WebKit Architecture

Alexandru Chiculita, Computer Scientist, WebKit team

1997

KDE HTML Widget

1998

KHTML

2001

Forked by Apple

2003

First Safari Release

Projects

JavaScriptCore

The JavaScript engine used in Safari

WTF - Web Template Framework

A bunch of C++ templates that tries to replace STL

WebCore

The rendering engine. Most of the code is here.

WebKit

Old platform integration code.

Used by Chromium which has its own cross-process code.

WebKit2

New platform integration code used in Safari since version 5.

Based on a cross process architecture.

Attention! **This is not used by Chromium.**

JavaScriptCore

Used in Safari

JavaSript Parser, JSON Parser

Byte Compiler
using internal bytecode format

Assembler
uses code patching at runtime -> it needs writeable code memory

Data flow graph
new initiative to optimize the generated code based on compile time speculations

Interpretor
runs the generated bytecode

Regexp engine
with JIT support

Garbage collector
mark-and-sweep

Runtime
all the JS global objects (Date, String, Number, etc.)

Debugger, Profiler

WebCore

Resource loaders

HTML & XML parsers, DOM

SVG & SMIL

CSS
Parser, Selectors, Animations

Rendering and layout

Bindings generator
IDL files: JSC, V8, ObjC

HTML5 features
audio, video, canvas, WebGL, notifications

WebInspector

Platform integration
Graphics, Fonts, Sound, Video

WebKit / WebKit2

The only task for these projects is to expose platform functionality to WebCore layer.

The port maintainers implement and extend this code.

Examples of APIs exposed

NSWebView from Mac

Android's Java WebView API

What happens when a page loads in WebKit?

Network request to load the page

The HTML parser creates HTML Elements

Elements are attached to DOM and RenderObjects are created

Resulting tree of Frames with RenderViews and Documents

Load events are fired and page is painted

Is it always creating RenderObjects?

No!

WebKit is optimized for performance!

Processing is postponed until the last moment

Think about performance regressions

How much memory is your patch using?

How is it impacting performance?

Is it useful for all the existing sites?

Remember

JavaScript should never see intermediate layout states!

WebKit is a secure runtime

HTML, JS, CSS can NOT be trusted

For example in Flex, the app developer makes sure the application code doesn't break the layout

Kernel of the Web

Do not accept incorrect CSS values or behaviors

It may happen as a side effect of parsing
It may be needed to maintain it forever

Always think about cross-domain access

GPU acceleration

Layer Blending

Activated using -webkit-transform and CSS animations

Implemented in RenderLayer

Drawing pipeline is platform dependent

2D drawing acceleration

No 2D GPU acceleration in WebKit

CoreGraphics on Lion and Skia have GPU accelerated 2D drawing

WebGL

Managed access to OpenGL ES APIs from JavaScript

Q & A

achicu@adobe.com