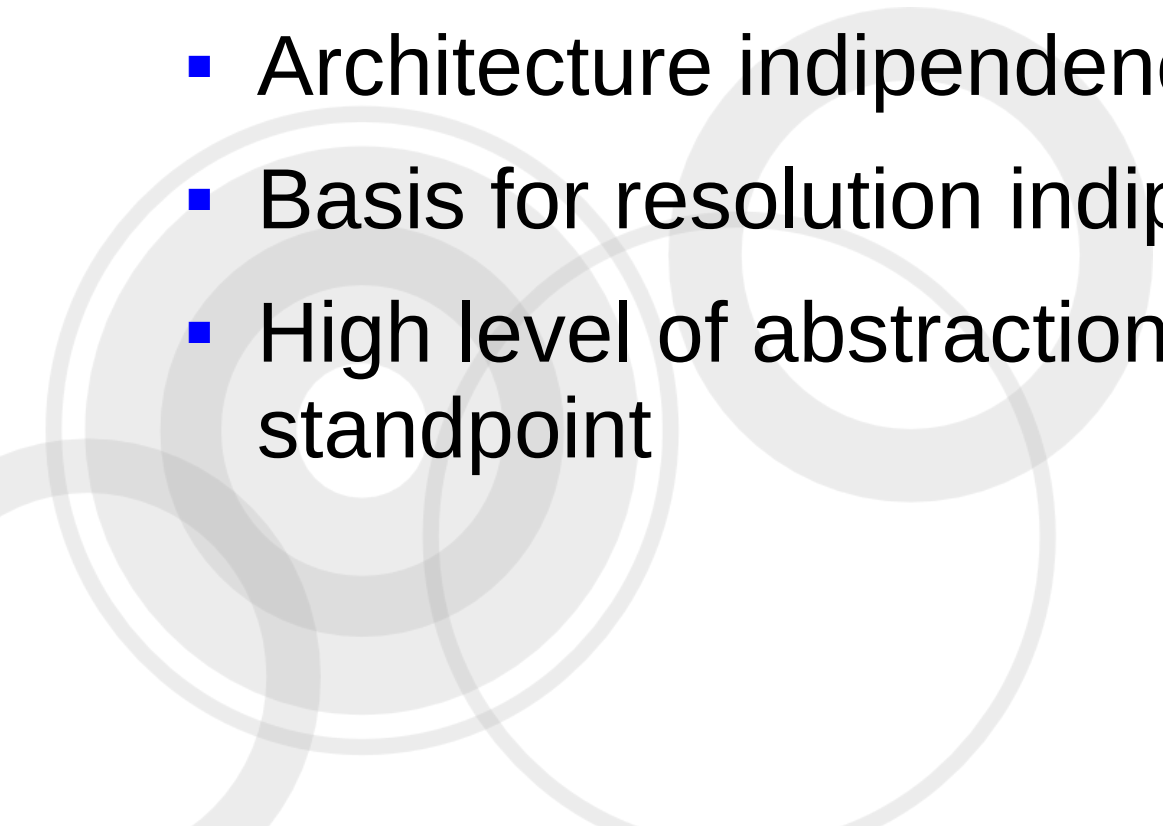




Themes, shells and other animals



Themes

- Plasma graphics is heavily based on the Svg standard
 - Easily themeable
 - Architecture independence
 - Basis for resolution independence
 - High level of abstraction from an API standpoint
- 
- A decorative graphic in the bottom-left corner consisting of several overlapping, semi-transparent circles of varying shades of gray, creating a layered, circular pattern.

Using Svgs from themes

- They are installed in the KDE prefix/share/apps/desktopthemes/themeName
- Subfolders: dialogs, widgets, applet names, opaque, locolor
- Loading them:
 - Plasma::Theme abstracts the loading of the proper path, with proper fallbacks
 - `m_svg->setImagePath("widgets/background");`

From packages

- A scripted applet can load a svg from its own package with `Package::filePath()`
 - In javascript you do `plasmoid.findSvg("foo")`
 - Will search `foo.svg(z)` in the images subfolder of your plasmoid

Some advices

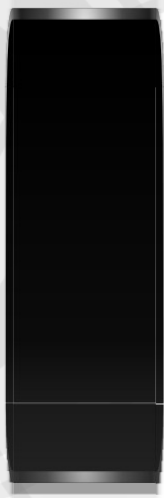
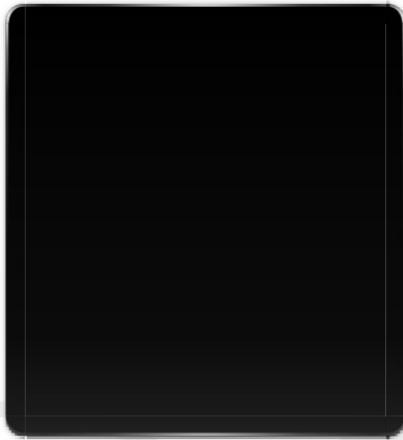
- Try to use graphics from the theme and default widgets as much as possible
- Install your own graphics (c++ case) or use from the package (scripted case) only when really necessary
- Try to make visually work your additional graphic with different themes (at least dark vs light)

Implementation details:

Plasma::Svg

- The central machinery that renders svgs, eases use of QSvgRenderer
- Several optimizations (a renderer is quite heavy):
 - Shared: for each svg file a single shared renderer
 - Cached: saves rendered svgs on disk: avoids completely creation of renderers in some cases
- Use it: several paint() functions, takes a Qpainter as argument

Plasma::FrameSvg



- Most Plasma theme elements style mostly-rectangular widgets
- Svg is scalable, but... not so much :)
- We want to keep the radius of rounded borders intact and each element exactly aligned to the pixel grid
- Plasma::FrameSvg comes to rescue

From a graphics standpoint



- We have svgs made of 9 elements: center, edges and corners
- The center can be scaled as we wish
- Corners are NEVER scaled
- Horizontal edges are scaled in the x axis, vertical in the y axis instead

From a code standpoint

- FrameSvg is a subclass of Svg, so usual setImagePath()
- ResizeFrame scales the final composition with said rules
- PaintFrame() paints the whole composition
- A svg can have multiple sets of 9 elements sets, with a different prefix in their name, switch with setElementPrefix()

Part 2: shells - Plasma-mid

- Plasma-mid is an experimental interface for mobile internet devices and other constrained little devices.
- Work began at Akademy-2008, mostly by Aaron, some work by Sebas in mid-specific containments, some patches by me and i'm forgetting somebody, sorry :)
- The improved sytem-tray we have now actually derives from thoughts and design around plasma-mid

Plasma-mid

- What does work
 - Plasma-mid is a barebone plasma app with a main view and a panel view in the same window
 - Loads correctly applet layouts from config files
 - Supports panels in the 4 locations
 - The main view can correctly switch containment
- What does not
 - More than 1 panel
 - Any configuration interface, hand editing configuration files is necessary

One size fits.. not much

- Different devices with different hardware capabilities will need to have a different-looking interface.. different applets layout same app
- Touchscreen or not, keyboard and mouse or not, makes needed stuff and Fitt's laws slightly different
- Also, a really tiny single-purpose device could have just plasma, a more general one like the Eee mixes up plasma and traditional apps

Current demo

- A plasma-based interface that mostly manages traditional apps
- Right vertical panel thumb friendly with an activitybar, an “activewindow”, a sytemtray and a clock
- 3 activities: an icon list to launch apps, a fullscreen taskbar and a traditional widgets space
- Ideally the windowmanager would have to be very barebone, even more than matchbox

Current demo

- It's a full screen non-desktop window, with the panel area reserved, so clicking on the panel brings the plasma interface on top
- Both the panel and the main screen containments are simplified mid-oriented code
- It's done with already existing applets, maybe not really adapt and not so pretty but we have a basic working thing from where we can start to build something

Media center

- Current media center solutions are applications that aims to turn a pc in an easy attach-to-your-tv single purpose box
- The characteristics of Plasma we're developing both in terms of good looking ui and network services transparency would make this possible, but...

We can do more than that

- Blend media center related stuff and the traditional desktop (media center activity?)
- Control a media center from an another machine with your plasma desktop (or even your plasma-mid portable device ;)
- Scripted widgets to control the media could be sent across the network
- Anything else that can't think about now

What there is now...

...And what realistically i would like to get in KDE 4.3:

- A video widget in libplasma usable even with few lines of javascript
- A media player plasmoid with basic audio/video playback capabilities and a dbus interface to be remote controlled

Questions?

