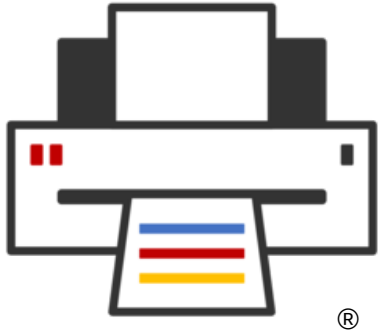




OpenPrinting

We make printing just work!

2001 – 2026: 25 years of OpenPrinting!

Till Kamppeter – OpenPrinting

Opportunity Open Source 4.0, August 29, 2026

OpenPrinting Funding



- **Laid off from Canonical**
 - June 2025 (Notice on Engineering Sprint May 2025)
 - Termination pay for further 3 months
- **Sovereign Tech Agency**
 - Sovereign Tech Fund Investment Oct 2025 – Dec 2026
- **Linux Foundation: OpenPrinting Governance and Foundation**
 - Started working on that already some years ago but intensified on it after lay-off
 - Foundation to accept sponsorships/memberships/donations
 - Charters for Foundation and Technical Charter
 - Needs to get finalized

What we are doing



- Development of printing **architecture, technology, infrastructure** on Linux, Unix, and POSIX-style operating systems
- Collaboration with **IEEE-ISTO Printer Working Group (PWG)** especially on the **Internet Printing Protocol (IPP)**
- **Driverless scanning** support in cooperation with SANE
- Integration of printing with the **desktop environment**, esp. print dialogs and printer setup tools
- Integration in **all types of operating systems** (classic DEB, RPM, .../immutable) with **all packaging methods** (Snap, Docker, ...)
- **Printer/driver compatibility look-up lists** (Driverless printers, Foomatic)
- Helping **manufacturers** on printer driver design and integration

What we are doing



- **Maintaining the components**
 - CUPS
 - cups-filters
 - ipp-usb (IPP over USB)
 - Common Print Dialog Backends (CPDB)
 - pappl-retrofit
- **Printer compatibility databases**
 - Foomatic
 - List of driverless printers
- **Contributions to other projects:** COSMIC, GNOME, KDE, LibreOffice, ...

Achievements



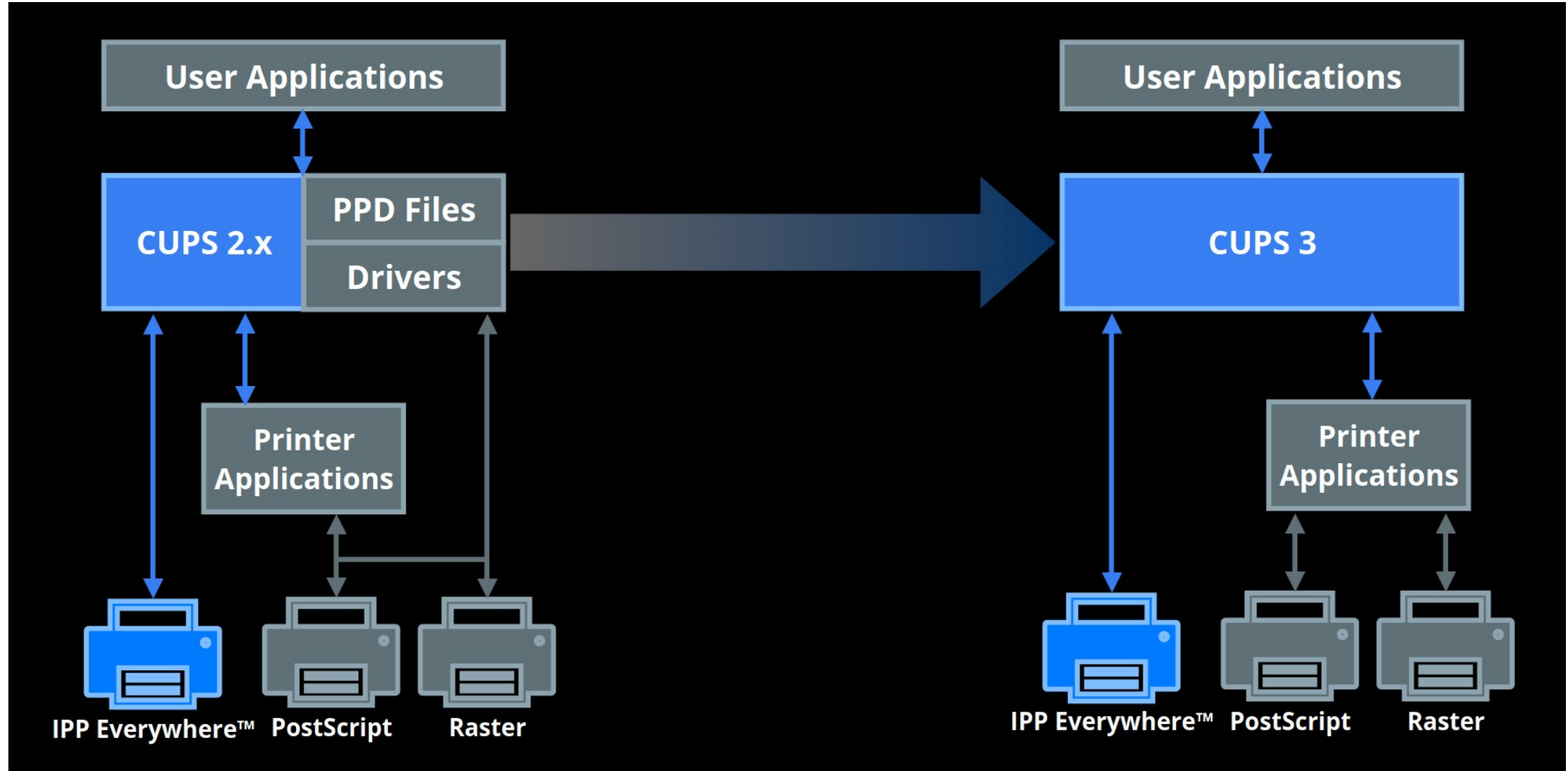
- Made **CUPS** the **standard printing system** for POSIX-style operating systems
- **All free printer drivers work with CUPS**
- Switched **standard print job format** from PostScript to **PDF**
- **Grand unified Ghostscript**: Merged Mike Sweet's ESP Ghostscript fork and all third-party drivers into upstream Ghostscript at Artifex.
- **system-config-printer** improvement and maintainership
- **cups-filters**: Continued filters and backends which Apple dropped
- **Common Print Dialog Backends**
- **All free printer drivers in Printer Applications**
- **Snaps** and **OCI container images** of **CUPS** and **Printer Applications**

New Architecture for Printing and Scanning



- **Modern printers are driverless IPP**
 - Auto-discoverable via DNS-SD
 - Tell full capabilities via IPP
 - Use standard job formats: PDF, PWG Raster, Apple Raster, PCLm
- **Stop** supporting **PostScript Printer Description** files (PPD, deprecated)
- **CUPS gets all-IPP**, no classic PPD/filter-based drivers (CUPS 3.x)
- Driver still needed? Use **Printer Applications**, a software **emulation of IPP printers**
- **Windows** goes similar way with “**Windows Protected Print**”, for security, but
 - **Mopria** (closed source), not CUPS
 - **No support concept for legacy printers** (But you can use WSL ...)

New Architecture for Printing and Scanning





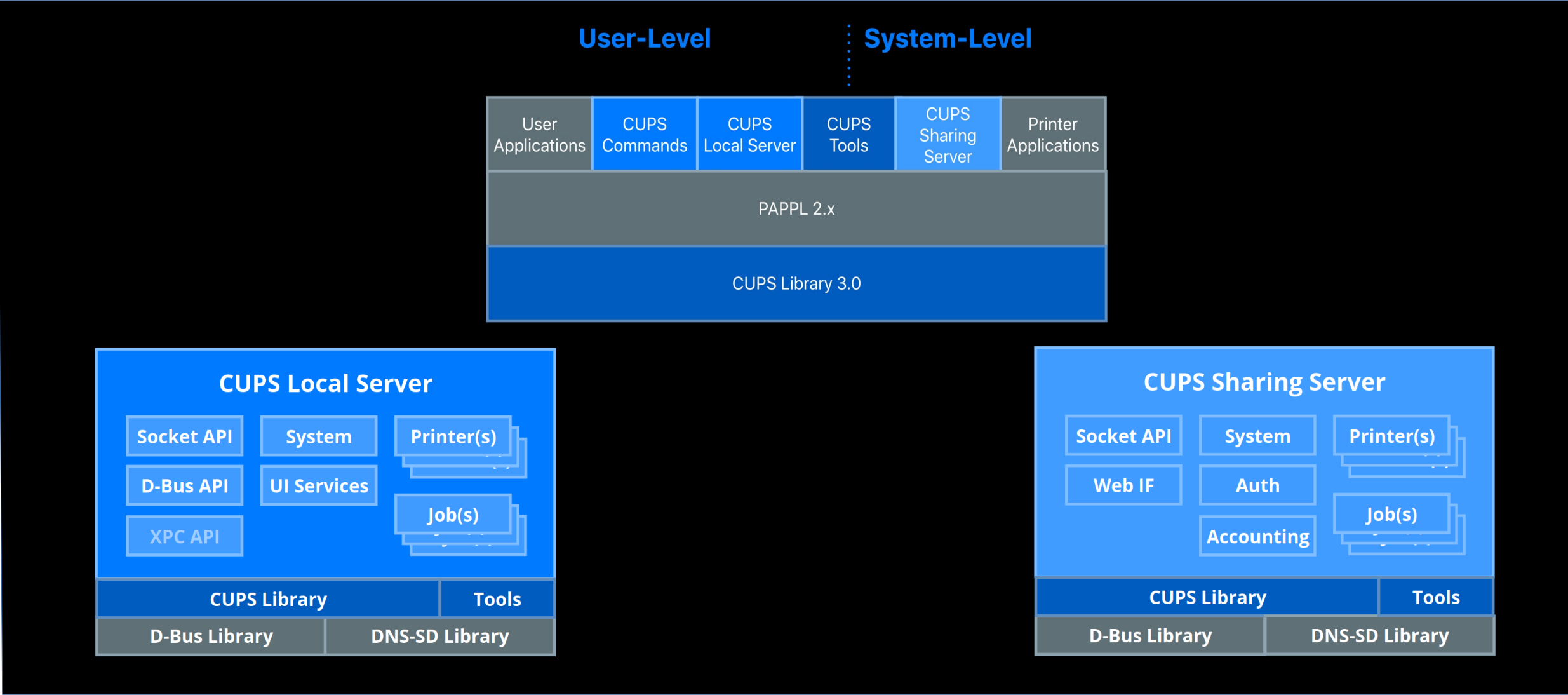
- **CUPS 2.5.x**
 - **Features**
 - **Oauth 2.0/OpenID** authentication
 - Discovery: **“Wide-area” DNS-SD** (DNS-SD over traditional DNS)
 - Localization: **Join Weblate, multi-language PPDs** for driverless IPP
 - **"job-sheets-col"** (banner pages on specific media using standard formats)
 - Security: **X.509 certificate** management improvements
 - **Web interface**: Modernization/"face lift"
 - **API backports from libcups3**: HTML form, IPP file, JSON/JWT, X.509 certificates
 - **Beta1 release**: Somewhere in the coming weeks



- **CUPS 3.x**
 - **Features**
 - **Modularization**
 - **libcups3** + CUPS tools (ipptool, ...) → **Stable release already: 3.0.1**
 - **Local server** + command line utilities (lp, lpstat, ...)
 - **Sharing server**
 - **All-IPP – New Architecture**
 - => **Printer Applications** required for legacy printers
 - To be **removed**: PPD support, Kerberos support
 - All components to be released up to **January/February 2027**
 - Switchover in possible in e. g. **Ubuntu 27.10** (27.04 very close after release)



- **CUPS 3.x**
 - **Local server**
 - Handles **local print requests** for desktop/mobile devices
 - Only **temporary IPP Everywhere print queues**
 - **Runs as user**
 - **UNIX domain socket** (D-Bus to poll socket path)
 - **Sharing server**
 - Handles **network print requests** and local printing on headless servers
 - Full print **accounting/ACLs/pre-processing** of documents
 - **OAuth 2** and **PAM**-based authentication/authorization
 - **IPP Shared Infrastructure Extensions/System** support



Printer APPLication framework – PAPPL



- **Simple libcups-based C framework/library** for developing **Printer Applications**
 - Also used to provide the **CUPS 3.x local and sharing servers**
 - Will be the **standard implementation of IPP server** in open source
- Supports **JPEG, PNG, PWG Raster, Apple Raster**, and "raw" printing
- **PAPPL 2.0 (end-2026)** - Features and Improvements:
 - **Authentication:** OAuth 2.0/OpenID
 - **Cloud Printing:** IPP INFRA and SYSTEM support
 - **Discovery:** More DNS-SD options (**local-only, disabled**)
 - Print: `finishings`, `job-password`, `job-release-action`, `job-sheets/-col`
 - **Multiple-document jobs, PDF** and **plain text** support via `ipptransform`
 - Other: **Idle-exit/on-demand** support, increased limits

Printer Setup Tools vs. CUPS 3.x



- **Main Window**
 - List **all IPP print services** as reported by DNS-SD
 - List Printer Applications and their queues or printer/fax in a group
 - No duplicates for IPv4/IPv6, IPPS, interfaces
 - Buttons for web interface, add new queue, show jobs ...
- **Add Printer Wizard**
 - List of discovered **non-driverless** USB/network printers
 - Button to see list of **Printer Applications** supporting the printer, installed ones and available in Snap Store (look-up service on OpenPrinting)
 - Buttons to setup printer with selected Printer Application and to install Printer Application from Snap Store
- **Do not remove support for permanent CUPS queues and classic drivers**

KDE Print Manager – Volunteer contributor → 2 times GSoC



- **KDE Print Manager** is printing part of **KDE Settings**
- **KDE Developer Mike Noe** opened feature request for **general improvements on Print Manager**
 - **KDE Developer Nate Graham** commented about **CUPS 3.x needs**
 - **I added detailed comment** on what is needed with **reference to GNOME and Mohit Verma's work**
- I also had **longer private mail thread with Mike Noe**
- **Mike Noe posted feature request for CUPS 3.0 support**
- **Tarun Srivastava looked for volunteer task at OpenPrinting, I directed him to KDE Print Manager**
- **Now he is enthusiastically on it, in GSoC 2025 and 2026 ...**



- **COSMIC: Desktop written completely in Rust**, for Pop! OS by System 76
- Ubuntu Summit 2024: **Carl Richell, CEO of System 76, presents COSMIC**
- **GSoC 2025**: I posted projects for **Rust bindings for libcups and cpdb-libs**, with COSMIC in mind, Michael Murphy (COSMIC developer at System 76) as mentor
- Mintu Gogoi (libcups) and Titiksha Bansal (cpdb-libs) **complete projects successfully**
- **I present the outcome to Carl Richell and he supports me**
- **GSoC 2026**: I posted **COSMIC Printer Setup Tool** and **COSMIC Print Dialog**, in collaboration with System 76
- Carl includes printing support in **COSMIC roadmap for Epoch 2 and 3**
- Abdelrahman Khalifa (Setup), Ayush Chauhan (Dialog) **enthusiastically on it ...**

Print Dialogs – CPDB, preview, and portals



- **Current print dialogs use obsolete CUPS API**
 - Do not support temporary queues for IPP print destinations, require manual queues, not supported any more by CUPS 3.x
 - Do not just use new CUPS API, but **Common Print Dialog Backends (CPDB)**, so that CUPS support is in separate backend, for future API changes
- **Modernize print dialogs**
 - **GNOME/KDE** print dialogs have no **embedded preview**, but LibreOffice, Firefox, Thunderbird, Chromium have; **COSMIC**: Design with preview right away ...
 - Print dialogs get **moved from GUI toolkits (GTK/Qt) to XDG Desktop Portal**, for better sandboxed packaging support, but **portal API does not support preview → Started discussion with all stakeholders**

Testing & Security: Continuous Integration (CI)



- **Tests to be automatically run on every GIT commit or Pull Request**
- To **discover regressions** caused by each change
- Triggered via **GitHub workflows**
- **Tests done**
 - **Static analysis** with CodeQL, GitHub's semantic code analysis engine
 - **Static analysis** with cppcheck, C/C++ code analysis
 - **Build tests** on x86-64, arm64, armhf, riscv64, for libcups 2.4.x, 2.5.x, 3.x
 - Run **unit tests** (test of compiled source, “make check”) on each architecture
 - Run **autopkgtests** of Debian/Ubuntu (test of installed package)
- On GitHub tests on x86-64 and arm64 run natively, for armhf and riscv64 emulations are used

Testing & Security:

Continuous Integration (CI)



- **CUPS, PAPPL, PDFio, ...** (projects maintained by Michael Sweet) had these CI tests already from the beginning on (but with Coverity instead of cppcheck)
- **CI addition** to OpenPrinting projects in **Winter of Code 5.0 (2026)**:
 - **libcupsfilters, cups-filters** (Rohit Kumar)
 - **cpdb-libs** (Bishal Barman)
 - **cpdb-backend-cups** (Souptik De)
- **CI addition** to OpenPrinting projects during **GSoC 2026**:
 - **libcupsfilters, cups-filters, cups-snap, pappl-retrofit, cups-browsed, cpdb-libs, cpdb-backend-cups** (Rohit Kumar)
 - **libppd** (Nayanjyoti Das, Rohit Kumar)
 - **libcups 3.x and 2.5.x** support in **libcupsfilters, cups-filters, libppd, cups-snap, pappl-retrofit, cups-browsed, cpdb-backend-cups** (Rohit Kumar)

Testing & Security: Continuous Integration (CI)



- **Special attention for cups-snap** (Rohit Kumar)
 - **Snap build test not working for emulated architectures** armhf and riscv64
 - **armhf** Snap build works remotely **on Launchpad**
 - **riscv64** needs hours/days on Launchpad, therefore no build test in CI but **check when riscv64 build appears in Snap Store** via cron job, test it, and report asynchronously
- **CI completion** by Rohit Kumar in **GSoC 2026**:
 - Add CI to **all remaining repos**
 - **Native, stand-alone C unit tests** for libppd, pappl-retrofit, and others
 - DESTDIR staging architecture to **upstreamize the autopkgtests of the Debian/Ubuntu packages** (on all repos)
 - Add support for build/use with **CUPS 2.4.x, 2.5.x, 3.x** and **PAPPL 1.4.x, 2.x**

Testing & Security: Google OSS-Fuzz



- Many bugs are not found by the usual **manual testing**, **CI/unit testing**, or **bug reports**
- **Fuzz testing** (“fuzzing”): Continued running of tests with **randomly modified input** to find unexpected behavior (aka as **bugs**)
- **Requires**
 - **Test harnesses**: Methods to call the software exercising as much as possible of it (XX% coverage), can be generated by AI/LLM (OSS-Fuzz-Gen)
 - **Input generator**: Generates randomized input which does not get trivially rejected
 - **High-performance cluster** for permanent test running (Google OSS-Fuzz)

Testing & Security: Google OSS-Fuzz



- OpenPrinting is **supplying an essential part of the ecosystem** and many orgs use OpenPrinting's work, software included **in all major OS distributions**
- 20 of 37 repos written in **memory-unsafe languages (C/C++)**
- Oxidizing (converting to **Rust**) **too much work and bug-prone**
- **OpenPrinting joined OSS-Fuzz** via GSoC 2024, continuing in GSoC 2025, 2026
- Projects: **cups, libcups, cups-filters, libcupsfilters, libcups3, ipp-usb, ...**
- **Success:** 41 issues reported with 21 resolved, 5000+ LoC changes in 2024
- From unit testing to fuzz harness
- Adapting various test input formats (e.g., **IPP** Protocol, **PDF** file, and **argv** input)
- **AI/LLM-based** harness generator **OSS-Fuzz-Gen**

Testing & Security: Security Audit



- OpenPrinting got **security-audited by 7A Security**
- Sponsored by the **Sovereign Tech Agency**
- Audit done in the following **work packages**:
 - WP1: OpenPrinting CUPS 2.5 Core & Admin APIs Audit
 - WP2: OpenPrinting Filters & Parsing Libraries Audit
 - WP3: OpenPrinting Discovery, USB & CPDB Audit
 - WP4: OpenPrinting Supply Chain & Release Process Review
 - WP5: OpenPrinting Lightweight Threat Model
- **6-head audit team** dedicated total of **77.35 working days**
- **Detailed 125-page report** of all the results

Testing & Security: Security Audit



- In WP1-WP3 **29 bugs/vulnerabilities** were found and **1 hardening recommendation**
- Each bug got **described in detail** with code snippets and **suggestions how to fix**, so we can give them as **assignments to newcomers**
- Most can be just treated as normal bugs (no CVE required), reported issues, then fixed, and the component released
- In WP4 it was investigated which security standards and methods we are using when managing our code base on GitHub, using the **Supply-chain Levels for Software Artifacts (SLSA)**
- In WP5 **threat scenarios** and recommendations for protection against them are shown
- **Thanks a lot to the Sovereign Tech Agency and 7A Security**

Further projects



- **Visual analysis of filter/print output** (GSoC 2025)
 - Check text content (OCR), margins, colors, resolution, ...
 - **Automated testing**, like CI or fuzzing also detects **functional problems**, not only crashes and errors
 - Inspired by **GNOME's openQA**
- **Emulation of real-life IPP multi-function device**
 - Not only emulating to IPP standards but also **emulating quirks**
 - Testing of **quirk handling**
 - Reproducing **user-reported bugs**

Further projects



- **Porting CUPS and Printer Applications to Zephyr RTOS** (GSoC 2025, 2026)
 - Printing from IoT devices
 - Cheap upcycling of legacy printers to driverless IPP
- **Rust bindings for libcups and CPDB** (GSoC 2025)
 - Printing support for desktop apps written in Rust, and also for the COSMIC desktop
- **Scanner Application Support in PAPPL**
 - Emulation of driverless Scanner (eSCL and/or IPP Scan) for scanner drivers in sandboxed/containerized environments

Get involved



- **Code, bug triage, CI testing, packaging, documentation, web site**
- **Google Summer of Code: Contributor, Mentor:**
<https://github.com/LinuxFoundationGSoC/ProjectIdeas/wiki/GSoC-2026-OpenPrinting>
- OpenPrinting **GitHub**: <https://github.com/OpenPrinting/>
Michael Sweet's GitHub: <https://github.com/michaelsweet/>
- OpenPrinting **mailing list**: printing-architecture@lists.linux.dev
<https://lore.kernel.org/printing-architecture/>
<https://subspace.kernel.org/lists.linux.dev.html>
- **OpenPrinting News**: <https://openprinting.github.io/news>
- **LinkedIn**: <https://www.linkedin.com/company/openprinting/>
- **Mastodon**: #OpenPrinting