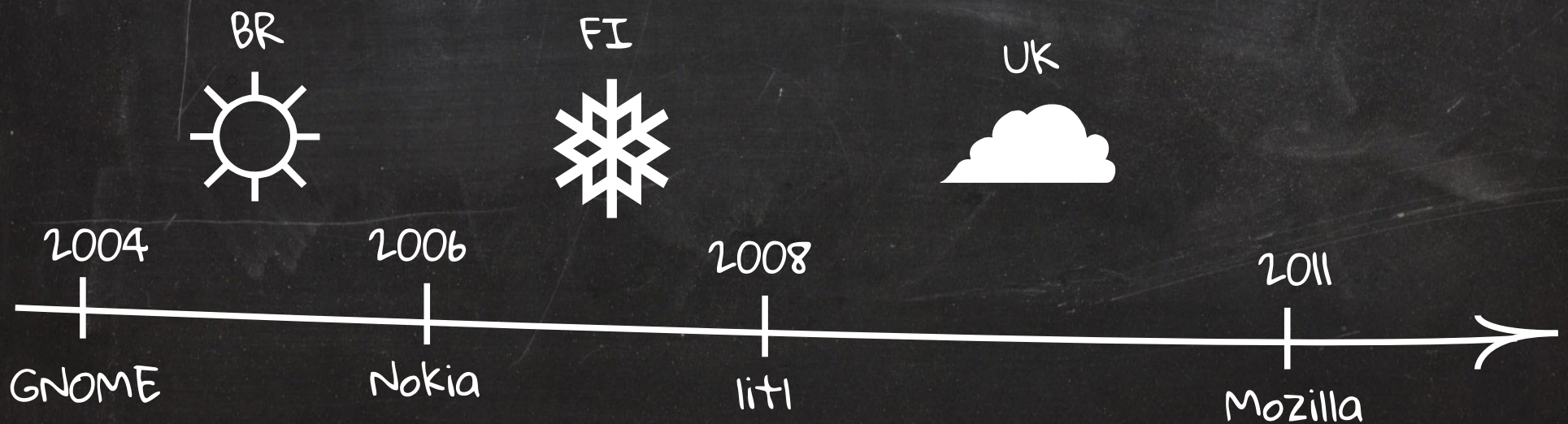


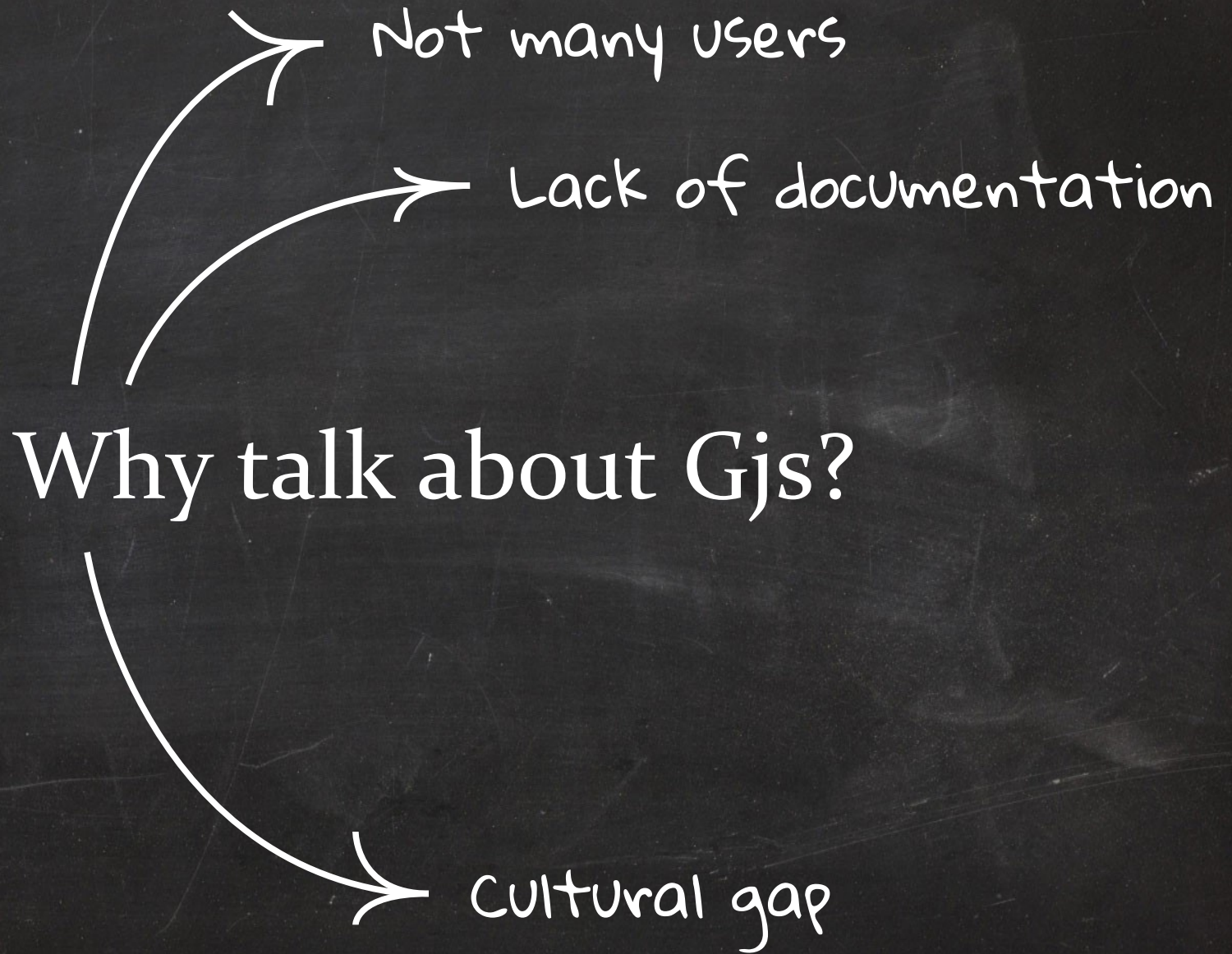
Thoughts on Gjs-based Development



Desktop Summit 2011

I am Lucas Rocha.





JavaScript is pretty cool!

JavaScript is misunderstood.

Java? What?

Bad code! Bad!

DOM API sucks!

Not reliable

Globals

Unexpected behaviour

It has bad parts.

No block scope

But it also has good parts.

Closures are central.


```
function fetchUrls(urls, callback) {
  var nUrls = urls.length;

  for (var i in urls) {
    doHttpGet(urls[i], function() {
      nUrls--;

      if (nUrls == 0) {
        callback();
      }
    });
  }
}
```

Functions are first-class objects.

```
function makeMultiplyFunction(times) {  
    return function(n) {  
        return n * times;  
    };  
}
```

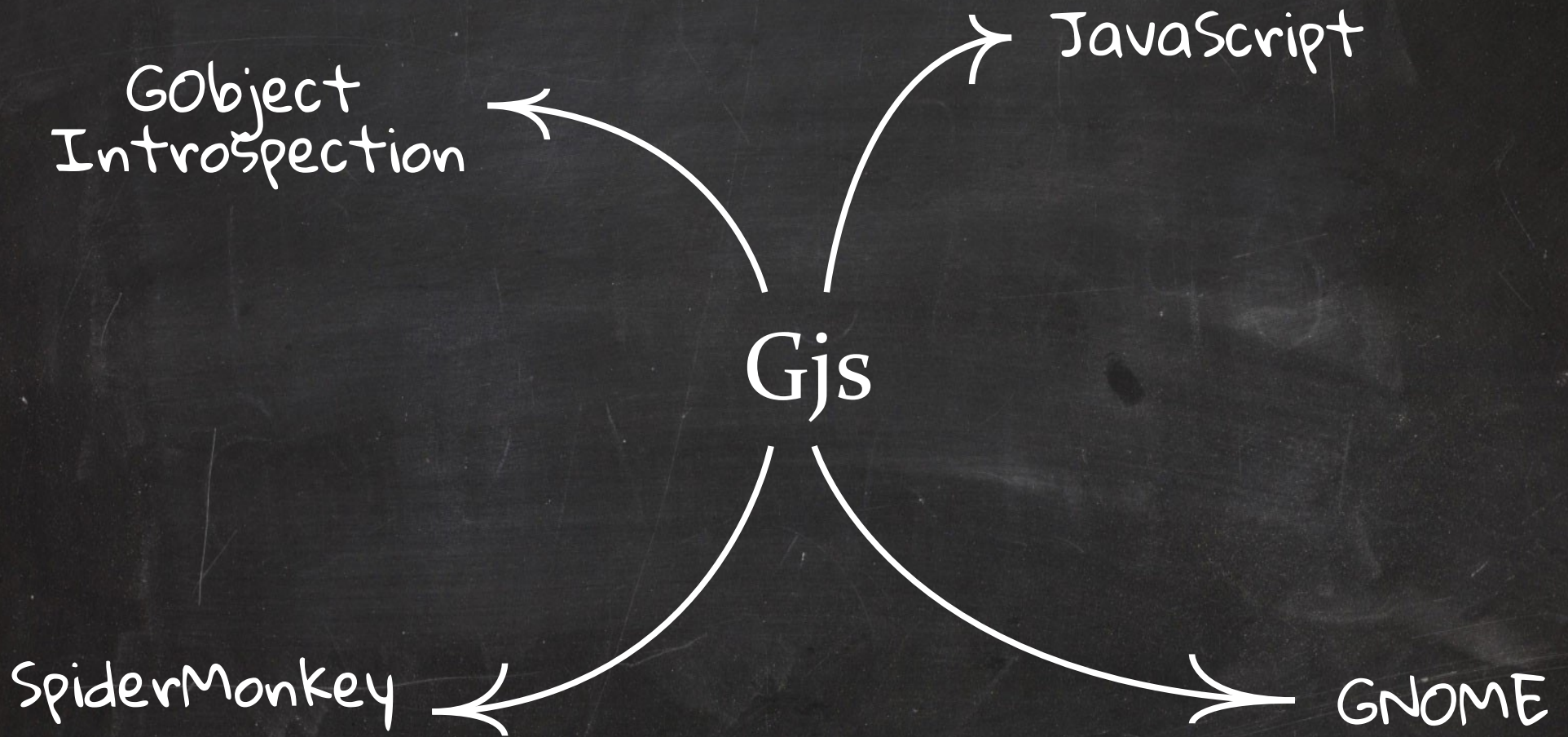
```
var multiplyByTwo = makeMultiplyFunction(2);  
print(multiplyByTwo(5)); // Prints 10
```

```
var multiplyByThree = makeMultiplyFunction(3);  
print(multiplyByThree(5)); // Prints 15
```

Prototypal inheritance.

```
var foo = {  
    one: 1,  
    two: 2  
};  
  
var bar = Object.create(foo);  
  
bar.three = 3;  
  
print(bar.one); // Prints 1  
print(bar.two); // Prints 2  
print(bar.three); // Prints 3
```

JavaScript is light, fast, everywhere.



Created by Havoc.

First released in 2008.



Well maintained.

Red Hat

Community



main.js

```
const Gtk = imports.gi.Gtk;  
const App = imports.app;
```

```
Gtk.init(0, null);
```

```
let win =  
    new Gtk.Window({ type: Gtk.WindowType.TOPLEVEL });
```

```
win.set_border_width(10);
```

```
function onDestroy(widget) {  
    Gtk.main_quit();  
}
```

```
win.connect('destroy', onDestroy);
```

```
...
```

main.js

```
...  
  
let model = new App.Model({ name: 'Lucas' });  
model.connect("name-changed", function() {  
    print('Name changed!');  
});  
  
let button = new Gtk.Button({ label: "Hello World" });  
button.connect("clicked", function() {  
    model.setName('Rocha');  
});  
  
win.add(button);  
  
button.show();  
win.show();  
  
Gtk.main();
```

app.js

```
const Signals = imports.signals;
```

```
function Model(args) {  
  this._init(args);  
}
```

```
Model.prototype = {  
  _init: function(args) {  
    this._name = args.name;  
  },  
  
  setName: function(name) {  
    this._name = name;  
    this.emit('name-changed');  
  }  
};
```

```
Signals.addSignalMethods(Model.prototype);
```

Running your app



```
$ ls  
main.js  
modules/
```

```
$ ls modules/  
app.js
```

```
$ gjs main.js
```

The “this” keyword

```
function Thing(foo) {  
  this._init(foo);  
}
```

```
Thing.prototype = {  
  _init: function(foo) {  
    this._bar = "Bar";  
  
    foo.connect("some-signal", function() {  
      print(this._bar);  
    });  
  },  
};
```



```
const Lang = imports.lang;
```

```
function Thing(foo) {  
  this._init(foo);  
}
```

```
Thing.prototype = {  
  _init: function(foo) {  
    this._bar = "Bar";  
  
    foo.connect("some-signal",  
                Lang.bind(this, this._onSomeSignal));  
  },  
  
  onSomeSignal: function() {  
    print(this._bar);  
  }  
};
```

Tween it!

```
...  
  
let stage = new Clutter.Stage();  
let texture = new ClutterTexture({ filename: 'img.jpg' });  
  
stage.add_actor(texture);  
  
let tweenArgs = {  
  x: 10,  
  y: 10  
  time: 500,  
  onComplete: function() { print('Done!') }  
};  
  
Tweener.addTween(texture, tweenArgs);  
  
...
```

D-Bus & GVariant

Built-in bindings.

Cairo

Nothing is perfect...

→ API coverage not 100%

→ No GType inheritance

That's all! Any questions?

lucasr.org

Find me here!

