

# Non-programmer view on Guix

Laszlo Krajnikovszkij  
2022

# Contents

- Introduction
- Road to free software
- Guix package manager
- Guix system distribution
- Guix home
- Malleable systems
- Compartmentalized systems
- Context-defined virtual environments
- Derivative ideas
- Conclusion

# Introduction

Name: Laszlo Krajnikovszkij

Email: [laszlo@laszlo.is](mailto:laszlo@laszlo.is)

# Road to free software

Mac OS → Ubuntu MATE → Emacs + WM → Manjaro → Guix

# Guix package manager

## **Pros**

- Easy to install on any distro
- Can complement or replace the native package manager

## **Cons**

- Building large programs can fail on weak hardware
- Many software packages are not available in Guix

# Guix system distribution

## **Pros**

- System configuration is fun and simple
- Systems can be replicated from the base config

## **Cons**

- Need to depend on other package managers
- Some more advance configuration is difficult to get into

# Guix home

## **Pros**

- Should complement system configuration perfectly
- Easy to init

## **Cons**

- Tendency to get back to usual management of dotfiles
- Some concepts are not clear

# Malleable systems

1. Software must be as easy to change as it is to use it
2. All layers, from the user interface through functionality to the data within, must support arbitrary recombination and reuse in new environments
3. Tools should strive to be easy to begin working with but still have lots of open-ended potential
4. People of all experience levels must be able to retain ownership and control
5. Recombined workflows and experiences must be freely sharable with others
6. Modifying a system should happen in the context of use, rather than through some separate development toolchain and skill set
7. Computing should be a thoughtfully crafted, fun, and empowering experience

# Compartmentalized systems

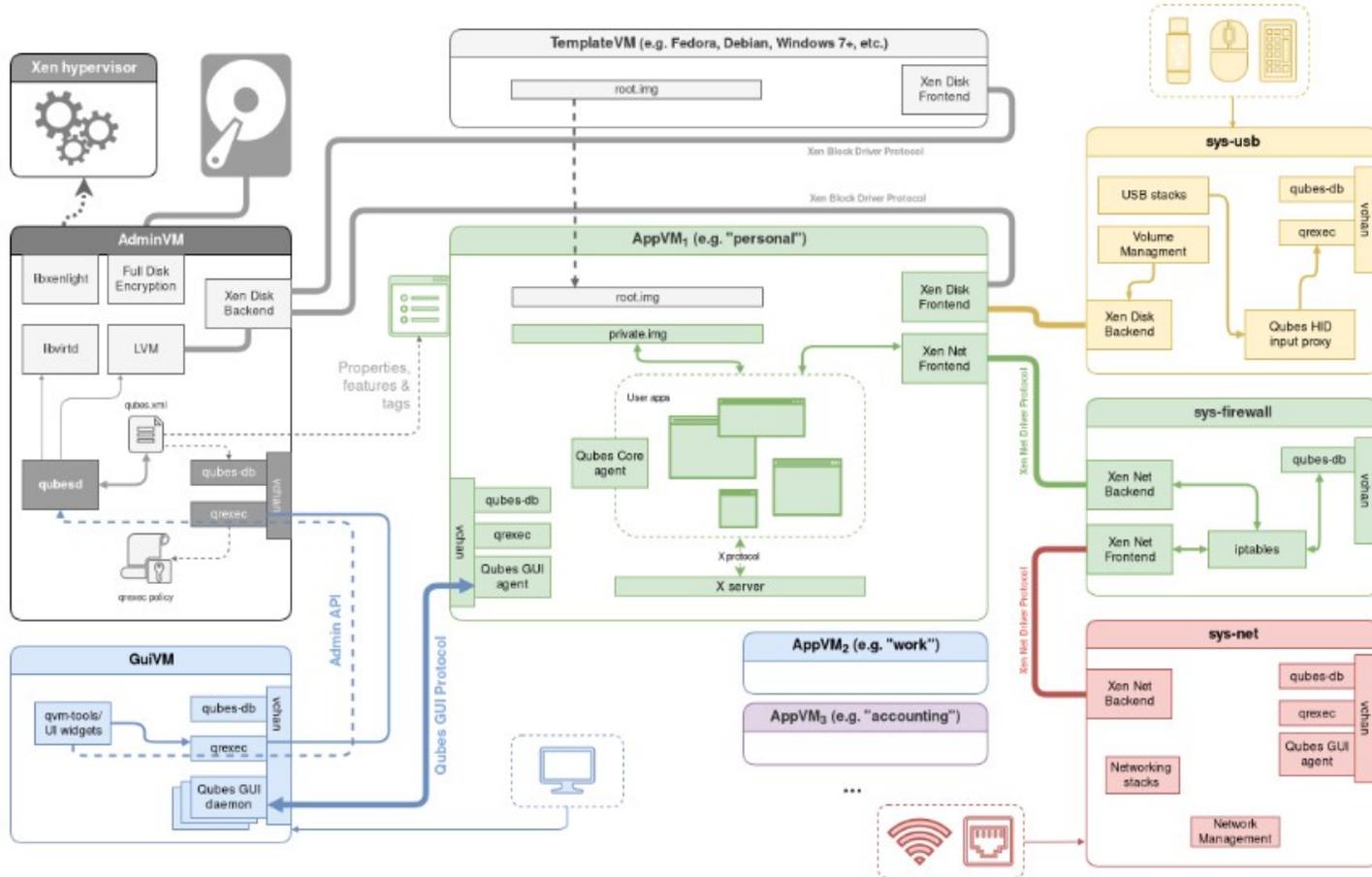
## **Qubes OS**

- Xen hypervisor
- Isolated filesystems
- Fedora images

## **Spectrum OS**

- Using Nix images
- Shared filesystem
- Per-app isolation

# Compartmentalized systems



# Context-defined virtual environments

**Lightweight Guix images**

**Emacs**

**WM tools**

**Browser**

# Derivative ideas

- **Self-hosted desktop container streaming**
- **Collaboration workspaces**
- **P2P network for Org mode based hypermedia**
- **Federated network for communities and SMEs**
- **No-code/low-code system configuration**

# Conclusion

- **Guix is very promising technology**
- **Needs to get more user friendly**
- **Needs more practical use cases for regular users**

**Thank you**