

Kartoza Desktop Strategy

A plan to normalise our work environment.

Introduction

About Me

- Extensive experience with Linux since 1998
- Highly adept and well-versed in Linux environments
- Good sysadmin skills
- Microsoft Certified Systems Engineer way back in ~2000
- Very good experience using and supporting Windows, macOS, Linux
- Expertise in managing desktop environments based on Linux

 Graphic Placeholder

Goals and Agendas

My Objectives

- Streamline ICT management
- Enhance system efficiency and security
- Promote open-source solutions in Kartoza
- Develop a plan for harmonization of approaches of company ICT, Development and Devops environments.

- **DevOps Environment (Lian)**

- Focus on deployment and operations
- Continuous integration and delivery

- **ICT Environment (Me)**

- Managing company infrastructure and user environments
- Ensuring security and efficiency

Bridging the Gap

- Create a seamless process from ICT to DevOps
- Align ICT management with DevOps requirements
- Use a well-established ICT framework
- Develop norms and guidelines for developers
- Ensure alignment between development environments and production deployments

- Outlines the benefits of NixOS for desktop environments
- Detailed analysis of stability, security, and maintainability
- Comparison with other operating systems

Key Points

- **Consistency:** Ensures consistent environments across all desktops
- **Reproducibility:** Easy to replicate environments for general staff and developers
- **Security:** Robust package management and isolation
- **Flexibility:** Highly customizable to meet specific needs

Why NixOS?

- Aligns with our goal of using open-source solutions
- Supports both ICT and Dev requirements

- **Building Desktop Environments:**

- NixOS allows for easy construction of desktop environments
- Ensures consistency and reproducibility across environments

- **Building Docker Containers:**

- NixFlex allows for easy construction of Docker containers
- Ensures consistency and reproducibility across environments

- **Provisioning for Deployment:**

- Secure and lightweight provisioning
- Supports frequent updates and deployments
- Enhances security through isolated environments

Benefits

- **Repeatability**
 - Ability to build the same container artifact repeatedly
 - Ensures consistency over time given the same recipe
- **Security**
 - Utilize tools like [Grype](#), [Syft](#) and [Grant](#), for:
 - Managing security
 - Maintain a bill of materials and bill of licenses
 - Provide an audit trail for all products and dependencies included in the container
- **Chain of Execution**
 - Use Jenkins for building containers
 - Employ a NixOS environment for the build process
 - Ensure the container has all necessary tools for building a NixOS flake

Key Features of the Desktop Environments

- **Secure Environment**
 - Ensure robust security measures
 - Protect user data and system integrity
- **Predictable Application Suite**
 - Provide a consistent set of applications
 - Align with the needs outlined in the white paper



Graphic Placeholder

Special Features of the Desktop Environment

ZFS

- **Push Backups**
 - Utilize ZFS send to push users' desktop home environments
 - Backup to NAS environments for data safety and redundancy

Management

- Every desktop will be on the VPN
- SSH running exclusively accessible from the VPN environment
- Enable remote management of machines
- Support users, handle security issues, and check machine statuses

Remote Desktop Access

- Use GNOME 46 for front-pointed login
- Provide remote desktop environment for users
- Access desktops from anywhere within the VPN
- Enable remote desktop access to cloud computers

Secure Environment

- Ensure robust security measures
- Protect user data and system integrity

Predictable Application Suite

- Provide a consistent set of applications
- Align with the needs outlined in the white paper

 Graphic Placeholder



Standardized Environments

Standardization: Simplifying Support & Enhancing Security



Security and Compliance

Built-in Security: Compliance with
Kartoza Standards

- KeePassXC
- Firewall
- VPN
- Disk encryption etc.



Reducing Variability

Reducing Variability: Consistent
Systems Across the Board



Ease of Support

Streamlined Support for Remote Workforce

- Deploy fixes via GIT
- Direct machine access when needed
- Single support target

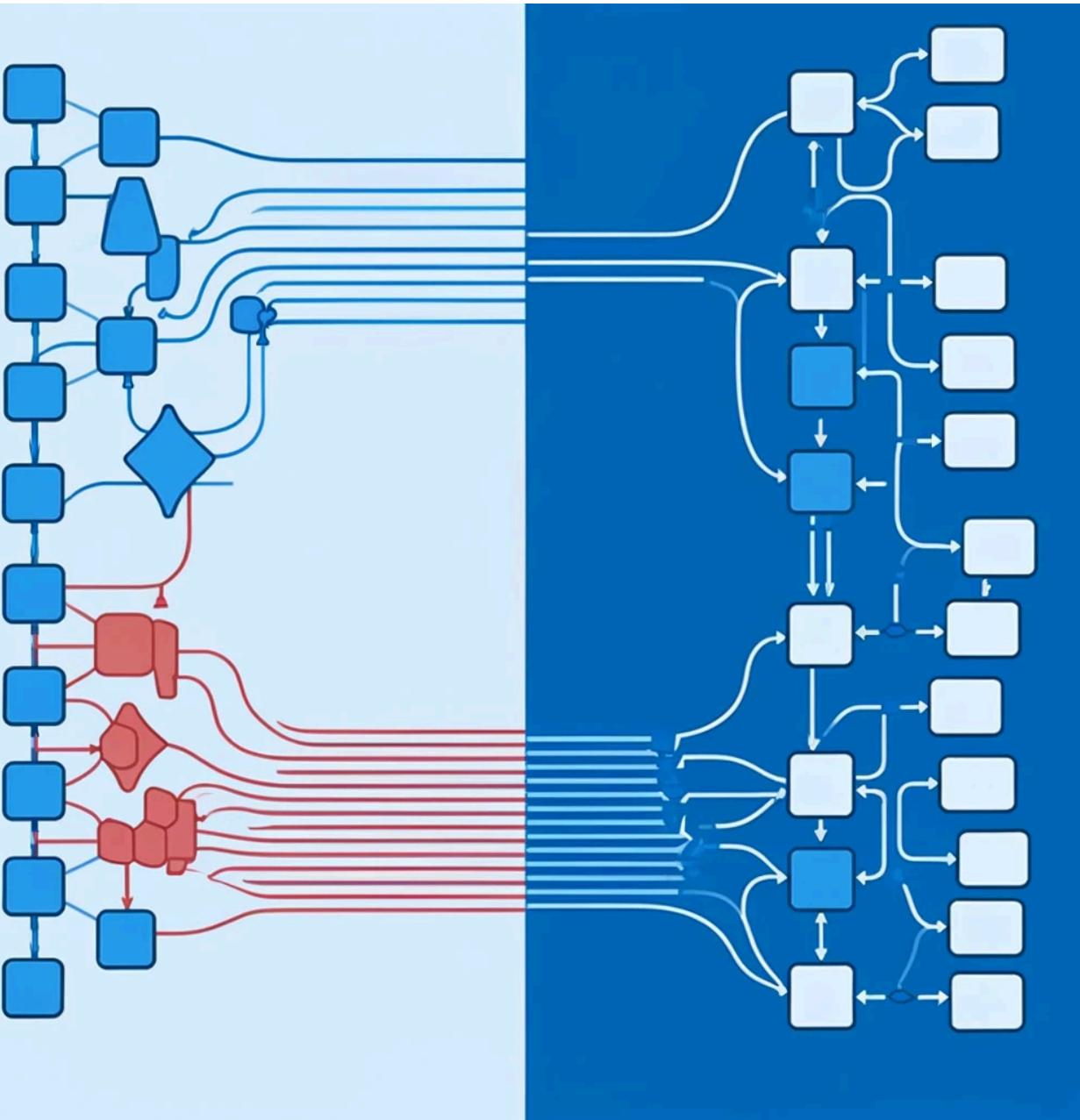


Branding and Professionalism

Our branding will show on:

- Screen shares
- Training sessions
- Screen captures

Staff will sense they are 'at work'



Efficiency

Efficient development env.

- Direct on NixOS, No VM needed for most cases
- 'Out the box' developer tools (docker, vm, shells, direnv)
- `nix.shell & shell.nix` is awesome!



Community and Flexibility

We can establish an internal community around 'our' operating system.

Staff can participate in shaping the environment we all use.



Misconceptions

NixOS is no harder to use than any other distro.

Most users will not even know they are using NixOS vs Ubuntu or another distro.



Remote Management

Systems with ZFS can push encrypted backups to a remote NAS.

We can support users remotely via VPN and SSH.



NixOS at Kartoza

Media Creation Experience

Optimized for Media: Create and Innovate Effortlessly



Supporting Work Modalities

- Admin staff
- Developers
- GIS staff
- Devops
- Interns
- Training
- Media creation



Conclusion

Another step in the growth and maturity of Kartoza following initiatives like:

- ERP and efficient admin
- Devops and carefully managed infrastructure
- Training strategy overhaul