24 Monate Immutable Desktop

Ein Erfahrungsbericht

Jan Wenzel / GONICUS GmbH

Since 2001

Linux & OSS Dienstleister

> 75 Mitarbeiter

Arnsberg im Sauerland (NRW)

Ausgangslage

- IT-Dienstleister
- Mitarbeiter haben Arbeitsgeräte (Notebooks) 🤯
- Notebooks sollten flexibel einsetzbar sein (Field Service)
- Linux-Dienstleister → Linux Notebooks

Ausgangslage

- Ubuntu LTS auf allen Notebooks (mit root-Rechten aka 🤠)
- Updates werden individuell installiert
- Regelmässig Problemsituationen, z.B.
  „Python deinstalliert“
  “Berechtigungen überschrieben“
  “etc.“

Quo Vadis?

- Root-Rechte wegnehmen → problematisch
- Sudo-Regeln selektiv steuern → für Notebooks quasi unendlicher Aufwand
- Remote Desktop → Keine Möglichkeit, ohne Netzverbindung zu arbeiten (Sauerland, Deutsche Bahn)
- Immutable Desktop → Relativ neu aber faszinierend

→ Evaluierung / Test

Immutable Desktop

- Updates / Upgrades durch Neustart
- Benutzer kann nicht Kernkomponenten entfernen
- Zentraler Bau der Images / Nutzung vorhander Infrastruktur:
  Gitlab
  Harbor
  Cloud-Native

  → Fehler fallen bereits während des Builds auf:
  Abhängigkeitsprobleme, Compile-Probleme, etc.
- Falls doch Probleme: Rollback auf vorherigen Stand

Immutable Desktop

- Container Image wird direkt gebootet
- Schreibbare Bereiche sind u.a. /etc und /var (/var/home)
- BTRFS
- /usr, /, usw. sind Read-Only (immutable)
- Unterstützung durch GRUB2 (Auswahl beim Booten)

Quo Vadis?

Entscheidung für Immutable Desktop

Erster Schritt: Technischer Durchstich

Manuelle Installation

Manuelle Anpassungen:

- Pakete
- Konfigurationen

Erste User (~5)

Erste User sammeln gute Erfahrungen und erstellen Bug Reports

Nächster Schritt:

Manuellen Aufwand reduzieren → Automatisierung

Immutable Desktop - Erstellung

Automatische Anpassung, Imageerstellung

Silverblue mit Makefile → OSTree image

Erste Iteration:

git-Workflow mit submodules

nachgelagerte Repos mit Gnome/KDE-Anpassungen

→ kompliziert

Wunsch: Mit Cloud-Native Technologien OCI-kompatible Images

Fedora ci-test Repo:

https://gitlab.com/fedora/ostree/ci-test

Gitlab-Pipeline + docker build, Push nach Image Registry

https://quay.io/organization/fedora-ostree-desktops

Universal Blue

Universal Blue setzt auf genau diese Images auf

Test von ublue-os sehr überzeugend

→ Adaption der Methoden

Immutable Desktop

- Gitlab CI/CD-Pipeline (docker build)
- Anpassung deklarativ durch yaml
- Module konsumieren yaml
  - Dateien
  - Flatpaks
  - RPMs (add und remove)
  - Fonts
  - Skripte
  - systemd Units (User + System)

Immutable Desktop

CI/CD-Pipeline → Merge Requests

Klassischer Workflow:

Topic Branch → Merge Request → Image zum Testen

Immutable Desktop

1. Installation von Fedora Silverblue von USB-Stick
2. Rebase auf GONICUS Desktop

Besser: Anaconda (Fedora Installer) / Kickstart:
Unterstützung von *ostree* und *ostreecontainer*

→ PXE-basierte Installation über Foreman

Cosign? https://github.com/sigstore/cosign

„*Code signing and transparency for containers and binaries*“

→ Signieren von Images durch Gitlab CI/CD

→ Container Policy

```
"registry/go-desktop": [{
            "type": "sigstoreSigned",
            "keyPath": "/etc/pki/containers/go-desktop.pub",
            "signedIdentity": {
                "type": "matchRepository"
```

Immutable Desktop

Wichtigstes Feature: A/B-Updates

Updates/Upgrades als Transaktion im Hintergrund

Einfacher Rollback möglich

Integration in Bootloader (Auswahl des Images beim Booten)

Pinning von Versionen

Updates


```
$ rpm-ostree status

State: idle

AutomaticUpdates: stage; rpm-ostreed-automatic.timer: last run 17min ago

Deployments:

  ostree-image-signed:docker://<REGISTRY_URL>/<PROJECT>/silverblue:41

                   Version: 41.20250320.0 (2025-03-20T14:09:23Z)

● ostree-image-signed:docker://<REGISTRY_URL>/<PROJECT>/silverblue:41

                   Version: 41.20250320.0 (2025-03-20T05:10:23Z)
```

Immutable Desktop

- ABER: Komplett neue Welt: Sandboxed Apps, Container-CLI
- Beispiele:

  “Wie läuft mein Chromium?“

  “Wie installiere ich Desktop-Erweiterungen?“

  “Wie installiere ich die Programme, die ich brauche?“

  “Wie kann ich Software XYZ testen?“

Immutable Desktop

Zentrale Antwort: *Kommt drauf an.*

Entweder: Container - distrobox/toolbx (funktioniert auch mit grafischen Anwendungen)

Oder: Sandboxed (Flatpak / AppImage)

Auch möglich: (rpm-ostree install / systemd-sysext / dnf5)

Immutable Desktop

Beispiel: Firefox

Variante 1: In Desktop integriert (wie upstream)

Bei Fedora per default keine Codecs
Sicherheit gegenüber Sandbox?

Variante 2: Flatpak

Codecs, Keine Gnome-Erweiterungen, KeepassXC?

Firefox als Flatpak

Beispiel für eine Lösung um KeepassXC und Firefox zu verbinden:

systemd-Unit, die Pfad überwacht (inotify) und umschreibt (Pfad zum Binary in Flatpak)

Extra-Berechtigung für Firefox, in KeepassXC-Sandbox Kommando auszuführen

Anwendungen

Beispiel LibreOffice:

Früher: verschiedene Versionen (verzögerte Updates, verschiedene Ubuntu-Releases → PPA)

Heute: Flatpak für alle (automatische Updates)

Welches Problem wird gelöst?

- Isolierte Arbeits- und Entwicklungsumgebung
- Spezifische Software / unterschiedliche Versionen
- Unterschiedliche Distributionen
- $HOME, Sockets (SSH, GPG, Wayland, Kerberos, etc.), D-Bus, etc.
- Alles in einem Container (podman, docker)

  → nicht nur für Immutable Desktops interessant

Distrobox / Toolbx

- Einfacher Export von grafischen Anwendungen
- Deklarative Erstellung von CLI-Containern über ini-File (distrobox assemble)

```
[ansible]
image=registry.fedoraproject.org/fedora-toolbox:41
additional_packages="git bash-completion vim python3-pip"
init_hooks="pip3 install ansible-core==2.17.*;"
```

Distrobox / Toolbx

- Stärker auf Fedora / RedHat ausgerichtet
- Deklarative Erstellung von Containern über Containerfile

```
FROM registry.fedoraproject.org/fedora-toolbox:41

RUN dnf -y install git bash-completion vim python3-pip

RUN pip3 install ansible-core==2.17.*
```

Probleme

- Mit Fedora 42 wurde auf composefs um, Rollback auf 41 nicht möglich
- OSTree kaputt
- Flatpak-Integration teilweise nicht optimal (Drucker, Kerberos, etc.), keine Unterstützung von Portalen

Ausblick

Hohes tägliches Download-Volumen (~1.6GB)

→ Mögliche Lösung: rechunk

https://github.com/hhd-dev/rechunk

Zerteilt Image und baut es anhand der geänderten Pakete (changelog) wieder optimiert zusammen

Idealfall: Nur neue/aktualisierte Pakete werden heruntergeladen

Weiterführende Links

- Universal Blue: Immutable OS, das Fedora Atomic Desktops erweitert, verschiedene Flavors (z.B. Bazzite für Gaming)
- Blue Build: Baukasten aus Universal Blue zur Erstellung eigener Immutable Distributionen: z.B. Secure Blue oder EU-OS
- Distrobox: Container als Shell
- Flathub: Registry für Flatpak-Apps
- AppImageHub: Registry für AppImage-Apps

The End

Vielen Dank!

Weitere Fragen?🤔

Jan Wenzel, Senior Technical Consultant

GONICUS GmbH

wenzel@gonicus.de