Aeon

Being different takes Aeons

A tale of the endless RC?

Richard Brown

About Me

Distribution Architect at SUSE

Former Sysadmin, Systems Manager, QA Engineer, Distribution/Release Engineer, openSUSE Board Member & Chair

10+ years at SUSE

20+ years in Open Source

Creator of Aeon

Agenda

Brief Introduction to Aeon Desktop

The Long Road Travelled

Will we ever release?

Lessons Learned

Open Discussion

Aeon Desktop

Traditional Linux is NOT good enough

Traditional Linux is like a Swiss Army Knife

Lots of Services & Features, all highly customisable

- With increased chance of incompatibilities between services
- And increased risk of complex failures
- Each installation can be wildly different from the next, sometimes even when installing from the same media
- All software has to be built assuming it could be run on Servers, Desktops, VMs, Edge, possibly multiple at the same time

Aeon

You Already Know This

Countless efforts made to build wonderfully complex, flexible Linux, just for users to take it and:

- Deploy Single Purpose VM/Cloud/IoT Instances
- Minimise / Disable / Uninstall default services
- Delay / Ignore patching
- Rip n' Replace instead of Upgrading
- Automation over Micro-Management
- Adding redundant instances

The Dream

Our operating systems should “just work”

They shouldn’t need complex policies, procedures, best practices to live with on a daily basis

They should be built in a way where the best practices are the standard operating model, and any remaining rough edges are mitigated as much as possible

The Dream

The “perfect” Linux should be optimised for what everyone already does:

- Single purpose installations
- Minimised Services
- Updates with zero impact to the running system & services
- Self-Healing over manual repair
- Use at scale
- Ready to be automated & orchestrated

The Dream

The “perfect” Linux should avoid:

- Requiring manual maintenance
- Operating under an ‘inconsistent’ or ‘unknown’ state
- Requiring complex additional orchestration or automation tooling

For Servers, the “perfect” Linux already exists

MicroOS

- Community Server/Container Host
- Rolling Release
- Fully Automated Updates
- OS is never altered during runtime
- OS always operates under a “known” state
- Designed to do “just one job”
- Minimised Footprint
- Ready for Automation/Orchestration

Desktops are special

Desktops cannot be automated & hidden away, only to be touched when something goes horribly wrong

They need to be an inviting, reliable platform for people to use for work, play, and everything in between

They need to be **as reliable**, if not more so, than any server

- Any failure is a huge problem for that user till it is resolved
- Users can’t really walk around with redundant laptops

Aeon

Introducing Aeon Desktop

Reliable, Predictable & Immutable

- Just like regular MicroOS

Opinionated

- Supports GNOME only
- “Chromebook-like” experience
- Image-based installation

Minimal, yet Functional

- Printing, Gaming, Development and much more must all work

Works straight “out of the box”

- Additional configuration must not be needed before being able to get to work

The Long Road

MicroOS Desktop

- Started in 2020 as a SUSE Hackweek Project
- Offered as a ‘System Role’ for regular MicroOS
- Implemented initially as a single RPM pattern pulling in GNOME
  - Dependant on all the regular MicroOS patterns
  - Shared MicroOS’s ISO, RPM-based installer, /etc/os-release, etc

MicroOS Desktop - Initial Technical Problems

Implemented initially as a single RPM pattern pulling in GNOME

- Dependant on all the regular MicroOS patterns
  - MicroOS targeted very different hardware (VMs, Cloud, Embedded) leading to both excess and missing drivers
  - Core Configuration didn't match desktop use cases (eg. sudo, polkit, SELinux)
  - MicroOS Feature Patterns such as Cockpit, Remote Attestation, IMA/EVM all interacted with the Desktop in weird ways, if installed (and not easy to stop tinkerers from installing it)
- Shared MicroOS's ISO, RPM-based installer, /etc/os-release, etc
  - Had to follow the same install workflow (eg. Must have a root account)
  - Couldn't deploy an image
  - Couldn't identify whether a system was MicroOS (Server) or MicroOS Desktop from any obvious config file like /etc/os-release

MicroOS Desktop - Initial Community Problems

Offered as a ‘System Role’ for regular MicroOS

“Cool! I’m going to contribute my favourite Desktop Environment!”
- (Some) MicroOS Desktop Users

MicroOS Desktop - Initial Community Problems

Offered as a ‘System Role’ for regular MicroOS

“Cool! I’m going to ~~contribute~~ force Richard & team to support my favourite Desktop Environment! And never be seen again!”

- (Some) MicroOS Desktop Users

MicroOS Desktop - Initial Community Problems

Offered as a ‘System Role’ for regular MicroOS

“Cool! I’m going to ~~contribute~~ force Richard & team to support my favourite Desktop Environment! And never be seen again!”
- (Some) MicroOS Desktop Users

And so MicroOS Desktop (KDE) was born

And MicroOS Desktop was renamed MicroOS Desktop (GNOME)

More Desktops, More Problems

Introduction of MicroOS Desktop (KDE) brought significant problems with few benefits

- Increased attention amongst the KDE userbase of openSUSE
  - But no new contributors..so more demands, more complaints, but no one to do anything with them
- Increased complexity of the MicroOS Desktop stack
  - Not just one pattern atop MicroOS, but three (GNOME, KDE, plus a ‘Common Desktop’ Pattern)
  - Constant debate and technical challenge as to what made sense to exist in which pattern
  - Debates were often pointless; was almost always implemented by the MicroOS Desktop (GNOME) contributors anyway
- Mixed messaging
  - Team had to constantly correct the assumptions that KDE was the “Main” flavour of MicroOS Desktop despite being all but unmaintained from Day 1

Why Not Drop MicroOS Desktop (KDE)?

We tried

- Initial MicroOS Desktop KDE contributor complained to openSUSE Board
- Despite being explicitly forbidden from directing development, openSUSE Board directed MicroOS Desktop team to continue offering KDE

MicroOS Desktop (KDE) remained practically unmaintained while MicroOS Desktop (GNOME) steadily progressed through it’s Alpha, Beta and reached a “Release Candidate” Stage

Glossary

In simple terms, I consider these to be the phases of software development

- Alpha
  - It’s a start, missing many features
  - It should work, but could easily break
- Beta
  - It’s almost complete
  - It should work, but no promises
- Release Candidate
  - It’s complete
  - It works, but may have some rough edges
- Release
  - It’s complete & works

MicroOS Desktop (GNOME) RC

In Early 2023, MicroOS Desktop (GNOME) was first considered “Feature Complete”

- Minimised GNOME Install (No RPM Apps besides Software and Terminal)
- Flatpak + Flathub out of the box for GUI Apps
- Distrobox out of the box for CLI Apps
- Desktop Notifications for background updates
- Minimised dependence on MicroOS Installer
    - GNOME Initial Setup for much desktop config on first boot

MicroOS Desktop (KDE) was still in [ALPHA]
(and still is today)

RC but not RC?

MicroOS Desktop GNOME RC retained some significant issues

- Still dependant on MicroOS Patterns, systemd-presets, polkit and SELinux policies
  - Inherited some ‘cruft’ from the shared ‘Desktop Common’ Pattern
  - Couldn’t run all Flatpaks (eg Steam) as seamlessly as desired
  - Needed a separate root password and was required a lot by GNOME Settings, Printers, etc
  - Couldn’t disable rebootmgr causing automatic reboots
  - Couldn’t reliably distinguish between MicroOS Desktop and regular MicroOS
- Still dependant on YaST as the Installer
  - Couldn’t prevent users from customising (aka breaking) the installed packages
  - Couldn’t prevent users from customising (aka breaking) the partition layout
  - Couldn’t disable the hard requirement for a root password
- Still had the MicroOS Desktop GNOME vs KDE confusion, and growing worse

Aeon

Aeon Desktop

May 2023 announced rename of MicroOS Desktop (GNOME) to Aeon Desktop

Immediately introduced a clearer, tighter definition (GNOME only, Desktop only)

Began addressing the technical debt & pursuing freedoms available by splitting from the MicroOS base

MicroOS Desktop (GNOME) RC became retroactively known as Aeon RC1

Aeon RC2

May 2024 saw the release of Aeon RC2

- Brand new installer (tik)
  - Image-based (dd or systemd-repart)
  - Modular (add your own install steps)
  - Supports ignition/combustion for first-boot config
  - Migrates /home & configs from MicroOS Desktop (or any other btrfs distro)
- No root password
- sudo permissions for first created user
- Polkit & SELinux tuned for Desktop
- No automatic reboots
- Programmatically identifiable as Aeon, not MicroOS

Aeon RC2

But wait, there's more

Aeon RC2 also addressed things we didn't care about in 2023

- systemd-boot
- Silent, flicker-free boot process
- ZRAM by default
- Btrfs compression by default
- SSH disabled by default, managed by GNOME Settings
- x86_64_v3 optimised libraries
- Distrobox automatic updates

Aeon RC3

July 2024 saw the release of Aeon RC3

- FDE by Default, in one of two modes
  - Default (TPM-backed automatic unlock)
  - Fallback (Passphrase, for systems without sufficient TPM support)

Has worked very well, but did take some time to adjust to the diversity of real hardware out there and differences in firmware update methods.

Will we ever “release”?

Yes..but..

There’s one thing we really must address first:

- openQA Testing

Without it we can’t be *sure* Aeon works every day. We’d love contributions!

And there’s a whole bunch we might also do meanwhile:

- UKI Support
- Improved Self Healing
- Remove TTY Consoles
- Installation/Re-installation from OCI Images

But..this is all stuff we probably could (and should) do post-release

Lessons Learned

Lessons Learned

- Tightly define your projects goals
  - Or have a plan for dramatic, disruptive redefinitions as you go along
- Don’t always listen to your community
  - Lots of people have great ideas, but they’re worthless if no one implements them
  - If you already have enough great ideas, focus on implementing those ones
  - It’s not your job to enable contributions in areas you’re not interested in
- Feature creep is real, a blessing, and a curse
  - Aeon is as awesome as it is today because we’ve adapted and adopted new things as they became available and/or popular
  - But that has recurrently kept us from being able to say its 100% “Ready”
- There are no absolutes in Distro Development
  - But with a clear goal, it’s a lot easier to decide what imperfect compromise is right for your Project

Open Discussion

When is something “Release Ready?”

Is it better to be stuck in RC than Release and make major changes?

Do terms like Alpha, Beta, RC, even make sense with Rolling Releases?

www.aeondesktop.org