WebKit GTK+

an alternative browser engine
for the free desktop

Alp Toker

2000, Zeitgeist

- Mozilla is starting to shape up
- Linux/Unix desktop users are migrating away from Netscape
- Cautious optimism about an open Web despite Microsoft Internet Explorer's dominance
- Meanwhile, KDE 2.0 launches with KHTML at the core of its browser shell, Konqueror. It does not go unnoticed.

And so at last the beast fell and the unbelievers rejoiced.
But all was not lost, for from the ash rose a great bird.
The bird gazed down upon the unbelievers and cast fire
and thunder upon them. For the beast had been
reborn with its strength renewed, and the
followers of Mammon cowered in horror.

from ***The Book of Mozilla****, 7:15*

2005

- Safari developer Dave Hyatt (incidentally the co-creator of Firefox) announced on his weblog that Apple was open sourcing WebKit
- Previously, only WebCore and JavaScriptCore were open source
- CVS (now SVN) repository and Bugzilla were opened up
- A big step forward from the minimal code drops previously coming from Apple to satisfy the LGPL

Today, 2007

- Firefox is in widespread use (estimated market share around 15%)
- Trend is to move away from monolithic “Internet suites” towards standalone applications and re-usable components
- There is an emerging demand for mobile Web browsers on phones, Internet tablets
- GNU/Linux has taken off and seems destined to become the commodity mobile and embedded OS

However...

- The emerging de-facto mobile browser, at the core of many of these new Linux-based devices, is also in many cases the only proprietary component of the platform
- Vendors are choosing the Opera browser engine
- The community has been concentrating on desktop market share
- ... in the mean time, missing out on the potentially much larger embedded space

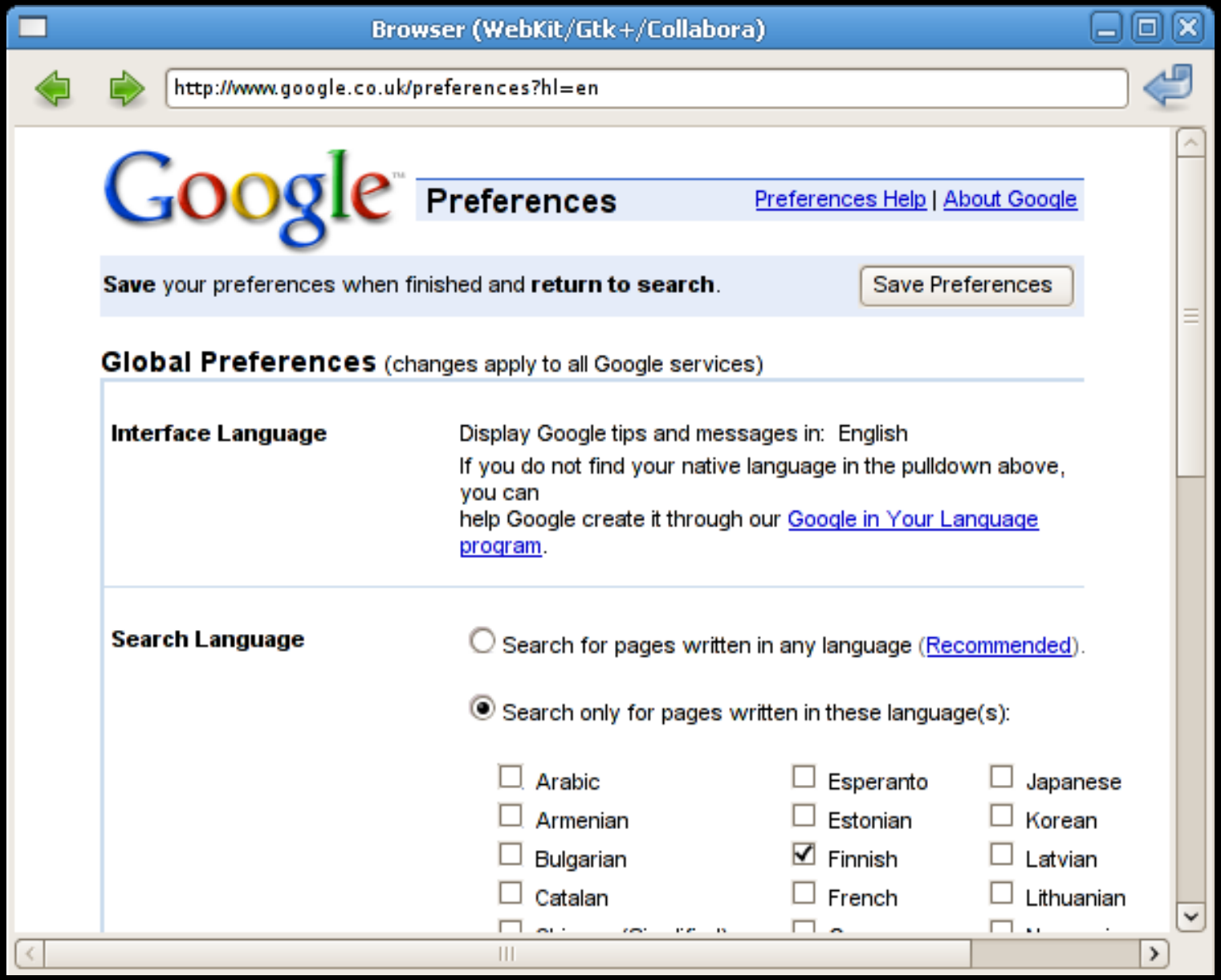
What is WebKit?

- An Open Source web browser engine
- The backend for Apple's Safari browser
- Officially available on OS X and Windows
- Used in Apple's iPhone
- Also ported to, ships with all Symbian S60 devices
- An Open Source project inviting contributions to the core engine as well as welcoming ports to new platforms and devices

What is WebKit/Gtk+?

- Gtk+ is the UI toolkit used by eg. the GNOME desktop
- Gtk+ lacks a modern native browser engine, though Mozilla provides a workable gtkmozembed component
- WebKit/Gtk+ is a port of the WebKit browser engine to both use the Gtk+ toolkit for its functionality and to then be available for embedding into host Gtk+ applications

Native widget theming



Why hack on browsers?

Personally...

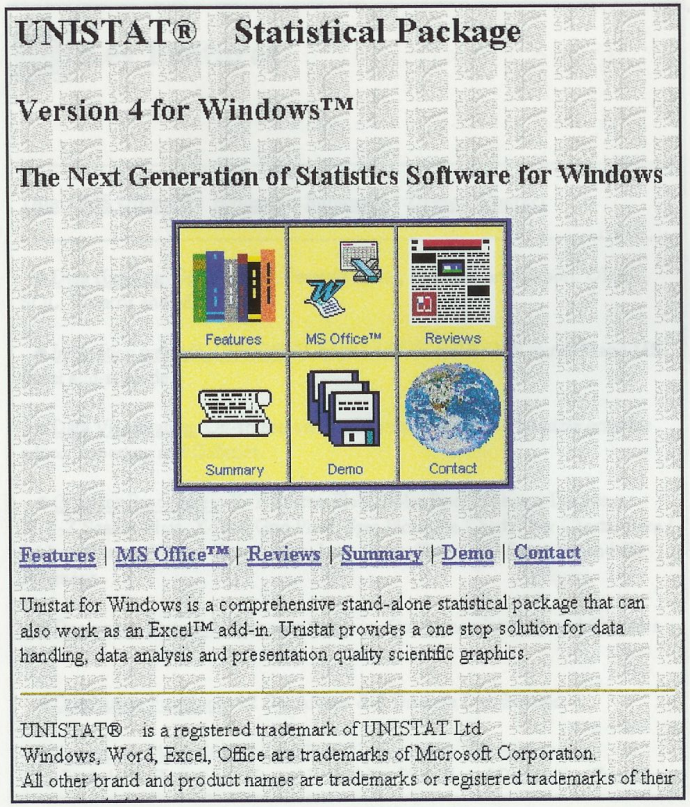
- Was a Netscape user in 1995
- Became frustrated with the lack of in-place editing support
- Started hacking on my own browser engine and UI with built-in editing features
- Took about 9 months (weekends and evenings) to complete

WebWorks...

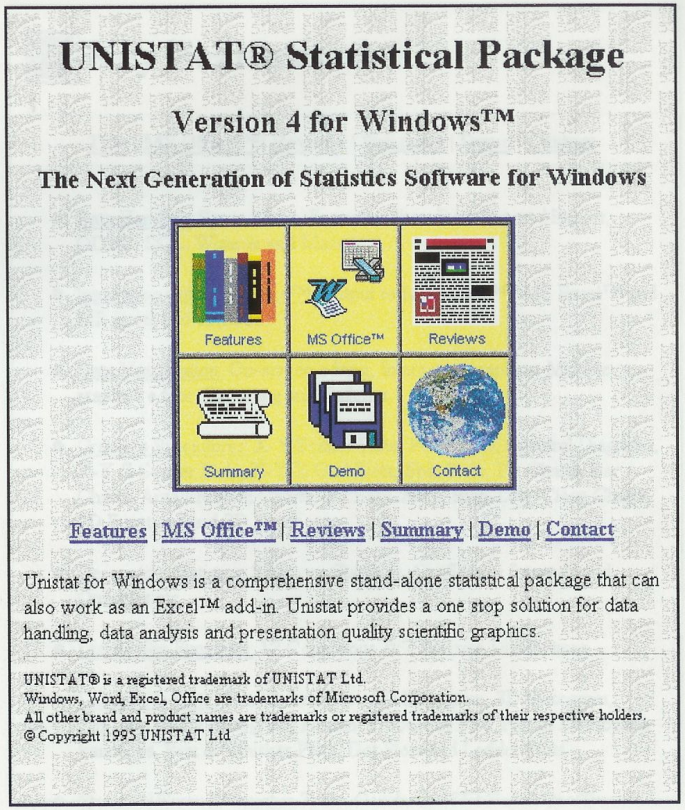
WebWorks (cc. 1995)

Figure C.II.1.

WebWorks Version 1.0



Netscape Version 1.2N



WebKit features at a glance

- Comprehensive Web standards support (WebCore, based on KHTML)
- JavaScript (JavaScriptCore, based on KJS)
- CSS, up to and incl. some CSS3
- “AJAX stuff”
- SVG
- Canvas
- Fast, lightweight, compatible, embeddable
- Under constant development

“WebKit is an engine, not a browser”

- WebKit has been developed as a re-usable browser engine component
- Intends to provide seamless integration with existing applications
- WebKit/Gtk+ shares these goals

WebKit: The code

- Written in a sensible subset of C++
- Limited use of templates
- Approachable to C hackers
- Follows a coherent coding style
- Dead code is removed liberally – there is always version control, after all
- All major internal APIs are “informally” abstracted for easy porting, but remain malleable, similar to those of the Linux kernel

Development workflow

- Formalised Open Source development process
- Strict patch review for all contributors and committers
- Commit access given only after significant contributions, and following necessary paperwork
- Currently 2 WebKit/Gtk+ committers: Alp Toker, Holger Freyther

A little story (“Stop energy”?)

- Late 2006: Adobe announces intent to use WebKit in their AIR platform
- Supposedly would be completing the Cairo rendering backend
- Several months later, nothing but minimal patches
- Luckily the WebKit/Gtk+ developers decided *not to sit around waiting* for this code drop, but instead forged ahead and completed the graphics backend
- Well aware that we might have been duplicating effort
- Promises of code can be damaging – real patches or solid discussion much more helpful

Forks considered harmful

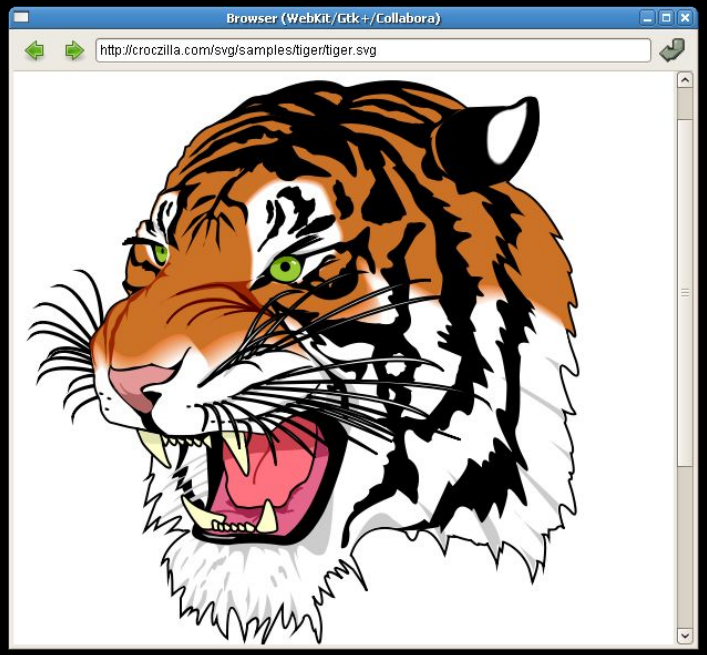
- Some kinds of projects seem to *thrive on forking* as an evolutionary force
- This does not seem to be the case in browser development beneath a certain threshold of activity
- Gtk+ porting efforts attempting to fork from mainline WebKit have *tended to fail*

Why don't forks succeed?

- Turns out that having an *expert developer team* (incl. Apple engineers) working directly on your shared code base is most valuable
- Maintainers of forked code bases are *alienated from the core developers*, who could otherwise offer advice and fix up their abstraction layers as needed
- Morale runs low with the realisation that your developers' *patches may never be merged*
- Lack of decent version control (git-svn may help here)

Features: SVG Support

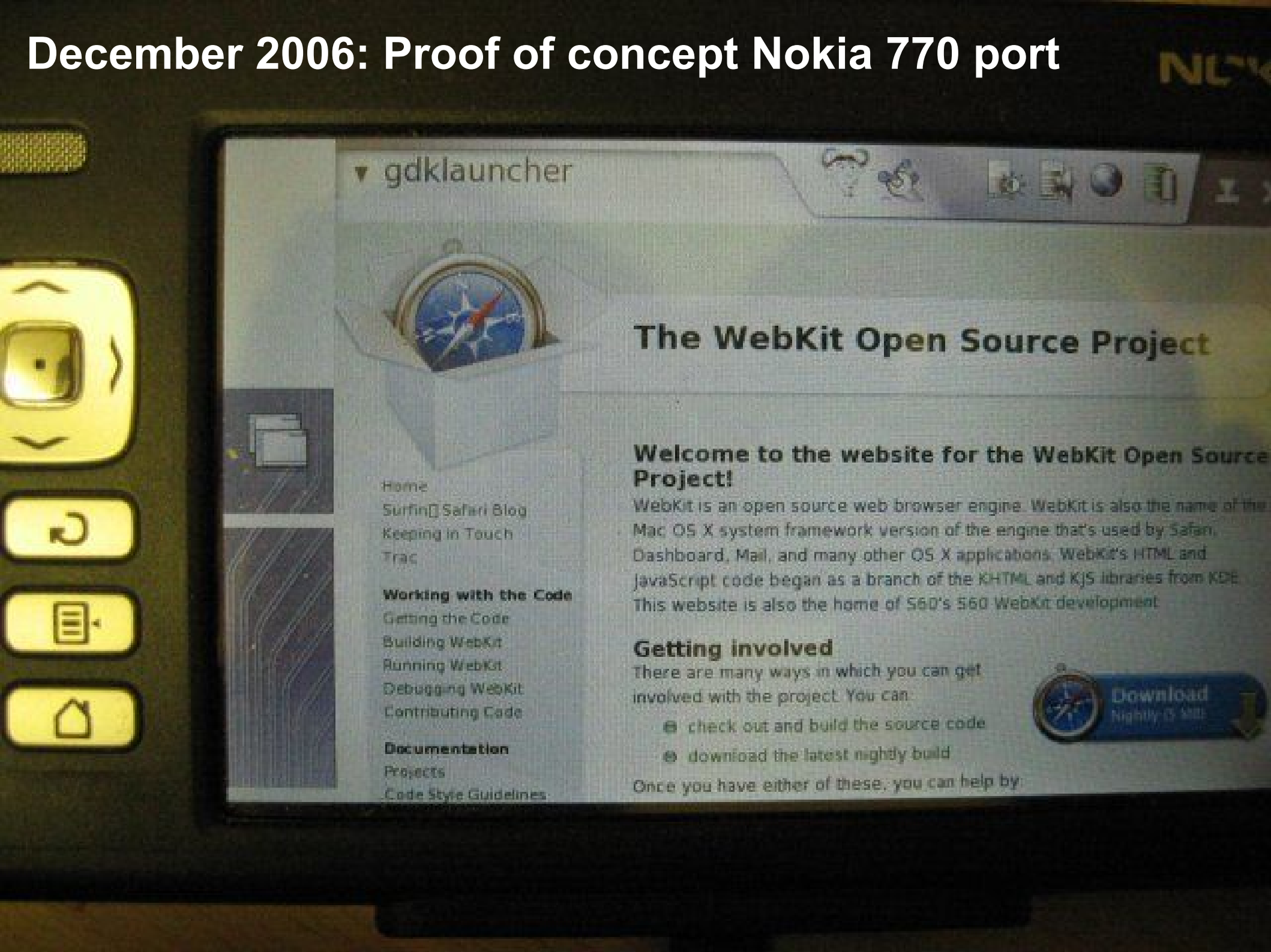
- SVG engine based on KSVG2
- SVG is supported as a core document type, not just an image format
- Vector graphics can be animated dynamically by manipulation of the DOM with JavaScript
- WebKit/Gtk+ implements its SVG backend using Cairo



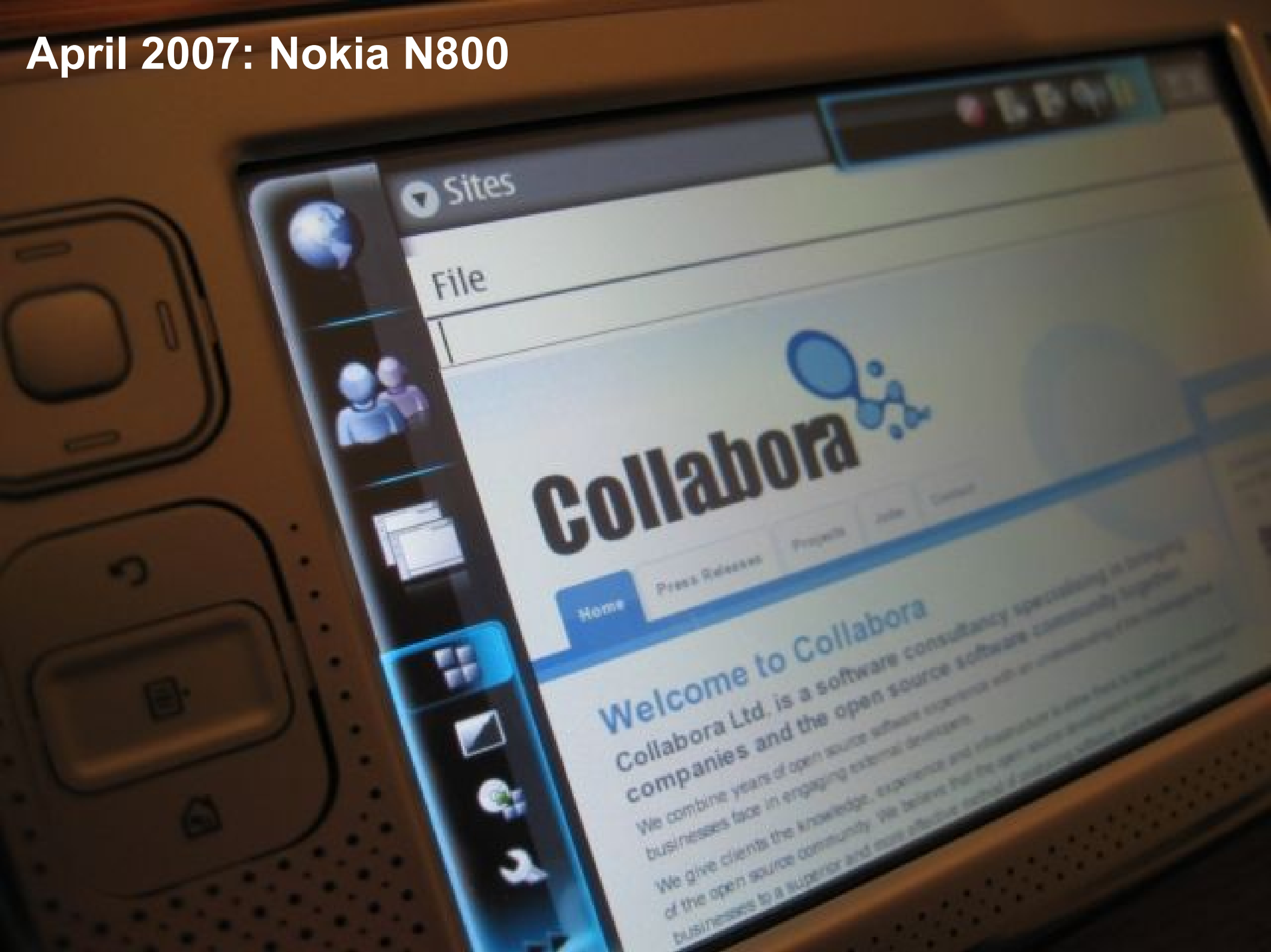
Features: The <canvas> element

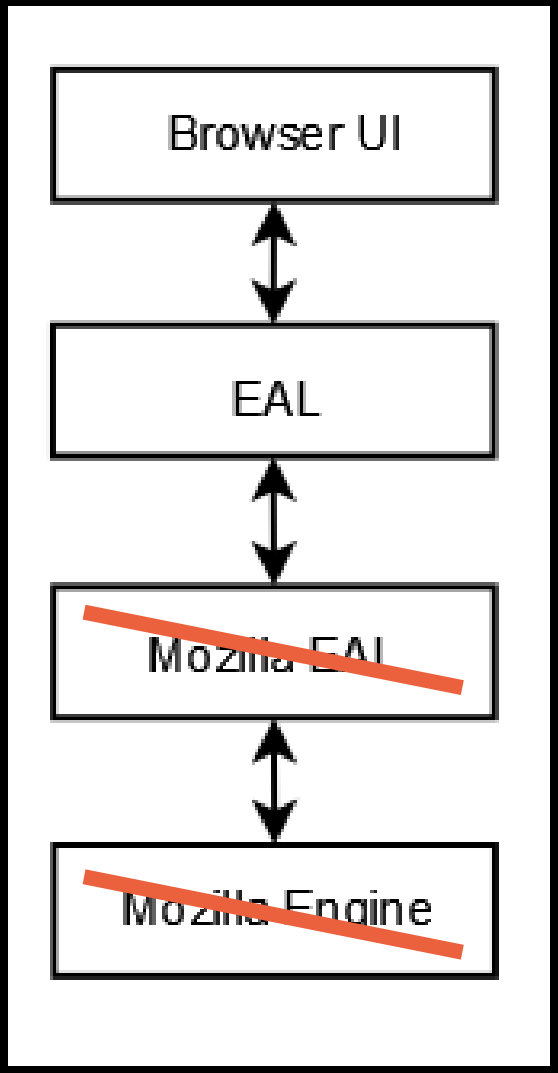
- Provides dynamic scriptable rendering of bitmap images
- Powers Apple's Dashboard widgets
- Originally a WebKit feature
- Now supported by Mozilla, Opera
- Now part of HTML5
- Coming soon to WebKit/Gtk+

December 2006: Proof of concept Nokia 770 port



April 2007: Nokia N800





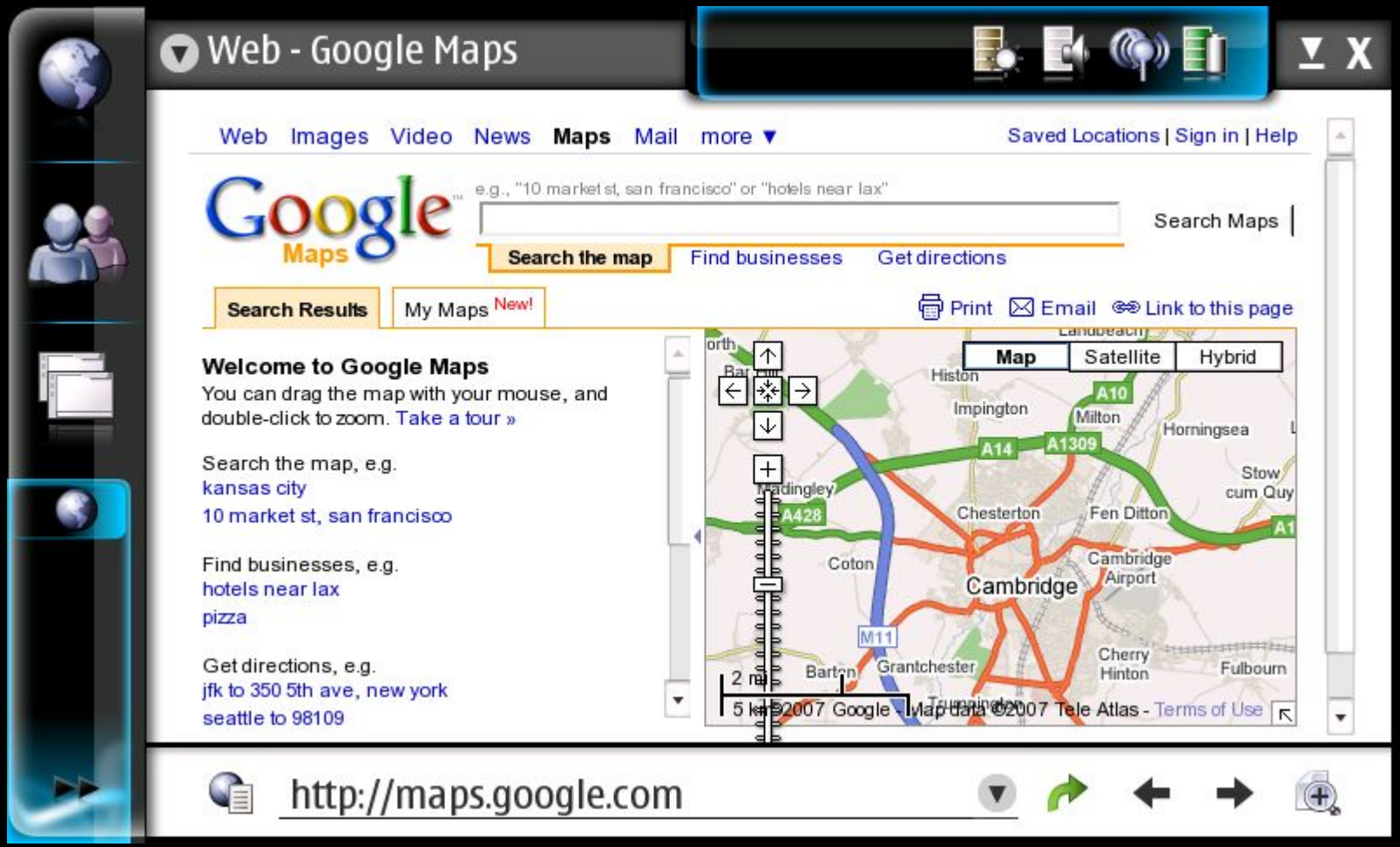
Graphics: Nokia

WebKit/EAL

- Nokia's Engine Abstraction Layer
- First introduced in the N800
- Allows users to switch between Opera and Nokia's Mozilla port
- We now have a WebKit/Gtk +-based EAL engine
- Integrates with device's buttons, input systems etc.

EAL, abstracting browser engines

Browsing on the N800



Full page scaling

Cairo graphics backend

- Cairo, device-independent 2D vector graphics library, used eg. by Gtk+
- Shared amongst ports, but the main consumer is the Gtk+ port, with Qt, Wx, Mac using their own graphics libraries
- High quality antialiased vector graphics and image composition
- Unified display, printing, PDF output etc.

State of the public API

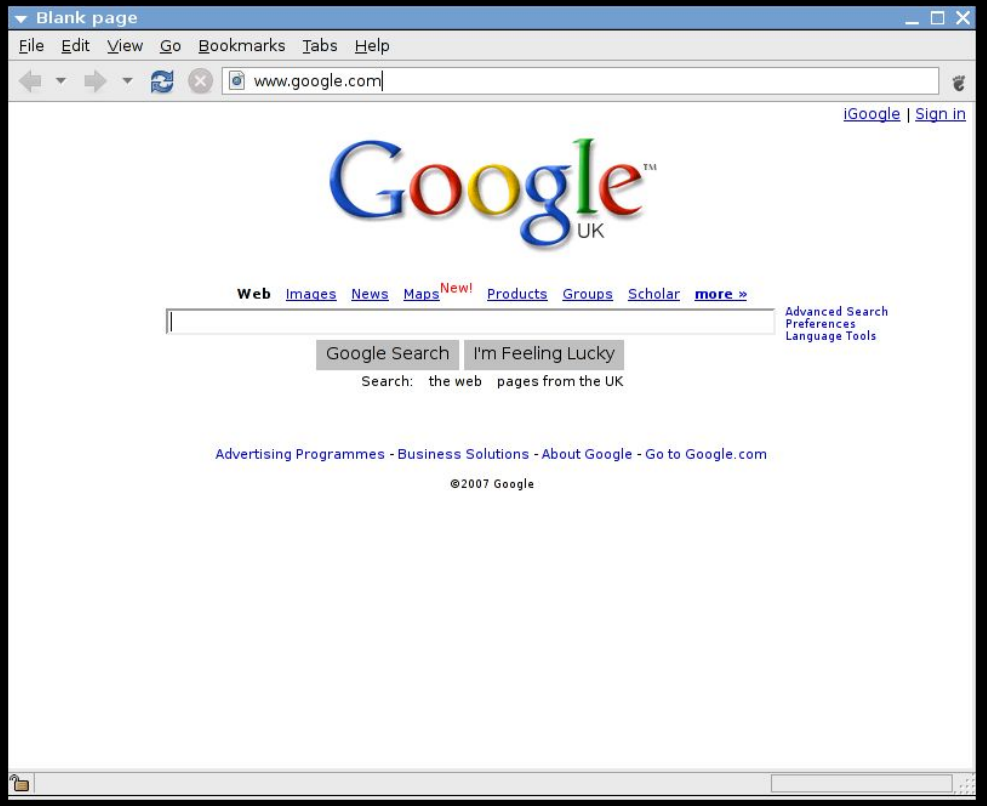
- Apple's public, stable WebKit API is written in Objective-C
- WebKit/Gtk+'s public API is C/GObject, with properties, signals in the style of the Gtk+ toolkit
- Goal is to match the functionality of the Objective-C API
- Still API unstable – we need to ask the developer community what they expect a browser API to look like

Seamless Gtk+ integration

- The browser widget follows Gtk+ conventions closely, eg. makes size requisitions and deals with allocations, but is usually packed in a GtkScrolledWindow
- There is no XUL-style platform and toolkit – the engine uses the standard Gtk+ look and feel
- Uses Gtk+ technologies wherever possible
- Avoids a GNOME dependency in the core without compromising potential for GNOME integration

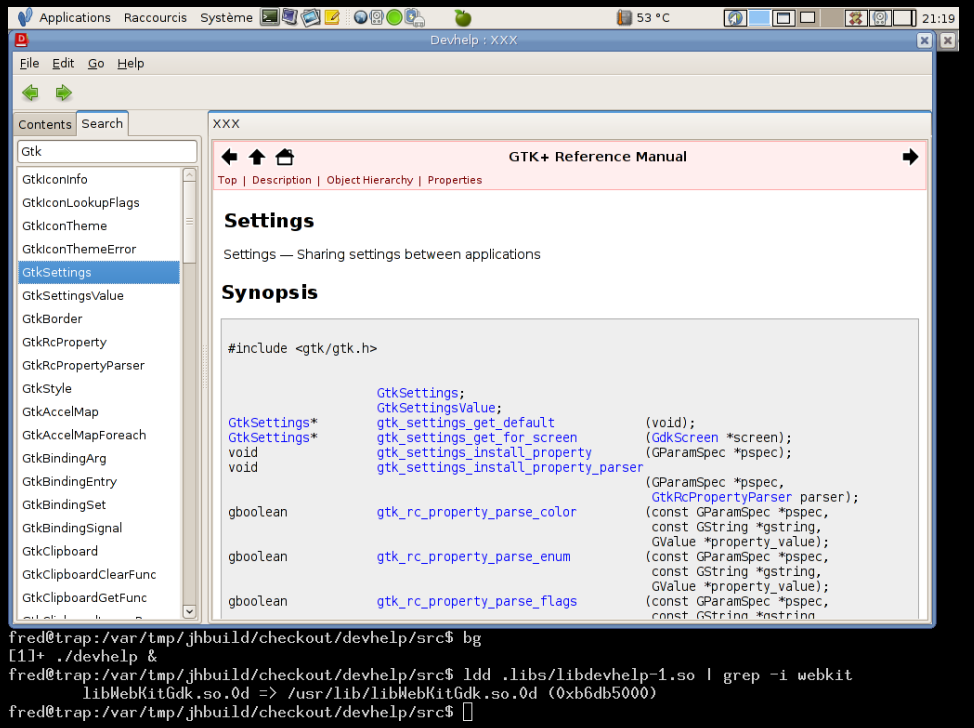
Applications: Epiphany
-with-engine=webkit

- Epiphany, a light-weight Web browser for the GNOME desktop
- Originally a Gtk+ UI around the Gecko rendering engine
- WebKit support added by Xan Lopez during GUADEC 2007

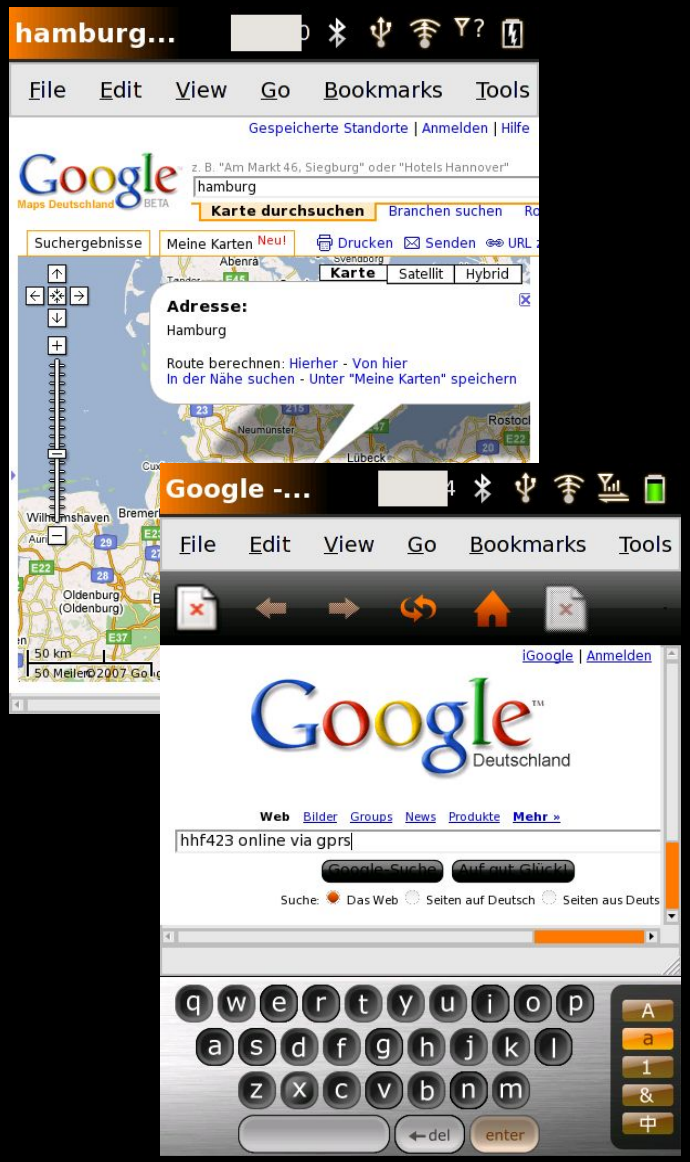


Applications: Devhelp

- A browser for Gtk+/GNOME API documentation
- Currently uses gtkmozembed – developers did not sound too happy with it
- WebKit/Gtk+ integration work going on
- Faster startup
- Lower memory footprint



WebKit/Gtk+ and OpenMoko



- Holger Freyther, Google Summer of Code student for the OpenMoko project
- Has been concentrating on improving the port as a whole, benefitting everyone

Proof of concept Sugar activity for the OLPC

WebKit/Gtk+ and OLPC?

- Proof of concept port of WebKit/Gtk+ as a Sugar activity done by Dan Winship, of Red Hat in *one day.* John Palmieri compared it with the existing Mozilla-fx3-based activity:
- "uses on average ***10 megs less in resident memory***"
- "starts up ***five seconds faster***"
- "feels a bit ***more responsive*** when scrolling"
- "just more aligned to our needs as a ***small*** and ***fast*** browser"
- Caveat: "the gtk port is just not finished yet [...] but it is close"

HTML5: The <video> tag: Coming soon?

- HTML5 is a WHATWG spec
- Backend support recently landed in the WebKit feature branch
- Apple's OS X and Windows implementations use QuickTime for audio/video
- Plans to implement audio/video in WebKit/Gtk+ using GStreamer
- Promises to displace Flash as the video delivery mechanism of choice

Getting involved

http://www.webkit.org/

IRC: #webkit, #webkit-gtk / FreeNode

WebKit GTK+