OpenPrinting

The New Printing GUIs

GNOME Control Center and Common Print Dialog Backends

Demo + AMA (Ask Me Anything)

Till Kamppeter – OpenPrinting

Linux App Summit, April 22, 2023

Introduction

- **The New Architecture – Pure IPP for Printing and Scanning**
  - CUPS 3.0/CUPS Snap: R. I. P. PPD files, all-IPP workflow
- **Printer Setup Tools**
  - IPP Services, not Queues; Printer Applications, not drivers
- **Print Dialogs**
  - IPP Attributes, not PPD options; Temporary Queues, Common Print Dialog Backends

The New Architecture

- For 22 years now, since its 1.0 launch, CUPS uses principally the same architecture:
  - **PostScript was standard job format** as printers typically used with UNIX were PostScript
  - Capabilities of a printer are described by a **PPD (PostScript Printer Description)** file
  - PPD describes all user-settable options, resources (trays, paper sizes, resolution, quality, color, ...) in a static text file
  - To cover non-PostScript printers **PPD format got extended** (by Michael Sweet) to specify a filter to generate printer's native format
  - Filters use **Ghostscript** to convert PostScript input
  - **Manually created queue** with driver (= PPD + filter)

The New Architecture

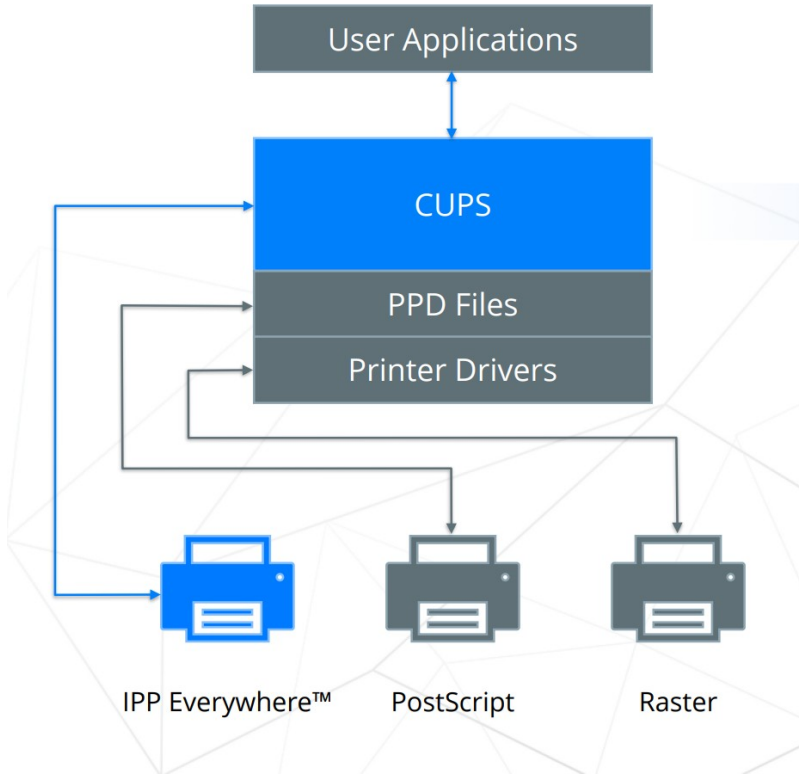
- Why do we want to do away with PPD files?
  - **In 1984 Adobe stopped development on PPD format**, so we started with an obsolete (but useful) format right away
  - In 2006 we **abolished PostScript** as print job format and **replaced it by PDF**
  - PPD files can represent user-settable options only as **enumerated choice or boolean**. Ugly workarounds for things like passwords or color adjustment

The New Architecture

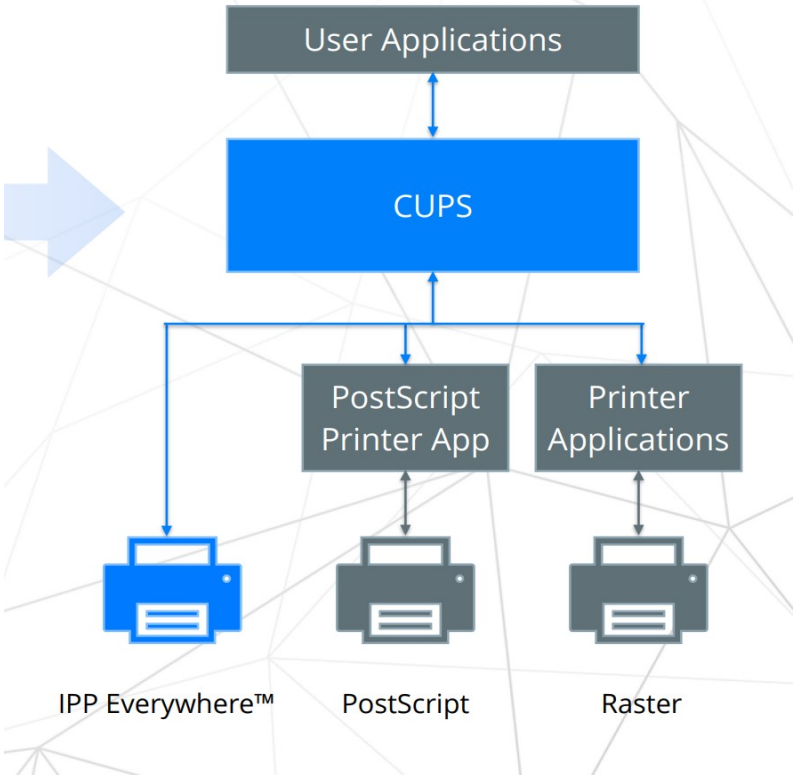
- **PPD-less CUPS – We are all-IPP now**
  - **CUPS 3.0.x will not support PPD files** from the ground up
  - The CUPS Snap does not allow adding PPDs and filters
  - Now **only driverless IPP printers** (IPP Everywhere, AirPrint, Mopria) are supported
  - **No manually created CUPS queues**: IPP printer discovered, temporary queue automatically created
  - Filtering only for driverless standard formats: PDF, PWG Raster, Apple Raster, PCLm output, no need to add filters
  - Legacy/specialty printers which need driver → **Printer Application** emulates IPP printer

The New Architecture

Old CUPS architecture



New CUPS architecture



Printer Setup Tool: How it works currently

- **Printer setup tools**
  - CUPS web admin interface `http://localhost:631/`
  - CUPS command line tools: **`lpadmin`**, **`lpinfo`**, **`lpstat`**
  - **`system-config-printer`** - GUI
  - GNOME Control Center – Print module - GUI
  - **`cups-browsed`** - daemon
- Tools **control CUPS**, the running **cupsd**
  - List available printers and drivers and create print queues
  - List queues and jobs
  - Modify queues
  - Server settings: Owner/everyone can cancel jobs, debug mode, ...

Printer management in the New Architecture

- We assume any form of the New Architecture
  - The **CUPS Snap** - OR -
  - **CUPS 3.x** or newer
- **All Printers are driverless IPP printers**, native or Printer Applications
- CUPS auto-creates virtual queue for each IPP printer → **No manual queue creation required**
- CUPS fully automatic → **Admin action moves to the IPP printers**
- **Tasks**
  - **List IPP services**
    - Buttons to web admin interfaces, IPP System Service, …
  - **Discover non-driverless printers**
    - Find Printer Applications, local and in Snap Store

Printer Setup Tool: GUI Design

- **Similarities** between old and new
  - **Main Window**
    - Old: List CUPS queues, buttons/pop-up to modify
    - New: List IPP devices, buttons to web IF/IPP System Service
  - **Add-Printer Window**
    - Old: List printer devices and drivers, create CUPS queue
    - New: List non-driverless printers, install Printer Application, open Printer Application’s web interface

Printer Setup Tool: GNOME Control Center

- **Support for classic CUPS AND New Architecture**
  - No hard dependency between GNOME and CUPS versions
  - Current CUPS already supports IPP services, Printer Applications, …
- **Main view**
  - CUPS queues with “Set options”, “Change driver”, “Remove queue”, …
  - IPP service with “Open web admin interface”
  - IPP: Group entries of same hardware device/Printer Application
- **“Add Printer” dialog**
  - Discover non-driverless printers
  - Search for both classic drivers and Printer Applications

Print Dialogs: Direct adaptation

- Print queues are usually **temporary**, for **discovered IPP services** (IPP printers or Printer Applications)
  - Some print dialogs still use **stone-old CUPS APIs**, not supporting temporary queues, and temporary queues exist for years
  - **GTK dialog** has this fixed
  - But applications with **too old GTK versions** still around
  - **cups-browsed** used as workaround, making all queues permanent, so be careful, some dialogs do well due to cups-browsed
- On CUPS 3.x there are **no PPD files at all**
  - Dialogs should not try to download the printer's PPD from CUPS. The APIs or URLs will go away with CUPS 3.x
  - Use **modern CUPS convenience APIs** or **IPP** to get capabilities/options

Print Dialogs: The problem

- To control printing, GUI applications use **print dialogs**
- **Many different print dialogs**, usually from the GUI toolkit used (GTK, Qt, ...), but also LibreOffice, Chrome, ...
- Each one has **its own implementation** to connect to CUPS, Print-to-File, and other print technologies
- Print dialog **development does not keep up** with changes, like temporary queues in CUPS, or addition of a new print technology (cloud service, ...)
  - Printing not considered very important
  - Newly introduced print technology not considered worthwhile
  - Developers do not have time
  - Long release cycles of GUI toolkit projects vs. fast pace in printing development

Print Dialogs: The idea → CPDB

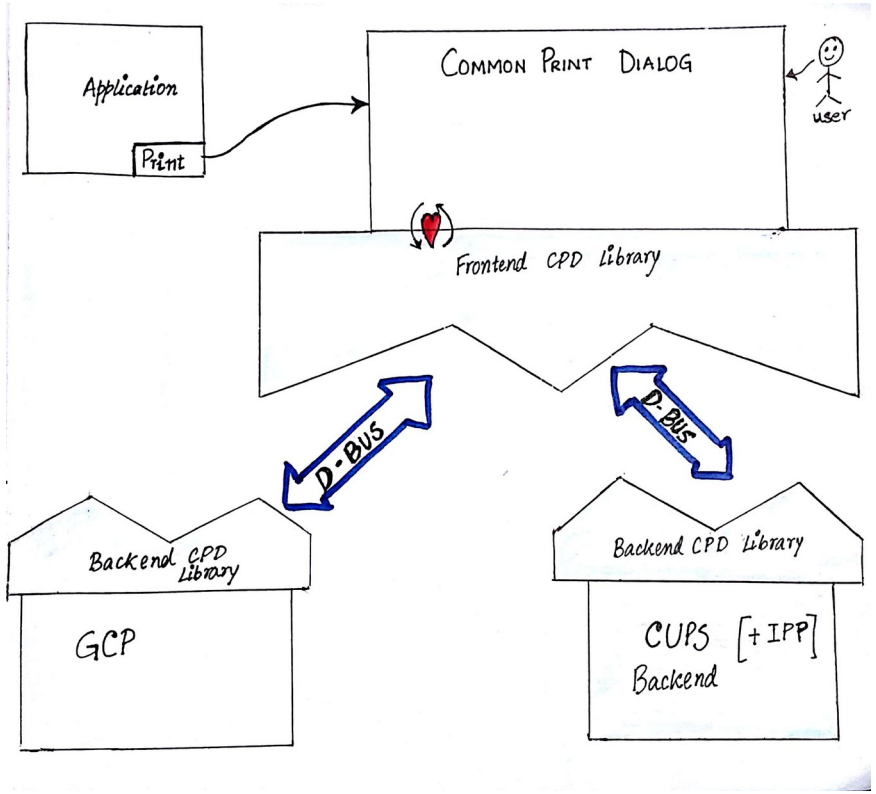
- Long time ago we tried a **Common Print Dialog**, but **failed** due to lack of human resources and/or funding (Flatpak did it finally)
- Later Aveek Basu remembered this project and **suggested a revival**, but I was unsure.
- Fixing a CUPS-related bug in the **GTK print dialog** I discovered that it uses **backends** for different print technologies
- All this brought up the idea of **Common Print Dialog Backends** in me:
  - Dialog itself still from the GUI toolkits (GTK, Qt, LibreOffice, ...)
  - GUI-independent backends for each print technology (CUPS, Print to file, ...)
  - Connection Dialog -- Backend: **D-Bus** (separately sandboxable)
  - Backend and frontend libraries

Print Dialogs: The idea → CPDB

- Backends maintained by **maintainer of print technology**
  - CUPS backend: OpenPrinting
  - GlobalCloud Print backend: GlobalCloud
  - …
- **Print dialog detects installed backends** and shows the printers of the respective print technologies
- **User sees always the same printers** with the same user-settable options in all print dialogs (GTK, Qt, LibreOffice, …)
- Print service provider can **supply backend via Snap Store**
- Maintainer of print technology changes something → He issues backend update and all print dialogs are up-to-date

Print Dialogs: CPDB - The implementation

- I posted this as a project idea in the **Google Summer of Code 2017** ...
- ... and **Nilanjana Lodh** picked it up and implemented it (her original drawing):



Print Dialogs: CPDB - The Implementation

- Libraries are on the **OpenPrinting GitHub**
  - Frontend/Backend libraries: **`cpdb-libs`**
  - CUPS backend: **`cpdb-backend-cups`**
  - Print to file backend: **`cpdb-backend-file`**
- There are also **packages in Ubuntu** (Universe, to be promoted to Main in Ubuntu 23.10)

Print Dialogs: CPDB

- This save us from problems like
  - CUPS added the new **`cupsEnumDests()`** API to **support its temporary queues** many years ago, GTK switched to it last year, **Qt (and perhaps others) did not switch yet** (needs checking).
  - The architecture of CUPS will significantly change with version 3.0 …
- In **GSoC 2022** Gaurav Guleria has added CPDB support to the **GTK** dialog and to the **Qt** dialog, merge request already **accepted in GTK**
- In **GSoC 2023** a contributor will work on the dialogs of **LibreOffice**, **Mozilla** (Firefox, Thunderbird), **Chromium Browser**
- **Modified GTK** in the **New Architecture PPA**:
  `https://launchpad.net/~till-kamppeter/+archive/ubuntu/new-arch-dev`

Demo: GNOME Control Center, Print dialogs, CPDB

- **Ubuntu Desktop 23.04** Lunar Lobster on amd64, arm64, or armhf
- Stop **cups-browsed**:
  `sudo systemctl stop cups-browsed`
- Install **GNOME Control Center**, **GTK**, and **Qt 6** from the **New Architecture PPA**:
  `https://launchpad.net/~till-kamppeter/+archive/ubuntu/new-arch-dev`
- Install **PostScript Printer Application** from the **Snap Store** to support our PostScript printer:
  `sudo snap install ps-printer-app`
- Install **focuswriter** as example for a Qt 6 app, Other options:
  `apt rdepends libqt6printsupport6`

Demo: GNOME Control Center, Print dialogs, CPDB

- Our **demo printer** supports **driverless IPP** via network or **IPP-over-USB** but is also a **classic PostScript printer**

- Activate ipp-usb: **Driverless IPP**
  `sudo systemctl stop ipp-usb; sudo systemctl disable ipp-usb`

- Deactivate ipp-usb: **Classic PostScript**
  `sudo systemctl start ipp-usb; sudo systemctl enable ipp-usb`
  - Create **2 print queues** via the web interface of the **PostScript Printer Application**:
    `https://localhost:8000/`

Demo: GNOME Control Center, Print dialogs, CPDB

- **Emulate** a driverless IPP printer via **ippeveprinter** (no hardware required):

```
ippeveprinter -s 10,10 -2 -f "image/urf,application/pdf" -d SPOOLDIR -k QUEUE
```

- The activities above **do not create any permanent CUPS queue**
- Printers show in **GNOME Control Center**
  - Button to open **web administration interface**
  - Queues of **PostScript Printer Application** grouped
- Printers show in **print dialogs**
  - **Job IPP attributes** (not PPD options) can be controlled via the options, so no PPD file data gets polled by the dialog/CPDB.

Questions / Discussion / Ask me Anything