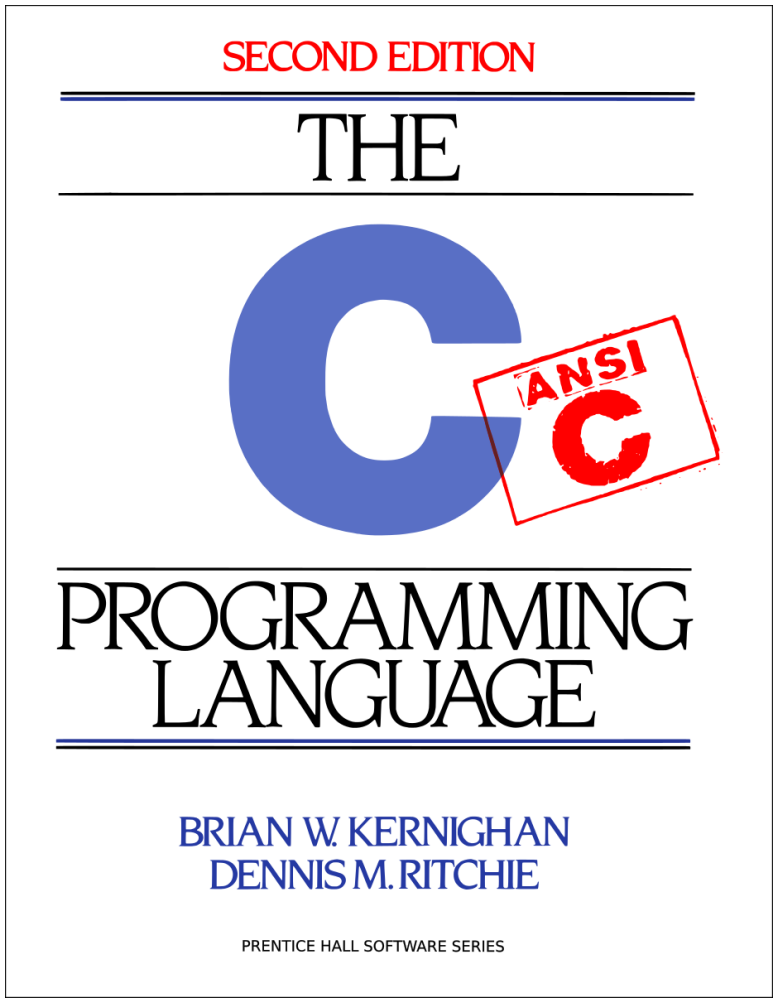
Analyzing JavaScript Web Applications in the Wild (Mostly) Statically

Sukyoung Ryu

Programming Language Research Group
KAIST

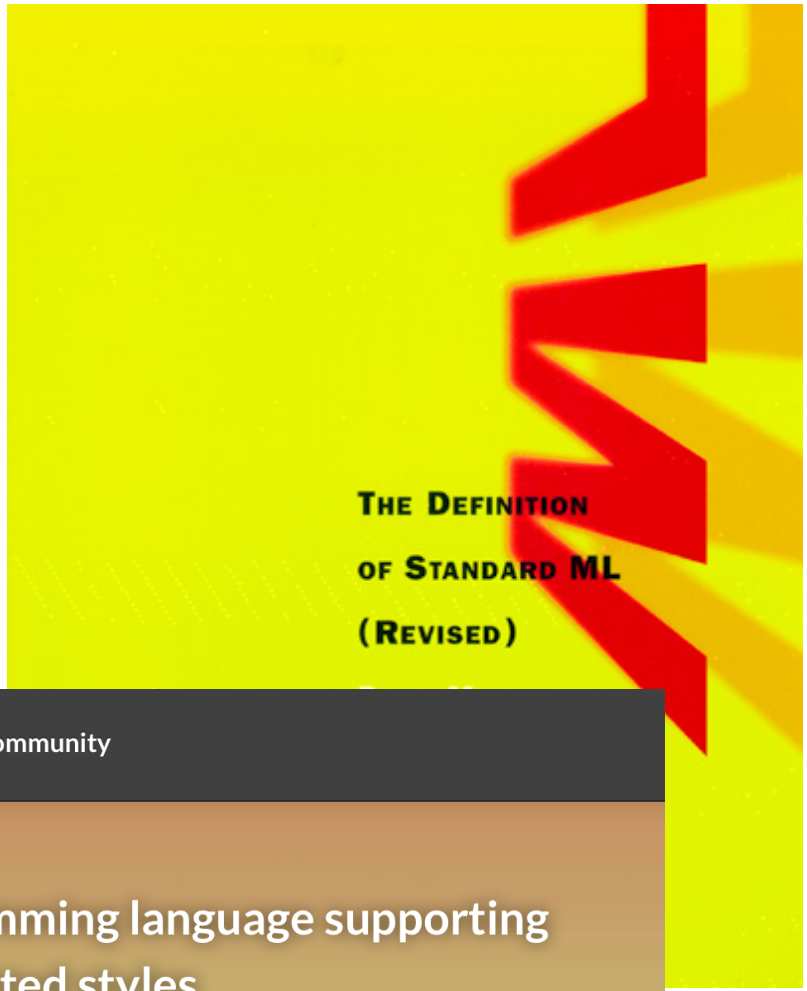
September 22, 2015

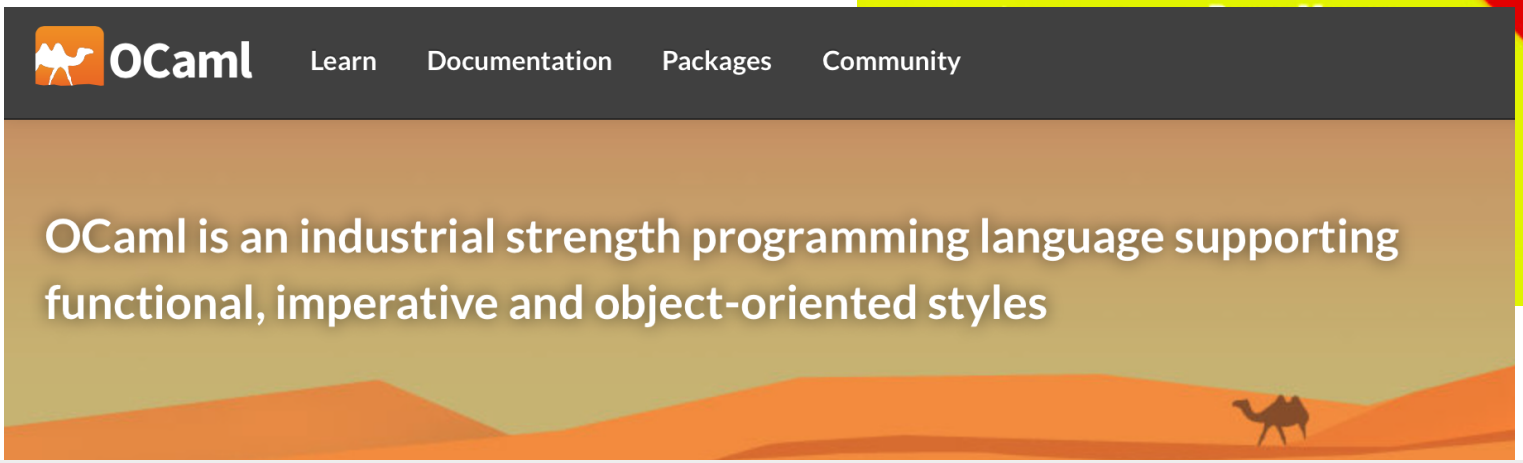
My First Programming Language: C



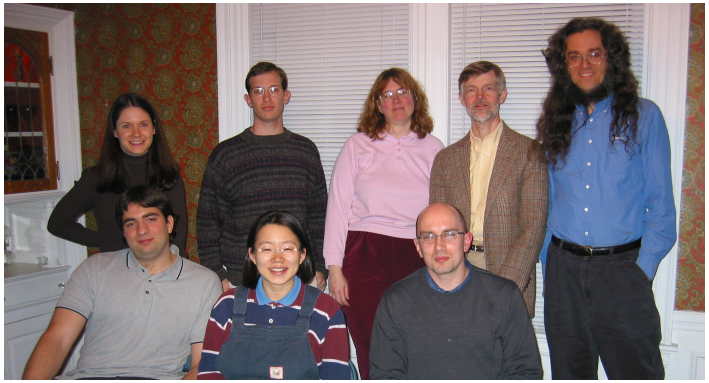
In Graduate School: SML

In Graduate School: OCaml





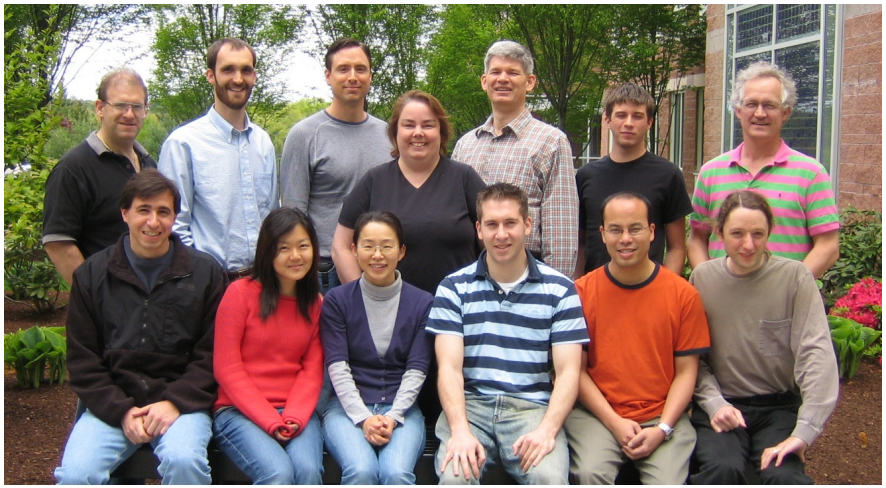
At Harvard: C



Debugging Everywhere

The goal of the *Debugging Everywhere* project is to make debugging a cheap, ubiquitous service. We intend to begin by getting compilers to emit *Active Debugging Information*, which we expect will support multi-language, multi-platform debugging much more readily than older approaches like Dwarf or dbx ``stabs."

At Sun Microsystems: Java/Scala



sources / ProjectFortress / src / com / sun / fortress / parser

Filename	Author	Revision	Modified	Log Entry
..				
Compilation.rats	chf	4575	over 4 years ago	Updated Copyright notices
Declaration.rats	chf	4575	over 4 years ago	Updated Copyright notices
DelimitedExpr.rats	sukyoungryu	4921	about 4 years ago	Revising
Expression.rats	Guy Steele	5051	almost 4 years ago	Completely redid assignment and ...
Fortress.rats	chf	4575	over 4 years ago	Updated Copyright notices

At Sun Microsystems: Java/Scala

```
protected def pExcInner(x: Type, y: Type)
                       (implicit negate: Boolean, history: Set[hType]): CFormula =
  (removeSelf(x), removeSelf(y)) match {
    case (s: BottomType, _) => pTrue()
    case (_, t: BottomType) => pTrue()
    case (s: AnyType, t) => pSub(t, BOTTOM)
    case (s, t: AnyType) => pSub(s, BOTTOM)
    case (s@SIntersectionType(_, elts), t) =>
      pOr(pSub(s, BOTTOM), pOr(elts.map(pExc(_, t))))
    case (s, t: IntersectionType) => exc(t, s)
    case (s@SUnionType(_, elts), t) =>
      pAnd(elts.map(pExc(_, t)))
    case (s, t: UnionType) => exc(t, s)
    case (i: _InferenceVarType, j: _InferenceVarType) if i==j => pFalse()
    case (i: _InferenceVarType, j: _InferenceVarType) =>
      pAnd(pExclusion(i,j), pExclusion(j,i))
    case (i: _InferenceVarType, t) => pExclusion(i, t)
    case (s, j: _InferenceVarType) => pExc(j, s)
    case (s@SVarType(_, sid, _), t@SVarType(_, tid, _)) if (s==t || sid == tid) =>
      pOr(pSub(s, BOTTOM), pSub(t, BOTTOM))
    case (s@SVarType(_, id, _), t) =>
      val hEntry = (negate, false, s, t)
```

JavaScript

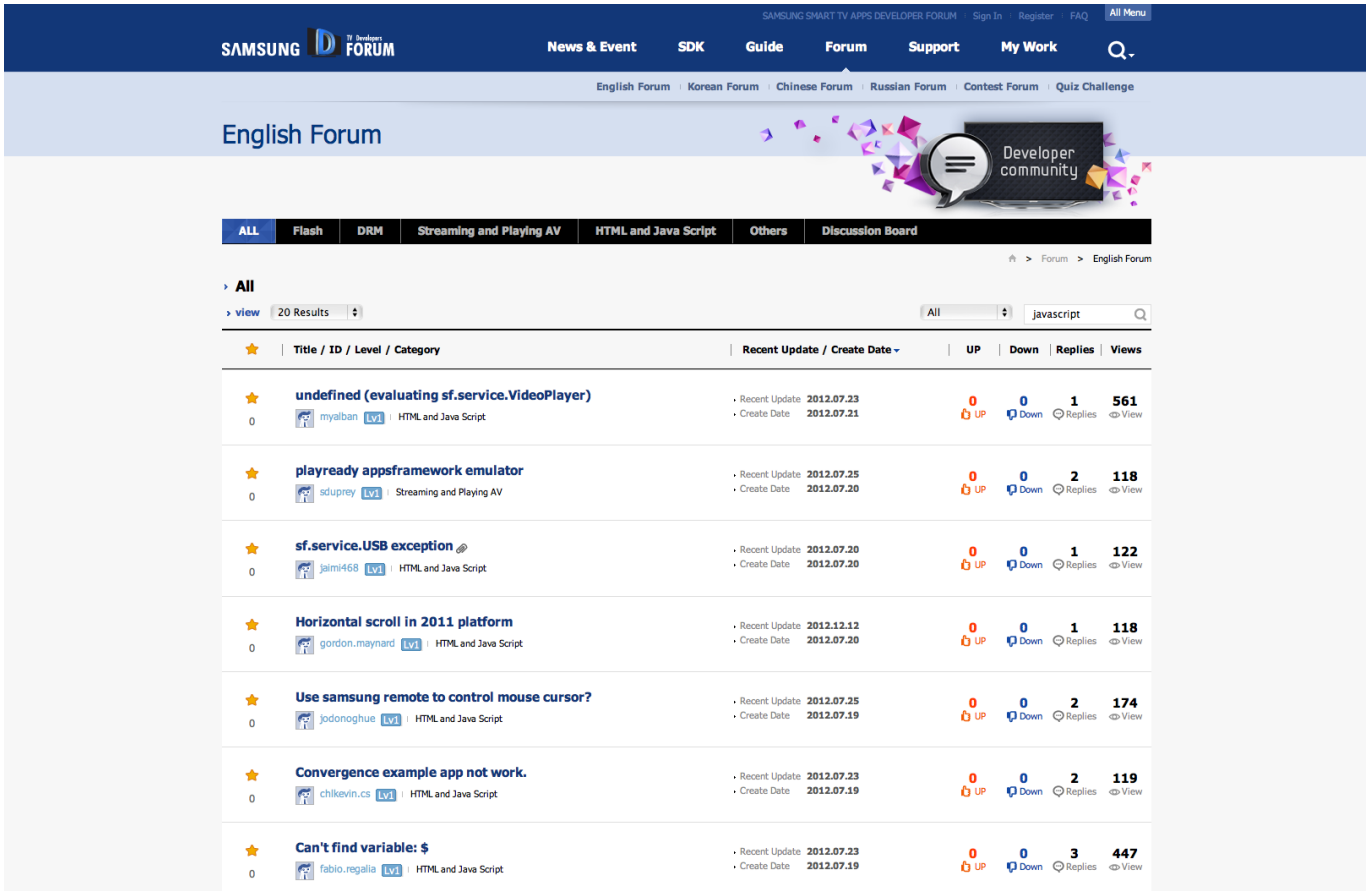
- Traditional scripting language
- ECMAScript language specification
- Java-style syntax, prototype-based, functions as values
- Dynamically typed language
  "Any type can be converted to any other reasonable type"

Issues with JavaScript

- No module system
- No user-defined types
- Unintuitive type conversions
  "A scripting language should never throw an exception [the script should just continue]" (Rob Pike, Google)
- Dynamic updates of webpage contents via HTML/DOM (Document Object Model) interface

JavaScript Web Applications Everywhere

JavaScript web application: Samsung Smart TV



JavaScript Web Applications Everywhere

JavaScript web application: Tizen Platform

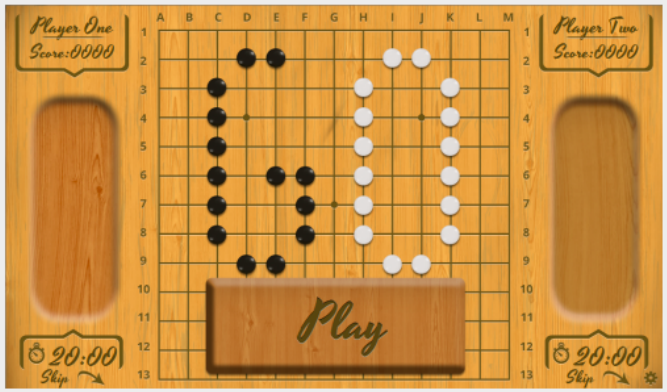
Play On Chrome: Go
Category: Games
Github Source: http://github.com/01org/webapps-go
Download Source: zip tar.gz
Installation Instructions: developer.tizen.org
Apache 2.0 License: https://github.com/01org/webapps-go/blob/master/LICENSE

Go is a board game for two players. It was implemented with HTML5/JavaScript technology.

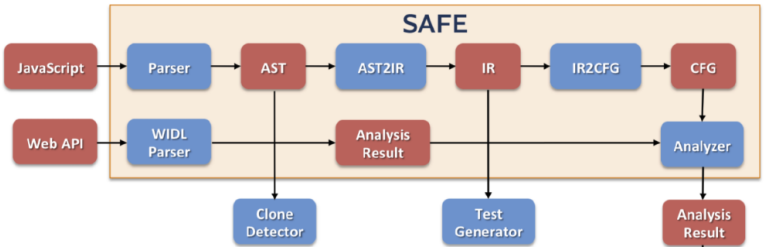
Features captured by this app include:

- Inset shadow effects using `-webkit-shadow`
- Use of `border-radius, overflow`
- Animations using webkit transitions, transforms
- Customizing scroll bar using `::-webkit-scrollbar, ::-webkit-scrollbar-track, ::-webkit-scrollbar-button:single-button`
- Manipulation of attributes and classes of the dom elements from JavaScript

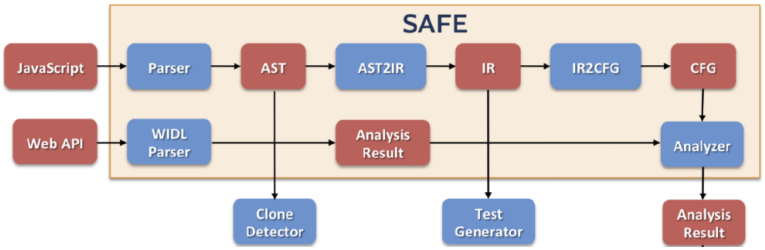
Screenshots:

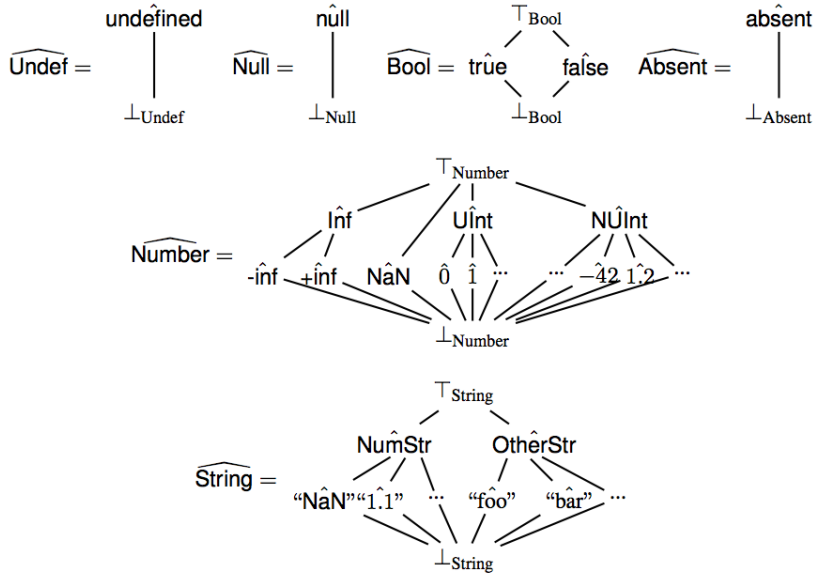


Analyzing JavaScript



Analyzing JavaScript





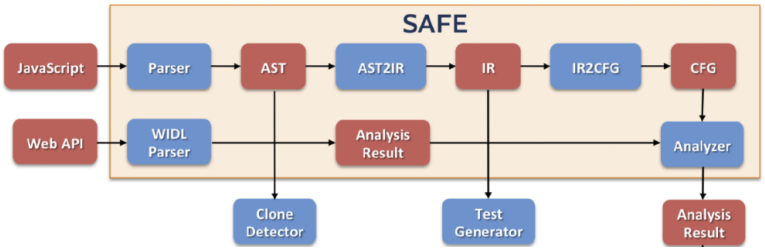
Hongki Lee, Sooncheol Won, Joonho Jin, Junhee Cho, and Sukyoung Ryu. SAFE: **Formal specification and implementation of a scalable analysis framework for ECMAScript** (FOOL'12)

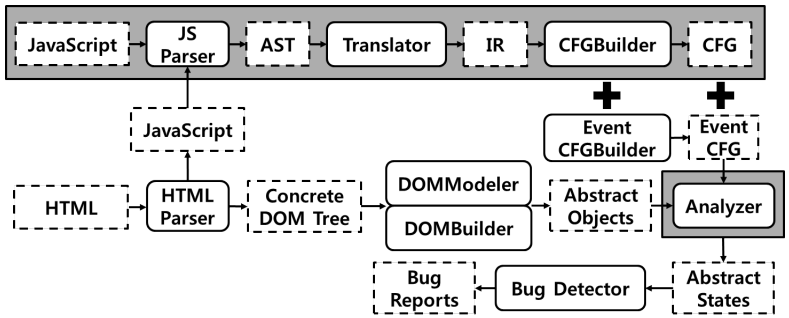
Seonghoon Kang and Sukyoung Ryu. **Formal specification of a JavaScript module system** (OOPSLA'12)

Changhee Park, Hongki Lee, and Sukyoung Ryu. **All about the `with` statement in JavaScript: Removing `with` statements in JavaScript applications** (DLS'13)

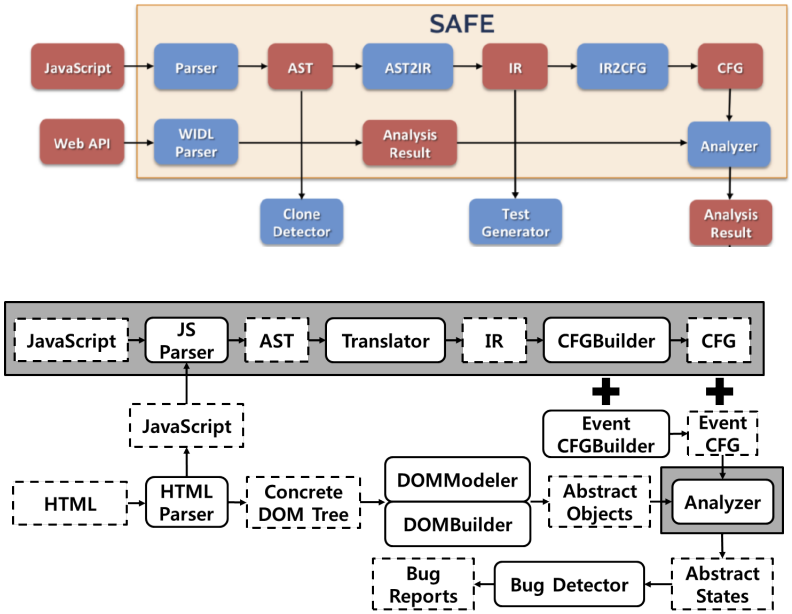
WaiTing Cheung, Sukyoung Ryu, and Sunghun Kim. **Development nature matters: An empirical study of code clones in JavaScript applications** (EMSE'15)

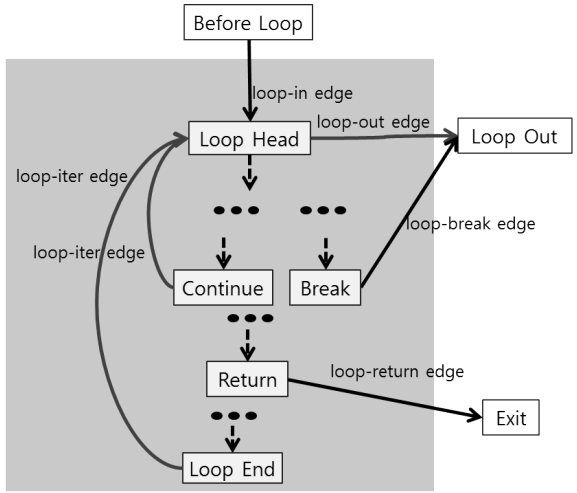
Analyzing JavaScript Web Applications





Analyzing JavaScript Web Applications

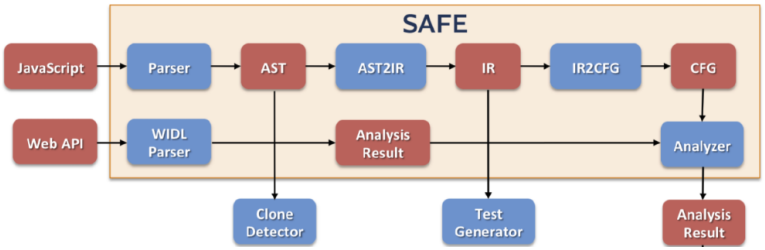


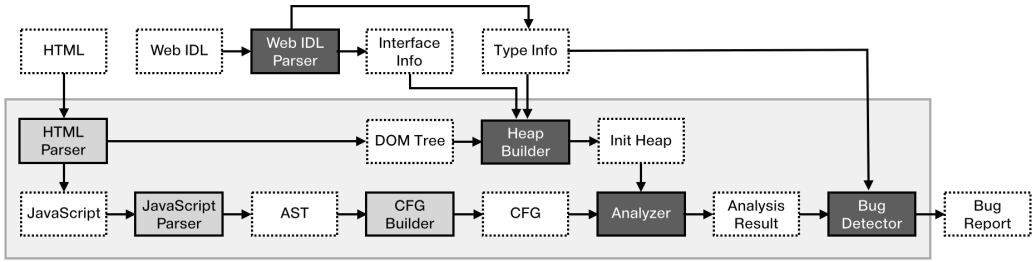


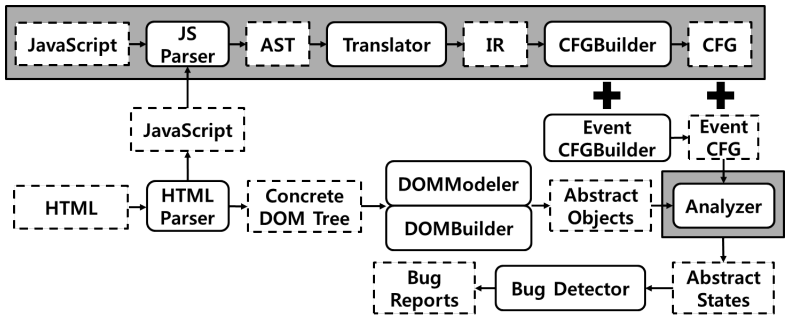
Changhee Park and Sukyoung Ryu. **Scalable and precise static analysis of JavaScript applications via loop-sensitivity** (ECOOP'15)

Changhee Park, Soocheol Won, Joonho Jin, and Sukyoung Ryu. **Static analysis of JavaScript web applications in the wild via practical DOM modeling** (ASE'15)

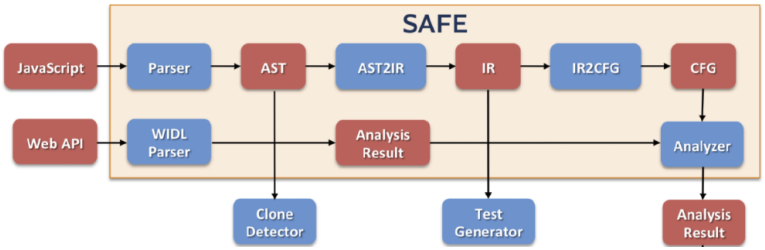
Analyzing JavaScript Web Applications in the Wild

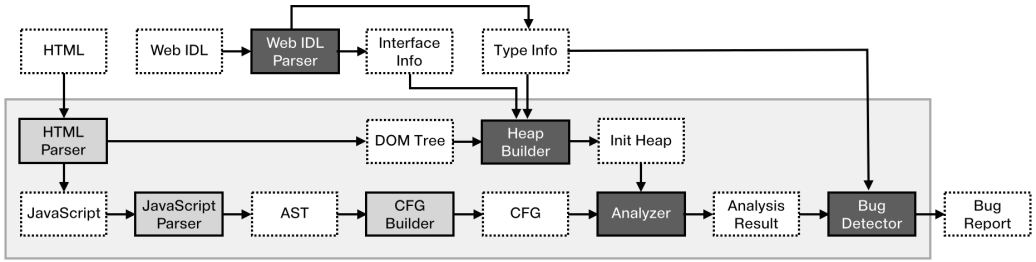


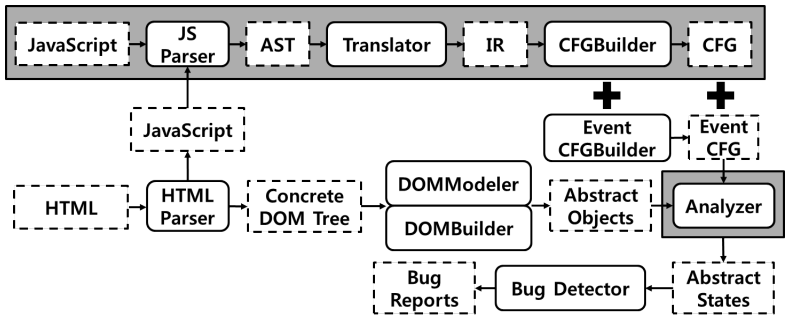




Analyzing JavaScript Web Applications in the Wild



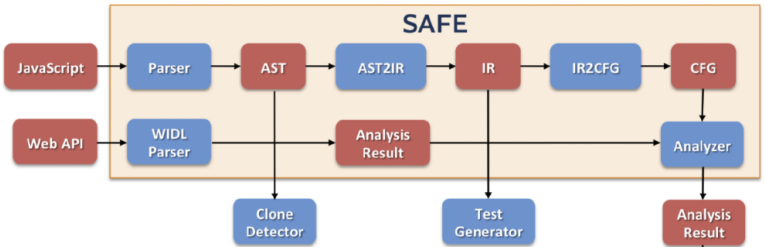


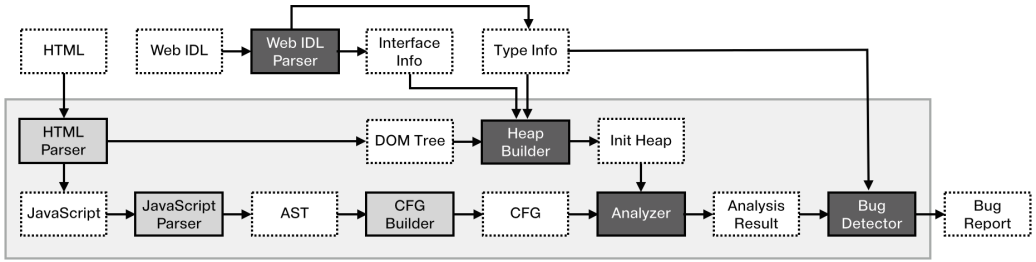


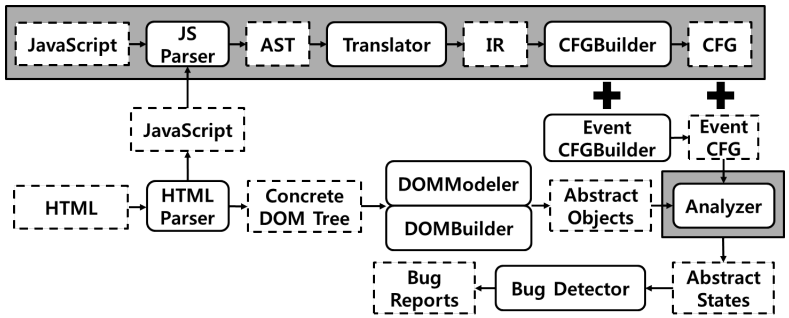
SungGyeong Bae, Hyunghun Cho, Inho Lim, and Sukyoung Ryu. SAFE$_{WAPI}$: **Web API misuse detector for web applications** (FSE'14)

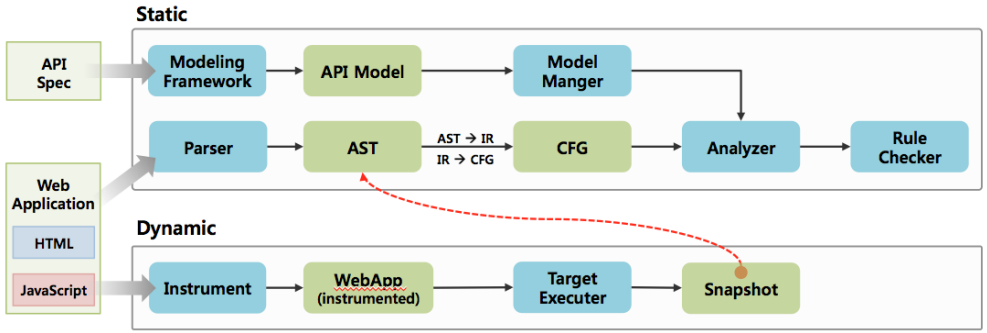
Yoonseok Ko, Hongki Lee, Julian Dolby, and Sukyoung Ryu. **Practically tunable static analysis framework for large-scale JavaScript applications** (ASE'15)

Analyzing JavaScript Web Applications in the Wild (Mostly) Statically









Analyzing JavaScript

JavaScript Bug Detection

```
3d-raytrace.js:118:19~118:23: [Warning] Trying to convert undefined to number. 'self[3]' can be undefined.
3d-raytrace.js:118:19~118:23: [Warning] Reading absent property '3' of object 'self'.
3d-raytrace.js:119:19~119:23: [Warning] Trying to convert undefined to number. 'self[7]' can be undefined.
3d-raytrace.js:119:19~119:23: [Warning] Reading absent property '7' of object 'self'.
3d-raytrace.js:120:19~120:24: [Warning] Trying to convert undefined to number. 'self[11]' can be undefined.
3d-raytrace.js:120:19~120:24:
```

```
function invertMatrix(self
    var temp = new Array(1
    var tx = -self[3];
    var ty = -self[7];
    var tz = -self[11];
    for (h = 0; h < 3; h++
        for (v = 0; v < 3;
            temp[h + v * 4
    for (i = 0; i < 11; i+
        self[i] = temp[i];
    self[3] = tx * self[0]
    self[7] = tx * self[4]
    self[11] = tx * self[8
    return self;
}
```

```
// this camera code is from notes i made ages ago, it is from
*somewhere* -- i cannot remember where
// that somewhere is
function Camera(origin, lookat, up) {
    var zaxis = normaliseVector(subVector(lookat, origin));
    var xaxis = normaliseVector(cross(up, zaxis));
    var yaxis = normaliseVector(cross(xaxis, subVector([0,0,0]
, zaxis)));
    var m = new Array(16);
    m[0] = xaxis[0]; m[1] = xaxis[1]; m[2] = xaxis[2];
    m[4] = yaxis[0]; m[5] = yaxis[1]; m[6] = yaxis[2];
    m[8] = zaxis[0]; m[9] = zaxis[1]; m[10] = zaxis[2];
    invertMatrix(m);
    m[3] = 0; m[7] = 0; m[11] = 0;
    this.origin = origin;
    this.directions = new Array(4);
    this.directions[0] = normalise([-0.7,  0.7, 1]);
    this.directions[1] = normalise([ 0.7,  0.7, 1]);
    this.directions[2] = normalise([ 0.7, -0.7, 1]);
    this.directions[3] = normalise([-0.7, -0.7, 1]);
```

Analyzing JavaScript Web Applications

JavaScript Bug Detection

```
3d-raytrace.js:118:19~118:23: [Warning] Trying to convert undefined to number. 'self[3]' can be undefined.
3d-raytrace.js:118:19~118:23: [Warning] Reading absent property '3' of object 'self'.
3d-raytrace.js:119:19~119:23: [Warning] Trying to convert undefined to number. 'self[7]' can be undefined.
3d-raytrace.js:119:19~119:23: [Warning] Reading absent property '7' of object 'self'.
3d-raytrace.js:120:19~120:24: [Warning] Trying to convert undefined to number. 'self[11]' can be undefined.
3d-raytrace.js:120:19~120:24:
```

```
function invertMatrix(self
    var temp = new Array(1
    var tx = -self[3];
    var ty = -self[7];
    var tz = -self[11];
    for (h = 0; h < 3; h++
        for (v = 0; v < 3;
            temp[h + v * 4
    for (i = 0; i < 11; i+
        self[i] = temp[i];
    self[3] = tx * self[0]
    self[7] = tx * self[4]
    self[11] = tx * self[8
```

```
// this camera code is from notes i made ages ago, it is from
*somewhere* -- i cannot remember where
// that somewhere is
function Camera(origin, lookat, up) {
    var zaxis = normaliseVector(subVector(lookat, origin));
    var xaxis = normaliseVector(cross(up, zaxis));
    var yaxis = normaliseVector(cross(xaxis, subVector([0,0,0]
, zaxis)));
    var m = new Array(16);
    m[0] = xaxis[0]; m[1] = xaxis[1]; m[2] = xaxis[2];
    m[4] = yaxis[0]; m[5] = yaxis[1]; m[6] = yaxis[2];
    m[8] = zaxis[0]; m[9] = zaxis[1]; m[10] = zaxis[2];
    invertMatrix(m);
    m[3] = 0; m[7] = 0; m[11] = 0;
    this.origin = origin;
```

Web Page Bug Detection (I)

Analyzing JavaScript Web Applications in the Wild

JavaScript Bug Detection

```
3d-raytrace.js:118:19~118:23: [Warning] Trying to convert undefined to number. 'self[3]' can be undefined.
3d-raytrace.js:118:19~118:23: [Warning] Reading absent property '3' of object 'self'.
3d-raytrace.js:119:19~119:23: [Warning] Trying to convert undefined to number. 'self[7]' can be undefined.
3d-raytrace.js:119:19~119:23: [Warning] Reading absent property '7' of object 'self'.
3d-raytrace.js:120:19~120:24: [Warning] Trying to convert undefined to number. 'self[11]' can be undefined.
3d-raytrace.js:120:19~120:24:
```

```
function invertMatrix(self
    var temp = new Array(1
    var tx = -self[3];
    var ty = -self[7];
    var tz = -self[11];
    for (h = 0; h < 3; h++
        for (v = 0; v < 3;
            temp[h + v * 4
    for (i = 0; i < 11; i+
        self[i] = temp[i];
    self[3] = tx * self[0]
    self[7] = tx * self[4]
    self[11] = tx * self[8
```

```
// this camera code is from notes i made ages ago, it is from
*somewhere* -- i cannot remember where
// that somewhere is
function Camera(origin, lookat, up) {
    var zaxis = normaliseVector(subVector(lookat, origin));
    var xaxis = normaliseVector(cross(up, zaxis));
    var yaxis = normaliseVector(cross(xaxis, subVector([0,0,0]
, zaxis)));
    var m = new Array(16);
    m[0] = xaxis[0]; m[1] = xaxis[1]; m[2] = xaxis[2];
    m[4] = yaxis[0]; m[5] = yaxis[1]; m[6] = yaxis[2];
    m[8] = zaxis[0]; m[9] = zaxis[1]; m[10] = zaxis[2];
    invertMatrix(m);
    m[3] = 0; m[7] = 0; m[11] = 0;
    this.origin = origin;
```

Web Application Bug Detection (I)

```
loading : function(i) {
        // alert("<img src='image/loading_"+i+".png>");
        var tmpdivB = document.getElementById("waitIcon");

        i = i == null ? 1 : i;

        // set timeout is 30s
        this.totalCnt++;
        if (this.totalCnt > 200) {
                this.endLoading();
                tmpdivB.className = "";
                this.showNoResultFound(tmpdivB);
                return;
        }
```

```
<div id="waitIconDiv" class="waitIconType" style="display:none;">
    <img src = "css/images/009.gif" width = "50px" height = "50px"/>
</div>
```

Web Page Bug Detection (I)

Analyzing JavaScript Web Applications in the Wild (Mostly) Statically

JavaScript Bug Detection

```
3d-raytrace.js:118:19~118:23: [Warning] Trying to convert undefined to number. 'self[3]' can be undefined.
3d-raytrace.js:118:19~118:23: [Warning] Reading absent property '3' of object 'self'.
3d-raytrace.js:119:19~119:23: [Warning] Trying to convert undefined to number. 'self[7]' can be undefined.
3d-raytrace.js:119:19~119:23: [Warning] Reading absent property '7' of object 'self'.
3d-raytrace.js:120:19~120:24: [Warning] Trying to convert undefined to number. 'self[11]' can be undefined.
3d-raytrace.js:120:19~120:24:
```

```
function invertMatrix(self
    var temp = new Array(1
    var tx = -self[3];
    var ty = -self[7];
    var tz = -self[11];
    for (h = 0; h < 3; h++
        for (v = 0; v < 3;
            temp[h + v * 4
    for (i = 0; i < 11; i+
        self[i] = temp[i];
    self[3] = tx * self[0]
    self[7] = tx * self[4]
    self[11] = tx * self[8
```

```
// this camera code is from notes i made ages ago, it is from
*somewhere* -- i cannot remember where
// that somewhere is
function Camera(origin, lookat, up) {
    var zaxis = normaliseVector(subVector(lookat, origin));
    var xaxis = normaliseVector(cross(up, zaxis));
    var yaxis = normaliseVector(cross(xaxis, subVector([0,0,0]
zaxis)));
    var m = new Array(16);
    m[0] = xaxis[0]; m[1] = xaxis[1]; m[2] = xaxis[2];
    m[4] = yaxis[0]; m[5] = yaxis[1]; m[6] = yaxis[2];
    m[8] = zaxis[0]; m[9] = zaxis[1]; m[10] = zaxis[2];
    invertMatrix(m);
    m[3] = 0; m[7] = 0; m[11] = 0;
    this.origin = origin;
```

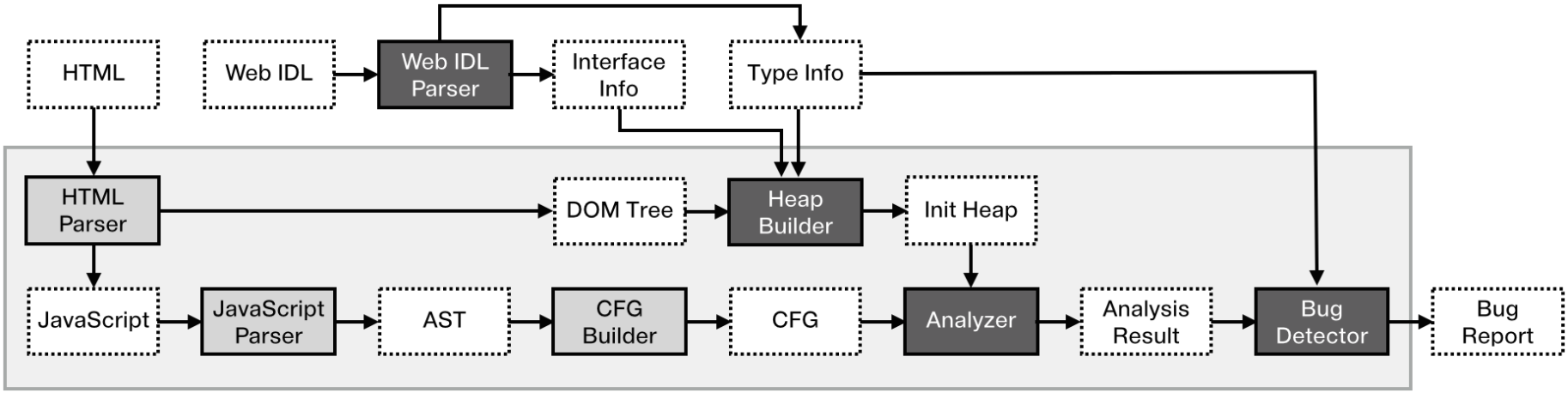
Web Application Bug Detection (I)

```
loading : function(i) {
        // alert("<img src='image/loading_"+i+".png>");
        var tmpdivB = document.getElementById("waitIcon");

        i = i == null ? 1 : i;

        // set timeout is 30s
        this.totalCnt++;
        if (this.totalCnt > 200) {
                this.endLoading();
                tmpdivB.className = "";
                this.showNoResultFound(tmpdivB);
                return;
        }
```

```
<div id="waitIconDiv" class="waitIconType" style="display:none;">
    <img src = "css/images/009.gif" width = "50px" height = "50px"/>
</div>
```

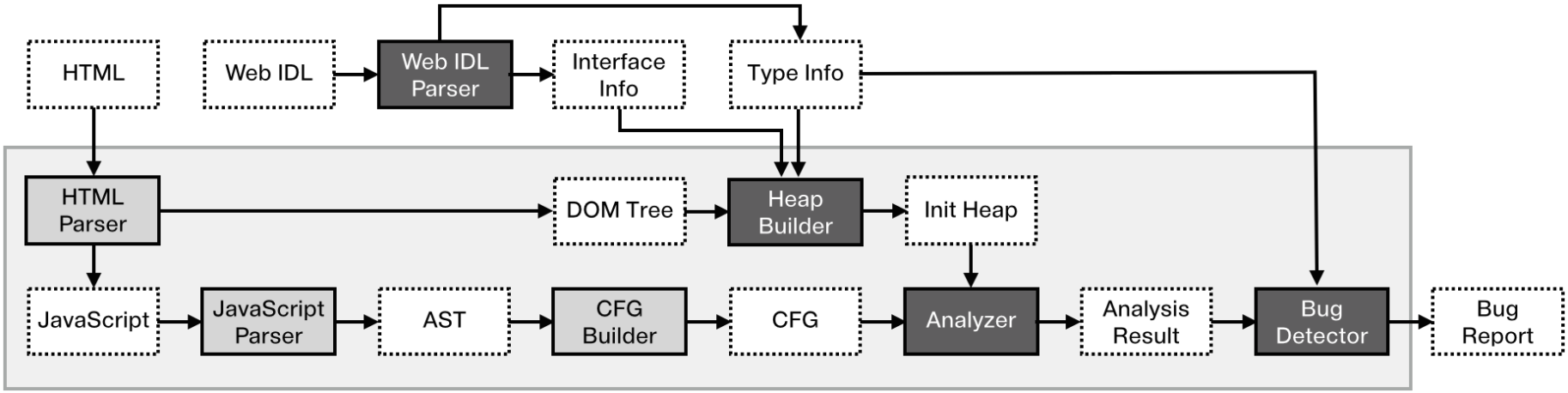
Web Page Bug Detection (I)

$\text{SAFE}_{\text{WAPI}}$: SAFE for Web Apps with Web APIs



- Web APIs specified in Web IDL
- Automatic modeling of platform-generated objects
- Automatic modeling of API functions

$\text{SAFE}_{\text{WAPI}}$: SAFE **for Web Apps with Web APIs**



- Web APIs specified in Web IDL
- Automatic modeling of platform-generated objects
- Automatic modeling of API functions

Analyzing JavaScript Web Applications in the Wild FSE'14

- Model platform APIs written in IDLs including Web IDL
- by *automatic modeling* from Web APIs
- to analyze real-world web applications interacting with platform libraries

```
[NoInterfaceObject] interface CalendarManager {
  void getCalendars(CalendarType type,
                    CalendarArraySuccessCallback successCallback,
                    optional ErrorCallback? errorCallback);

  Calendar getUnifiedCalendar(CalendarType type);

  Calendar getDefaultCalendar(CalendarType type);

  Calendar getCalendar(CalendarType type, CalendarId id);
};
```

```
enum CalendarType { "EVENT", "TASK" };
```

```
[Callback=FunctionOnly, NoInterfaceObject]
interface CalendarArraySuccessCallback {
  void onsuccess(Calendar[] calendars);
};
```

```
function successCB(calendars) {
  calendars[0].foo;
}
var bar = "EVEN";
webapis.calendar.getCalendars(
                    bar, successCB);
```

How?

Compare API function requirements and analysis results

- Exception handling
- Number of arguments
- Types of arguments
- Callback function calls
- ...

How?—Types

JavaScript values vs Web IDL types

- JavaScript types
  Undefined, Null, Boolean, String, Number, Object
- Web IDL types
  interface, dictionary, enumeration, union, user-defined types, ...

JavaScript Type Conversion Table

Throws TypeMismatchError

Throws TypeMismatchError conditionally

Argument Value{}	Supplementary*		Parameter Type								
	nullable	optional	integer*	float point*	DOMString	boolean	enum	array	dictionary	callback	interface
undefined	-	*default value as spec. described*	0	NaN	"undefined"	false	*if the value of DOMString column is not available in enum.* *(but, setting the invalid value to an attribute will ignore the exception)*				
null	*follows spec.*	-	0	0.0	"null"	false					
true / false	-	-	1 / 0	1.0 / 0.0	"true" / "false"	true / false					
"" (empty string)	-	-	0	0.0	""	false					
"1.2" (numeric)	-	-	1	1.2	"1.2"	true					
"one" (non-numeric)	-	-	0	NaN	"one"	true					
0	-	-	0	0.0	"0"	false					
Infinity	-	-	0	NaN	"Infinity"	true					
1 (non-zero)	-	-	1	1.0	"1"	true					
NaN	-	-	0	NaN	"NaN"	false					
{ } (any object)	-	-	*by result of toString() or valueOf()*		"[object Object]"	true			{ }		*if it is not a callback interface*
[9] (1 numeric elt)	-	-	9	9.0	"9"	true		[9]			
['a'] (any other array)	-	-	0	NaN	"a"	true		['a']			
function() {}	-	-	0	NaN	"function() {}"	true			new Object(fu		*if it is not FunctionOnly inte*

How?—Functions

Mockup values and call flows

- Return values from API functions
- Argument values for callback functions

```
function sCB(calendars){
        calendars[3].foo;
}
function eCB(calendars){
}
x = "EV" + "ENT";
webapis.calendar.getCalendars(x, sCB, eCB);
```

How?—Functions: Abstract Type Values

```
[NoInterfaceObject] interface File {
  readonly attribute File? parent;

  readonly attribute boolean readOnly;

  readonly attribute boolean isFile;

  readonly attribute boolean isDirectory;

  readonly attribute Date? created;

  readonly attribute Date? modified;

  readonly attribute DOMString path;

  readonly attribute DOMString name;

  readonly attribute DOMString fullPath;

  readonly attribute unsigned long long fileSize;

  readonly attribute long length;

  DOMString toURI() raises(WebAPIException);
```

```
#FileMockupLoc -> {
  @parent : #FileMockupLoc,
  readOnly : T_Bool,
  isFile : T_Bool,
  isDirectory : T_Bool,
  @created: #DateMockupLoc,
  @modified: #DateMockupLoc,
  path: T_Str,
  name: T_Str,
  fullPath: T_Str,
  fileSize: T_UInt,
  length: T_Int,
  toURI: #39712,
}
```

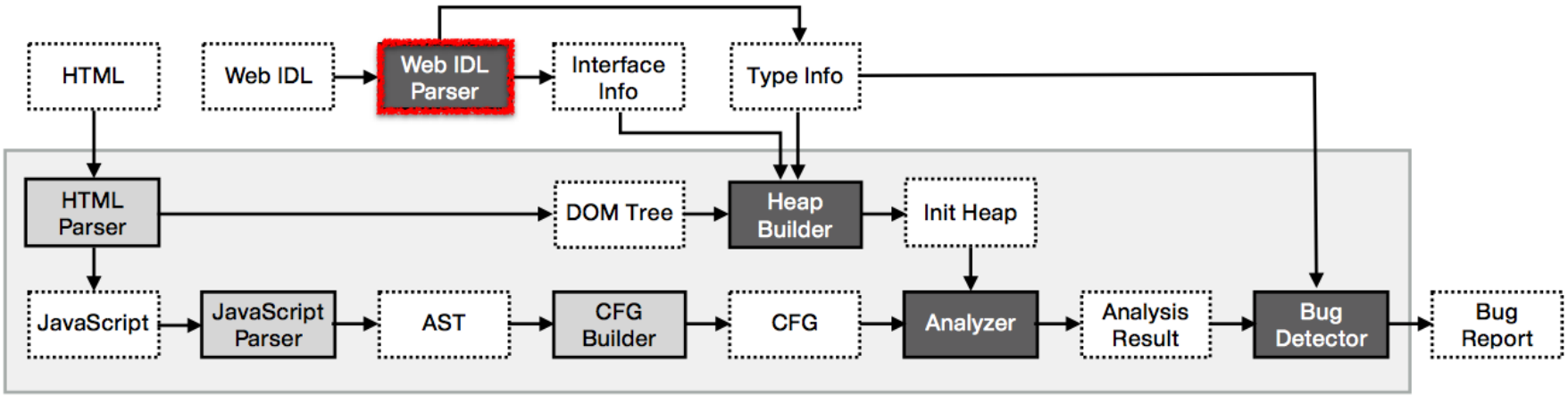
How?—Functions: Callback Flows

Callback function calls with abstract type values returning abstract type values

```
1  function API(arg_1, ..., CB_1, ..., CB_n) {
2      if (...) {
3          CB_1(arg_11, ...);
4      } else if (...) {
5          CB_2(arg_21, ...);
6      } .
7        .
8        .
9      } else if (...) {
10         CB_n(arg_n1, ...);
11     }
12     return (return_value);
13 }
```

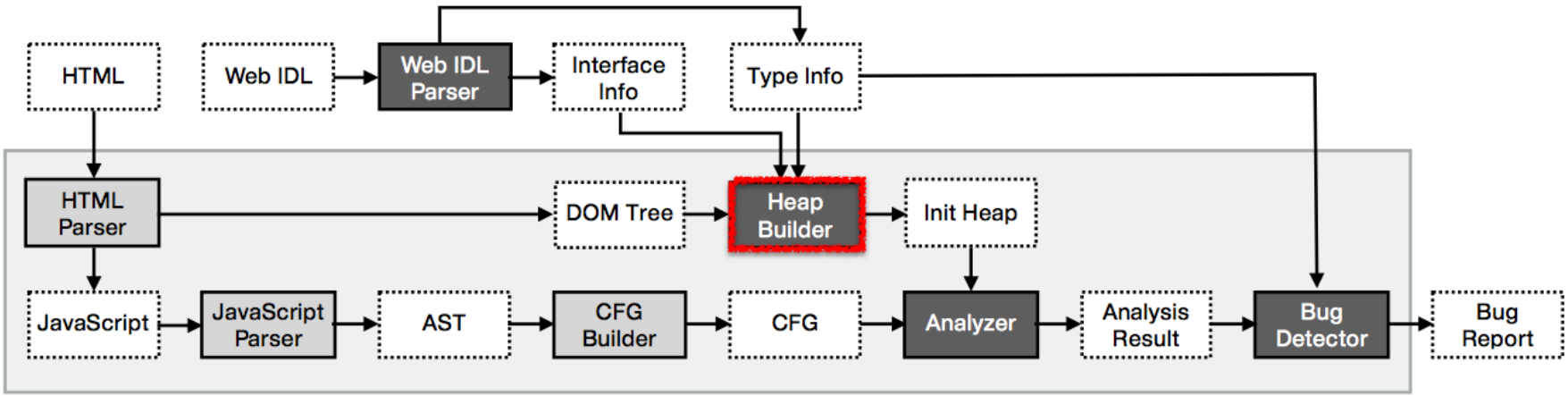
Implementation

Automatic extraction of APIs from API specifications



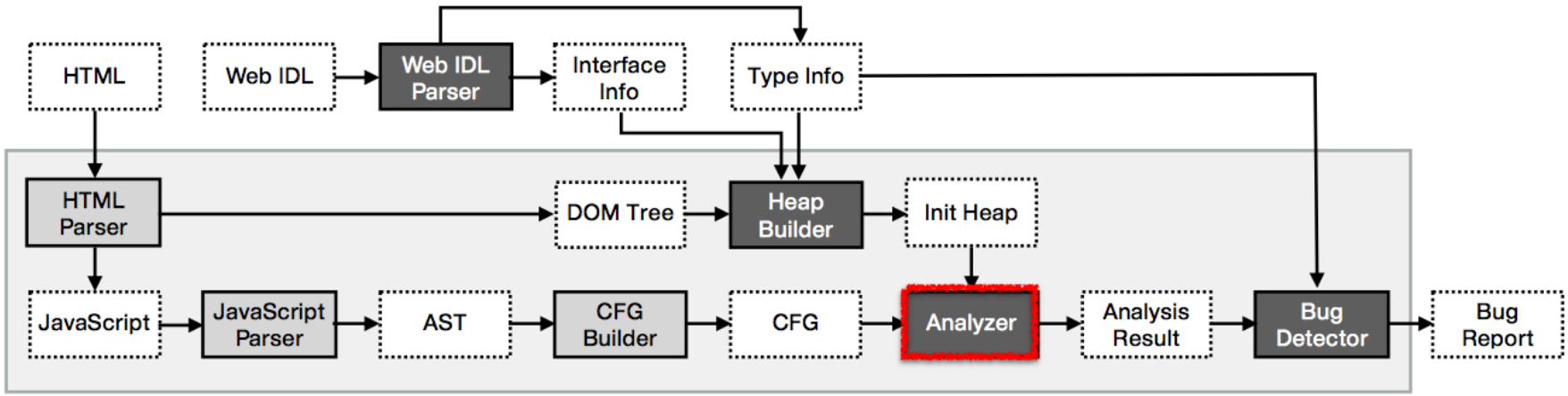
Implementation

Automatic modeling of types in APIs into the analysis heap



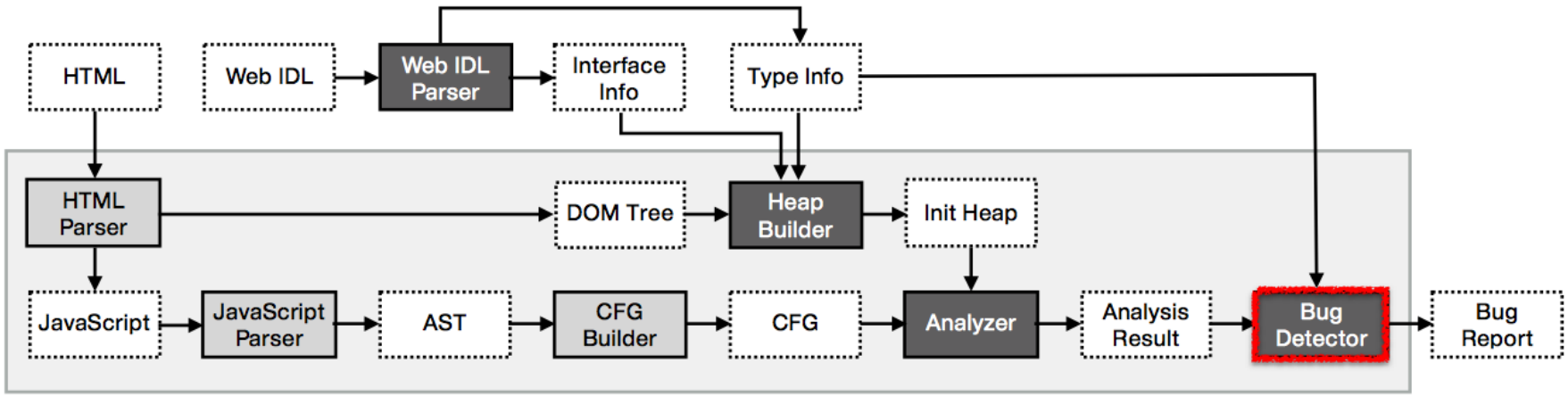
Implementation

Automatic modeling of callback function calls

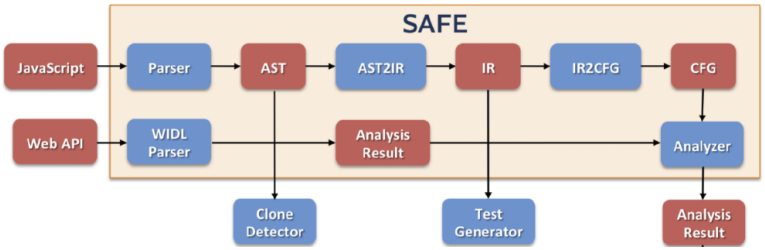


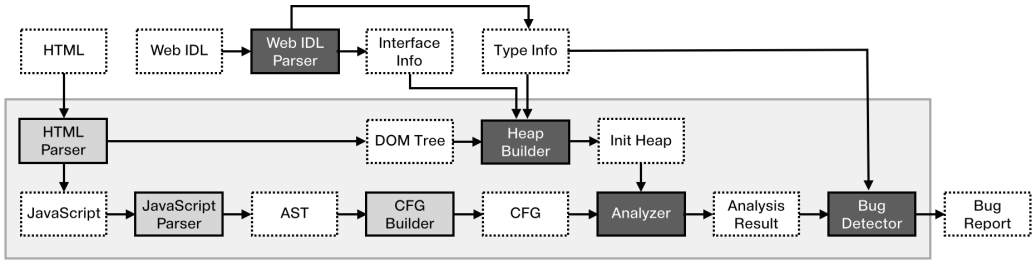
Implementation

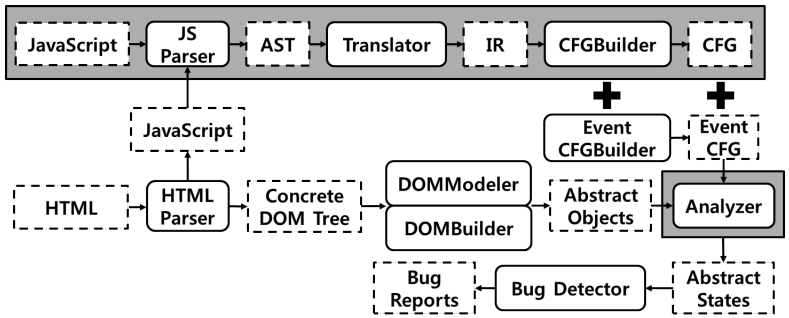
Automatic detection of API misuses

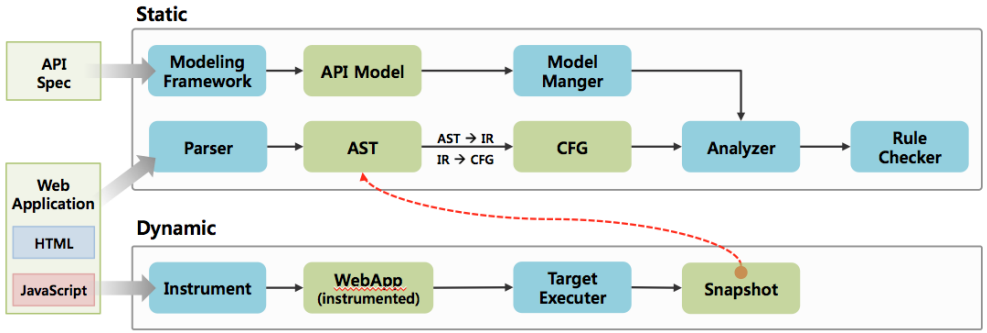


Analyzing JavaScript Web Applications in the Wild (Mostly) Statically



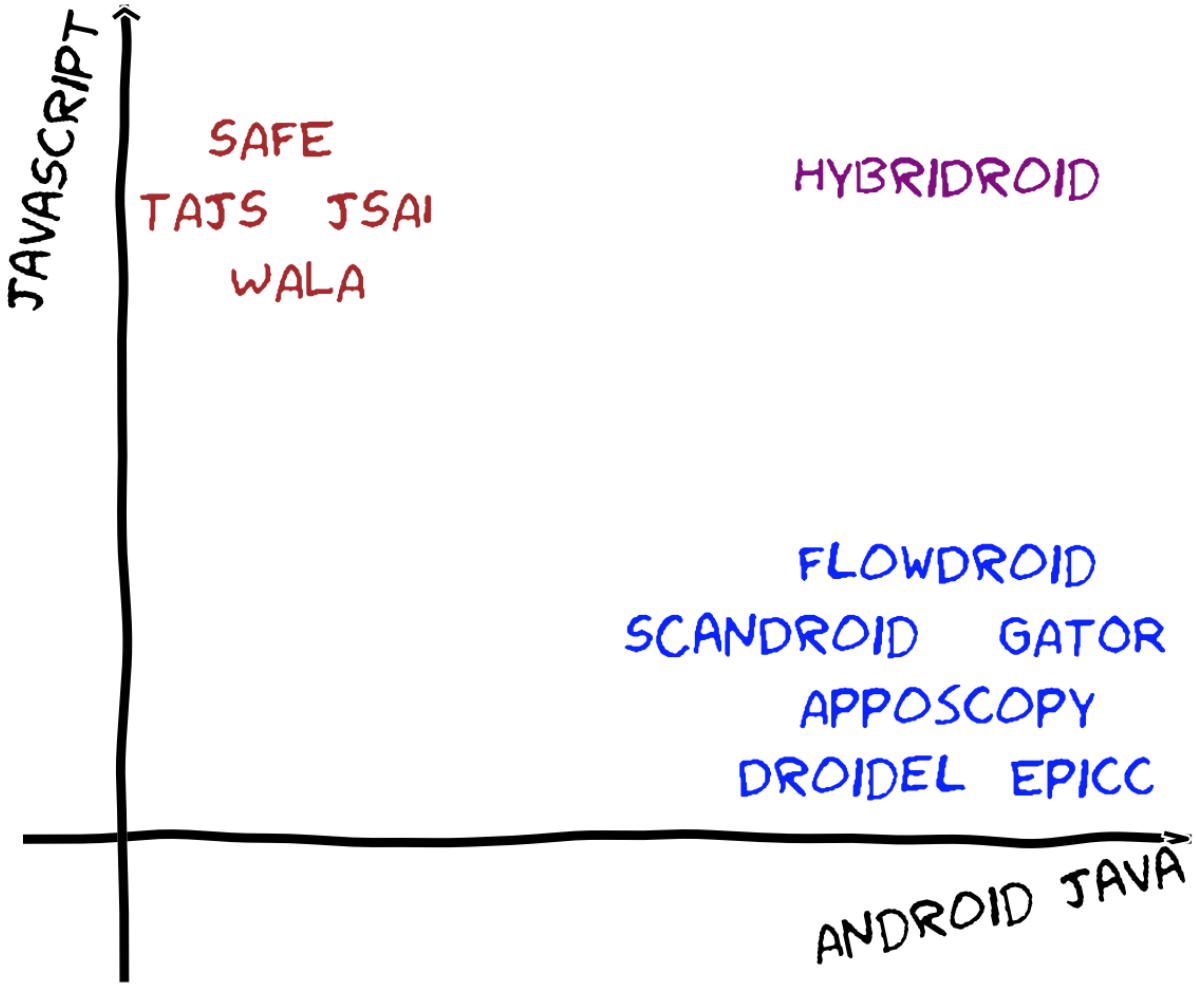






http://safe.kaist.ac.kr

Analyis of Android Hybrid Applications



Many Applications for Multiple Platforms

Many mobile platforms out there.

Many Applications for Multiple Platforms

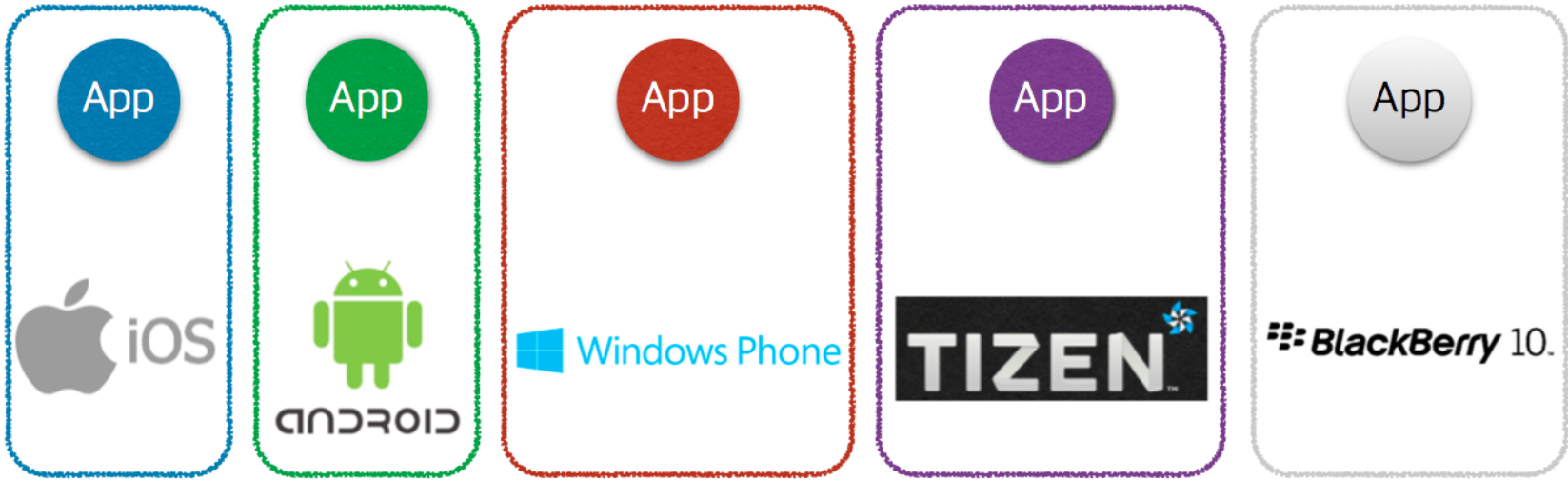
Many mobile platforms out there.

Many Applications for Multiple Platforms

To support multiple platforms with native applications,

- need to implement one application per platform;
- need to repeat application development multiple times.

Web applications cannot use device features.

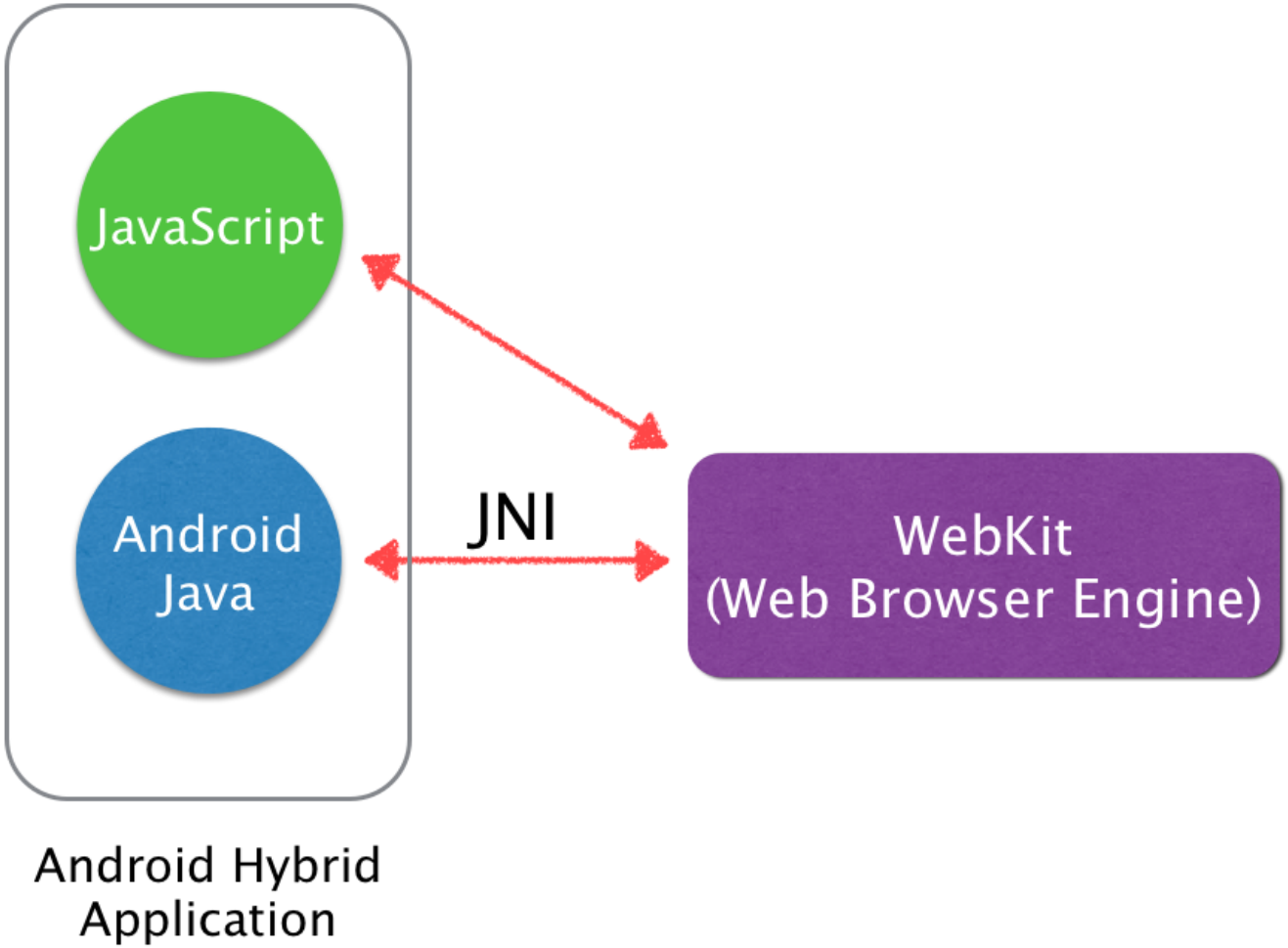


Hybrid Applications for Multiple Platforms

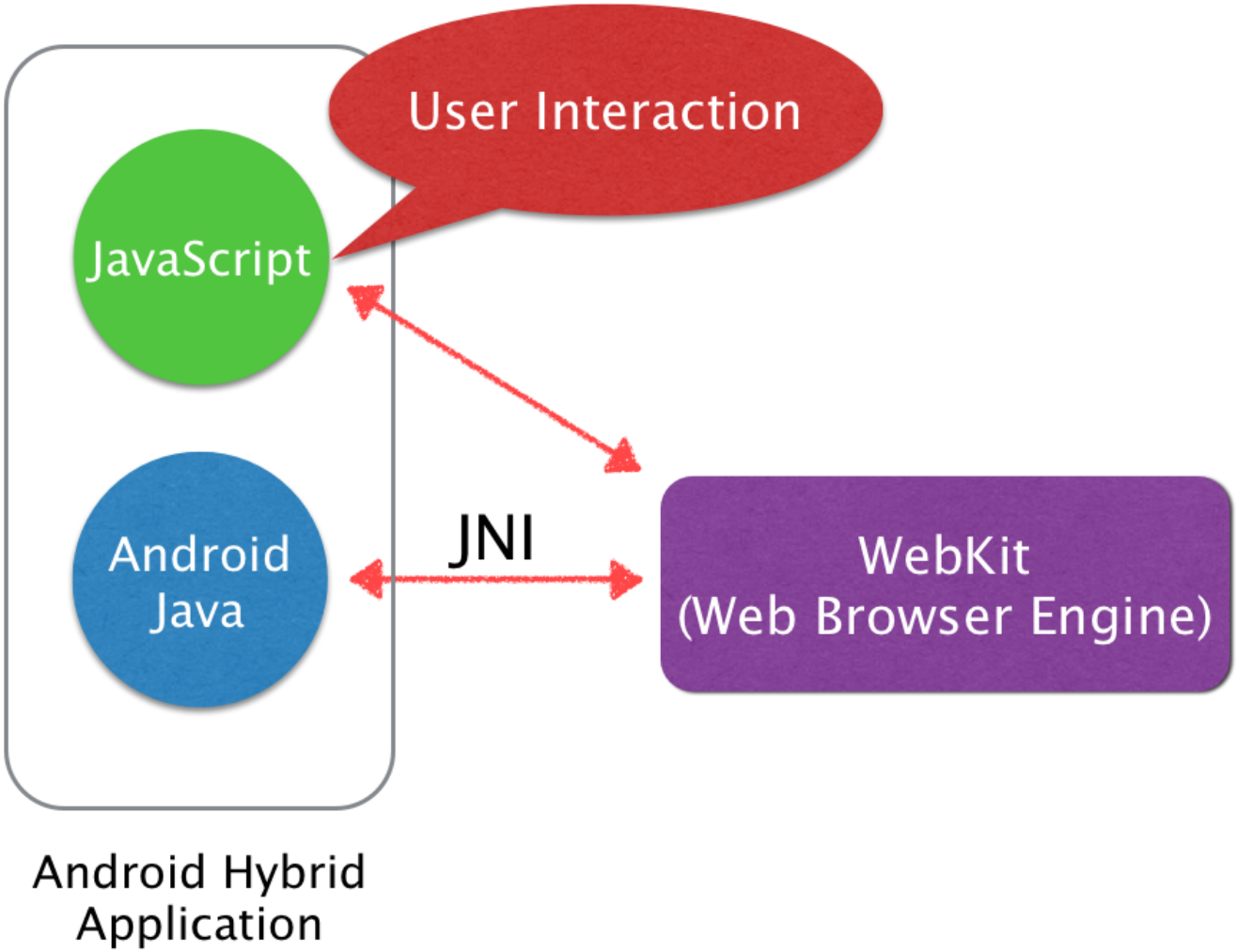
Hybrid applications could be one solution.

- Both HTML5 code (HTML, CSS, and JavaScript) and native device features, such as a camera or accelerometer.

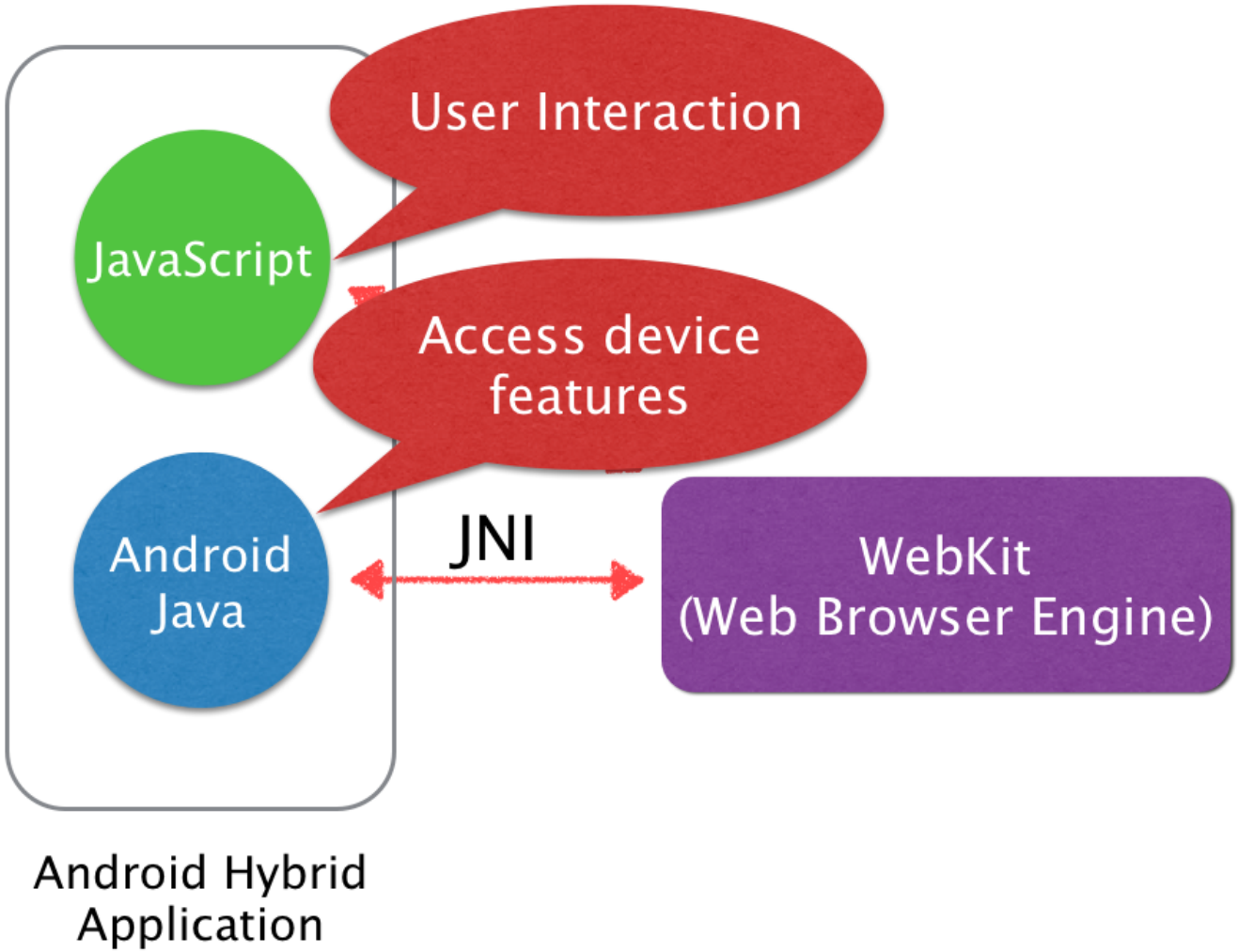
Hybrid Applications in Android



Hybrid Applications in Android



Hybrid Applications in Android



Hybrid Applications for Multiple Platforms

Among 1,402,894 (1,186,488 free) Android applications from PlayDrone[1], we downloaded and decompiled

- 151 Android applications
- 56 (out of 151) hybrid applications
- 47 (out of 56) using `addJavascriptInterface`
- 13 (out of 56) using Apache Cordova

[1] "A Measurement Study of Google Play" Nicolas Viennot, Edward Garcia, and Jason Nieh.
`http://www.cs.columbia.edu/~nieh/pubs/sigmetrics2014_playdrone.pdf`
`https://github.com/nviennot/playdrone`

Problems with Hybrid Applications

One malware for multiple platforms!

Problems with Hybrid Applications in Android

One malware for multiple platforms!

- Silent misbehaviors due to API misuse
  - Use of `void` results from Java methods in JS
  - Passing values of incompatible types between Java & JS
  - Wrong number of arguments to Java methods from JS
- Private data leakage between Java & JS

Problems with Hybrid Applications in Android

One malware for multiple platforms!

- Silent misbehaviors due to API misuse
  - Use of `void` results from Java methods in JS
  - Passing values of incompatible types between Java & JS
  - Wrong number of arguments to Java methods from JS
- Private data leakage between Java & JS

Questions | Tags | Users | Badges

Stack Overflow is a question and answer site for professional and enthusiast programmers. It's 100% free, no registration required.

Passing a JavaScript object using addJavascriptInterface() on Android

Problems with Hybrid Applications in Android

One malware for multiple platforms!

- Silent misbehaviors due to API misuse
  - Use of `void` results from Java methods in JS
  - Passing values of incompatible types between Java & JS
  - Wrong number of arguments to Java methods from JS
- Private data leakage between Java & JS

1 Does anybody know if there is actually any documentation on exactly what can be taken as parameters and what can be returned from a method which is part of the injected java object? – Wayne Uroda May 21 '13 at 4:07

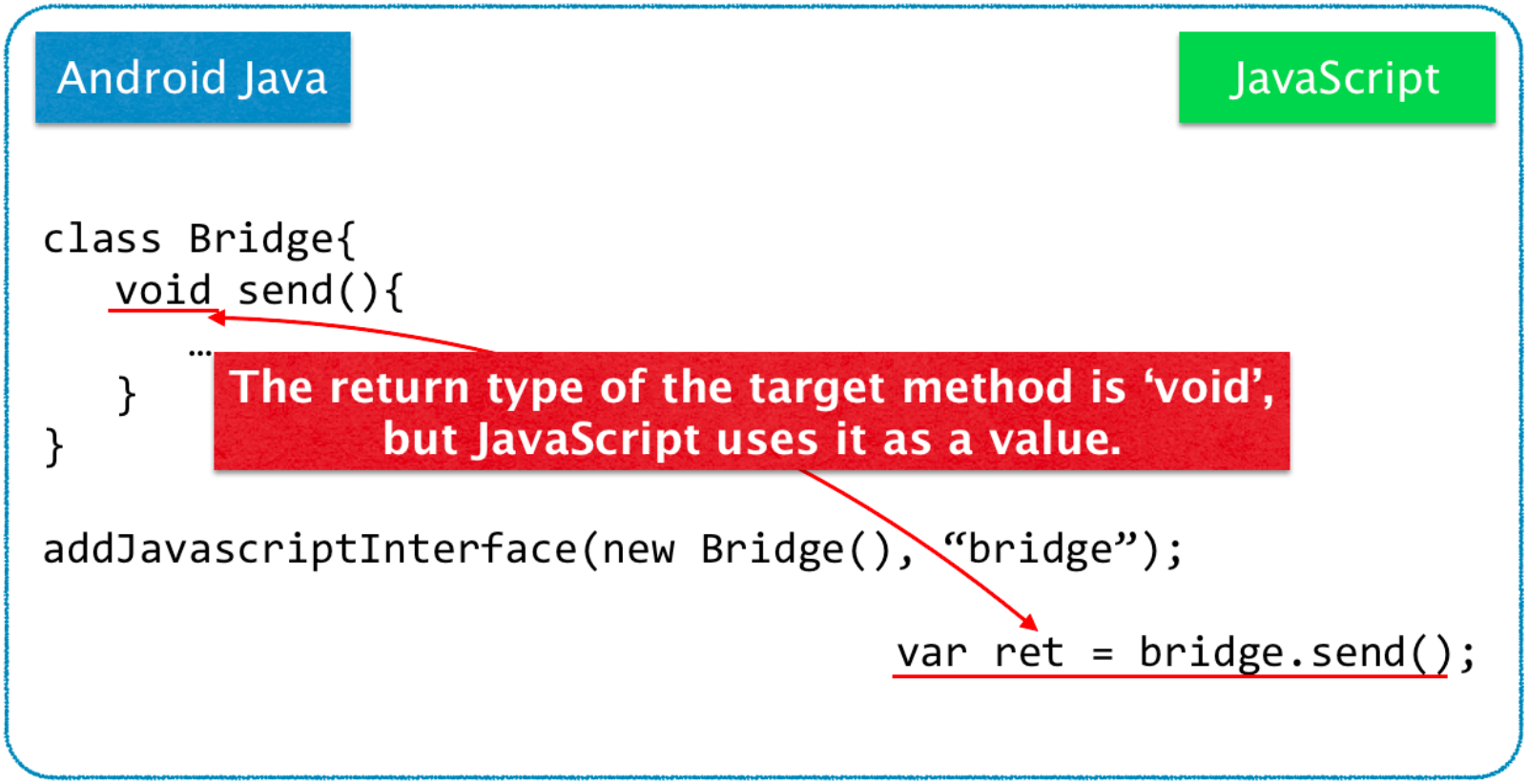
@WayneUroda: Alas, I am not aware of any formal documentation. – CommonsWare May 21 '13 at 4:10

1 What madness... – Wayne Uroda May 21 '13 at 5:12

From my tests, you can pass any primitive, string, of array of those. in case of polymorphism, though, it appears that the method that accepts float has priority (probably a matter of ordering the methods). I'll post something about that when I have investigated further – njzk2 Sep 3 '13 at 10:16

@WayneUroda I'm also looking for some documentation. Does anybody know something in the mean time? – domi May 14 '14 at 11:37

Use of `void` from Java in JS



Incompatible Types between Java & JS

Android Java

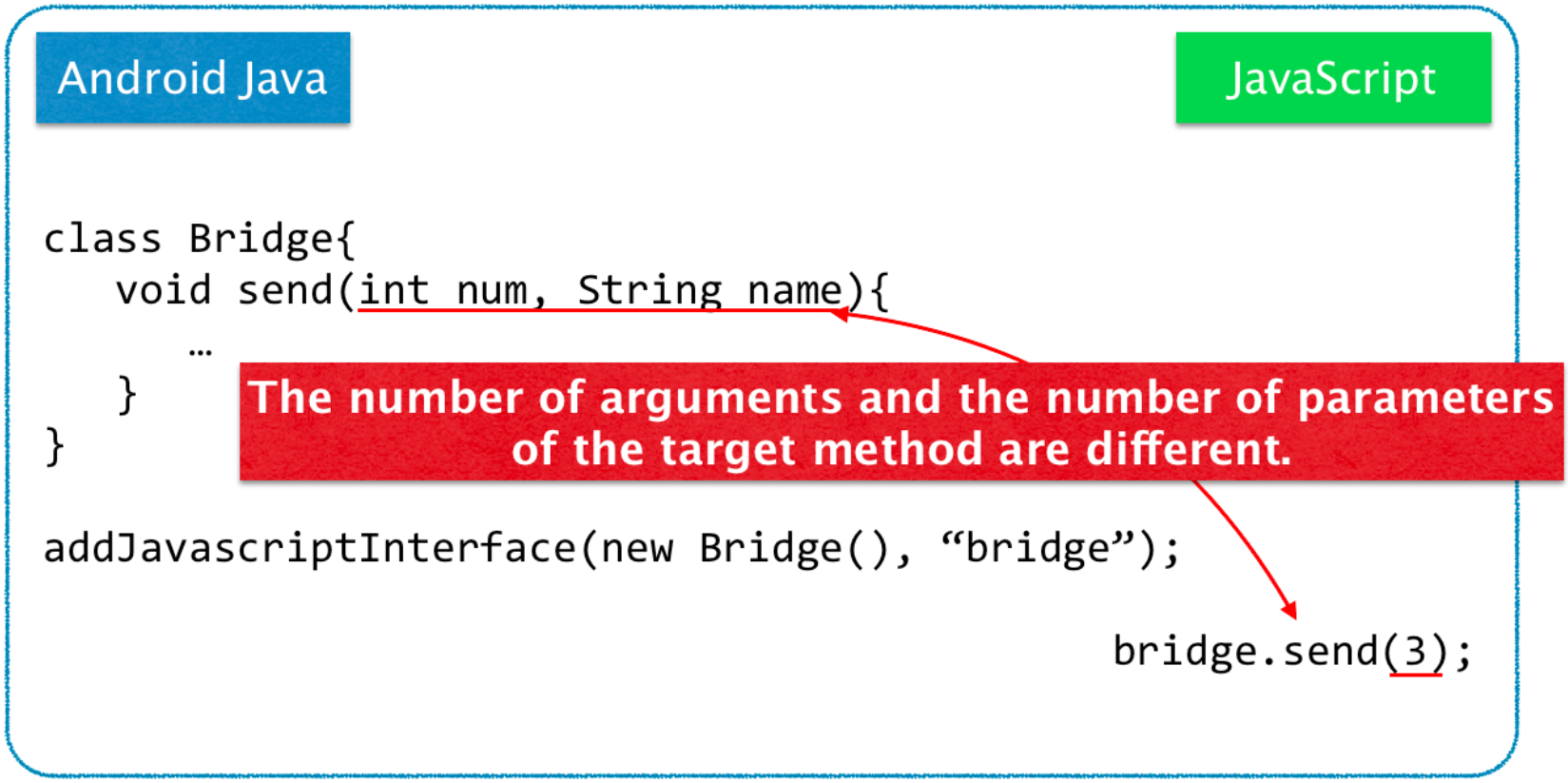
JavaScript

```
class Bridge{
    void send(int num){
        …
    }
}

addJavascriptInterface(new Bridge(), “bridge”);
```

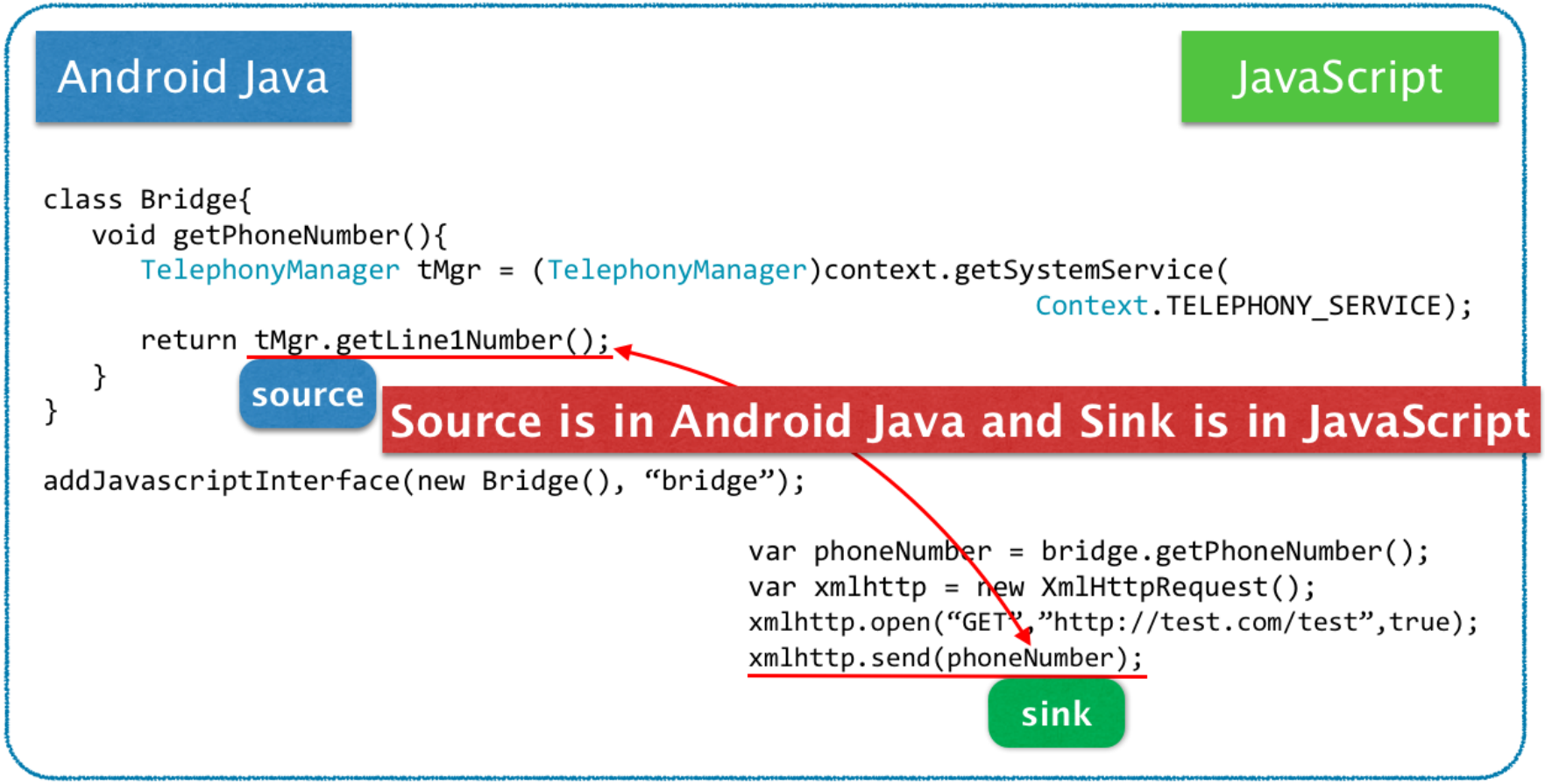
```
bridge.send(“Hybrid”);
```

The argument type and the parameter type of the target method are not compatible.

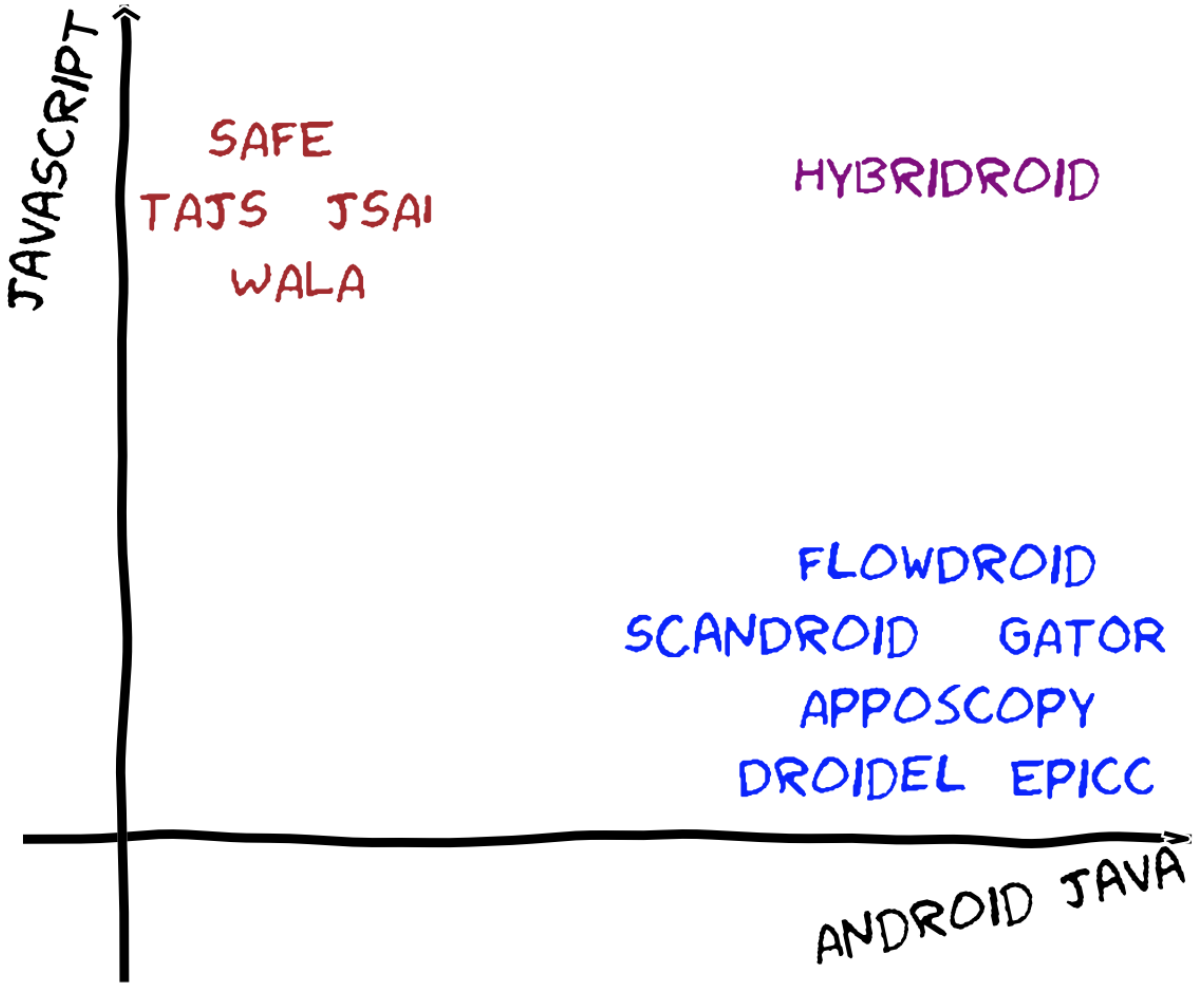
Wrong Number of Args between Java & JS



Private Data Leakage between Java & JS

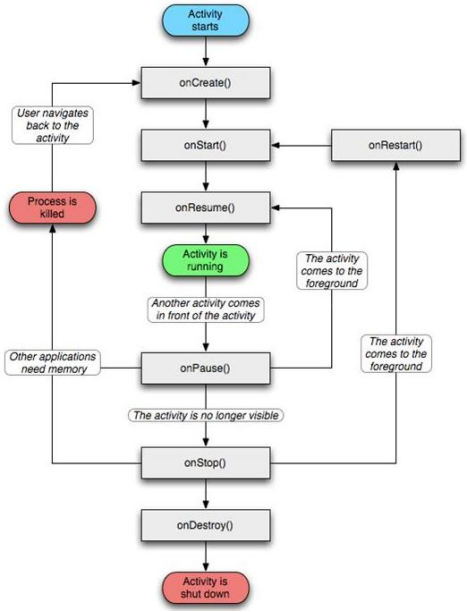


Analyis of Android Hybrid Applications



Analysis of Android Applications

	Impl.	Inter-App	Inter-Comp	Views	Lifecycle
DROIDEL[2]	WALA			✓	✓
SCANDROID	WALA	✓	✓		
FLOWDROID	Soot				✓
DROIDSAFE	Soot		✓	✓	✓
GATOR	Soot		✓	✓	✓
SMARTDROID	hybrid		✓	✓	
COMDROID			✓		
EPICC	Soot	✓	✓		
APPOSCOPY	Soot		✓		



- ScanDal, CHEX, LeakMiner, AndroidLeaks, ···
- TaintDroid, CopperDroid, Aurasium, DroidScope, ···

[2] "Droidel: A General Approach to Android Framework Modeling" Sam Blackshear, Alexandra Gendreau, and Bor-Yuh Evan Chang.
`https://www.cs.colorado.edu/%7Esabl4745/papers/droidel.pdf`
`https://github.com/cuplv/droidel`

Analysis of Android Hybrid Applications

- Implicit inter-language control flows
- Different types, values, and semantics in 2 languages
- Lack of documentation

Analysis of Android Hybrid Applications

- Implicit inter-language control flows
  - Android semantics by Droidel and other tools
  - Hybrid semantics by modeling inter-language flows
- Different types, values, and semantics in 2 languages
  - WALA's cross language support
  - Pointer-based analysis for Java & field-based for JS
- Lack of documentation
  - Web docs, blogs, Dalvik VM code, and experiments

Implicit Inter-Language Control Flows

Android Java ⇒ JavaScript

- `WebView.loadUrl("javascript:request();")`
- `WebView.loadUrl` is usually for loading a given URL.
- When the prefix of a string argument of `WebView.loadUrl` is "`javascript:`", it acts like the `eval` function.

Implicit Inter-Language Control Flows

JavaScript $\Rightarrow$ Android Java

- `WebViewClient.shouldOverrideUrlLoading`
- `WebChromeClient.onJsPrompt`
- `WebView.addJavascriptInterface`

(from hybrid applications developed in the Cordova framework)

Type Compatibility (by Experiments)

JavaScript ⇒ Android Java: function argument types

	int	float	String	boolean	Object	Array
Null	✗(null)	✗(null)	✗(null)	✗(null)	✗(null)	✗(null)
Undefined	✗	✗	✗("undefined")	✗	✗	✗
Number	✓	✓	✓(type conversion)	✗(false)	✗(null)	✗(null)
Boolean	✗(0)	✗(0)	✓(type conversion)	✓	✗(null)	✗(null)
String	✗(0)	✗(0)	✓	✗(false)	✗(null)	✗(null)
Object	✗(0)	✗(0)	✗("undefined")	✗(false)	✗(null)	✗(null)
Array	✗(0)	✗(0)	✗("undefined")	✗(false)	✗(null)	✤

✤ = ✓ if the Array element type is one of primitive types;
null if the Array element type is Object;
0 if the Array element type is int or float;
false if the Array element type is boolean; or
"undefined" if the Array element type is String.

Type Compatibility (by Experiments)

Android Java ⇒ JavaScript: function return types

	`int`	`float`	`String`	`boolean`	`Object`	`Array`
JavaScript	✓	✓(inexact)	✓	✓	✗(`{}`)	✗(`undefined`)

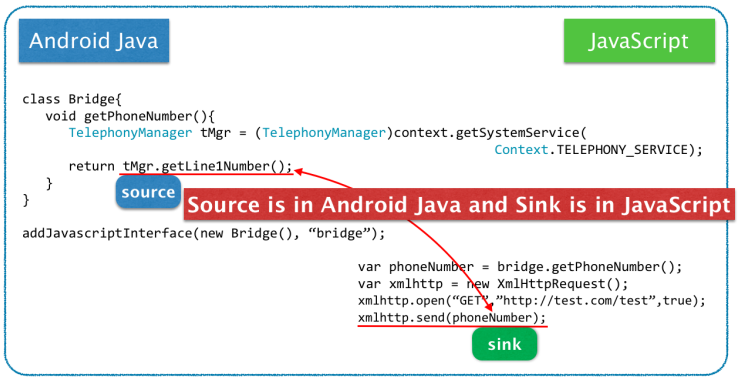
HybriDroid

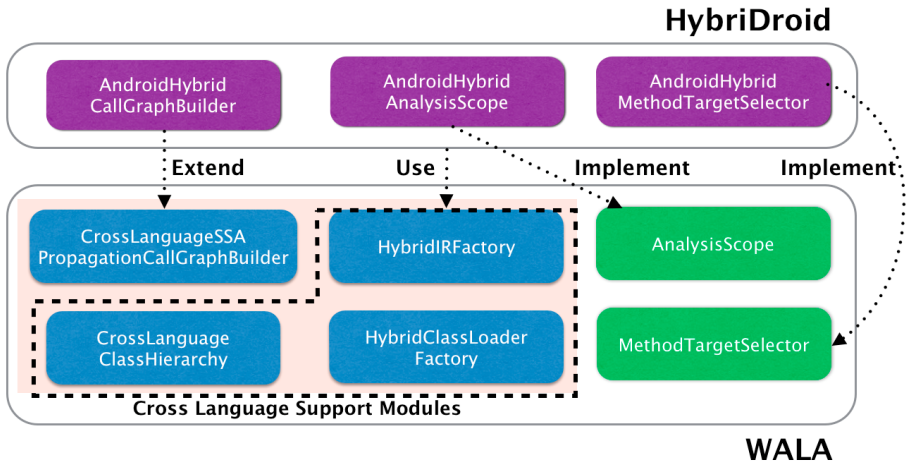
- *Soundy* analysis framework for Android hybrid applications
- Support for partial but most implicit inter-language flows backed by APIs, blogs, and Dalvik VM source code
- Support for partial but most type compatibility backed by experiments with trials & errors
- Implementation on top of WALA

  `https://github.com/SunghoLee/WALA/tree/master/HybriDroid/src/kr/ac/kaist/hybridroid/callgraph`

HybriDroid

- First tool to analyze Android hybrid applications
- Prototype implementation of detecting API misuse and private data leakage
- Work in progress
  - Experiments with real-world hybrid applications
  - Support for more Android specific semantics





Questions ?

Sukyoung Ryu
sryu.cs@kaist.ac.kr