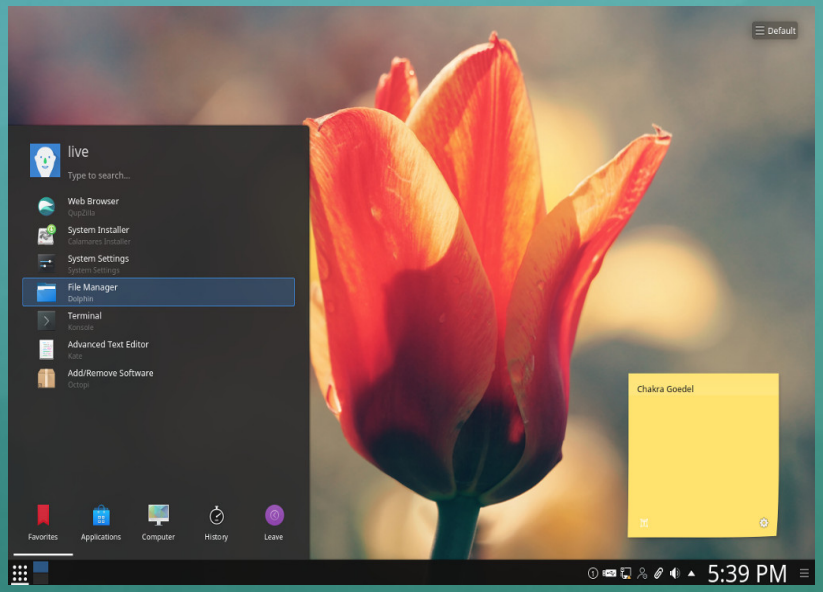
The **Half Rolling** repository model

The golden intersection for desktop users?

About Chakra

- Focus on ***KDE*** and ***Qt*** Software
- ***Independent***, using ***Arch*** technologies
- ***Half-Rolling*** repository model



Chakra 2017.10 ‘Goedel’

About me

Neofytos Kolokotronis

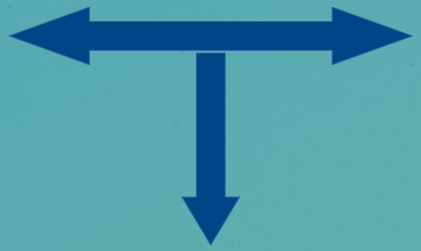
- Chakra team member
- Community & Project management
- Collaborator at Free Software and Open Data/Government projects
- neofytosk.com, @tetris4

Popular Models

Fully Rolling  **Non Rolling - Fixed**

Popular Models

Fully Rolling



Non Rolling - Fixed

Half Rolling

Fully Rolling

Advantages

- *Latest software* versions with new:
  - functionalities
  - security updates
  - bug fixes
- *Continuous* upgrades

Fully Rolling

Disadvantages

Each upgrade comes with a *risk*:

- regression
- bug
- broken system

Fully Rolling

Examples

- ArchLinux
- Gentoo

‘Racing Skaters’ by Alternate Skate
https://unsplash.com/photos/_tH3YCjPCCE

Ideal for

- *Enthusiasts* after the latest & greatest
- *Experienced* users

Non Rolling - Fixed Releases

Advantages

- Stability ⇒ less risks from upgrading
- Easier maintenance

Non Rolling – Fixed Releases

Disadvantages

- End of Life of a release ⇒ major upgrade risks
- Slower availability of new software versions

Non Rolling – Fixed Releases

Examples

- Debian
- Ubuntu
- Fedora

Ideal for

- *Workstations*
- *Servers*
- *Casual users*

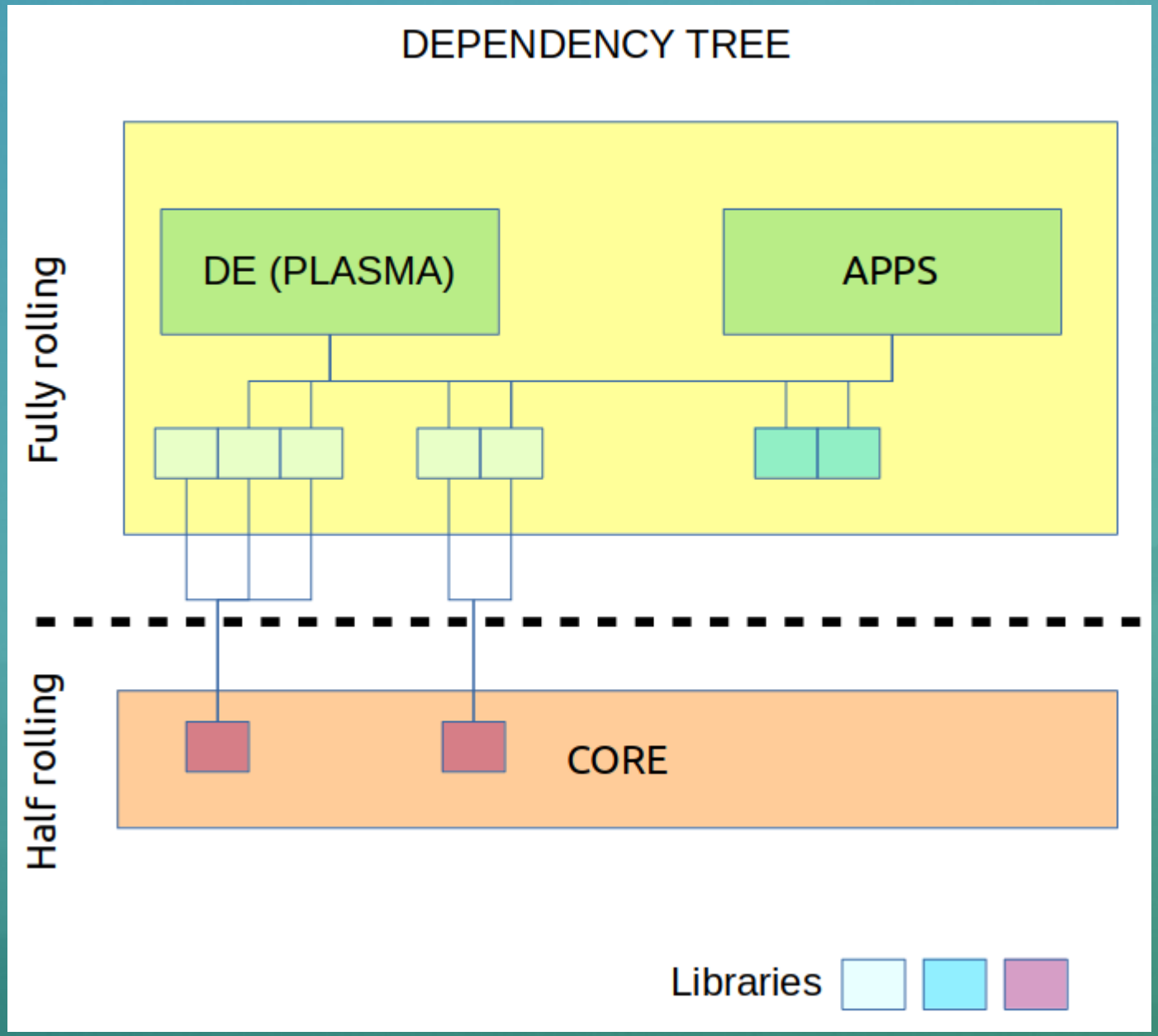
'fixed?' by D. Midgley
https://www.flickr.com/photos/petrichor/406667698/

Half Rolling - Introduction

Two layers of software

1. A stable ***core*** of software updated *periodically*.
2. *Fully rolling* ***DE*** and ***applications*** on top of it.

Half Rolling - Introduction



Half Rolling - Implementation

1. **Core**
   - Software *critical* for an operational system
   - *Periodically* updated on *scheduled* intervals

Half Rolling - Implementation

1. Core

Groups of packages

- kernel & drivers
- xorg
- multimedia
- boost
- glib
- python
- ...

Half Rolling - Implementation

1. Core Updates

Not all package groups are created equal

- kernel, graphics drivers ⇒ 3-4 times per year
- xorg, multimedia ⇒ 1-2 times per year

Half Rolling - Implementation

2. Desktop Environment & Applications

What users interact with

- Plasma DE and Frameworks by KDE
- Applications
- Any related dependencies

Half Rolling - Implementation

Testing Repository

- Unified
- Packages kept for several days or weeks
- Move to stable repositories accordingly: core, desktop, gtk, lib32

Half Rolling – Advantages

- *Balance* in **stability** as updates are:
  - periodical
  - foreseeable
- **Latest** versions of *applications*
- One time installation

Half Rolling – Disadvantages

VS **Fully Rolling**

Slower upgrades of core components

VS **Non Rolling Fixed Release**

Increased upgrade-related *risks*

Half Rolling – Challenges

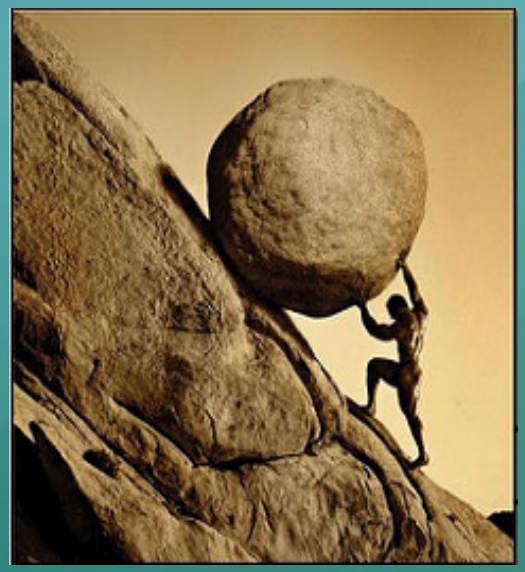
***1.** Application requiring **a newer library** version*

- new library version breaks compatibility?
- library part of a group of packages in core?

Half Rolling – Challenges

***2. Security vulnerabilities** in core packages*

- Is an update really required?



'Sisyphus' by Gerard Van der Leun
https://www.flickr.com/photos/1000photosofnewyorkcity/8819982782/

Half Rolling - Challenges

3.** Sticking to the **schedule

- Lack of packaging resources
- Unsheduled required upgrades
- Unpredictable issues
- Need to keep rolling

Half Rolling – Challenges

Solutions

- *Update* application and library
- *Patch*
- *Recompile* the whole group
- Application and library version *freeze*
- *Postpone* or *skip* a group

Half Rolling – Is it for you?

Ideal for

- Casual desktop users
- Gamers
- Small offices and businesses
- School laboratories

Questions?

Thank you!

Let’s connect:

- neofytosk.com
- @tetris4