

# LightDM: Cross Desktop Display Manager

## Robert Ancell



# What is a Display Manager?

- High level:
  - Run display servers (X)
  - Authenticate users (greeter)
  - Start sessions (GNOME, KDE, XFCE, ...)
  - Remote login (XDMCP)



# What is a Display Manager?

- Low level:
  - Persistent daemon
  - Manage display server and session processes
  - Support multiple logins
  - Switch users



# In the past...



# ...there was XDM



# Today...

XDM

GDM

KDM

LXDM

SLiM

uxlaunch

## ..we have many



# What's the difference?

XDM

GDM

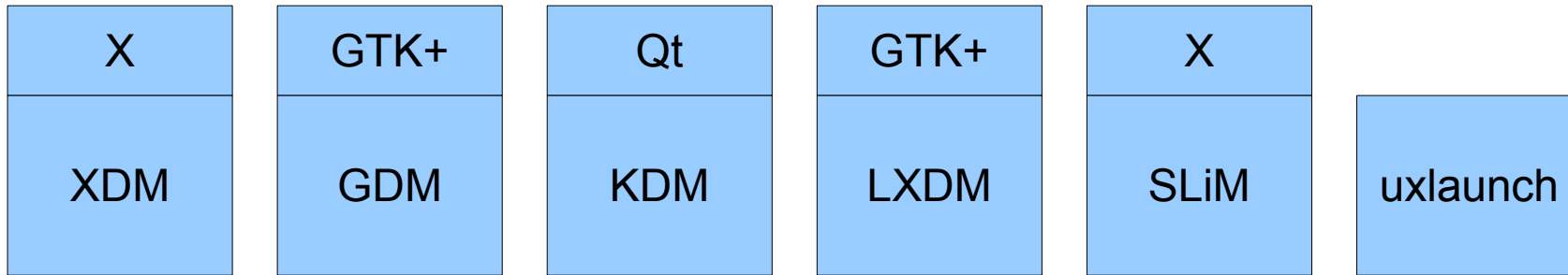
KDM

LXDM

SLiM

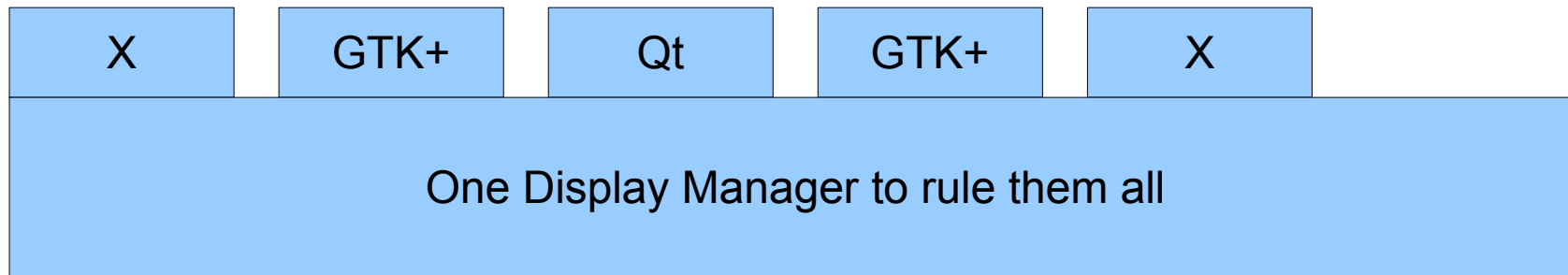
uxlaunch

# What's the difference?



## Mostly different GUIs

# Why not this?





# Introducing LightDM

- Freedesktop project
  - Specification (TODO)
  - Sponsored by Canonical for Ubuntu
- Ip:lightdm
- GPL3
- C/GObject



# Core features

- GObject / Qt greeter support
- Automatic login
- Guest accounts
- User switching
- XDMCP
- Multi-seat
- Test mode
- Regression tests



# Planned features

- Wayland support
- VNC / RDP login
- Improved two-factor authentication
- Fallback session support



# Example greeter (GObject)

```
#include <lightdm.h>

static void show_prompt (LightDMGreeter *greeter, const gchar *text, LightDMPromptType type)
{
    lightdm_greeter_respond (greeter, "password");
}

static void authentication_complete_cb (LightDMGreeter *greeter)
{
    if (lightdm_greeter_get_is_authenticated (greeter))
        lightdm_greeter_start_session_sync (greeter, NULL, NULL);
}

int main (int argc, char **argv)
{
    GMainLoop *main_loop = g_main_loop_new (NULL, FALSE);
    LightDMGreeter *greeter = lightdm_greeter_new ();
    g_signal_connect (greeter, "show-prompt", G_CALLBACK (show_prompt_cb), NULL);
    g_signal_connect (greeter, "authentication-complete", G_CALLBACK (auth_complete_cb), NULL);
    lightdm_greeter_connect_sync (greeter, NULL);
    lightdm_greeter_authenticate (greeter, "username");
    g_main_loop_run (main_loop);
}
```



# Example greeter (Qt)

```
#include <TestGreeter>
#include <QtCore/QCoreApplication>

void TestGreeter::onShowPrompt (QString text, QlightDM::PromptType type)
{
    respond ("password");
}

void TestGreeter::onAuthenticationComplete ()
{
    if (isAuthenticated ())
        startSessionSync ();
}

int main (int argc, char **argv)
{
    QCoreApplication app (argc, argv);
    TestGreeter *greeter = new TestGreeter ();
    connect (greeter, SIGNAL (showPrompt (QString, QlightDM::PromptType)),
            SLOT (onShowPrompt (QString, QlightDM::PromptType)));
    connect (greeter, SIGNAL (authenticationComplete ()),
            SLOT (onAuthenticationComplete ()));
    greeter.connectSync ();
    greeter.authenticate ("username");
    app.exec ();
}
```



# Example greeter (Python)

```
from gi.repository import GObject
from gi.repository import LightDM

def show_prompt (greeter, text, type):
    greeter.respond ("password");

def authentication_complete_cb (greeter):
    if (greeter.is_authenticated)
        Greeter.start_session_sync ();

main_loop = GObject.MainLoop ();
greeter = LightDM.Greeter ();
greeter.connect ("show-prompt", show_prompt_cb)
greeter.connect ("authentication-complete", auth_complete_cb)
greeter.connect_sync ()
greeter.authenticate ("username")
main_loop.run ()
```



# Example greeter (Vala)

```
class TestGreeter
{
    public static void show_prompt (LightDM.Greeter greeter, string text, LightDM.PromptType type)
    {
        greeter.respond ("password");
    }

    public static void authentication_complete_cb (LightDM.Greeter greeter)
    {
        if (greeter.is_authenticated)
            try { greeter.start_session_sync (); } catch (Error e) {}
    }

    public static int main (string[] args)
    {
        var main_loop = new MainLoop ();
        var greeter = new LightDM.Greeter ();
        greeter.show_prompt.connect (show_prompt_cb);
        greeter.authentication_complete.connect (auth_complete_cb);
        try { greeter.connect_sync (); } catch (Error e) {}
        greeter.authenticate ("username");
        main_loop.run ();
    }
}
```



# Demonstration





# How can you help?

- Distributors – package LightDM
- Upstream projects – consider if LightDM is right for you.
- Developers – Write new greeters
- Users – test and feedback



# Thanks

- Brian Cameron (Oracle)
- David Edmundson (KDE)
- Guido Berhoerster (OpenSUSE)
- Yves-Alexis Perez (Debian)

Still have a low bus factor...



# Questions?

