

# Writing JavaScript Apps to Scale

Phil @leggetter  
phil@leggetter.co.uk  
Caplin Systems

# BladeRunnerJS

@BladeRunnerJS

'Scale'

# What is a large-scale JavaScript app?

*In my view, large-scale JavaScript apps are non-trivial applications requiring significant developer effort to maintain, where most heavy lifting of data manipulation and display falls to the browser.*

Addy Osmani - Large-Scale JavaScript

# What factors affect scaling?

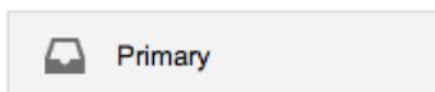
# 1. Complexity



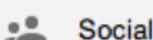
C

More ▾

1–50 of 57



Primary



Social



Promotions



Updates



Forums

   Gmail Team

Welcome to the new Gmail inbox - Hi Philip Meet the new inbox Inbox tabs put you back in control with simp

   ProjectLocker

[ProjectLocker] Faster Access with

   e-Documents

New eDocument arrived - Please ac

   Adam Todd

Newly Business Information - Hi, M

   service, me (2)

Receipt for your PayPal payment to

   Microsoft BizSpark

Call for Startup videos, where deve

   e-Documents

New eDocument arrived - Please ac

   Adam Todd

Business Leads - Hi, We are a leadi

   FollowUp

Your email account, phil@pusher.

   Expensify

[Expensify] Please approve expens

   Pooja

New Business Enquiry - Hi Phil, I b

   Joe Benjamin

Your Invitation to the 2013 New En

   e-Documents

New eDocument arrived - Please ac

   Boomerang for Gmail

Grab your Capes! Boomerang for A

New Message

To

Subject

Send



**CAPLIN**

My Tiles Other Tile Set Other Tile Set

EURUSD +

USDDKK		SP FW SW	X
SELL USD	enter amount	GFA 8.888M	RFS ODR
5.71 SPOT 655	20/12/12 SPOT	5.72 -1.75 080	
5.71 -2.5 405	27/12/12 1W	5.71 -4.75 180	
5.71 -4.75 180	03/01/13 2W	5.71 -4.25 185	
5.70 -11.25 530	24/01/13 1M	5.71 -10.25 230	
5.68 -26.25 805	21/02/13 2M	5.69 -26.25 605	
5.66 -51.00 555	21/03/13 3M	5.67 -49.00 355	
5.58 -134.00 255	20/05/13 5M	5.59 -124.00 855	
5.48 -230.00 655	19/09/13 6M	5.51 -210.00 255	

USDCHF		SP FW SW	X
SELL USD	BUY USD	SP FW SW	X
0.96	0.96	45 <sub>7</sub> 65 <sub>0</sub>	
USD - 5M	GFA 8.888M	RFS	
20/12/2012 (SPOT)			

EURJPY		SP FW SW	X
SELL EUR	BUY EUR	63 <sub>5</sub> 85 <sub>2</sub>	
108.	108.		
EUR - enter amount	GFA 8.888M	RFS	
20/12/2012 (SPOT)			

USDMUR		SP FW SW	X
SELL USD	BUY MUR	65 <sub>5</sub> 35 <sub>5</sub>	
5.71 -2.5 655	27/12/12 1W	5.71 -4.75 355	
5.71 -4.75 155	03/01/13 2W	5.71 -4.25 155	
USD - enter amount	GFA 8.888M	RFS	
20/12/2012 (SPOT)			

New Tile Set  
Open Tile Set  
Save Tile Set As...

1 Click Trading   Standard Trading

Close All Tiles

N. Europe Europe Asia Sparkline

CCY Pair Bid Ask Sparkline

EURRUB ▲ 29.3423 ▼ 29.3643

EURRUB 29.3423 ▲ 29.3643

EURRUB 29.3423 29.3643

EURRUB ▲ 29.3423 ▼ 29.3643

EURRUB 29.3423 ▲ 29.3643

EURRUB 29.3423 29.3643

Other Watchlist My Watchlist

EURUSD +

CCY Pair	Bid	Ask	Sparkline
EURRUB ▲ 29.3423 ▼ 29.3643			
EURRUB 29.3423 ▲ 29.3643			
EURRUB 29.3423 29.3643			
EURRUB ▲ 29.3423 ▼ 29.3643			
EURRUB 29.3423 ▲ 29.3643			
EURRUB 29.3423 29.3643			

News Other Tile Set

Top Stories

Barclays to shut wealth management services in 130 countries Reuters

Loans to eurozone businesses 'drop sharply' Reuters

As rupee hits hard times, India cajoles and woos offshore currency players Reuters

Japan shares rally, U.S. budget fuels caution elsewhere Reuters

Activity Blotter Historic Blotter Orders Positions

1y Sep 25, 2012 - Sep 25, 2013

Vol Nov '12 Jan '13 Mar '13 May '13 Jul '13 Sep '13

## Gmail & Caplin Trader

- Large Single Page Apps (SPAs)
- Complex functionality
- Complex interactions
  - Technology
  - User

# 2. Size (Large Codebase)

More functionality = More Code

# Caplin Trader

- SDK:
  - 1,004 JavaScript files
  - 130,594 LoC
  - ~100 lines per file (inc. comments + whitespace)
- Tests:
  - 641 files
  - 95,000 LoC
- Reference implementation:
  - 425 JavaScript files
  - ~50,000 LoC
  - 200 test files
  - 21,000 LoC

# 3. Contributors

The Human Factor

# Who contributes to an app?

- Multiple devs
- Multiple teams
- Front-end devs
- Back-end devs
- Designers
- QA
- Infrastructure

## Scale apps to account for:

1. Complexity
2. Size (codebase)
3. Contributors (human factors)

# What causes scaling problems?

# Scaling problems caused by

- Inconsistency
- Unnecessary Complexity
- Overlapping concerns need a better term

# Problems avoided through

- Consistency
- Simplicity
- Separation of Concerns (SoC)

Enabled through Principles, Practices, Libraries & Tooling

# Patterns & Practices

These are the patterns & practices we've considered when:

- Defining our development workflow
- Defining our application architecture
- Defining & building our development tools

# Tooling & Libraries

- Building our tooling & libraries: [BladeRunnerJS](#)

# Tooling & Libraries enable, encourage (and enforce) good practices

- Consistent Coding Style
- Consistent Code Structure
- A Maintainable Application Architecture
  - Technology & Human Factors
- Code quality through testing

# Coding Style

# Coding Style Guide

*Programming style is a set of rules or guidelines used when writing the source code for a computer program. It is often claimed that following a particular programming style will help programmers to read and understand source code conforming to the style, and help to avoid introducing errors.*

aka [Programming Style](#)

note: "[Coding Conventions](#)" can cover much more

# Static Code Analysis

- Define a Coding Style
- Check with JSHint

# Editor Integration

## Realtime feedback

The screenshot shows a code editor window with a dark theme. On the left is the code file `NetMonitorPanel.jsm`. The code is annotated with several error markers (yellow triangles) and highlights (underlines). The right side of the editor displays a list of errors and warnings from a static analysis tool. The errors are numbered and include:

- 8:65 'Components' is not defined.
- 10:1 Possible strict violation.
- 15:1 'XPCOMUtils' is not defined.
- 15:35 Possible strict violation.
- 18:1 Possible strict violation.
- 26:3 'EventEmitter' is not defined.
- 27:2 Missing semicolon.
- 29:1 'NetMonitorPanel' is not defined.
- 43:17 'Promise' is not defined.
- 54:43 'aReason' is defined but never used.
- 56:24 'reason' is not defined.
- 56:46 'reason' is not defined.
- 61:14 'function closure expressions' is only available in

The code itself is a Mozilla add-on manifest file (`.jsm`) containing logic for initializing a panel and handling promises.

```
1 /* -*- Mode: javascript; tab-width: 2;
2 /* vim: set ft=javascript ts=2 et sw=2
3 /* This Source Code Form is subject to
4 * License, v. 2.0. If a copy of the MF
5 * file, You can obtain one at http://
6 "use strict";
7
8 const { classes: Cc, interfaces: Ci, ut
9
10 this.EXPORTED_SYMBOLS = ["NetMonitorPar
11
12 Cu.import("resource://gre/modules/XPCOM
13 Cu.import("resource:///modules/devtools
14
15 XPCOMUtils.defineLazyModuleGetter(this,
16   "resource://gre/modules/commonjs/sdk/
17
18 this.NetMonitorPanel = function NetMoni
19   this.panelWin = iframeWindow;
20   this._toolbox = toolbox;
21
22   this._view = this.panelWin.NetMonitor
23   this._controller = this.panelWin.NetMonitorController;
24   this._controller._target = this.target;
25
26 EventEmitter.decorate(this);
27 }
28
29 NetMonitorPanel.prototype = {
30   /**
31    * Open is effectively an asynchronous constructor.
32    *
33    * @return object
34    *         A Promise that is resolved when the NetMonitor completes opening.
35    */
36   open: function() {
37     let promise;
38
39     // Local monitoring needs to make the target remote.
40     if (!this.target.isRemote) {
41       promise = this.target.makeRemote();
42     } else {
43       promise = Promise.resolve(this.target);
44     }
45 }
```

# Workflow Integration

- Feedback on:
  - File save
  - Local build / F5

# CI Build Integration

- Build fails if style guide rules are not met

# Tooling

- [JSHint](#) support for:
  - VIM
  - Emacs
  - Sublime text
  - TextMate
  - Visual Studio
- [jshint-eclipse](#)
- Grunt plugin: [grunt-contrib-jshint](#)
- [WebStorm jshint](#)

# Guides

- [Caplin's Coding Style Guide](#)
- [Airbnb's JavaScript Style Guide](#)

# Code Structure

# JavaScript === Flexible

# Use a Programming Paradigm

- All developers
- Across all teams

# We use Classical OOP

- Classes
- Inheritance
- Interfaces
- Everything is a Class or Interface
- Support for this using [Topiarist library](#)

# Classes

```
// Define a class
function ArnoldRimmer() {
    this.name = 'Arnold Judas Rimmer';
}

// Define functions
ArnoldRimmer.prototype.answerExamPaper = function() {
    var answer = '';
    for( var i = 0; i < 500; ++i ) {
        answer += 'I am a fish.';
    }
    return answer;
};

var rimmer = new ArnoldRimmer();
```

# Inheritance

```
// Define a specialisation (subclass)
function AceRimmer() {
    ArnoldRimmer.call( this );

    this.name = 'Ace';
}
topiarist.extend( AceRimmer, ArnoldRimmer );

// Override functions
AceRimmer.prototype.answerExamPaper = function() {
    return 'Something very clever and brave.';
};
```

# Interfaces

```
// Define an Interface
function SmegHead() {
}
SmegHead.prototype.lie = function() {
  throw new Error( 'lie must be implemented by Class' );
};

// Override functions
function ArnoldRimmer() {
  this.name = 'Arnold Judas Rimmer';
}
// Arnold Judas Rimmer is a Smeg Head
topiarist.implement( ArnoldRimmer, SmegHead )

ArnoldRimmer.prototype.lie = function() {
  return 'I aced my engineering exam';
};
```

# OO Benefits

- Widely understood
- Developed for complex systems
- Reusable, maintainable, single responsibility principle
- Modularity (LSP)
- Simple, Consistent, Promotes SoC

**Use a consistent mechanism to define functionality**

# Libraries & Tooling

- OO
  - [Topiary](#) library to help with OOP
  - Shims
  - Multiple OO libraries available
  - Multiple libraries provide ways of achieving this
- Scaffolding: CLI to help generate classes, interfaces etc.
  - BRJS CommandPlugin<sup>†</sup>
  - [Yeoman](#) generators
  - [Loom](#)

<sup>†</sup> May offer this

# Modules / Packages / Namespaces / Libraries

No time :o(

# Application Architecture

# Application Structure

- Functional components
- Application plumbing & foundations
- Assets

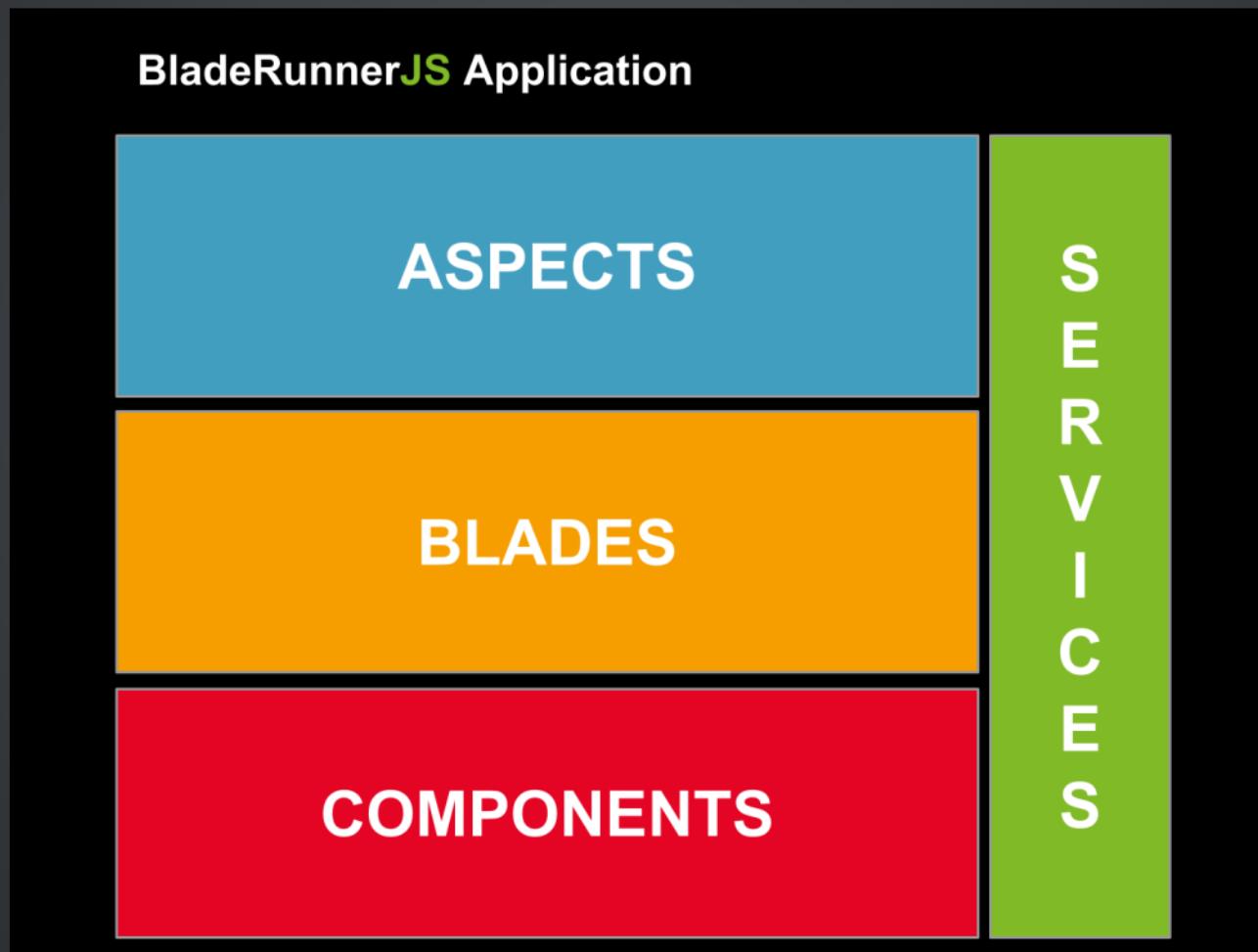
# Architecture Goals

a.k.a Buzz Terms

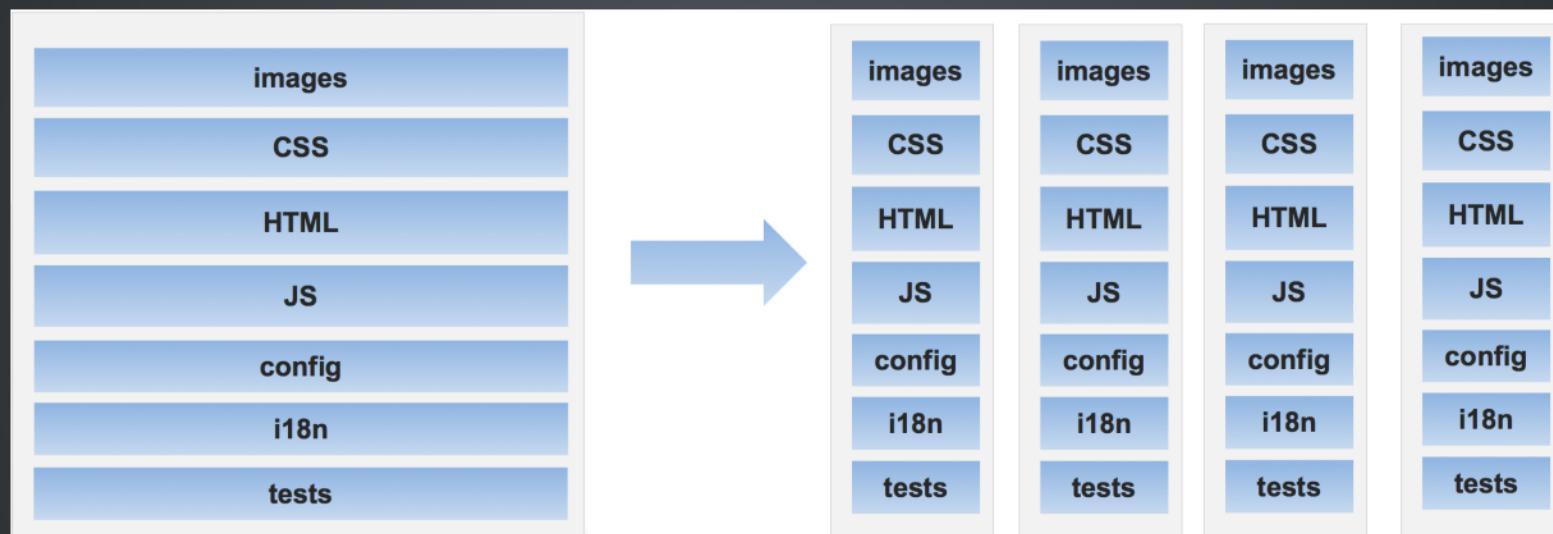
- Separation of Concerns
  - Code, Application Architecture and Assets
- Loosely coupling
- High cohesion

## Divide & Conquer

# Application Overview



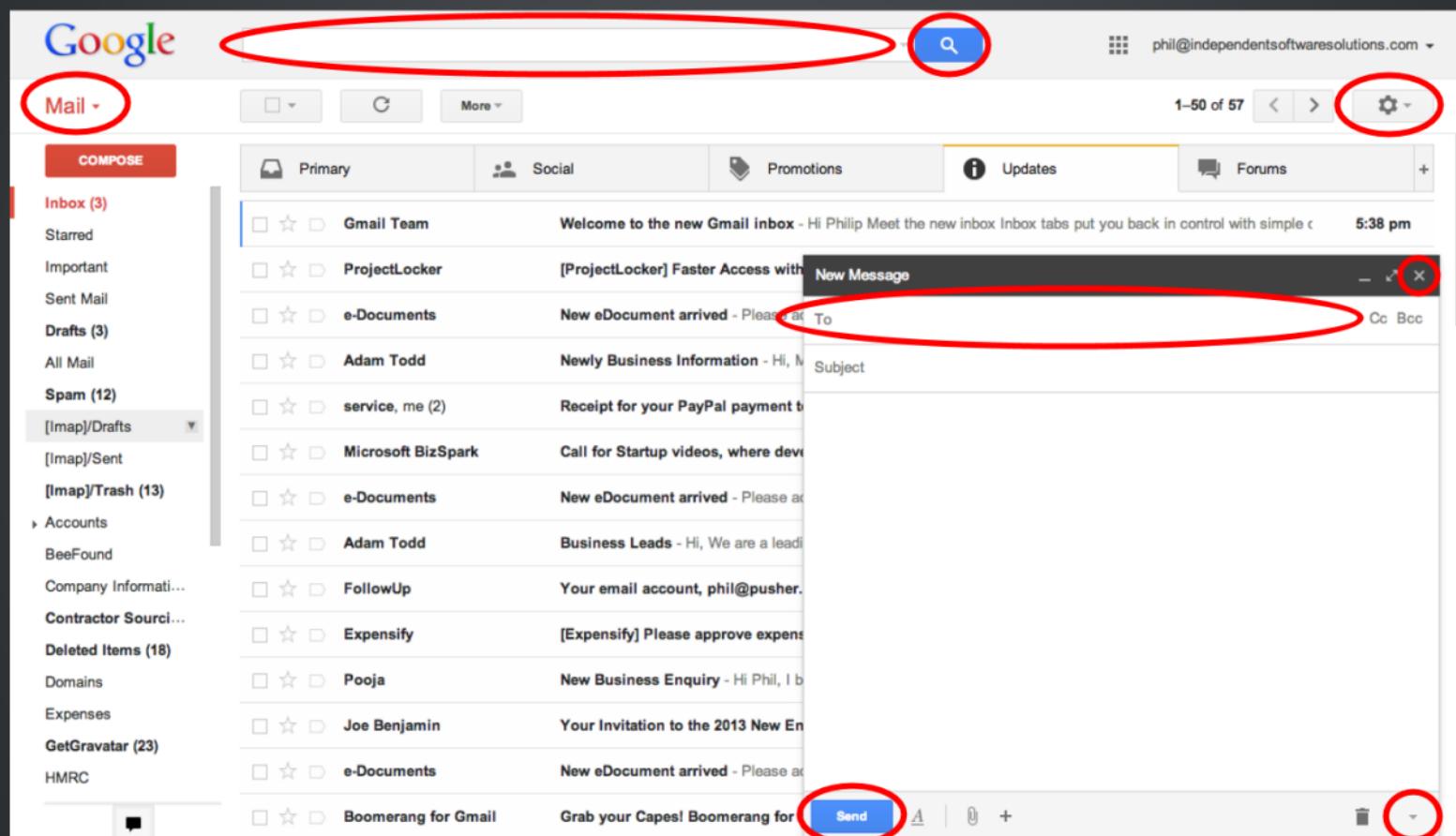
# Application Assets



## All those Assets!!!

- Managing the inclusion of these would be hard
- BRJS analyses dependencies and creates bundles

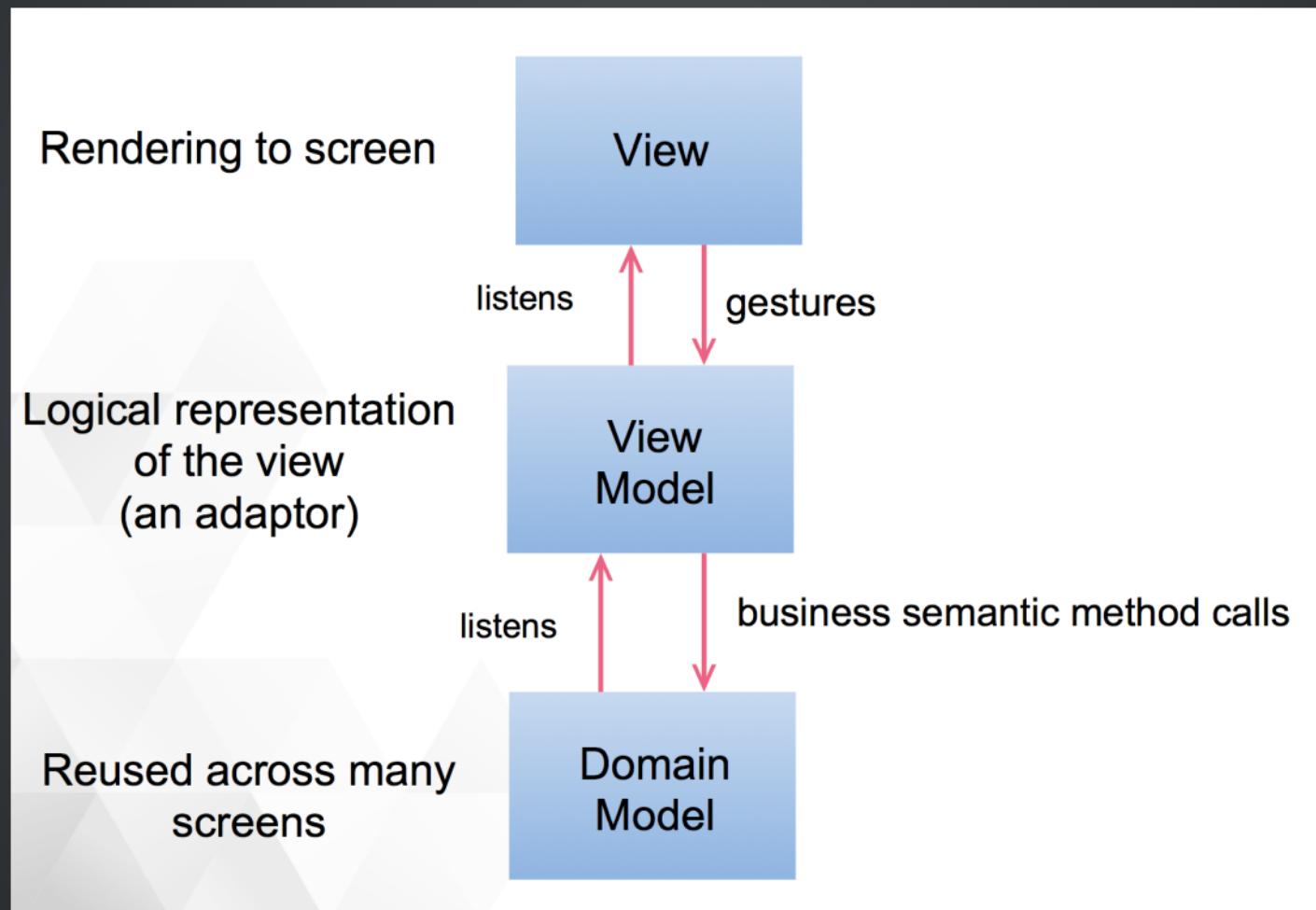
# Components



# MV\*

- MVC, MVP, MVVM
- We use Presenter, built on KnockoutJS
- Provides SoC

# MVVM



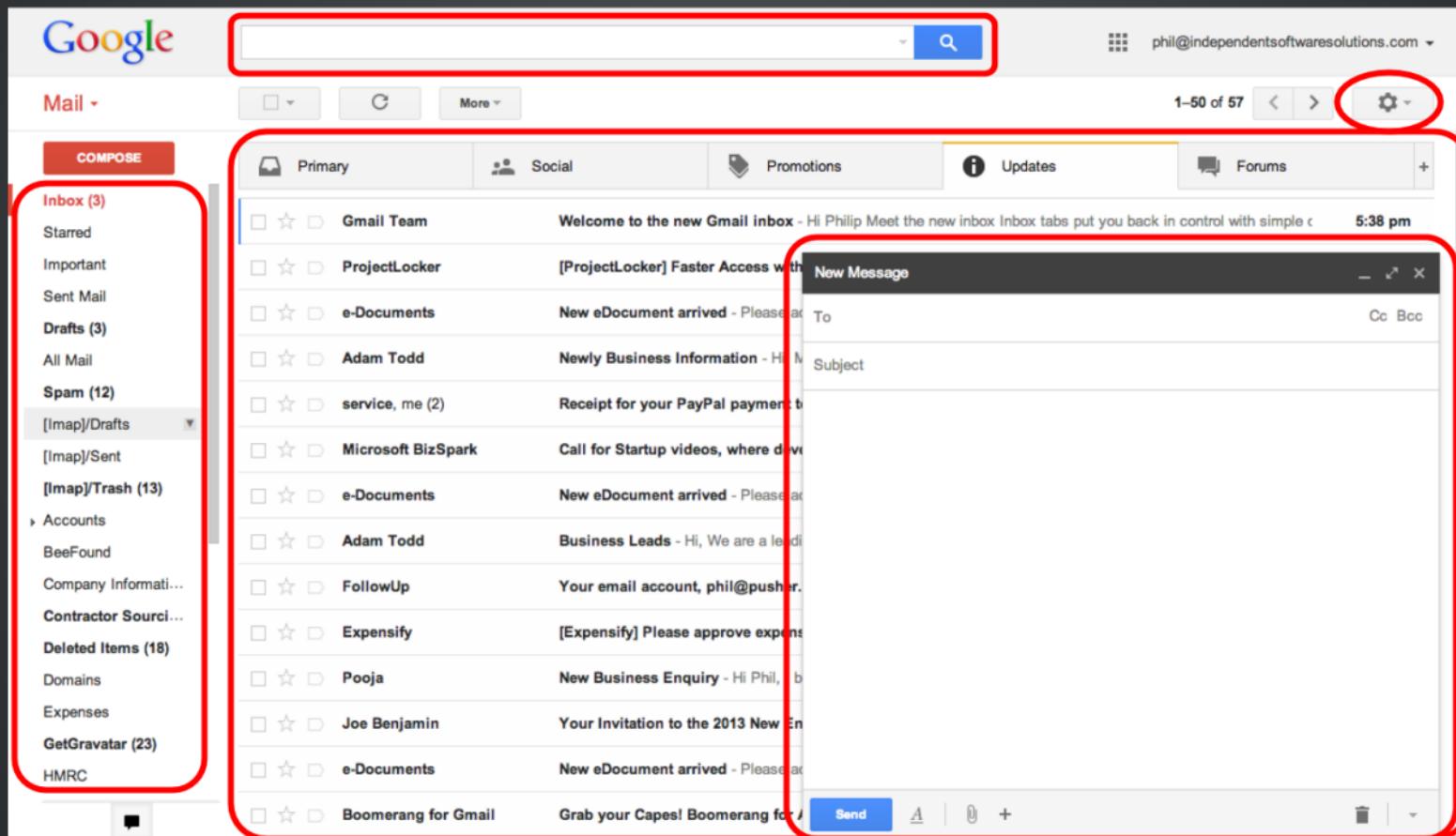
# Components

- Reusable
- Contain all required assets:
  - JavaScript
  - HTML
  - CSS
  - i18n
  - Other resources
- Composite components
  - Components containing other components
- MVVM / MV\*

# Tools, Libs & Other Component Solutions

- BRJS Presenter
- Web Components
- Polymer
- Montage
- Flight
- & the usual suspects: KnockoutJS, Backbone, Angular, Ember
  - But, these are more than just component libraries

# Blades



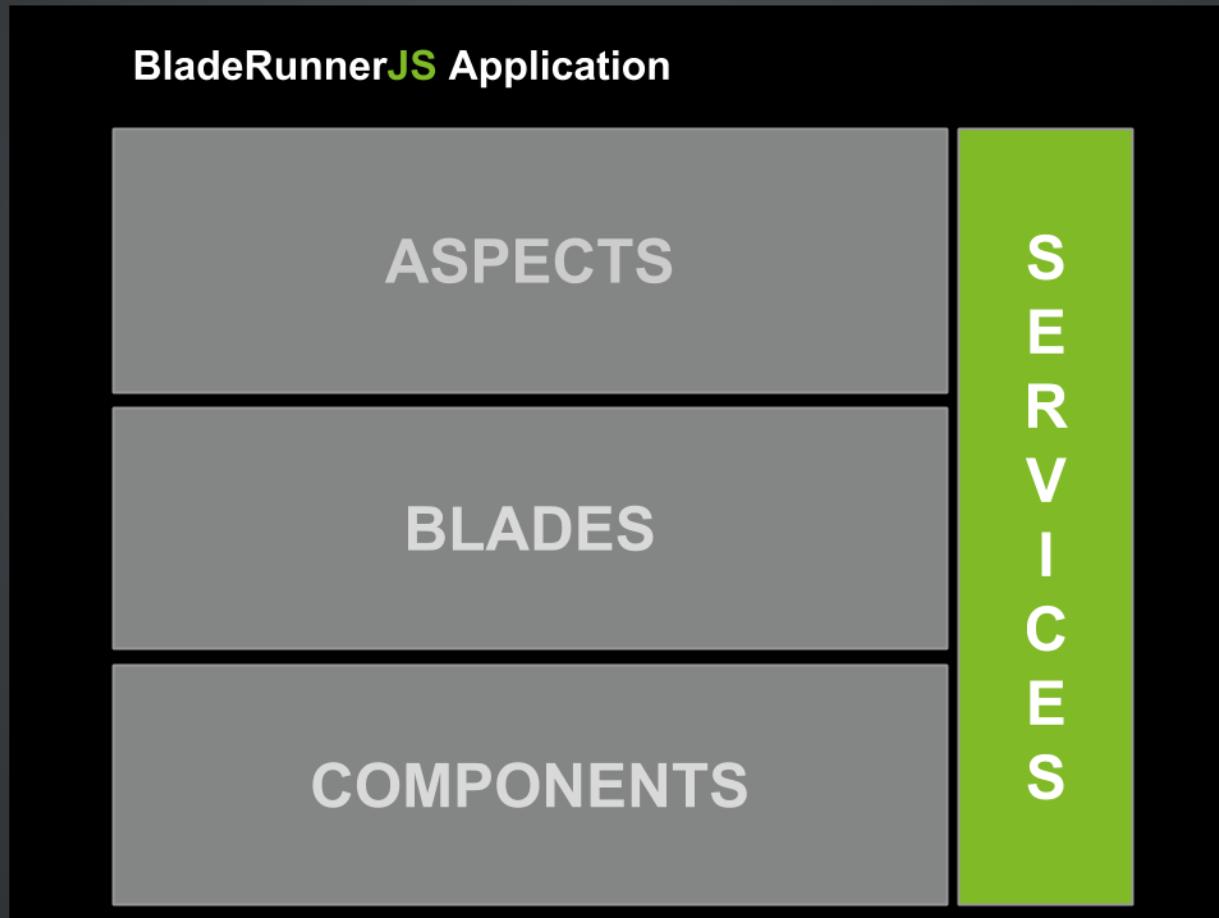
# Blades

- Reference components in libraries
- Define app-specific components
- Provide app-specific functionality
- Unlikely to have a defined API
- Contain all required assets:
  - JavaScript
  - HTML
  - CSS
  - i18n
  - Other resources
- MVVM / MV\*

# Libraries & Tooling

- A BRJS specific concept
- BRJS CLI has a `create-blade` command

# Services



# Services

- Use services to access resources
- Perform common non-UI tasks
- e.g.
  - PersistenceService
  - RealtimeService
  - LoggingService
- Services registered and accessed via a ServiceRegistry
- IoC / Dependency Injection

# Services

- Uses Interfaces

```
// Contact Service Interface
function ContactService() {
}

ContactService.prototype.search = function( value ) {
    throw new Error( 'search must be implemented by Class' );
};

ContactService.prototype.save = function( contact ) {
    throw new Error( 'save must be implemented by Class' );
};

ContactService.prototype.delete = function( contact ) {
    throw new Error( 'delete must be implemented by Class' );
};
```

# Setting Up Services

## Define Service

```
function RestfulContactService() {  
}  
topiarist.implements( RestfulContactService, ContactService );  
  
// implementation overrides
```

## Register Service

```
var ServiceRegistry = require( 'br/ServiceRegistry' );  
var RestfulContactService = require( 'myapp/RestfulContactService' );  
  
ServiceRegistry.registerService( 'contact-service',  
    new RestfulContactService() );
```

# Using Services (cont)

## Access & Use Service

```
var ServiceRegistry = require( 'br/ServiceRegistry' )

var contactService =
  ServiceRegistry.getService( 'contact-service' );

var rimmer = contactService.search( "Arnold Judas Rimmer" );
contactService.delete( rimmer );

var Contact = require( 'myapp/Contact' );
contactService.save( new Contact( 'Kristine Kochanski' ) );
```

# Services enable Loose Coupled Communication

- Interact with Interfaces and not implementation
- Accessed via ServiceRegistry (no new MyThing( ))
- Implementations can be swapped out
- We use this for in-app communication too

# Event Hub

- Communication between application components
- e.g. Inter-Blade Communication
- By basic PubSub message passing

# Tooling & Libraries

- Service Registry library
- Event Hub library
- [AngularJS Dependency Injection](#)
- [Marionette](#) offers optional IoC
- [SomaJS DI](#)
- [di.js](#)
- See "The Mediator Pattern" (EventHub)

# Bonus Dev Tooling

## BRJS Workbenches

- Run Blades in Isolation
- Interact with Blade via:
  - Services
  - View Model (MVVM)

**CAPLIN**

My Tiles Other Tile Set Other Tile Set

EURUSD +

**USDDKK** SP FW SW X

SELL USD 5.71 SPOT 655 20/12/12 SPOT 5.71 -2.5 405 27/12/12 1W GFA 8.888M 5.71 -4.75 180 03/01/13 2W GFA 8.888M 5.71 -4.25 185

SELL USD 5.70 -11.25 530 24/01/13 1M GFA 8.888M 5.71 -10.25 230 5.68 -26.25 805 21/02/13 2M GFA 8.888M 5.69 -26.25 605 5.66 -51.0 555 21/03/13 3M GFA 8.888M 5.67 -49.0 355 5.58 -134 255 20/05/13 5M GFA 8.888M 5.59 -124 855 5.48 -230 655 19/09/13 6M GFA 8.888M 5.51 -210 255

**USDCHF** SP FW SW X

SELL USD 0.96 45<sub>7</sub> BUY USD 0.96 65<sub>0</sub>

**EURJPY** SP FW SW X

SELL EUR 108. 63<sub>5</sub> BUY EUR 108. 85<sub>2</sub>

**USDMUR** SP FW SW X

SELL USD 5.71 SPOT 655 20/12/12 SPOT 5.71 -2.5 355 27/12/12 1W GFA 8.888M 5.71 -4.75 155 03/01/13 2W GFA 8.888M 5.71 -4.25 185

New Tile Set Open Tile Set Save Tile Set As...

1 Click Trading Standard Trading

Close All Tiles

**N. Europe** Europe Asia Sparkline

CCY Pair Bid Ask Sparkline

EURRUB ▲ 29.3423 ▼ 29.3643

EURRUB 29.3423 ▲ 29.3643

EURRUB 29.3423 29.3643

EURRUB ▲ 29.3423 ▼ 29.3643

EURRUB 29.3423 ▲ 29.3643

EURRUB 29.3423 29.3643

**Other Watchlist** My Watchlist

EURUSD +

CCY Pair Bid Ask Sparkline

EURRUB ▲ 29.3423 ▼ 29.3643

EURRUB 29.3423 ▲ 29.3643

EURRUB 29.3423 29.3643

EURRUB ▲ 29.3423 ▼ 29.3643

EURRUB 29.3423 ▲ 29.3643

EURRUB 29.3423 29.3643

News Other Tile Set

Top Stories

Barclays to shut wealth management services in 130 countries Reuters

Loans to eurozone businesses 'drop sharply' Reuters

As rupee hits hard times, India cajoles and woos offshore currency players Reuters

Japan shares rally, U.S. budget fuels caution elsewhere Reuters

Activity Blotter Historic Blotter Orders Positions

EURUSD +

EURUSD Other Tile Set Other Tile Set

1y Sep 25, 2012 - Sep 25, 2013

Open 1.3208

Vol. Nov '12 Jan '13 Mar '13 May '13 Jul '13 Sep '13

Blades-and-Motifs

Fx-Motif

Blade-FX-Tile

Workbench

Launch App

Run Tests

18

21

16

Panels



## FX Tile Workbench

- ▶ Outputs
- ▶ SL4B Console
- ▶ Connectivity
- ▶ Permissions

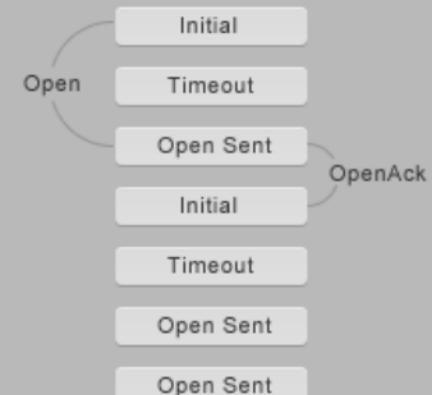


## ▼ Events

 OFF Auto Server Events

Client Side

Server Side



## ▼ Prices

 ON Stream Prices

Subscription

FX/Unidentified 21

- |              |  |  |                                     |
|--------------|--|--|-------------------------------------|
| FX/GBPUSD 2M | <input checked="" type="checkbox"/> Send Single Update |  | <input checked="" type="checkbox"/> |
| FX/GBPUSD 2M | <input type="checkbox"/>                               |  | <input type="checkbox"/>            |
| FX/GBPUSD 2M | <input checked="" type="checkbox"/>                    |  | <input checked="" type="checkbox"/> |
| FX/GBPUSD 2M | <input checked="" type="checkbox"/>                    |  | <input checked="" type="checkbox"/> |
| FX/GBPUSD 2M | <input type="checkbox"/>                               |  | <input type="checkbox"/>            |

# Aspects

- A view into the application
- Groups Blades together
- Example Aspects:
  - Desktop
  - Mobile
- Bundles are created for each Aspect on deployment

# Code Quality

## Code Quality Ground work

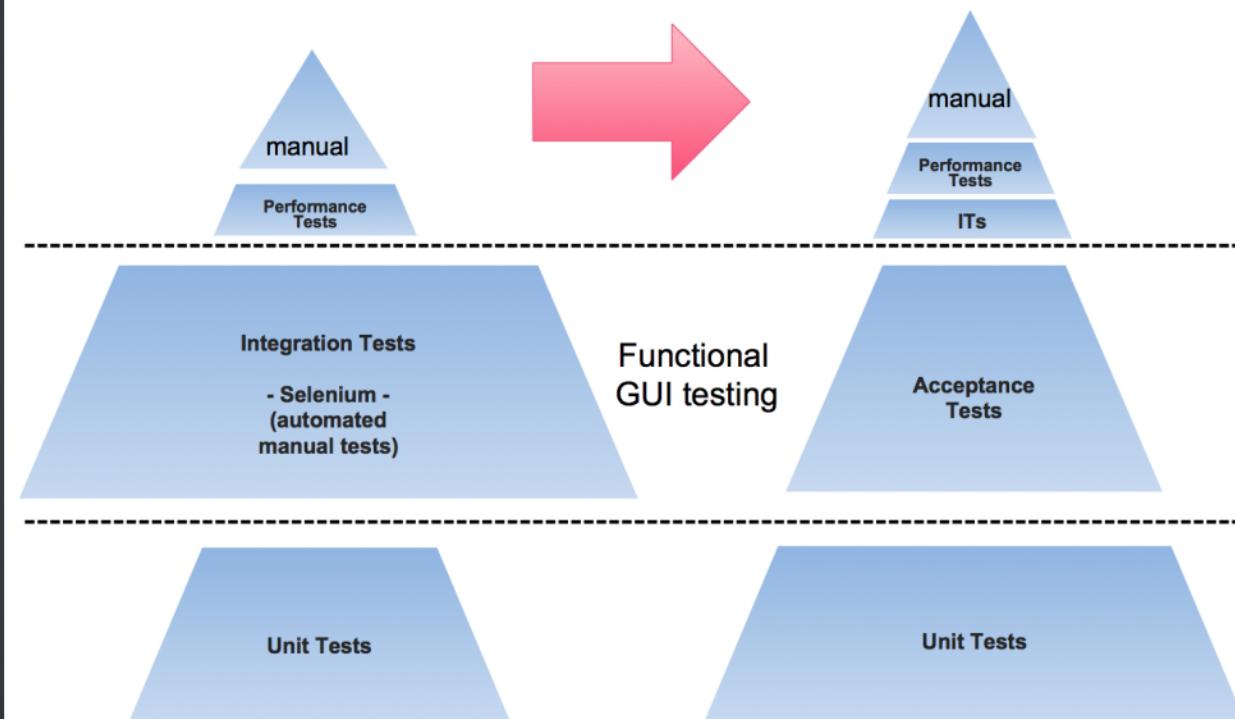
- Coding Style
- Code Structure
  - OOP
  - Modules, Packages, Libraries
- Architecture
  - Divide & Conquer
  - Services
  - Event Hub
- Asset Structure
  - Group by feature
  - Develop in isolation in Workbenches

And of course  
**Testing**

# Testing

- Unit Test All Classes
- Test App functionality (Blades) in isolation
  - View Model changes -> Mocked Services
  - Dummy Service interactions -> View Model
  - Blade UI in Web Browsers in isolation -> Mocked Services
- Enabled by ServiceRegistry
- Some full app ATs and Manual tests

# The Testing Triangle



## Benefits of Services and MVVM

- Full suite of tests used to take 24 hours
- Now takes less 30 minutes

# Tooling & Libraries

- BRJS
  - Presenter library for MVVM tests
  - ServiceRegistry for mocking services
  - BRJS CLI wraps JSTestDriver to execute tests
- Other
  - Any MVVM library
  - Karma test runner
  - Jasmin Node

# Conclusion

# Benefits of Tooling & Libraries

- Coding Style: enforcement
- Code Structure: automation and enforcement
- Architecture: automation and enforcement
- Asset Structure: automation and enforcement
- Code Quality: automation and enforcement

# Consistency, Simplicity, Separation of Concerns

- Complexity will increase
- Following Principles & Practices controls
  - Complexity & Consistency
- Divide & Conquer!
- Tooling & Libraries are the glue that makes things easier
- Enabling a Front-end JS App that Scales

# Thanks / Questions

Phil @leggetter

[phil@leggetter.co.uk](mailto:phil@leggetter.co.uk)

Caplin Systems

# BladeRunnerJS

@BladeRunnerJS