Too Many VCLPlugs

Caolán McNamara,
Red Hat
2016-09-07

- Too Much Stuff

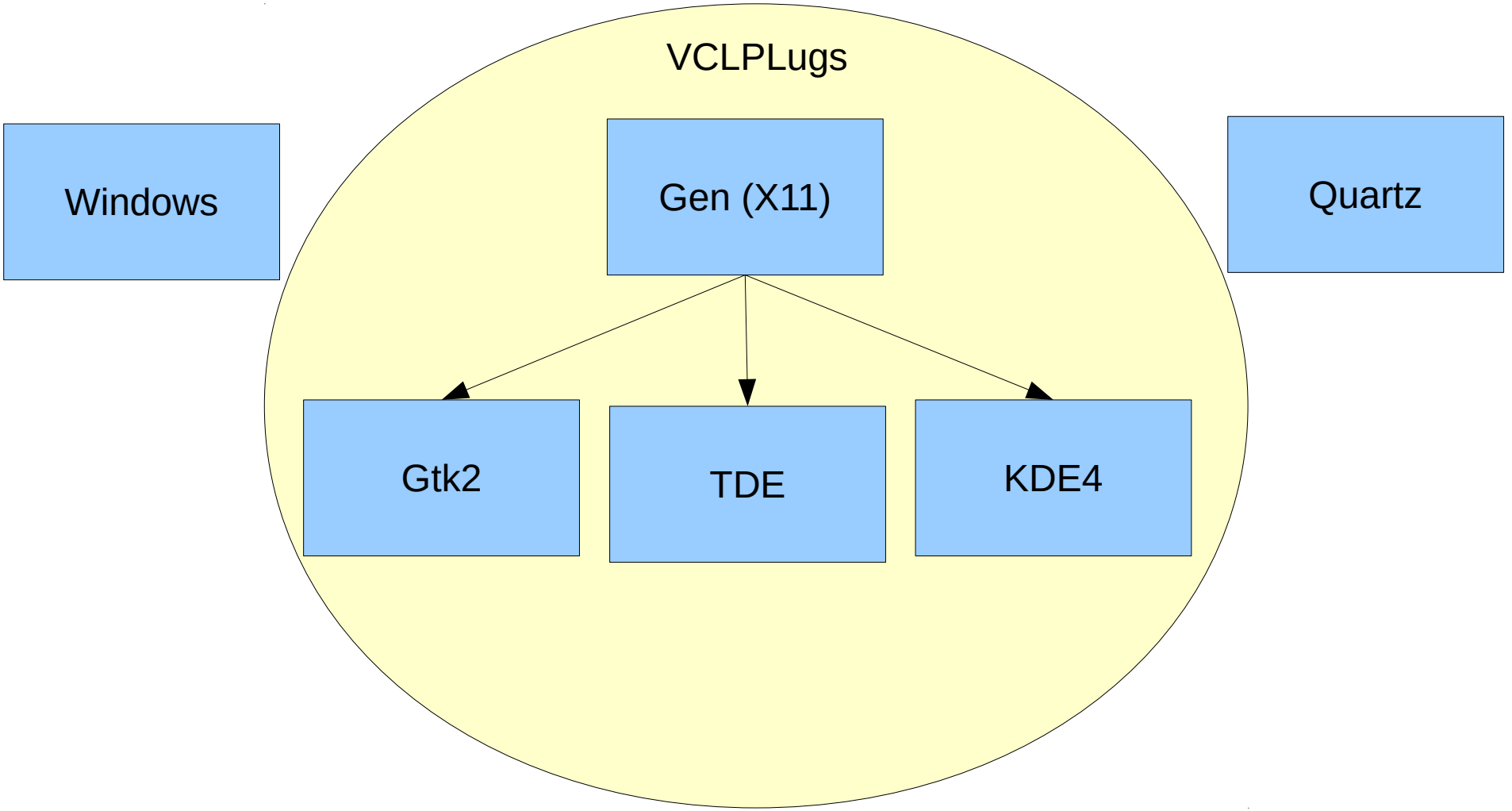
SalInstance, SalFrames

- Each platform has to implement a SalInstance
- A SalInstance mostly consists of Create/Destroy pairs for SalFrames, SalPrinters, SalVirtualDevice, etc.
- Each platform has to provide concrete implementations of SalFrames, SalPrinters and SalVirtualDevices, etc
- SalFrames are system windows (X11 Window)
- SalVirtualDevices are non visible drawables/buffers (X11 Pixmap)

SalGraphics

- SalFrames and SalVirtualDevices must implement AcquireGraphics which returns a SalGraphics
- Each port has to implement a SalGraphics which enables drawing to the SalFrame/SalVirtualDevice
- Apis like drawLine, drawRect
- Some of the drawing apis are optional
- Some of these apis are somewhat “fat”
  - DrawEPS
  - isNativeControlSupported/drawNativeControl for native widget framework

VCL Implementations Then



Gtk2

- GtkSalGraphics inherited from the X11SalGraphics
  - Mostly reused X11 code, except added native widget support
- GtkSalFrame inherited from X11SalFrame
  - In many places grabbed the underlying xid of the GtkWindow and tweaked it directly
- Printing inherited from generic cups backend
- Entirety of cut-and-paste and draw-and-drop inherited from X11 equivalents.

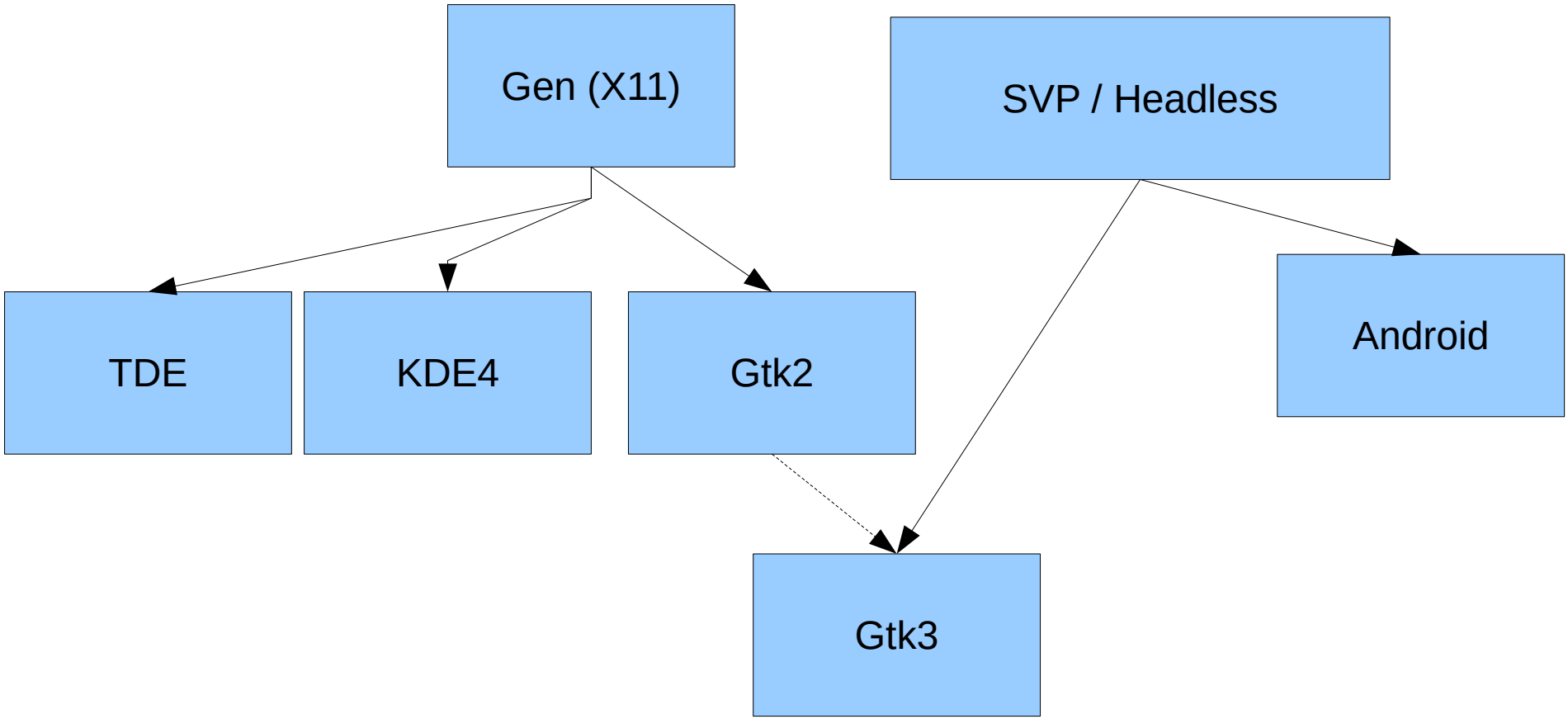
Gtk3

- Obviously lots of overlap with the Gtk2 vclplug, where we didn't just use X directly
- But we need something to back our virtual devices
- And we also can no longer draw directly to windows
- Need a SalGraphics implementation that can draw into those replacements.
- Have to implement in native gtk a bunch of stuff that previously inherited from generic X, e.g. d-n-d, c-n-p.

Headless/svp

- We have a headless mode
- Originally intended for server applications
  - e.g. Document conversion hubs
- Forms a part of the android port and libreofficekit tiled render work
- Headless mode is implemented as a SalInstance etc
- Implements a virtual device bitmap buffer and a SalGraphics impl to render to it

VCLPlug Implementations Now



Text Layout 1

- Text layout done with Harfbuzz on Linux
  - Harfbuzz replaces ICU layout
  - Used to have another “Simple” layout which is gone
- A generic “GraphiteLayout” engine
- Three text layouts on Windows
  - Uniscribe layout
  - “Simple” layout
  - And a Graphics Specific Windows one
- MacOSX CoreText layout

Text Layout 2

- Unifying low-level text layout using HarfBuzz
  - Akash Jain, Khaled Hosny
- GSOC 2016
- Looking forward to seeing that

Text Rendering Linux

- Text is rendered with cairo under all vclplugs

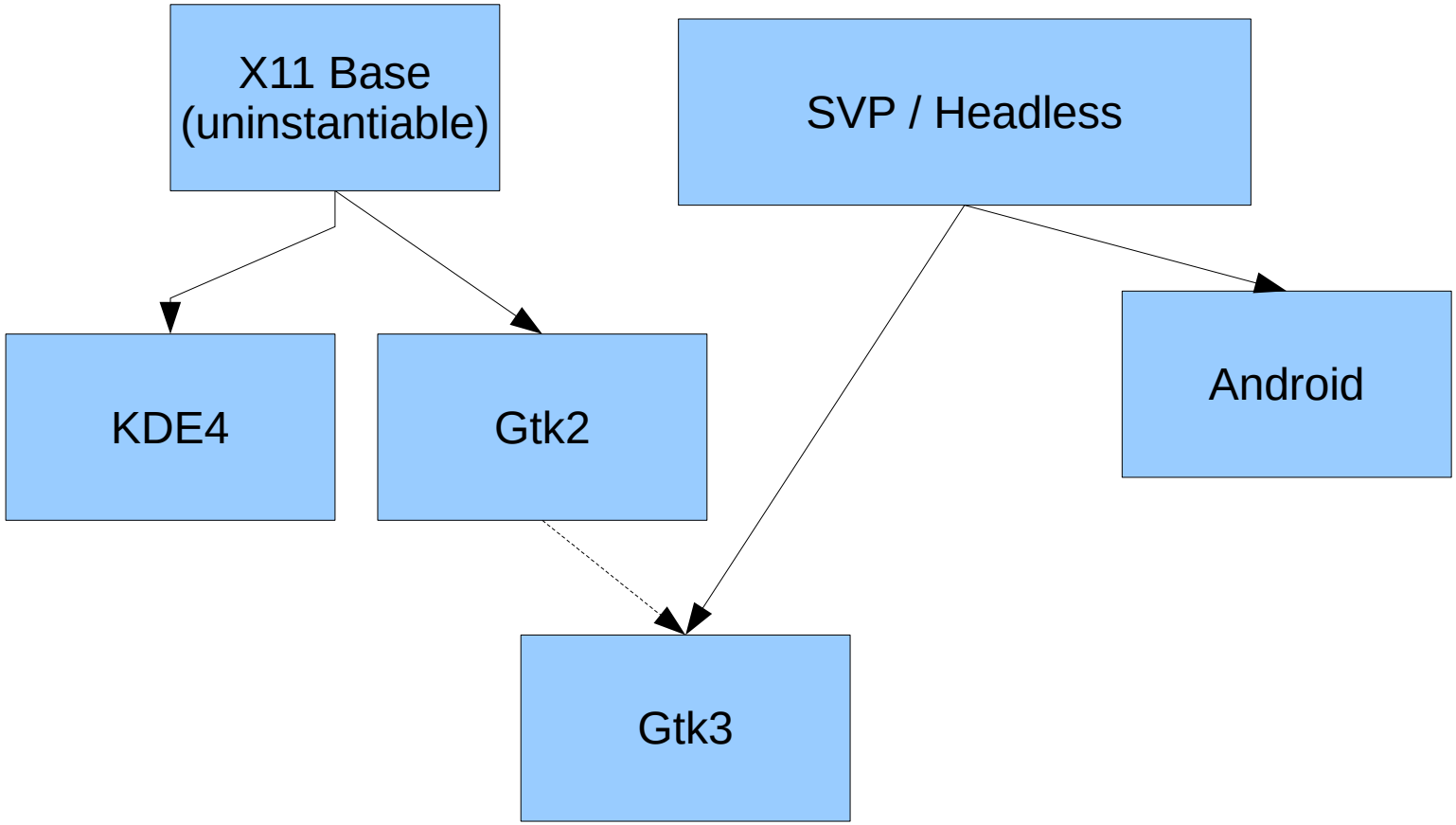
Graphic Rendering Linux

- svp graphics are rendered with cairo
- gen graphics are rendered with X
  - except when they are rendered with cairo, maybe
  - unless they are rendered with opengl, sometimes

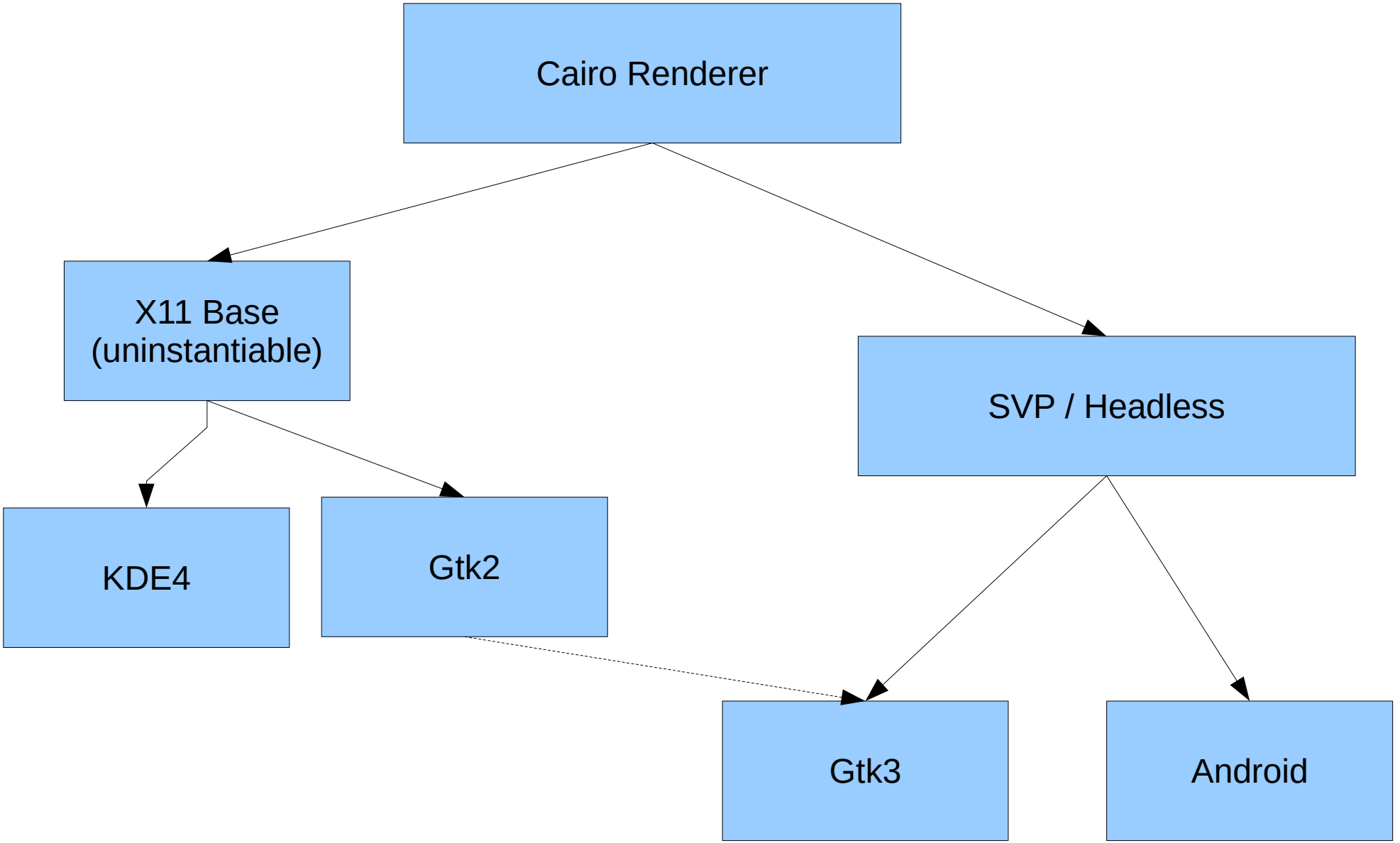
Drop archaic elements

- Remove direct instantiation of the gen vclplugin
  - default to gtk2 in the absence of anything else
- Drop tde vclplug, controversial ?

That gives us something like this



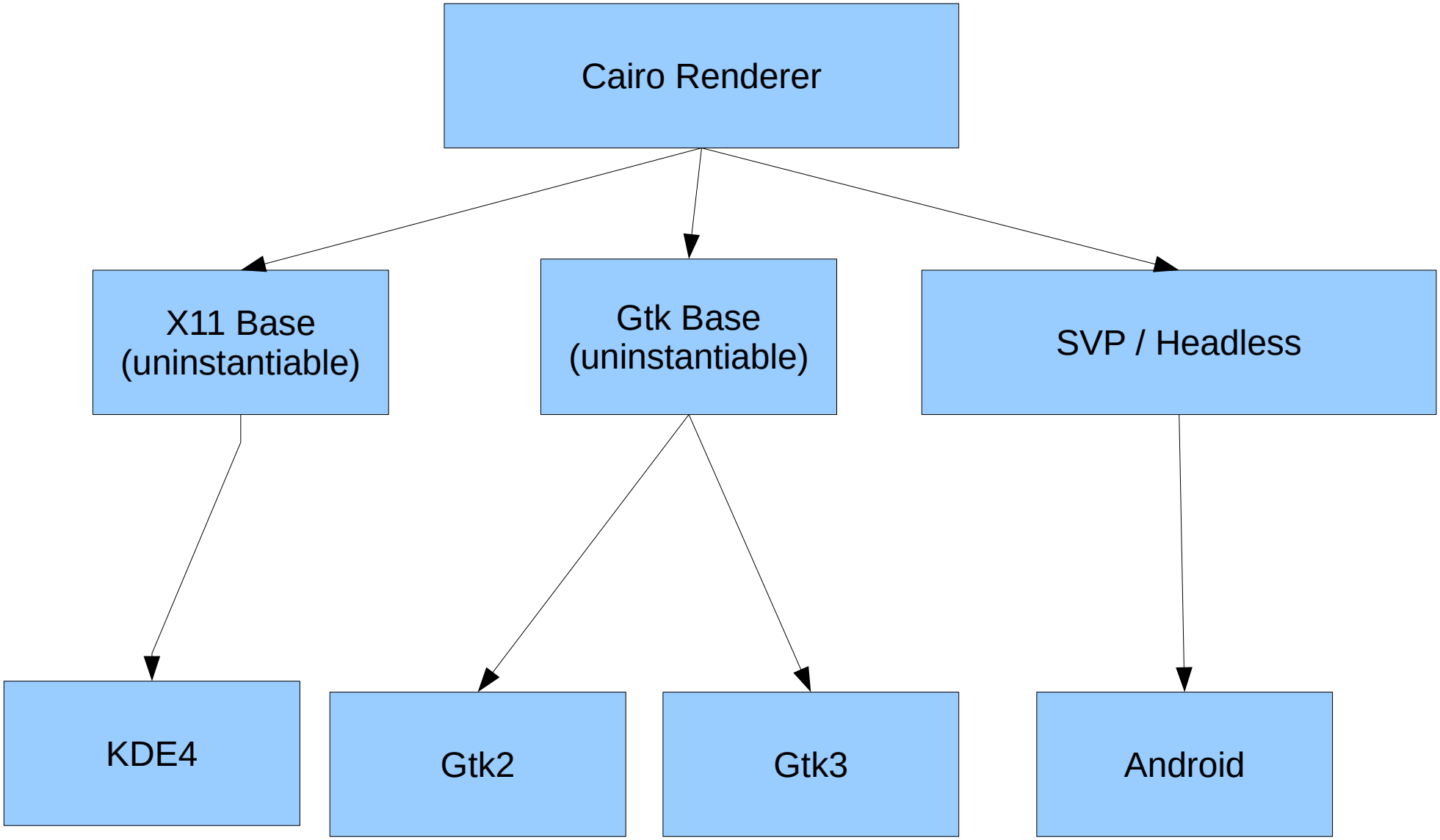
Replace X drawing impl with Cairo impl



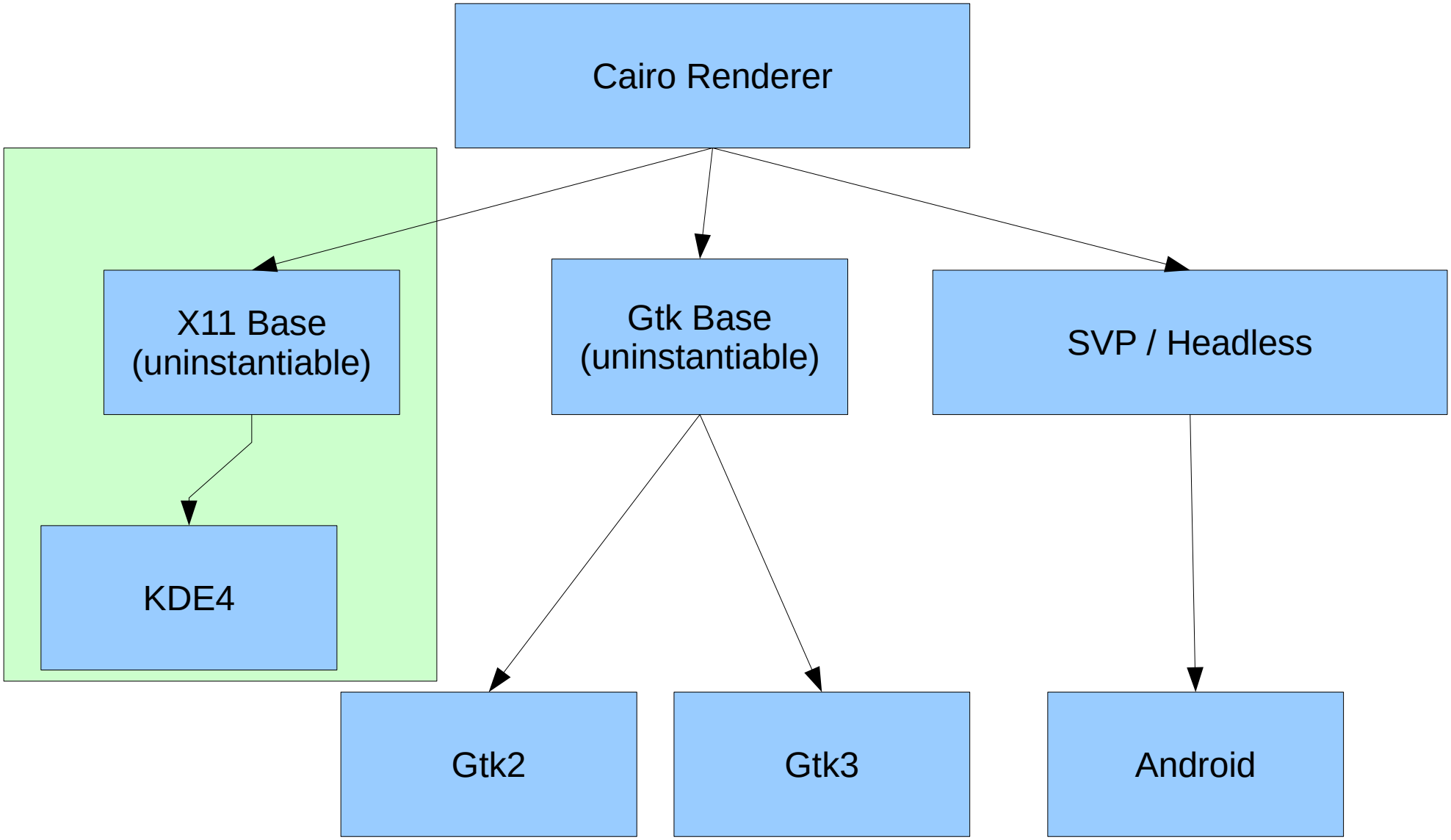
Decouple gtk2 impl from the X impl

- Pull out code implemented for gtk3 using apis that exist in gtk2
- So, effectively complete the gtk2 vclplug with the gtk3 vclplug code and remove the inheritance from the gen/X one

Looks something like this



What's KDE's future?



KDE

- I’m happy enough to leave it alone
- Perhaps merge X11 specific stuff into it to remove a layer
- Definitely needs love from someone to separate it from its X impl baseclass underpinnings

What's the gain

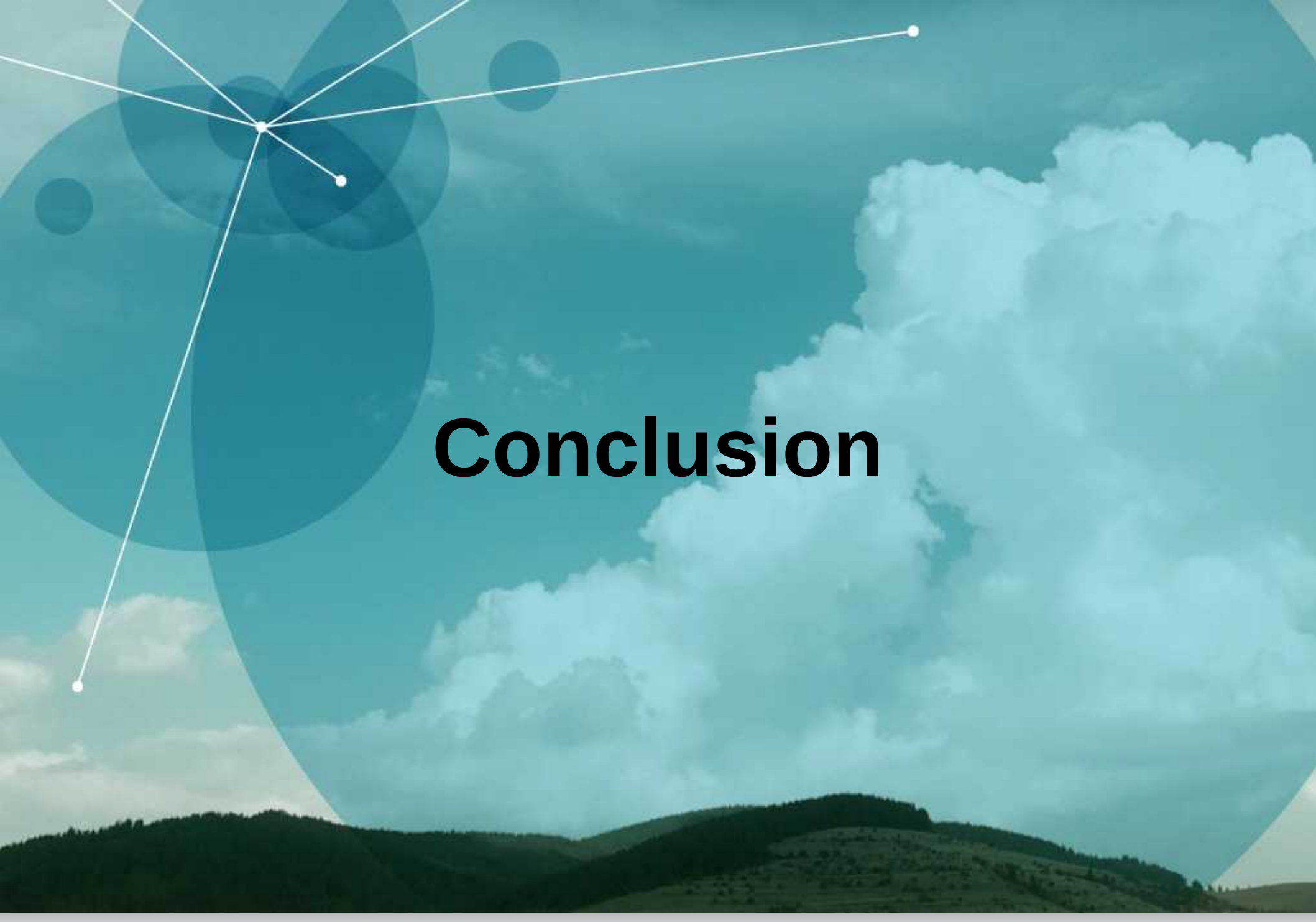
- One base graphics rendering layer for Linux
  - A set of current optional vcl rendering paths will now always exist, so the old ones can go
- Same rendering path for gtk2, gtk3, headless, kde4
- Less code
- Less complexity
- Share gtk3 c-n-p and d-n-d glitches in gtk2
- Does cairo canvas still make sense if generic vcl canvas is backed by effectively the same cairo calls ?

Another possibility

- Provide Linux universal build only as flatpak
- Delete all vclplugs except gtk3

Go the whole hog

- Same base graphical rendering with cairo on all platforms
- Use gtk3 on all platforms
  - Fix gtk3 theming on various platforms
  - Maybe keep the file pickers



Conclusion

Proposal

- SAL_USE_VCLPLUGIN=gen would do nothing
- Replace X drawing impl with the svp drawing impl
- Drop the inheritance from X impl for the gtk2 impl
- Fill in missing gtk2 bits with shared code from gtk3 impl
- Drop TDE