Wayland intro

with i18n hacks

Presented by

Peng Wu

Red Hat

Today's Topics

1. X Window Background
2. Wayland intro
3. Wayland with input method frameworks
4. Wayland i18n hacks
5. Summary

X Window Background

X Window History

- X originated at the MIT in 1984
- Since then, never touched the core protocol

Why not X?

- parts of the core protocol seldom used today
  - X11 core graphics functionality seldom used
  - XIM protocol bypassed by input method frameworks, such as ibus, scim, fcitx.

Client Rendering Steps

- clients render locally
- clients tell the X server what they've drawn
- the server asks the WM to display it
- the window manager decides what to draw and where
- the X server displays what the WM rendered
- So, the window manager is the new X server

Wayland intro

Wayland dissected

- Wayland - Wayland Compositor Infrastructure
- Weston - The Weston Wayland Compositor
- libinput - input device management and event handling library

Wayland intro

- RPC mechanism designed for window system
  - Utility functions for client/server development
- Generate codes from XML protocol definition
  - some client/server codes for communicating with the display server.
- Define the core wayland protocol
  - in protocol/wayland.xml

Weston display server

- The Reference Wayland Compositor
  - Reference Implementation
- Support additionally protocols
  - in weston/protocol
  - Maybe we could call these as extensions?
- For input method support
  - See text.xml and input-method.xml

Client Rendering Steps

- clients render locally
- clients tell the server what they've drawn
- the server decides what to draw and where
- So, Wayland just cut out the middle man

Wayland Pros

- it's tractable
- 'every frame is perfect'
  - a set of pixels that should be shown in a window coherently at one time

solely frame-based

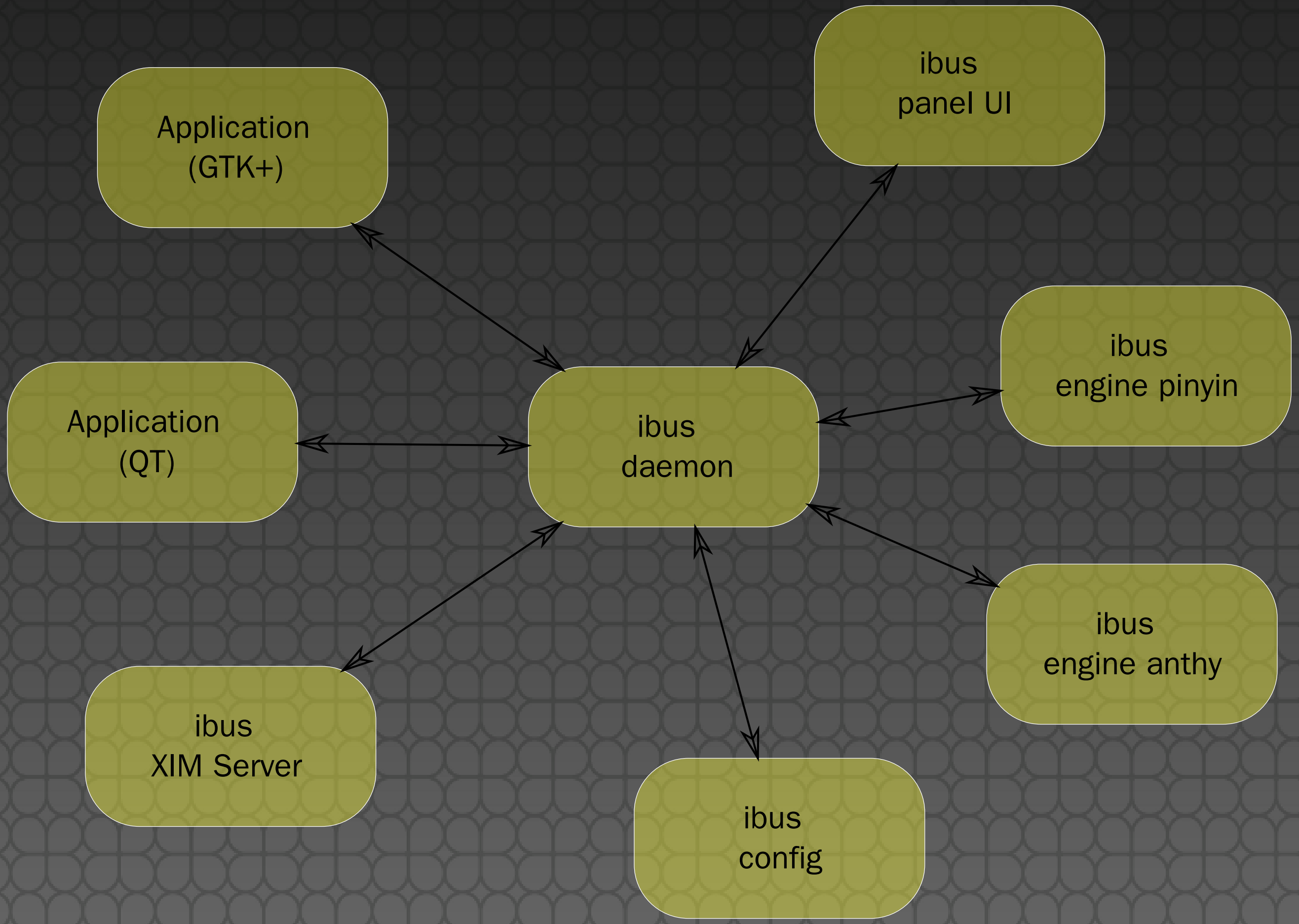
- client says, 'display this'
- server displays it
- job done.

Wayland with input method frameworks

ibus intro

- Bus-Centric Architecture
  - It has an ibus-daemon, which manages all clients.
  - all engines, panel, config modules & clients are clients of ibus-daemon.
- iBus is based on dbus IPC protocol.

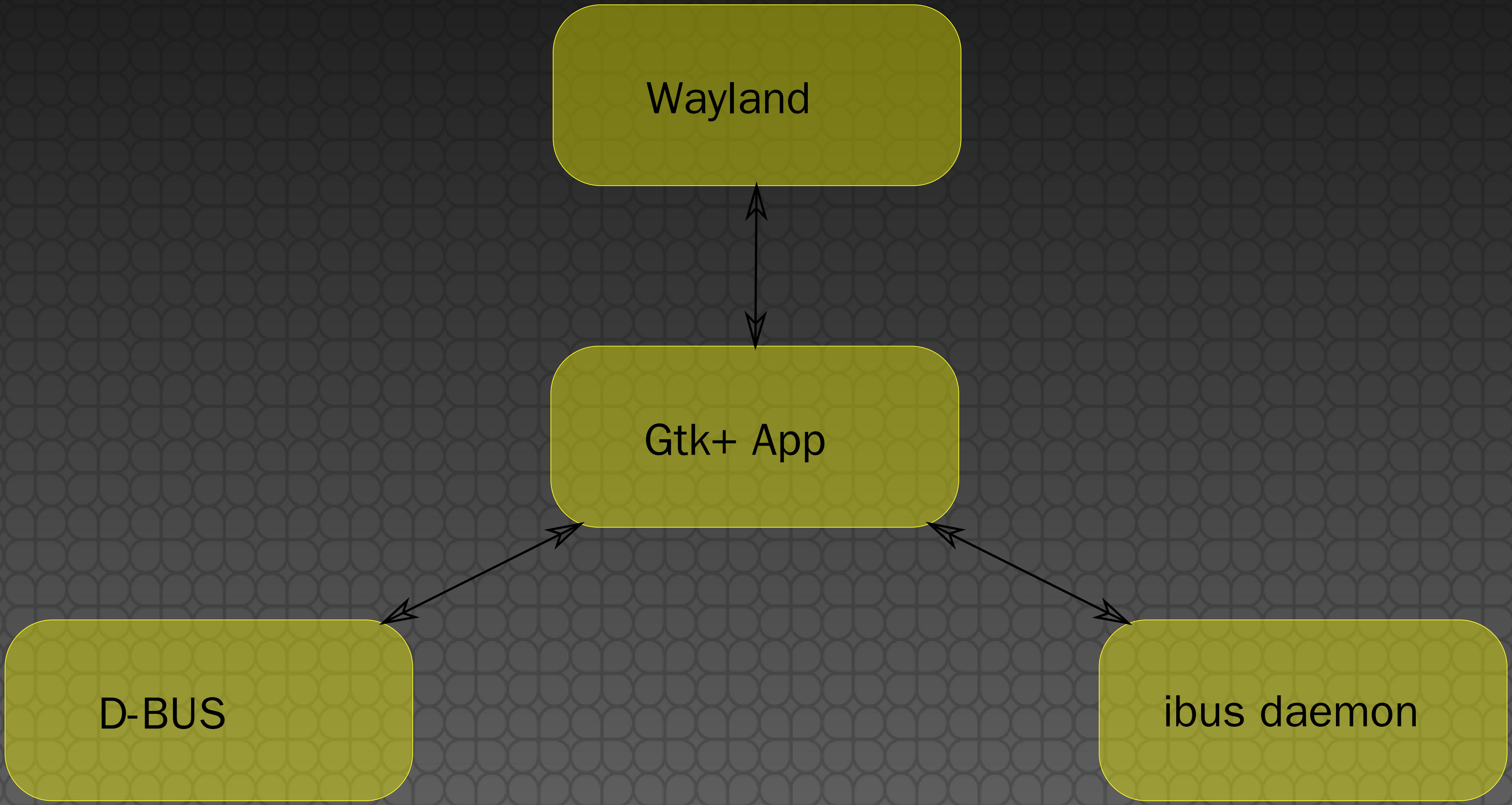
ibus architecture



ibus with X

- XIM protocol is old, and have many problems.
- Many GUI ToolKits implement own IMModules
  - Gtk+ - GtkIMContext
  - QT – QinputContext
- The XIM is by-passed by default configuration.

Proposed architecture



Advantages

- Every daemon serves one aspect of functionality.
  - weston for display
  - dbus for RPC
  - ibus for input
- Gtk+ Apps talks to the specific daemon only when needed.
- Why not by-pass display server again? ;-)

Wayland i18n hacks

weston terminal hacks

- fixes dual-width glyph spaces problem
  - learned it from vte widget
- Fixes cursor problem with vim, emacs
  - respect the virtual terminal spec

Summary

- Wayland is wonderful
- more works on Wayland i18n supports
  - input methods
  - font rendering

Questions?

Contact:
example@fedoraproject.org