

Flatpak Next

Adrian Vovk (adrianvovk)
fosstodon.org/@AdrianVovk
avovk@redhat.com

Sebastian Wick (swick)
hachyderm.io/@swick
sebastian.wick@redhat.com

...but let's start with "Next in Flatpak"

Incremental Improvements: systemd-appd

- systemd-appd: Centralized app identity and permission storage
- Services can query appd to learn about who their peer is and what the peer is allowed to do
- Move sandboxing enforcement directly into the services
- No more proxies or protocol-specific security contexts
- Mount services directly into the sandbox
- Enables subsandboxing (user namespaces)

Incremental Improvements: other

- Entitlements
- Replace flatpak-builder with BuildStream
- Network Sandboxing
 - Localhost, LAN, Internet, or some combo of all three
 - For localhost: restrict access by port
 - DNS
- x11 (Xwayland-satellite), x11-less runtime/x11 extensions, rootful Xwayland fallback

New Portals

- Audio (Pipewire) Portal
- Writing Assistance
- hidraw devices, serial devices
- GPU daemon (XDC talk)
- Document Portal improvements
- VPN Portal
- Password managers, autofill
- Credentials, FIDO, Passkeys

Flatpak Next: Why

- Backwards compatibility $\Rightarrow$ cruft, complexity, security issues
- So we need something to break backwards compat...
- ... but can be co-installed with Flatpak v1 (no flag days!)
- (and keep all of the Flatpak v1 and Portal improvements from earlier)

Flatpak Next: What

- Rust 🦀
- Design from the beginning to resist path traversal
- No more libflatpak, expose Varlink service instead
- Pluggable storage backends
- Include collection IDs in app identity

Flatpak Next: What

- Rethink how we do static permissions
- Separate distribution from sandboxing
 - unconfined mode, but only for remotes the user trusts
 - everything else is always completely sandboxed
- Distros now consistently support unprivileged user namespaces, so we can stop using bubblewrap

Portals Next

- Complete rearchitecture, but again coexist with current portals
- Reminder: systemd-appd for app authentication and permission management
- Remove frontend-backend split
- Use Varlink instead of D-Bus
 - Direct p2p Unix socket connections!
 - No more need to work around D-Bus via Request and Session abstractions
- Desktop environment is responsible for starting portals like other session services (systemd socket activation?)
- Shared repo for interface definitions and tests

Conclusion

- No timeline
- Please give feedback
- Transparent for end-users
- For app developers, it will mostly be about following Flatpak v1 best practices
- The future is bright! ✨

Adrian Vovk (adrianvovk)
fosstodon.org/@AdrianVovk
avovk@redhat.com

Sebastian Wick (swick)
hachyderm.io/@swick
sebastian.wick@redhat.com