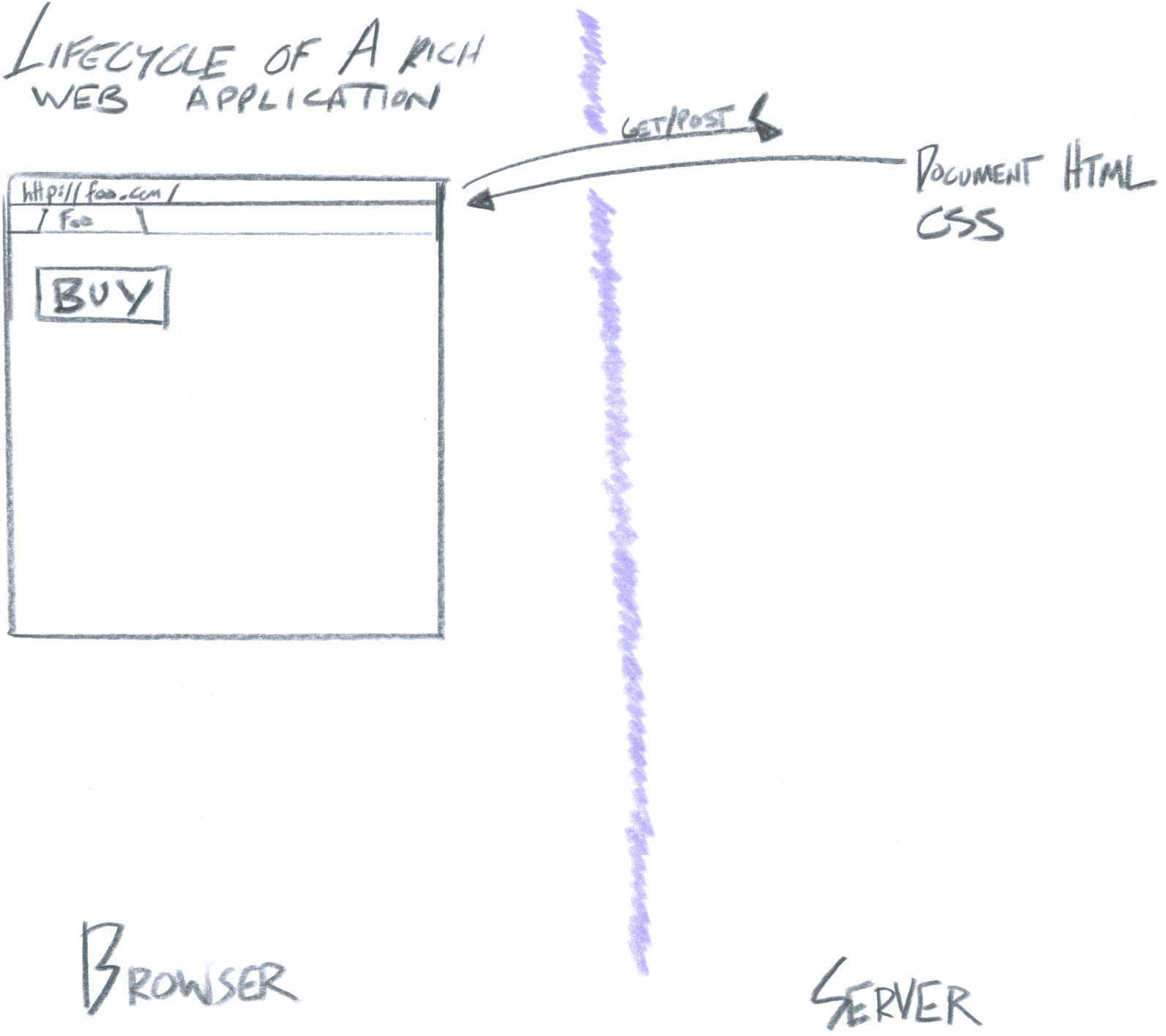
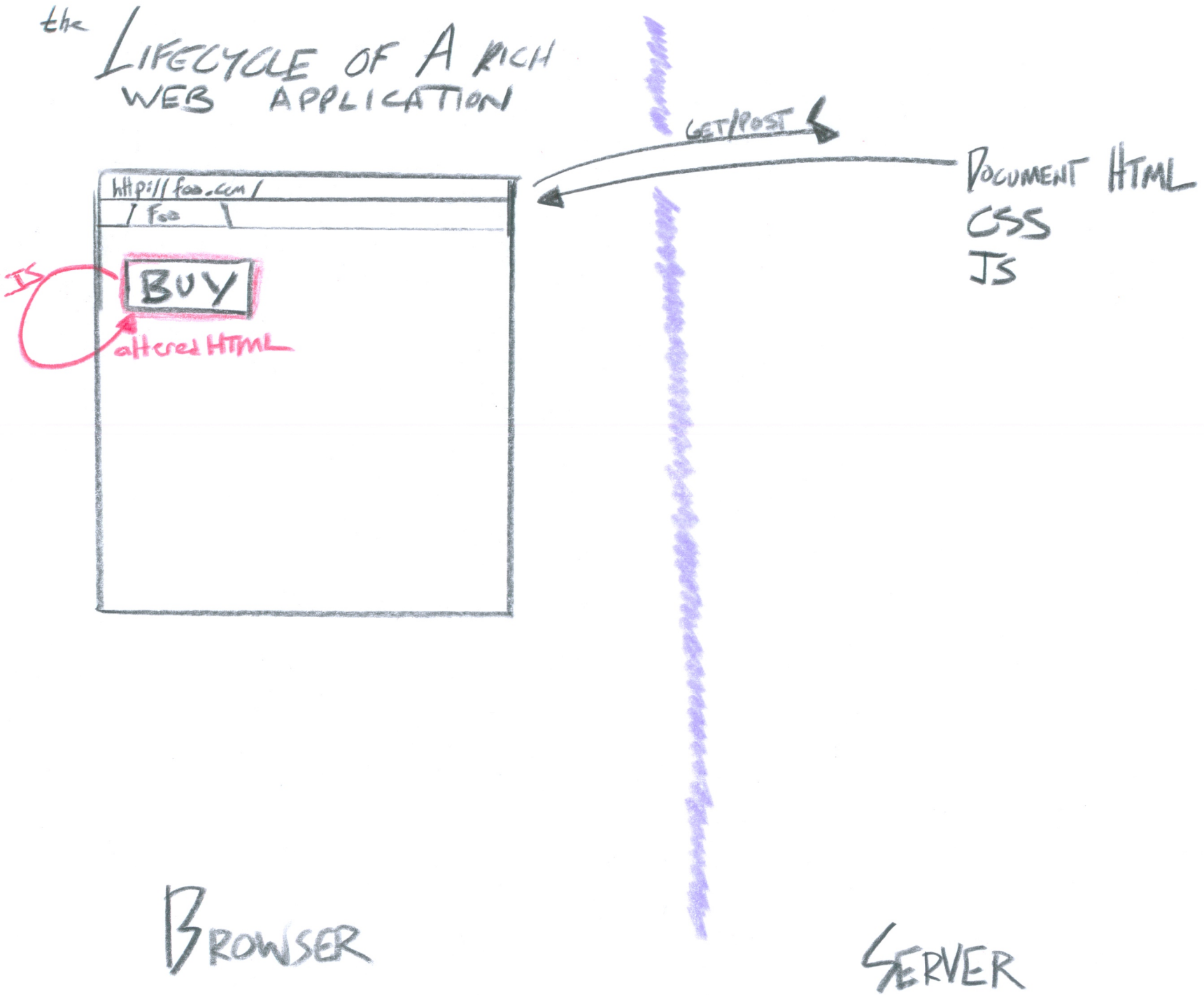
Rich web applications & Google Closure

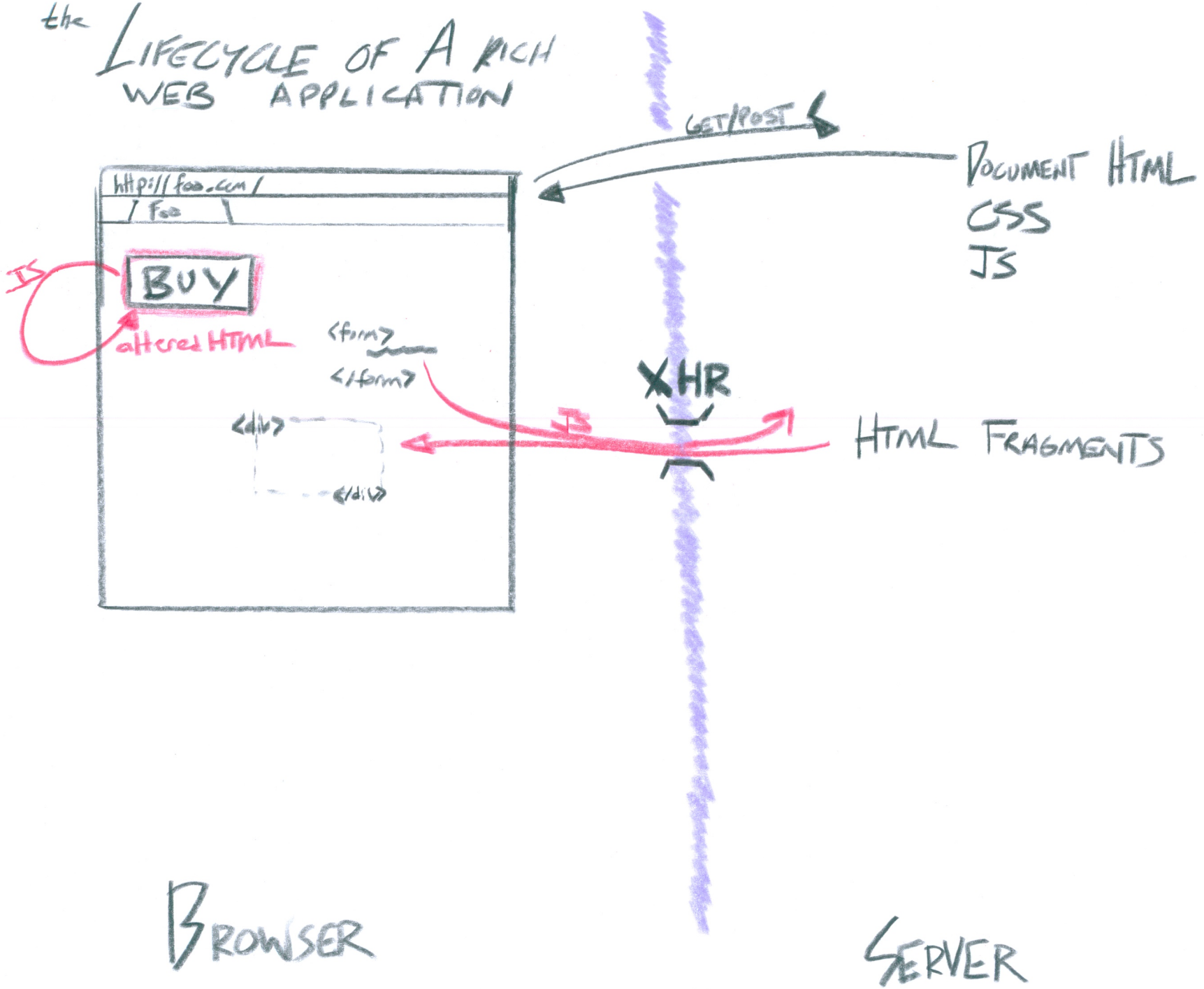
Hunter Blanks, hunter@twilio.com / @tildehblanks
github.com/hblanks/talks/

Twilio. April 26, 2011

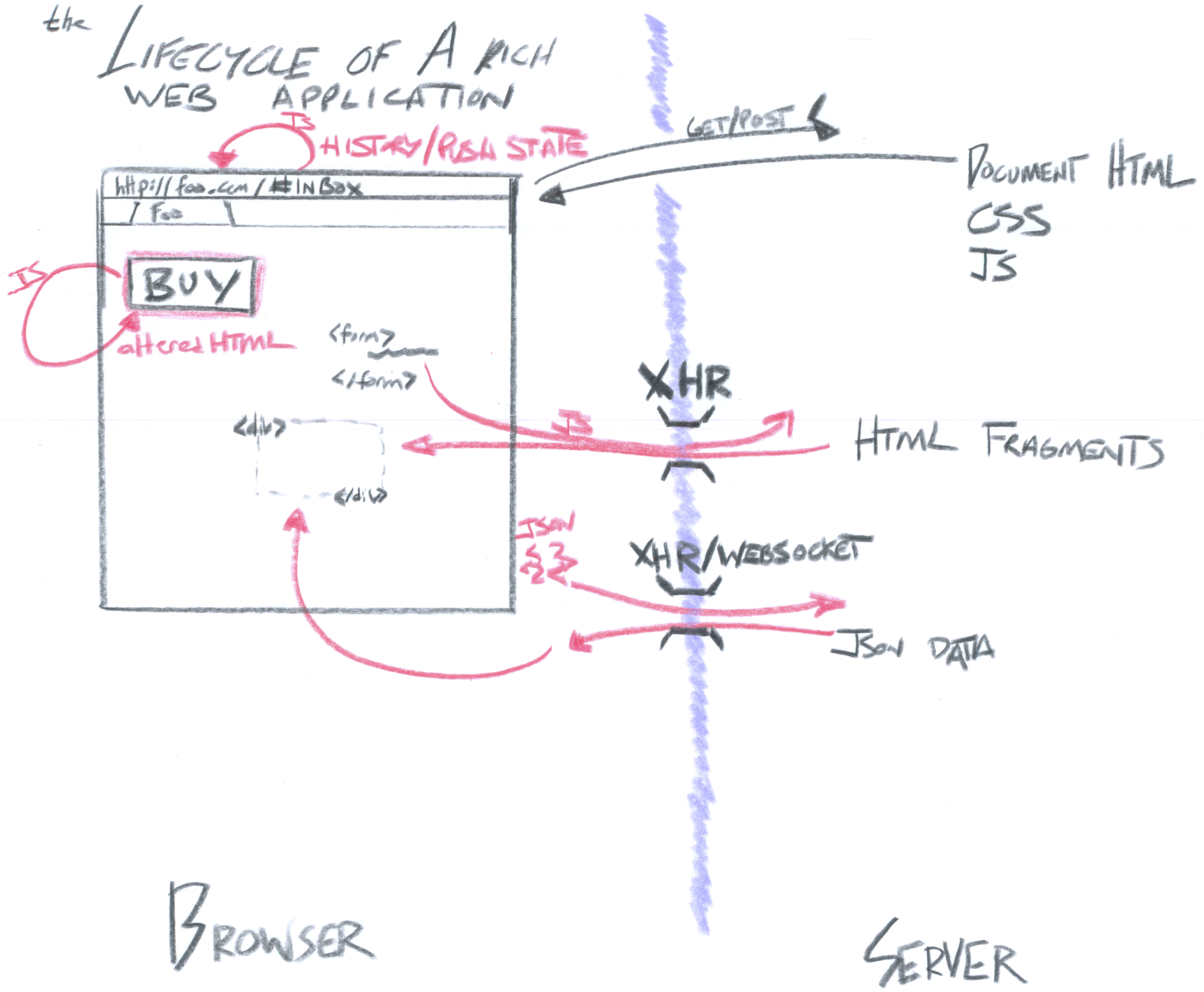
the LIFECYCLE OF A RICH WEB APPLICATION



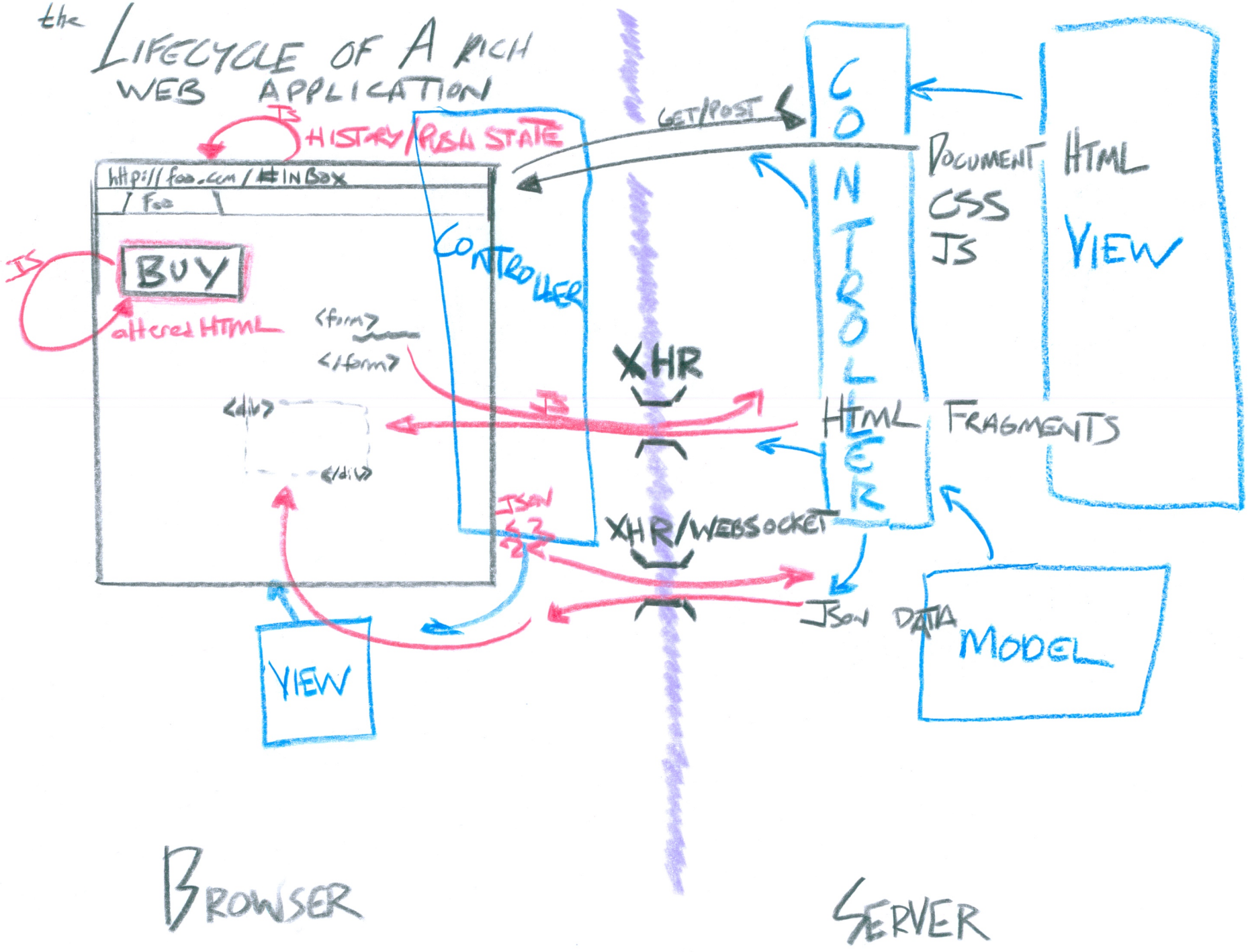

the LIFECYCLE OF A RICH WEB APPLICATION
GET/POST
DOCUMENT HTML
CSS
JS
http://foo.com/
/ Foo
BUY
JS
altered HTML
BROWSER
SERVER


the LIFECYCLE OF A RICH WEB APPLICATION
GET/POST
DOCUMENT HTML
CSS
JS
http://foo.com/
/ Foo
BUY
JS
altered HTML
<form>
</form>
XHR
JS

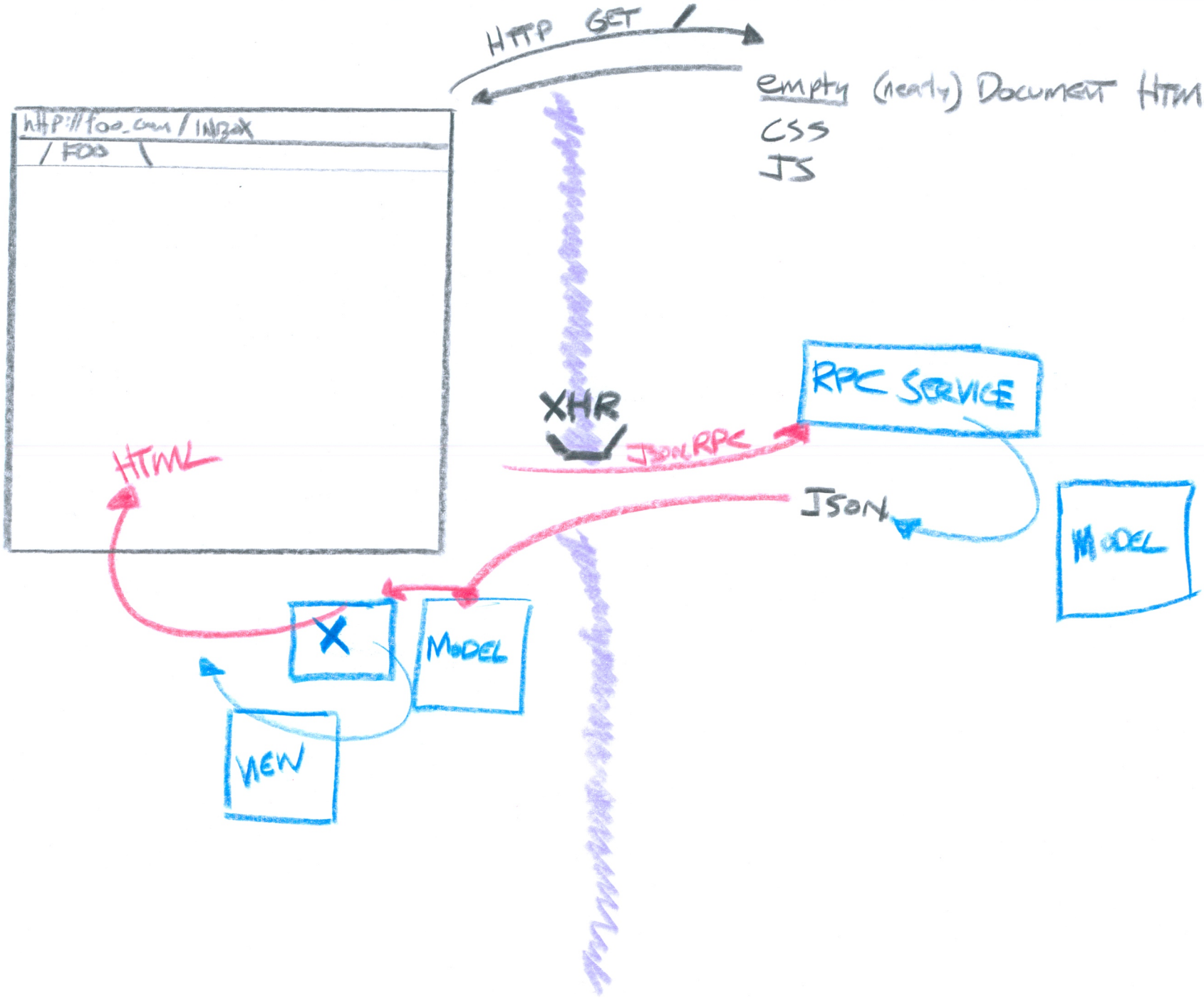
HTML FRAGMENTS
BROWSER
SERVER


the LIFECYCLE OF A RICH WEB APPLICATION
JS
HISTORY/RSH STATE
GET/POST
DOCUMENT HTML
CSS
JS
http://foo.com/#INBOX
Foo
BUY
JS
altered HTML
<form>
</form>
XHR
JS
HTML FRAGMENTS

JSON
XHR/WEBSOCKET
JSON DATA
BROWSER
SERVER


the LIFECYCLE OF A RICH WEB APPLICATION
HISTORY/RSLA STATE
http://foo.com/#InBox
/ Foo
BUY
altered HTML
CONTROLLER
<form>
</form>

JS
XHR
JSON
XHR/WEBSOCKET
VIEW
GET/POST
CONTROLLER
DOCUMENT
CSS
JS
HTML
VIEW
HTML FRAGMENTS
JSON DATA
MODEL
BROWSER
SERVER


HTTP GET /
empty (nearly) Document HTML
CSS
JS
http://foo.com/inbox
/ foo /
XHR
JSON RPC
RPC SERVICE
HTML
JSON
MODEL
X
MODEL
VIEW

An incomplete survey of your tools and options

- Backbone.js

- AngularJS

- Google Web Toolkit

- Cappucino

- SproutCore

- (roll your own) with jQuery

- (roll your own) with CoffeeScript

- (roll your own) with Closure Tools

“Something that has been ignored in [previous] responses is the fact that Closure and jQuery are totally different beasts. jQuery is the industry standard way of writing nice, user friendly web pages and user controls. Closure is an application development framework.

Closure comes with a hell of a lot of overheads that are only worthwhile after a certain scale is reached.

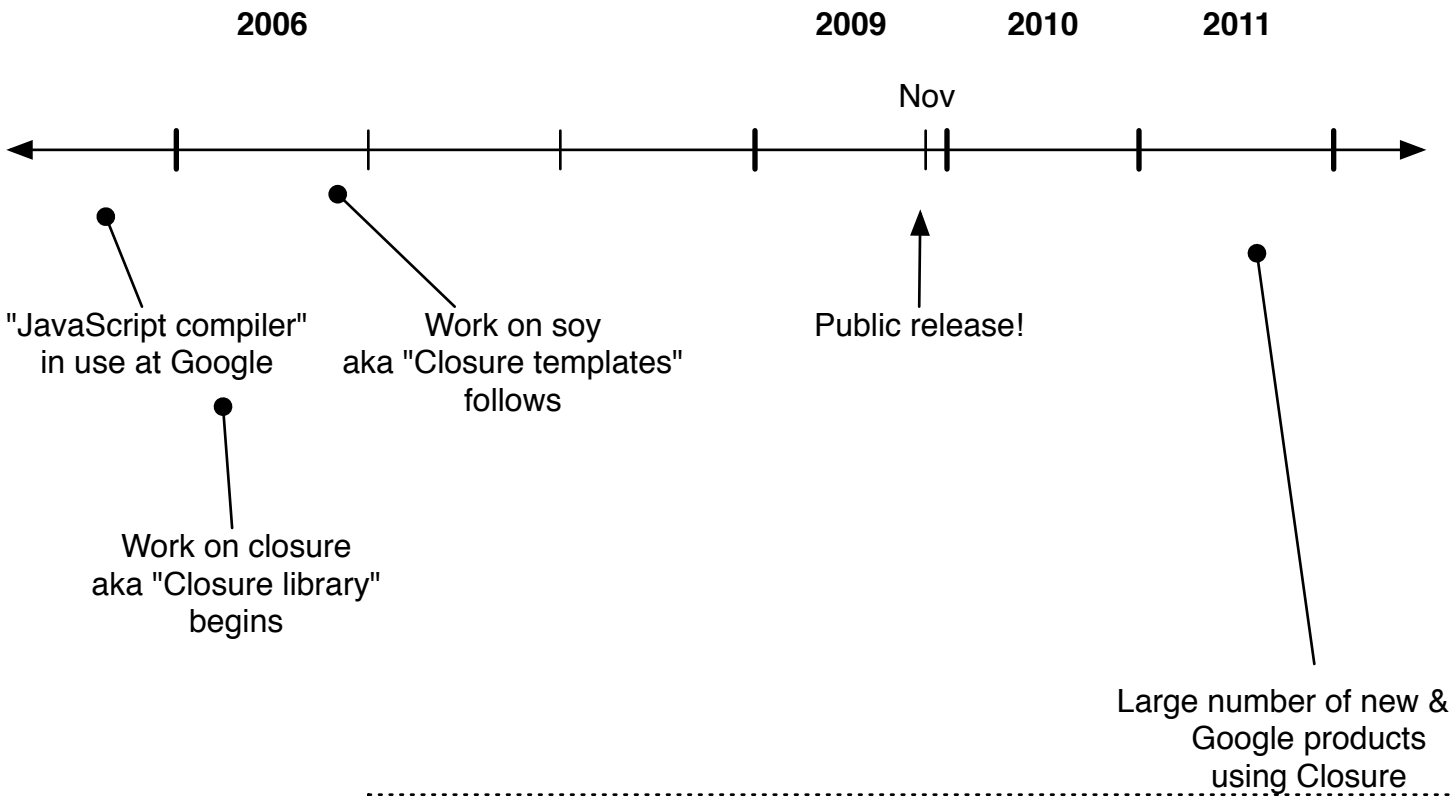
However try writing a complex system (say... 30k+ LOC) in jQuery and you'll feel pain unlike you've never imagined. This is where closure comes into its own (you can always then bring jQuery into a closure project also).

So different tools for different jobs. Both with great support and strong development.”

–Guido Tapia

http://groups.google.com/group/closure-library-discuss/browse_thread/thread/498d8fb44957ff75

An outsider's timeline of Google Closure



Who uses Closure Library?

The Closure Library serves as the base JavaScript library for many Google products, including:

Gmail Maps Docs Sites Books Reader Blogger Calendar Google+ Photos

(http://code.google.com/closure/library/)

What is it?

1) **Closure Compiler**. A JavaScript -> Javascript compiler. Key features are:

- dead code elimination
- type checking
- numerous optimizations
- warnings/errors on unsafe code

2) **Closure Library**. A giant, heavily documented, cross-platform JS library.
Key features are:

- cross browser event framework
- many, many UI components
- Gmail's rich text editor
- cross-page & server communication libraries
- canvas/SVG/VML graphics libraries
- numerous other libraries (e.g. goog.array, goog.structs.Map, etc.)

3) **Closure templates**. A JS & Java template language/compiler.
Key features are:

- HTML structure can be kept separate from your JavaScript
- you find out about syntax errors and unused varables at compile time!

Highlights from using the compiler

1) The compiler **automatically finds and links in namespaces**, letting you easily build libraries for common code, and naturally split out code into individual files and directories.

```
/** @fileoverview Foo.com lightbox library. */
goog.provide('foo.Lightbox');

goog.require('goog.style');
goog.require('foo.dom');
goog.require('foo.events');
```

2) **JSDoc annotations**, together with the Closure compiler, make code more explicit and help you catch *more bugs at compile time.*

```
/**
 * @constructor
 * @extends {goog.Disposable}
 * @param {number} width Width of lightbox.
 * @param {number} height Height of lightbox.
 */
foo.Lightbox = function(width, height) {
    this.width_ = width;
    ...
};
```

3) Among other things, Closure's **advanced optimizations remove functions that are never called and inline others**. This is key, since for any website, small (<16KB) script payload sizes are best.

Highlights from using templates (aka soy)

1) Templates are **compiled**. So you find out about syntax errors at compile time instead of runtime.

```
Command failed: '(cd soy && java -jar /Users/hblanks/personal/files/arma/cloister/templates/
SoyToJsSrcCompiler.jar --shouldProvideRequireSoyNamespaces --shouldGenerateJsdoc --
outputPathFormat "/Users/hblanks/personal/git/arma/build/soy/{INPUT_DIRECTORY}/
{INPUT_FILE_NAME_NO_EXT}.js" foo.soy textril-arma-templates.soy)' returned 1

Exception in thread "main" com.google.template.soy.base.SoySyntaxException: In file foo.soy:
Lexical error at line 8, column 0.  Encountered: <EOF> after : ""
```

2) Templates **also have JSDoc annotations**, and all inputs are explicit! So you always know what variables you should go into your template (and the template compiler will complain if you try to use other ones, or don't use all the ones you specify).

```
/**
 * @param world
 */
{template .gar}hello {$worl}{/template}

Exception in thread "main" com.google.template.soy.base.SoySyntaxException: In file foo.soy:6,
template arma.templates.foo.gar: Found references to data keys that are not declared in
SoyDoc: [worl]
```

3) Templates can also be rendered in Java. (If for some reason you wanted to.)

Highlights from using the library

1) **Almost anything you ask for** has been implemented, and it's **heavily documented**.

```
goog.require('goog.ui.Editor');           // <- gmail's rich text editor
goog.require('goog.ui.HsvaPallette');     // <- arbitrary hex color (and alpha) picker
goog.require('goog.ui.DragListGroup');    // <- drag & drop UI component
goog.require('goog.events');              // <- Cross-platform event framework
goog.require('goog.events.KeyHandler');   // <- Cross-platform/OS key handler
goog.require('goog.net.BrowserChannel');  // <- Cross-platform websocket implementation
goog.require('goog.net.CrossDomainRpc');  // <- Cross-platform inter-page communication
```

Moreover, most things have been running in production for a long time, and are highly performant.

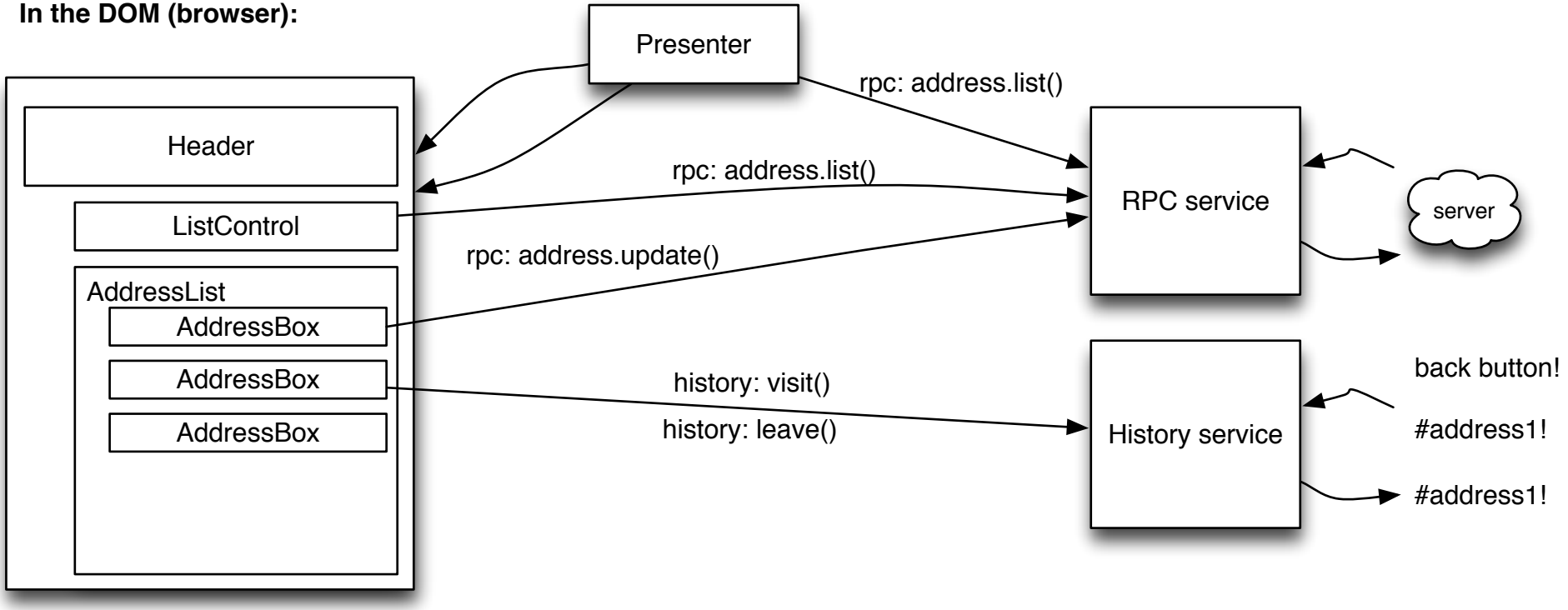
2) goog/base.js gives you **a real object model** with class-based inheritance.

```
/**
 * A custom button control.  Identical to {@link goog.ui.Button},
 * [[[ Lots of JSDOC was here ]]]
 */
goog.ui.CustomButton = function(content, opt_renderer, opt_domHelper) {
  goog.ui.Button.call(this, content, opt_renderer ||
      goog.ui.CustomButtonRenderer.getInstance(), opt_domHelper);
};
goog.inherits(goog.ui.CustomButton, goog.ui.Button);
```

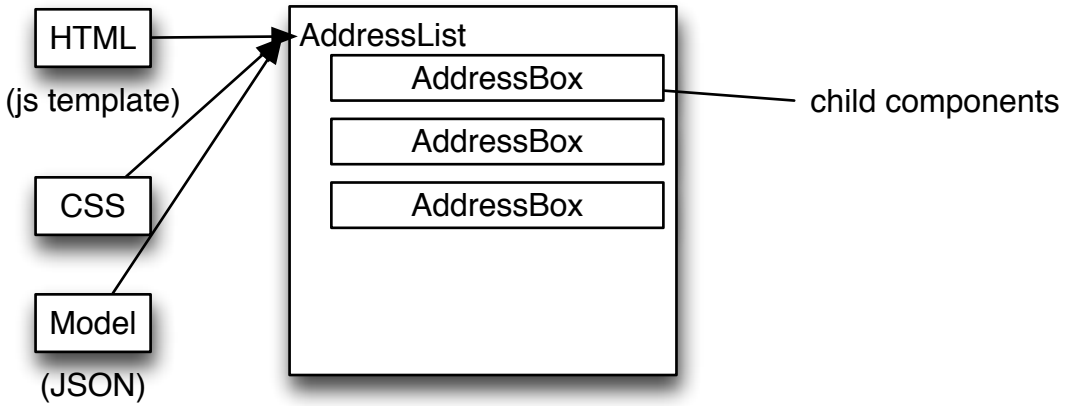
3) Once you understand it, **goog.ui.Component** is a great tool for building UI's.

How you might sketch out an addressbook app w/Closure

In the DOM (browser):



… and in your code (JS):



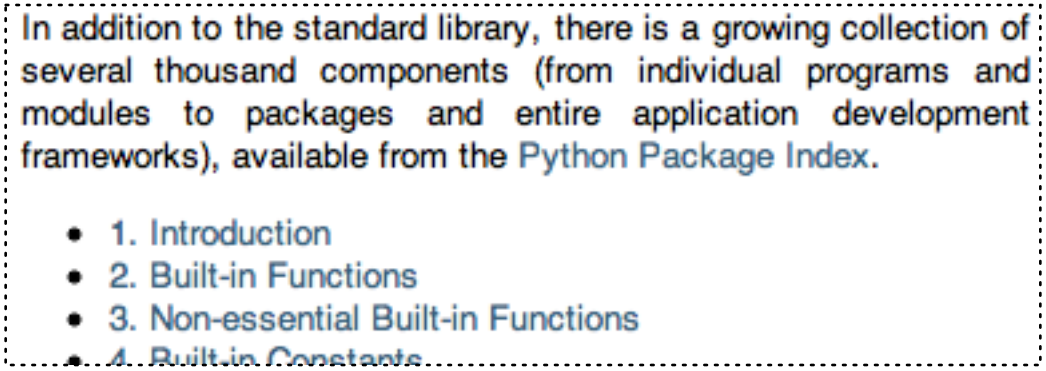
Remarks on the painful parts of using Closure, too

1) You're going to need a build system. It helps to support loading compiled scripts for production and uncompiled scripts (with your own deps.js file) for debugging. Options include

   - plovr
   - closure script
   - your own make/rake/jake/waf/ant(?)/Scons/...
   - ???

2) Documentation, although rich, is anti-compact. (This is not Python, people.) Michael Bolin's book, *Closure: The Definitive Guide*, is a real help, though.

≠



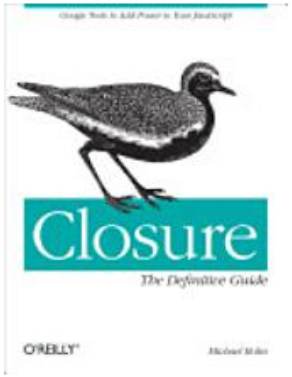
3) Closure library, although tied to Google's own version control and build system, has no versioned releases, only svn/trunk. So be ready for changes.

Where to get started

1) If you just want to try out the compiler, visit:

http://closure-compiler.appspot.com/home

2) If you want to work with the UI, buy this book:



Closure: The Definitive Guide. Michael Bolin. (O'Reilly)

Be sure to **read the Component chapter early**. Ray Ryan's talk on Best Practices for GWT is also a **very good video to watch**. (Yes, it is on GWT.)

3) For help getting started with the compiler & closurebuilder.py, see

http://code.google.com/closure/library/docs/closurebuilder.html
http://code.google.com/closure/compiler/docs/js-for-compiler.html

4) Or, come to the first Closure Library meetup on June 5th at Building 42!

Thank you!

Hunter Blanks, hunter@twilio.com / @tildehblanks
github.com/hblanks/talks/