stackoverflow Questions Tags Users Badges

Add a number to another number in JavaScript

0

hallo

I have got a number in my JavaScript variable! Now how do I add another number to it? Please

javascript

3 Answers

oldest

Add a number to another number in JavaScript

hallo

0

I have got a number in my JavaScript variable! Now how do I add another number to it? Please

javascript

3 Answers

oldest

You should definitely use jQuery. It's really great and does all things

22

link | edit | flag

answered

I<3

1,

I agree, jQuery is really the best, it solves all kinds of browser problems and is good, as well – ||sumc0d

+1 jquery is best quality code ever, if you don't use your a idiot – Werry_Togan 4 mins ago

add comment

I think there's a jQuery plugin for that. Google for jQuery basic arithmetic plugin.

answered

22

link | edit | flag

answered

I<3

1,2

I agree, jQuery is really the best, it solves all kinds of browser problems and is good, as well – ||sumc0d

+1 jquery is best quality code ever, if you don't use your a idiot – Werry_Togan 4 mins ago

add comment

I think there's a jQuery plugin for that. Google for jQuery basic arithmetic plugin.

4

link | edit | flag

answered

Tin

4,3

yeah, jQuery is definately the way to go – fishnipples 5 mins ago

I used the jQuery diet plugin and lost 10kg in a week – jfatty 4 mins ago

add comment

-2

To add numbers together you should use the + operator, for example:

```
var a= 1;
var b= a+2;
alert(b);  // 3
```

I think there's a jQuery plugin for that. Google for jQuery basic arithmetic plugin.

4

link | edit | flag

answered

Tim

4,

yeah, jQuery is definately the way to go – fishnipples 5 mins ago

I used the jQuery diet plugin and lost 10kg in a week – jfatty 4 mins ago

add comment

-2

To add numbers together you should use the + operator, for example:

```
var a= 1;
var b= a+2;
alert(b);  // 3
```

link | edit | delete | flag

answered

bo

so

-1 not enough jQuery – ||sumc0da 30 secs ago

you suck – Timothy Goatse 3 secs ago

http://www.doxdesk.com/updates/2009.html

Design, techniques and tools

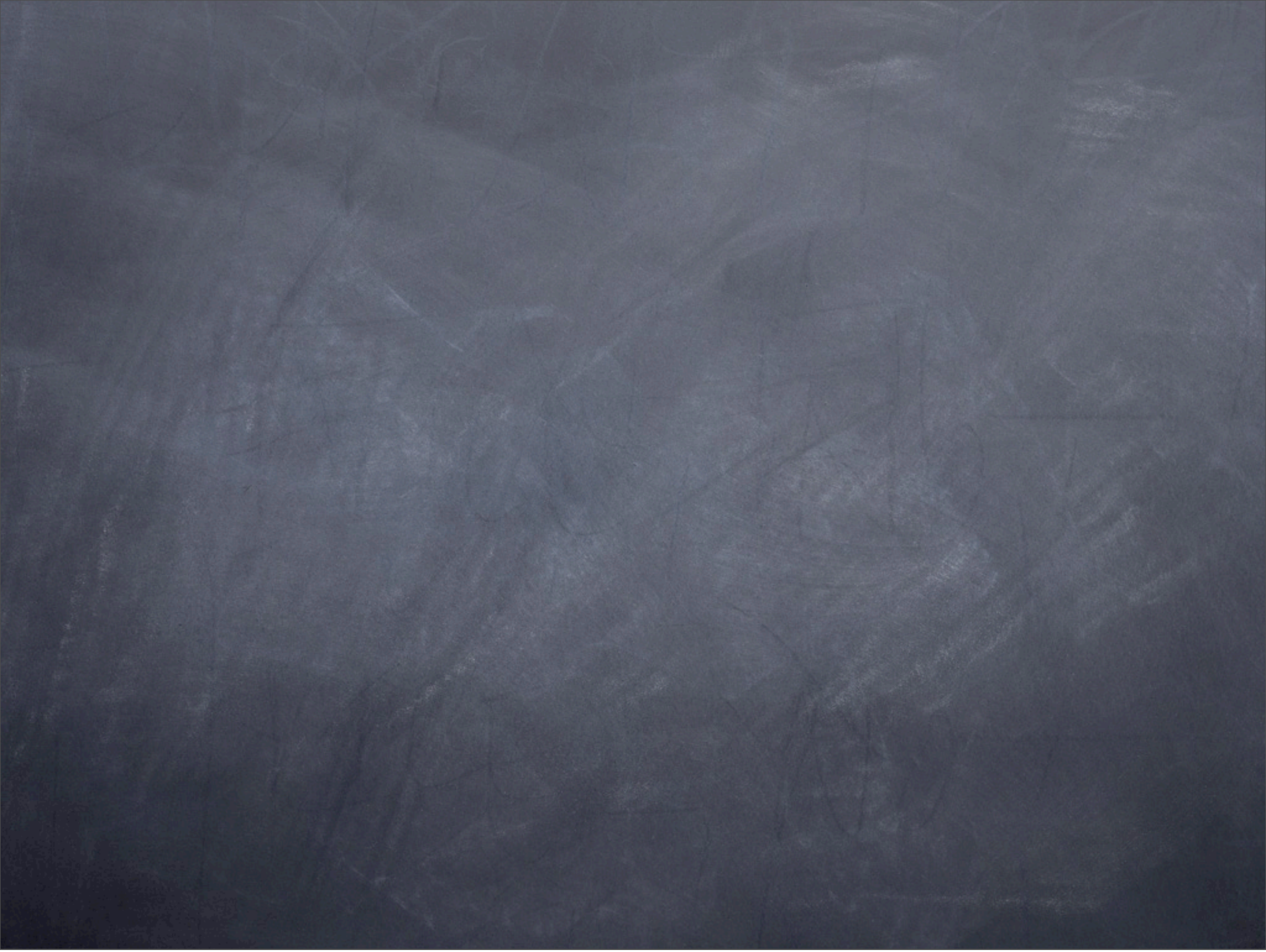
for larger JavaScript applications

```
Object.create = (function() {
    function F() {}
    return function(p) {
        F.prototype = p;
        return new F();
    };
})();
```

Karl Krukow (kkr@trifork.com),
Goto Amsterdam, Oct. 13th, 2011

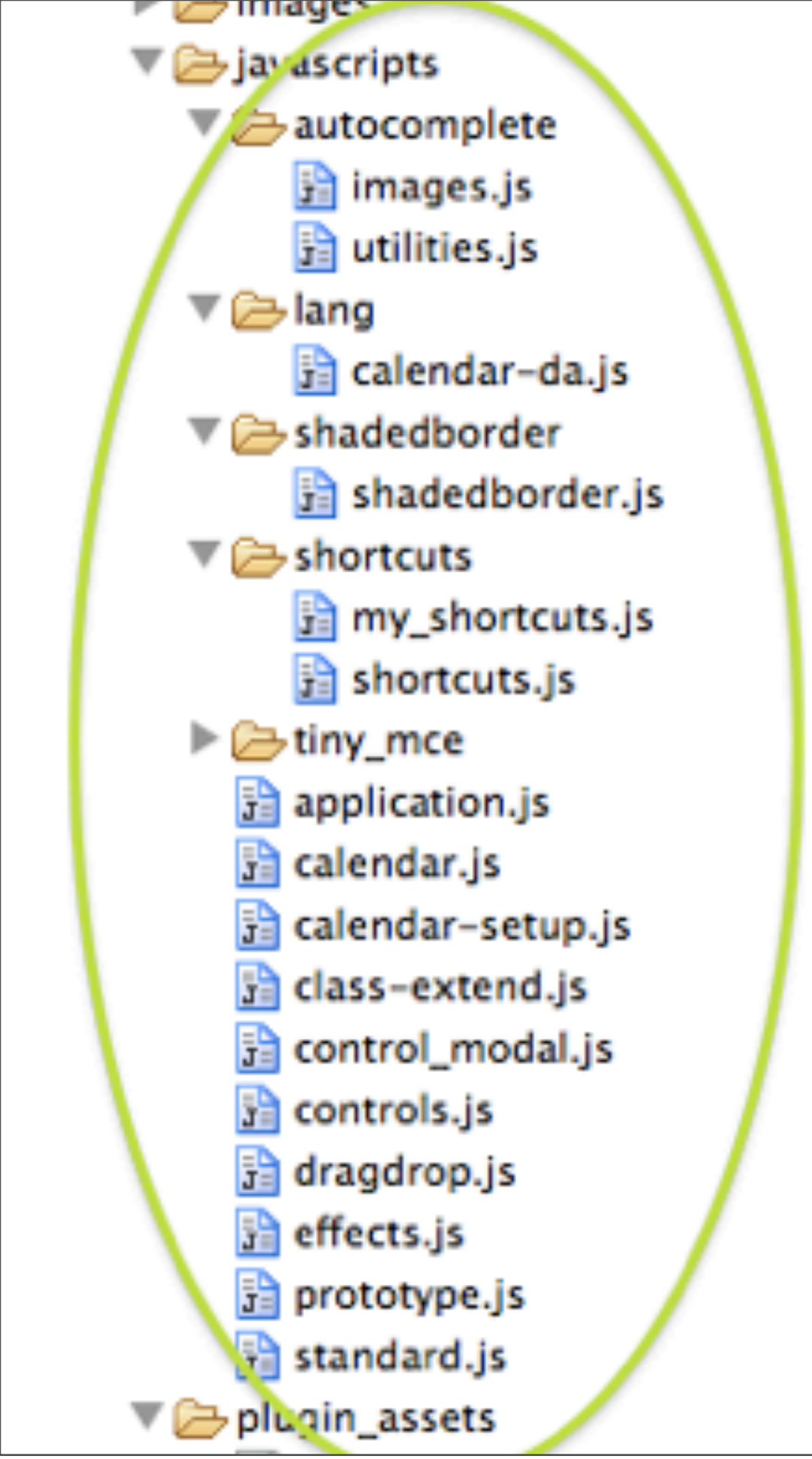
About me

- PhD, University of Aarhus, Theory-stuff :)
- Working at Trifork for about 5 years on Web, JavaScript, Java/JEE, Ruby/Rails, Clojure, Mobile, Conferences and Training.
- Last two years on iOS on Trifork in the financial sector.
- Recently part of a start-up doing mobile automated testing: LessPainful ApS (http://www.lesspainful.com)



Have you ever
found yourself
in a project like this...

Development Time




```
23      inputField  : "entry_event
24      ifFormat    : "%d-%m-%Y --
25      button      : "calendar-tr
26      showsTime          : "true"
27    }
28 );</script>
29 <br/>
30          <%= label_with_validatio
31          <%= text_area 'entry', '
32          <br />
33          <div style="float:right"
34              <%= submit_tag m(:so
35          </div>
36
37      </div>
38
39
40
41      <% end %>
42      </td>
43
44      </tr>
45 </table>
46
47 <script type="text/javascript" l
48      function upd() {
```

```
<script type="text/javascript">
Calendar.setup(
  {
    inputField  : "entry_event_occurred_at",
    ifFormat    : "%d-%m-%Y -- %H:%M",    // the date format
    button      : "calendar-trigger",
    showsTime         : "true"
  }
);</script>
<br/>
        <%= label_with_validation 'entry','contents', m(:new_note) %><br/>
        <%= text_area 'entry', 'contents',{:class=>"mceEditor", :value => pa
        <br />
        <div style="float:right">
            <%= submit_tag m(:save_entry) %>  
        </div>

    </div>



    <% end %>
    </td>

    </tr>
</table>
```

```
    <% end %>
    </td>

    </tr>
</table>

<script type="text/javascript" language="javascript">
    function upd() {
        selected_people = new Array();
            var textElement = $('persons_persons');
            textElement.value.split(',').each(function(e) {
                e = e.strip().toLowerCase();
                var id = person_name_2_id[e];
                if (id != null) {
                        selected_people[id] = id;
                }
            });
            synchronizeImages(selected_people);
    }
    upd();
    Event.observe('persons_persons', 'keyup', upd);
    var now = new Date();
</script>
```

Development Time

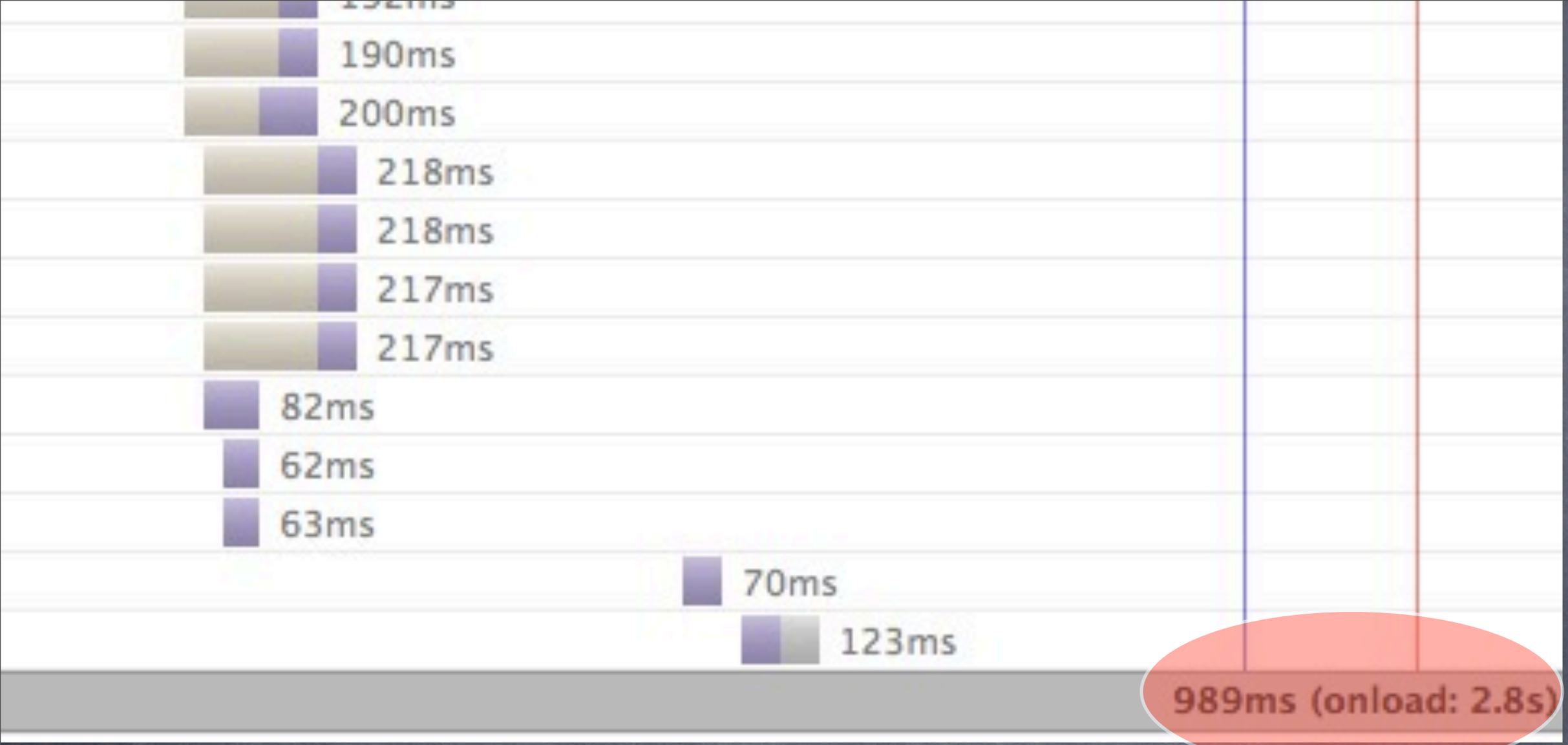
Unstructured
No clear architecture
No consistency

Runtime

Console HTML CSS Script DOM Net

Clear Persist All HTML CSS JS XHR Images Flash Media

URL	Status	Domain	Size	Timeline
GET l10n.js?ver=20101110	200 OK	businessnetwork.co.uk	221 B	73ms
GET jquery.js?ver=1.4.4	200 OK	businessnetwork.co.uk	26.5 KB	362ms
GET jquery.galleriffic.js?ver=3.1	200 OK	businessnetwork.co.uk	7.4 KB	170ms
GET jquery.opacityrollover.js?ver	200 OK	businessnetwork.co.uk	480 B	106ms
GET bp-moderation.js?ver=0.1.4	200 OK	businessnetwork.co.uk	693 B	103ms
GET global.js?ver=3.1	200 OK	businessnetwork.co.uk	11.5 KB	165ms
GET widget-groups.js?ver=3.1	200 OK	businessnetwork.co.uk	497 B	104ms
GET group-tags.js?ver=3.1	200 OK	businessnetwork.co.uk	505 B	148ms
GET bp-follow.js?ver=3.1	200 OK	businessnetwork.co.uk	408 B	147ms
GET store.js?ver=2.1.1	200 OK	businessnetwork.co.uk	828 B	148ms
GET comment-reply.js?ver=200	200 OK	businessnetwork.co.uk	412 B	188ms
GET bp-like.min.js?ver=3.1	200 OK	businessnetwork.co.uk	676 B	189ms
GET engine.js?ver=3.1	200 OK	businessnetwork.co.uk	1 KB	192ms
GET bp-share-it.js?ver=3.1	200 OK	businessnetwork.co.uk	264 B	190ms
GET effects.core.min.js?ver=3.1	200 OK	businessnetwork.co.uk	3.4 KB	200ms
GET effects.blind.min.js?ver=3.1	200 OK	businessnetwork.co.uk	524 B	218ms
GET wdfb_connect_widget.js?ver	200 OK	businessnetwork.co.uk	280 B	218ms
GET wdfb_facebook_login.js?ver	200 OK	businessnetwork.co.uk	282 B	217ms
GET actions.js?ver=3.1	200 OK	businessnetwork.co.uk	997 B	217ms
GET jsapi	200 OK	google.com	5.7 KB	82ms
GET t13n?form=cse-search-box	200 OK	google.co.uk	507 B	62ms
GET brand?form=cse-search-bo	200 OK	google.co.uk	793 B	63ms
GET uds?file=elements&v=1&pa	200 OK	google.com	286 B	
GET transliteration.l.js	200 OK	google.com	64.9 KB	
24 requests			128.9 KB	


190ms
200ms
218ms
218ms
217ms
217ms
82ms
62ms
63ms
70ms
123ms
989ms (onload: 2.8s)

With JavaScript

Unless we do something:
The way we **organize** our source code at **development** time...
Significantly affect its behavior at **runtime**!

Server side

Server side

- We often have non-functional requirements
  - Maintainability, extensibility, understandability, quality,
  - Productivity, Performance, ...

Server side

- We often have non-functional requirements
  - Maintainability, extensibility, understandability, quality,
  - Productivity, Performance, ...
- We have techniques and tools to help
  - Architecture, modularity, reuse, separation of concerns
  - automated testing & continuous integration
  - Tool support (static analysis, compilers, IDEs, profilers)

Why so different?

Some key properties of JavaScript as a language

- Linkage of different scripts/modules via global variables.
- Delivered as source code, as opposed to executable binaries.
  - Compilation is done by browser: compiles scripts as it receives them
- Dynamically typed
- Dynamic Objects (general containers)
- Prototypal inheritance
  - Objects inherit from objects (no classes)

JavaScript in the large

JavaScript in the large

- JavaScript(ECMA-262 3rd edition)is focused on **flexibility** and **ease** of use:
  - poorly suited for writing large & complex applications.

JavaScript in the large

- JavaScript(ECMA-262 3rd edition)is focused on **flexibility** and **ease** of use:
  - poorly suited for writing large & complex applications.
- Here are some of the problems for large-scale use
  - language flaws (e.g., type conversion, scope rules, numbers...)
  - global namespace / missing module system (packages)
  - missing encapsulation
  - everything is mutable - even methods on objects
  - program information hard to use by tools (e.g., static typing)

"Larger" projects

(whatever that means)

"Larger" projects

(whatever that means)

- This may or may not be a problem for you, **except** when your app scales in one or more of:

"Larger" projects

(whatever that means)

- This may or may not be a problem for you, **except** when your app scales in one or more of:
  - size and complexity

"Larger" projects

(whatever that means)

- This may or may not be a problem for you, **except** when your app scales in one or more of:
  - size and complexity
  - development team: size and composition (new people, different skills)

"Larger" projects

(whatever that means)

- This may or may not be a problem for you, **except** when your app scales in one or more of:
  - size and complexity
  - development team: size and composition (new people, different skills)
  - time: stretches over years of development and maintenance.

Theme of this talk:
What can we do?

Theme of this talk: What can we do?

- **Rationale** part ("the theory")
  - intro: helpful technologies and techniques
- **Practical** part ("the code"):
  open source sample application with a strict architecture and a JavaScript tool-chain.
  - a modular MVC architecture (using custom events)
  - advanced tooling (compilation, type-check, lazy-loading)
  - automated testing (unit, integration and functional)

Compiler+VM Technology

Compiler+VM Technology

"JavaScript is assembly language for the web"

-Eric Meijer

Compiler+VM Technology

- Google Dart
- ClojureScript (Clojure -> JS)
- Cappuccino (Objective-J -> JS)
- CoffeeScript (lightweight, "local" JS compilation)
- Google Closure Compiler (JS -> JS),
- Traceur (JS.Next->JS)
- Google Web Toolkit (Java -> JS)
- JSIL (.NET/CIL -> JS)
- ...

"JavaScript is assembly language for the web" -Eric Meijer

Differences in compilers

Differences in compilers

Differences in compilers

This talk assumes you are staying in JavaScript.

Google Closure Compiler

Google Closure Compiler

- Extensible optimizing JavaScript-to-JavaScript compiler

Google Closure Compiler

- Extensible optimizing JavaScript-to-JavaScript compiler
- Several modes - the interesting one is "**advanced mode**".
  - **whole-program** analysis (all js files + extern must be supplied).
  - optimizations (e.g., dead-code elimination, constant folding/ propagation)
  - optional type-checking, @private access, ...
  - lazy-loading of modules

Google Closure Compiler

- Extensible optimizing JavaScript-to-JavaScript compiler
- Several modes - the interesting one is "**advanced mode**".
  - **whole-program** analysis (all js files + extern must be supplied).
  - optimizations (e.g., dead-code elimination, constant folding/ propagation)
  - optional type-checking, @private access, ...
  - lazy-loading of modules
- Requires writing JavaScript code in a particular way.

Google Closure Compiler

- Extensible optimizing JavaScript-to-JavaScript compiler
- Several modes - the interesting one is "**advanced mode**".
  - **whole-program** analysis (all js files + extern must be supplied).
  - optimizations (e.g., dead-code elimination, constant folding/ propagation)
  - optional type-checking, @private access, ...
  - lazy-loading of modules
- Requires writing JavaScript code in a particular way.
- Very under-appreciated! Not just yet another minifier!

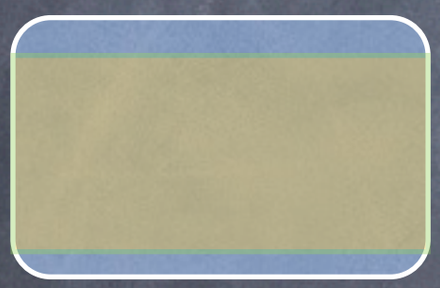
Typical JS Application composition

Total amount of code

Code actually used at runtime

Typical JS Application composition

Your code



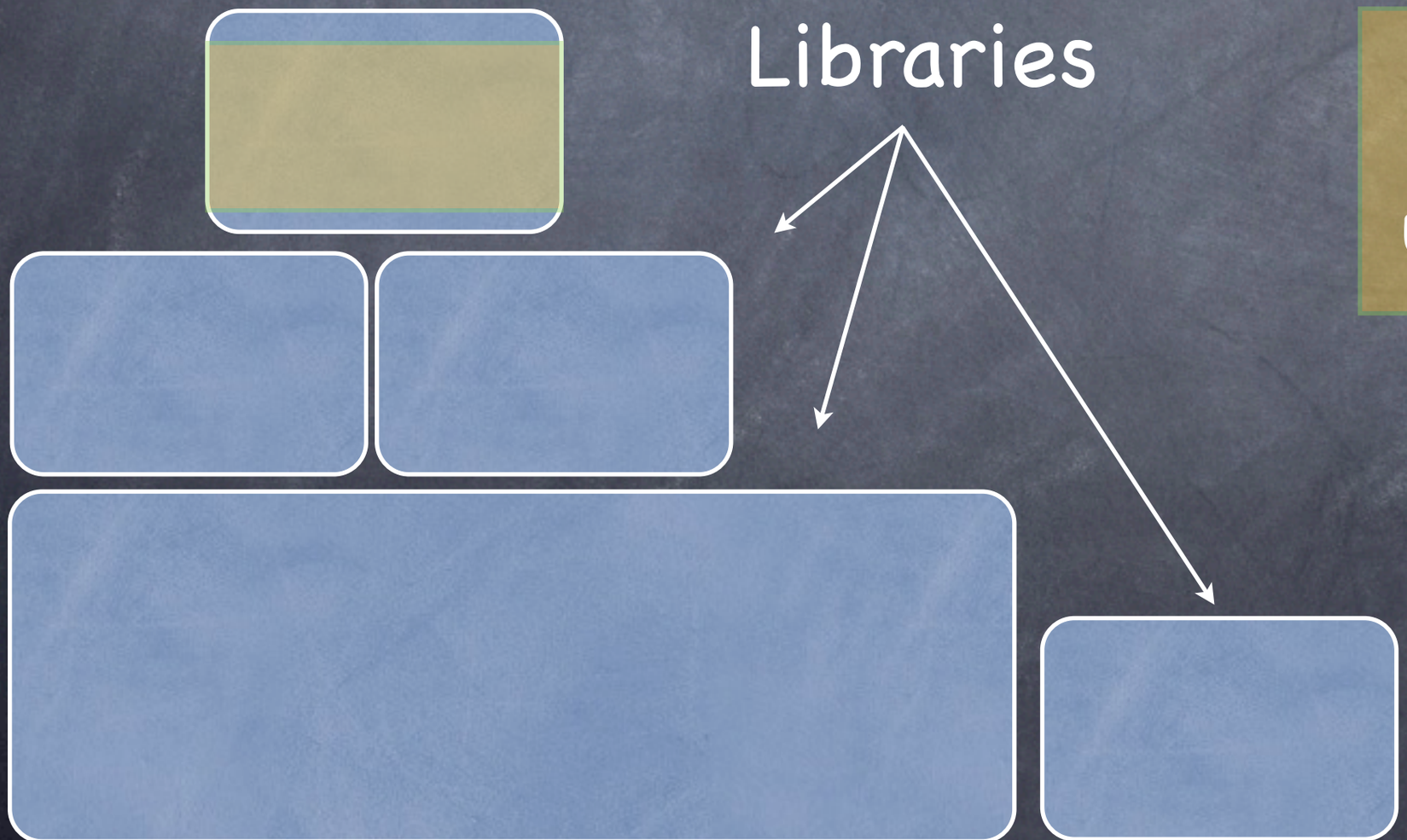
Total amount of code

Code actually used at runtime

Typical JS Application composition

Your code

Libraries



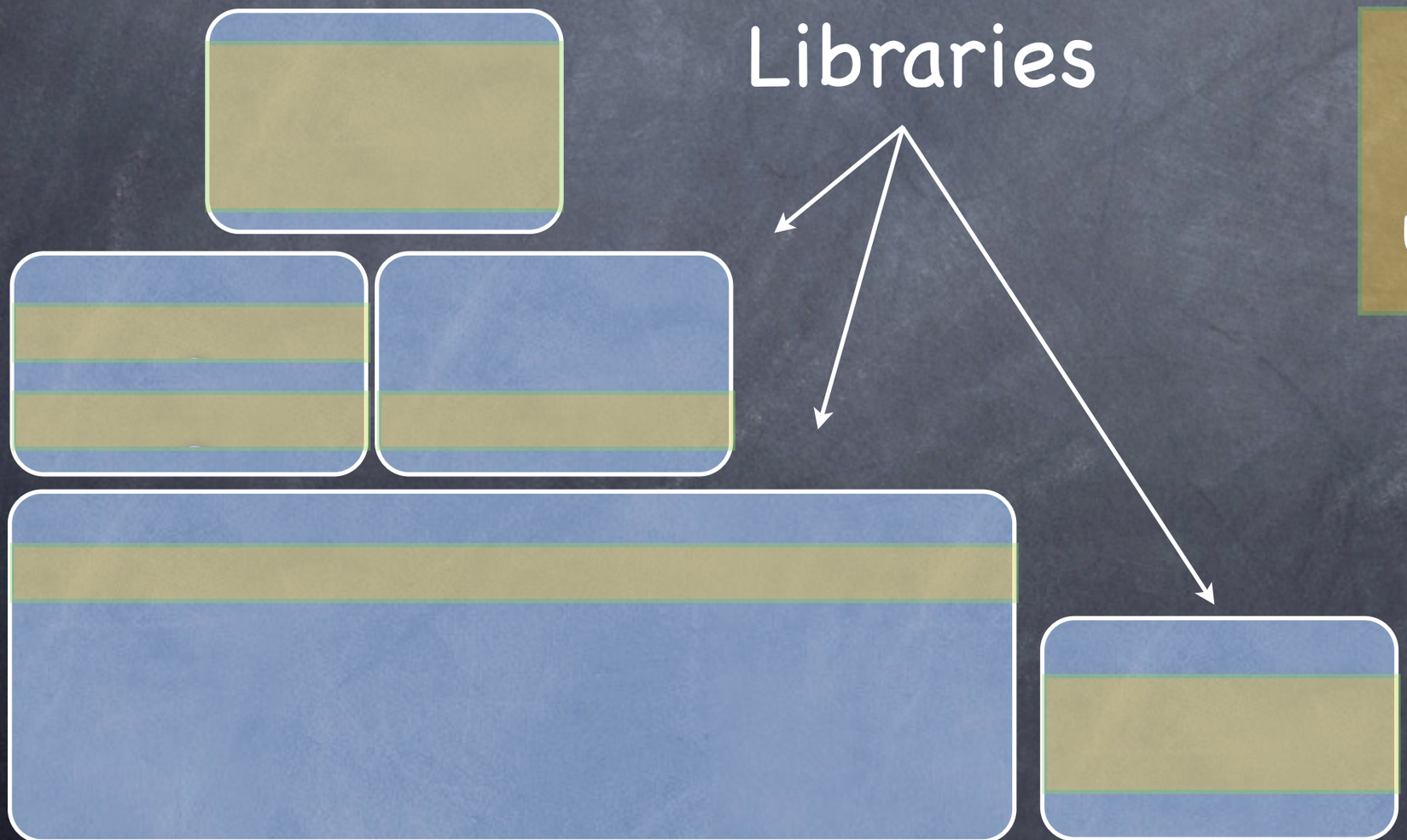
Total amount of code

Code actually used at runtime

Typical JS Application composition

Your code

Libraries

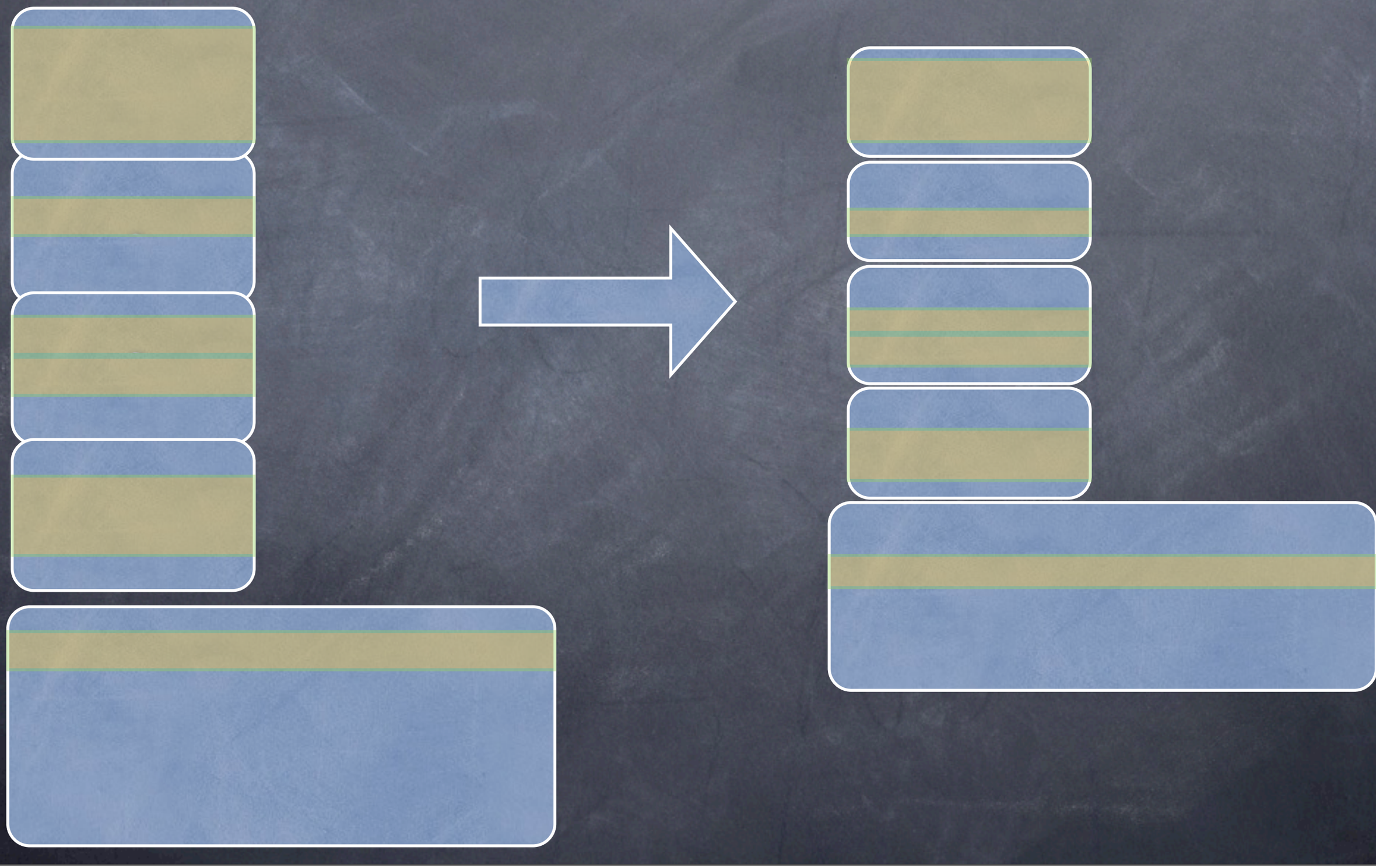


Total amount of code

Code actually used at runtime

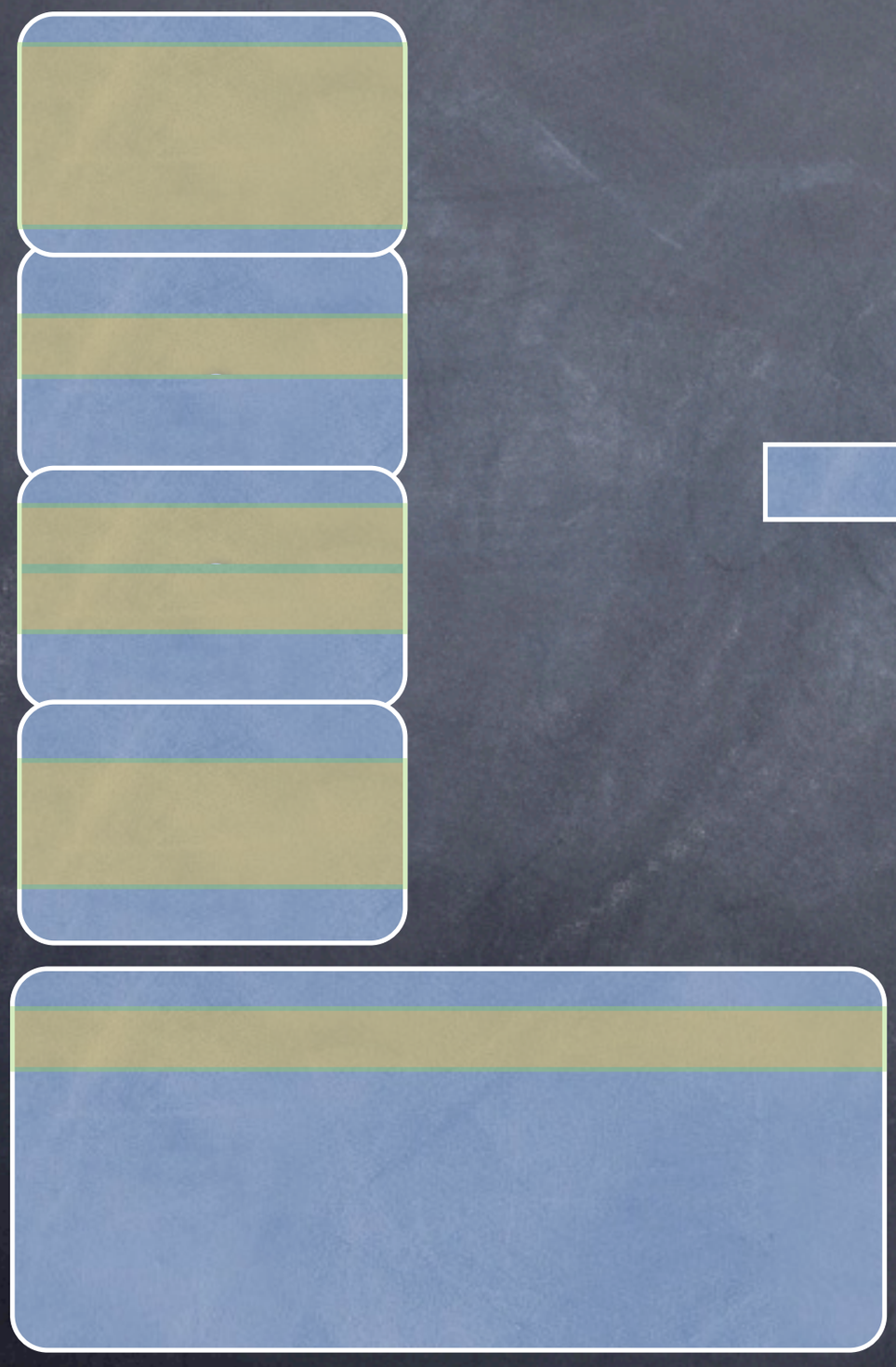
Local optimization

Local optimization



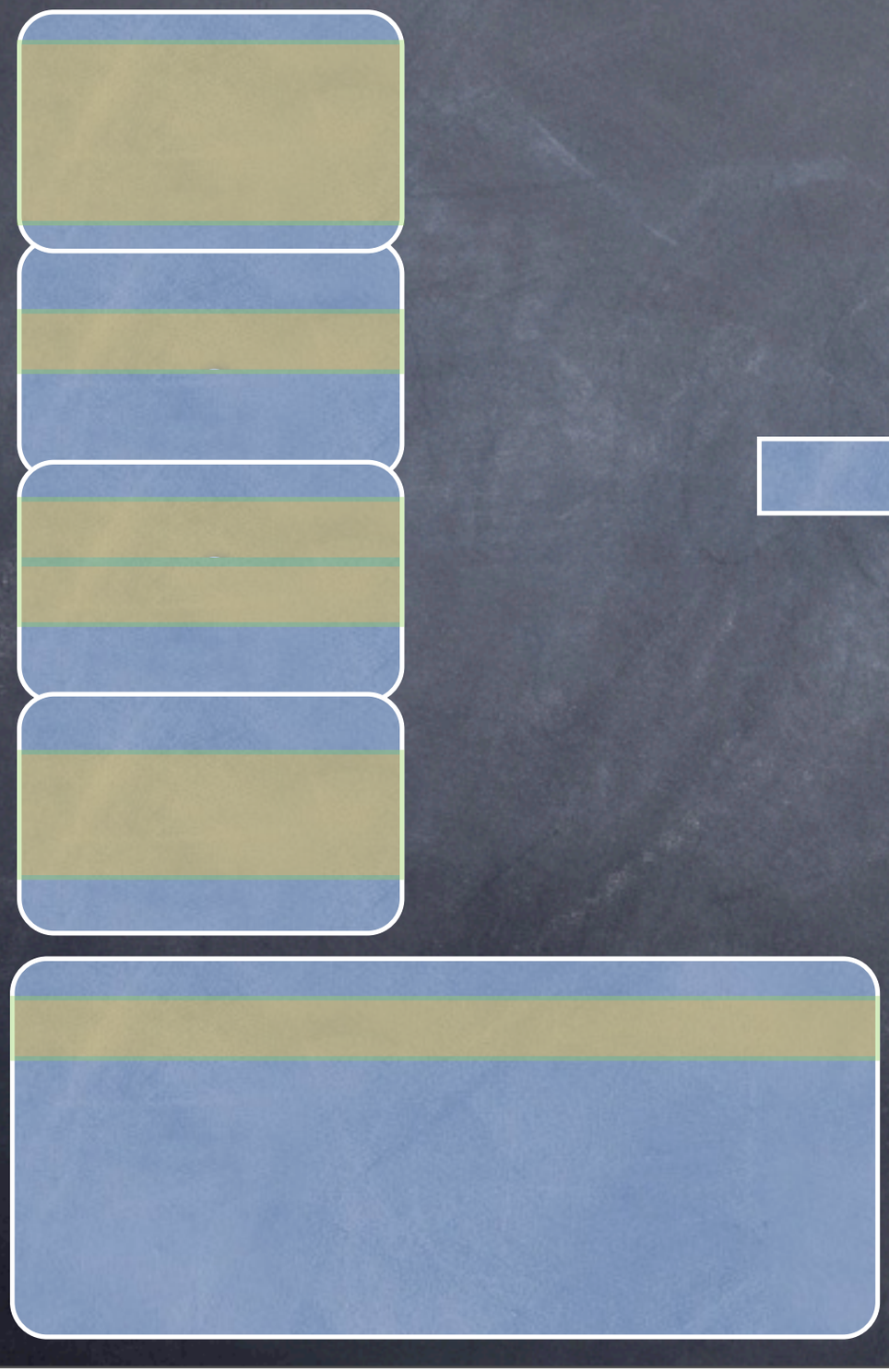
Whole program analysis & Closure Tools

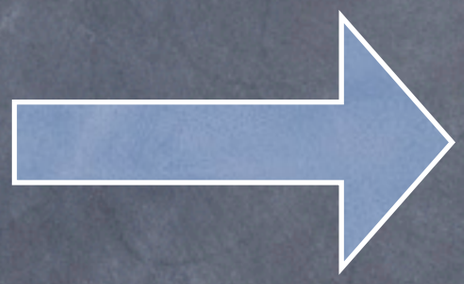
Whole program analysis & Closure Tools



App. base modules

Whole program analysis & Closure Tools





Other Modules loaded by need

App. base modules

Closure Library

Closure Library

- HUGE JavaScript library used by Google.
  - Symbiotic with closure compiler advanced mode, which makes it SMALL when used with compiler.

Closure Library

- HUGE JavaScript library used by Google.
  - Symbiotic with closure compiler advanced mode, which makes it SMALL when used with compiler.
- Highlights (apart from expected of a JS library)
  - Dependency/module system goog.provide, goog.require.
  - “Interfaces”, “Enums”, “Class”-based inheritance system.
  - (custom) event system supporting bubbling, capture, preventDefault and cancel.
  - extensible UI components with a well-defined lifecycle, and which support custom events.

JSDoc annotations and tooling

JSDoc annotations and tooling

- Several advantages to using JSDoc:
  - Standardized => tool support
  - Google uses JSDoc annotations for "talking" to the compiler (e.g., denoting types and access/visibility).
  - Many IDEs understand them, and can help with warnings, code-completion, inline documentation.
  - Makes you document your code and think about what your public API is.

JSDoc annotations and tooling

- Several advantages to using JSDoc:
  - Standardized => tool support
  - Google uses JSDoc annotations for "talking" to the compiler (e.g., denoting types and access/visibility).
  - Many IDEs understand them, and can help with warnings, code-completion, inline documentation.
  - Makes you document your code and think about what your public API is.
- Example IDEs: WebStorm, RubyMine, Spket, Aptana

JSDoc annotations and tooling

- Several advantages to using JSDoc:
  - Standardize
  - Google uses compiler (e.
  - Many IDEs warnings, c
  - Makes you what your

```
/**
 * Constructor for Connection objects that establishes
 * a WebSocket-based connection to the chat server
 * and sends and parses messages and converts them to
 * application-level events.
 *
 * @param {string} url to connect to websocket server.
 * @constructor
 * @extends {goog.events.EventTarget}
 */
chatpp.controller.conn.Connection = function(url) {
    goog.base(this);
    this.url_ = url;

    /**
     * @type {stomple.Client}
     * @private
     */
    this.client_ = new stomple.Client(url);
};
goog.inherits(chatpp.controller.conn.Connection, goog.events.EventTarget);
```

- Example IDEs: WebStorm, RubyMine, Spket, Aptana

Automated Testing

- js-test-driver (http://code.google.com/p/js-test-driver/)
  - unit tests executing in real browsers
  - ide and commandline
  - SinonJS for spies,stubs,mocks
- Selenium Webdriver ("selenium 2.0")
  - De-facto standard for automated functional/acceptance testing.

JavaScript Application Architecture

- In the past few years there have been focus on JavaScript application architecture. People like:
  - Nicholas Zakas (ex YAHOO, JS app arch)
  - Rebecca Murphey (jQuery vs "enterprise")
  - Peter Michaux (MVC Architecture)
  - Ray Ryan (Google, GWT app arch.)

Common theme

Common theme

- **Modularity - reusable modules, loose coupling.**
  - clearly def. public interface and private state+functions

Common theme

- **Modularity - reusable modules, loose coupling.**
  - clearly def. public interface and private state+functions
- **Strict file organization - matches modules**
  - Many small files, each file one concern (like good class design)

Common theme

- **Modularity - reusable modules, loose coupling.**
  - clearly def. public interface and private state+functions
- **Strict file organization - matches modules**
  - Many small files, each file one concern (like good class design)
- **Separation of concerns (e.g., MVC, MVP)**
  - view management, communication/network access, domain model,...

Common theme

- **Modularity - reusable modules, loose coupling.**
  - clearly def. public interface and private state+functions
- **Strict file organization - matches modules**
  - Many small files, each file one concern (like good class design)
- **Separation of concerns (e.g., MVC, MVP)**
  - view management, communication/network access, domain model,...
- **Custom events**
  - extend the DOM Level 2 event model to app-specific events

A note on custom events

A note on custom events

- Custom events: like DOM-Level 2 **except**
  - App.-specific and -generated
  - Logical rather than physical

A note on custom events

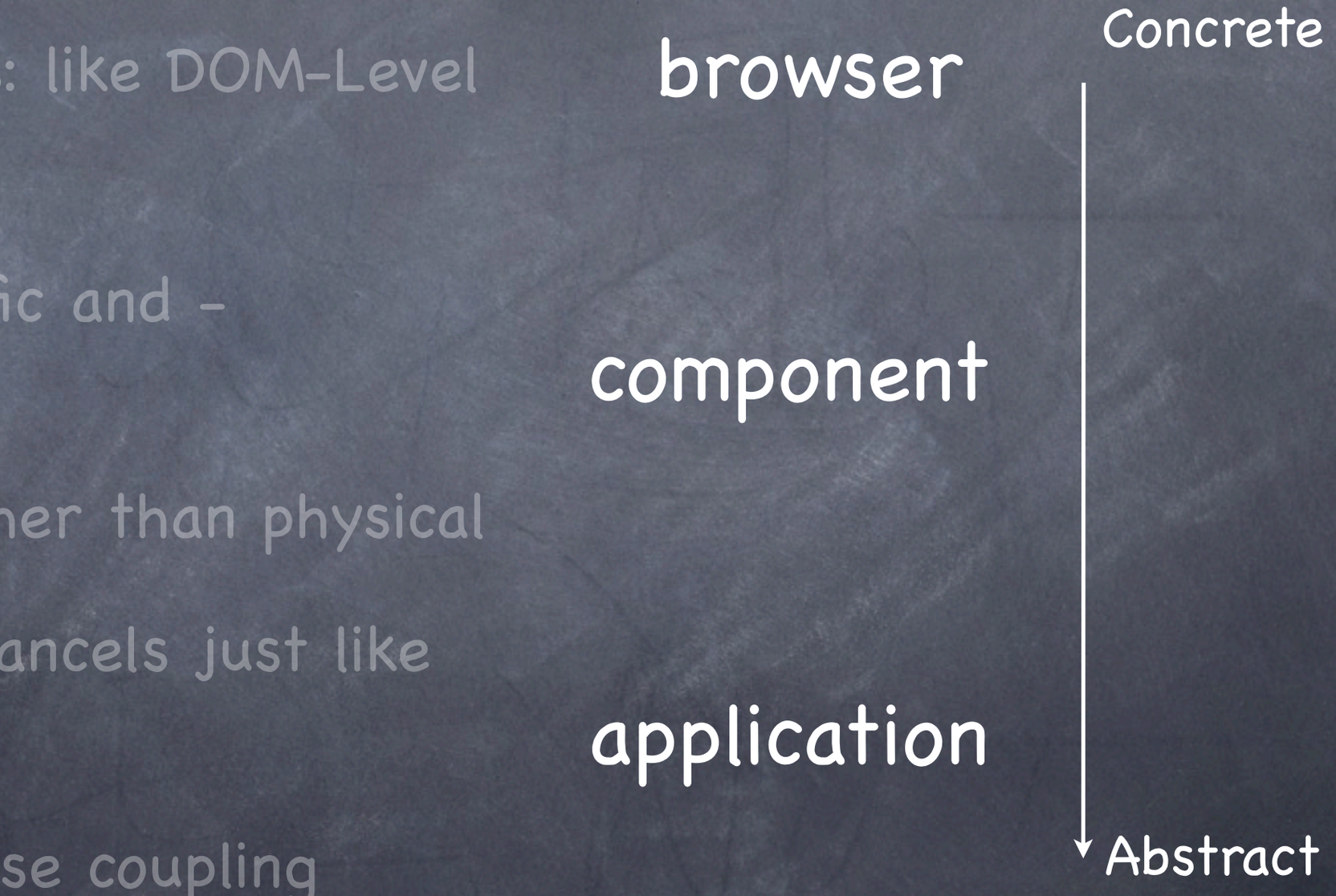
- Custom events: like DOM-Level 2 **except**
  - App.-specific and -generated
  - Logical rather than physical
- Bubbles and cancels just like DOM events.

A note on custom events

- Custom events: like DOM-Level 2 **except**
  - App.-specific and -generated
  - Logical rather than physical
- Bubbles and cancels just like DOM events.
- Helps with loose coupling

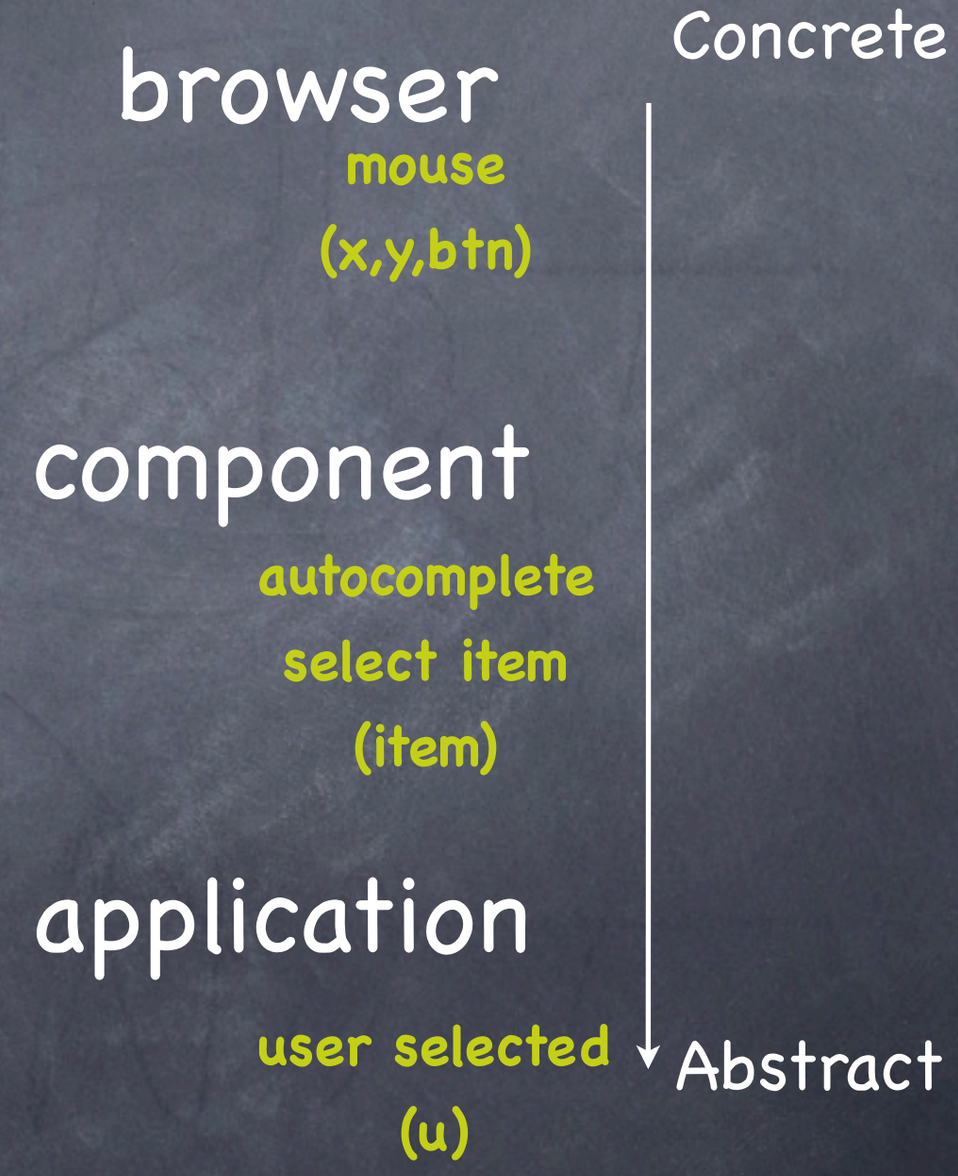
A note on custom events

- Custom events: like DOM-Level 2 **except**
  - App.-specific and -generated
  - Logical rather than physical
- Bubbles and cancels just like DOM events.
- Helps with loose coupling



A note on custom events

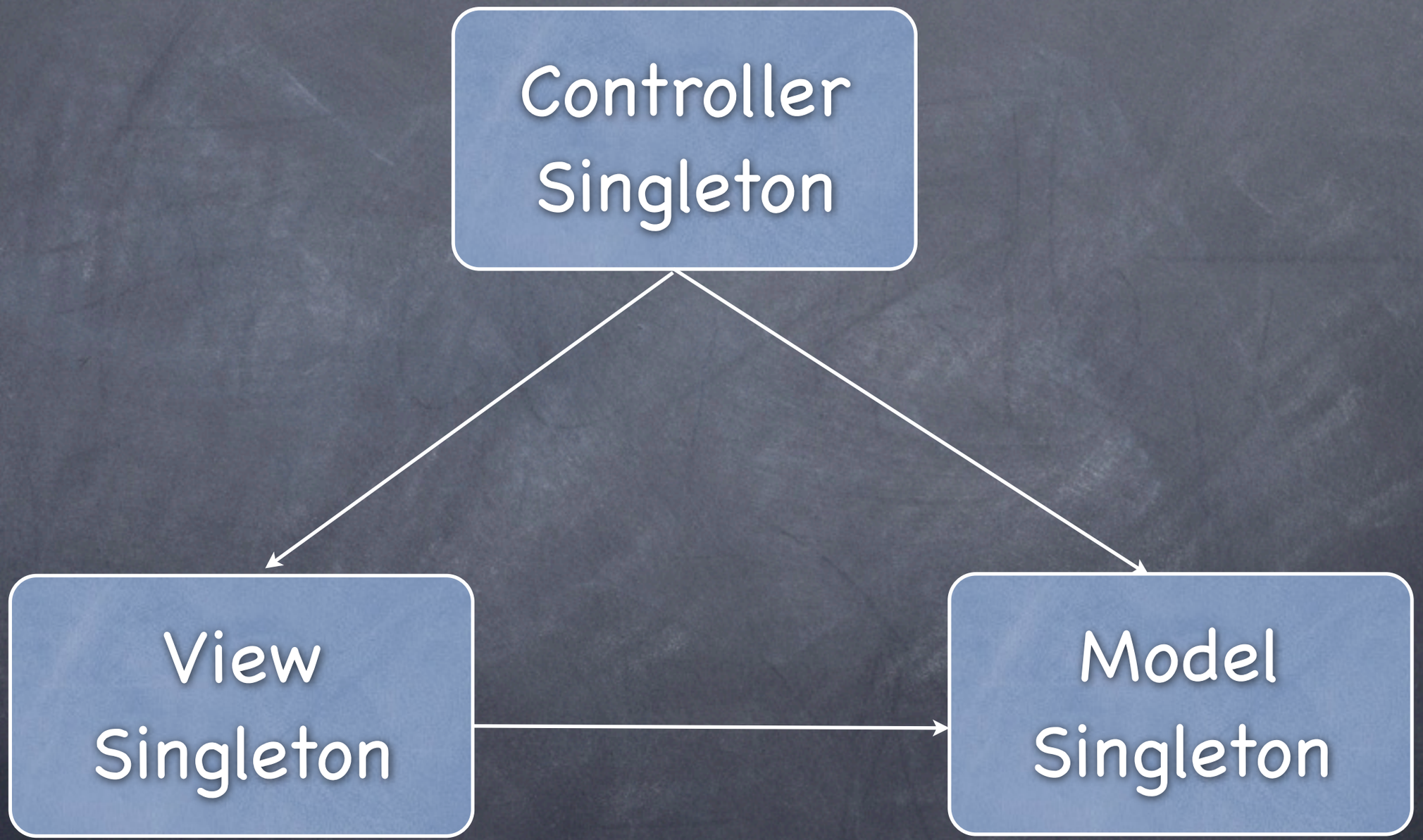
- Custom events: like DOM-Level 2 **except**
  - App.-specific and - generated
  - Logical rather than physical
- Bubbles and cancels just like DOM events.
- Helps with loose coupling



Dependency

MVC

(Yes, again)

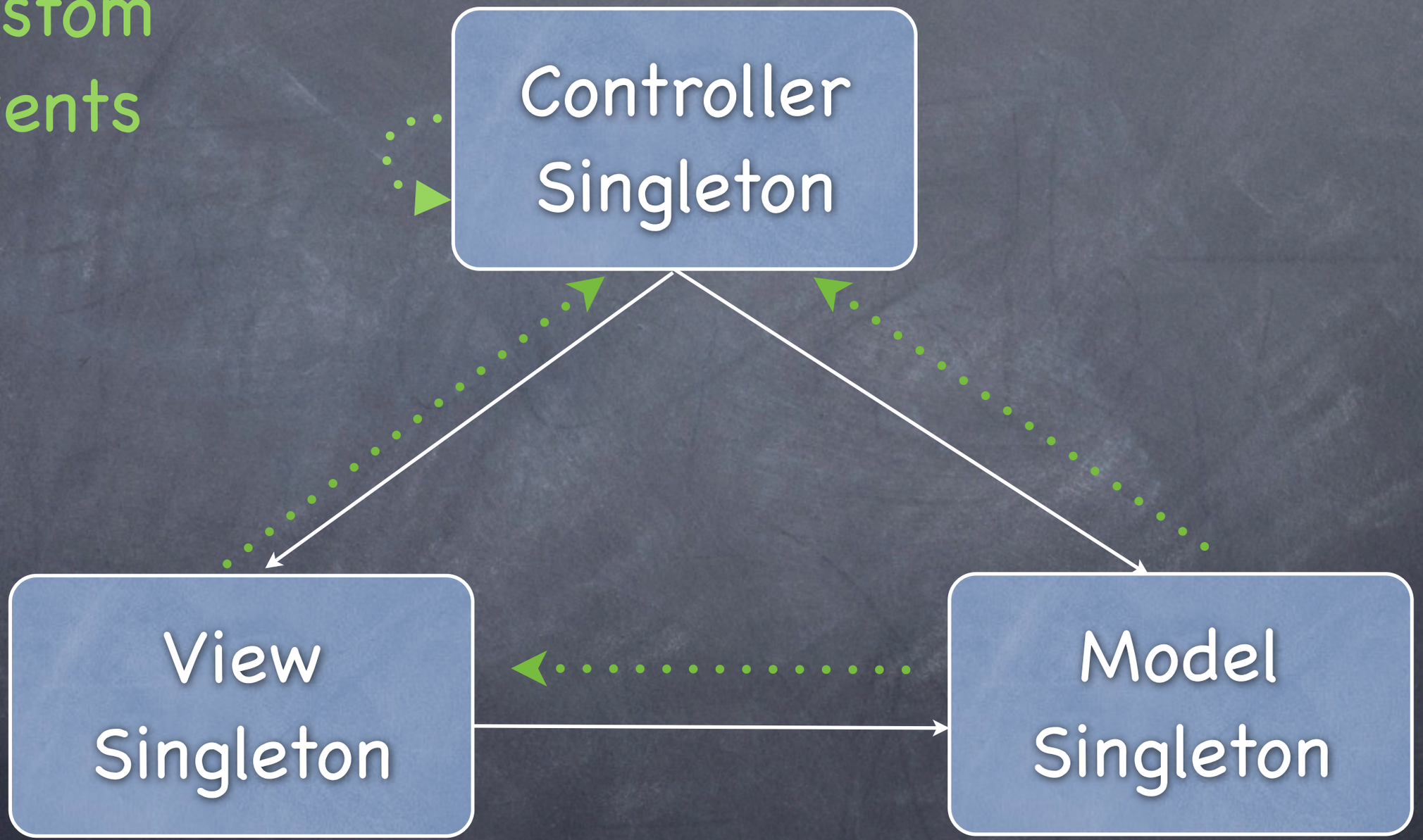


MVC

(Yes, again)

Dependency

Custom Events

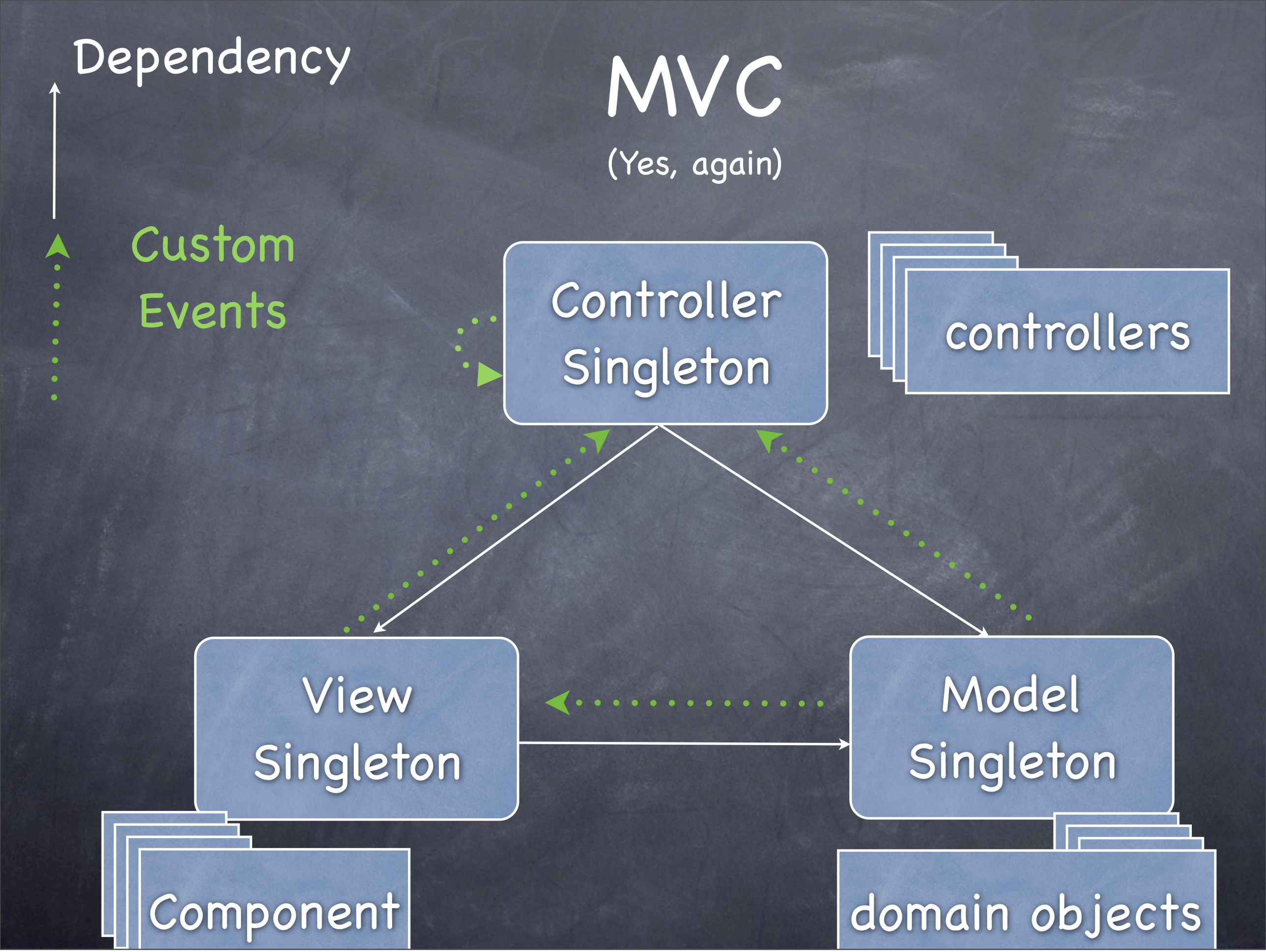


MVC

(Yes, again)

Dependency

Custom Events

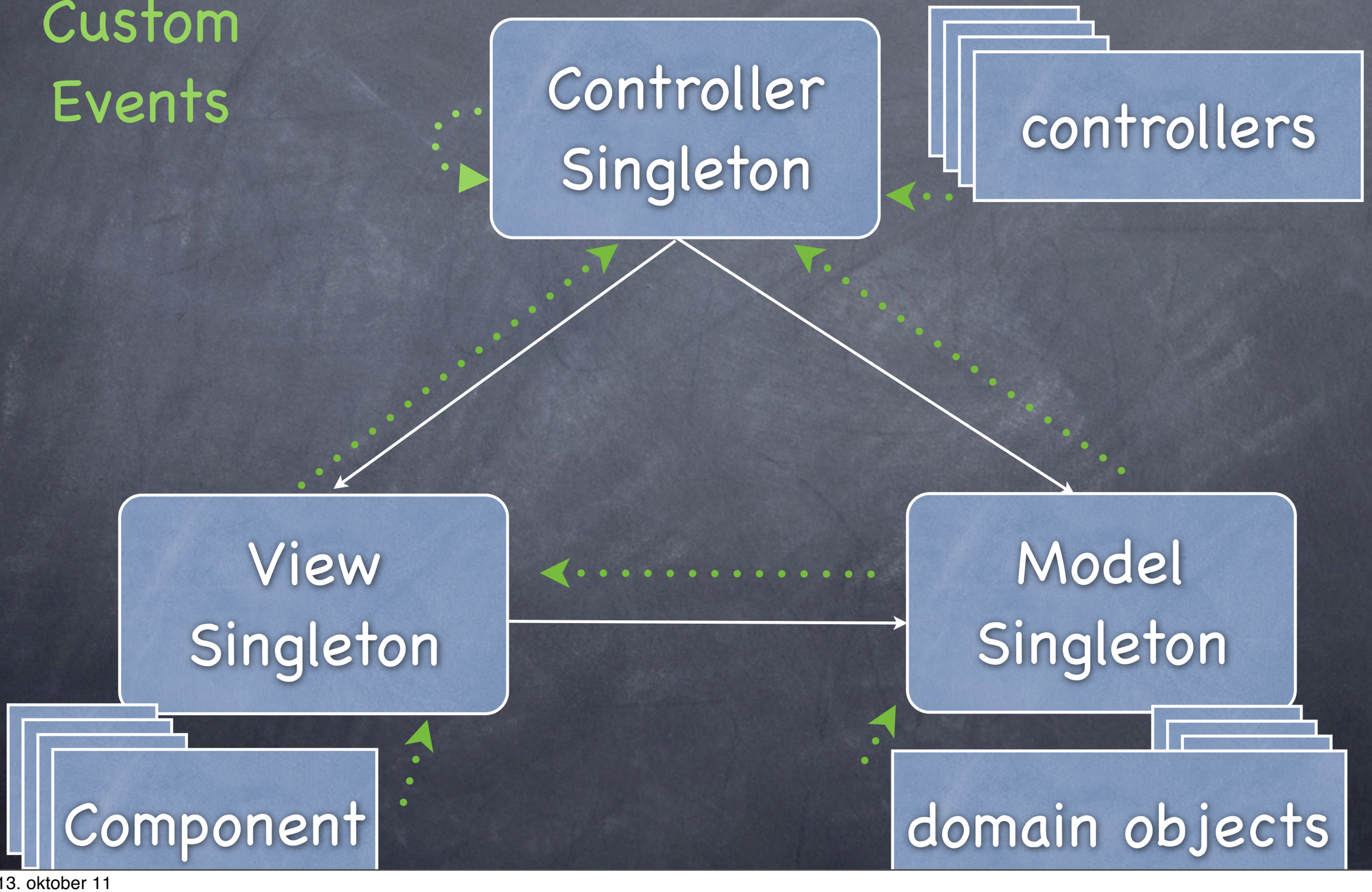


MVC

(Yes, again)

Dependency

Custom Events



How to do it in practice?

Example: chatroom++

Example: chatroom++

- Communication uses WebSockets+Stomp on Torquebox
  - Messages are 'instant' & can be transactional with reliable delivery (JMS via STOMP on HornetQ).

Example: chatroom++

- Communication uses WebSockets+Stomp on Torquebox
  - Messages are 'instant' & can be transactional with reliable delivery (JMS via STOMP on HornetQ).
- JavaScript Tool-chain
  - Google closure compiler and linter/checker.
  - Jetbrains' RubyMine as IDE (JSDoc comments, code "intelligence")
  - functional tests: selenium-webdriver for automating browsers
  - unit tests: js-test-driver + SinonJS

Example: chatroom++

- Communication uses WebSockets+Stomp on Torquebox
  - Messages are 'instant' & can be transactional with reliable delivery (JMS via STOMP on HornetQ).
- JavaScript Tool-chain
  - Google closure compiler and linter/checker.
  - Jetbrains' RubyMine as IDE (JSDoc comments, code "intelligence")
  - functional tests: selenium-webdriver for automating browsers
  - unit tests: js-test-driver + SinonJS
- JavaScript Client:
  - Google Closure Library for compatibility with Closure compiler.
  - Stomple JavaScript library (STOMP over websockets).
  - Model-View-Controller with Custom Events.
  - Lazy-module loading, strict file organization, optional types.

DEMO

Two Solutions Compared

- One: jQuery, jQuery-UI, and stilts-stomp.js.
  - Minified using UglifyJS.
  - No strict architecture, but does separate view and "everything else", uses custom events.
- Two: Google Closure Library and Stomple-0.99.
  - Both libraries are written to be compatible with Closure Compiler Advanced mode.
  - Modular Model-View-Controller arch. using custom events.

Development time

Total: 603,3 kB

app.js (two files) (6.3kB)

stilts-stomp.js (7kB)

jQuery-UI-min
(367kB)

jQuery (233kB)

Total: ~ 1.3 MB

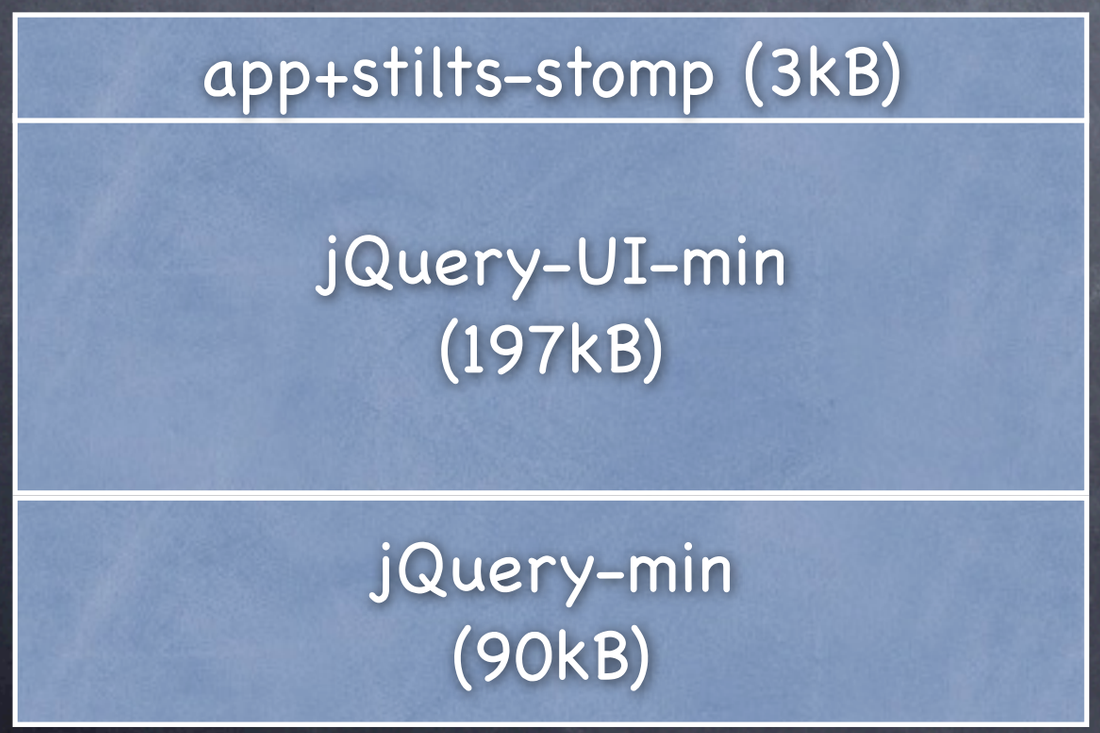
"App" (several files...)
(35kB)

Stomple-0.99.js (39,6kB)

Closure Library
(approx. 1.2MB)

Production

Total: 290 kB



Production

Total: 290 kB

Total: 71 kB

app+stilts-stomp (3kB)

jQuery-UI-min
(197kB)

jQuery-min
(90kB)

Closure compiled: 71kB

Production

Total: 290 kB

app+stilts-stomp (3kB)

jQuery-UI-min
(197kB)

jQuery-min
(90kB)

Total: 71 kB

or optionally

Webkit Closure compiled: 64kB

Closure compiled: 71kB

Summary

Summary

- JavaScript is not well-suited for large-scale application development. We must add "something"
  - Requires much discipline, structure, convention.
  - Poor tradition and literature about client-side architecture.

Summary

- JavaScript is not well-suited for large-scale application development. We must add "something"
  - Requires much discipline, structure, convention.
  - Poor tradition and literature about client-side architecture.
- Tooling is not great, but there **are** tools that can help:
  - Compiler technologies are superior tools. There are significant differences in compiler techs. Closure is not just another minifier.
  - Although not perfect, IDEs and testing tools are getting better.

Summary

- JavaScript is not well-suited for large-scale application development. We must add "something"
  - Requires much discipline, structure, convention.
  - Poor tradition and literature about client-side architecture.
- Tooling is not great, but there **are** tools that can help:
  - Compiler technologies are superior tools. There are significant differences in compiler techs. Closure is not just another minifier.
  - Although not perfect, IDEs and testing tools are getting better.
- Custom events, MVC-like patterns, file-organization help with structuring your application.

Summary

- JavaScript is not well-suited for large-scale application development. We must add "something"
  - Requires much discipline, structure, convention.
  - Poor tradition and literature about client-side architecture.
- Tooling is not great, but there **are** tools that can help:
  - Compiler technologies are superior tools. There are significant differences in compiler techs. Closure is not just another minifier.
  - Although not perfect, IDEs and testing tools are getting better.
- Custom events, MVC-like patterns, file-organization help with structuring your application.
- Example: https://github.com/krukow/advanced_javascript_tooling

Google Tools to Add Power to Your JavaScript

Closure

The Definitive Guide

O'REILLY®

Michael Bolin