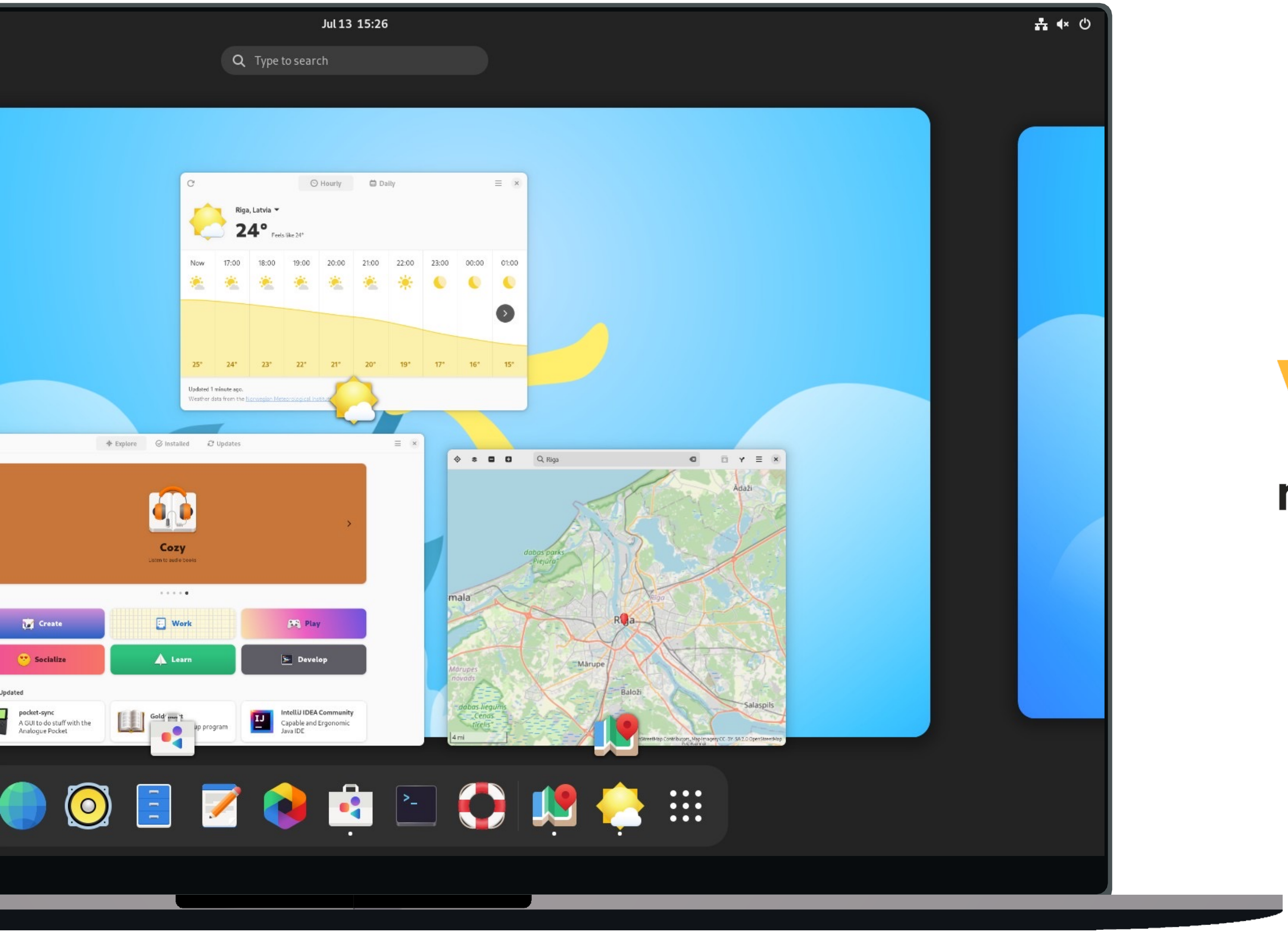


Vanilla OS is not just another distribution.

> whoami

Pietro di Caprio

Vanilla OS

- Core Developer
- Public Relations Manager

Vanilla OS today

Vanilla OS is an immutable and atomic Linux distribution based on Ubuntu, with the goal of providing a consistent and smooth user experience. Its stability and security allow users to use their devices without worries.

To date, we have partially achieved our goal.

Vanilla OS today

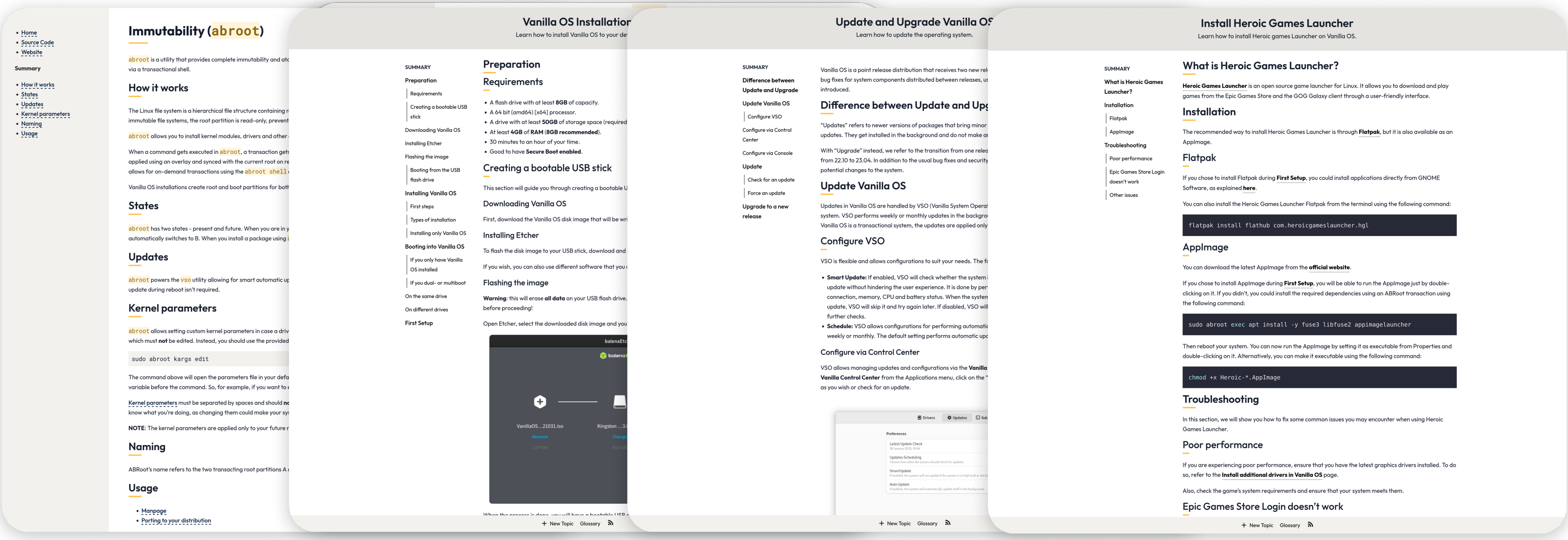
Thanks to our technologies and utilities, the user is able to use their personal devices without having to worry about updates and maintenance.

[A B] Root

APX

Vanilla OS today

And if they ever need help, we have thorough documentation and a handy handbook to help solve many common issues.



- Home
- Source Code
- Website

Summary

- How it works
- States
- Updates
- Kernel parameters
- Naming
- Usage

Immutability (abroot)

abroot is a utility that provides complete immutability and ato
via a transactional shell.

How it works

The Linux file system is a hierarchical file structure containing r
immutable file systems, the root partition is read-only, prevent

abroot allows you to install kernel modules, drivers and other

When a command gets executed in abroot, a transaction gets
applied using an overlay and synced with the current root on re
allows for on-demand transactions using the abroot shell

Vanilla OS installations create root and boot partitions for both

States

abroot has two states - present and future. When you are in y
automatically switches to B. When you install a package using

Updates

abroot powers the vso utility allowing for smart automatic up
update during reboot isn't required.

Kernel parameters

abroot allows setting custom kernel parameters in case a driv
which must **not** be edited. Instead, you should use the provided

```
sudo abroot kargs edit
```

The command above will open the parameters file in your defa
variable before the command. So, for example, if you want to e

Kernel parameters must be separated by spaces and should **no**
know what you're doing, as changing them could make your sys

NOTE: The kernel parameters are applied only to your future r

Naming

ABRoot's name refers to the two transacting root partitions A

Usage

- Manpage
- Porting to your distribution

Vanilla OS Installation

Learn how to install Vanilla OS to your de

SUMMARY

Preparation
- Requirements
- Creating a bootable USB stick

Downloading Vanilla OS

Installing Etcher

Flashing the image
- Booting from the USB flash drive

Installing Vanilla OS
- First steps
- Types of installation
- Installing only Vanilla OS

Booting into Vanilla OS
- If you only have Vanilla OS installed
- If you dual- or multiboot

On the same drive

On different drives

First Setup

Preparation

Requirements

- A flash drive with at least **8GB** of capacity.
- A 64 bit (amd64) [x64] processor.
- A drive with at least **50GB** of storage space (required
- At least **4GB** of **RAM** (**8GB recommended**).
- 30 minutes to an hour of your time.
- Good to have **Secure Boot enabled**.

Creating a bootable USB stick

This section will guide you through creating a bootable U

Downloading Vanilla OS

First, download the Vanilla OS disk image that will be wri

Installing Etcher

To flash the disk image to your USB stick, download and

If you wish, you can also use different software that you

Flashing the image

Warning: this will erase **all data** on your USB flash drive.
before proceeding!

Open Etcher, select the downloaded disk image and you

When the process is done, you will have a bootable USB

+ New Topic Glossary

Update and Upgrade Vanilla OS

Learn how to update the operating system.

SUMMARY

Difference between Update and Upgrade

Update Vanilla OS
- Configure VSO

Configure via Control Center

Configure via Console

Update
- Check for an update
- Force an update

Upgrade to a new release

Vanilla OS is a point release distribution that receives two new rel
bug fixes for system components distributed between releases, us
introduced.

Difference between Update and Upg

"Updates" refers to newer versions of packages that bring minor
updates. They get installed in the background and do not make a

With "Upgrade" instead, we refer to the transition from one relea
from 22.10 to 23.04. In addition to the usual bug fixes and security
potential changes to the system.

Update Vanilla OS

Updates in Vanilla OS are handled by VSO (Vanilla System Operat
system. VSO performs weekly or monthly updates in the backgro
Vanilla OS is a transactional system, the updates are applied only

Configure VSO

VSO is flexible and allows configurations to suit your needs. The f

- **Smart Update:** If enabled, VSO will check whether the system i
update without hindering the user experience. It is done by per
connection, memory, CPU and battery status. When the system
update, VSO will skip it and try again later. If disabled, VSO will
further checks.
- **Schedule:** VSO allows configurations for performing automatic
weekly or monthly. The default setting performs automatic upd

Configure via Control Center

VSO allows managing updates and configurations via the **Vanilla**
Vanilla Control Center from the Applications menu, click on the "
as you wish or check for an update.

Preferences

Latest Update Check
26 January 2023, 16:44

Updates Scheduling
Choose how often the system should check for updates.

SmartUpdate
If enabled, the system will not update if the system is in high load or the ba

Auto Update
If enabled, the system will automatically update itself in the background.

+ New Topic Glossary

Install Heroic Games Launcher

Learn how to install Heroic games Launcher on Vanilla OS.

SUMMARY

What is Heroic Games Launcher?

Installation
- Flatpak
- AppImage

Troubleshooting
- Poor performance
- Epic Games Store Login doesn't work
- Other issues

What is Heroic Games Launcher?

Heroic Games Launcher is an open source game launcher for Linux. It allows you to download and play games from the Epic Games Store and the GOG Galaxy client through a user-friendly interface.

Installation

The recommended way to install Heroic Games Launcher is through **Flatpak**, but it is also available as an AppImage.

Flatpak

If you chose to install Flatpak during **First Setup**, you could install applications directly from GNOME Software, as explained **here**.

You can also install the Heroic Games Launcher Flatpak from the terminal using the following command:

```
flatpak install flathub com.heroicgameslauncher.hgl
```

AppImage

You can download the latest AppImage from the **official website**.

If you chose to install AppImage during **First Setup**, you will be able to run the AppImage just by double-clicking on it. If you didn't, you could install the required dependencies using an ABRoot transaction using the following command:

```
sudo abroot exec apt install -y fuse3 libfuse2 appimagelauncher
```

Then reboot your system. You can now run the AppImage by setting it as executable from Properties and double-clicking on it. Alternatively, you can make it executable using the following command:

```
chmod +x Heroic-*.AppImage
```

Troubleshooting

In this section, we will show you how to fix some common issues you may encounter when using Heroic Games Launcher.

Poor performance

If you are experiencing poor performance, ensure that you have the latest graphics drivers installed. To do so, refer to the **Install additional drivers in Vanilla OS** page.

Also, check the game's system requirements and ensure that your system meets them.

Epic Games Store Login doesn't work

+ New Topic Glossary

Vanilla OS today

Common problems

All classic distributions are affected by

System Unreliability

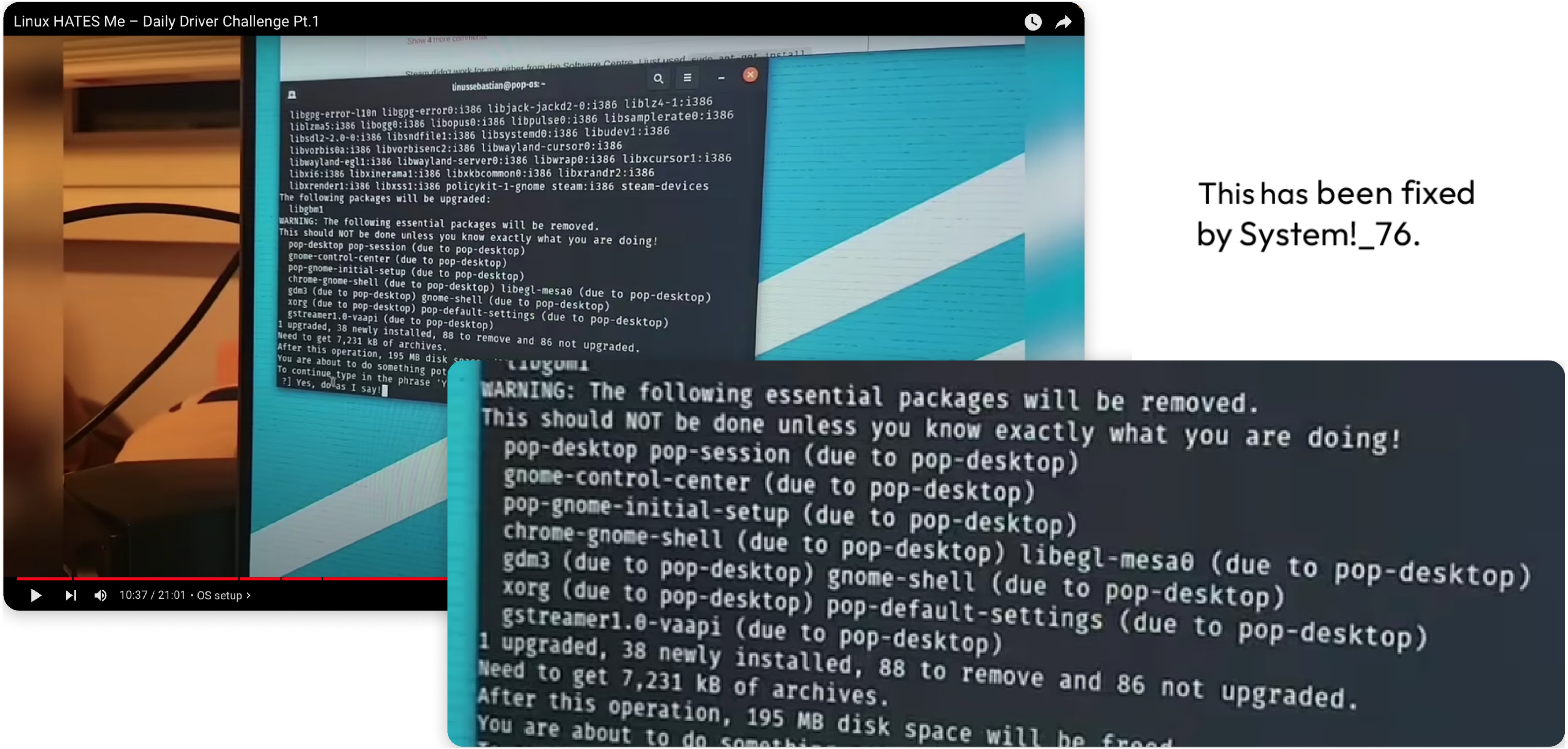
Update Inconsistency

Overfreedom

System Unreliability

The problems

Providing a system package manager results in having a high chance of conflicts, which affects the whole system. These conflicts are difficult to track and the final result can't be predicted.


Linux HATES Me – Daily Driver Challenge Pt.1
10:37 / 21:01 • OS setup ›
linussebastian@pop-os: ~

```
  libgpg-error-l10n libgpg-error0:i386 libjack-jackd2-0:i386 liblz4-1:i386
  liblzma5:i386 libogg0:i386 libopus0:i386 libpulse0:i386 libsamplerate0:i386
  libsdl2-2.0-0:i386 libsndfile1:i386 libsystemd0:i386 libudev1:i386
  libvorbis0a:i386 libvorbisenc2:i386 libwayland-cursor0:i386
  libwayland-egl1:i386 libwayland-server0:i386 libwrap0:i386 libxcursor1:i386
  libxi6:i386 libxinerama1:i386 libxkbcommon0:i386 libxrandr2:i386
  libxrender1:i386 libxss1:i386 policykit-1-gnome steam:i386 steam-devices
The following packages will be upgraded:
  libgbm1
WARNING: The following essential packages will be removed.
This should NOT be done unless you know exactly what you are doing!
  pop-desktop pop-session (due to pop-desktop)
  gnome-control-center (due to pop-desktop)
  pop-gnome-initial-setup (due to pop-desktop)
  chrome-gnome-shell (due to pop-desktop) libegl-mesa0 (due to pop-desktop)
  gdm3 (due to pop-desktop) gnome-shell (due to pop-desktop)
  xorg (due to pop-desktop) pop-default-settings (due to pop-desktop)
  gstreamer1.0-vaapi (due to pop-desktop)
1 upgraded, 38 newly installed, 88 to remove and 86 not upgraded.
Need to get 7,231 kB of archives.
After this operation, 195 MB disk space
You are about to do something pot
To continue type in the phrase 'Y
?] Yes, do as I say!
```

```
  libgbm1
WARNING: The following essential packages will be removed.
This should NOT be done unless you know exactly what you are doing!
  pop-desktop pop-session (due to pop-desktop)
  gnome-control-center (due to pop-desktop)
  pop-gnome-initial-setup (due to pop-desktop)
  chrome-gnome-shell (due to pop-desktop) libegl-mesa0 (due to pop-desktop)
  gdm3 (due to pop-desktop) gnome-shell (due to pop-desktop)
  xorg (due to pop-desktop) pop-default-settings (due to pop-desktop)
  gstreamer1.0-vaapi (due to pop-desktop)
1 upgraded, 38 newly installed, 88 to remove and 86 not upgraded.
Need to get 7,231 kB of archives.
After this operation, 195 MB disk space will be freed.
You are about to do somethi
```

This has been fixed by System!_76.

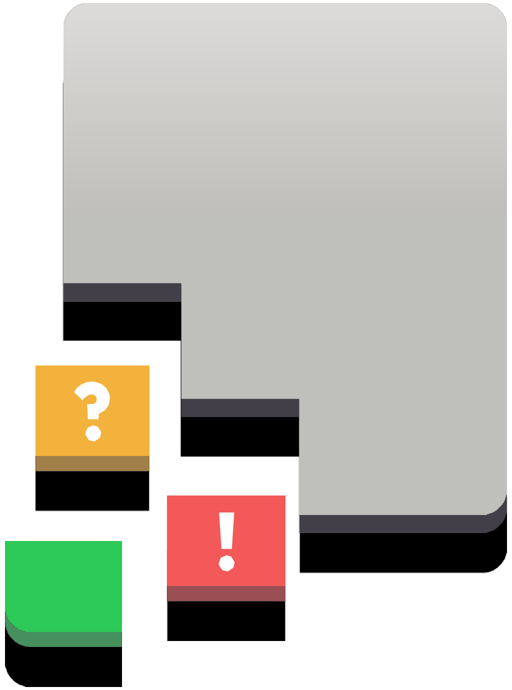
Update Inconsistency

The problems

Due to the nature of package managers, updates are the result of multiple dependencies and package collisions.

This increases the risk of obtaining different updates than those provided and intended by the developers.

The result is an experience that does not conform to the intended one.



Overfreedom

The problems

Freedom is one of the foundations of Linux and Open Source worlds.

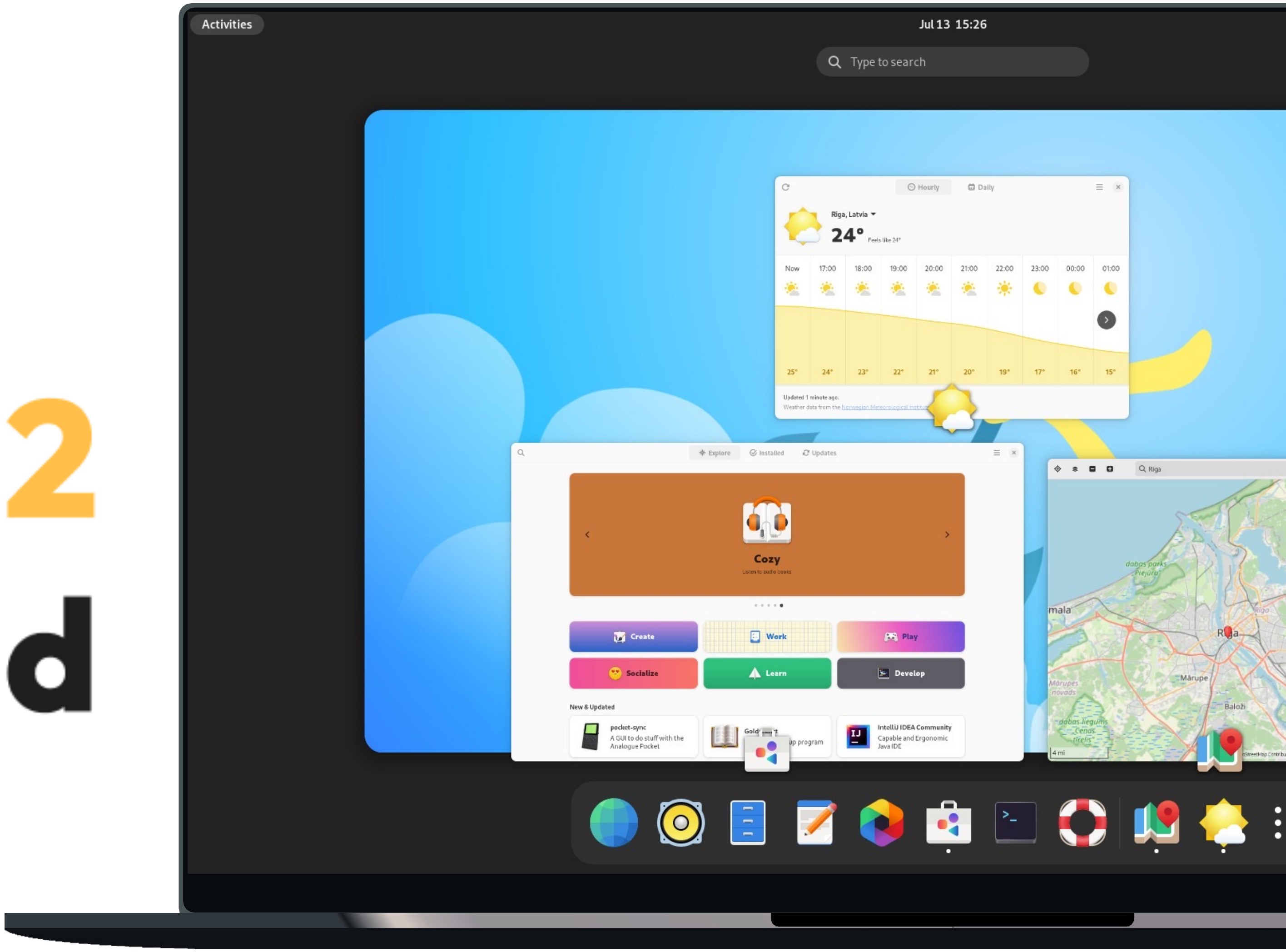
Inexperienced users can modify every aspect of their system, risking breakage.

Developers can't replicate an unknown state: we need standards.

So, what?

Vanilla OS 2

Orchid



Vanilla OS 2

Vanilla OS 2 is a complete rewrite and revision of the project.

We have transitioned from Ubuntu to Debian Sid as our base, with an internal repository freeze system, giving us the freedom to plan and release updates when they are ready.

This version is designed to meet the needs of every type of user, giving them the ability to perform any task without worrying about system stability and maintenance as it is performed automatically in the background.

Technologies and Utilities

The technologies and utilities behind Vanilla OS have grown, a lot.

Vib [A B] Root APX

Ikaros Albius Prometheus

ABRoot v2

Immutability protects the root system from unwanted changes.

Atomicity ensures that updates only reach the user’s system if they don't fail.

OCI images are used to distribute consistent, reliable and reproducible updates.

I SEE CONTAINERS
CONTAINERS EVERYWHERE

OCI Images

ABRoot v2

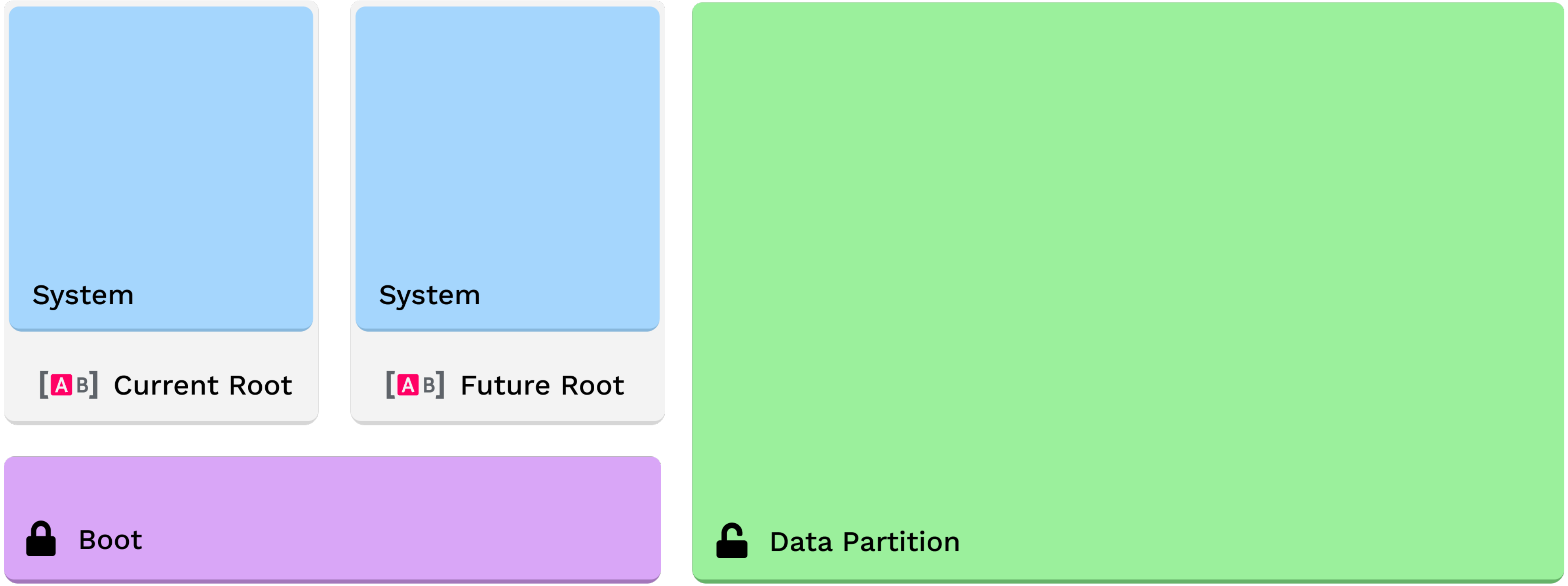
OCI (Open Container Initiative) images are widely used in cloud computing to provide production and development environments that are easy to replicate and distribute.

In ABRoot, we have used this technology to distribute system updates, thus achieving consistent and reproducible updates, as well as logistically intelligent updates thanks to data deduplication.

System structure

ABRoot v1

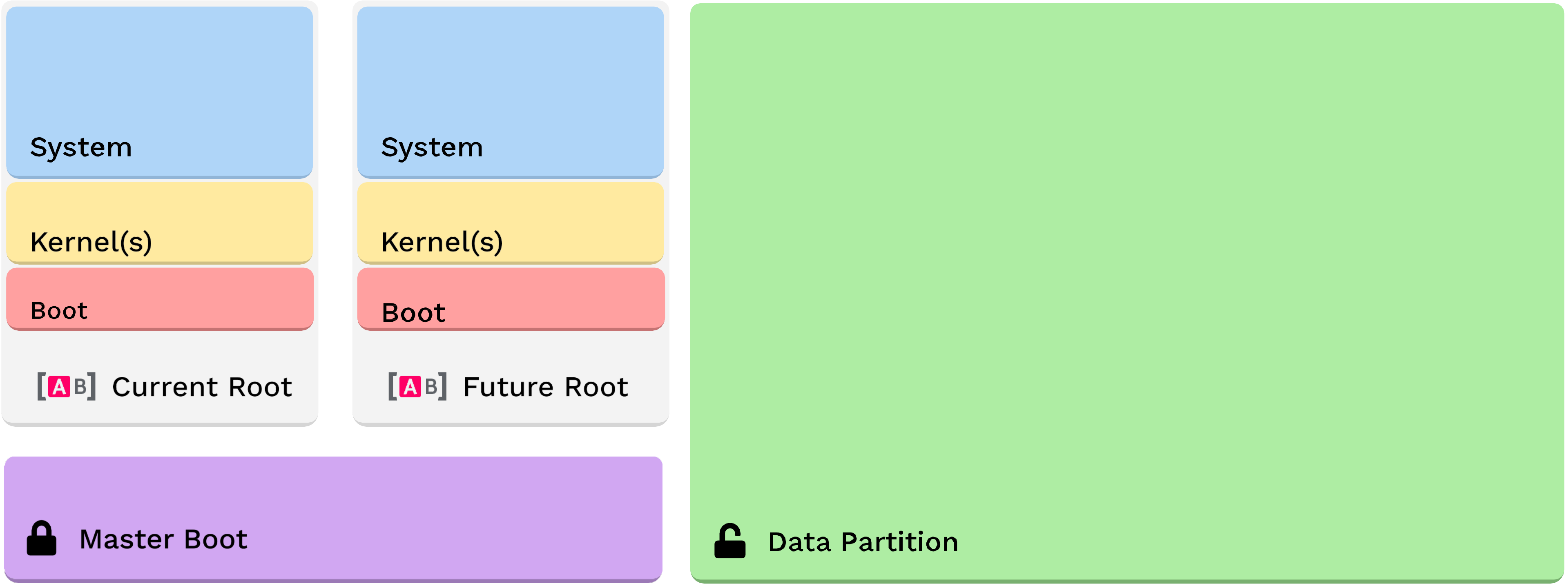
The file system structure of ABRoot is very different from other distributions.



System structure

ABRoot v2

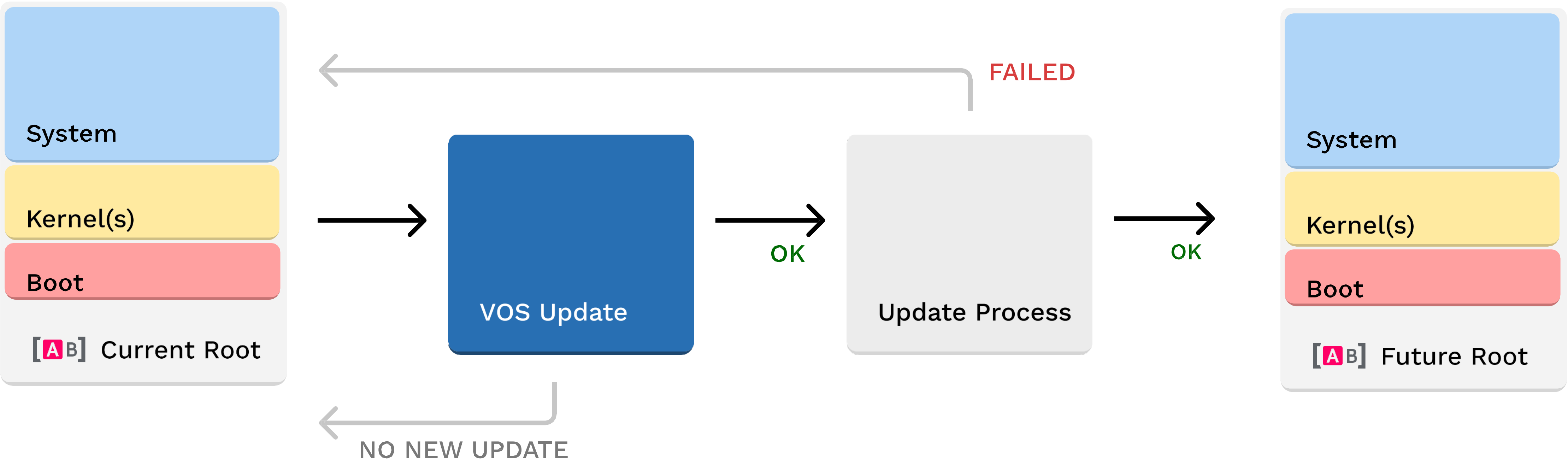
The file system structure of ABRoot 2 is even more different.



System upgrade

ABRoot v2

The system gets upgrades through the use of atomic transactions, which means that if just one stage of the upgrade fails, the whole transaction is dropped.



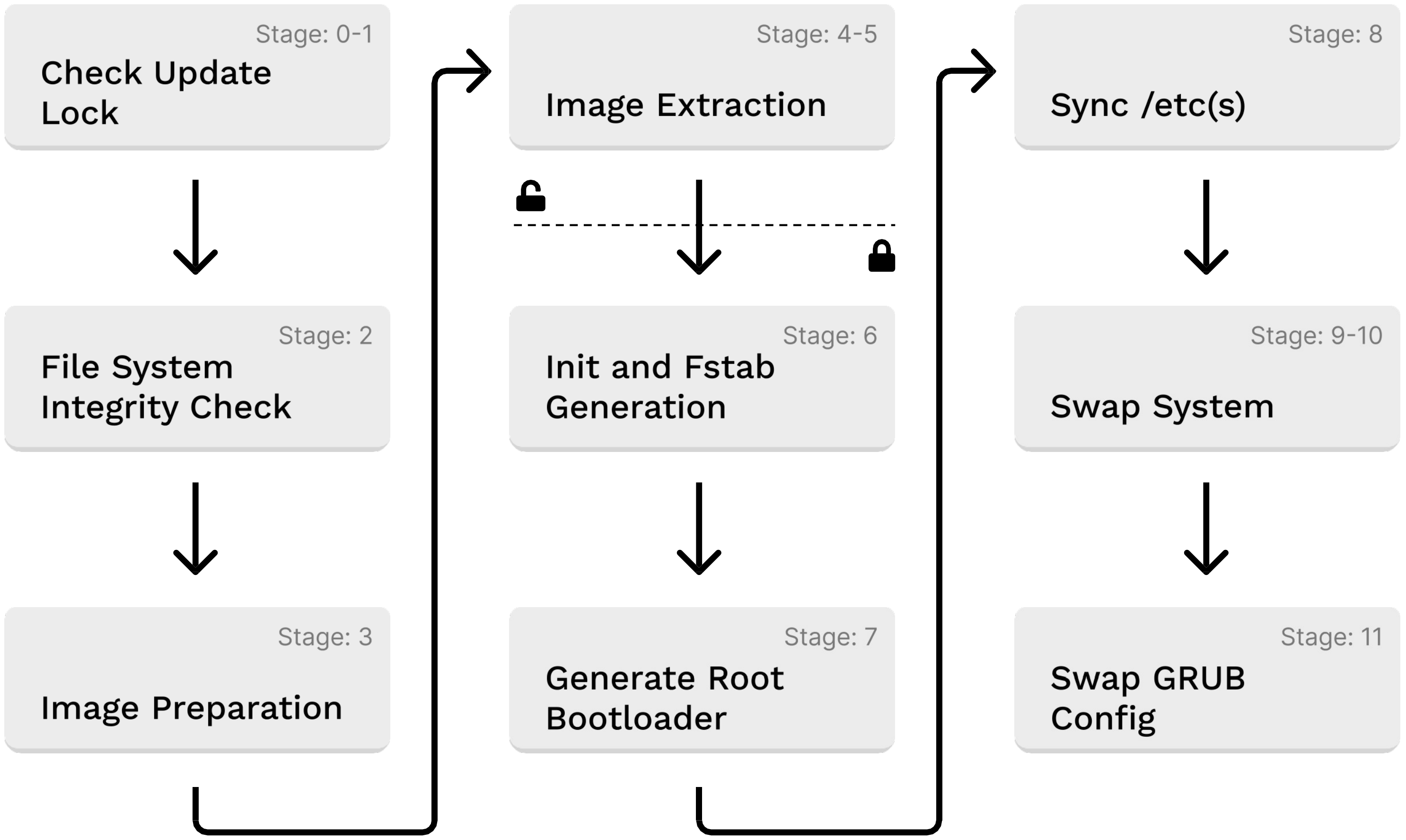
Update process

ABRoot v2

The upgrade process consists of several stages.

Each stage is critical and causes the whole update to get dropped if anything goes wrong.

The upgrade process can be safely* stopped until the 5th stage is reached.



Stage: 0-1
Check Update Lock

Stage: 2
File System Integrity Check

Stage: 3
Image Preparation

Stage: 4-5
Image Extraction

Stage: 6
Init and Fstab Generation

Stage: 7
Generate Root Bootloader

Stage: 8
Sync /etc(s)

Stage: 9-10
Swap System

Stage: 11
Swap GRUB Config

*safe as no data has been written yet. As of stage 6, the user can still stop the upgrade, but there is a risk that the root partition will be left in an unknown state while the current root partition is never touched.

Kernel Flags

ABRoot v2

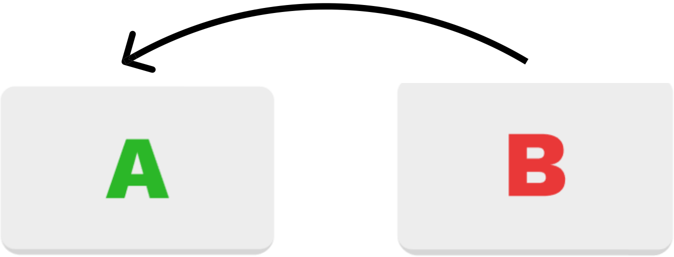
Given the complexity of the file system and its particular structure for booting, ABRoot also handles the management of kernel flags. Advanced users can use a built-in manager to add, remove, and update the variables that are passed to the kernel during the boot phase.

The flags become part of the atomic update process, ensuring that they are applied only from the future root, thus always allowing access to the previous ones.

Rollback

ABRoot v2

Accidents happen. If a driver update has a critical bug or some dependency breaks compatibility with the user's device, they can use ABRoot to easily rollback to the previous state, even if they cannot boot into their system.

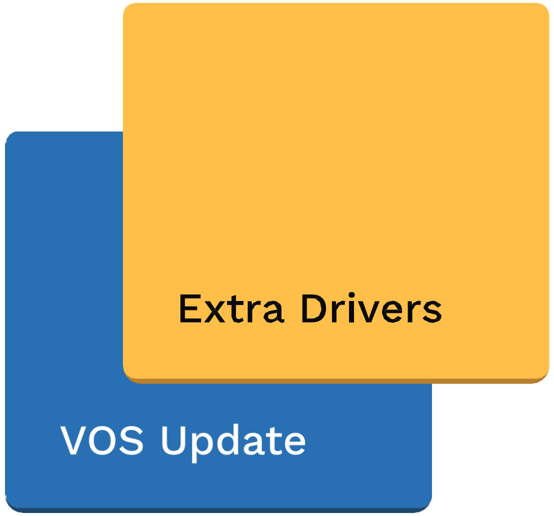


Need more drivers?

ABRoot v2

Touching the underlying system is discouraged, but drivers and firmware are essential; for this reason, we have created a kind of package manager in ABRoot that allows adding and managing packages safely.

When the user requests a change, the added packages list is used to build a local OCI image based on top of Vanilla OS; this image is then atomically kept up to date like the standard one.



Extra Drivers

VOS Update

Albius

Technologies and Utilities

Vanilla OS uses an unusual filesystem structure with OCI images. This makes using an existing installer backend a challenge.

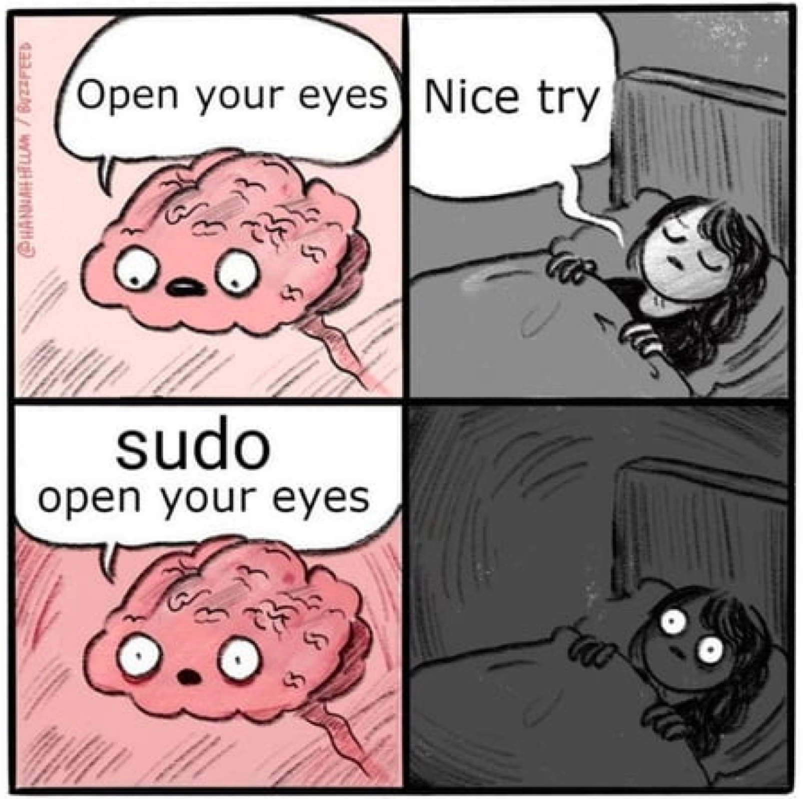
It led us to write a distro-agnostic custom installer backend called Albius. It uses recipes which define setup steps, mount points, installation options, and post-installation steps.

sudon't

By eliminating sudo, the risk of accidental or harmful operations performed by unauthorized entities would get reduced.

Since Vanilla OS promotes a secure and stable environment, we are limiting privileged access to help preserve the system’s integrity.

The absence of sudo would encourage the adoption of a more structured approach to privilege management. (e.g. polkit)



VSO and Apx

VSO and Apx were two distinct utilities in Vanilla OS 22.10.

They now share the purpose of providing the user with software, both in their own distinct ways, to cater towards both less experienced and more advanced users.

Meet their new roles.

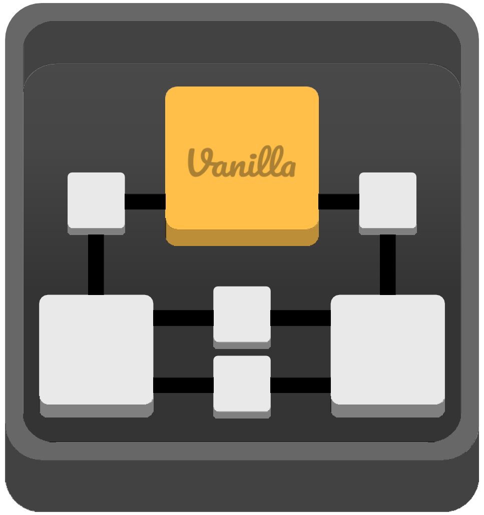
VSO v2

VSO and Apx

VSO is the meeting point between the system and the user.

It allows the user to install applications within a container that is deeply integrated with the system while remaining separated from the root partition.

The end result is a native experience, allowing the user to use Vanilla OS as if it were a conventional Linux distribution.

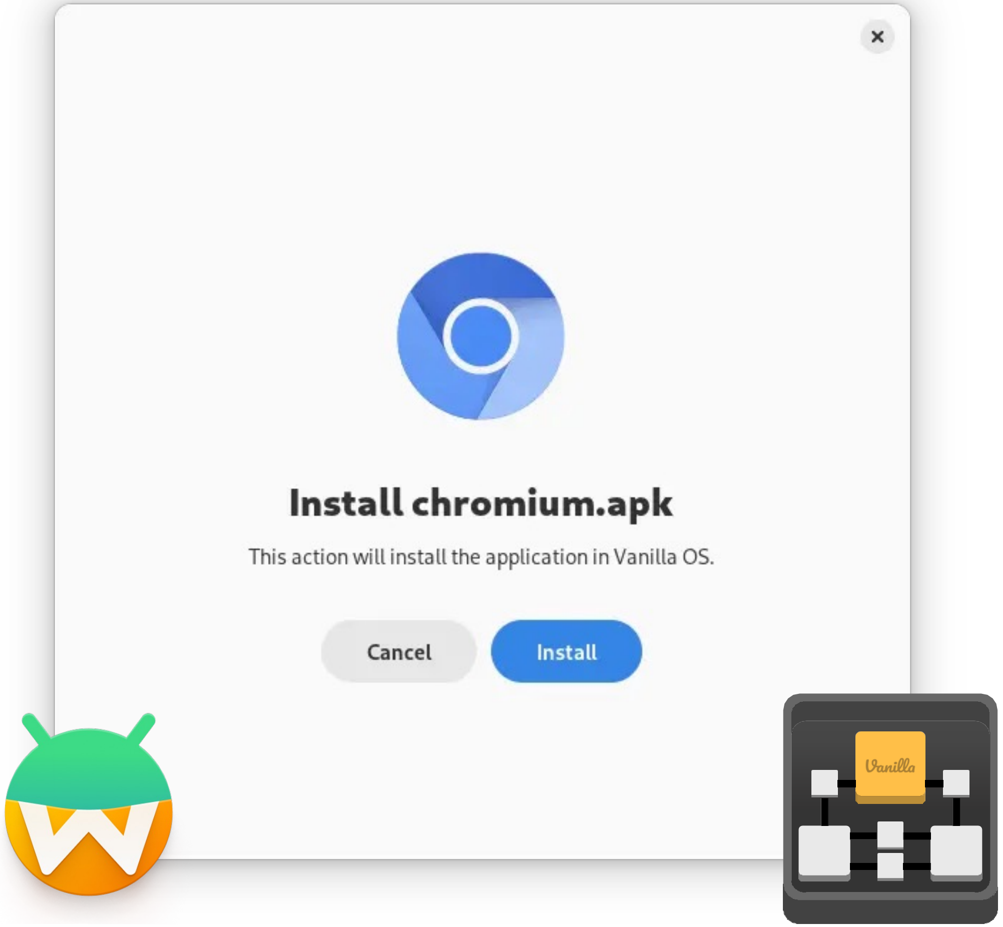


VSO - Deb and..APK?

VSO and Apx

By default, VSO installs applications through the Vanilla OS repositories or via .deb packages.

Thanks to its integration with Waydroid, VSO also supports Android applications (.apk), further expanding the software ecosystem supported by Vanilla OS.



Install chromium.apk

This action will install the application in Vanilla OS.

Apx v2

VSO and Apx

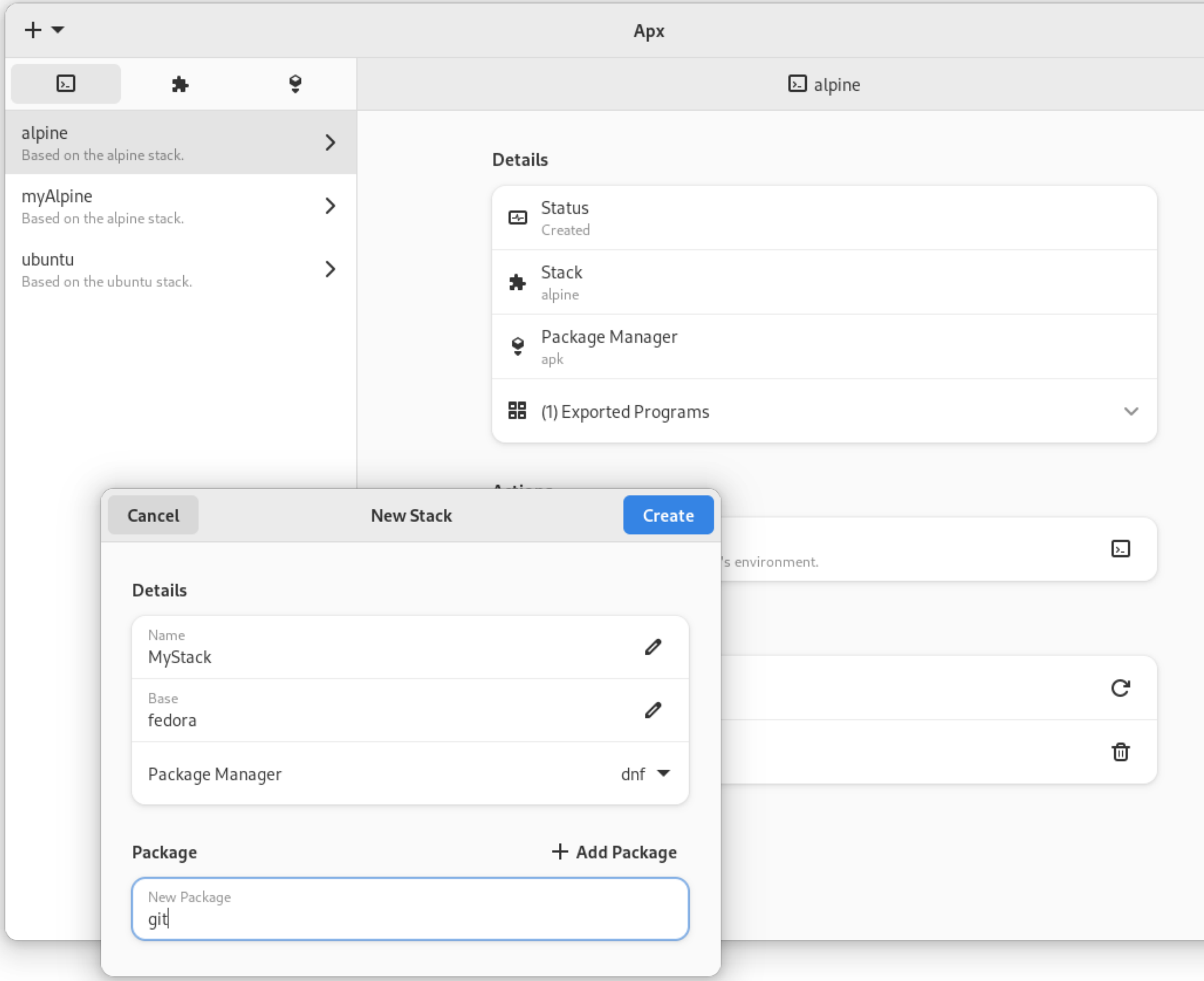
Apx builds upon Distrobox to let the user create subsystems based on any Linux distribution. These subsystems use Apx's common interface for package management (following APT structure), simplifying software management.

Users define subsystems using stacks, which specify the base distribution and optional packages to preinstall. These stacks can be made for specific purposes, such as programming, graphic design, or education.

Apx GUI

VSO and Apx

Since Apx is a feature-rich tool, we have developed the Apx GUI, an interface to manage all of your subsystems in a convenient and graphical way.



alpine
Based on the alpine stack.

myAlpine
Based on the alpine stack.

ubuntu
Based on the ubuntu stack.

Details

Status: Created
Stack: alpine
Package Manager: apk
(1) Exported Programs

New Stack – Details

Name: MyStack
Base: fedora
Package Manager: dnf

Package

New Package: git

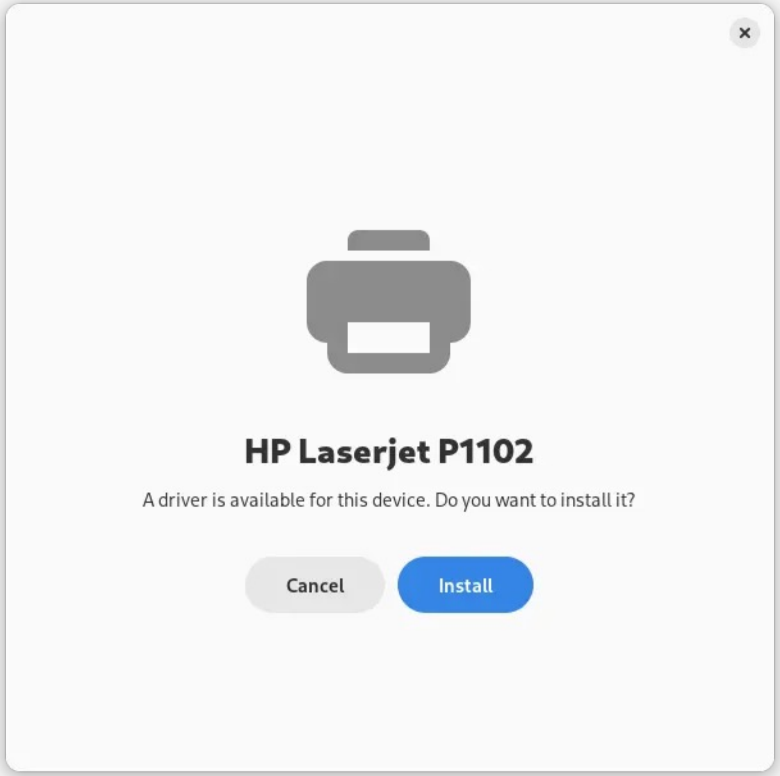
Ikaros

Technologies **and Utilities**

With the move from Ubuntu to Debian, we noticed the lack of the ubuntu-drivers-common package, or something similar.

Driver installation is often one of the least accessible parts of Linux, which is why we made Ikaros, together with a GUI, as a replacement.

Ikaros automatically detects your hardware and proposes to install the available drivers for your system.



HP Laserjet P1102

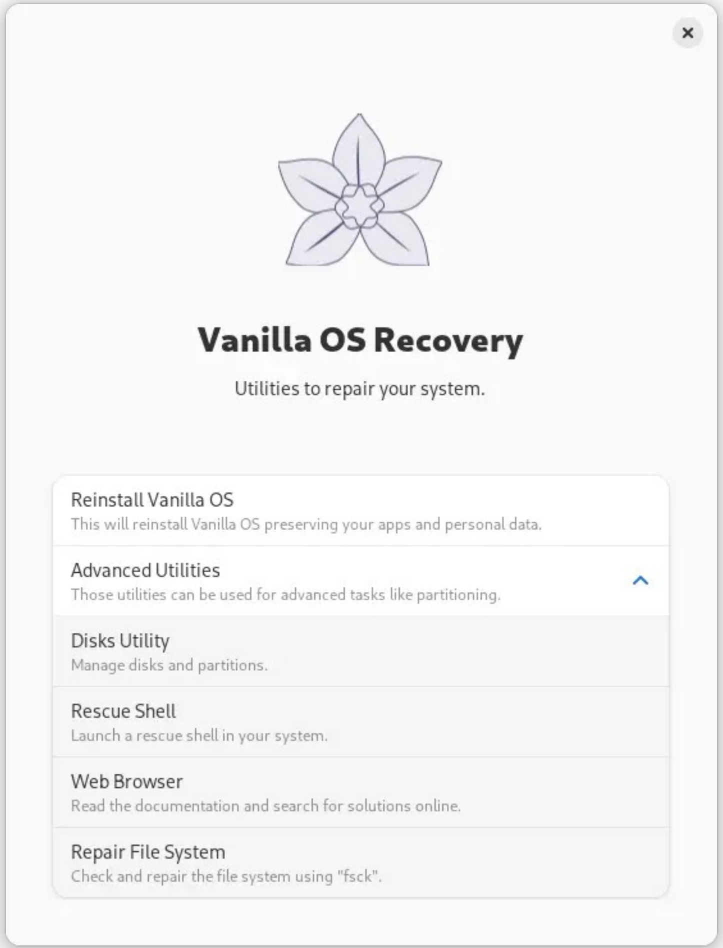
A driver is available for this device. Do you want to install it?

Recovery Utility

Technologies and Utilities

Vanilla OS is by design an extremely stable and reliable system, breaking it is extremely difficult.

Even if the system breaks through unforeseen circumstances, we have made a recovery utility to rescue your system with various tools.



Vanilla OS Recovery

Utilities to repair your system.

Reinstall Vanilla OS
This will reinstall Vanilla OS preserving your apps and personal data.

Advanced Utilities
Those utilities can be used for advanced tasks like partitioning.

Disks Utility
Manage disks and partitions.

Rescue Shell
Launch a rescue shell in your system.

Web Browser
Read the documentation and search for solutions online.

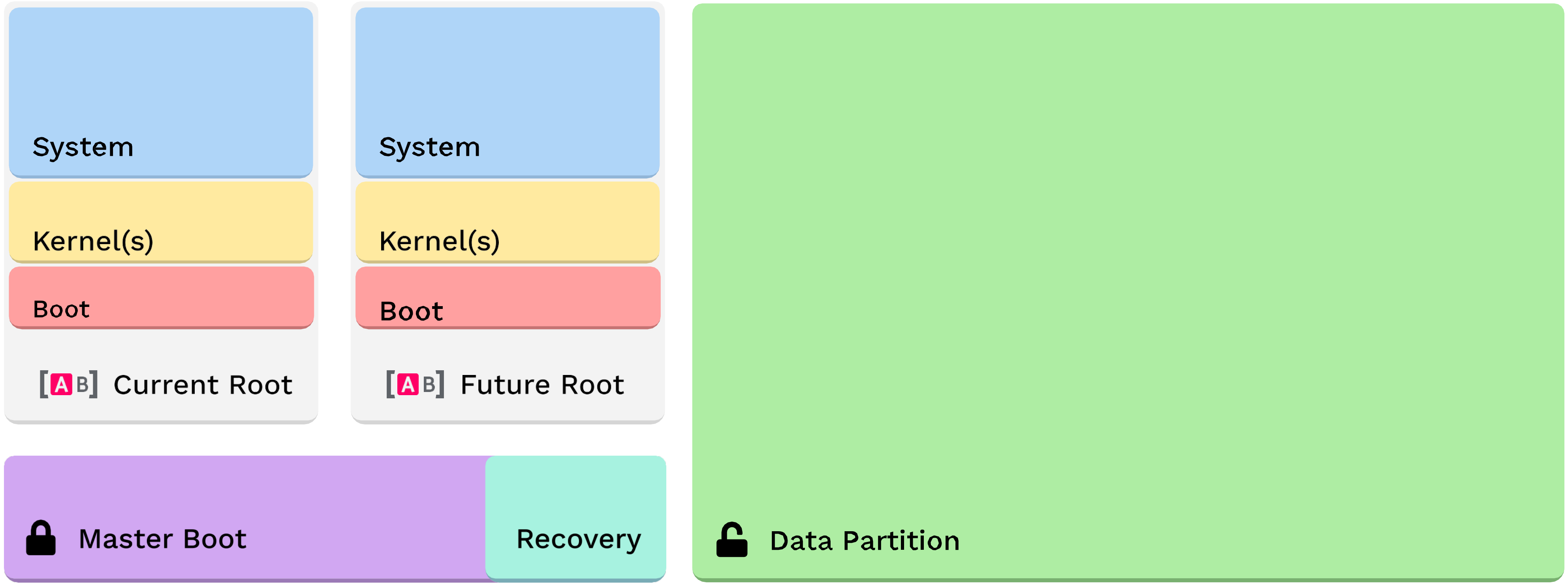
Repair File System
Check and repair the file system using "fsck".

Recovery Utility

Technologies and Utilities

The recovery is located in a small partition (~300MB)...

...or in a pen-drive.



System

Kernel(s)

Boot

[AB] Current Root

System

Kernel(s)

Boot

[AB] Future Root

Master Boot

Recovery

Data Partition

OEM

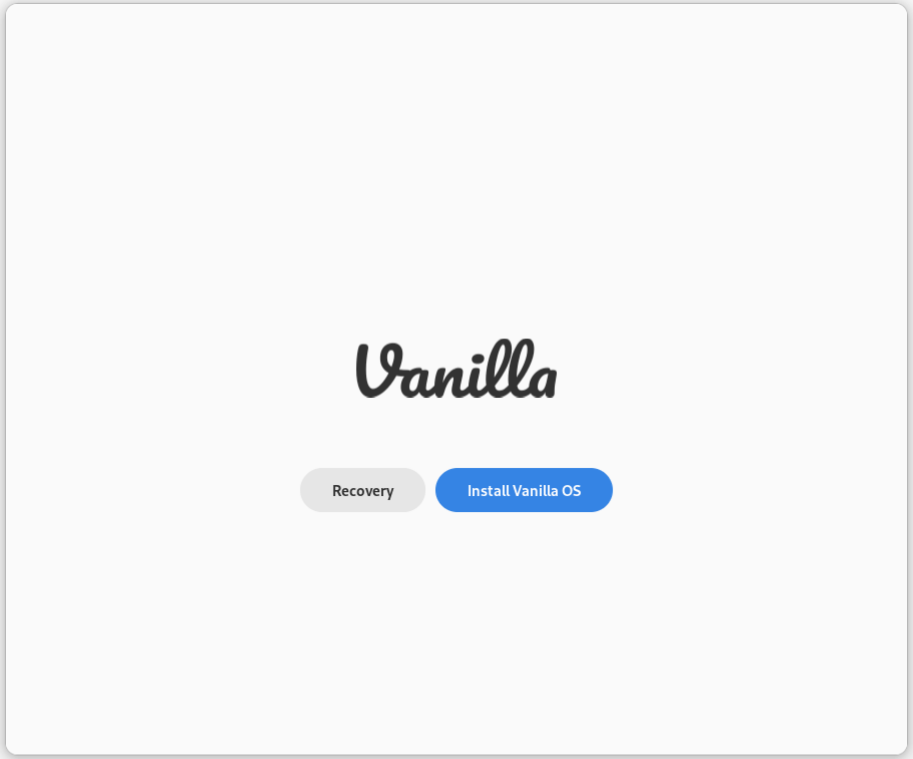
Our goal is to reach as many people as possible with Vanilla OS. This is why we put effort into making our distribution as OEM compatible as possible.

Vanilla Installer

OEM

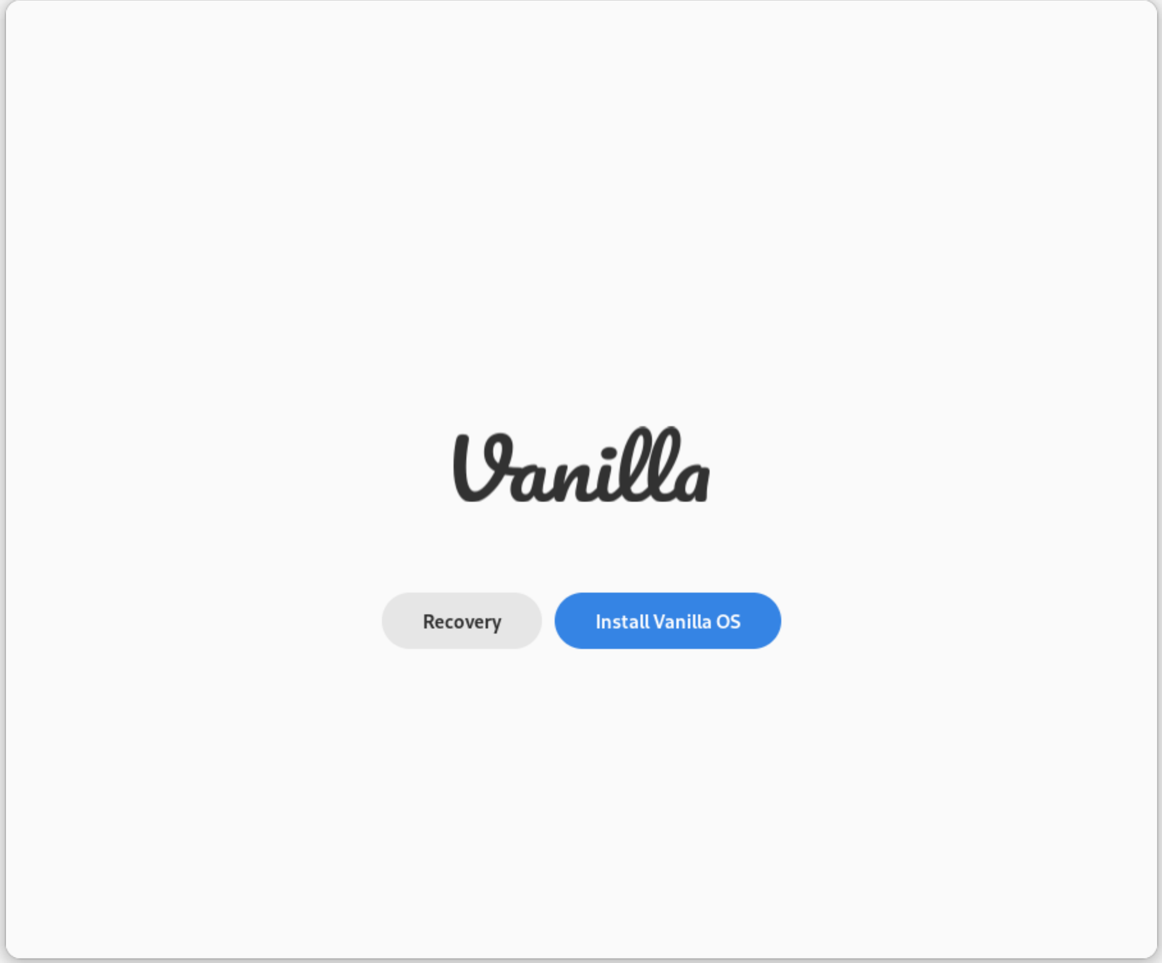
The Vanilla Installer is a GTK 4 application, utilizing Libadwaita to provide a great user experience.

The application is explicitly used to install the system.



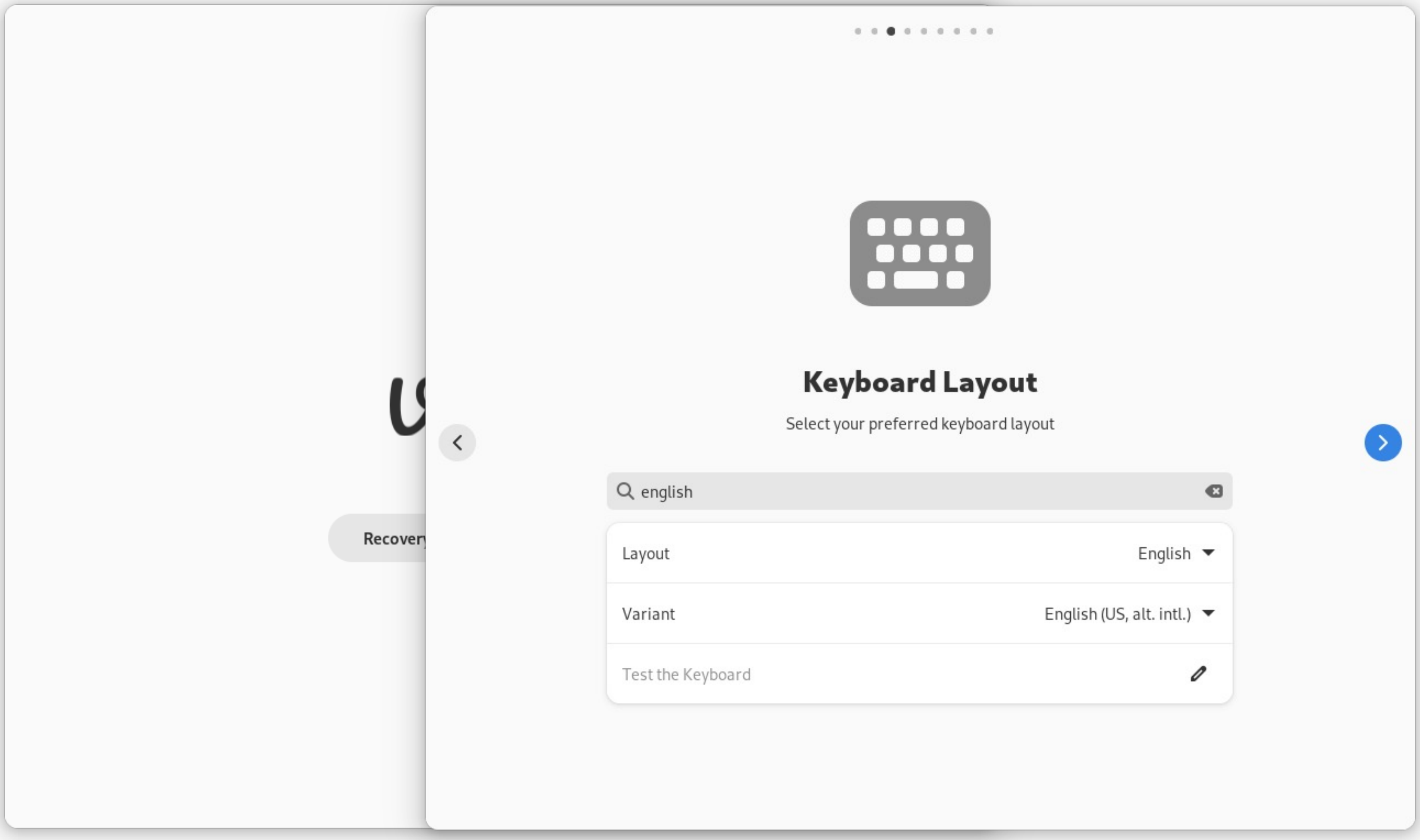
Vanilla Installer

OEM



Vanilla Installer

OEM



Keyboard Layout

Select your preferred keyboard layout

english

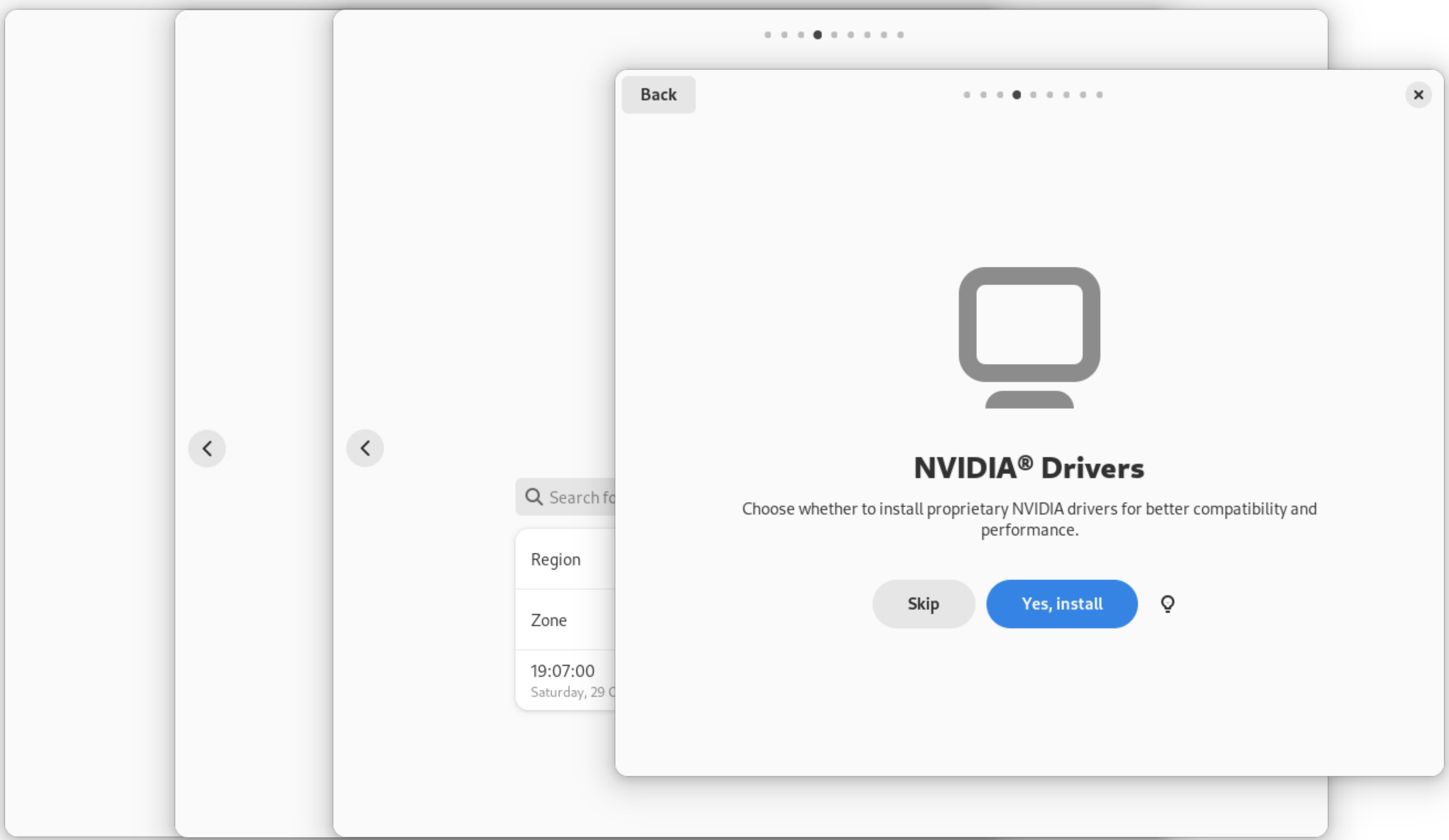
Layout: English

Variant: English (US, alt. intl.)

Test the Keyboard

Vanilla Installer

OEM



Back

NVIDIA® Drivers

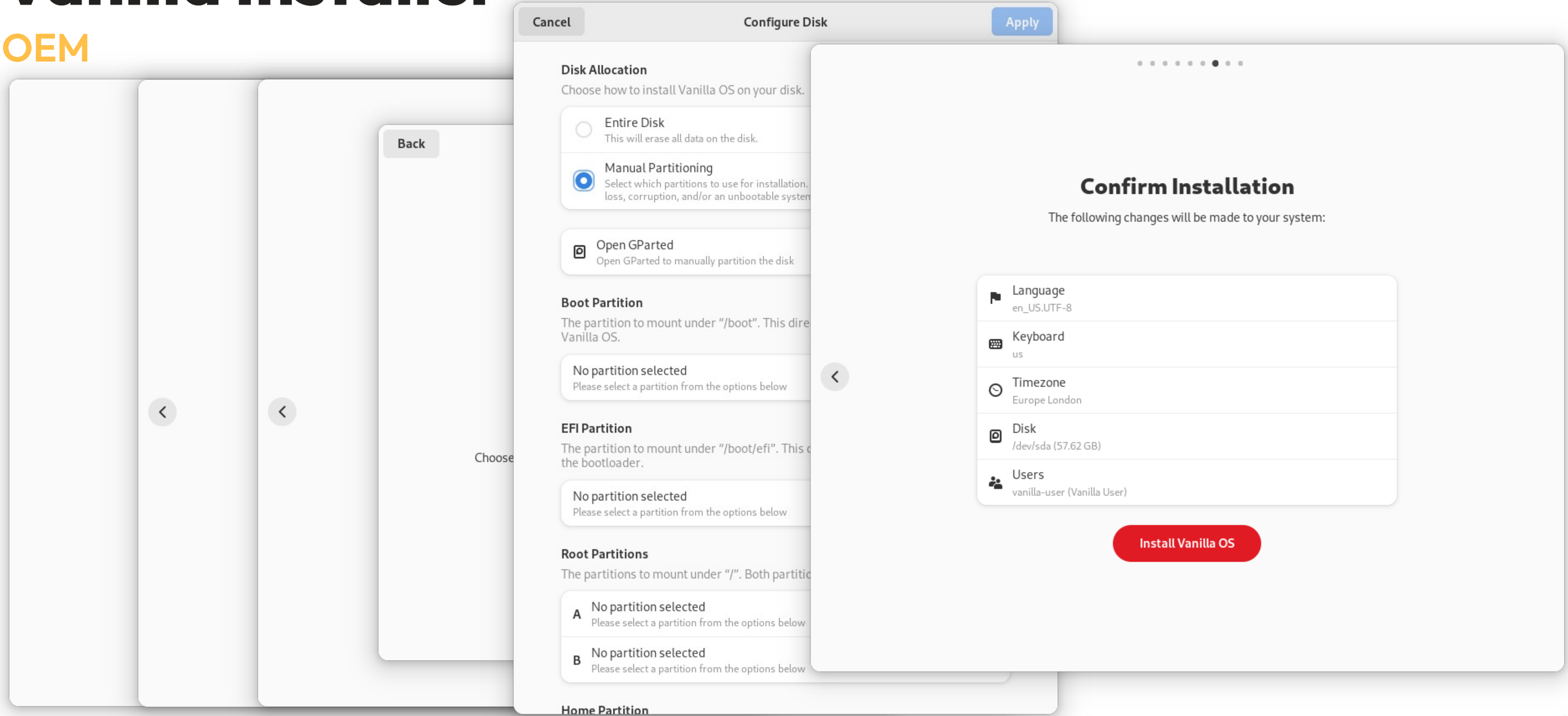
Choose whether to install proprietary NVIDIA drivers for better compatibility and performance.

Skip

Yes, install

Vanilla Installer

OEM



Back

Choose

Cancel

Configure Disk

Apply

Disk Allocation

Choose how to install Vanilla OS on your disk.

○ Entire Disk
This will erase all data on the disk.

● Manual Partitioning
Select which partitions to use for installation.
loss, corruption, and/or an unbootable system

Open GParted
Open GParted to manually partition the disk

Boot Partition

The partition to mount under "/boot". This dire
Vanilla OS.

No partition selected
Please select a partition from the options below

EFI Partition

The partition to mount under "/boot/efi". This c
the bootloader.

No partition selected
Please select a partition from the options below

Root Partitions

The partitions to mount under "/". Both partitio

A No partition selected
Please select a partition from the options below

B No partition selected
Please select a partition from the options below

Home Partition

Confirm Installation

The following changes will be made to your system:

Language
en_US.UTF-8

Keyboard
us

Timezone
Europe London

Disk
/dev/sda (57.62 GB)

Users
vanilla-user (Vanilla User)

Install Vanilla OS

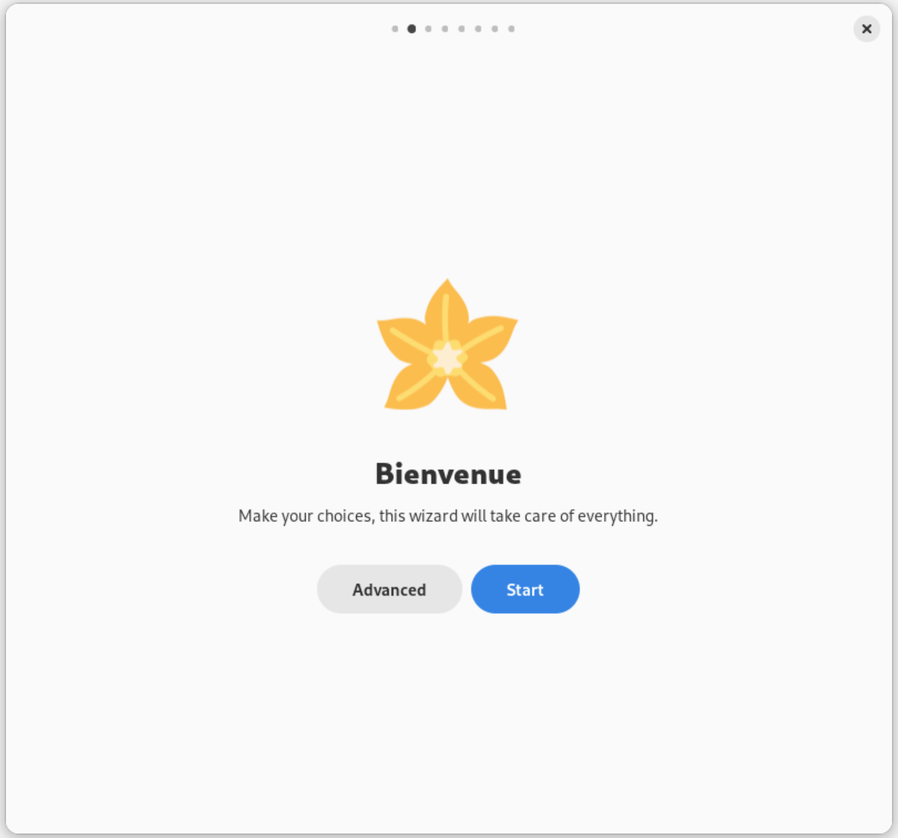
Vanilla First Setup

OEM

The First Setup is, just like the installer, a GTK 4 application utilizing Libadwaita.

The application is used to alter the initial Vanilla OS installation to the user's needs.

It contains both an express and advanced setup process, providing the best experience for both newcomers and more advanced users.

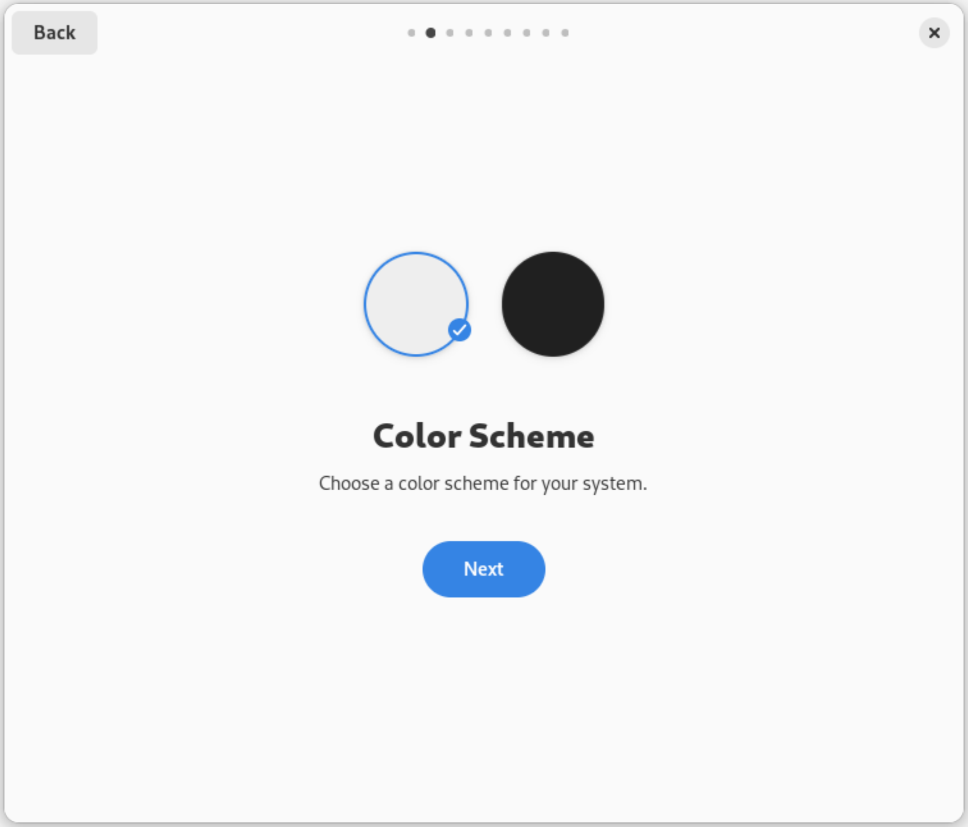


Bienvenue

Make your choices, this wizard will take care of everything.

Vanilla First Setup

OEM

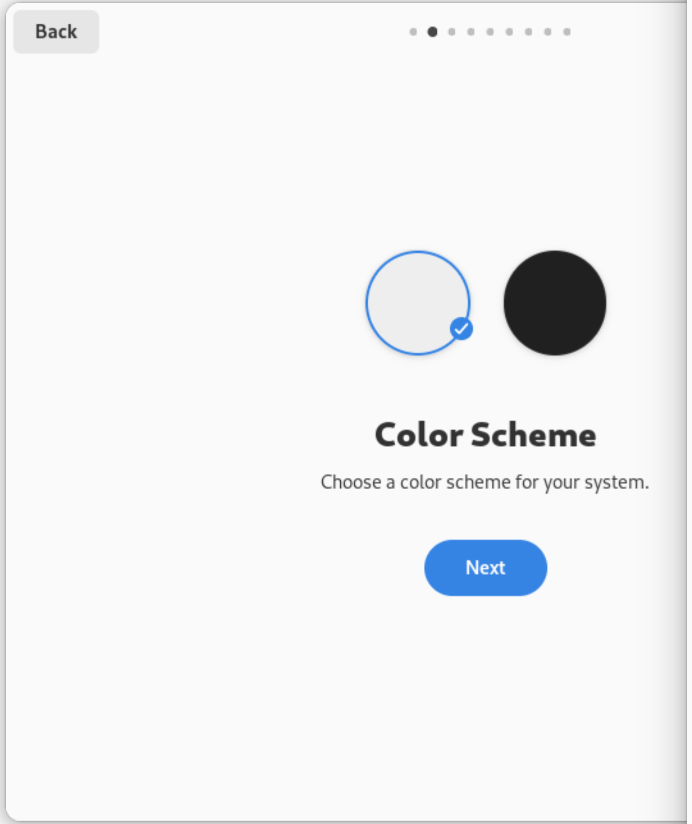


Color Scheme

Choose a color scheme for your system.

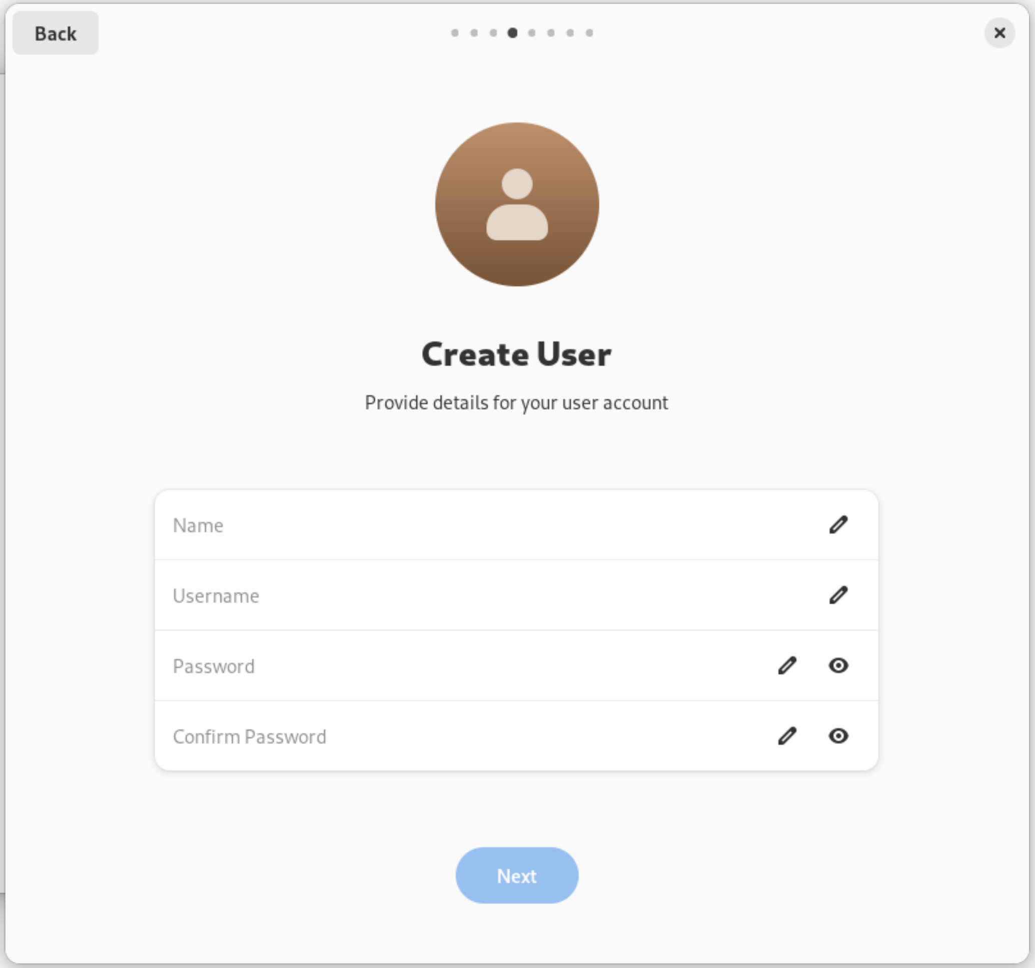
Vanilla First Setup

OEM



Color Scheme

Choose a color scheme for your system.



Create User

Provide details for your user account

Name

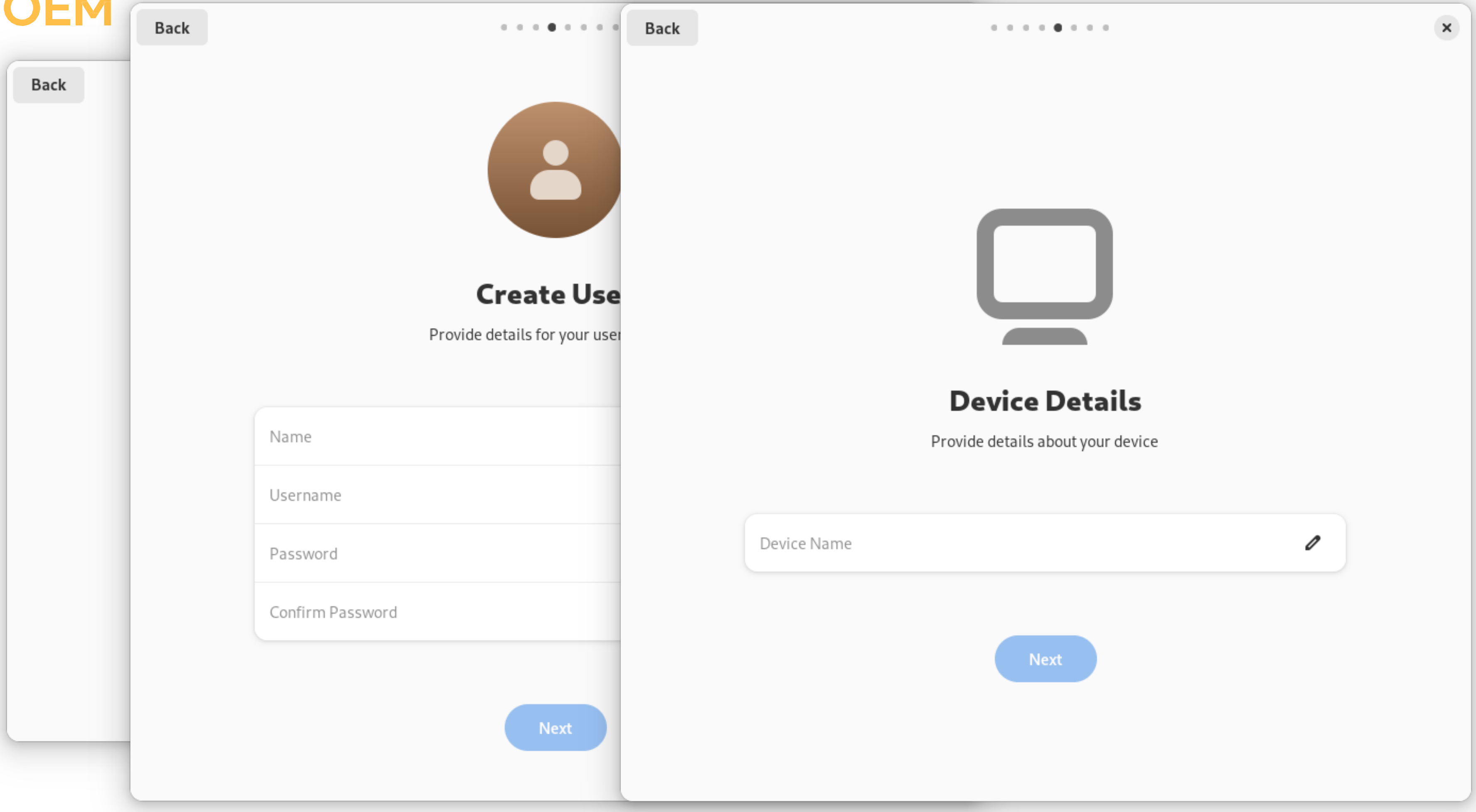
Username

Password

Confirm Password

Vanilla First Setup

OEM



Create Use

Provide details for your user

Name

Username

Password

Confirm Password

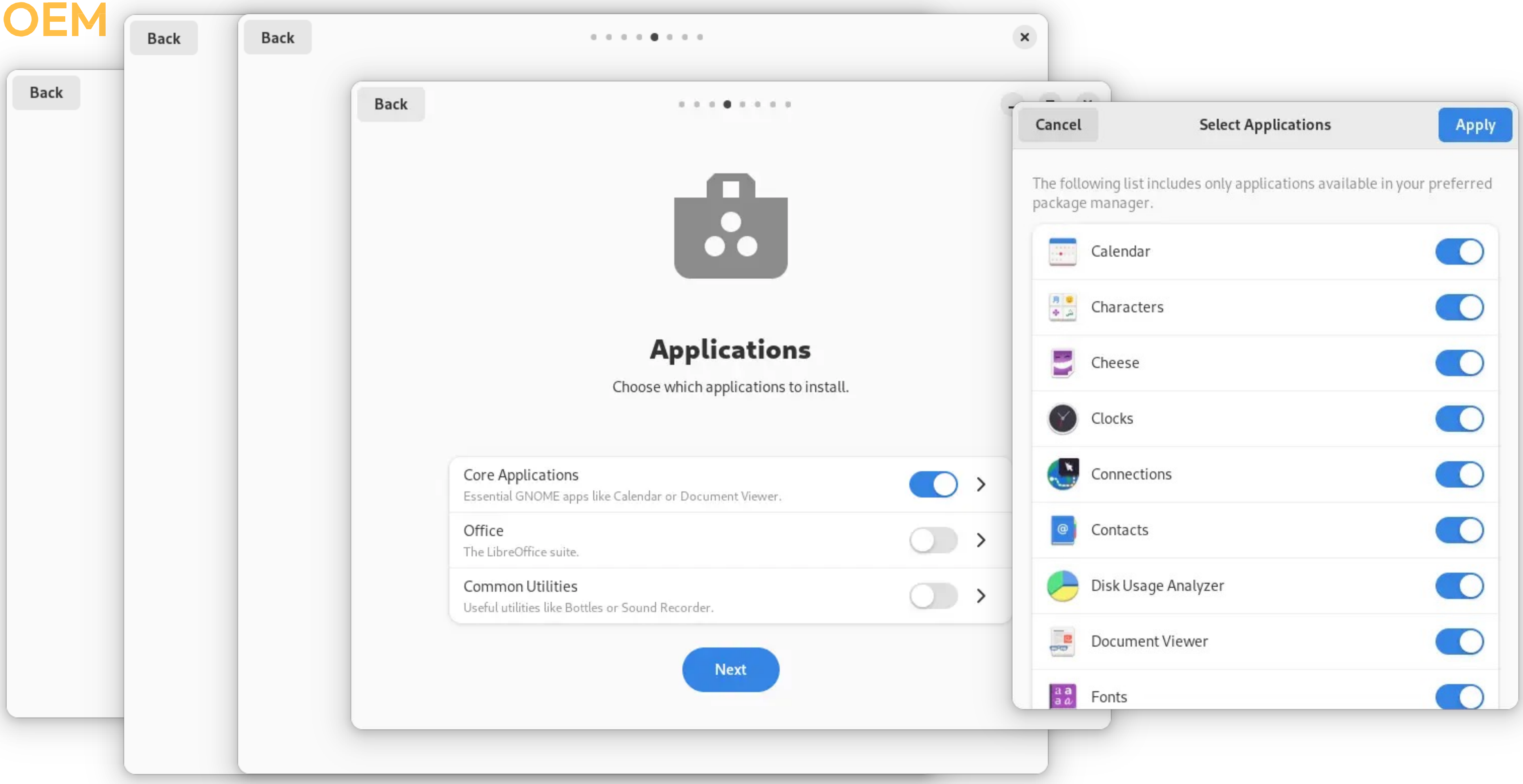
Device Details

Provide details about your device

Device Name

Vanilla First Setup

OEM



Applications

Choose which applications to install.

- ☑ Core Applications — Essential GNOME apps like Calendar or Document Viewer.
- ☐ Office — The LibreOffice suite.
- ☐ Common Utilities — Useful utilities like Bottles or Sound Recorder.

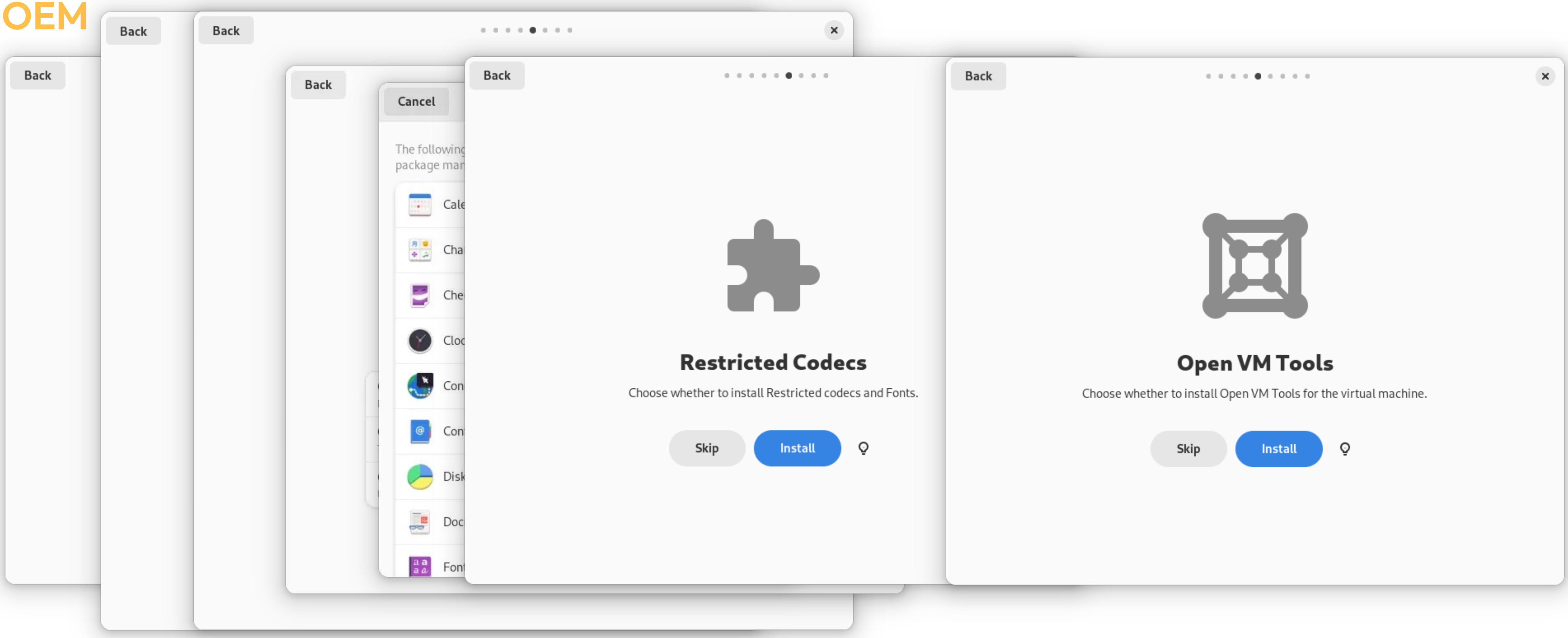
Select Applications

The following list includes only applications available in your preferred package manager.

- ☑ Calendar
- ☑ Characters
- ☑ Cheese
- ☑ Clocks
- ☑ Connections
- ☑ Contacts
- ☑ Disk Usage Analyzer
- ☑ Document Viewer
- ☑ Fonts

Vanilla First Setup

OEM



The following
package man

Restricted Codecs

Choose whether to install Restricted codecs and Fonts.

Open VM Tools

Choose whether to install Open VM Tools for the virtual machine.

The underlying structure raises some question

What if I need to make more complex changes?

How can flavors be made?

What can I do if I want to make a reproducible image that I can use for my devices?

The solution?

Vib

Vib (Vanilla image builder) is a tool that streamlines the creation of container images. It does so through YAML recipe files, which use a similar syntax to Flatpak manifests.

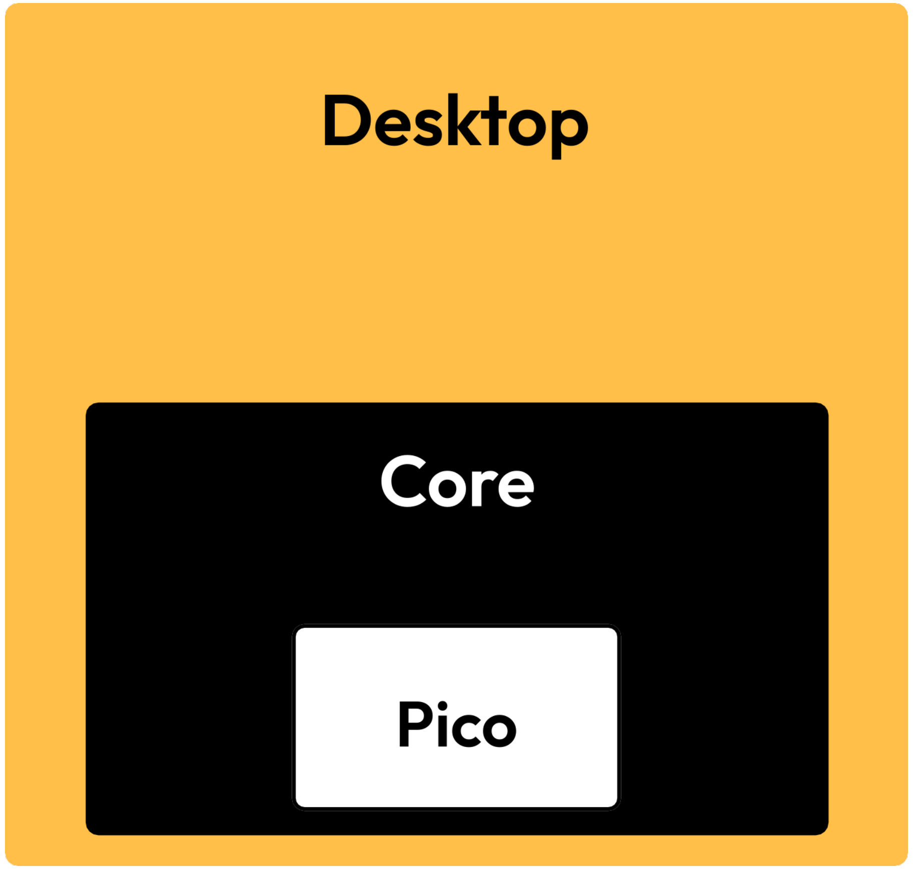
Thanks to modules and build instructions, it is possible to easily create custom images based on Vanilla OS, or any other Linux distribution.

System Composition

Vib

Vib can be used with any Linux distribution that provides an OCI image. Vanilla OS offers three images: Pico, Core, and Desktop.

These images can be used as a base to create derivatives of Vanilla OS. For instance, you can create a Server version or one with KDE using Core, or add extra software using Desktop. Meanwhile, Pico allows the greatest flexibility.



□ Pico is a Debian rootfs built using the Vanilla OS repository during the debootstrap process

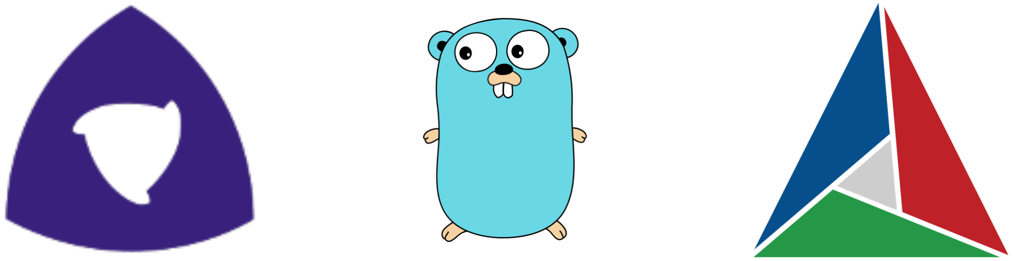
■ Core is based on Pico and pre-install all the Core components of Vanilla OS

🟨 Desktop is based on Core and adds all the Vanilla OS features and configurations.

Modules

Vib

Modules are a convenient way to define what to include in the image. There are modules for compiling software, copying files, expanding archives, using the package manager, and more.



apt   dpkg   dnf   rpm   >shell

Modules

Vib

```
modules:
- name: mandb
  type: shell
  commands:
  - apt update
  - apt install -y man-db
  - mandb -c
```

>shell

Modules

Vib

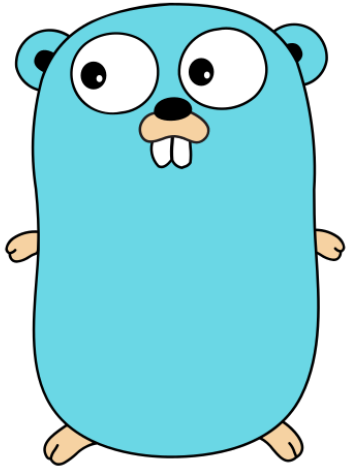
```
modules:
- name: nvidia-driver
  type: apt
  source:
    packages:
    - nvidia-driver
    - libnvidia-cfg1
```

apt

Modules

Vib

```yaml
modules:
- name: abroot
  type: go
  source:
    type: git
    url: github/abroot.git
    tag: 2.0
```



Release date?

vanillaos.org

it's all about
user experience

Contributing

Money

Code

github.com/Vanilla-OS

Community

Our team

Mirko Brombin

Founder

Head of Design (UI/UX)

Luca di Maio

Co-Founder

Software Engineer

Pietro Di Caprio

Public Relations Manager

Software Engineer

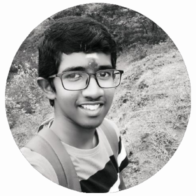
Muqtadir

Software Engineer

UI Designer

Mateus Melchiades

Software Engineer



K.B.Dharun Krishna

Documentation Team Lead

Localization Team Lead

Bart Gravendeel

Documentation Writer

Moderator

Hari Rana

UX Designer

Moderator

Kramo

UI Designer

Illustrator

Dallas Strouse

Website Accessibility

axtlos

Software Engineer

Lit Contributor

...you?

Thank you!

vanillaos.org

twitter.com/VanillaOSLinux

fosstodon.org/@vanillaos

Questions?