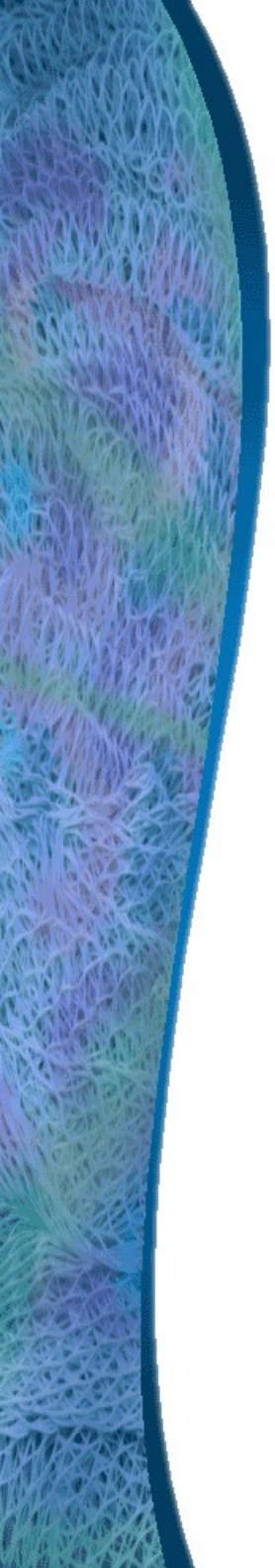


Validating Javascript from Realworld Websites in a Browser-like Environment

Sander Sõnajalg

JavaScript (ECMAScript)

- Object-oriented / functional scripting language
- Dynamic typing
- Prototype inheritance
- Objects are dictionaries of properties
- Standardized by ECMA, spec. 262.

JS in the Web - Example

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Tra..
<html xmlns="http://www.w3.org/1999/xhtml" xml:l..
<head>
<meta http-equiv="content-type" content="text/h..
<title>Images for David Bowie</title>

<script type="text/javascript"
  src="/js/global-v77.js" />

<script type="text/javascript">
DISCOGS.set("Session", {loggedin: false, userna..
DISCOGS.set('static_server', '');
</script>

</head>
```

LINKED

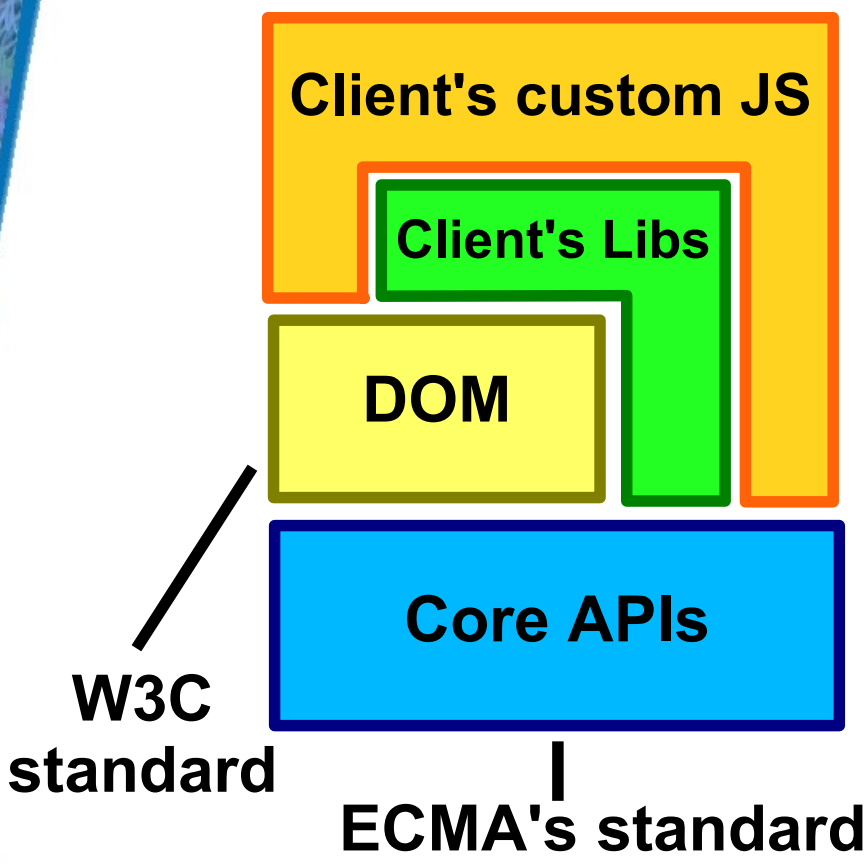
EMBEDDED

JavaScript in the Web

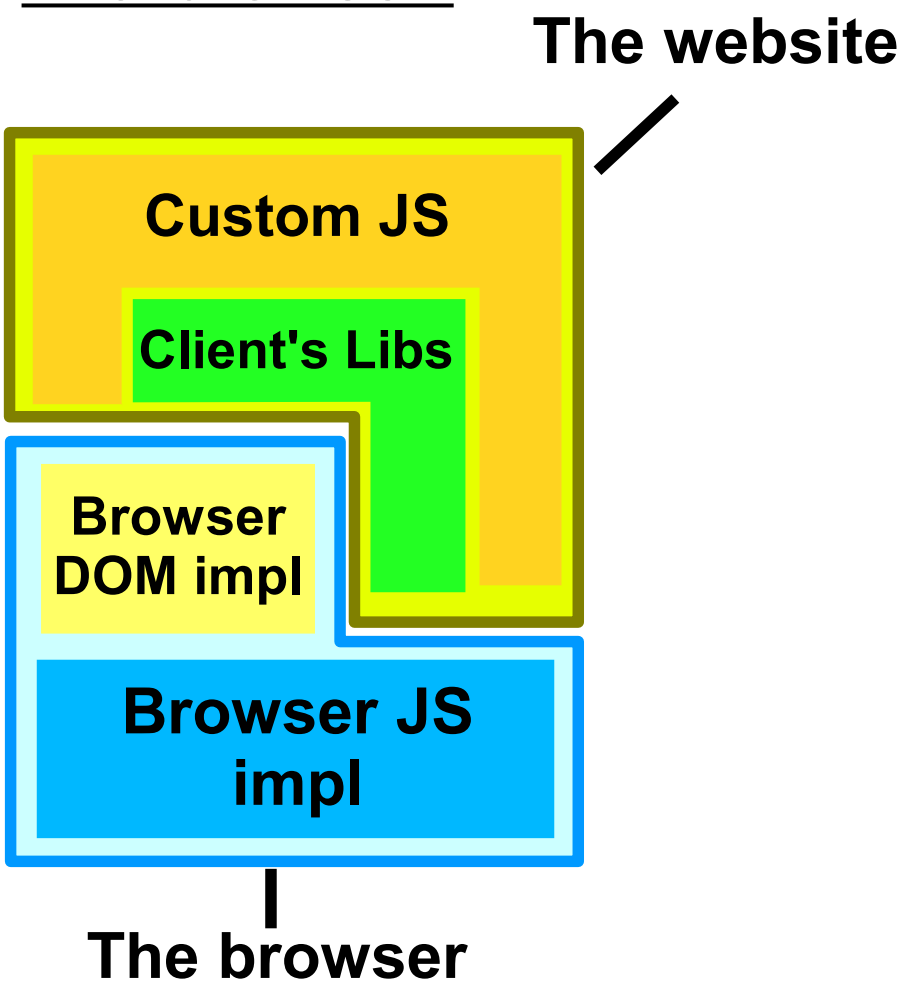
- Every web page is a JavaScript “program” of:
  - JS core functions (ECMA standard)
  - Browser's DOM implementation
  - All linked and embedded snippets of JavaScript
- Browsers use different JS engines
- .. and different DOM implementations

Browser Environment

General model:



In a browser:



The Goals

(1) Statically validate web site JavaScript's syntax

(2) Simulate the execution inside a browser, detect runtime errors

(3) Report usage of properties/functions that might not work in *some* browsers

Towards Goal 1: syntax errors

- Basically equal to parsing
- Implement myself?
- Reuse something available?
  - Some browser's JS engine?
  - Which one? Google's V8? Mozilla's SpiderMonkey?

Rhino

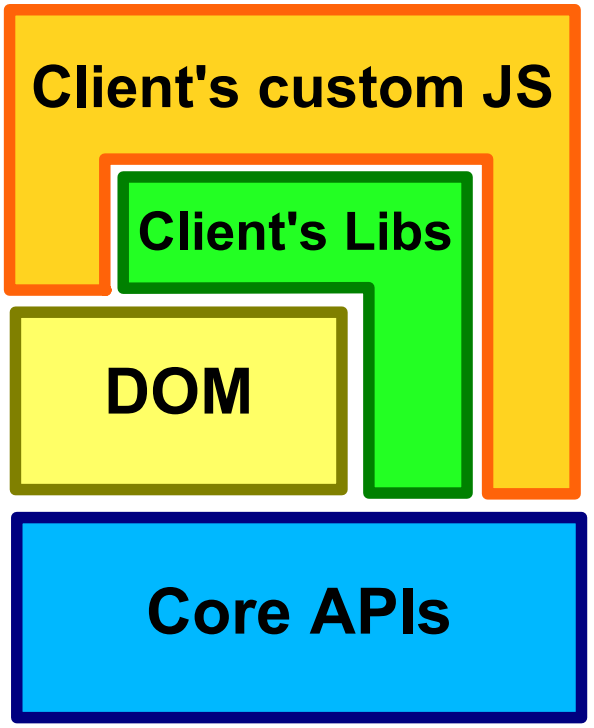
- Mozilla's alternative JavaScript engine
- Open-source
- Parser detects syntactical mistakes

GOAL 1 ✓

- Evaluation detects some API-level mistakes
- Tests for language-extensions can be implemented

Simulated browser environment

General model:



Simulated environment:

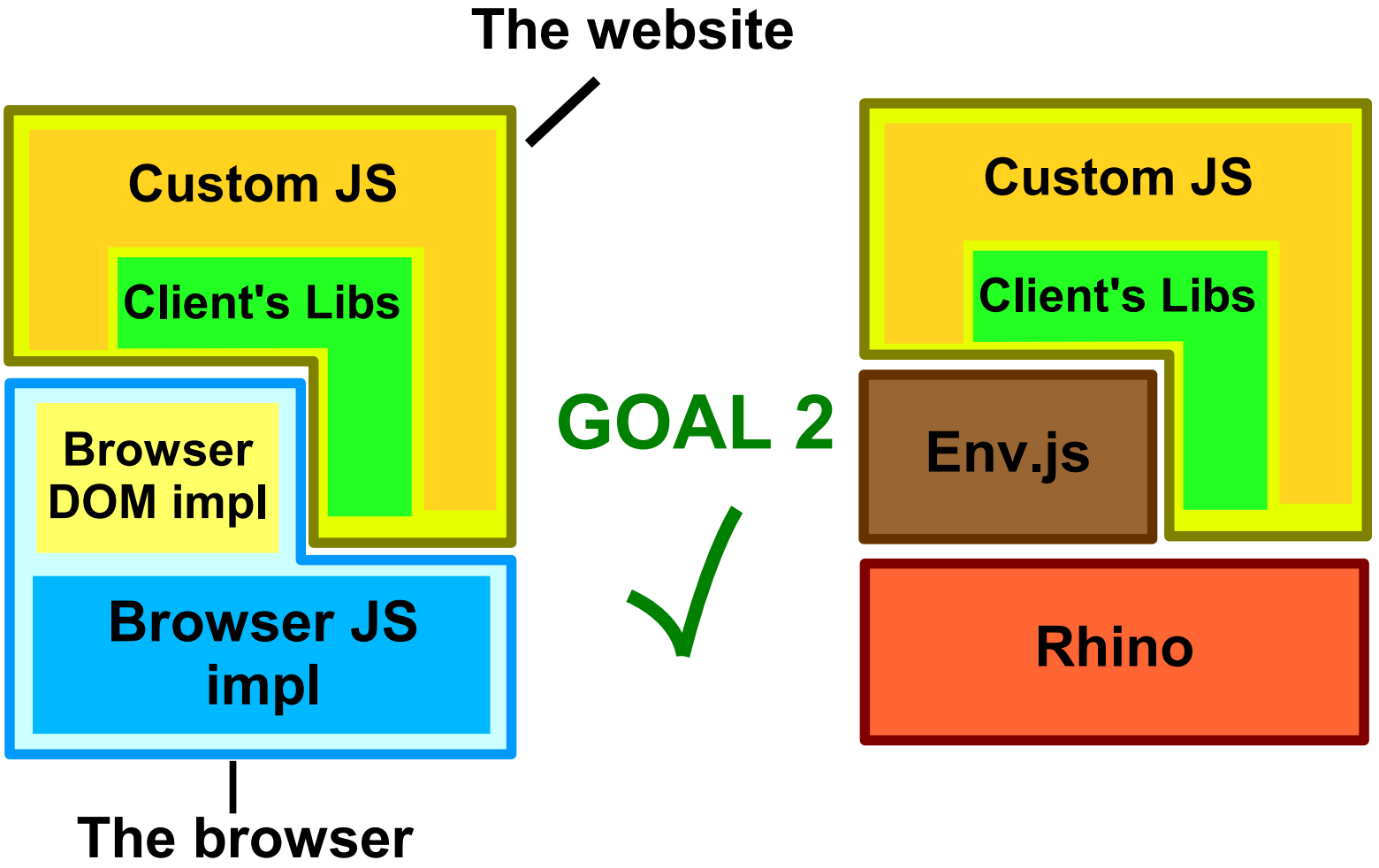
Env.js

- A JavaScript implementation of DOM
- Can be loaded with Rhino

Simulated Browser Environment

In a browser:

Simulated environment:



Browser-Specific Code

```
// Firefox
if (window.addEventListener) {
  myObject.addEventListener("mouseover", myEventHandler,
                                                    false);
// IE
} else {
  myObject.attachEvent("onmouseover", myEventHandler);
}
```

- Quite impossible to distinguish real errors from false positives
- Mostly in JS libraries

Challenge 1: Eliminating libraries, generated code

- No point to validate them
- Contain lot of browser-specific code
- Various ways:
  - By names of linked scripts
  - By headers
  - If compressed
  - Case-by-case

Challenge 2: Unknown types

- Example: property **srcElement** of class **Event** is undefined in some browsers

```
// ======= e is an Event
var e = window.event

function MyElement(src) {
    this.srcElement = src;
}

// ======= f is something different
var f = new MyElement("foo", "bar");


var g = e.srcElement;

var h = f.srcElement;
```

NEED TO REPORT

FALSE POSITIVE IF REPORTED

Challenge 2: Unknown types State of the Art

- Type-inference on formalized subset of JS [Anderson et al]
- Abstract interpretation of JS programs [Jensen et al]

- *Can we do it in a simpler way?*

Challenge 2: Unknown types Suggested Approach

Rewrite this ...

```
var g = e.srcElement;
var h = f.srcElement;
```

GOAL 3

√ ??

.. to this:

```
if (e instanceof Event) {
  _report_call$Event$srcElement("myScript.js", 233);
}
var g = e.srcElement;

if (f instanceof Event) {
  _report_call$Event$srcElement("myScript.js", 234);
}
var h = f.srcElement;
```

Demo output

Demo Output (1)

```
==========================================================
***    VALIDATION ERRORS FOR SCRIPT : [embedded javascript, 25 lines]
==========================================================

[ERROR] uncaught JavaScript runtime exception: TypeError: Cannot call method
"substring" of undefined (around line 2)
[WARNING] Reference to undefined property "language" (around line 11)
[WARNING] Reference to undefined property "browserLanguage" (around line 11)
[WARNING] Reference to undefined property "characterSet" (around line 11)
[WARNING] Reference to undefined property "charset" (around line 11)
[WARNING] Reference to undefined property "_domain" (around line 5754)
[WARNING] Reference to undefined property "domain" (around line 5754)
```

Demo output (2)

S U M M A R Y
==========

Script	Warnings	Errors	Runtime errors	Browser-specific
fEpost.js	0	0	0	0
embedded script 'var bannersFor	0	1	0	0
js.js	0	0	0	0
embedded script 'var gaJsHost =	0	1	0	0
embedded script 'var pageTracke	6	1	0	0