

Developing Native Applications with GTK



These slides

<https://tinyurl.com/developing-apps-with-gtk>





Felipe Borges - GNOME Developer / Red Hat Developer

feborges at irc.gnome.org

✉ felipeborges@gnome.org / feborges@redhat.com

Who is using Gtk?

- Desktop - GNOME: Fedora, RHEL/Red Hat, Ubuntu/Canonical, Suse
- Just drawing - Firefox / Chrome / WxWidgets on Linux
- Embedded devices - Konami
- Multiplatform .Net IDE - Microsoft with Mono / Xamarin
- Valve / Steam - Dialogs on Linux



Key concepts

- Widgets / Containers / Constraints
- Events / callbacks
- C based with GObject
- Multi platform
- Multi language with GObject Introspection
- Single threaded



Language bindings

- C / C++
- Python
- JS
- Vala
- C# / Mono
- Rust



API

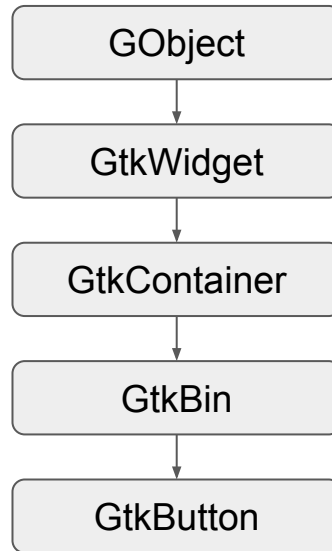
Prefixed snake case

\$project_\$object_\$action:

g_ | hash_table | _new

gtk_ | button | _set_label

Inheritance



Pros

- Language agnostic
- Low overhead
- Stability
- Flexibility by being able to extend it
- FOSS





Cons

- Not good multiplatform story
- Limited in some areas such as animations
- Not the best documentation (yet)



Target

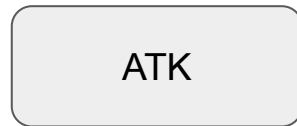
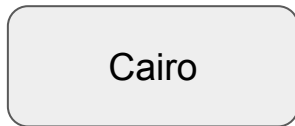
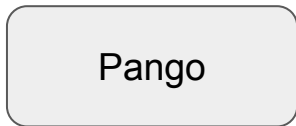
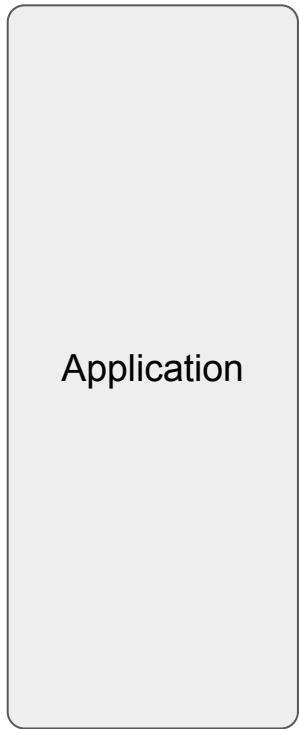
- Desktop applications
- Embedded medium powerful devices
- Cross platform not being an important factor



Tooling

- GNOME Builder as IDE
- GtkInspector
- DevHelp
- HIG
- Glade





High level

Low level

Gtk - GLib - seriously, why C...

- C is very flexible
- C is the de facto common programming language
- Interaction with the kernel and low level libraries
- Allows easy introspection
- Low overhead



Signals and event handling

- Types of events
- Connecting and disconnecting signal handlers
- Signal emission and propagation



Resources

- Gtk and HIG: <https://developer.gnome.org>
- Python Gtk: <https://lazka.github.io/pgi-docs>
- Help: irc.gnome.org #newcomers and #gtk



Demo time!



Thank you!

