WebKit and Blink: Open Development Powering the HTML5 Revolution

Juan J. Sánchez

LinuxCon 2013, New Orleans

Myself, Igalia and WebKit

- Co-founder, member of the WebKit/Blink/Browsers team
- Igalia is an open source consultancy founded in 2001
- Igalia is Top 5 contributor to upstream WebKit/Blink
- Working with many industry actors: tablets, phones, smart tv, set-top boxes, IVI and home automation.

Outline

- The WebKit technology: goals, features, architecture, code structure, ports, webkit2, ongoing work
- The WebKit community: contributors, committers, reviewers, tools, events
- Contributing to WebKit: bugfixing, features, new ports
- Blink: history, motivations for the fork, differences, status and impact in the WebKit community

WebKit: the technology

The WebKit project

- **Web rendering engine** (*HTML*, *JavaScript*, *CSS*...)
- The engine is the product
- Started as a fork of KHTML and KJS in 2001
- *Open Source* since 2005
- Among other things, it's **useful for**:
    - Web browsers
    - Using web technologies for UI development

Goals of the project

- **Web Content Engine**: HTML, CSS, JavaScript, DOM
- **Open Source**: BSD-style and LGPL licenses
- **Compatibility**: regression testing
- **Standards Compliance**
- **Stability**
- **Performance**
- **Security**
- **Portability**: desktop, mobile, embedded...
- **Usability**
- **Hackability**

Goals of the project

NON-goals:

- "It's an engine, not a browser"
- "It's an engineering project not a science project"
- "It's not a bundle of maximally general and reusable code"
- "It's not the solution to every problem"

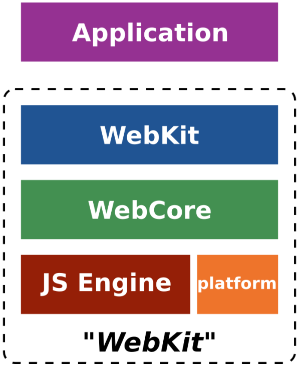
`http://www.webkit.org/projects/goals.html`

WebKit features

- HTML and XML support
- JavaScript support (ECMAScript 5.1)
- CSS 2.1, CSS 3 support
- SVG support
- Support for Plugins (NPAPI, WebKit Plugins)
- HTML5 support: multimedia, 3D graphics, advanced CSS animations and transformations, drag'n'drop, offline & local storage, connectivity...
- Accessibility support
- Q&A infrastructure: review process, continuous integration, 30.000 regression tests, API tests...
- Passing ACID3 with 100/100 tests since March 2008

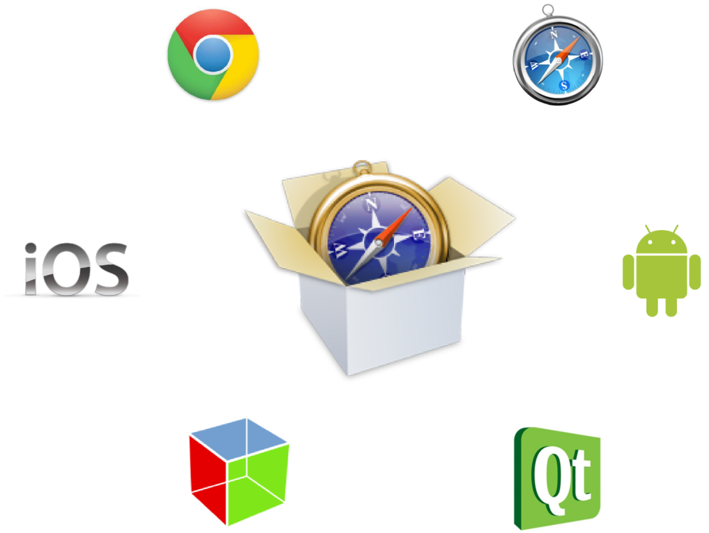
WebKit Architecture

From a simplified point of view, WebKit is structured this way:



- **WebKit**: thin layer to link against from the applications
- **WebCore**: rendering, layout, network access, multimedia, accessibility support...
- **JS Engine**: the JavaScript engine. JavaScriptCore by default, but can be replaced (e.g. V8 in Chromium)
- **platform**: platform-specific *hooks* to implement generic algorithms

What is a WebKit port?



How many WebKit ports are there?

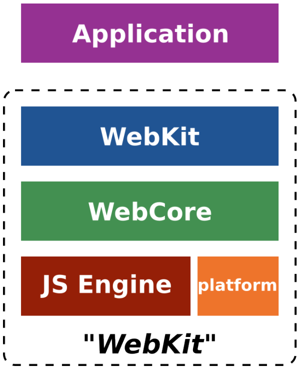
WebKit is available for different platforms:

- Main upstream ports during the past 2 years:
    - Mac OS X, iOS
    - GTK+ based platforms (GNOME)
    - Qt based platforms (KDE)
    - Enlightenment Foundation Libraries (EFL, Tizen)
    - Google Chromium / Chrome
    - New proposal: WebKitNIX
- Other ports: wxWidgets, Brew MP, Symbian devices (S60), Win32, BlackBerry, Adobe Integrated Runtime (Adobe AIR)

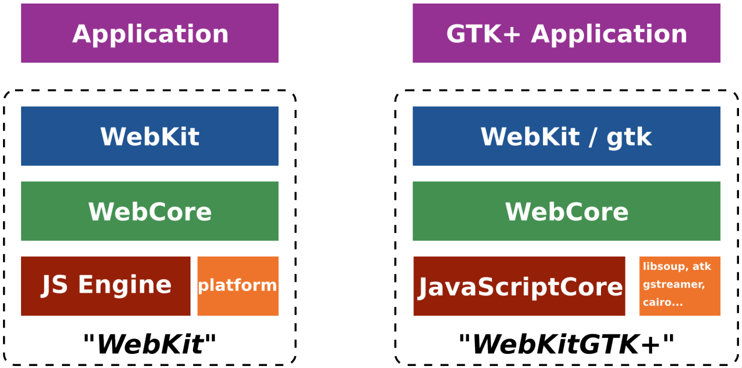
Some WebKit based browsers

- Amazon Kindle
- Arora
- BOLT browser
- Epiphany browser
- Google Chrome
- iCab (version >= 4)
- Iris Browser
- Konqueror
- Midori
- Nintendo 3DS
- OWB
- OmniWeb
- PS3 web browser
- RockMelt
- Safari
- SRWare Iron
- Shiira
- Sputnik for MorphOS
- Stainless
- Steel for Android
- TeaShark
- Uzbl
- Web Browser for S60 (Nokia)
- WebOS Browser

Architecture of a WebKit port



Architecture of a WebKit port



How do we use a WebKit port?

- **The WebView** ***widget***:
  A platform-specific widget that renders web content.

  It's the **main component** and it's useful for:
  - Loading URIs or data buffers pointing to HTML content
  - Go fullscreen, text/text+image zooming...
  - Navigate back and forward through history...

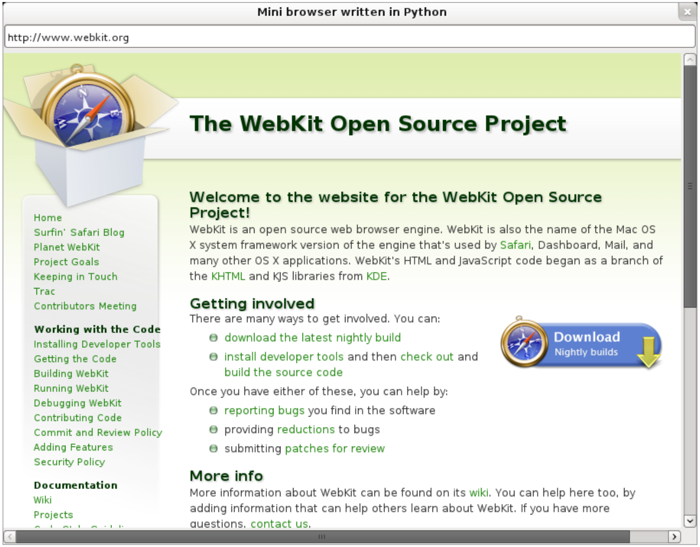
- **Events handling**:
  Allows embedders to get notified when something important happens or when some input is needed.

  Some examples of these events:
  - Getting notified when a load finished or failed
  - Asking permission for navigating to an URI
  - Requesting authorization for something..

A minibrowser written in Python

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

import gtk
import webkit

def entry_activated_cb(entry, embed):
    embed.load_uri(entry.get_text())

# Widgets and signals
window = gtk.Window()
window.set_default_size(800, 600)
window.set_title("Mini browser written in Python")
embed = webkit.WebView(); # WebKit embed
entry = gtk.Entry()
entry.connect('activate', entry_activated_cb, embed)
scroller = gtk.ScrolledWindow()
scroller.add(embed)

# Pack everything up and show
vbox = gtk.VBox(False, 5)
vbox.pack_start(entry, False, False)
vbox.pack_start(scroller)
window.add(vbox)
window.show_all()

# Load a default URI and run
embed.load_uri("http://www.webkit.org")
gtk.main()
```

A minibrowser written in Python


Mini browser written in Python

http://www.webkit.org

The WebKit Open Source Project

Home
Surfin' Safari Blog
Planet WebKit
Project Goals
Keeping in Touch
Trac
Contributors Meeting

Working with the Code
Installing Developer Tools
Getting the Code
Building WebKit
Running WebKit
Debugging WebKit
Contributing Code
Commit and Review Policy
Adding Features
Security Policy

Documentation
Wiki
Projects

Welcome to the website for the WebKit Open Source Project!

WebKit is an open source web browser engine. WebKit is also the name of the Mac OS X system framework version of the engine that's used by Safari, Dashboard, Mail, and many other OS X applications. WebKit's HTML and JavaScript code began as a branch of the KHTML and KJS libraries from KDE.

Getting involved

There are many ways to get involved. You can:

- download the latest nightly build
- install developer tools and then check out and build the source code

Once you have either of these, you can help by:

- reporting bugs you find in the software
- providing reductions to bugs
- submitting patches for review

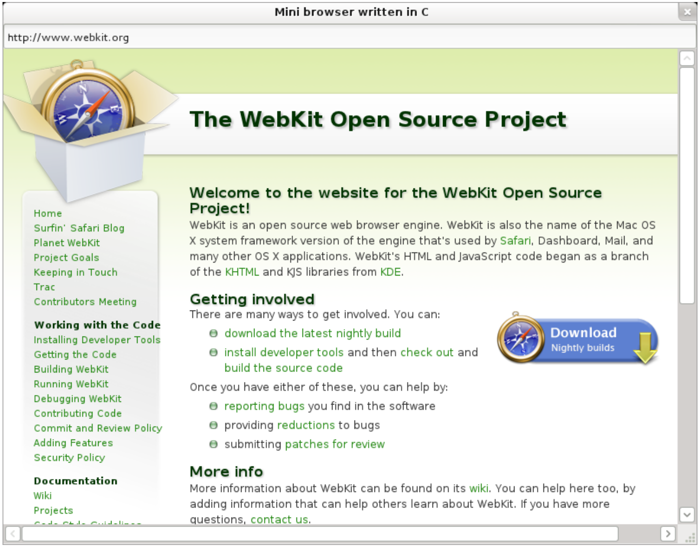
More info

More information about WebKit can be found on its wiki. You can help here too, by adding information that can help others learn about WebKit. If you have more questions, contact us.

A minibrowser written in C

```c
#include <webkit/webkit.h>
static void entry_activated (GtkEntry *entry, WebKitWebView *embed)
{
  webkit_web_view_load_uri (embed, gtk_entry_get_text (entry));
}
int main (int argc, char** argv)
{
  gtk_init (&argc, &argv);

  /* Widgets and signals */
  GtkWidget *window = gtk_window_new (GTK_WINDOW_TOPLEVEL);
  gtk_window_set_default_size (GTK_WINDOW (window), 800, 600);
  gtk_window_set_title (GTK_WINDOW (window), "Mini browser written in C");
  GtkWidget *embed = webkit_web_view_new();
  GtkWidget *entry = gtk_entry_new();
  g_signal_connect (entry, "activate", G_CALLBACK (entry_activated), embed);
  GtkWidget *scroller = gtk_scrolled_window_new(NULL, NULL);
  gtk_container_add (GTK_CONTAINER(scroller), embed);

  /* Pack everything and show */
  GtkWidget *vbox = gtk_box_new (GTK_ORIENTATION_VERTICAL, 0);
  gtk_box_pack_start (GTK_BOX(vbox), entry, FALSE, FALSE, 0);
  gtk_box_pack_start (GTK_BOX(vbox), scroller, TRUE, TRUE, 0);
  gtk_container_add (GTK_CONTAINER(window), vbox);
  gtk_widget_show_all (window);

  /* Load a default URI and run */
  webkit_web_view_load_uri (WEBKIT_WEB_VIEW (embed), "http://www.webkit.org");
  gtk_main();
  return 0;
}
```

A minibrowser written in C



The WebKit Open Source Project

Home
Surfin' Safari Blog
Planet WebKit
Project Goals
Keeping in Touch
Trac
Contributors Meeting

Working with the Code
Installing Developer Tools
Getting the Code
Building WebKit
Running WebKit
Debugging WebKit
Contributing Code
Commit and Review Policy
Adding Features
Security Policy

Documentation
Wiki
Projects

Welcome to the website for the WebKit Open Source Project!

WebKit is an open source web browser engine. WebKit is also the name of the Mac OS X system framework version of the engine that's used by Safari, Dashboard, Mail, and many other OS X applications. WebKit's HTML and JavaScript code began as a branch of the KHTML and KJS libraries from KDE.

Getting involved

There are many ways to get involved. You can:

- download the latest nightly build
- install developer tools and then check out and build the source code

Once you have either of these, you can help by:

- reporting bugs you find in the software
- providing reductions to bugs
- submitting patches for review

More info

More information about WebKit can be found on its wiki. You can help here too, by adding information that can help others learn about WebKit. If you have more questions, contact us.

What is WebKit2?

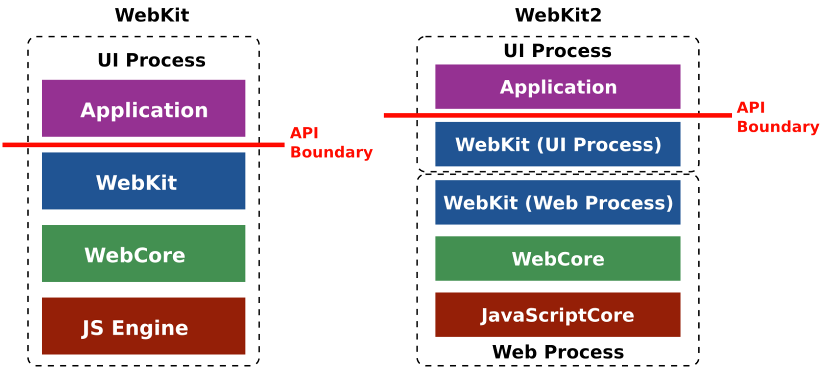
- New API layer designed to support a split process model (First release by Apple on April 8th, 2010[1]).
- Different to Chromium's multi-process implementation

  It's bundled in the framework (reusable)
- Different processes take care of different tasks:
  - **UI process**: the WebView *widget*, application UI
  - **Web process**: loading, parsing, rendering, layout...
  - **Plugin process**: each plugin type in a process
- It comes with Inter-Process Communication (IPC) mechanisms to communicate those processes bundled-in

`http://trac.webkit.org/wiki/WebKit2`

igalia

WebKit VS WebKit2



WebKit2 vs WebKit1

Pros:

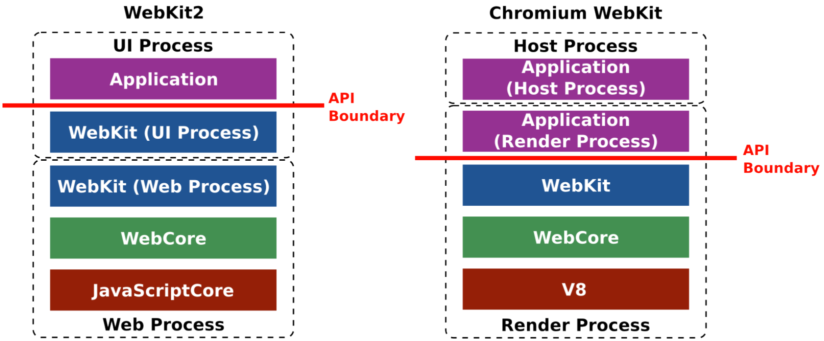
- Isolation
- Security
- Performance
- Stability

WebKit2 vs WebKit1

Cons:

- Higher resource requirements
  - Multiple processes instead of just one
  - At the same time easier to release resources
- Complexity
  - Coding
  - Debugging
- WebKit1 and WebKit2 maintenance

WebKit2 VS Chromium



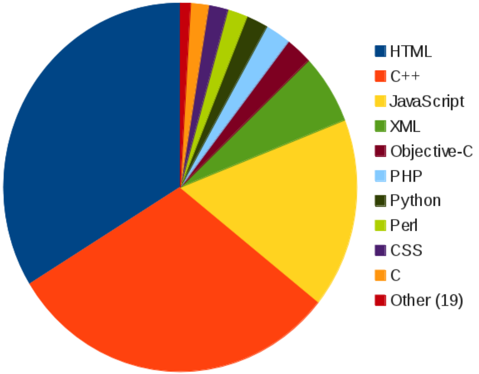
WebKit2: current status

- Apple, Qt and GTK+ already released WebKit2 browsers
- WebKit1 moving to maintenance mode for most ports
- Cross-platform and non-blocking C API available
- Most challenges of the split process model solved
- Lots of new architectural changes about to come

The Source Code in numbers

According to Ohloh on 2013 Sep 10th, **lines of code per language**, without considering blank lines nor comments:

Language	LoC	%
HTML	1,707,944	30.3 %
C++	1,355,784	31.0 %
JavaScript	886,308	20.5 %
XML	163,012	2.8 %
Objective-C	119,890	2.6 %
C	105,259	2.9 %
PHP	97,627	2.4 %
CSS	90,581	1.7 %
Python	76,040	2.0 %
Perl	75,988	1.9 %
OpenGL Shad	26,234	1.0 %
Other (18)	50,000	0.9 %
Total	4,754,103	



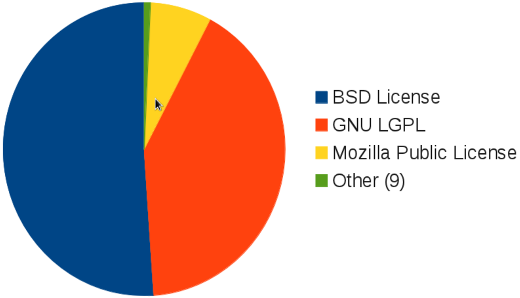
https://www.ohloh.net/p/WebKit/analyses

Just considering C++, Objective-C and C files, we have almost 1.6M LoC!

The Source Code in numbers

According to Ohloh on 2013 September 10th, **files per license**:

License	Files	%
BSD License	4427	51.09 %
GNU LGPL	3568	41.18 %
MPL	607	7.01 %
Other (9)	63	0.73 %
Total	8665	



`https://www.ohloh.net/p/WebKit/analyses`

New WebKit code will always be under the terms of **either the LGPL 2.1+ or the BSD license**. Never GPL or LGPL 3.

High Level source code overview

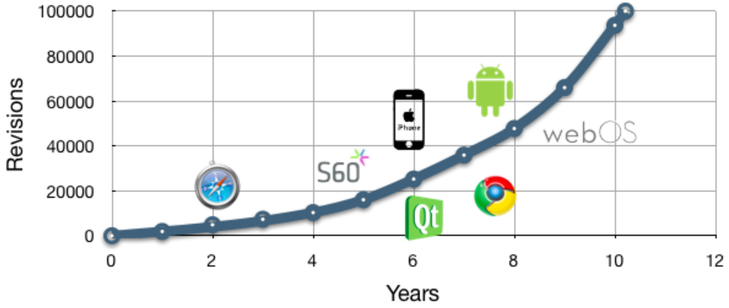
- **Source/**: the code needed to build WebKit. That is, WebCore, JavaScriptCore, WebKit and WebKit2
- **LayoutTests/**: layout tests reside here (More than 30000!). They test the correctness of WebKit features
- **ManualTests/**: specific cases not covered by automatic testing
- **PerformanceTests/**: measure run-time performance and memory usage
  See `perf.webkit.org` for results
- **Tools/**: tools and utilities for WebKit development. Small test applications, tools for testing, helper scripts...
- **Websites/**: code and pages for WebKit related sites

Tracking ongoing work in WebKit

- Webkit is a big beast and a lot of organizations with different agendas are working in different parts:
  - Implementing new standards (like the CSS shaders from Adobe, or CSS3 GCPM from Apple)
  - Improvements in architecture, performance and internal code (WebKit2)
- On top of this there is the maintenance work (testing, continuous integration, bugfixing)
- No single centralized place to follow all the information: blogs, mailing lists, IRC, etc.

WebKit: The community

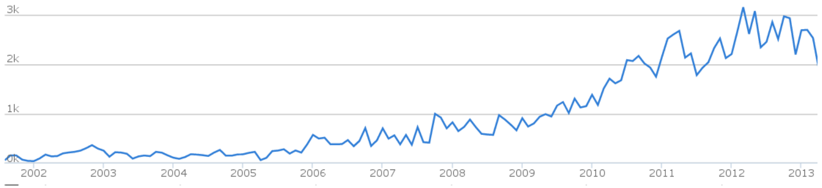
A bit of history



Source: `http://ariya.ofilabs.com/2011/11/one-hundred-thousand-and-counting.html`

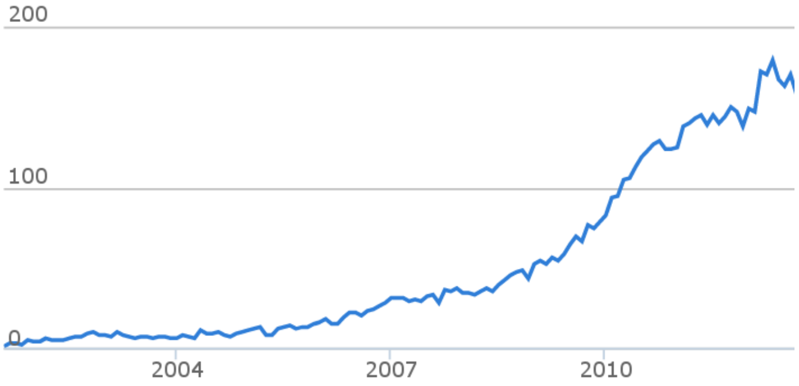
The WebKit Project in numbers

Commits per month till 2013:



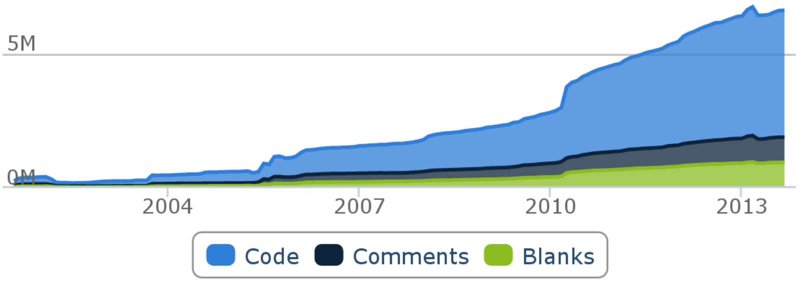
The WebKit Project in numbers

Contributors per month::



The WebKit Project in numbers

Evolution in the number of lines of code



Activity of Companies

- Based on Bitergia's report[2]
- Based on reviewed commits
- "Gardening" commits filtered out
- From the beginning of the project till beginning of 2013

[2] http://blog.bitergia.com/2013/02/06/report-on-the-activity-of-companies-in-the-webkit-project/

Activity of Companies

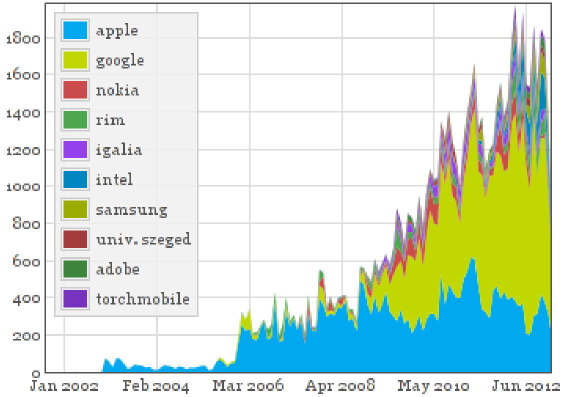


Figura: Commits per company (monthly)

Activity of Companies

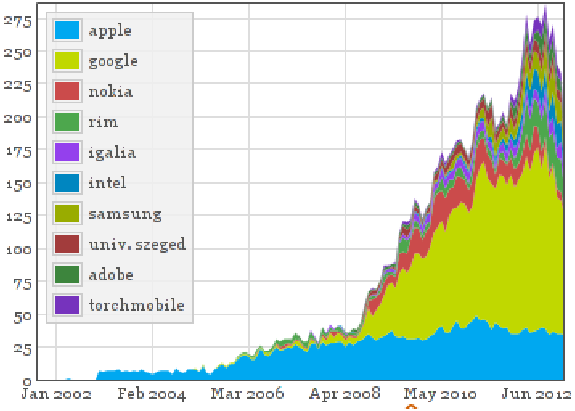


Figura: Active authors per company (monthly)

Activity of Companies

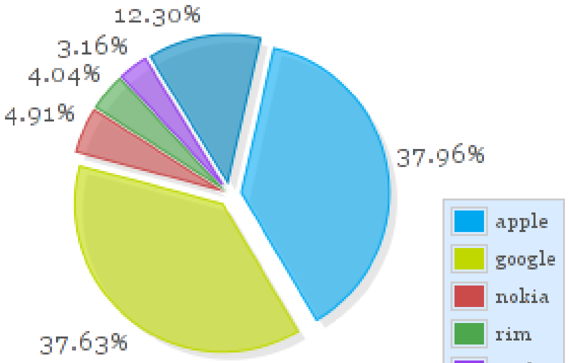


Figura: Commits per company

Activity of Companies

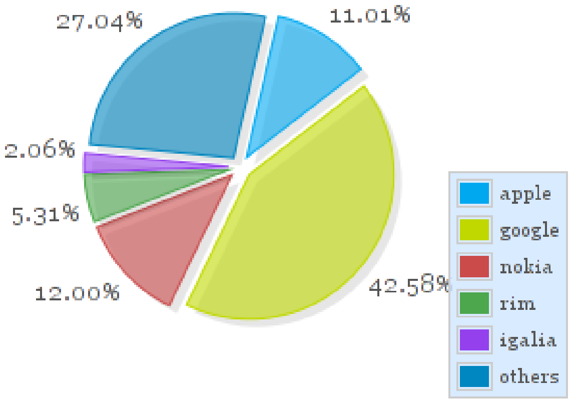


Figura: Active authors per company

Activity of Companies

Some conclusions from the authors:

- Google and Apple leading the project
- The diversity of the project has been increasing
- Contributions from other parties >25 % and growing
- > 20 companies actively contributing to WebKit
- 1500-2000 commits per month
- 387 committers from 29 different institutions

Committers and Reviewers

WebKit Committer

A WebKit Committer should be a person we can trust to follow and understand the project policies about checkins and other matters.

Has commit rights to the public SVN repository.

WebKit Reviewer

A WebKit Reviewer should be a person who has shown particularly good judgment, understanding of project policies, collaboration skills, and understanding of the code.

A WebKit Committer who can review other's patches.

Copyright for contributions

- There is no copyright transfer for the contributions
- Committers sign some papers where they commit to good behaviour

Releases

- There are no releases of WebKit itself
- Each port manages the release cycle, typically aligned with the target platform schedule

Coordination and communication tools

- Website: `http://www.webkit.org/`

  Port specific Websites (e.g. `http://webkitgtk.org/`)
- Wiki: `http://trac.webkit.org/wiki`
- Blogs: `http://planet.webkit.org/`
- Source Code:
  - SVN: `http://svn.webkit.org/repository/webkit`
  - Git mirror: `git://git.webkit.org/WebKit.git`
- Bugzilla: `https://bugs.webkit.org/`
- Buildbots: `http://build.webkit.org/`
- Mailing lists: `http://lists.webkit.org/mailman/listinfo.cgi`
- IRC (*irc.freenode.net*): *#webkit* and *#webkitgtk+*

The WebKit Contributors Meeting

- Meeting for contributors to the WebKit project
- Organized in an “unconference”-like format
- Extremely useful to advance on some topics: Implementation of new APIs, WebKit2, accelerated compositing, helper tools, QA infrastructure...
- Yearly held in Cupertino, California. Hosted by Apple

How to contribute

Types of contributions

- Bugfixing and new features in:
  - An existent port
  - The core components: webcore and JSC/V8
- Creation and maintenance of a new port

Guidelines for contributing patches to WebKit

1. Get and build the code from the SVN repository
2. Choose or create a bug report to work on
3. Code your changes and make sure you include new regression or unit tests if needed
4. Create a patch for your changes and submit it asking for review over it to appropriate reviewers
5. Update and change your patch as many times as needed
6. Once approved, land your patch or ask a committer/reviewer to do it
7. Watch for any regressions it might have caused

Creating a port: what needs to be done

- High level API (WebKit1 and WebKit2)
- Low level backend specific implementation
  - Web Template Framework (WTF): memory management, threading, data structures (vectors, hash tables, bloom filters, ...) numerical support, etc.
  - JSC vs V8
  - Networking: HTTP, DNS, cookies, etc.
  - Graphics: 2D/3D rendering, compositing, theming, fonts
  - Multimedia: media player for audio and video tags
  - DOM bindings
  - Accessibility
  - Smaller tasks: clipboard, popup and context menus, cursors, etc.
- Other things: favicons, plugins, downloads, geolocation, settings, navigation policies, etc.

Creating a port: the social side

- Difficult without full-time reviewers
- Reuse from other ports as much as possible
- Try to work upstream from the very beginning
- The risk of forking is big, the project moves fast
- Focus on testing and continuous integration
- Port-specific events and communication tools

Present and future

Google's Departure. Blink

- Google announced on April 3rd that they would be forking WebKit and creating Blink
- Motivations according to Google:
    - They were not using WebKit2 anyway
    - Easier to do ambitious architectural changes after the fork
    - Simplification of the codebase in Blink
- Tension between Apple and Google before the fork
    - Architectural decisions: NetworkProcess
    - Code governance: Owners
- Big shock within the WebKit community

Differences between WebKit and Blink

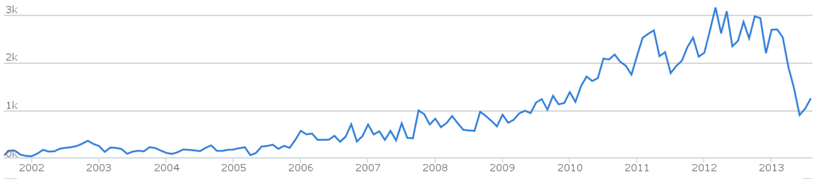
- Removes the concept of ’port’ as it was defined in WebKit (deep platform integration): Skia, V8 and other libraries cannot be replaced
- Still possible to use Blink in other platforms, but now integration happens at Content level
- Only the rendering engine. Multi-process architecture is still in Chromium
- Peer review process (committers, reviewers, owners) more flexible in Blink
- Many architecture changes are started to be implemented

Consequences of Blink for WebKit

- Google was the main contributor by # of commits
- Opera joined WebKit then moved to Blink. Other companies and communities thinking what to do.
- Apple's position more dominant. Likely to relax Owners policy
- Several WebCore modules left orphan. Other hackers assuming WebCore modules maintainership
- WebKit developers porting patches from/to Blink
- Many hacks to accomodate Chromium removed. Engines likely start to diverge at faster pace

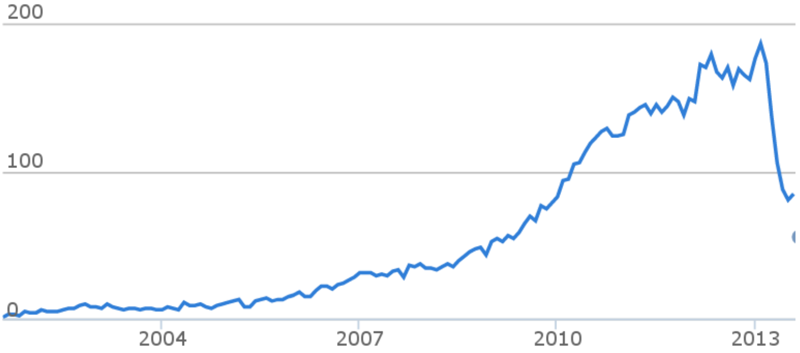
Impact of Blink in numbers

Commits per month in WebKit, including last months:



Impact of Blink in numbers

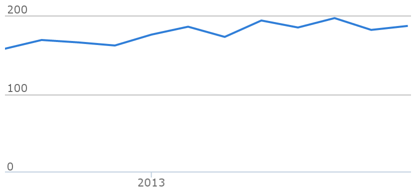
Contributors per month in WebKit, including last months::



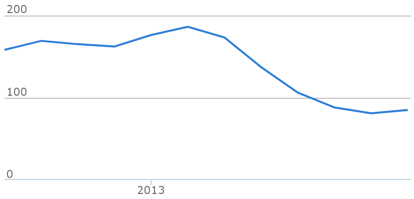
Impact of Blink in numbers

Contributors per month in 2013::

Blink:

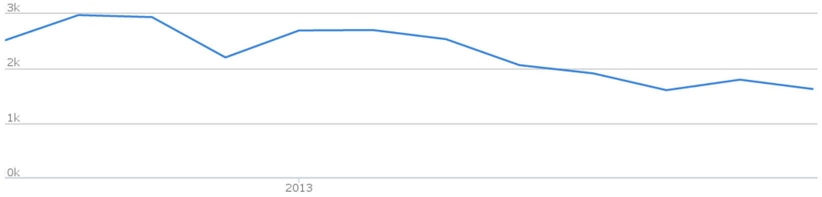


WebKit:

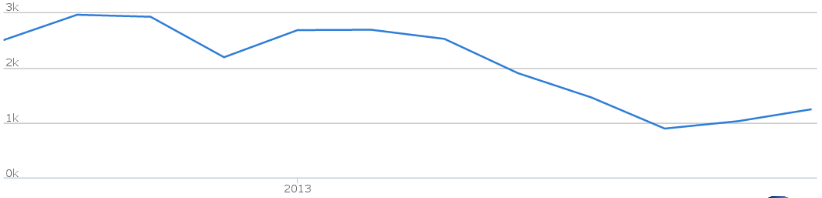


Impact of Blink in numbers

Commits per month in 2013, Blink::



Commits per month in 2013, WebKit::



The Blink community is consolidating

- Dynamics of the community still being consolidated
- Opera already actively contributing and using Blink
- Qt recently announced that they are moving to Blink
- In a few weeks/months we will know how everything settles down
- BlinkOn event, end of September at Google's office in San Francisco

Open questions about Blink for WebKit

- How open the dynamics of each community will be?
- Which project will innovate faster and keep higher quality and performance standards?
- Which option will upstream ports be based on?
- Which option will companies developing platforms and solutions use?

Conclusions

- WebKit is an Open Source web engine powering lots of applications (not only browsers!) out there
- The only true Open Source alternative for embedders for many years. Clearly defined and modular architecture. Port friendly. Complex and fast moving project
- Developed by a community of organizations and individuals with different interests, collaborating together. Lots of contributors. Appropriate policies in place to handle permissions and responsibilities in the project
- The recent announcement of Blink is changing the WebKit community a lot and it is still to be seen how the situation in terms of open source dynamics and project health and quality will be in a few months.

Thank you!

Juan J. Sánchez
jjsanchez@igalia.com