

Production Ready JavaScript

Elliott Sprehn (Google, Inc.)

esprehn@gmail.com

<http://www.elliotsprehn.com/>

Apps of the future, apps of the past!?

How much
JavaScript?

Apps of the future, apps of the past!?

1 000 lines?

Apps of the future, apps of the past!?

10k lines?

Apps of the future, apps of the past!?

Google Feedback
is 15k lines.

Apps of the future, apps of the past!?

100k lines?

Apps of the future, apps of the past!?

Gmail is over
400k lines.

Code reusability

- Use a library (or several)
- Keep it modular
- Be consistent

Make sure it's tested

- Unit tests
 - JsTestDriver, QUnit, ...
- End to End tests
 - WebDriver, PhantomJS

Seriously write some tests

Tests!!

Post processing

- Concatenation
- Minification
 - YUI Compressor
 - **Uglify**
- Compilation
 - Closure Compiler

Dependency management

- `goog.require` / `goog.provide`
- `dojo.require` (< 1.7)
- AMD

Google Closure

```
goog.provide('ns.tools');
```

```
goog.require('goog.string');
```

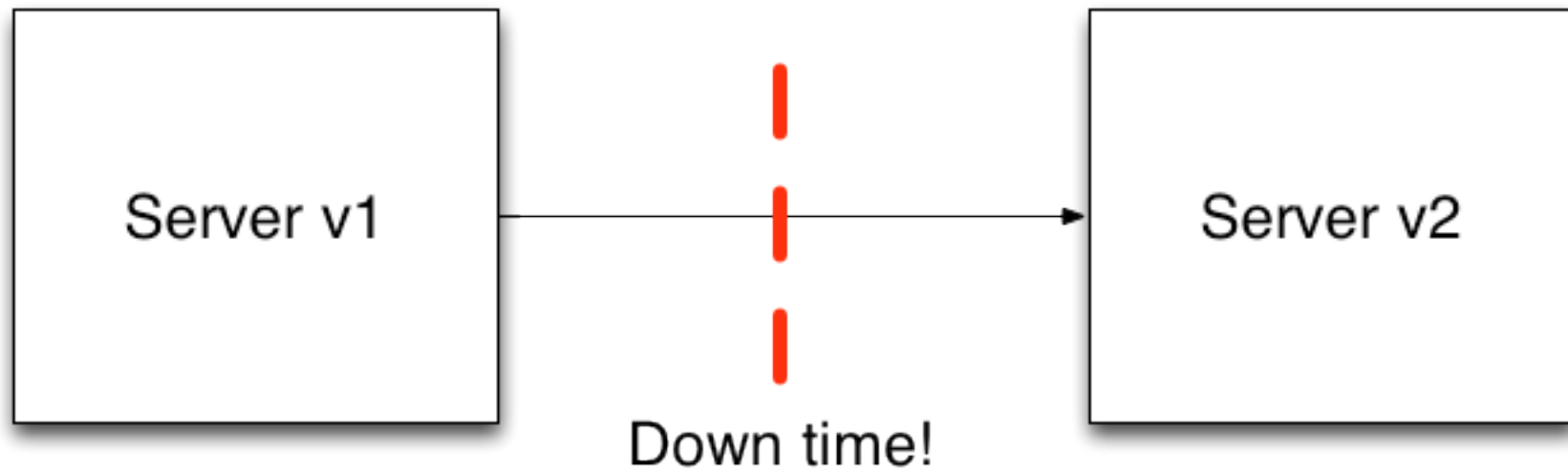
```
ns.tools.methodName = function(value) {  
  return goog.string.collapseWhitespace(value);  
};
```

AMD

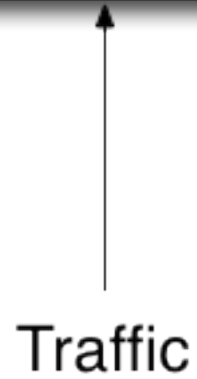
```
define(['exports', 'tools'], function(exports, tools) {  
  exports.method = function(value) {  
    return tools.transform(value);  
  };  
});
```

```
define(function (require, exports) {  
  var tools = require('tools');  
  exports.method = function(value) {  
    return tools.transform(value);  
  };  
});
```

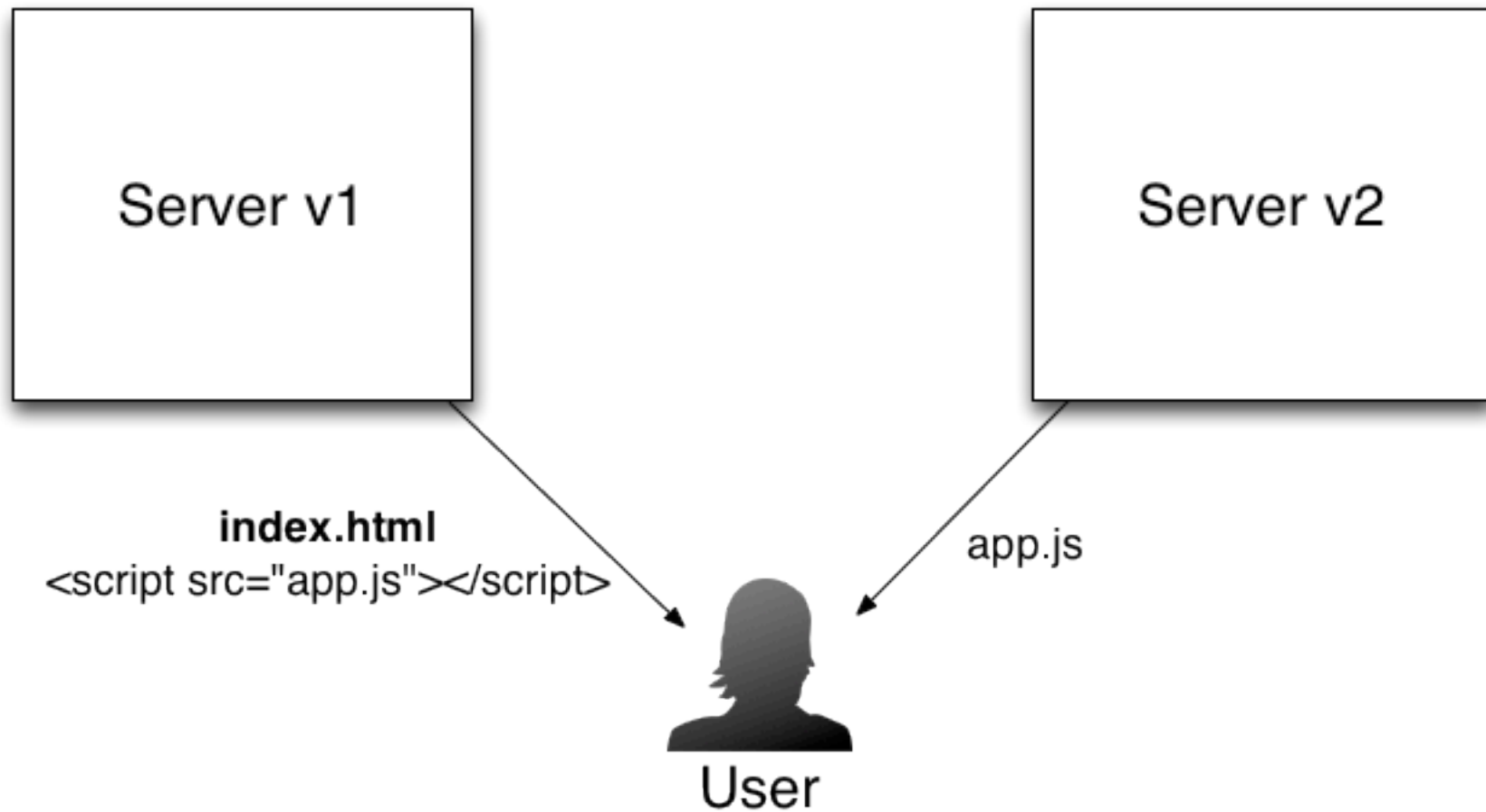
Deployment



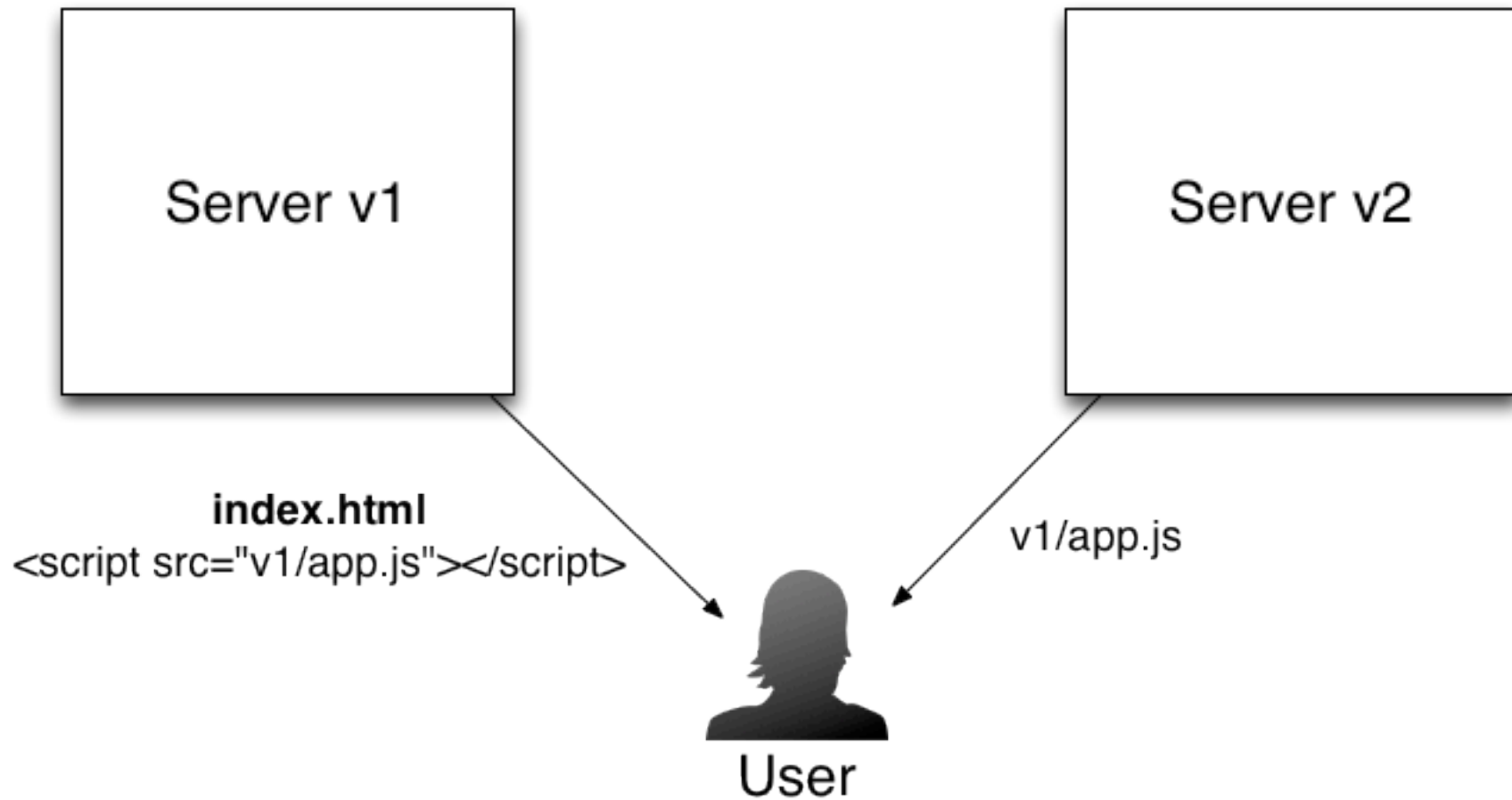
Deployment



Deployment



Deployment



Versioning

- Version by release
 - /js/v12/feedback.js
- Version by hash
 - /js/297a34d_feedback.js

XHR APIs

- Keep it backwards compatible
- Extend, then remove
- Don't explicitly version

Extend APIs

Version 1

```
{  
  "email": "example@gmail.com",  
  "name": "Elliott Sprehn"  
}
```

Extend APIs

Version 2

```
{  
  "email": "example@gmail.com",  
  "name": "Elliott Sprehn"  
  "firstName": "Elliott"  
  "lastName": "Sprehn"  
}
```

Extend APIs

Version 3

```
{  
  "email": "example@gmail.com",  
  "firstName": "Elliott"  
  "lastName": "Sprehn"  
}
```

Debugging and monitoring

- Logging
 - console.log / error
- Instrumentation
 - Events (user actions, errors)
 - Timing

Instrumentation

```
function createScreenshot() {  
  var timer = new Timer('screenshot');  
  try {  
    // ...  
    timer.tick('success');  
  } catch (e) {  
    timer.tick('error', e);  
  } finally {  
    timer.report();  
  }  
}
```

Instrumentation

```
var events = new UserEventReporter();
```

```
window.onload = function() {  
  events.report('load');  
};
```

```
function showPhotos(...) {  
  events.report('photos');  
}
```

Error Reporting

- Send stack traces to server
- *window.onerror*

Performance

- GZIP compression
- Public caching
- Bundle resources

Questions

