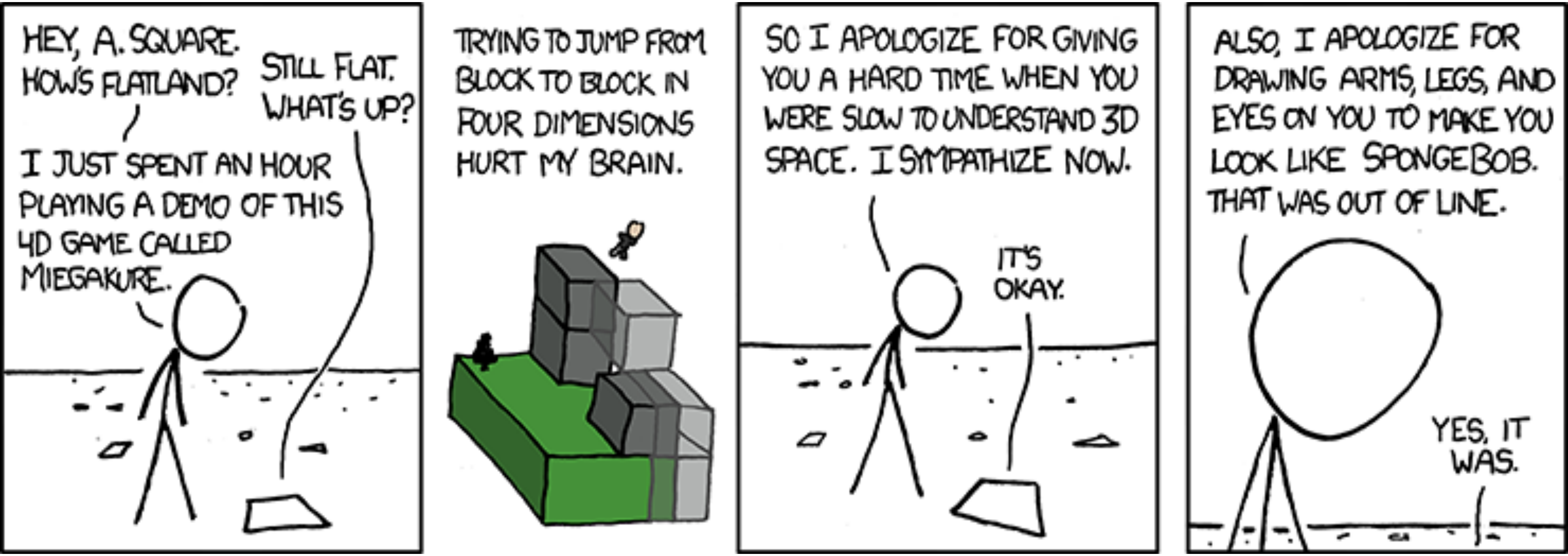
Cairo: 2D in a *3D* World


HEY, A. SQUARE. HOW'S FLATLAND?
STILL FLAT. WHAT'S UP?
I JUST SPENT AN HOUR PLAYING A DEMO OF THIS 4D GAME CALLED MIEGAKURE.
TRYING TO JUMP FROM BLOCK TO BLOCK IN FOUR DIMENSIONS HURT MY BRAIN.
SO I APOLOGIZE FOR GIVING YOU A HARD TIME WHEN YOU WERE SLOW TO UNDERSTAND 3D SPACE. I SYMPATHIZE NOW.
IT'S OKAY.
ALSO, I APOLOGIZE FOR DRAWING ARMS, LEGS, AND EYES ON YOU TO MAKE YOU LOOK LIKE SPONGEBOB. THAT WAS OUT OF LINE.
YES, IT WAS.

Who am I?

- part of Intel's Open Source Technology Centre, developing drivers for Intel Graphics. (Don't ask me what I think of Poulsbo.)

- De facto maintainer for Cairo

- Top 20 GNOME committer.

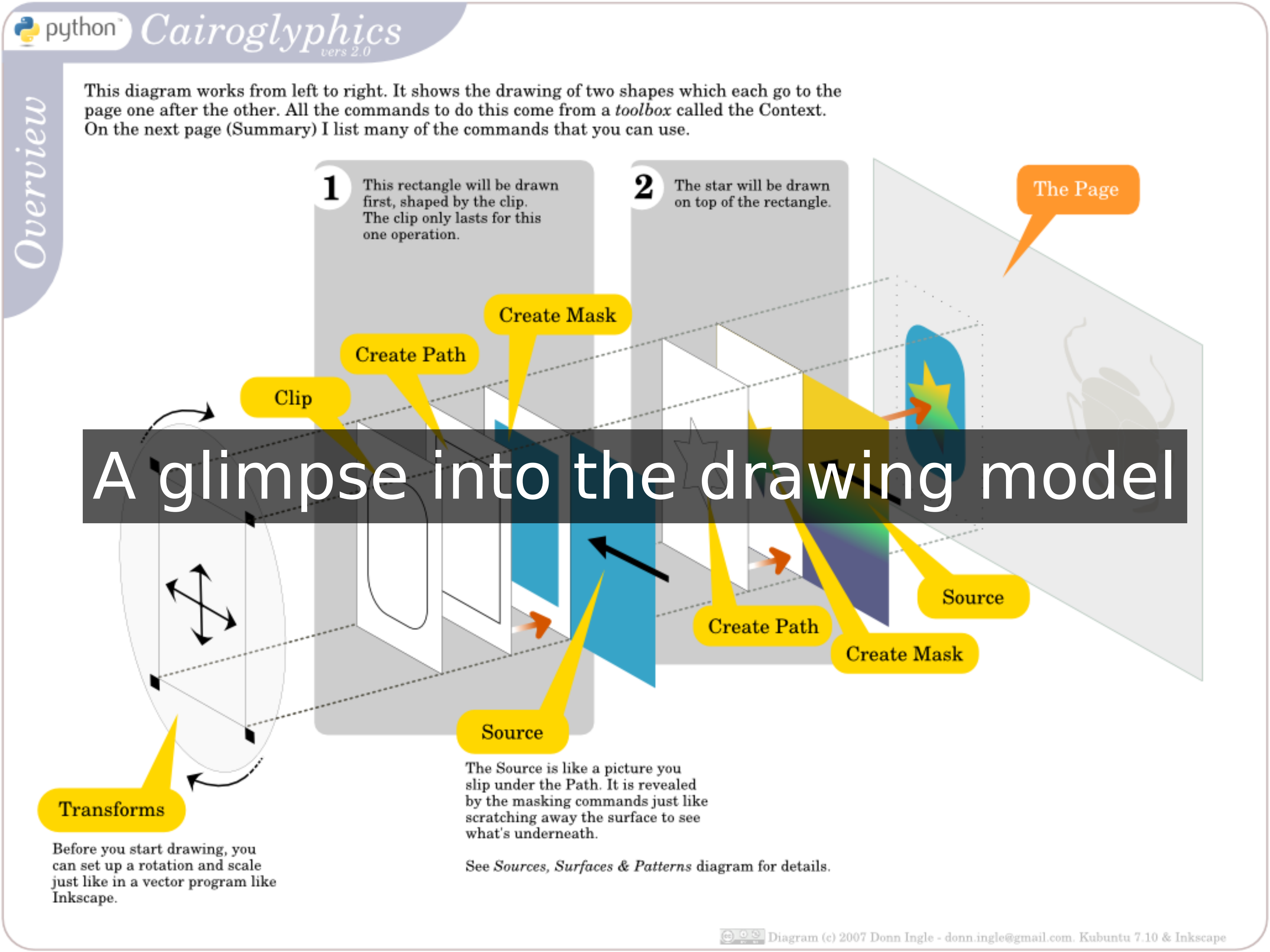
What is Cairo?

- Pen based drawing model developed from PS/PDF

- Originally convenience library for XRender

- Now a full Page Description Library
  - software rasteriser
  - native display acceleration
  - vector output for printing

A glimpse into the drawing model



This diagram works from left to right. It shows the drawing of two shapes which each go to the page one after the other. All the commands to do this come from a *toolbox* called the Context. On the next page (Summary) I list many of the commands that you can use.

1 This rectangle will be drawn first, shaped by the clip. The clip only lasts for this one operation.

2 The star will be drawn on top of the rectangle.

Source: The Source is like a picture you slip under the Path. It is revealed by the masking commands just like scratching away the surface to see what's underneath.

See *Sources, Surfaces & Patterns* diagram for details.

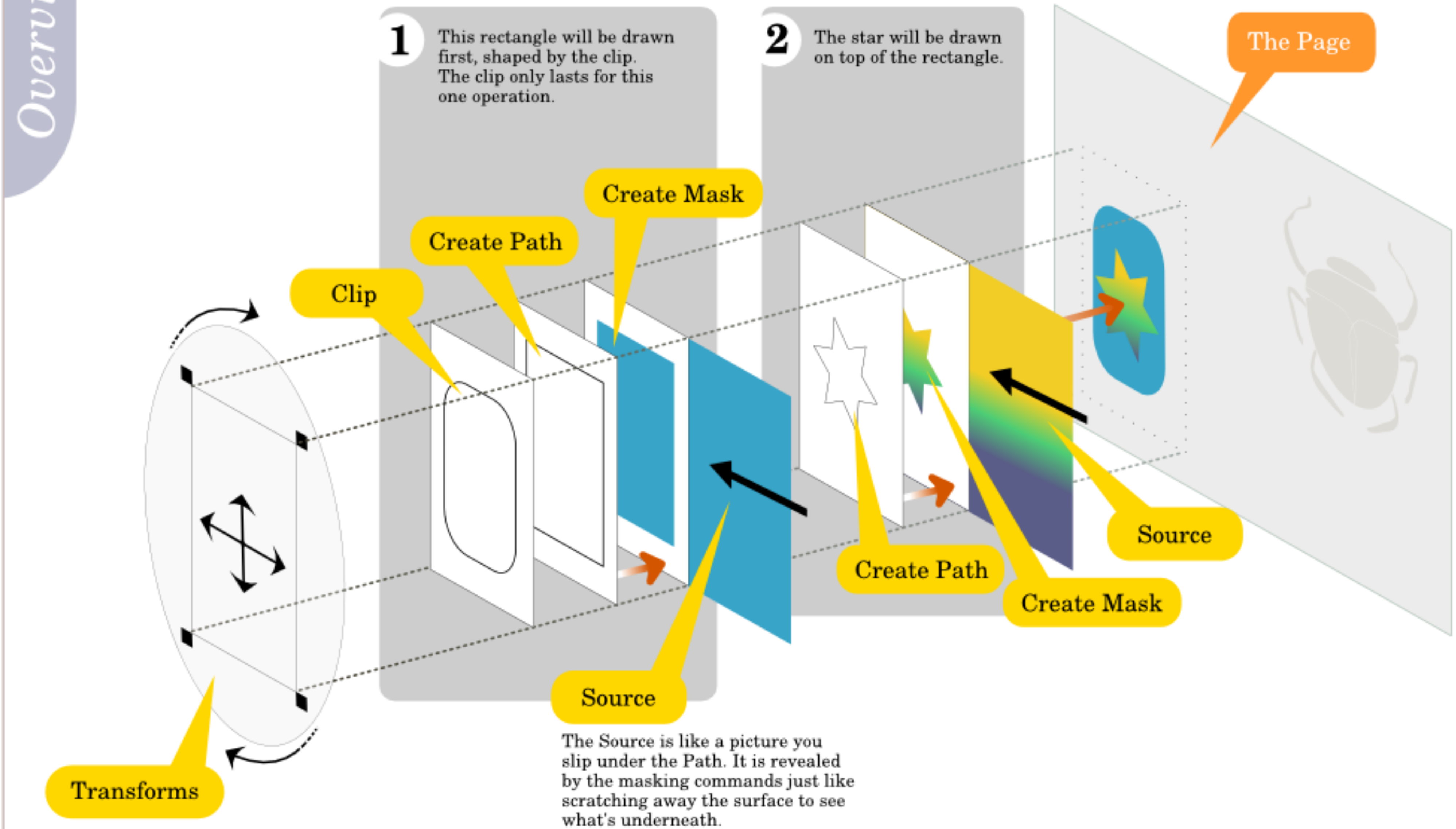
Transforms: Before you start drawing, you can set up a rotation and scale just like in a vector program like Inkscape.

python Cairoglyphics

vers 2.0

Overview

This diagram works from left to right. It shows the drawing of two shapes which each go to the page one after the other. All the commands to do this come from a *toolbox* called the Context. On the next page (Summary) I list many of the commands that you can use.



1 This rectangle will be drawn first, shaped by the clip. The clip only lasts for this one operation.

2 The star will be drawn on top of the rectangle.

Source

The Source is like a picture you slip under the Path. It is revealed by the masking commands just like scratching away the surface to see what's underneath.

See *Sources, Surfaces & Patterns* diagram for details.

Transforms

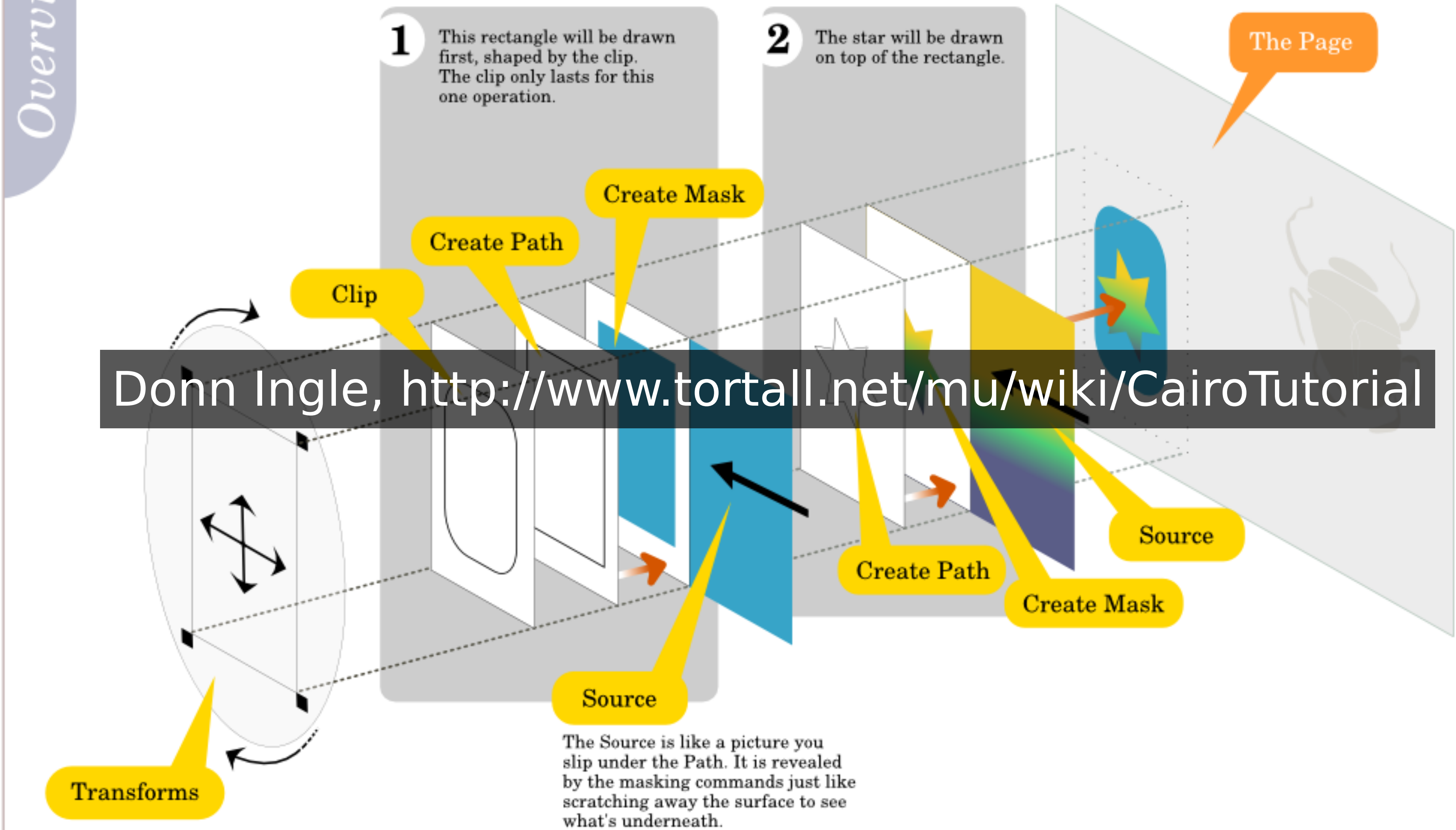
Before you start drawing, you can set up a rotation and scale just like in a vector program like Inkscape.

Cairoglyphics

vers 2.0

Overview

This diagram works from left to right. It shows the drawing of two shapes which each go to the page one after the other. All the commands to do this come from a *toolbox* called the Context. On the next page (Summary) I list many of the commands that you can use.



1 This rectangle will be drawn first, shaped by the clip. The clip only lasts for this one operation.

2 The star will be drawn on top of the rectangle.

The Source is like a picture you slip under the Path. It is revealed by the masking commands just like scratching away the surface to see what's underneath.

See *Sources, Surfaces & Patterns* diagram for details.

Before you start drawing, you can set up a rotation and scale just like in a vector program like Inkscape.

What does Cairo look like to the GPU?

ICANHASCHEEZBURGER.COM

cairo

~~Glyphs~~ **the good**

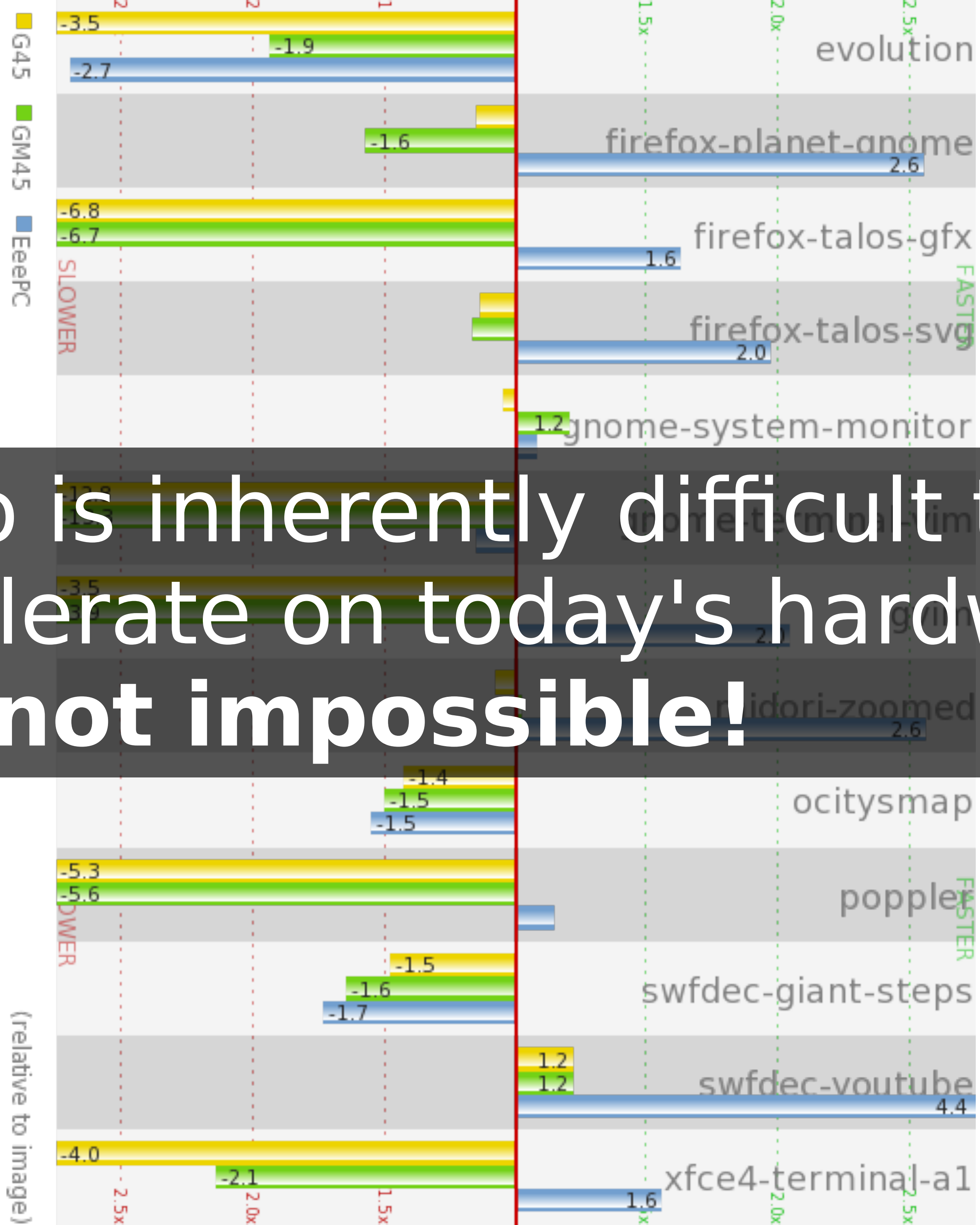
~~Fill~~ **the bad**

~~Stroke~~ **the ugly**

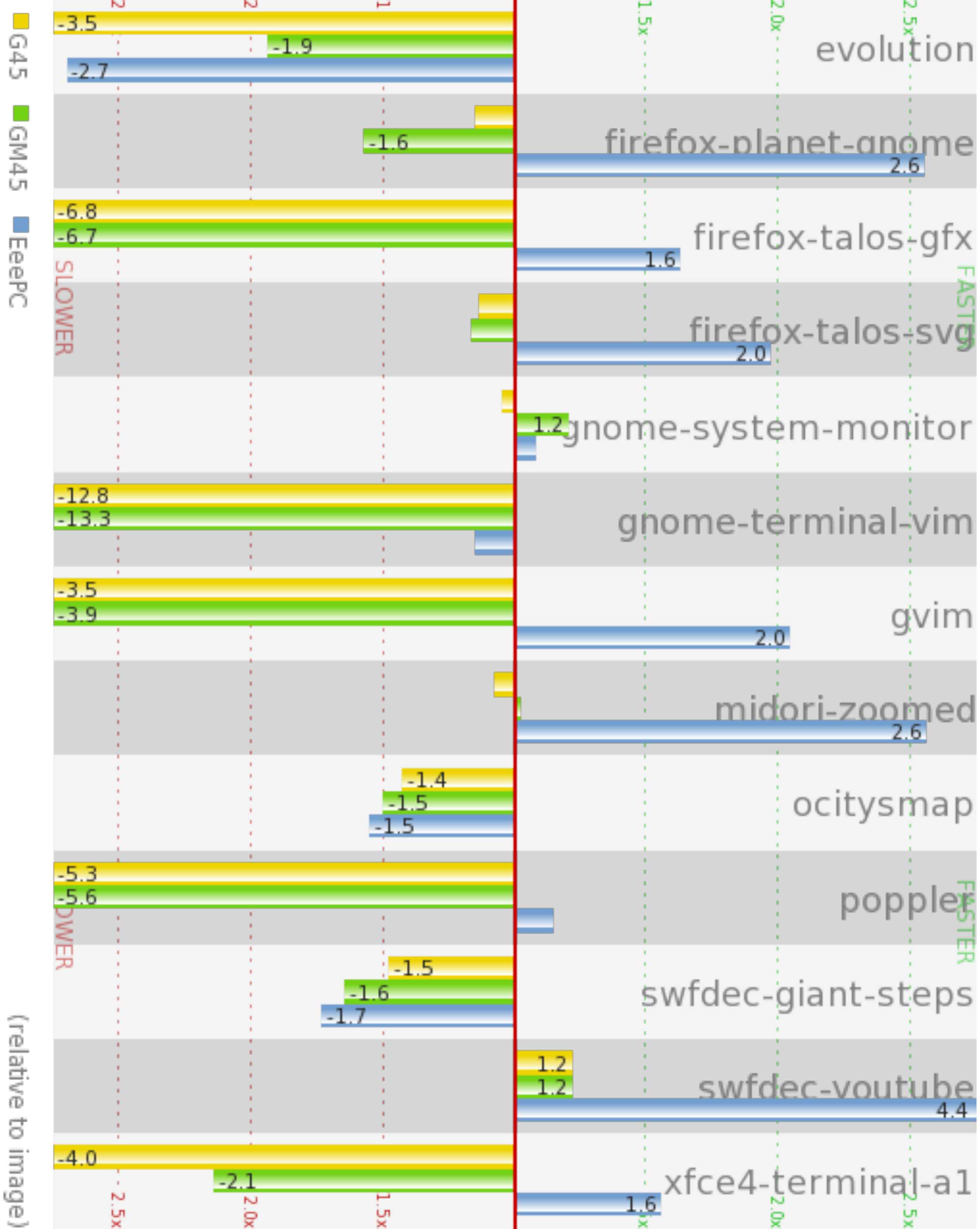
To go fast, think like a GPU.

- GPUs are very fast many-core stream processors with long pipelines.

- They do not like state changes; tiny operations with lots of cache flushes.



Cairo is inherently difficult to accelerate on today's hardware **but not impossible!**


G45
GM45
EeePC
SLOWER
FASTER
(relative to image)
2.5x
2.0x
1.5x
1.5x
2.0x
2.5x
evolution
-3.5
-1.9
-2.7
firefox-planet-gnome
-1.6
2.6
firefox-talos-gfx
-6.8
-6.7
1.6
firefox-talos-svg
2.0
gnome-system-monitor
1.2
gnome-terminal-vim
-12.8
-13.3
gvim
-3.5
-3.9
2.0
midori-zoomed
2.6
ocitysmap
-1.4
-1.5
-1.5
poppler
-5.3
-5.6
swfdec-giant-steps
-1.5
-1.6
-1.7
swfdec-youtube
1.2
1.2
4.4
xfce4-terminal-a1
-4.0
-2.1
1.6

cairo

My ideal hardware would do 31x33 FSAA.

Include sub-pixel filtering.

And process an entire composite DAG in a single pass.

The future

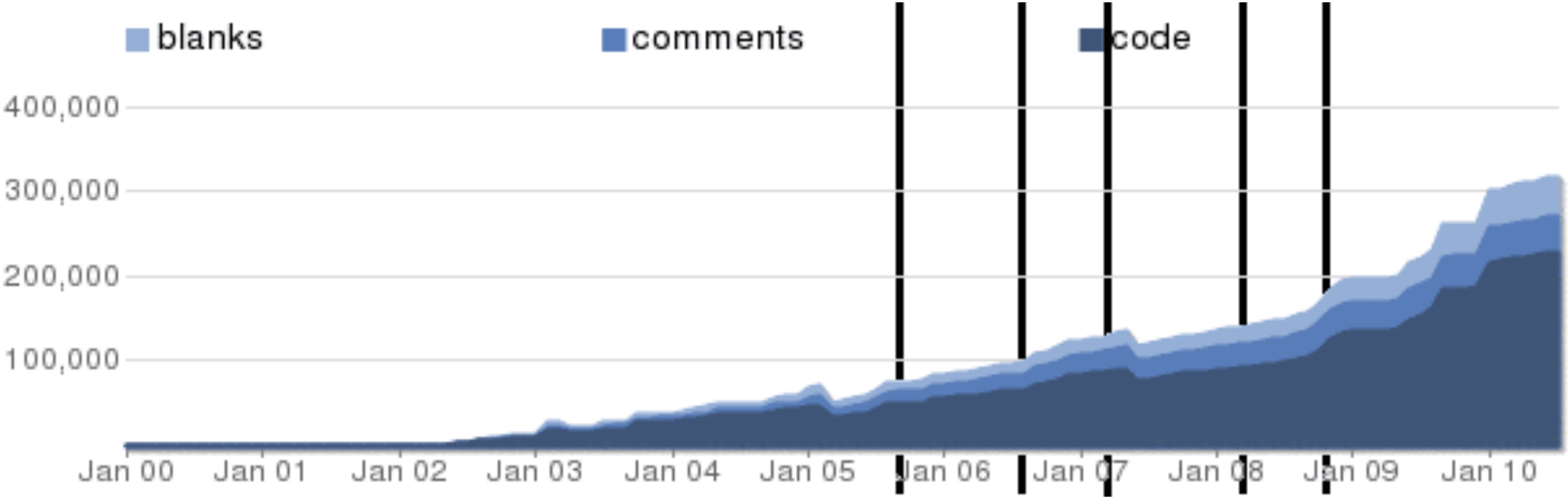
- Focus on what Cairo does best:
Page Description and Presentation

- Leave the animation to libraries that
were explicitly designed for the task

- Share the core routines for the common tasks?

Overcoming the release drought

- Can anyone remember the last Cairo release?
- Development however did not stall.
- With every patch the amount of code to stabilise increased, and the desire reduced.



http://www.ohloh.net/p/cairo

New Release Plan!

- Make 1.10.

- Time based releases: every 6 months.

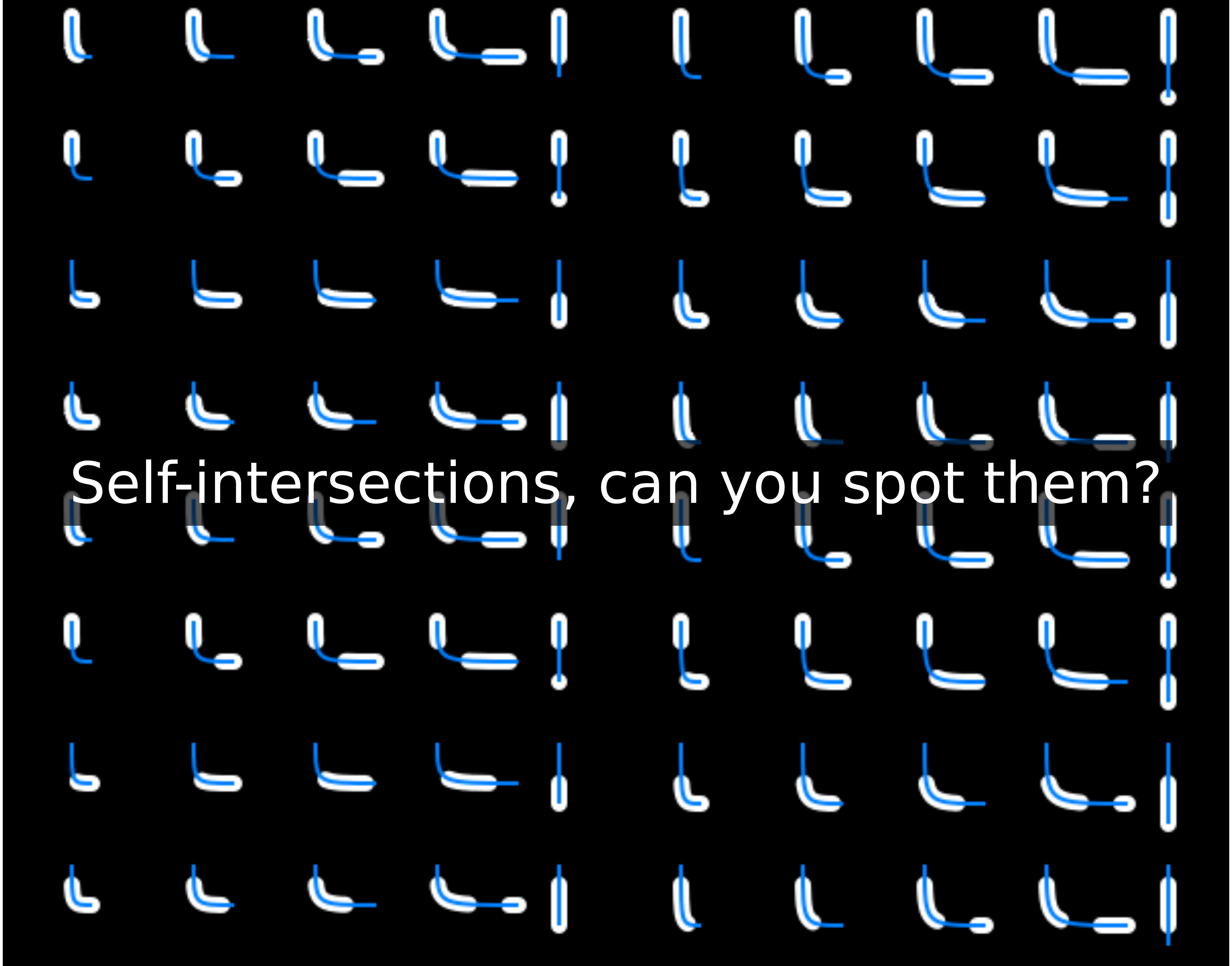
- Get updates to distros quicker, and maintain continual development, yet stable base.

1.10, what does it do for me?

- Self-intersection removal

- Much faster software rasteriser
  - scan line based
  - direct rendering optimisations
  - [pixman has much improved as well]

- Server-side gradients



Self-intersections, can you spot them?

What's next?

buildbot

cairo_backend_t

color spaces

compositor/blitter interface

direct2d backend

mime surface

cairo_shape_t

stroke-to-path

type7 gradients

yuv and planar images

Images courtesy of

- http://xkcd.com/721/ - Attribution-NonCommercial 2.5 Generic
- http://icanhascheezburger.com/
- http://cuteoverload.com/

Render fast, die young.