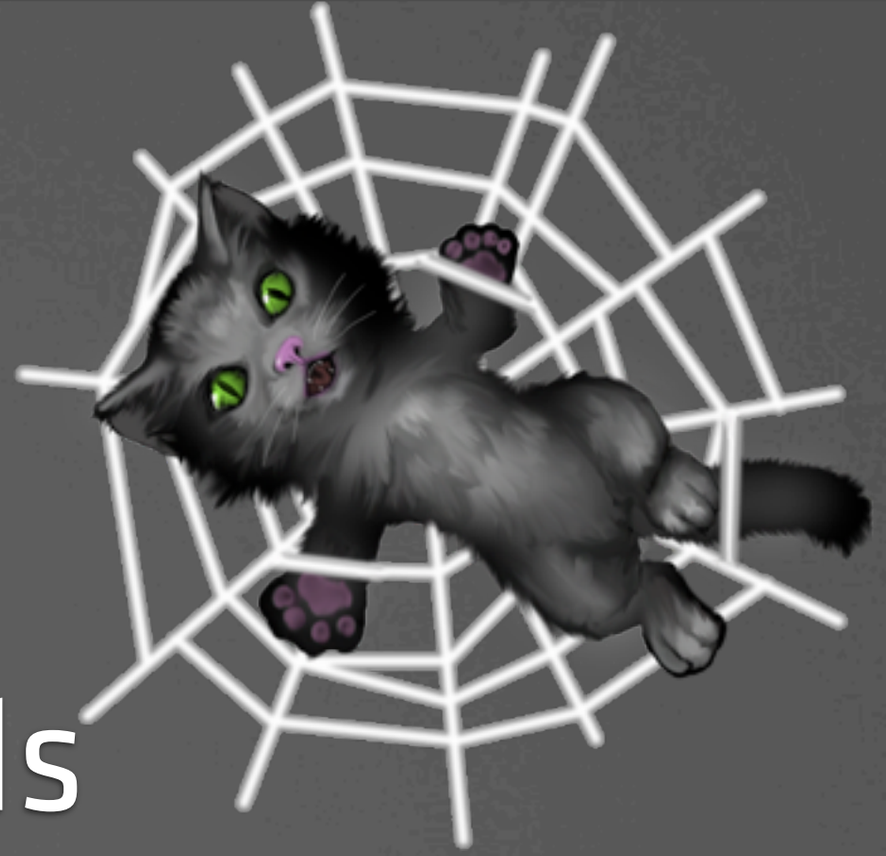
Hacking WebKit & Its JavaScript Engines



Jarred Nicholls
Work @ Sencha
WebKit Committer

Doing webkitty things...

Our Mission if we choose to accept it

Define: What is WebKit?

Some Simple Ways to Leverage WebKit

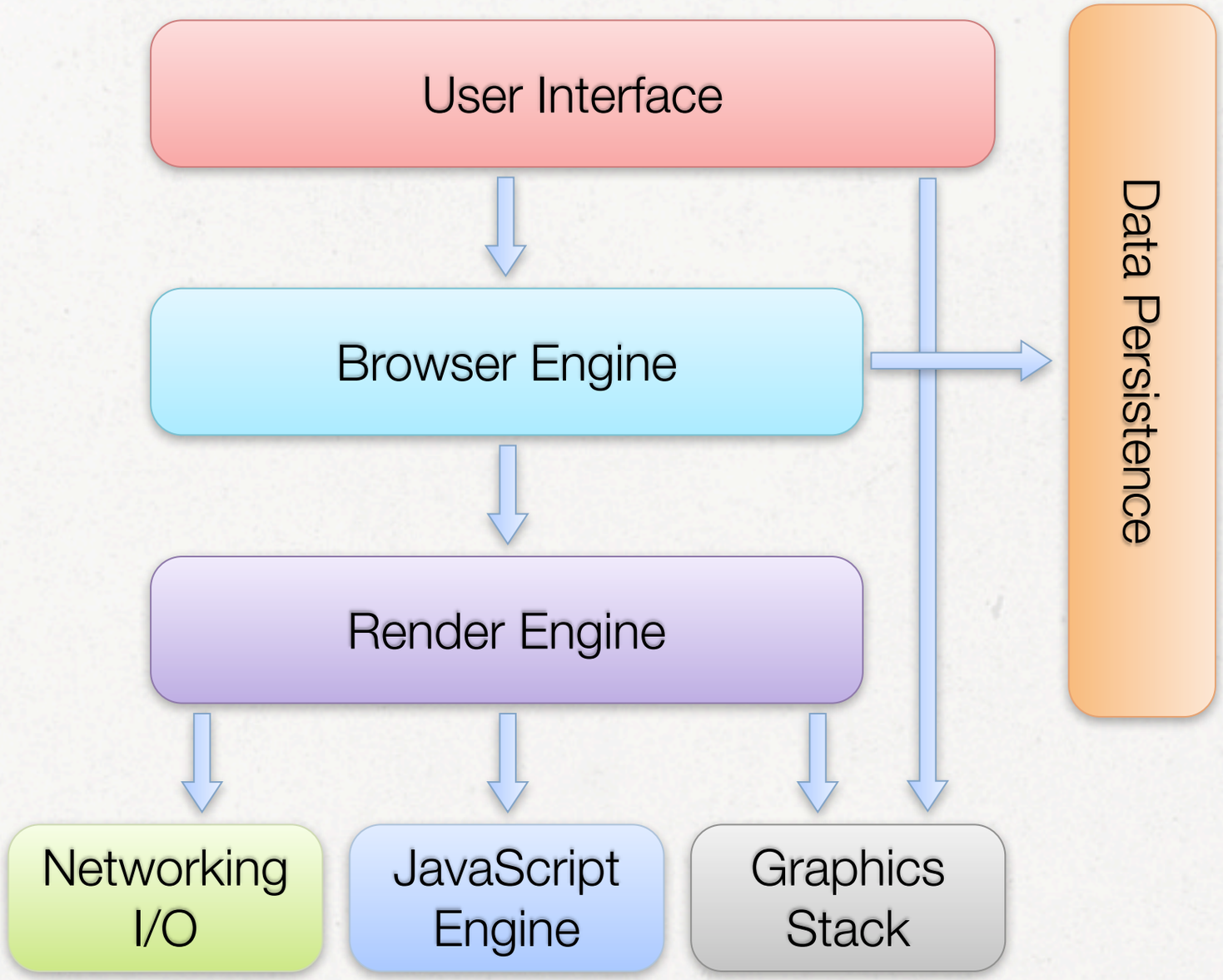
- Little investment, Large ROI
- Web Inspector Hacking
- “Headless” WebKit

What is WebKit?

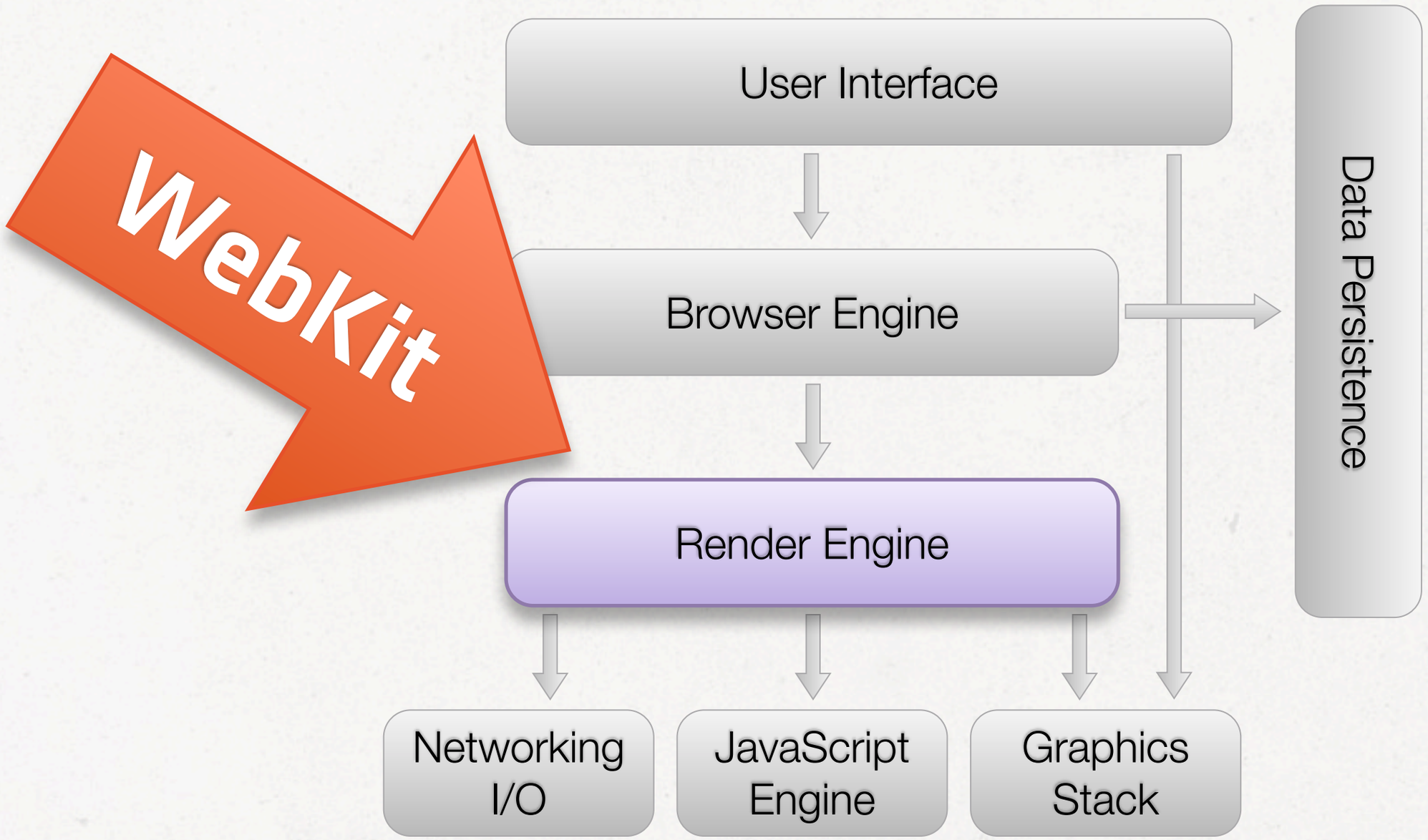
≠


N
NE
NW
E
W
SE
S
SW

Browser at a High Level



Browser at a High Level



≠


N
NE
NW
E
W
SE
S
SW


N
NE
E
SE
S
SW
W
NW

WebKit is EVERYWHERE!

Even in your living room!

Portability

The WebKit project seeks to address a variety of needs. ***We want to make it reasonable to port WebKit to a variety of desktop, mobile, embedded and other platforms.*** We will provide the infrastructure to do this with tight platform integration, reusing native platform services where appropriate and providing friendly embedding APIs.

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

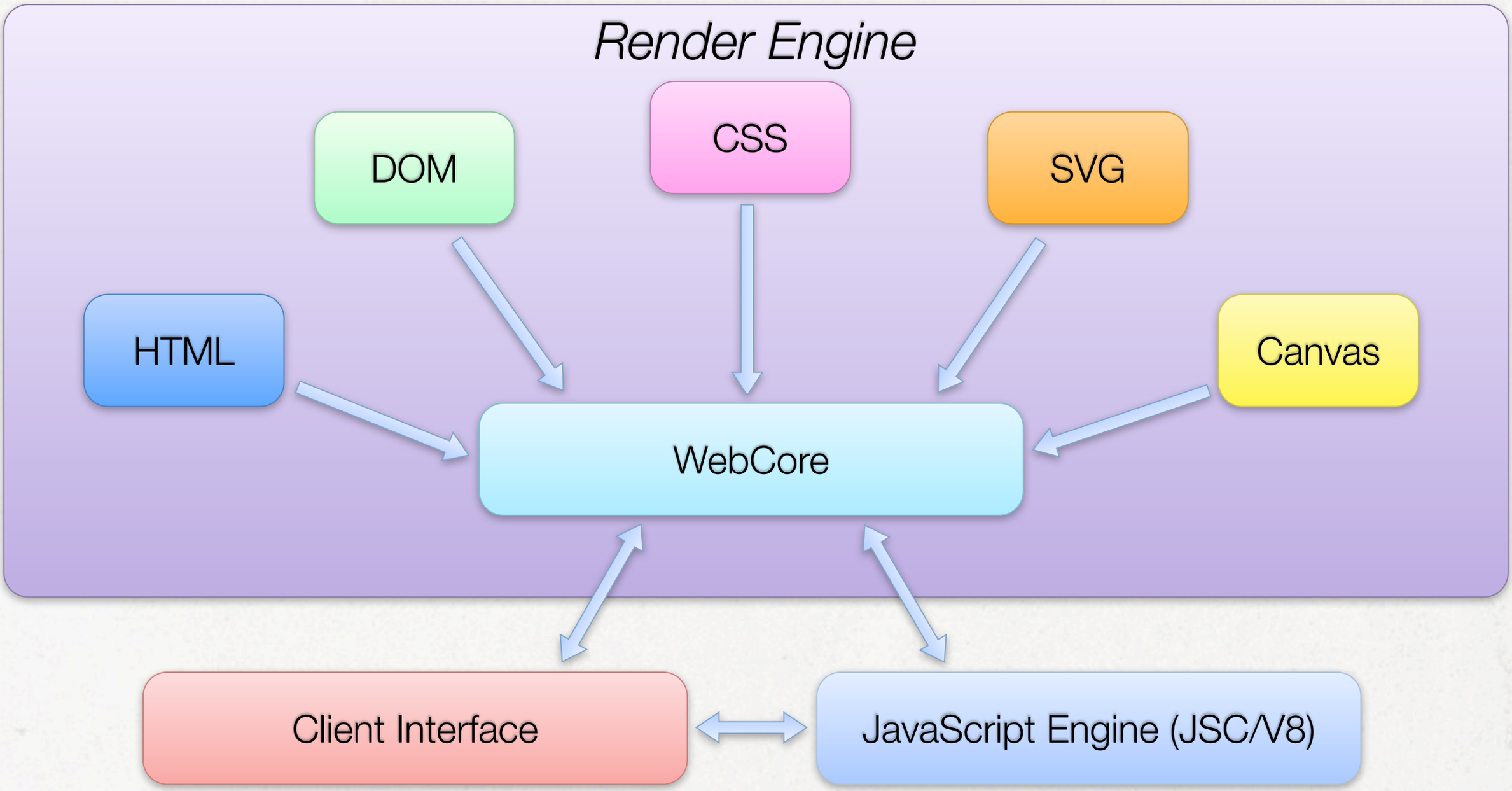
BlackBerry

iOS

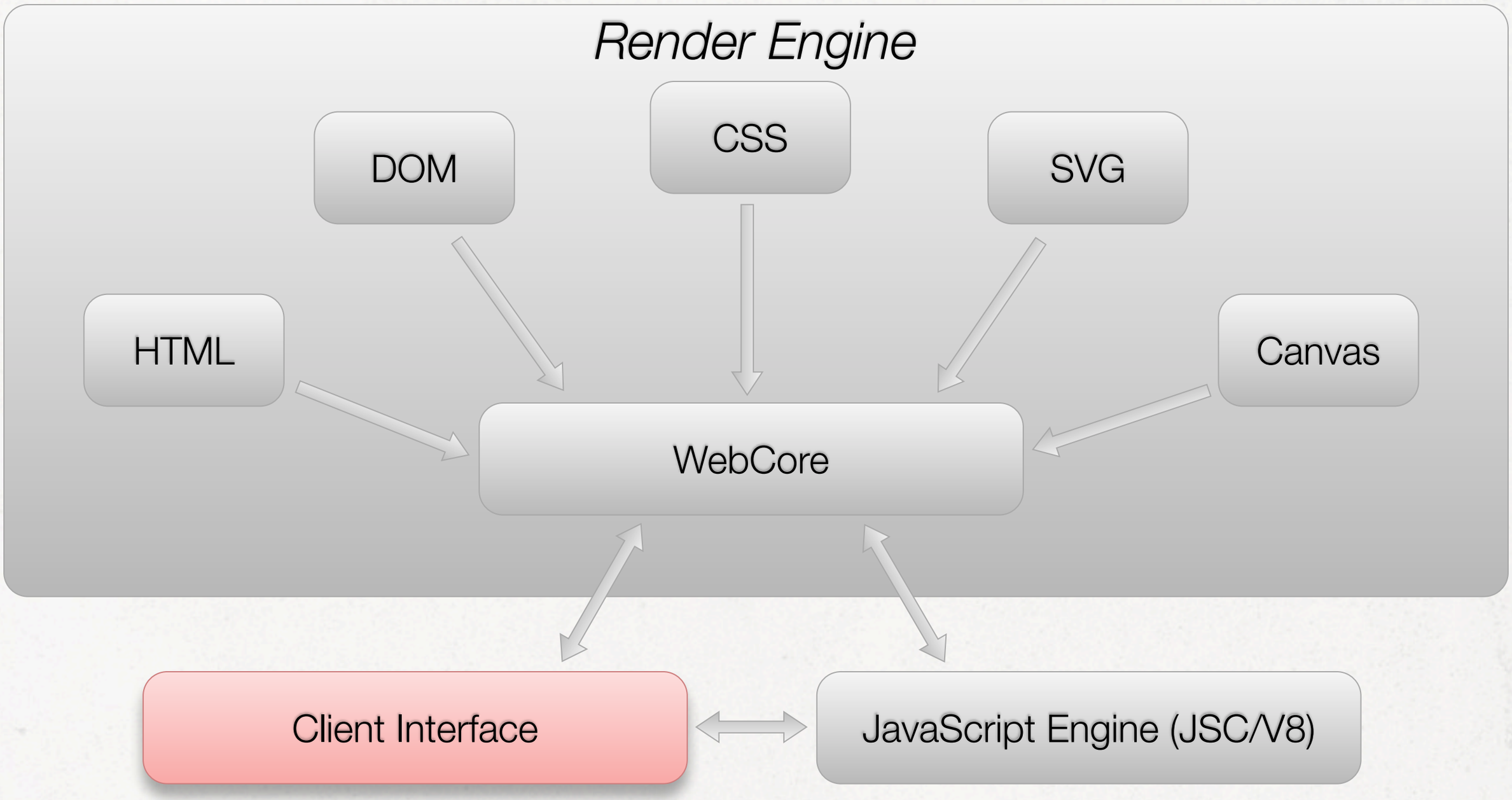
How?

Client Interface

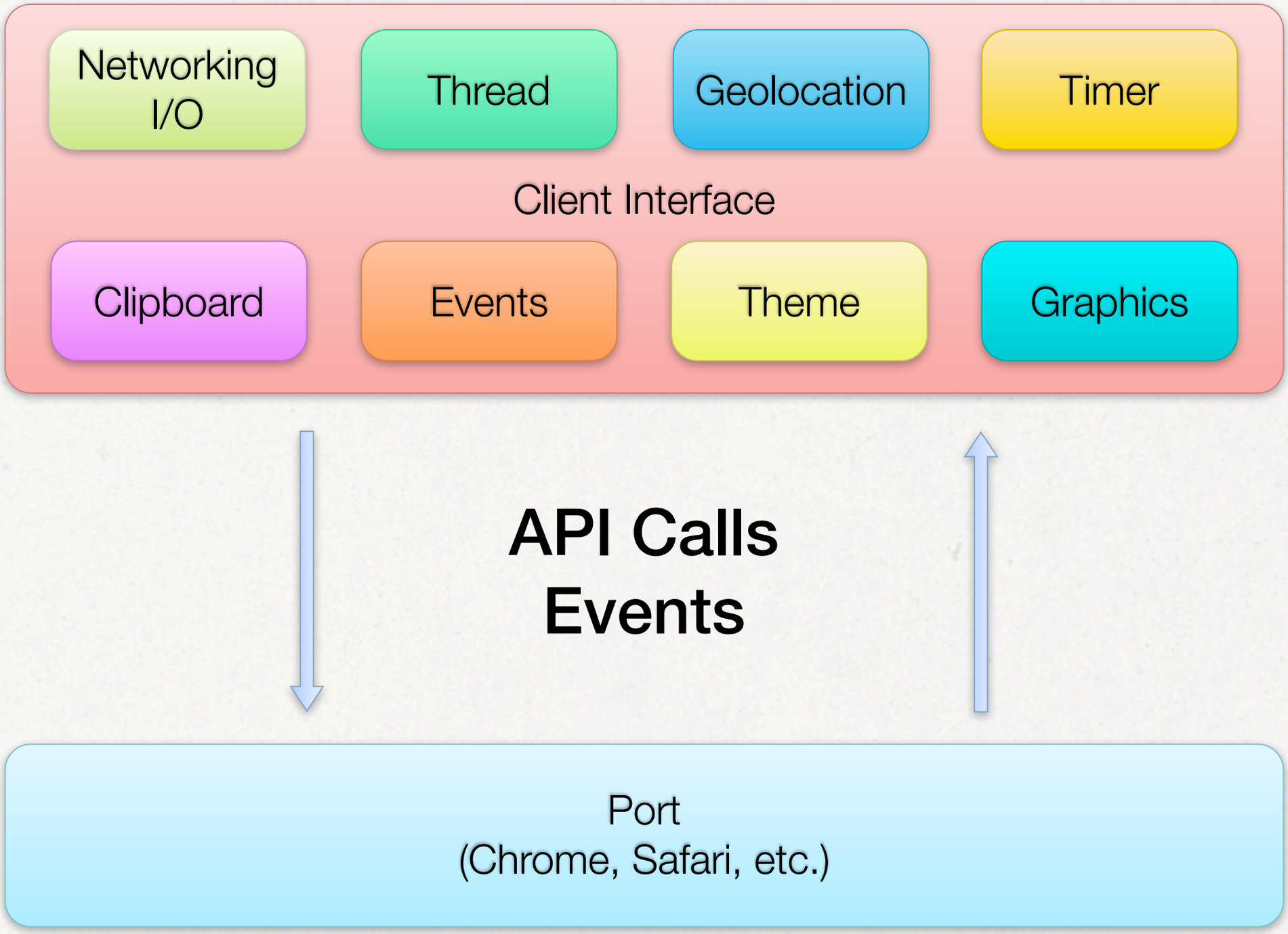
WebKit Components



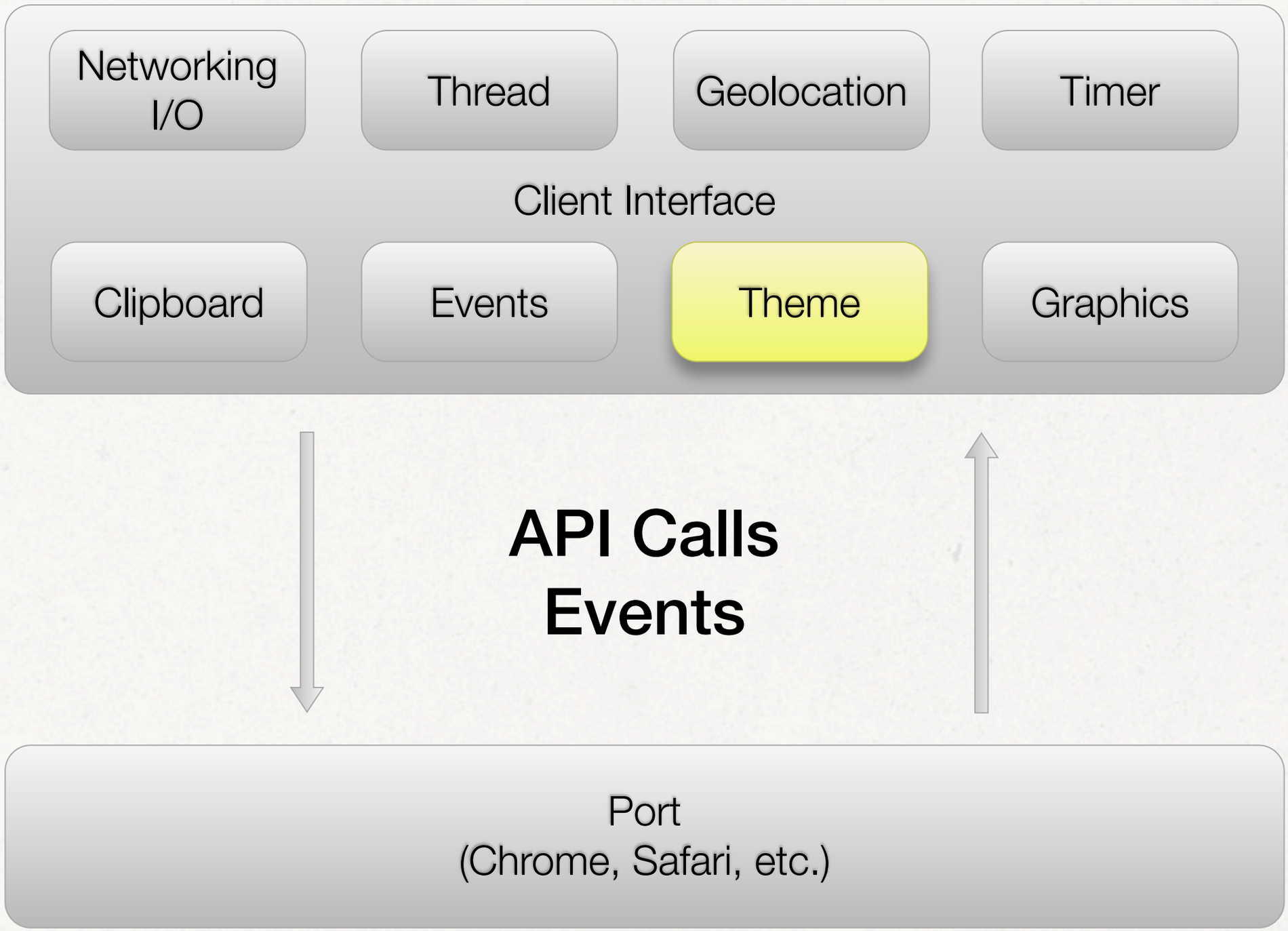
WebKit Components



Port Abstraction



Port Abstraction



<input type=”number” />

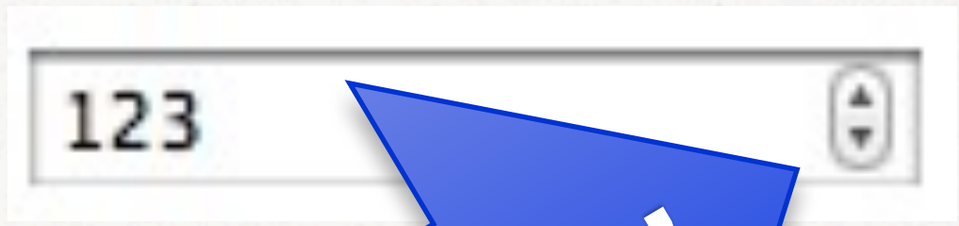
WebKit

Port

<input type=”number” />

WebKit

<input type=”number” />



lookin’ sexy!

Piece of cake!

Port

<input type=”number” />

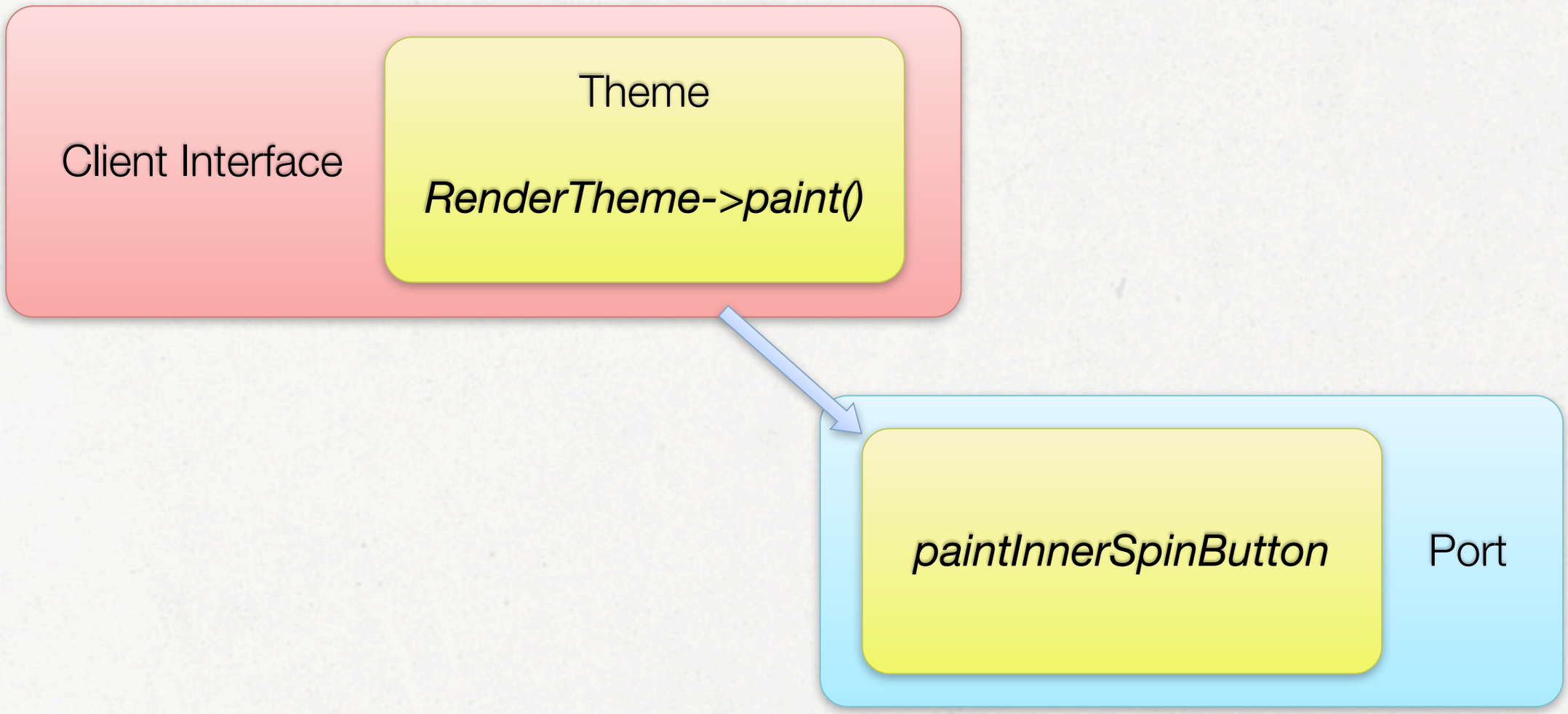
Theme Interface

```
bool paintInnerSpinButton(RenderObject*, const
                          PaintInfo&, const IntRect&);
```

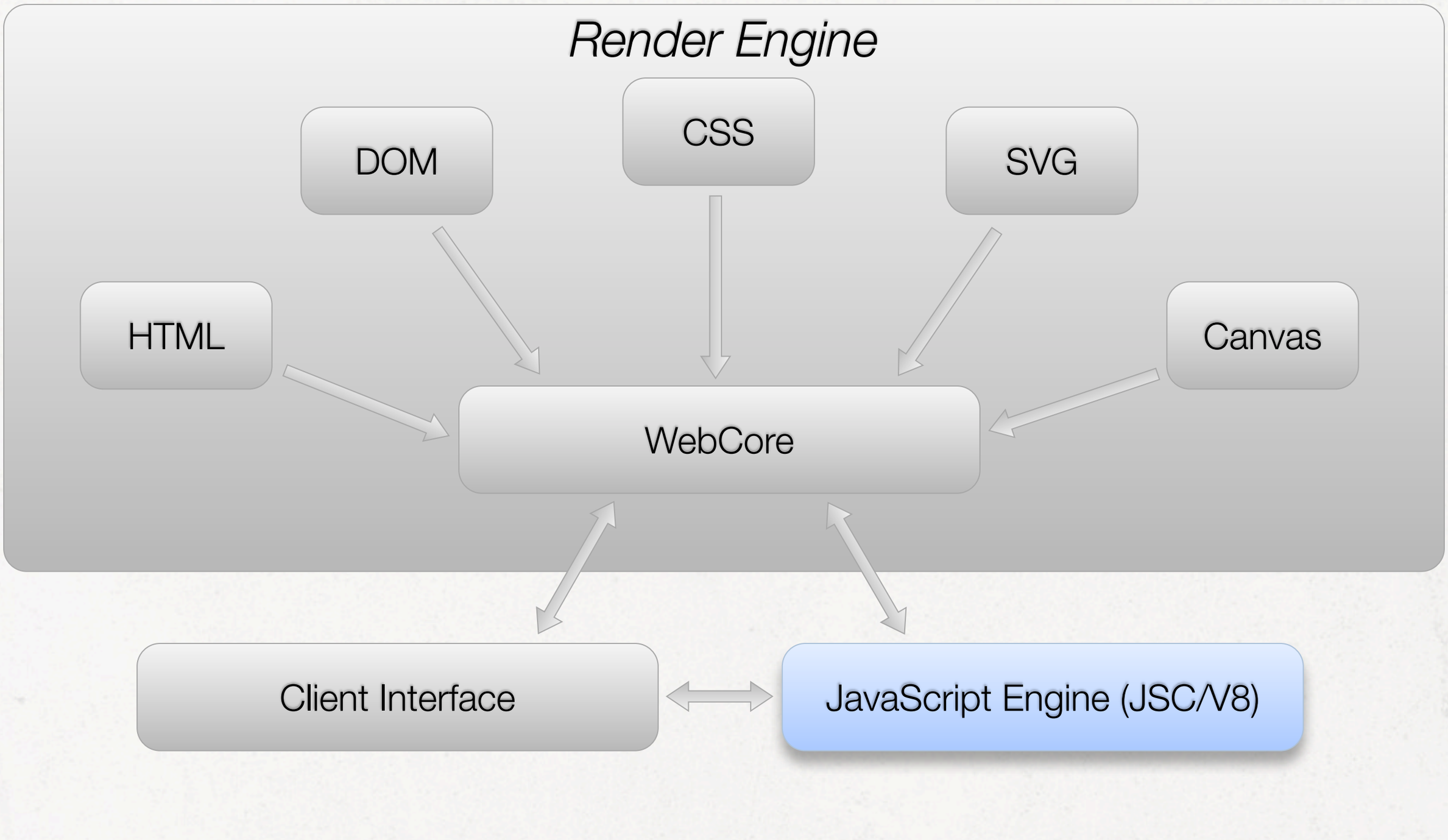
<input type=”number” />

Theme Interface

```
bool paintInnerSpinButton(RenderObject*, const
                          PaintInfo&, const IntRect&);
```

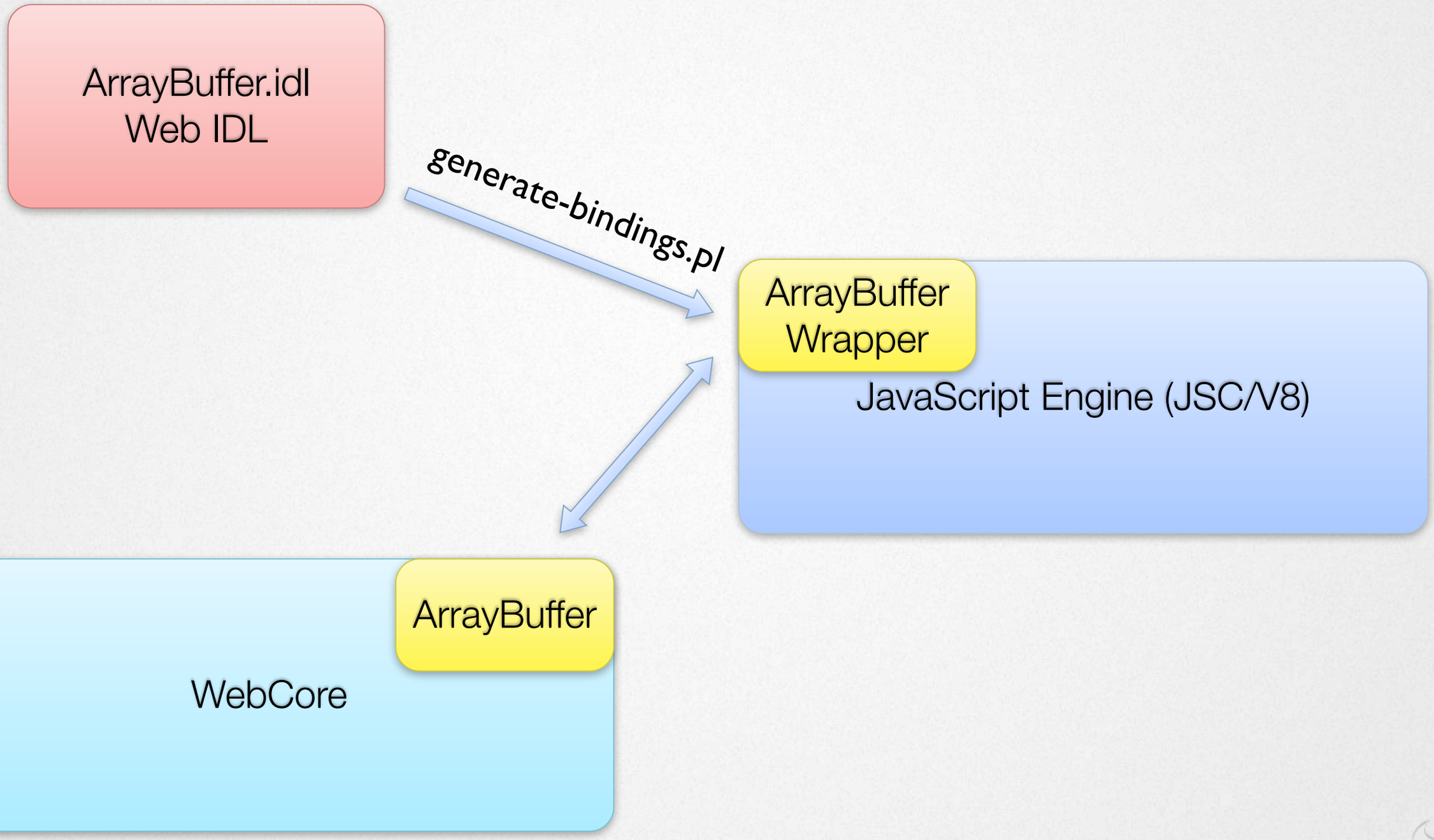


WebKit Components



JavaScript Bindings

JavaScript Bindings



ArrayBuffer.h

```
class ArrayBuffer : public RefCounted<ArrayBuffer> {
public:
    // ...
    inline unsigned byteLength() const;

    inline PassRefPtr<ArrayBuffer> slice(int begin, int end) const;
    inline PassRefPtr<ArrayBuffer> slice(int begin) const;
    // ...
};
```

Web[Kit]IDL

```
interface [
    JSGenerateIsReachable=Impl,
    CustomConstructor,
    ConstructorParameters=1,
    JSNoStaticTables
] ArrayBuffer {
    readonly attribute int byteLength;
    ArrayBuffer slice(in long begin, in [Optional] long end);
};
```

```
> perl generate-bindings.pl --generator JS ArrayBuffer.idl
> perl generate-bindings.pl --generator V8 ArrayBuffer.idl
```

JavaScriptCore

```
JSValue jsArrayBufferByteLength(ExecState* exec, JSValue slotBase, PropertyName)
{
    JSArrayBuffer* castedThis = jsCast<JSArrayBuffer*>(asObject(slotBase));
    UNUSED_PARAM(exec);
    ArrayBuffer* impl = static_cast<ArrayBuffer*>(castedThis->impl());
    JSValue result = jsNumber(impl->byteLength());
    return result;
}
```

V8

```
static v8::Handle<v8::Value> byteLengthAttrGetter(v8::Local<v8::String> name,
    const v8::AccessorInfo& info)
{
    INC_STATS("DOM.ArrayBuffer.byteLength._get");
    ArrayBuffer* imp = V8ArrayBuffer::toNative(info.Holder());
    return v8::Integer::New(imp->byteLength());
}
```

Hacking WebKit

For Fun & Profit

Building WebKit Nightly

Prerequisites

Prerequisites

Mac

Xcode 3.1.4+ & command-line tools

> which g++ => /usr/bin/g++

Windows

Don’t even bother (see “Building Chromium” :)

...but if you’re really brave: http://www.webkit.org/building/tools.html

Get the Source

Get the Source

SVN

svn checkout http://svn.webkit.org/repository/webkit/trunk WebKit

Git (RECOMMENDED!!!)

git clone git://git.webkit.org/WebKit.git

or

git clone git://github.com/WebKit/webkit.git WebKit

Build WebKit

Build WebKit

> cd WebKit
> Tools/Scripts/build-webkit

Go have lunch...

> Tools/Scripts/run-safari

Build WebKit

> cd WebKit
> Tools/Scripts/build-webkit

Go have lunch...

runs local safari, but w/ new WebKit framework

> Tools/Scripts/run-safari

Building Chromium

Prerequisites

Prerequisites

Depot Tools: http://www.chromium.org/developers/how-tos/install-depot-tools

Mac

Xcode 4.3+ & command-line tools

> which clang => /usr/bin/clang

Windows

Visual Studio (Express)

Windows SDK

DirectX SDK

Windows Driver Development Kit

Details: http://www.chromium.org/developers/how-tos/build-instructions-windows#TOC-Prerequisite-software

Get the Source

Get the Source

```
> mkdir chromium && cd chromium
> gclient config http://src.chromium.org/chrome/trunk/src
> gclient sync
```

Details: http://dev.chromium.org/developers/how-tos/get-the-code

Build Chromium

Build Chromium

```
> cd chromium/src
> GYP_GENERATORS=ninja gclient runhooks --force
> ninja -C out/Release chrome
```

Go on vacation...

```
> open out/Release/Chromium.app
```

Build Chromium

> cd chromium/src
> GYP_GENERATORS=ninja gclient runhooks --force
> ninja -C out/Release chrome

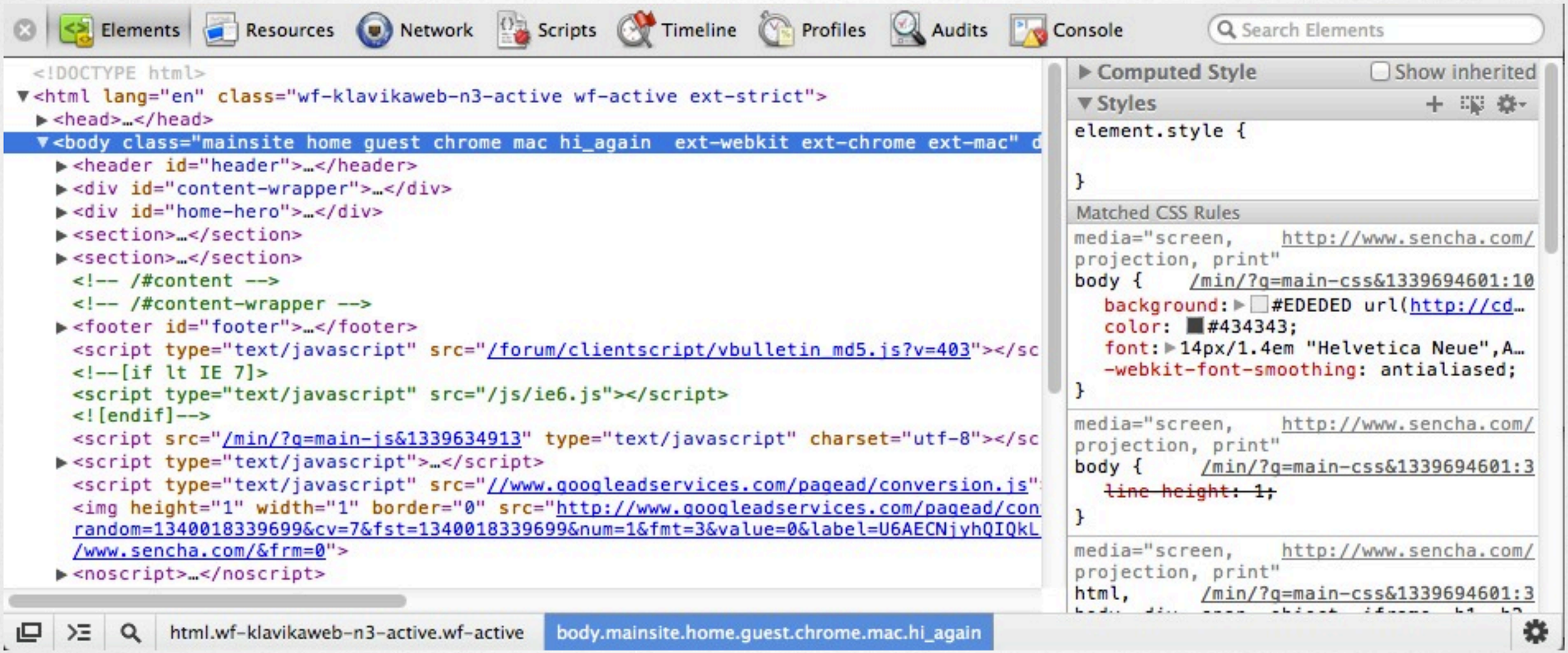
super fast "make" equivalent
comes with depot tools

Go on vacation...

> open out/Release/Chromium.app

Let’s Hack

Web Inspector



Add in Hacks

inspector.js

```
var re = /command=([^&]+)/,
    match = re.exec(location.href),
    command;

if (!match) {
    return;
}

try {
    command = eval('(' + decodeURIComponent(match[1]) + ')');
} catch (e) {
    return;
}

switch (command.type) {
    default:
        break;
}
```

command handler

Add in Hacks

inspector.js

persist data

```
var saveData = function(filePath, data) {
    filePath = '/Users/jarred/qconny/' + filePath;

    var xhr = new XMLHttpRequest();
    xhr.onload = function() {
        alert('Data saved successfully to ' + filePath);
    };
    xhr.onerror = function() {
        alert('Data failed to save to ' + filePath);
    };
    xhr.open('GET', '/savedata');
    xhr.setRequestHeader('DevTools-Data', JSON.stringify(data));
    xhr.setRequestHeader('DevTools-FilePath', filePath);
    xhr.send(null);
};
```

Add in Hacks

devtools_http_handler_impl.cc

```
void DevToolsHttpHandlerImpl::OnHttpRequest(
    int connection_id,
    const net::HttpServerRequestInfo& info) {
  // ...

  if (info.path.find("/savedata") == 0) {
    BrowserThread::PostTask(
        BrowserThread::FILE,
        FROM_HERE,
        base::Bind(&DevToolsHttpHandlerImpl::OnSaveDataRequestFILE,
                   this,
                   connection_id,
                   info));
    return;
  }
  // ...
}
```

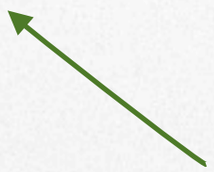
persist data

Add in Hacks

devtools_http_handler_impl.cc

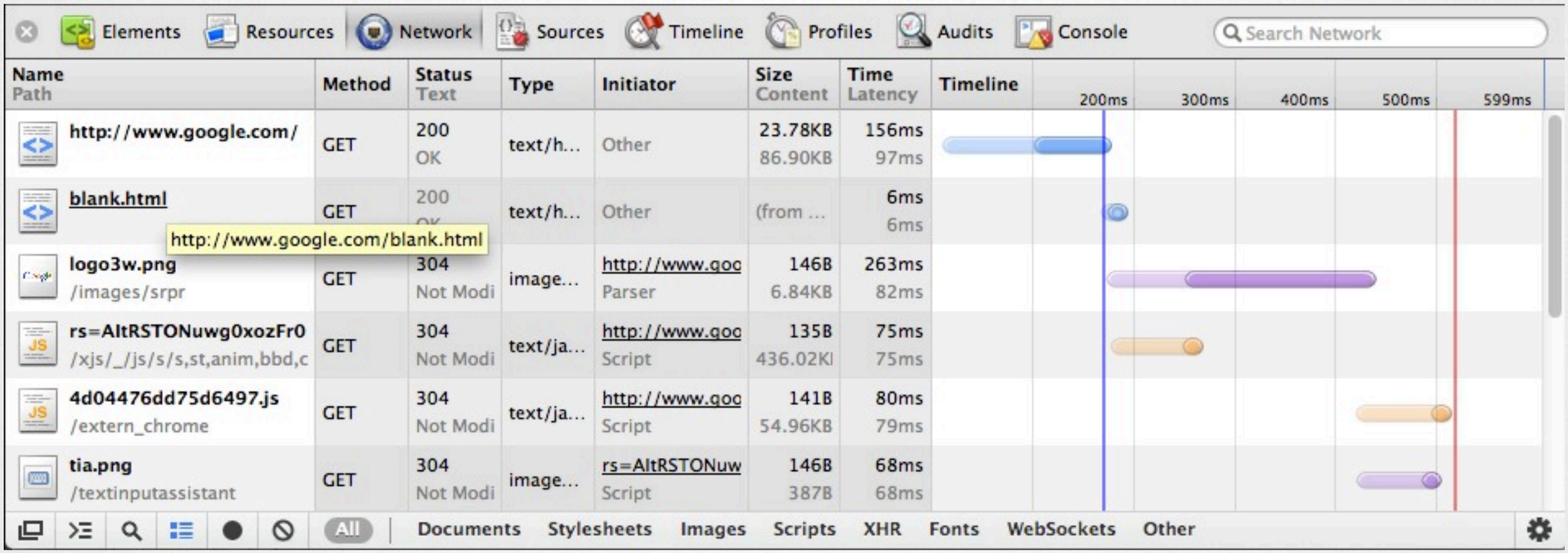
```
void DevToolsHttpHandlerImpl::OnSaveDataRequestFILE(int connection_id, const
net::HttpServerRequestInfo& info) {
  bool success = false;
  FILE* fp = fopen(info.GetHeaderValue("DevTools-FilePath").c_str(), "wb");
  if (fp) {
    std::string data = info.GetHeaderValue("DevTools-Data");
    fwrite(data.data(), 1, data.size(), fp);
    fclose(fp);
    success = true;
  }

  Send200(connection_id, success ? "true" : "false", "application/json; charset=UTF-8");
}
```

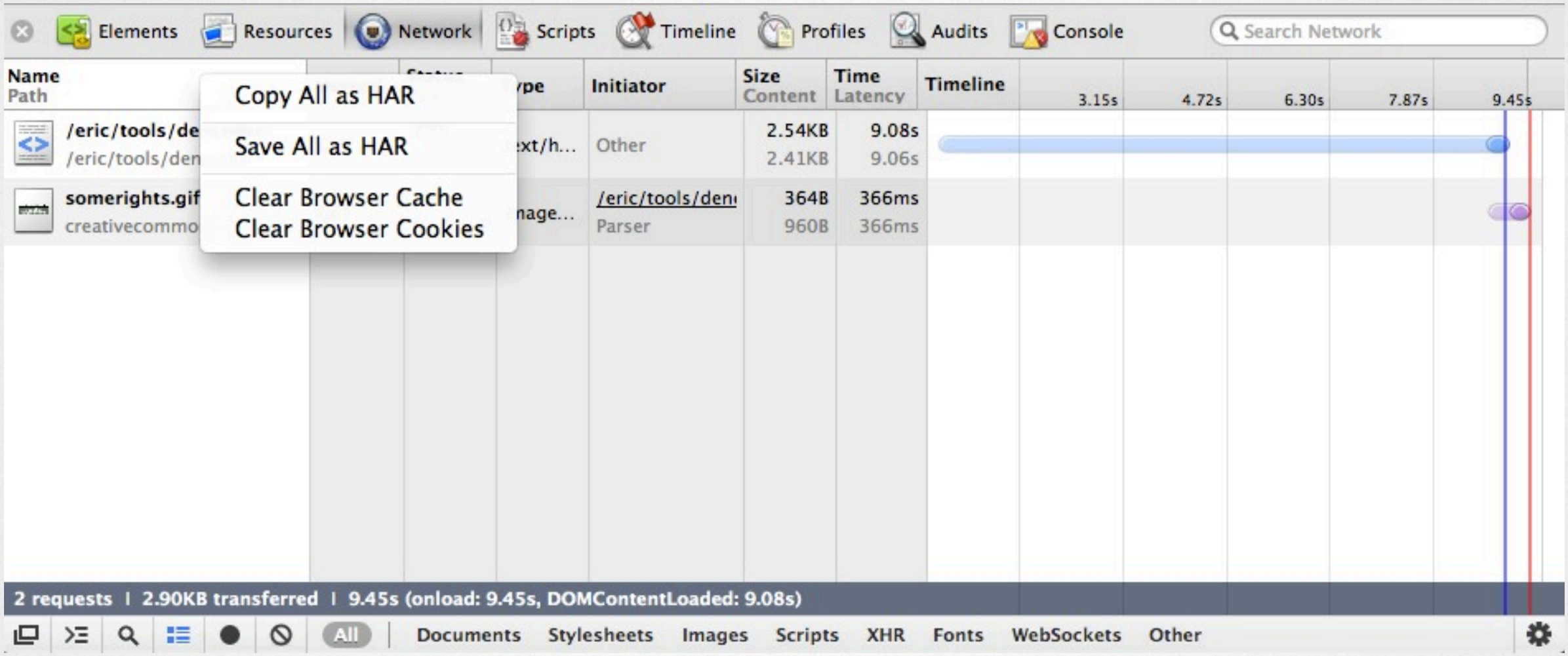


persist data

Web Inspector
Network



Export Network HAR Data



More info: http://www.softwareishard.com/blog/har-12-spec/

Add in Hacks

inspector.js

exportHAR cmd

```
var exportHAR = function(cmd) {
    // Force a reload of the page to grab network stats.
    PageAgent.reload(true /* bypass cache */);

    setTimeout(function() {
        var networkData =
WebInspector.panels.network._networkLogView._getExportData();
        saveData((cmd.filename || WebInspector.inspectedPageDomain)
            + '.har', networkData);
    }, cmd.timeout);
};

case 'exportHAR':
    exportHAR(command);
    break;
```

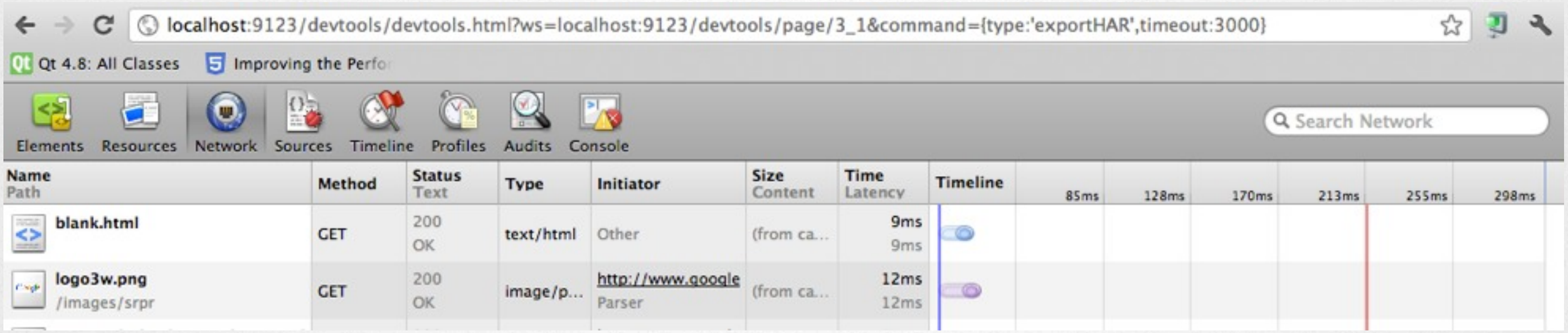
Add in Hacks

NetworkPanel.js

export data fn

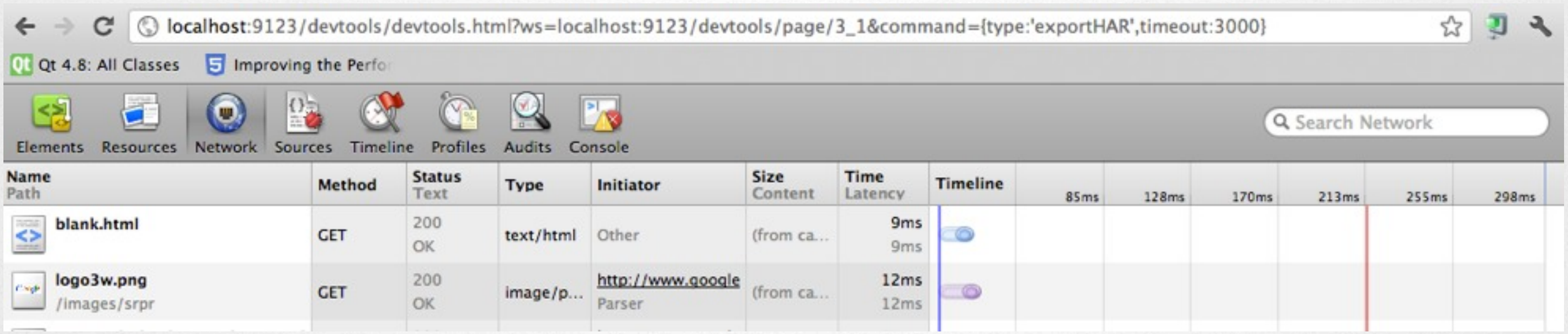
```
_getExportData: function() {
    return { log: (new WebInspector.HARLog(this._requests)).build() };
}
```

Trigger Hack



Trigger Hack

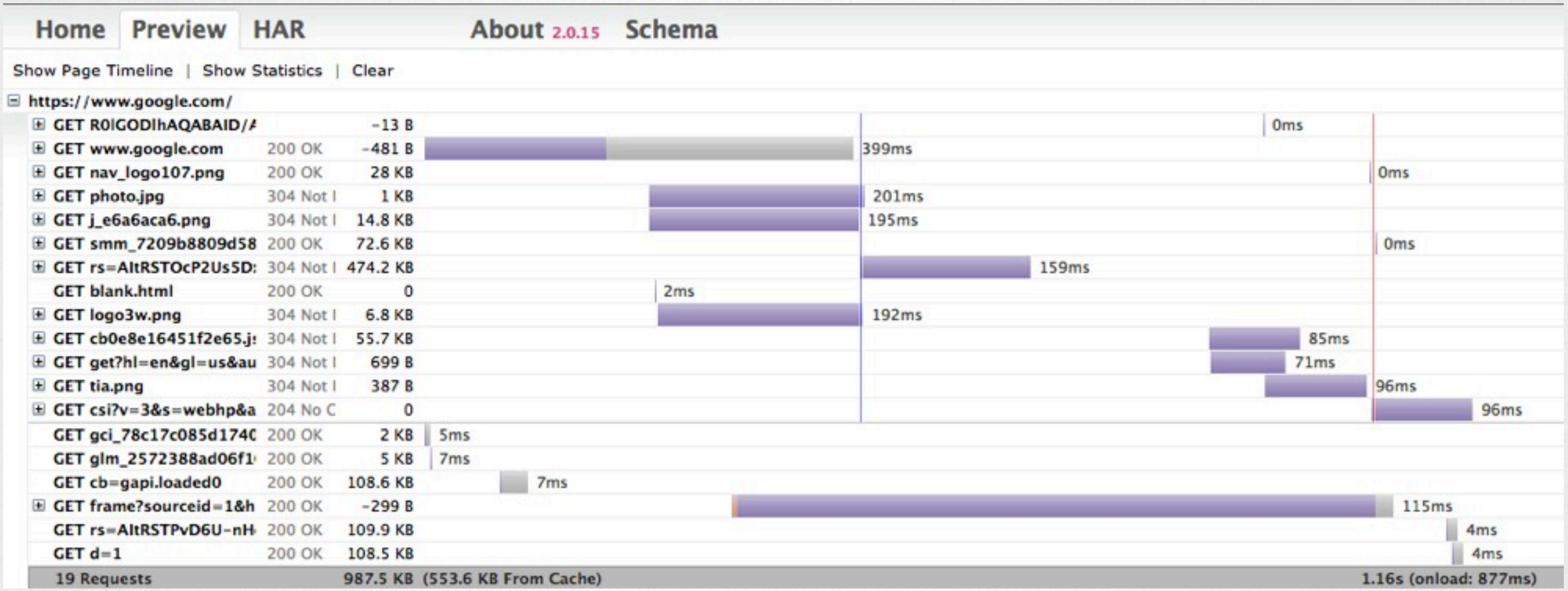
http://localhost:9123/devtools/devtools.html?ws=localhost:9123/devtools/page/3_1**&command={type:'exportHAR',timeout:3000}**



HAR Output


```
{"log":{"version":"1.2","creator":
{"name":"WebInspector","version":"536.5"},"pages":[],"entries":
[{"startedDateTime":"2012-06-19T18:02:16.852Z","time":8,"request":
{"method":"GET","url":"http://www.google.com/
blank.html","httpVersion":"HTTP/1.1","headers":[],"queryString":
[],"cookies":[],"headersSize":47,"bodySize":0},"response":
{"status":200,"statusText":"OK","httpVersion":"HTTP/1.1","headers":
[],"cookies":[],"content":{"size":0,"mimeType":"text/
html"},"redirectURL":"","headersSize":17,"bodySize":0},"cache":
{},"timings":{"blocked":
0,"dns":-1,"connect":-1,"send":-1,"wait":-1,"receive":0,"ssl":-1}},
```

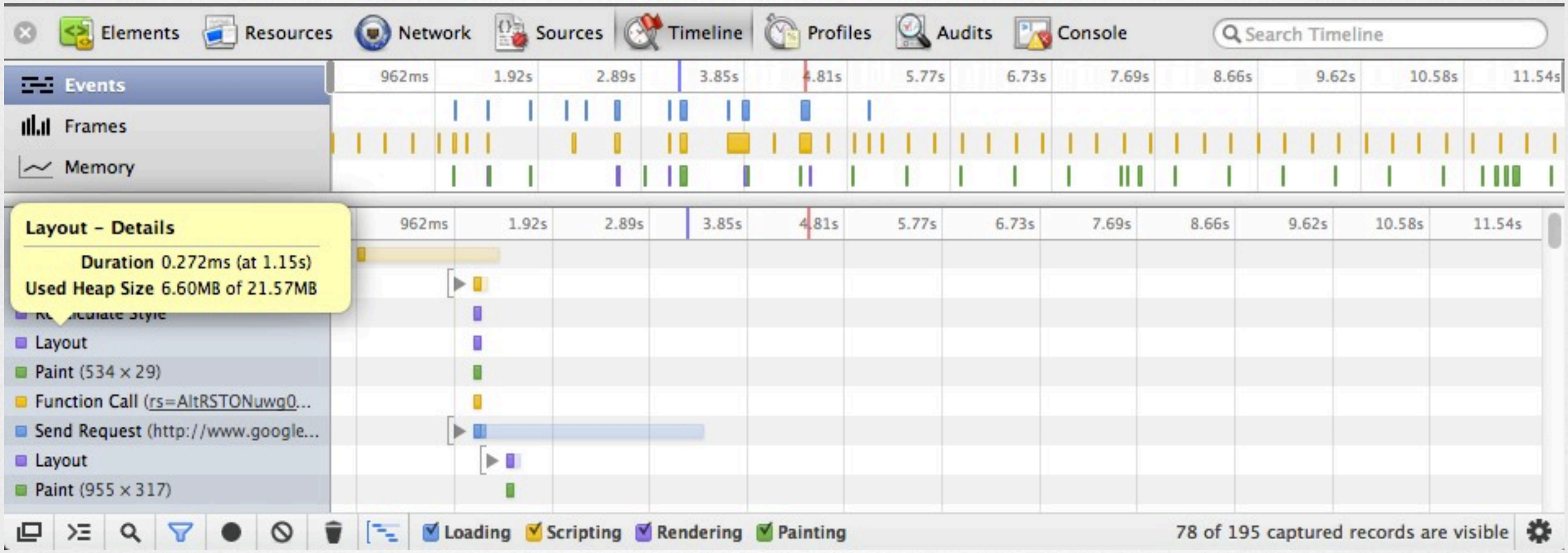
HAR Output



http://www.softwareishard.com/har/viewer/

Demo

Export Timeline Data



Add in Hacks

inspector.js

exportTimeline cmd

```
var exportTimeline = function(cmd) {
    setTimeout(function() {
        // Start the timeline profiler.
        WebInspector.panels.timeline._toggleTimelineButtonClicked();
        // Force a reload of the page to grab latest timeline stats.
        PageAgent.reload(true /* bypass cache */);

        setTimeout(function() {
            // Stop the timeline profiler.
            WebInspector.panels.timeline._toggleTimelineButtonClicked();

            var data = WebInspector.panels.timeline._model.getExportData();
            saveData(cmd.filename || (WebInspector.inspectedPageDomain + '-
timeline.json'), data);
        }, cmd.timeout);
    }, 1000);
};

case 'exportTimeline':
    exportTimeline(command);
    break;
```

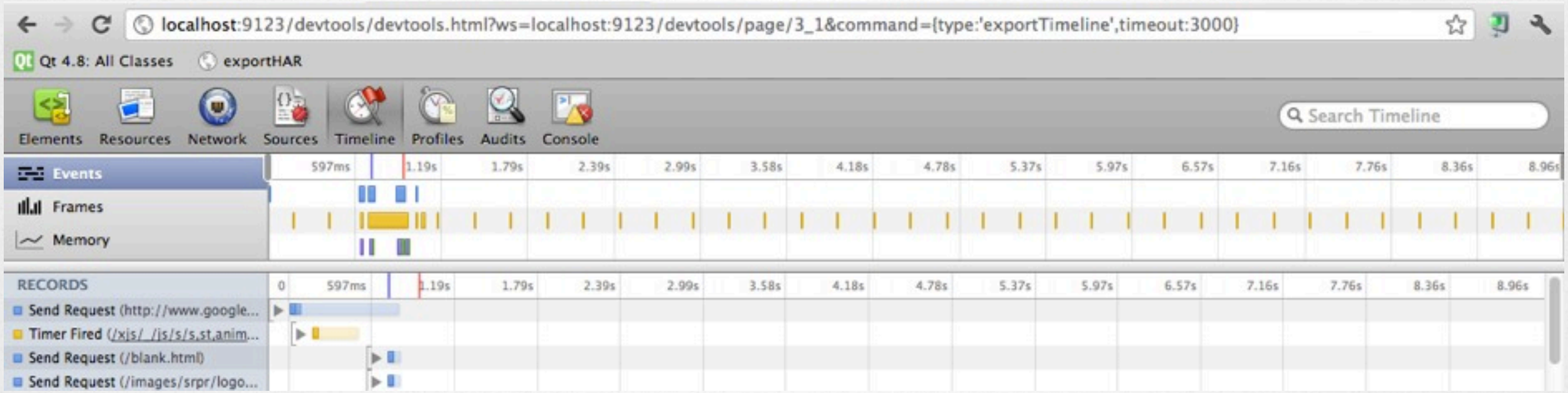
Add in Hacks

TimelineModel.js

export data fn

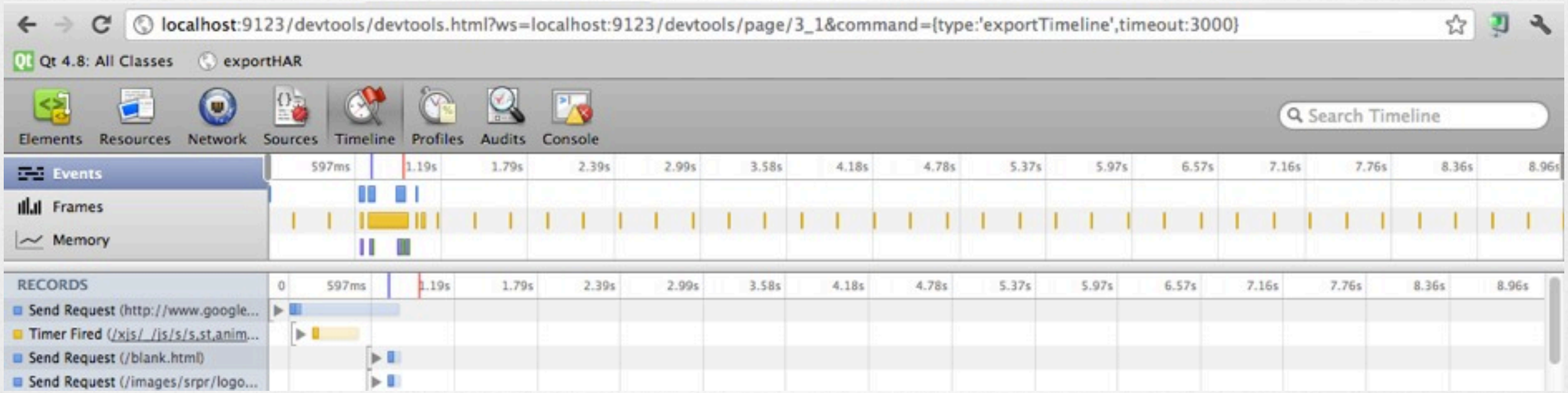
```
getExportData: function() {
    var records = [];
    for (var i = 0; i < this._records.length; i++) {
        records.push(this._records[i]);
    }
    return records;
}
```

Trigger Hack

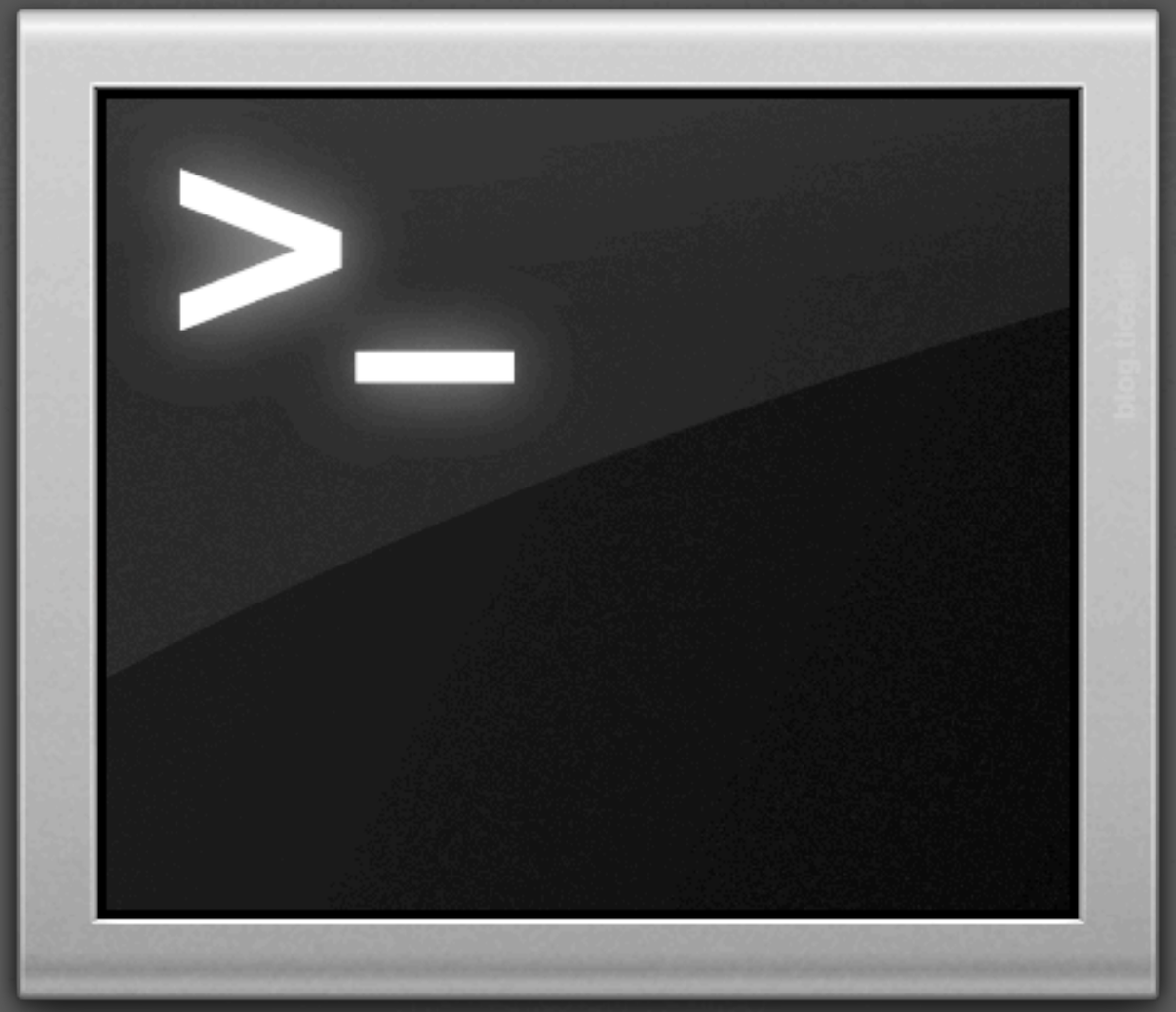


Trigger Hack

http://localhost:9123/devtools/devtools.html?ws=localhost:9123/devtools/page/3_1**&command={type:'exportTimeline',timeout:3000}**



Demo



Command Line WebKit

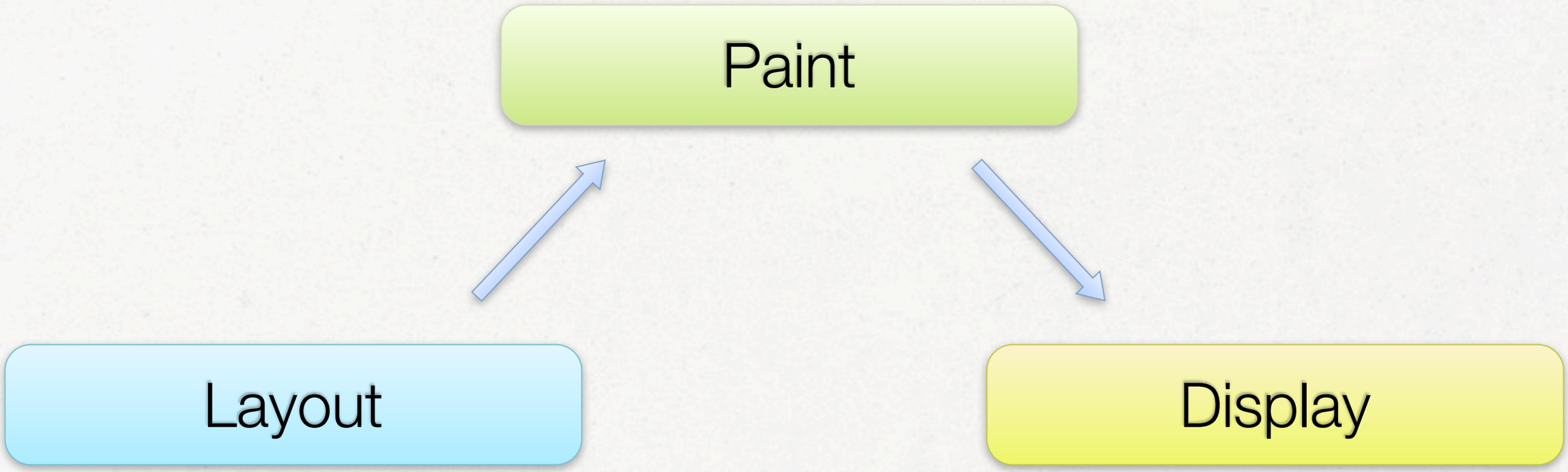
Demo

How’d you do that!?

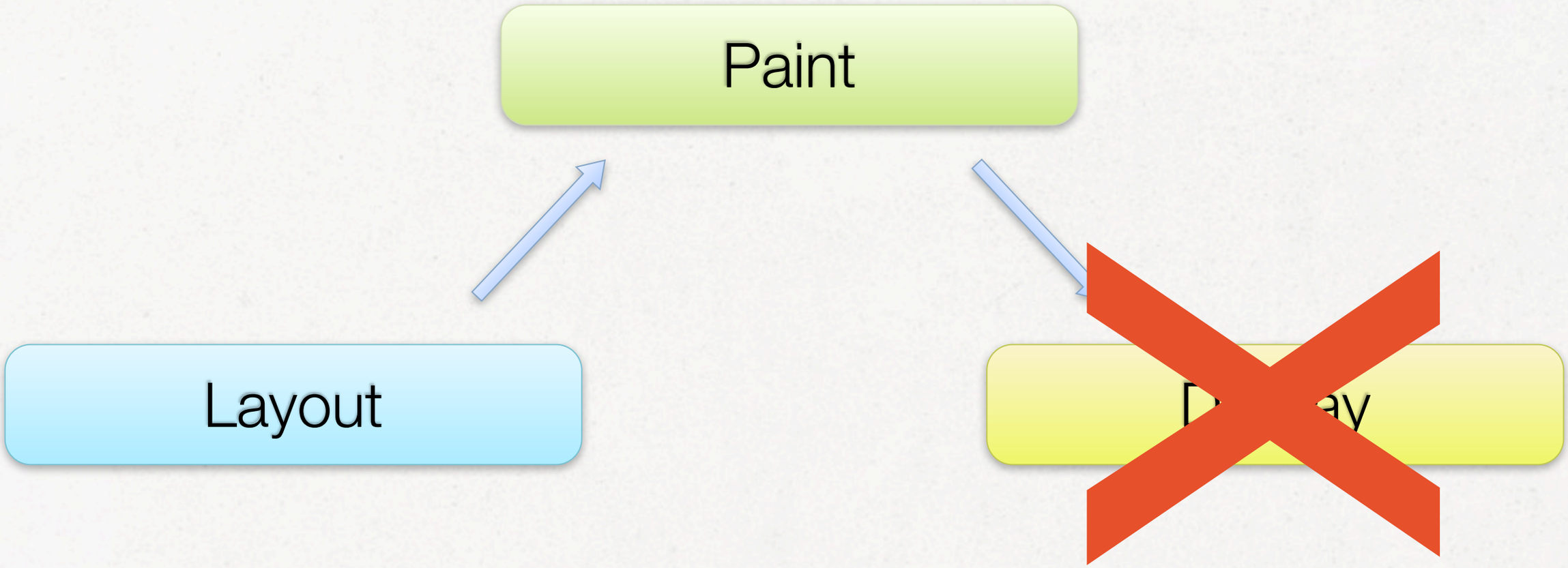
Headless WebKit

“Headless”

Normal Browser



Headless



What’s in it for me?

What's in it for me?

Automated "smoke" tests

Continuous integration

Pixel-perfect regression tests

Web scraping

...

PhantomJS

Headless WebKit

Based on QtWebKit port

Full Web Stack

JS, DOM, CSS3, Canvas, SVG, etc.

Command Line Interface

JavaScript APIs

License: BSD / Open Source

http://www.phantomjs.org

Demo

Wrap Up

Hacking WebKit

Automation

- Page scraping
- Behavior/input emulation
- Web Inspector hijacking

Quality Assurance / Continuous Integration

- Smoke tests
- Visual/pixel regression tests
- Unit tests
- Performance or Load regression tests

Jarred Nicholls

@jarrednicholls

Thanks!