Rudra Saraswat
rs2009

atomicity and **immutability** in Linux distributions

what will we be discussing?

- what, why, and how immutable and atomic Linux distributions work
- existing approaches and implementations
- blendOS v4 → v5, a base-independent atomic distribution

why not a traditional distro?

- package operations mutate the live filesystem
- partial updates → inconsistent state
- package managers assume they are the only source of truth
- (desktop) system updates should be predictable

transactional system updates >>>>>>

immutability vs atomicity

immutability

- (parts of) root filesystem is read-only or difficult to modify
- user configuration and data remains mutable
- does not necessitate atomicity

atomicity

- updates are indivisible transactions
- take place as a single unit, no in-between state
- immutability can usually be assumed in a desktop context

existing approaches

libostree

- Fedora Atomic Desktops
- Endless OS
- openSUSE Aeon

...

bootc

- Bazzite
- Project Bluefin
- Fedora Atomic Desktops

...

A/B layouts

- SteamOS
- VanillaOS
- Android

...

existing approaches

libostree

- content-addressed filesystem tree, like git commits
- active deployment is read-only
- deduplication of unchanged files during updates
- requires OSTree-specific tooling

bootc

- operating system packaged as an OCI container image
- distributed through standard image registries
- atomic image-based upgrades with rollback support
- reuses existing container tooling and CI/CD workflows

A/B layouts

- two identical root filesystem partitions
- updates are applied to the inactive partition
- bootloader switches partitions atomically after reboot
- duplicate partitions → increased storage usage

blendOS v4

- atomic and immutable
- Arch-based
- system controlled via declarative config

sample track configuration (XFCE) →

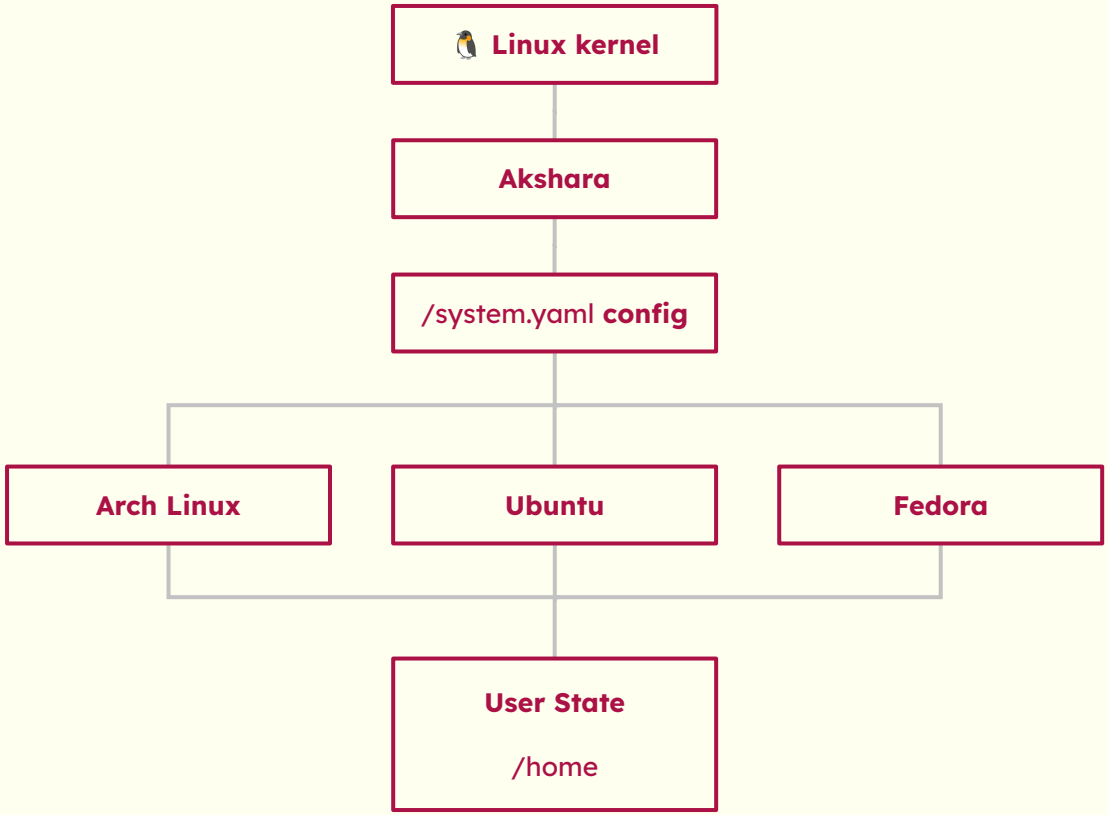
```
impl:
'https://github.com/blend-os/tracks/raw/main'

track: 'blendos-base'

packages:
  - 'xfce4'
  - 'firefox'
  - 'lightdm'
  - 'lightdm-gtk-greeter'

services:
  - 'lightdm'

commands:
  - 'systemctl enable lightdm'
```

blendOS v5

- atomic and immutable, again
- any (!) base Linux distribution
- declarative configuration
- image-based or traditional package manager

! – well, not quite, but we're getting there…

v5 devlog explaining design decisions



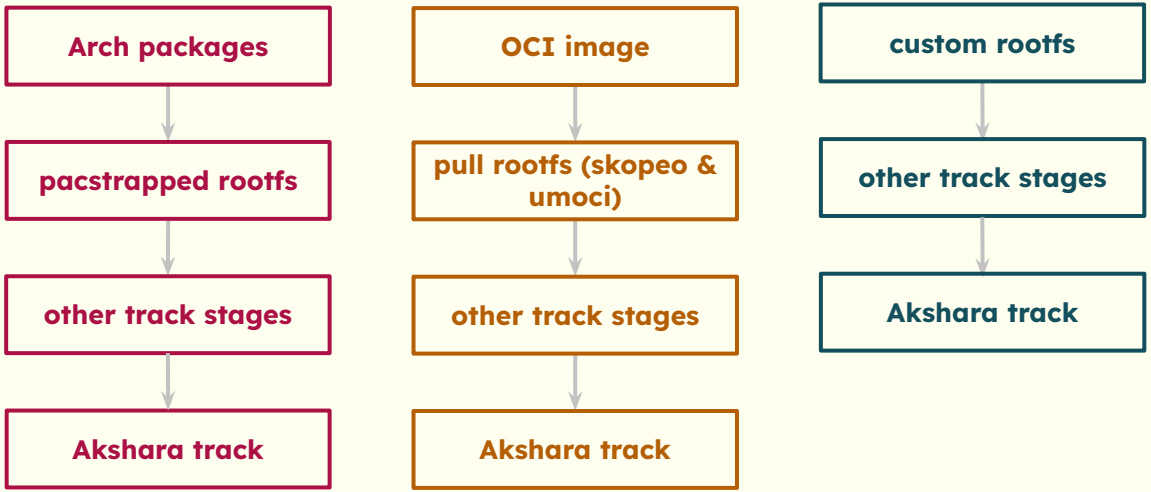
v5 devlog explaining design decisions

tracks in blendOS v5

- defines how to create the next system state
- creates filesystem tree
- determines origin of base filesystem – OCI image, pacstrap, debootstrap, dnf, etc.
- controls how files are preserved and updates are detected

v5 devlog explaining design decisions

a pluggable base



v5 devlog explaining design decisions

...and more!

OCI ∈ tracks

- a convenient way to describe a base filesystem: registry → OCI image → filesystem tree
- BUT atomic system != OCI image
- i.e. just another useable base track taking an image identifier as an input

v5 devlog explaining design decisions

a universal packages cli

- a single interface for installing software across multiple base distros
- abstracts away distro-specific package managers
- resolves packages from most suitable source and environment
- handles dependencies and compatibility automatically

v5 devlog explaining design decisions

where are we thus far?

- daily-driveable Arch Linux and Ubuntu (OCI image-based) GNOME and Plasma tracks; Fedora base track soon
- universal package manager in the works, expansion of bpkg from blendOS v4
- Calamares installation images already available

v5 devlog explaining design decisions

tl;dr

- immutable = read-only rootfs
  atomic = transactional system operations
- various implementations of atomic updates, even for the same given distro
- blendOS v5 combines atomicity with multiple Linux bases – Arch Linux, Ubuntu, etc.
- OCI images are just another blendOS track

https://blendos.co

thank you! questions?

>2036

>year of the linux desktop

>no more windows

>no more macos

>no more distro wars

>wait

>we put all the distro wars in one distro