

KDE & Distros

Marcus D. Hanwell cryos@gentoo.org
M. Gökmen Göksel gokmen@pardus.org.tr

Camp KDE 2009
18 January, 2009



What Is Gentoo Linux?

- Gentoo Linux is a source based distribution
- Large repository of ebuilds to facilitate building from source
- The ebuild repository updates from the CVS tree live
- Portage tracks dependencies, downloads source and compiles
- Large range of supported hardware architectures
- Emphasis on providing choice through USE flags
- Compile in only the features you want



What Is Pardus Linux?

- Pardus is a from scratch Linux distribution;
 - It has own package management,
 - Configuration sub-system,
 - Series of managers,
 - Installer and also own init system,
- Most of projects are written in Python
- Default and one supported desktop environment is **KDE**
- Supported by Scientific & Research Council of Turkey (TUBITAK)
- We have just released our minor version of Pardus that is 2008.2 Code name : **Gaza**
 - Install CD with **KDE 3.5.10**
 - Live CD with **KDE 3.5.10**
 - Live CD with **KDE 4.1.96**



Why Build From Source?

- Why build from source when
 - It is time consuming
 - Aggressive compiler flags cause weird bugs, `-fomg-faster`
 - Automagic build files break dependency tracking
- Each binary can be custom built
 - Compiler flags optimize to use the full power of the CPU
 - USE flags allow custom binaries to be compiled
 - Gives more power to users rather than packagers
- A large array of possible builds are possible from one source package



The KDE Team

- The KDE team takes care of KDE applications and libraries
- Gentoo and the KDE team is run by volunteers
- Small team managing a large number of ebuilds
- Gentoo is desktop environment agnostic—can be tough striking the right balance
- Always on the look out for volunteers and better ways to collaborate



Splitting Modules

- All (?) distros split the KDE modules for users
- Gentoo moved to split ebuilds only for KDE 4
- Most people want, not the whole module
- Many distros end up working around modules



Splitting Source Tarballs

- Am I the only developer who tends to work on a single application in a module?
- Would a move to git also prompt splitting?
- Gentoo already splits at compile time
- Matters much less to binary distros
- May lower the barrier for contributors



Automagic Is Bad

- Automagic detection of optional deps is bad!
 - Deterministic build systems are essential
 - Enable and disable flags should be provided
 - Automagic detection and inclusion breaks the dependency tree
 - All distros have similar issues with automagic detection—broken dependency trees
- CMake seems to have settled on `-DENABLE_FEATURE=ON`
- All developers should be aware of the dependencies of their applications *and* if they are optional



Multiple KDE Installs

- KDE 3 and 4 on the same system
- Stable (trunk/snapshot) and development install of KDE
 - Useful for developers who want a stable desktop
 - Useful for power users trying bleeding edge builds
 - Installed in a separate prefix
 - Default config dir set at kdelibs compile time
- Mostly works but there are some issues
 - KDE 3 session with KDE 4 installed in /usr
 - Build systems don't always respect PATH



Issues With Packaging

- Plasma can be tough to debug—tips?
- Amarok—MySQL on x86_64 and small systems
- The kdesupport packages
- Language bindings can be tough



Distro Patching

- Are we getting better at sharing?
- Now area in techbase for sharing
- KDE 3.5—can distros share bigger patch sets?
- What do we do to support legacy users?



Patches—Customizations

- Gentoo has a philosophy of using minimal patch sets
- Struggled with the branding question
- More and more packages are getting branding
- KDE is following this trend
- How much should we patch and customize KDE?
- Could generic branding mechanisms be added?



Language Bindings

- Bindings are getting more important
- Have always been tougher to package
- Python appears to be the most important
- Kdebase apps are beginning to use bindings



Basic Operations that we need

System
Services

Network
Management

User
Management

Boot Loader
Management

Xorg
Management

Disk
Management

Time
Management

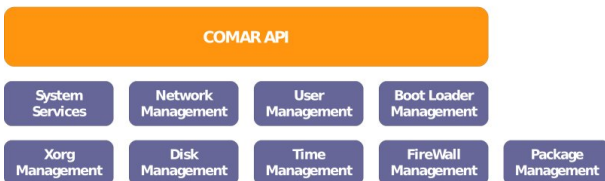
FireWall
Management



Package Manangement

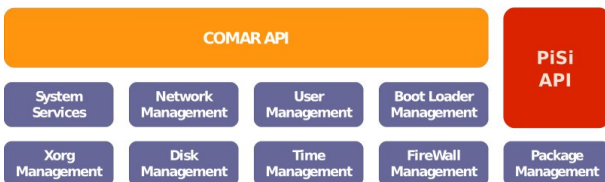


Comar: Configuration Manager

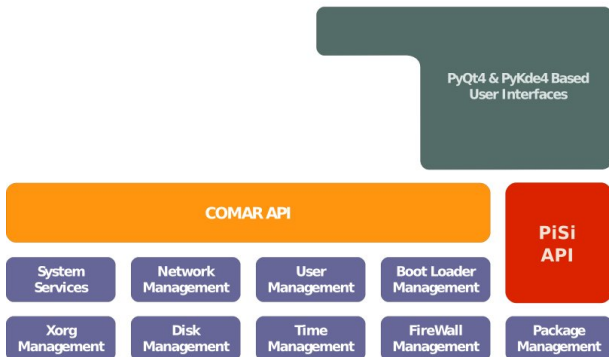


Pisi: Packages Installed Successfully as Intended

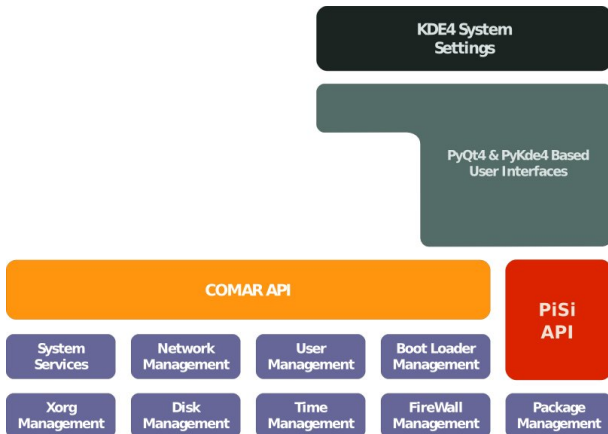
Pisi: Packages Installed Successfully as Intended



Managers suitable for operations

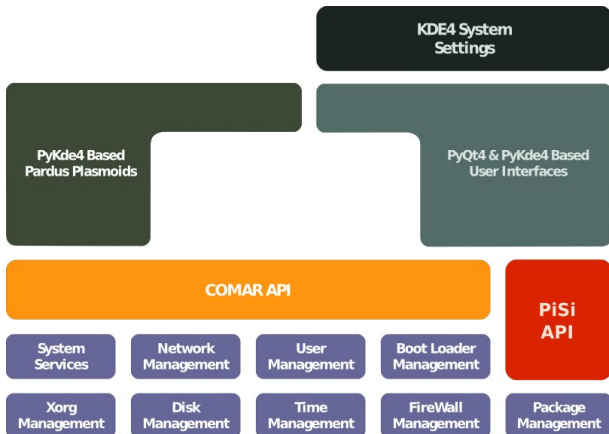


KDE 4 System Settings

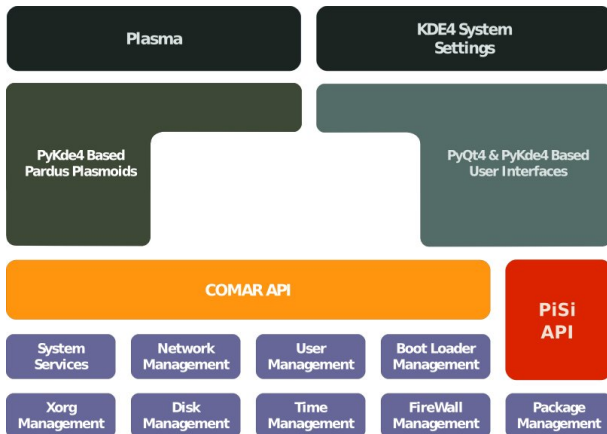


Plasmoids suitable for operations

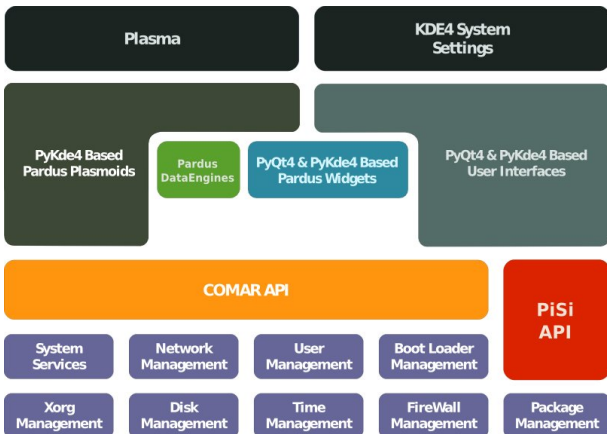
Plasmoids suitable for operations



KDE 4 Plasma

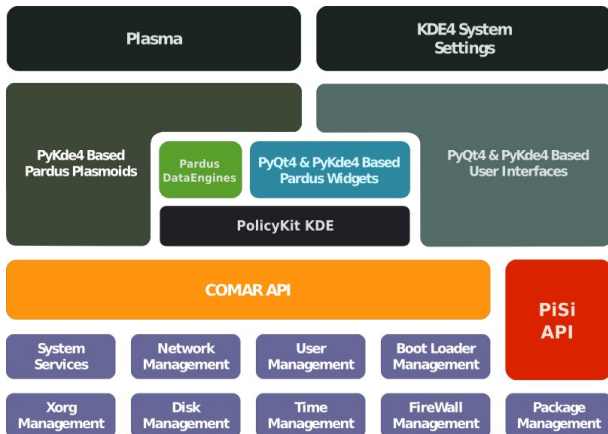


Data engines and shared widgets



Authentication: PolicyKit KDE

Authentication: PolicyKit KDE



Upstream KDE Developers

- How do we currently interact?
 - IRC
 - Mailing lists
 - Bugzilla
 - Blogs
 - Conferences
- Do we need to improve communication?
- What is the optimal way to request urgent patches or ask questions?



Linux Distributions

- Could we share patches more effectively?
- Ensure general patches are pushed upstream quickly
- Could we better coordinate security and other bug fixes?
- Could we be more open?



How Can We Improve?

- Going forward how could we improve interactions?
- The open source world is filled with passionate people
 - Don't let it get personal
 - Disagreements should try to remain technical
 - Most of us are volunteers—time is in short supply
- Interaction can be tough to get right
 - Don't be an island



Thank You

- We would like to thank you all for listening
- This presentation was created in \LaTeX using the 'beamer class
- Could we work more closely?
- Questions?

