XCanvas

Migrating OpenOffice to state-of-the-art graphics output

Thorsten Behrens
Software developer
Sun Microsystems, Inc.

What's (X)Canvas?

- 'X' because it's a UNO interface
- new rendering subsystem for OOo
- going to replace VCL's OutputDevice for rendering application content:

1. Impress slideshow (OOo 2.0)
2. Draw/Impress edit views (planned for OOo 2.x)
3. Calc/Writer/Math (OOo 3.0?)

What's Cairo Canvas?

- Implementation of OOo's XCanvas interface
- Based on the new high-quality graphics library cairo
  - > Cairo is increasingly gaining grounds as a high-quality rendering backend for unixoid systems – there are ports of the mozilla suite, gtk, scribus, inkscape and other applications under way.

The Issues that Sparked XCanvas

- Core functionality was missing in VCL:
  - anti-aliasing
  - (affine) transformations
  - color management
  - alpha compositing for all primitives
- Impress slideshow performance was bad

Reasons for XCanvas

- UNO API for rendering
- Significantly better portability
  - low impedance towards modern graphics APIs
  - easy to start with, for contributors
- Separation of concerns
  - XCanvas: rendering
  - toolkit: controls & windowing
- Speed
  - low impedance towards contemporary graphics hardware
- Quality
  - ubiquituous alpha compositing
  - anti-aliasing
  - color management

Key XCanvas Features

- Contemporary set of render primitives
- Multitude of backends feasible
- Stateless, concurrency-friendly design
- Flexible caching concept

Today & Tomorrow

OOo 2.0:

- VCL canvas

StarOffice 8.0:

- VCL canvas
- DirectX/GDI+ canvas

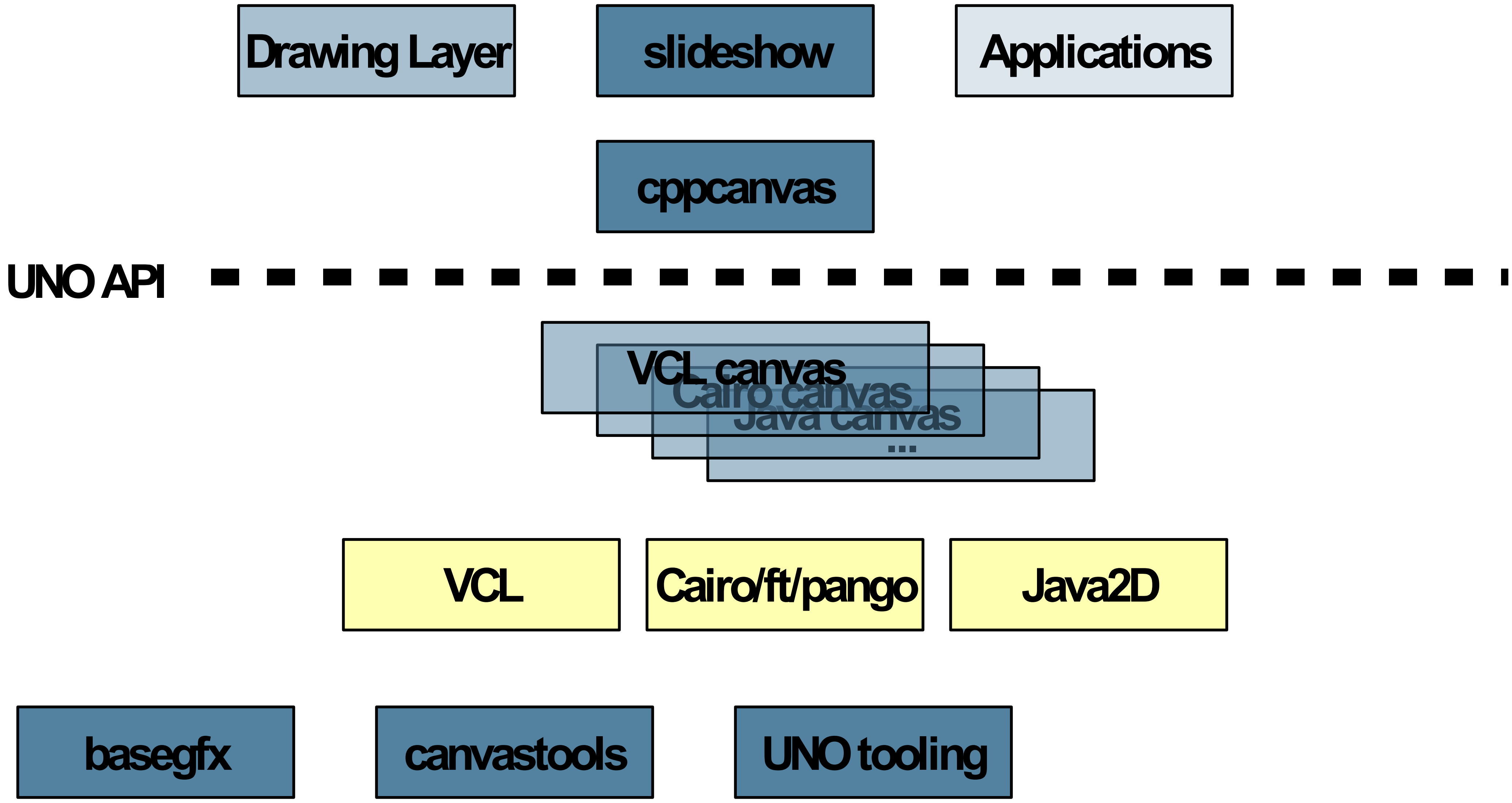
OOo 2.x.y:

- VCL canvas
- Cairo canvas
- VCL + backend (cairo/agg/gdi+ ...)

StarOffice 8++:

- VCL canvas
- Cairo canvas
- VCL + backend (cairo/agg/gdi+ ...)
- DirectX + backend (cairo/agg/gdi+ ...)

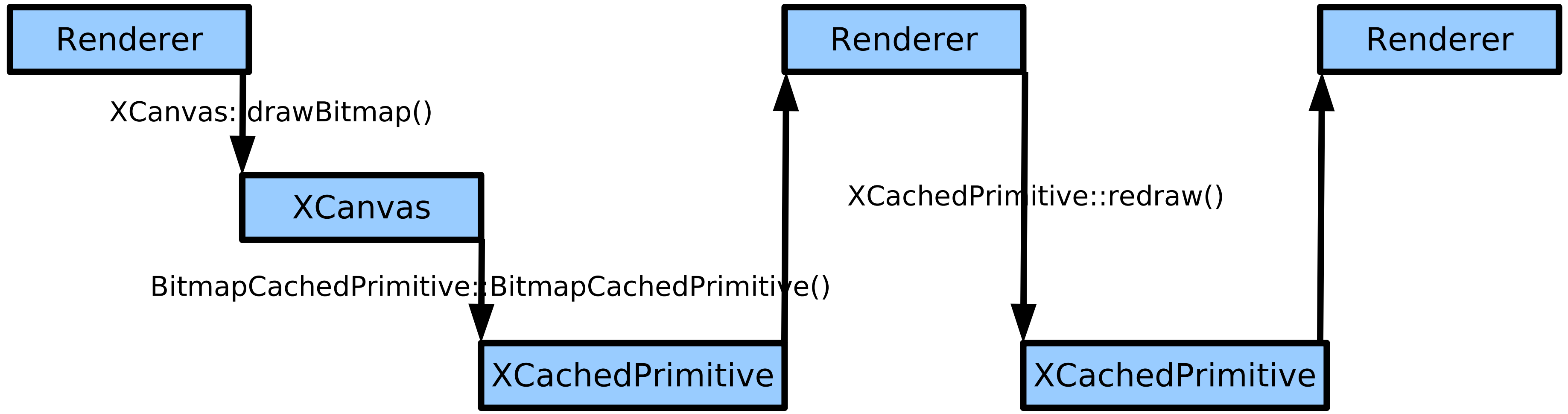
Canvas Modules



Core XCanvas Methods

drawPoint	Draw a point in device resolution on the device.
drawLine	Draw a line in device resolution width (i.e. one device pixel wide).
drawBezier	Draw a cubic bezier curve in device resolution width (i.e. one device pixel wide).
drawPolyPolygon	Draw a poly-polygon in device resolution line width (i.e. the lines are one device pixel wide).
strokePolyPolygon	Stroke each polygon of the provided poly-polygon with the specified stroke attributes.
strokeTexturedPolyPolygon	Stroke each polygon of the provided poly-polygon with the specified stroke attributes, fill the stroked outline with the specified texture graphics.
queryStrokeShapes	Query the polygonal representation of the stroke outlines, as it would be generated by the strokePolyPolygon methods.
fillPolyPolygon	Fill the given poly-polygon.
fillTexturedPolyPolygon	Fill the given poly-polygon with a texture.
createFont	Create a suitable font for the specified font description.
queryAvailableFonts	Query font information, specific to this canvas.
drawText	Draw the text given by the substring of the specified string with the given font.
drawTextLayout	Draw the formatted text given by the text layout.
drawBitmap	Render the given bitmap.

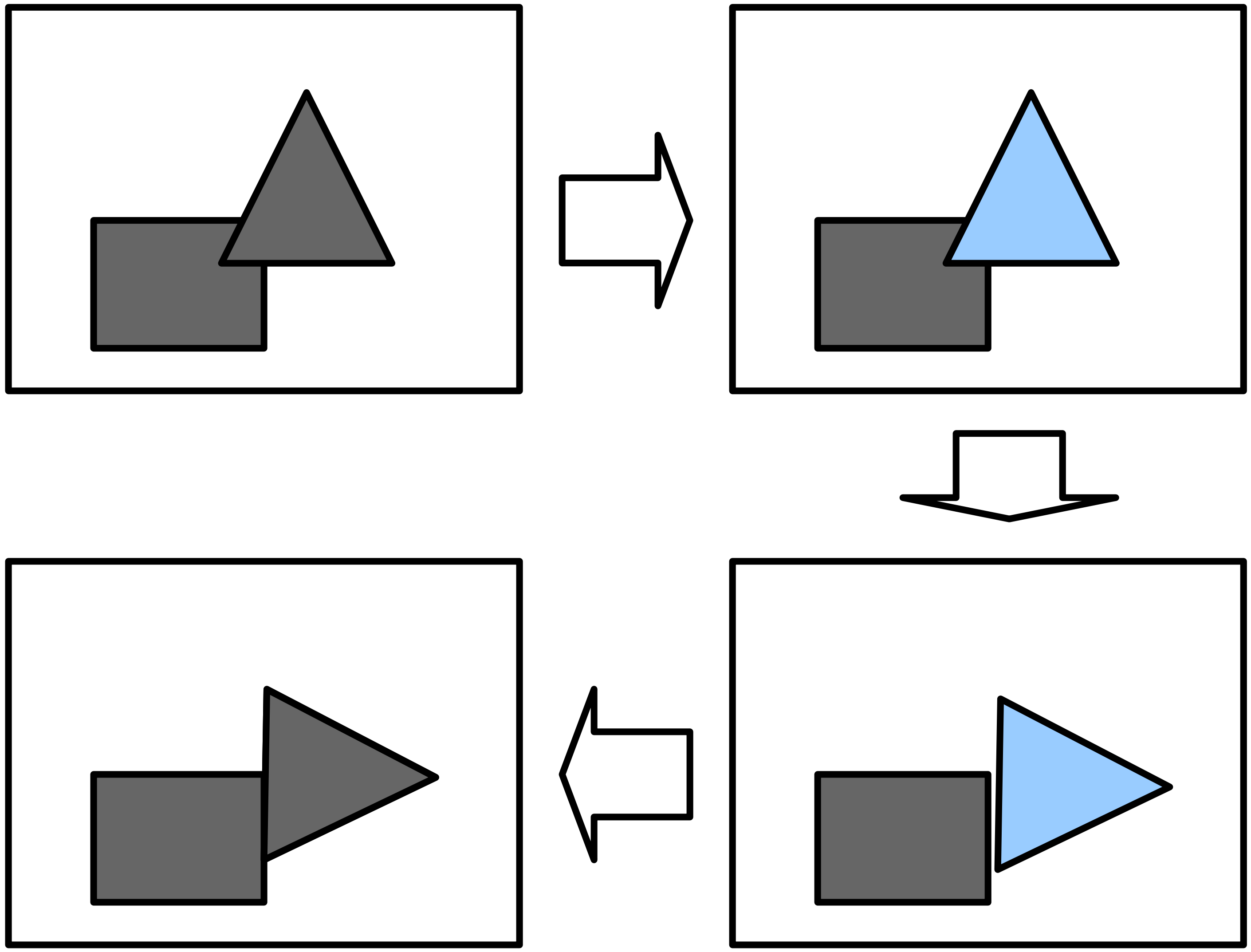
Call Sequence: Caching Output



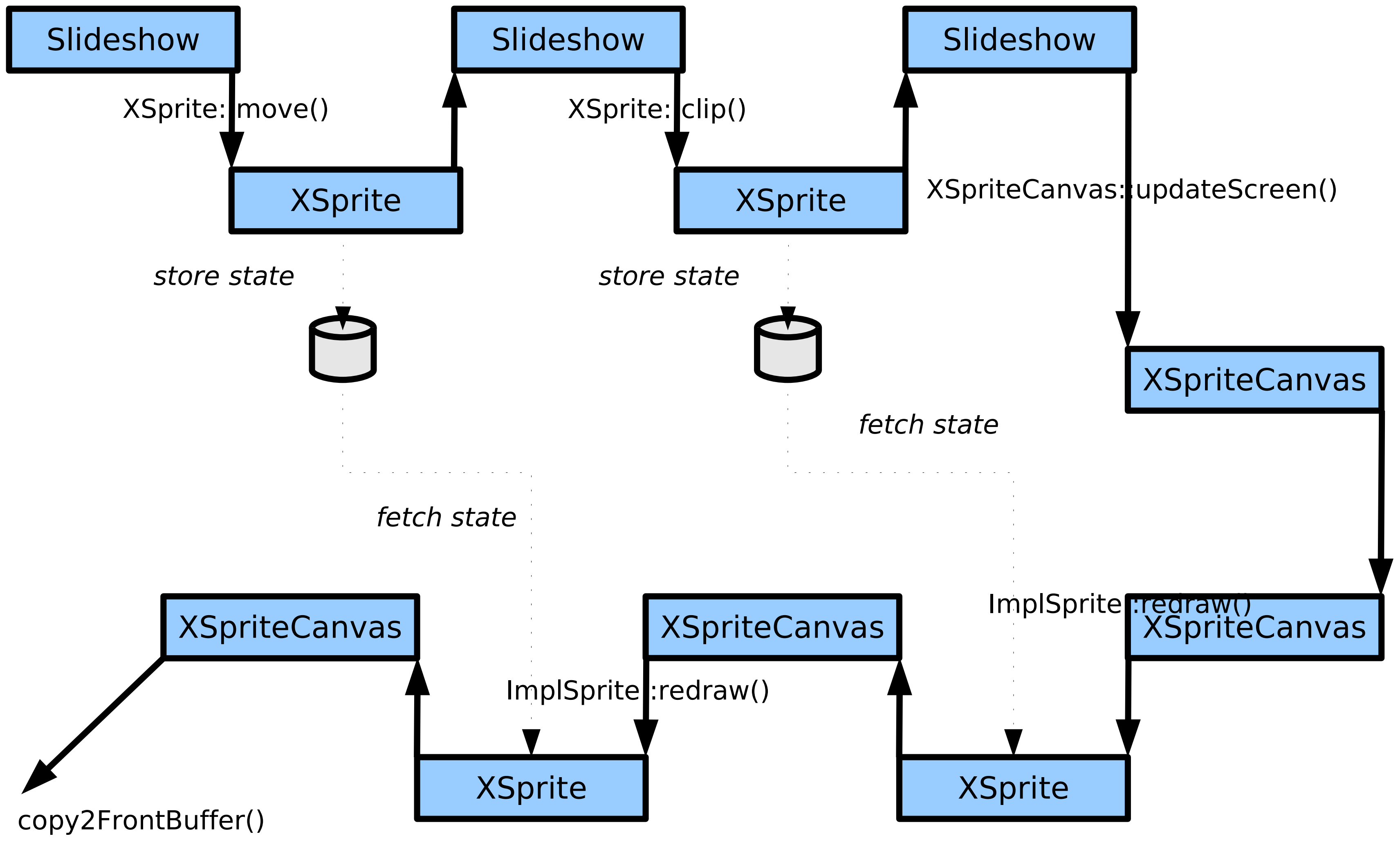
XSpriteCanvas Methods

createSpriteFromAnimation	Create a sprite object from the specified animation sequence. A sprite is a back-buffered object with its own, independent animation.
createSpriteFromBitmaps	Create a sprite object from the specified animation sequence. A sprite is a back-buffered object with its own, independent animation.
createCustomSprite	Create a custom, user-handles-it-all sprite object. A sprite is a back-buffered object with its own, independent animation.
createClonedSprite	Create a cloned version of an already existing sprite object. The cloned sprite always shows the same content as its original. Furthermore, cloned copies of a hidden original are never visible, although cloned copies of a visible original can of course be invisible.
updateScreen	Tells the sprite canvas to now update the screen representation. Required to display rendered changes to the canvas, and updates to stopped animations and XCustomSprites in general. This method will return only after the screen update is done, or earlier if an error happened.

Sprites & Animations



Call Sequence: Sprite Animations


Slideshow
XSprite::move()
XSprite
store state
Slideshow
XSprite::clip()
XSprite
store state
Slideshow
XSpriteCanvas::updateScreen()
XSpriteCanvas
fetch state
ImplSprite::redraw()
XSpriteCanvas
XSprite
XSpriteCanvas
ImplSprite::redraw()
fetch state
XSprite
XSpriteCanvas
copy2FrontBuffer()