

# Custom Debian Live CDs

Debian, CDDs and Morphix:

## The road to World Domination

(...well, one of the roads at least)

Alex de Landgraaf

# Overview

- Why Live CDs?
- Introduction to Morphix
- Live CD architecture
- Autobuilding – the way forward
- From CDD to CDL and back again

# Why Live CDs?

- Cooler: superior means of indoctrination!
- Faster: up and running asap!
- Easier: pre-selected, pre-configured packages on 1 CD/DVD
- Cons: limited architectures, large reliancy on hardware detection & configuration, painful to upgrade after installing to harddisk (most livecds ignore/discourage installing)

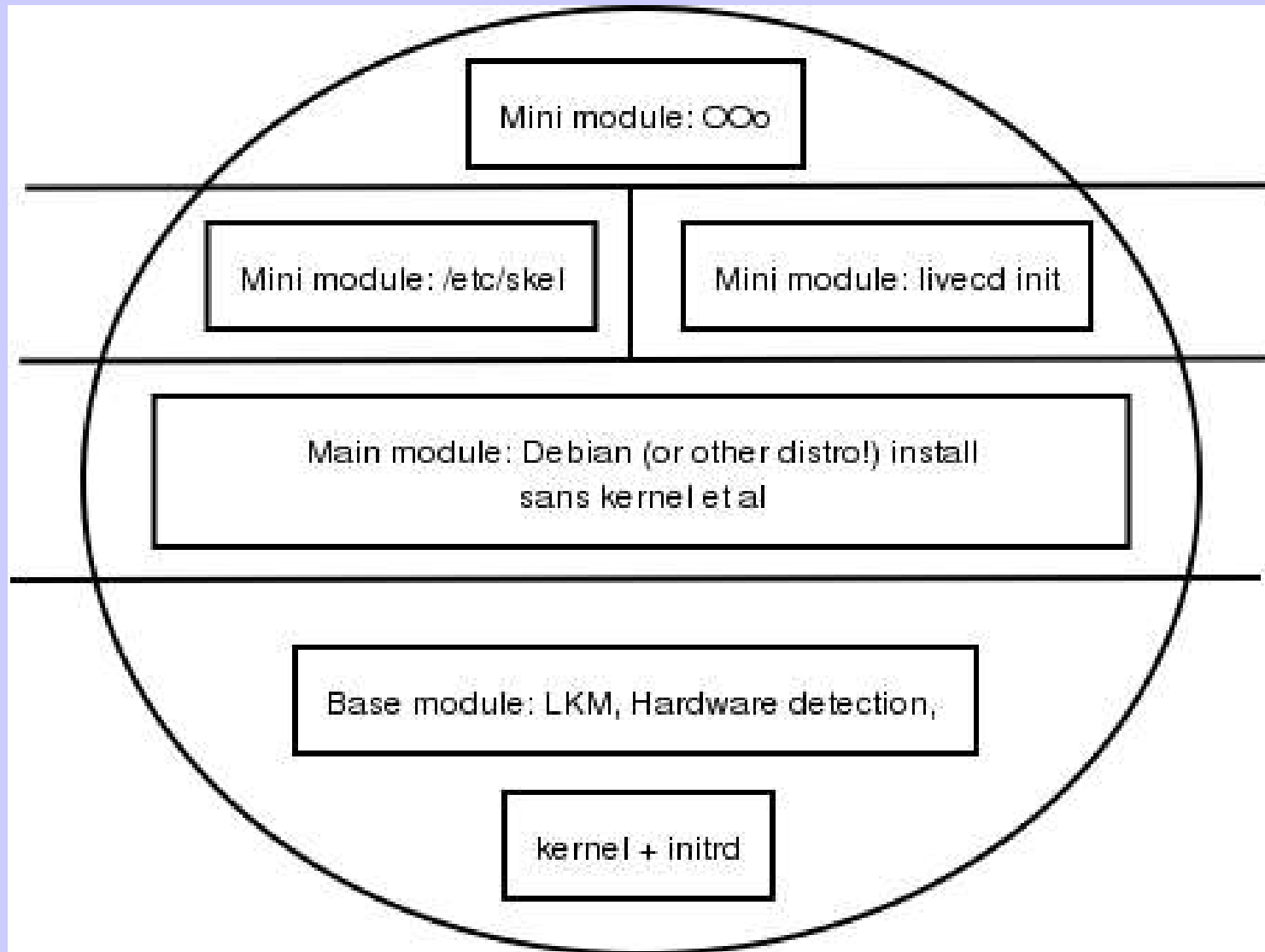
# Introduction to Morphix

- the standard for Live CD hardware detection: Knoppix
- PITA to hack away at: Let's morph!
- Need for inter-exchangeable, smaller, compressed images
- Result: flexibility for us lazy live cd builders, more choice for distributors / users
- Heaps of derivatives! (uhoh...)

# Morphix: GTK+ stuff

- Installer
- Partitioner
- Morphable/Morphix control panel
- some post-install config tools ported from RH/Fedora (reasonable state, not yet on official morphix livecds, haven't had time to release full isos)

# Live CD architecture, overview



# Live CD architecture, cdrom layout

- /base
- /mainmod
- /minimod (optional)
- /deb (optional, install .debs at boot time)
- /copy (optional, copy files at boot time)
- /exec (optional, execute scripts at boot time)
- (can use xcdroast, mkisofs, or any cd burn tool to put these together)

# Live CD architecture, benefits

- Update separate modules
- Exchange of modules with other live cds
- Cleaner install possible
- Overlaying of modules possible

# Live CD architecture, translucency

- Kernel module (2.4 only... for now)
- Works by redirecting syscalls: lovely hack
- Maps one filesystem tree over another
- Result: makes read-only trees writeable (by mapping to tmpfs), makes overlaying separate modules possible

# Autobuilding modules

- Building modules manually... brrr...
- m(odule)maker tool: one file -> one module
- XML layout: contains info about module, list of repositories for debian packages, list of (meta)packages (handles dependancies)
- Works for mainmodules, base almost done, separate minimod generator (will be integrated in time). mmaker is still under quite some development, documentation lacking

# Template example: gnome.xml

```
<comps> <group>
<name> Morphix Gnome</name><description>...</description>
<type>mainmod</type><suite>sid</suite>
<repositorylist>
  <repository type='debian'>ftp://ftp.debian.org/debian</repository>
  ... (your repository here)...
</repositorylist>
<packagelist>
  <packagereq>gnome-core</packagereq>
  ... (your packages here)...
</packagelist></group></comps>
```

# Template example: and now?

- `m maker gnome.xml gnome.mod`

(this takes a while, go drink some coffee...)

- Add, replace `/mainmod/*.mod` with `gnome.mod`
- Rebuild iso, et voila: a new custom debian live CD... hopefully

(should get easier, as module building is basically a library: having a GUI frontend will even make this fun :-)

# Combining CDD and CDL:

- CD building for CDD's: repositories and package list
- Live CD building for CDL's: repositories and package list
- Combined world domination! Woot!

# What's next?

- Get out a base with Linux 2.6 asap
- Extend mmaker framework for CDL: making Debian Live CDs should be as easy as possible
- Installer, Partitioner still need some work
- Upgrading hell... (would need separate repositories, more general CDD issue)
- Some place to store cdrom images (sf.net loves us... not)

# Nearly done, hang in there...

Questions?

More info:

<http://www.morphix.org> (wiki, howtos, forums)

<http://am.xs4all.nl/phpwiki/index.php/ModuleMaker>

debian-custom or morphix mailinglists

#morphix, #debian-custom, #debian-np on  
[irc.freenode.net](http://irc.freenode.net)

and you can contact me 24/7:

alextrreme at xs4all dot nl