Wayland intro and ibus support

Presented by

Peng Wu

Software Engineer

Today's Topics

1. wayland intro
2. ibus wayland support

wayland intro

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.

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

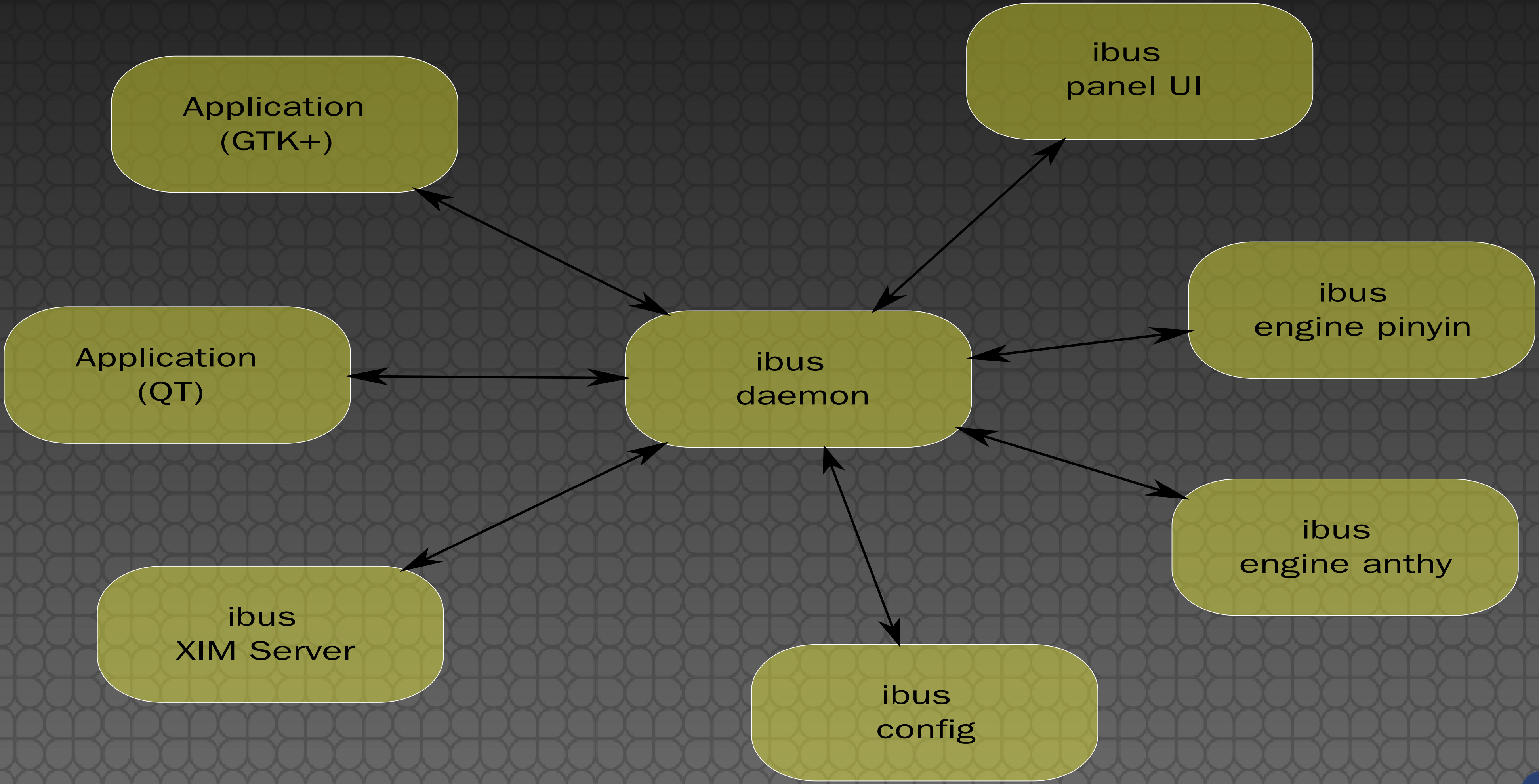
- The Reference Wayland Compositor
  - Reference Implementation
- Support additionally protocols
  - in weston/protocol
  - Maybe we could call these as extensions?
- For input method support
  - See text.xml and input-method.xml

ibus wayland support

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 base on dbus IPC protocol.

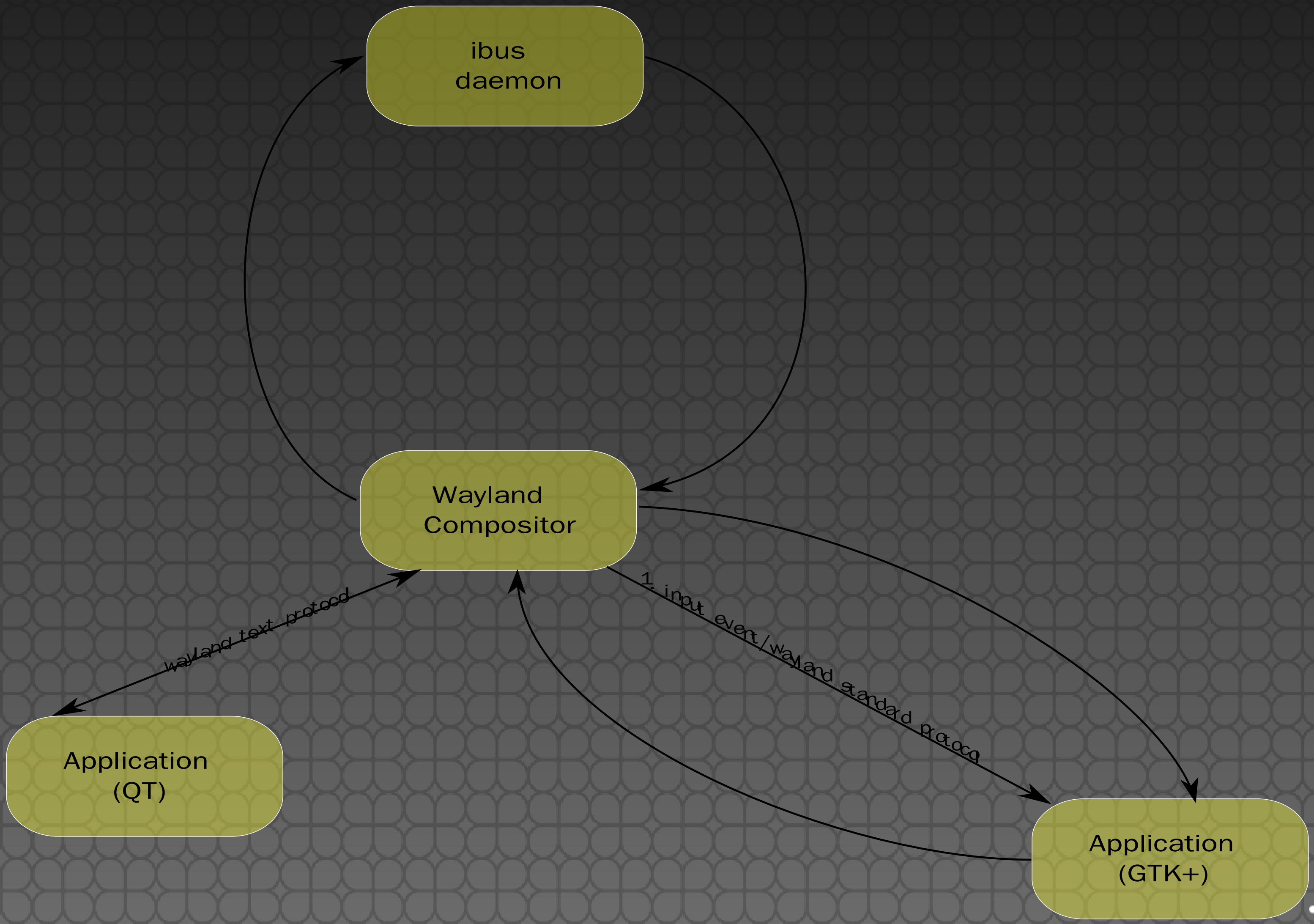
ibus architecture



Advantage of iBus

- iBus uses dbus RPC for communication.
  - Clients can use any languages to communicate with ibus
  - Engines can be wrote by any languages
- The daemon, engines, clients & UI, all partitions are running in separated processes
  - One of them has problem, it will not impact all system
  - All partitions can be restarted on the fly.
  - Engine can be loaded & unloaded on demand.

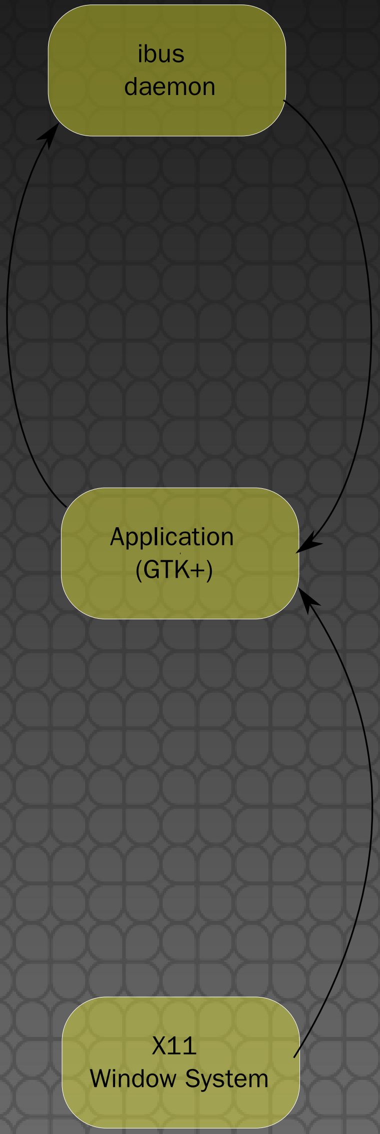
The openismus design



openismus event flow

1. the wayland compositor send the input event to the application;
2. the application sends the input event to the wayland compositor by wayland text protocol;
3. wayland compositor sends the input event to the ibus daemon by wayland input method protocol;
4. ibus daemon sends back the input method event to wayland compositor by wayland input method protocol;
5. wayland compositor sends back the input method event to the application by wayland text protocol;

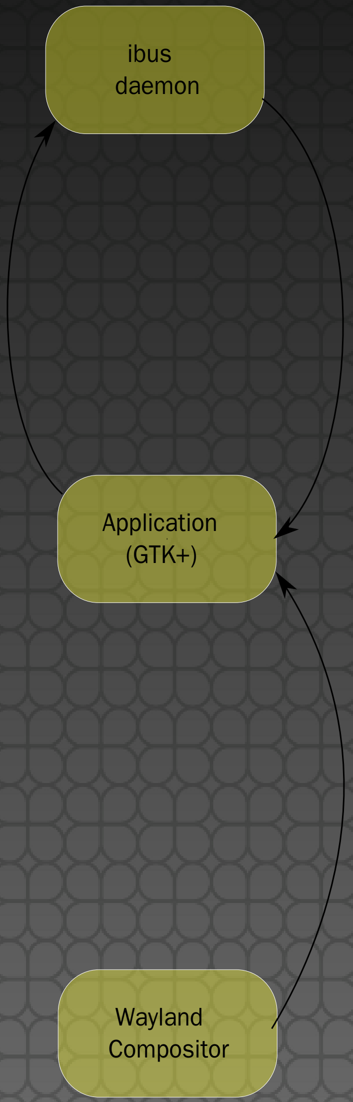
The GTK+ design for X11



GTK+ for X11 event flow

1. the X11 Window System send the input event to the application;
2. the application send the input event to ibus daemon by ibus protocol;
3. the ibus daemon send back the input method event to the application by ibus protocol;

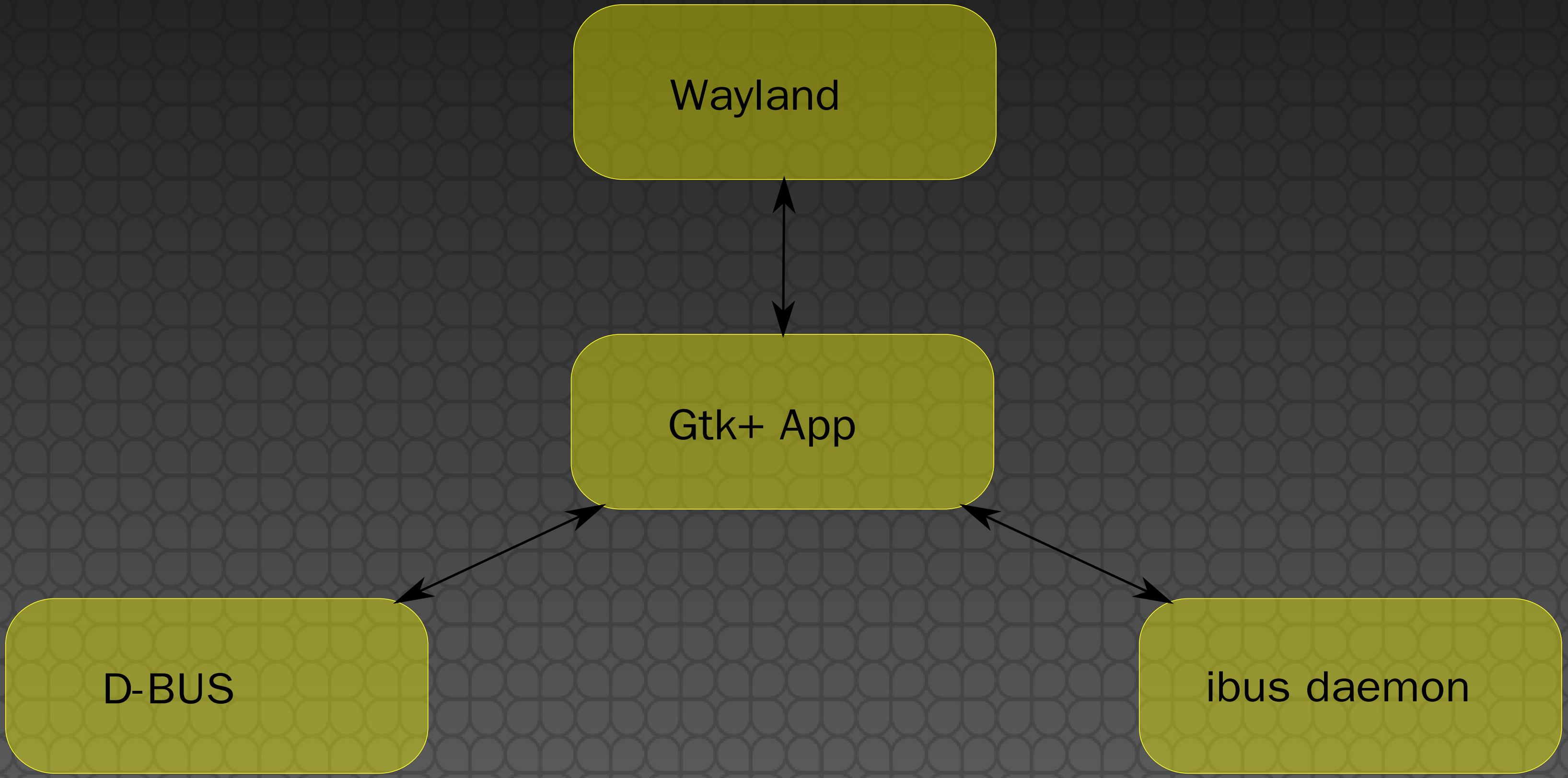
Proposed GTK+ design



Proposed event flow

1. the wayland compositor send the input event to the application;
2. the application send the input event to ibus daemon by ibus protocol;
3. the ibus daemon send back the input method event to the application by ibus protocol;

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.

Summary

- wayland intro
- ibus wayland support
- Proposed a new architecture for input support

Questions?

Contact:
example@fedoraproject.org