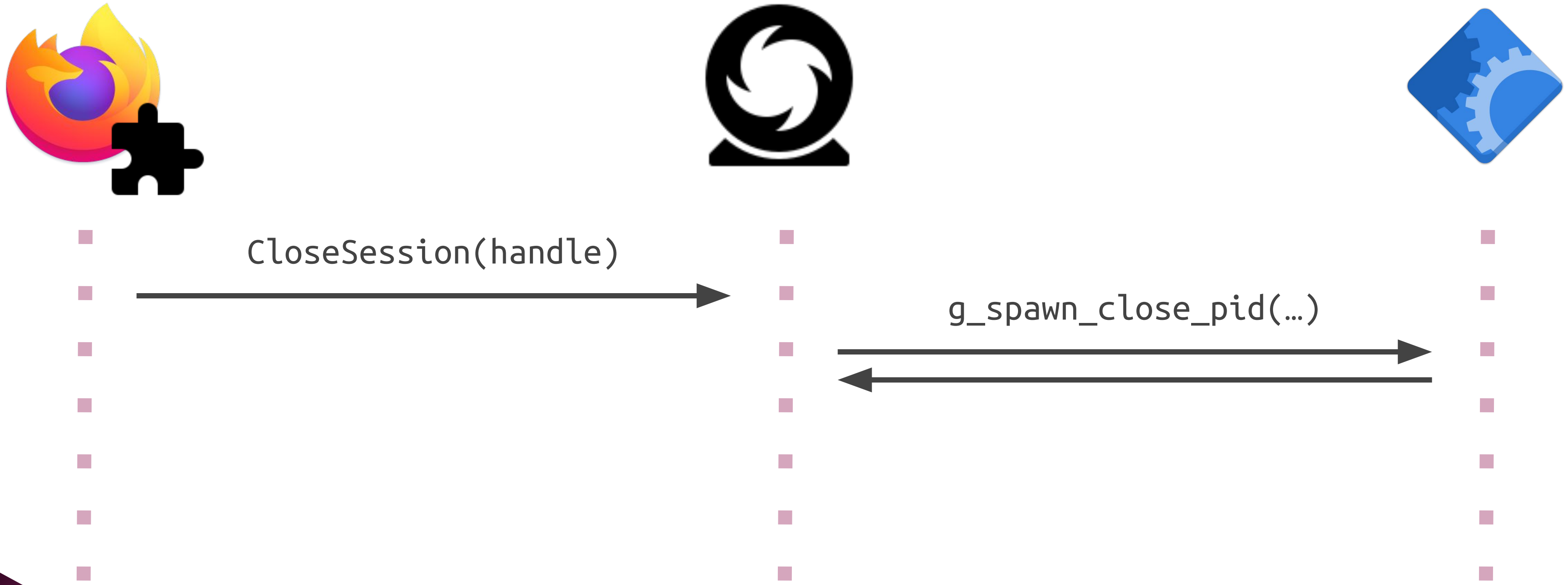
How does this work?



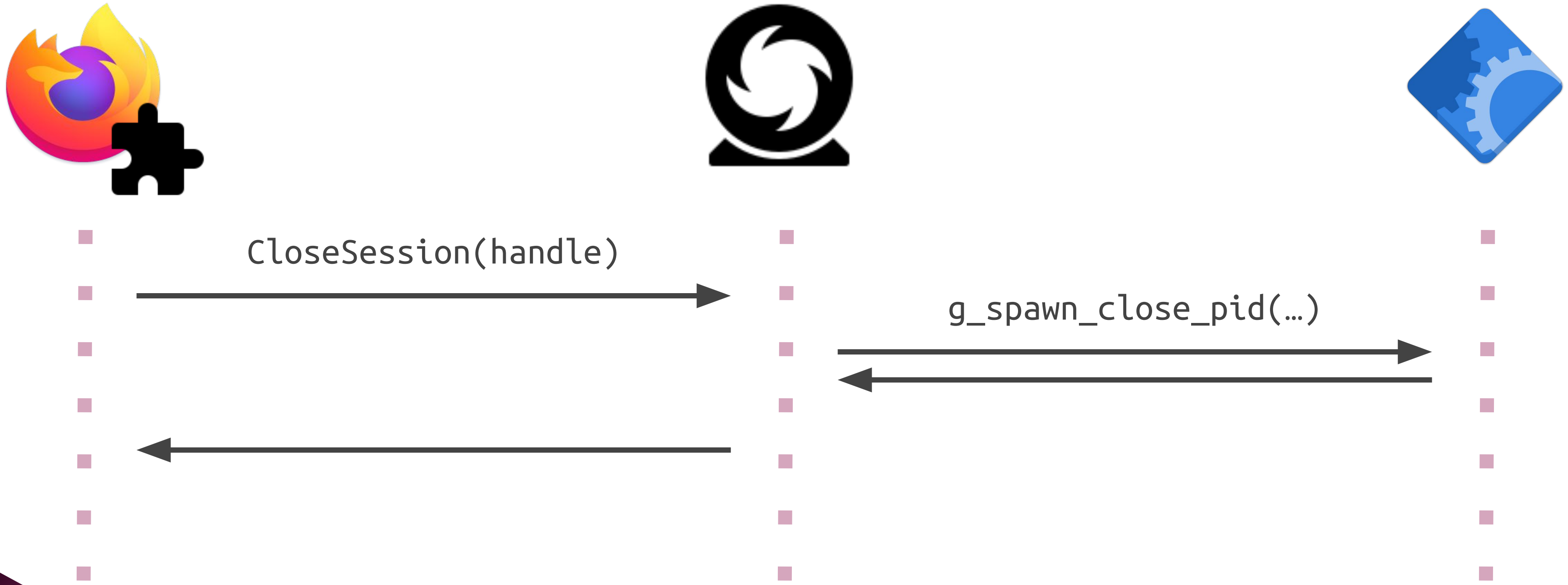
How does this work?



How does this work?



How does this work?



Demo

- chrome-gnome-shell (deb)
- KeepassXC (snap)

Future work

- Get consensus on the portal and get it merged upstream
- Land the firefox patch
- Extract CLI tool to manipulate the permissions db out of flatpak
- Create a UI in the settings app to allow changing those permissions
- Replicate this work for chromium and possibly other browsers
- Remove the KeepassXC warning about snaps

Thank you. Questions?