



# WebKit GTK+

Developing hybrid Web/GTK+  
rich internet applications

February 2008  
FOSDEM  
Brussels

Alp Toker



**JavaScriptCore**  
portable C API

**WebCore**  
content engine



**WebKit**  
GObject API



**GTK+ applications**  
C, C#, C++, Vala, Python



# WebKit language bindings

## 🐾 Vala bindings

- Maintained by Jürg Billeter
- Covers core API

## 🐾 Python bindings

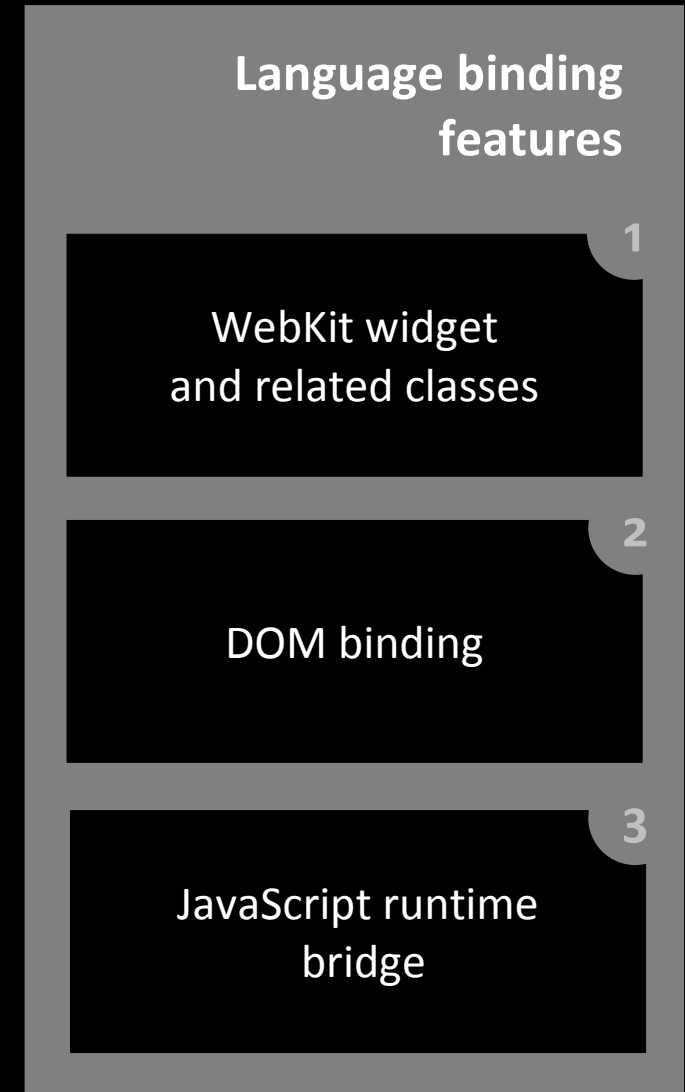
- Created by the OLPC team
- Maintained by Jan Alonzo

## 🐾 C#/CLR bindings (WIP)

- JS bridge available

## 🐾 JavaScript bindings

- DOM only (part of JavaScriptCore)



# Introducing WebView

☞ Write a browser in a dozen lines

```
import gtk
import webkit

view = webkit.WebView()

sw = gtk.ScrolledWindow()
sw.add(view)

win = gtk.Window(gtk.WINDOW_TOPLEVEL)
win.add(sw)
win.show_all()

view.open("http://planet.gnome.org/")
gtk.main()
```

# WebView modes: Scrollable

## ☞ Packed in a **GtkScrolledWindow**

- Provides a full-featured browser engine for document display and editing



# WebView modes: Packed

- ☞ Packed directly into the UI
  - Acts as an **integral part of the surrounding GTK+ UI**
  - Place Web content in amongst GTK+ widgets
- ☞ Web content / GTK+ size request interop  
(landing soon)
- ☞ Develop and design in parallel
  - Let programmers work on the core
  - Let designers produce UI elements using Web skills

# When to use a WebView

- When do you use straight GTK+?
- When does Web content enrich the experience?
- Great power; great responsibility
- **WebView** isn't the right tool for every job
  - Continue to use **GtkTextView** for light viewing and editing
  - Use **GtkTreeView** and **GtkIconView** unless you really need a custom look

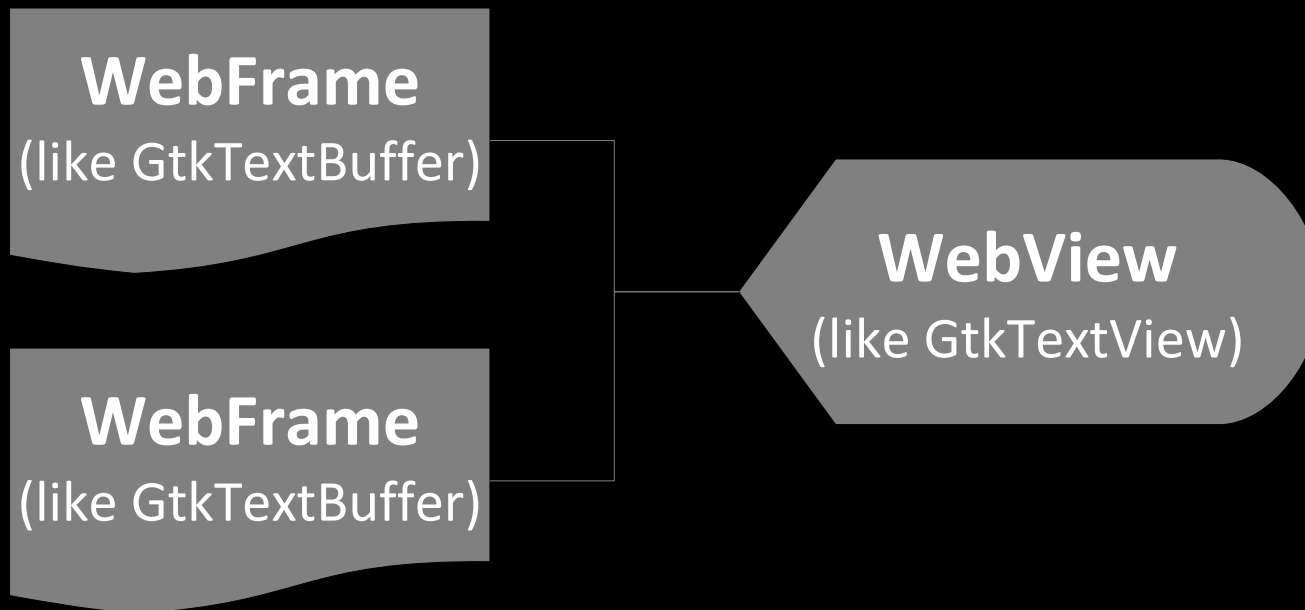
“The idea was that anybody who used the web would have a space where they could write and so **the first browser was an editor**, it was a **writer as well as a reader.**”

Tim Berners-Lee



# A writer as well as a reader

- Enable WebKit's powerful content editor with one line of code
- **WebView** is designed from the ground up to work as an **enhanced GtkTextView**



# Edit with style

- `webkit_web_view_set_editable (WEB_VIEW (view), TRUE);`
- Push and retrieve HTML/SVG content with simple accessors or use the upcoming **GIO** streaming interface
- Perform formatting operations
  - With the basic editing command API
  - Or by manipulating the upcoming **GObject DOM** directly
- Work on your application's killer features and leave formatting to `WebView`

# Using WebFonts

- ☞ SVG/TTF custom fonts are a W3C recommendation
- ☞ Apply a *distinctive* look **without compromising usability**
- ☞ Continue to internationalise with gettext and .po files
- ☞ Text selection and editing works as usual
- ☞ **No installation required**

```
@font-face {  
    font-family: 'Bitstream Vera Sans';  
    src: url('http://www.freedesktop.org/~alp/tmp/Vera.ttf') format(truetype);  
}
```

```
h1 {  
    font-family: 'Bitstream Vera Sans', sans-serif;  
}
```

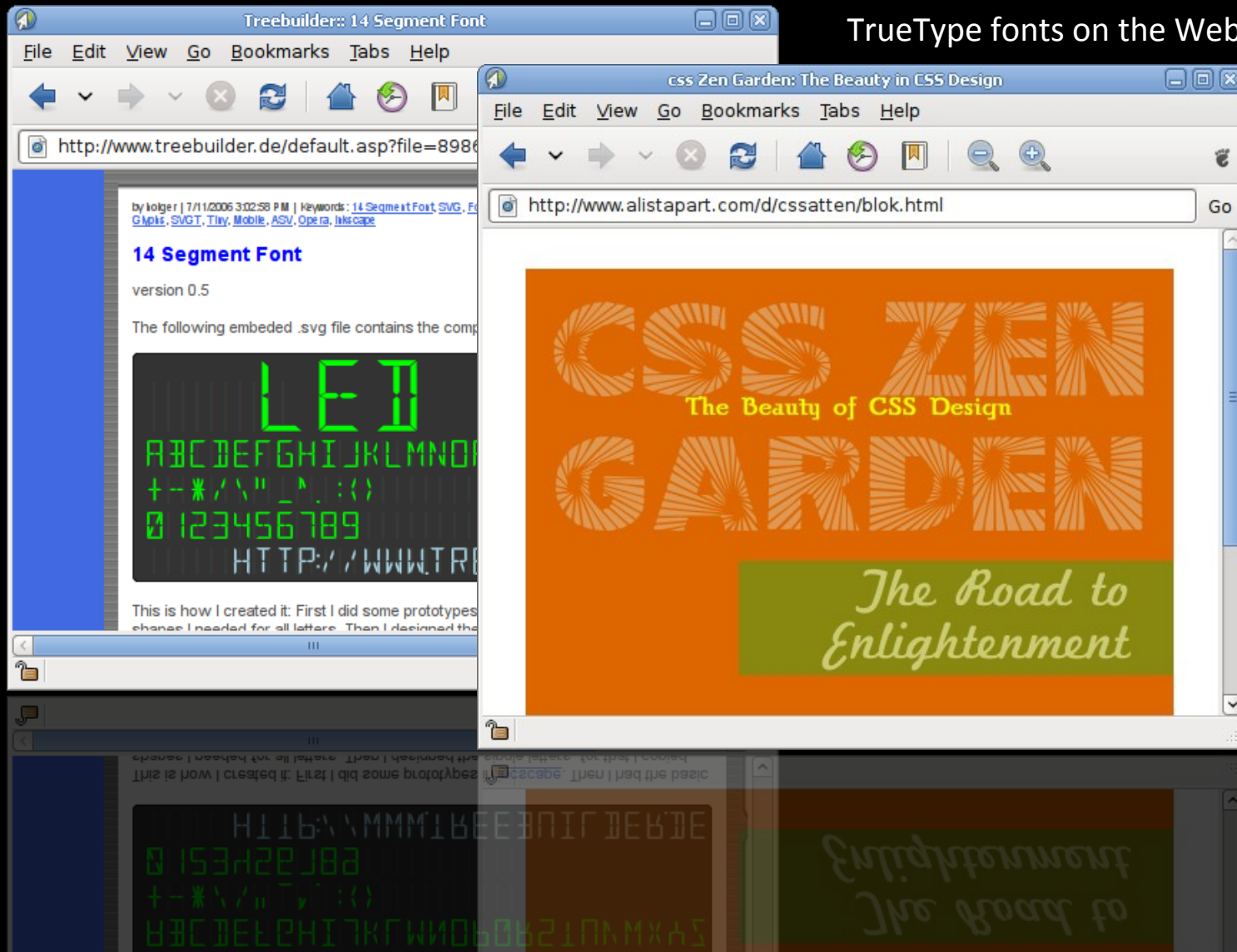
# Custom fonts in action

## SVG fonts

Easy to design with tools like Inkscape

## WebFonts

TrueType fonts on the Web



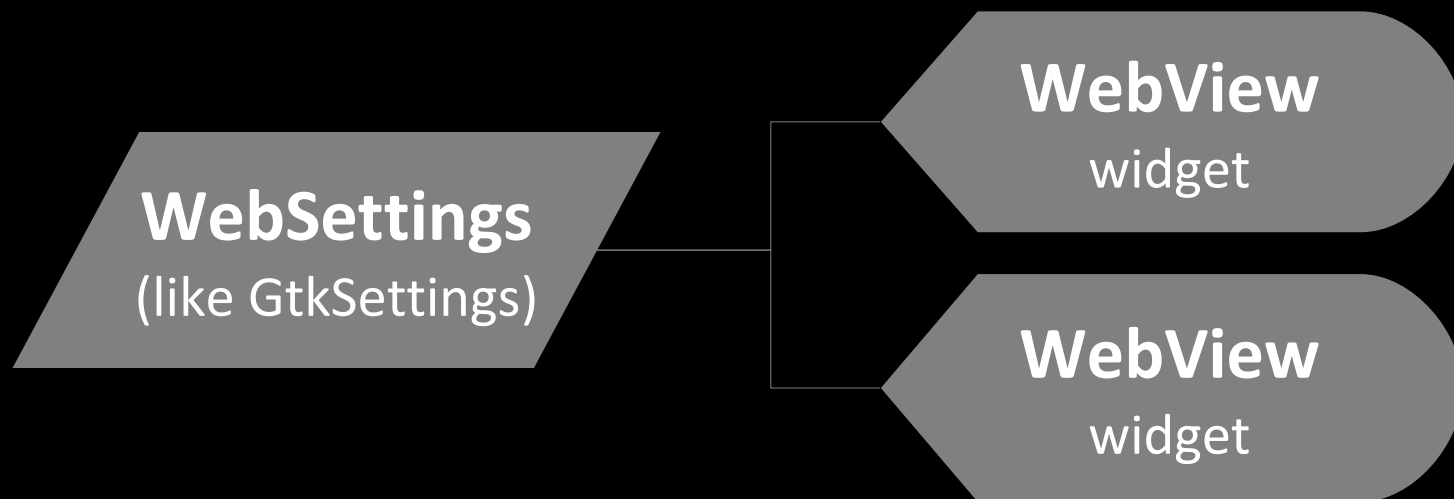
# GObject DOM

- ☞ Existing DOM bindings
  - **ObjC DOM**, used extensively in Safari/Mac
  - **COM DOM** (new), used on Windows
- ☞ Upcoming GObject DOM features
  - Complete access to the DOM (all levels)
  - Stable API

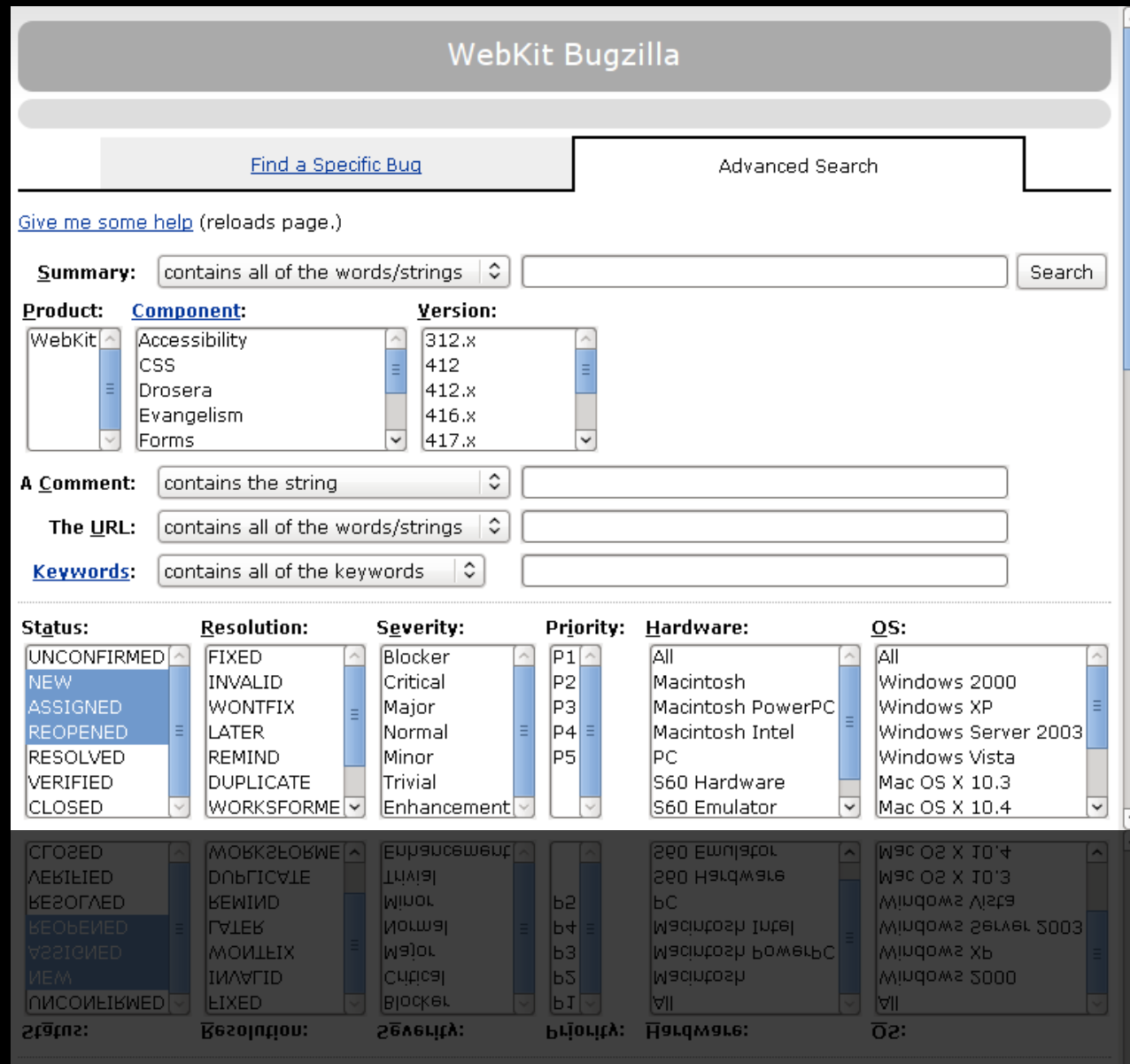
**LIVE DEMO**

# Apply settings with ease

- Use **WebSettings** to group settings for multiple WebViews
- Keep **granular settings** per WebView or enforce global settings when necessary  
(makes porting Gecko-based applications a breeze)



# Perfectly native widget styling



# Go asynchronous

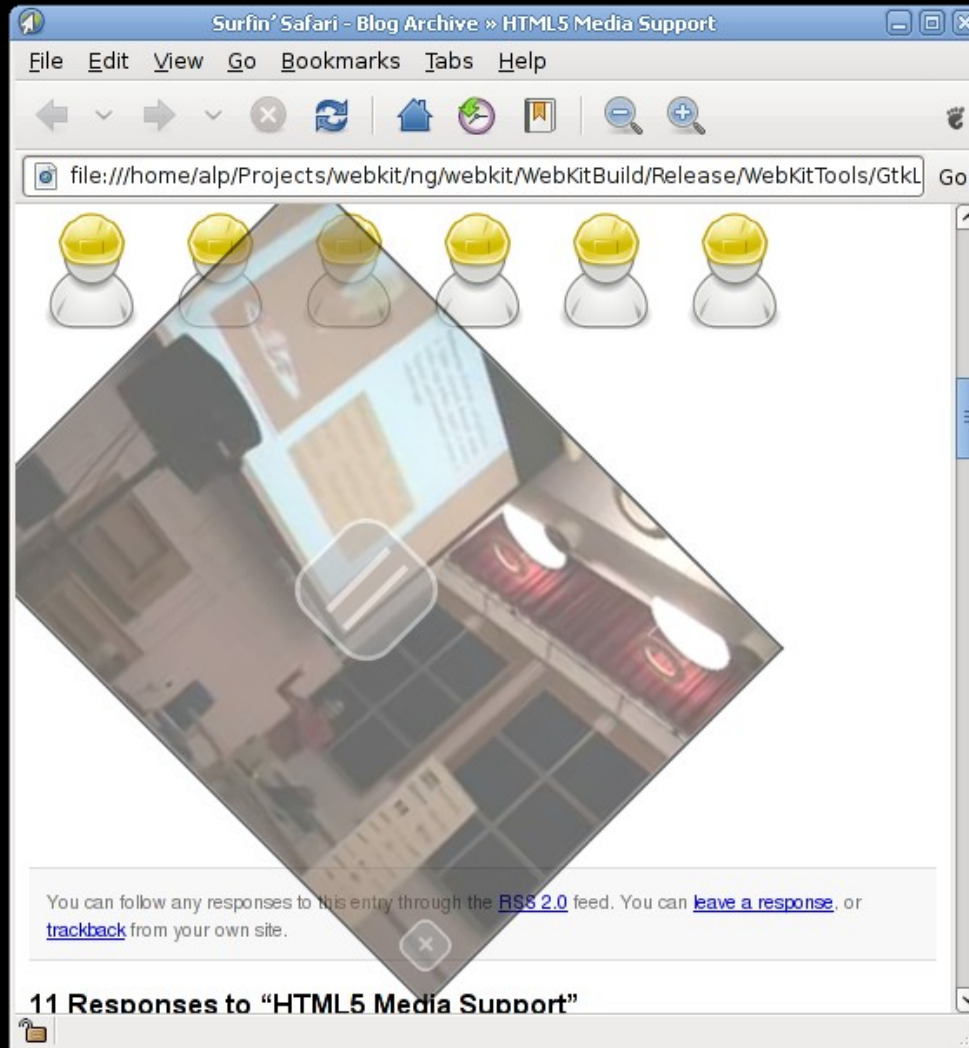
- ☞ Use **WebDecision** objects to delegate actions requiring user input or network queries
  - Authentication challenges
  - Navigation requests
  - Script alert and print dialogs
- ☞ A dream come true for browser developers
- ☞ Allows programmers to **eliminate modality**
- ☞ Lets users get on with what they're doing



# HTML5 video with GStreamer

## Open Source

Web video without  
proprietary plugins



Feature developed by  
Collabora

## Versatile

Create stylish DVD/DVB  
players, video  
conferencing tools

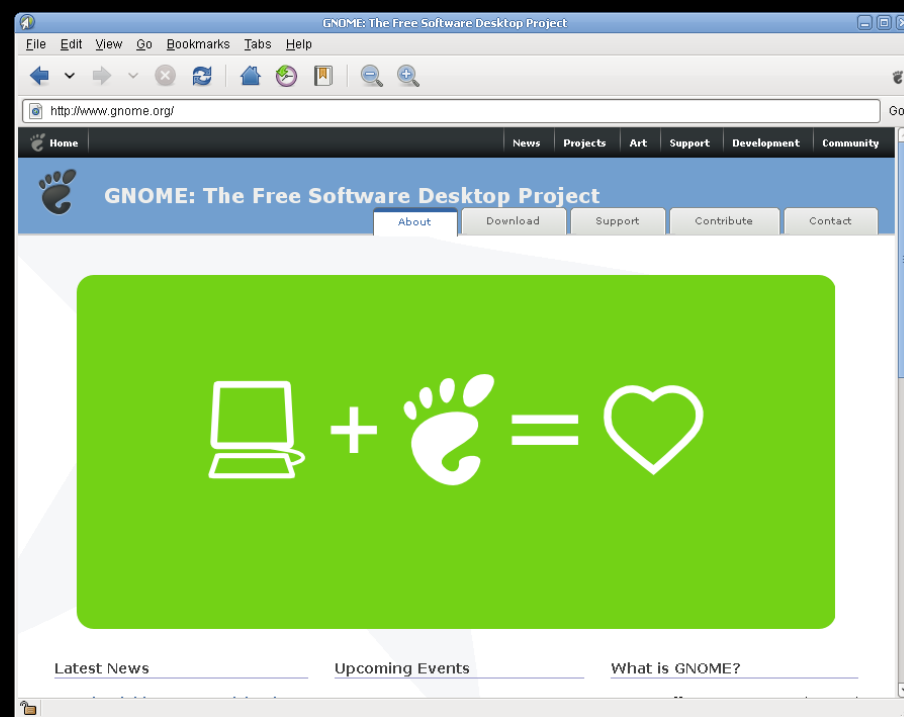
# WebKit for browser engineers

- Engine core written in a sensible dialect of C++
- Approachable to C hackers
- Follows a coherent coding style
- Project-wide refactoring and reorganisation is encouraged
- Internal APIs are “informally” abstracted and change frequently while the public API is strictly stable (similar policy to the Linux kernel)

# Browse with Epiphany



- 🐾 Epiphany, a light-weight Web browser for the GNOME desktop
- 🐾 Originally a GTK+ UI around the Gecko rendering engine
- 🐾 Experimental WebKit support added by Xan Lopez at GUADEC 2007
- 🐾 WebKit backend is well maintained
- 🐾 Seeing rising popularity



# WebKit and Yelp

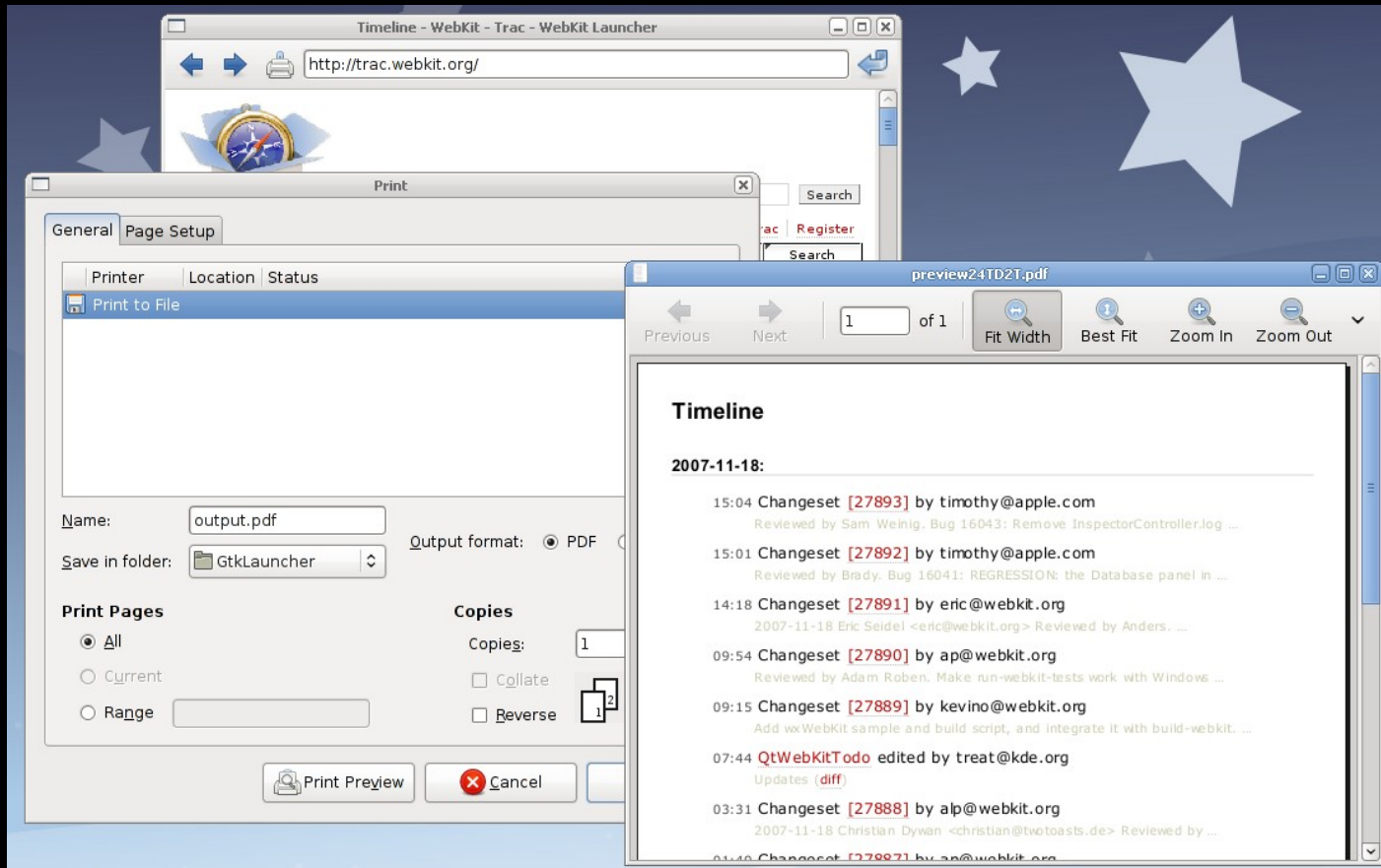


- 🐾 Yelp is the GNOME documentation browser
- 🐾 Initial WebKit port completed
- 🐾 Maintainer Don Scorgie says
  - “Blazing fast. Startup goes from 2.8s to 1.9s.”
  - “API rocks. It's like a real **gtk+ API**. I can understand what's going on in it.”



# GtkPrint: Beautiful on paper

**Print API**  
Customise  
printing from  
your application  
(headers,  
footers, page  
settings coming  
soon)



**JavaScript**  
Print using the  
standard JS  
function

# Make it awesome

- 🐾 Use GTK+ to provide the core UI and shell
- 🐾 Use **WebView** to emphasise content
- 🐾 Allow users to customise their applications
- 🐾 **Create a community** around your application

Pidgin IM client

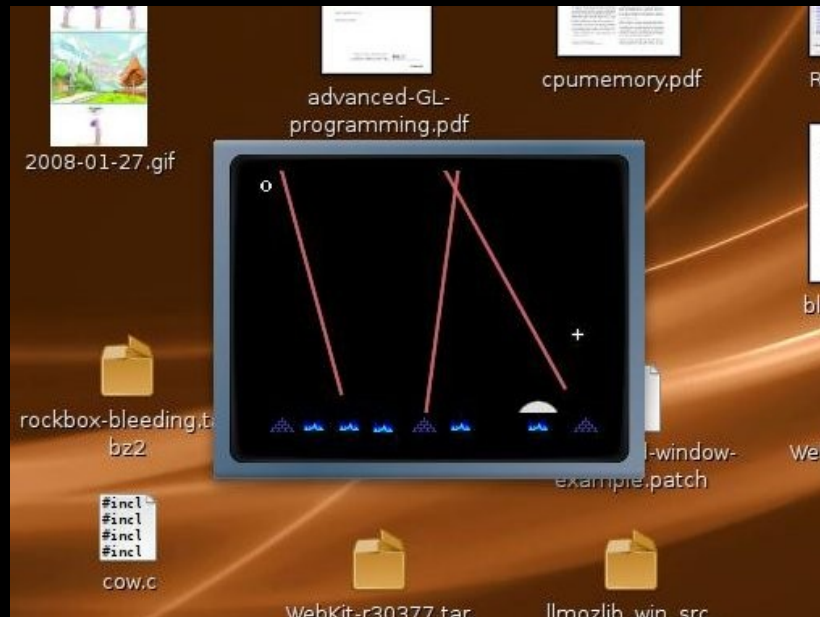
WebKit support by Sean Egan



# Host desktop widgets

```
GtkWidget *web_view = webkit_web_view_new ();  
webkit_web_view_set_transparent (  
    WEBKIT_WEB_VIEW (web_view), TRUE);
```

**HTML5 canvas**  
for drawing



**Shell access**  
for local operations

**D-Bus IPC**  
coming soon

**HTML5 local storage**  
for data persistence

# Invent a new look and feel

Composited desktop  
SVG circles



**Experiment**  
with fun new UI  
concepts



# Cross-platform

## ☞ Supported platforms

- GNU/Linux (X11, DirectFB)
  - All major distributions (**Debian** and **Gentoo** packaging teams in particular have helped a lot upstream)
- FreeBSD, DragonFlyBSD (X11)
- OS X (Native, X11, DirectFB)
  - Imendio working to complete the native GTK+ backend
- Windows (Native, X11)
  - Patches to be merged soon

# JavaScriptCore C API

## 🐾 Integrates with the GObject API

- Stable
- Fully documented

## 🐾 Portable

- Installed with WebKit/GTK+
- Ships with OS X

## 🐾 Standalone use

- Provides a light scripting engine for any application

## 🐾 Fastest mainstream JavaScript implementation

## 🐾 Use it to

- Export functions to JavaScript
- Invoke JavaScript
- Build dynamic language bindings

Portable application code		
OS X	Linux	Windows
WebKit	n/a	
WebKit/GTK+		

# Location-aware Web apps

- External module to expose location metadata to Web apps
- Implements the **locationaware.org** spec
- Spec not final
- Uses Gypsy GPS library (alternative location sources possible)
  - <http://folks.o-hand.com/iain/gypsy/>

```
var geolocation = navigator.getGeolocation();
geolocation.request(function(location) {
    alert(location.latitude+', '+location.longitude);
});
```

# WebKit and OpenMoko

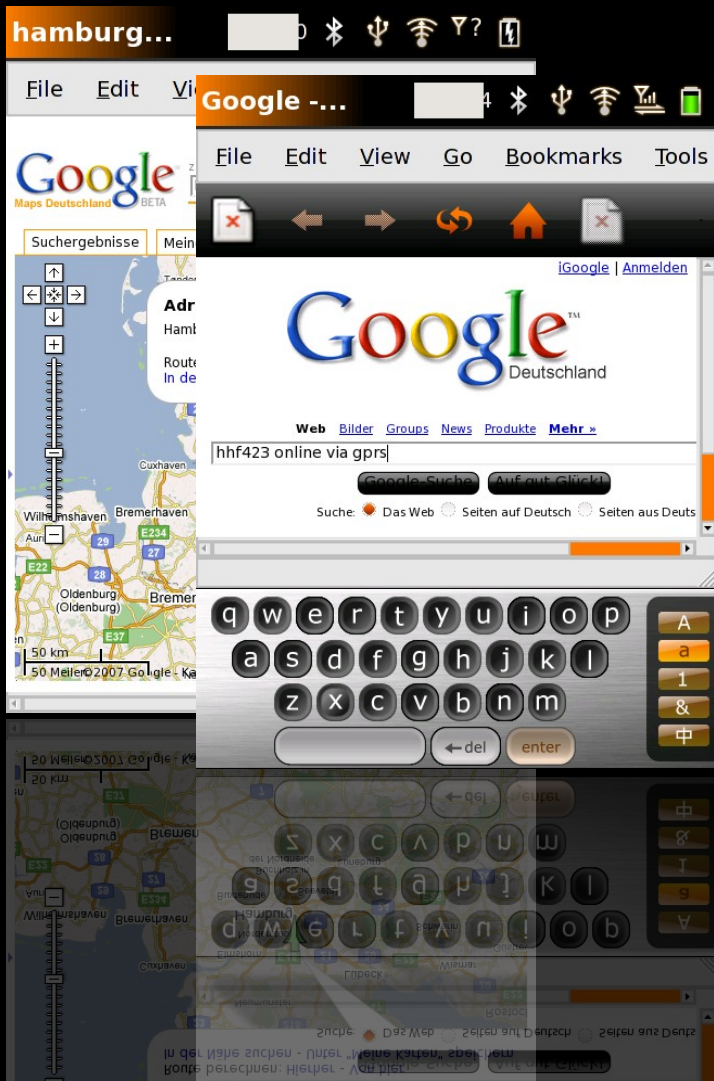
## 🐾 OpenMoko WebKit applications

– Browser

– Feed reader

🐾 Maintained by Holger Freyther  
(also a WebKit developer)

🐾 OpenMoko supports the WebKit  
project by providing a build server  
for continuous integration



# OLPC Sugar activity

Fast  
Loads fast  
renders fast

The screenshot shows a web browser window displaying the OLPC Wiki homepage. The browser's address bar shows 'BrowseKit Activity' and 'Share with: Private'. The page title is 'The OLPC Wiki' and it is noted as being redirected from the Home page. A green banner at the top states, 'This page is maintained by the OLPC team. Please use the discussion tab for your comments.' Below this, there are language selection options: deutsch, english, español, français, italiano, 日本語, 中文, монгол, português, română, and русский. A 'New!' badge indicates a recent update with ID# 56624. The main content area features a 'Welcome to the OLPC Wiki' message, followed by a 'What's new' section with several bullet points: Intel joining the team, B4 machines being distributed, a new skin, an OLPC badge, a report from Nigeria, a Bridge Builder Award, a new milestone, the latest release (Build 406.15 and Q2C18 firmware), and localization efforts in Spanish, Simplified Chinese, and Portuguese. A 'Latest Releases' box highlights the stable build and firmware versions, along with links for updates and bug reports. The page also includes a sidebar with navigation links for 'About OLPC', 'About the laptops', 'About the site', and a 'Toolbox'.

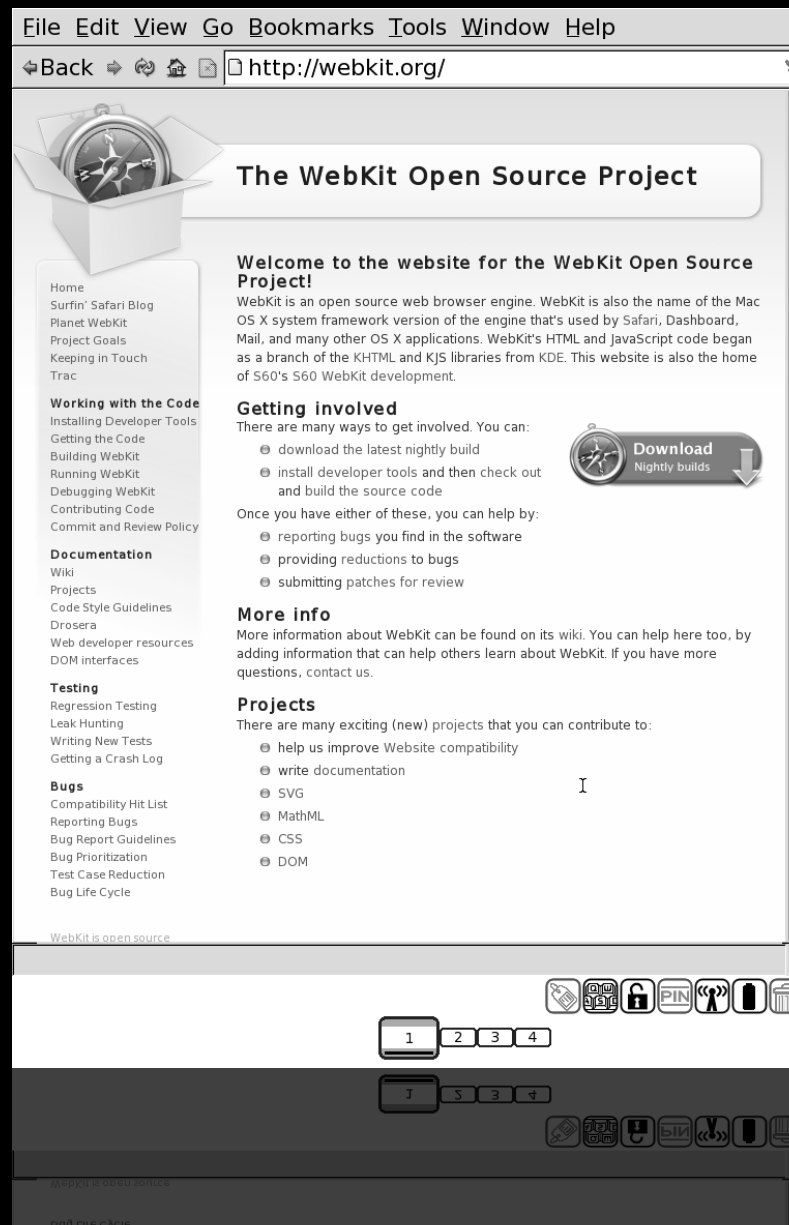
Light  
Small memory  
footprint

# WebKit and OLPC

- WebKit/GTK+ Sugar activity done by Dan Winship (Red Hat) in one day
- John Palmieri compared it with the existing Mozilla-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”

# WebKit e-paper with the Irex iLiad

Midori browser on iLiad  
WebKit port by Adam Boeglin



## Open questions

How do we reduce repaints?  
Can we save memory for grayscale displays?

# WebKit for Maemo

**Hildon extensions**  
for text entry  
and UI elements



**4.7X**  
faster than  
**MicroB**  
SunSpider  
JS/AJAX testsuite



# Putting the Web in GTK+

- ☞ Berlin, March 2008, GTK+ hackfest goals
  - **Enhance** the GTK+ core to meet browser needs
  - **Extend** the GTK+ toolchain with Web capabilities
- ☞ GTK+ *Web integration squad*
  - Alp Toker
  - Christian Persch

# The return of the online desktop?

- ☞ Why did the GNOME online desktop fail?
  - The data model was there
  - The IPC system was in place (D-Bus)
  - Did the *lack of a Web content engine* before 2007 kill the online desktop?
- ☞ Reviving the online desktop; a good idea?
  - WebView could make the online desktop practical today

“The *next big thing* is the one that makes the last big thing usable.”

**Blake Ross**

# Announcing the offline desktop

## ☞ Let users **take back their data**

- Provide hybrid Web/GTK+ user-interfaces as a frontend to *local and personal-area data stores*
- Use GTK+ to deliver rich internet applications that aren't riddled with branding and advertising

## ☞ **Pack a WebView in your application and get started today**

# Get involved!

<http://live.gnome.org/WebKit>

<http://www.webkit.org>

IRC: #webkit, #webkit-gtk / FreeNode

**WebKit** GTK+

