

Building Drag'n'Drop Interfaces

with Backbone

Who is this dude?

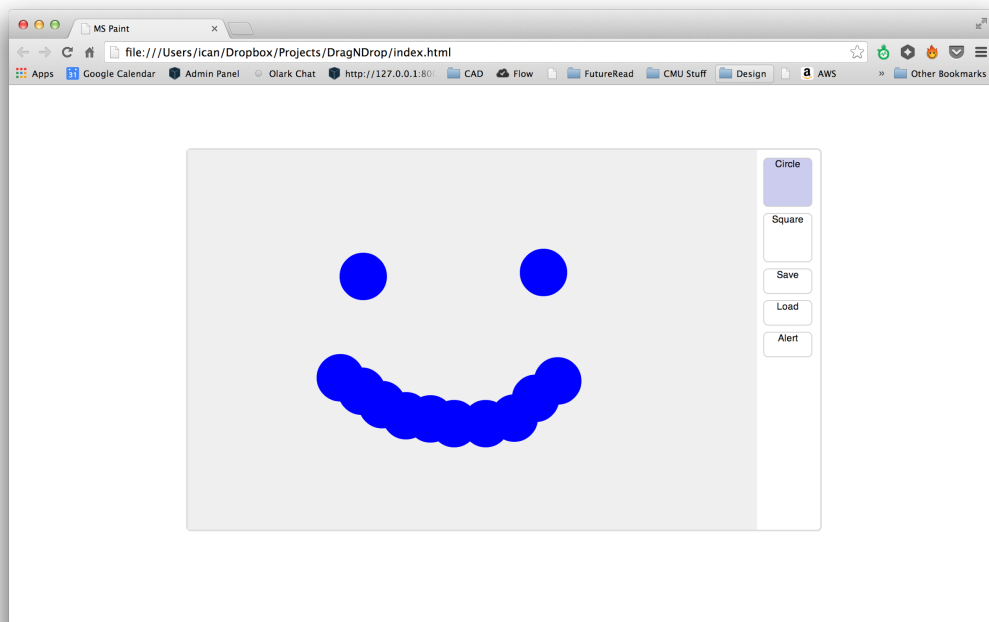
Why Backbone?

- It's light weight
- Doesn't do magic
- Awesome inheritance

Backbone let's you have your own architecture

Philosophical dogma doesn't interest me.

- Bob Dylan



I assume that:

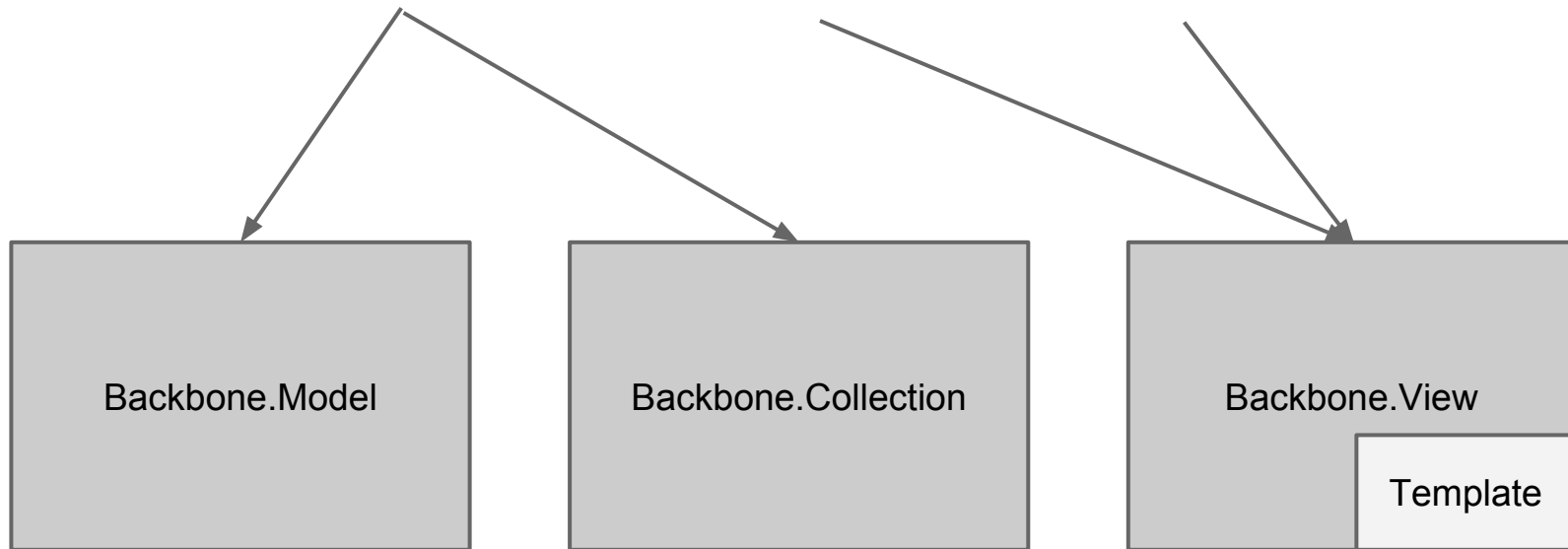
- You're know basics of Javascript and meaning of JSON
- You're know what Model, View, and Template means in general
- You're familiar with Backbone

<https://github.com/icanberk/DragNDrop>



BACKBONE.JS

Model - View - Template



Models (*{objects in json}*)

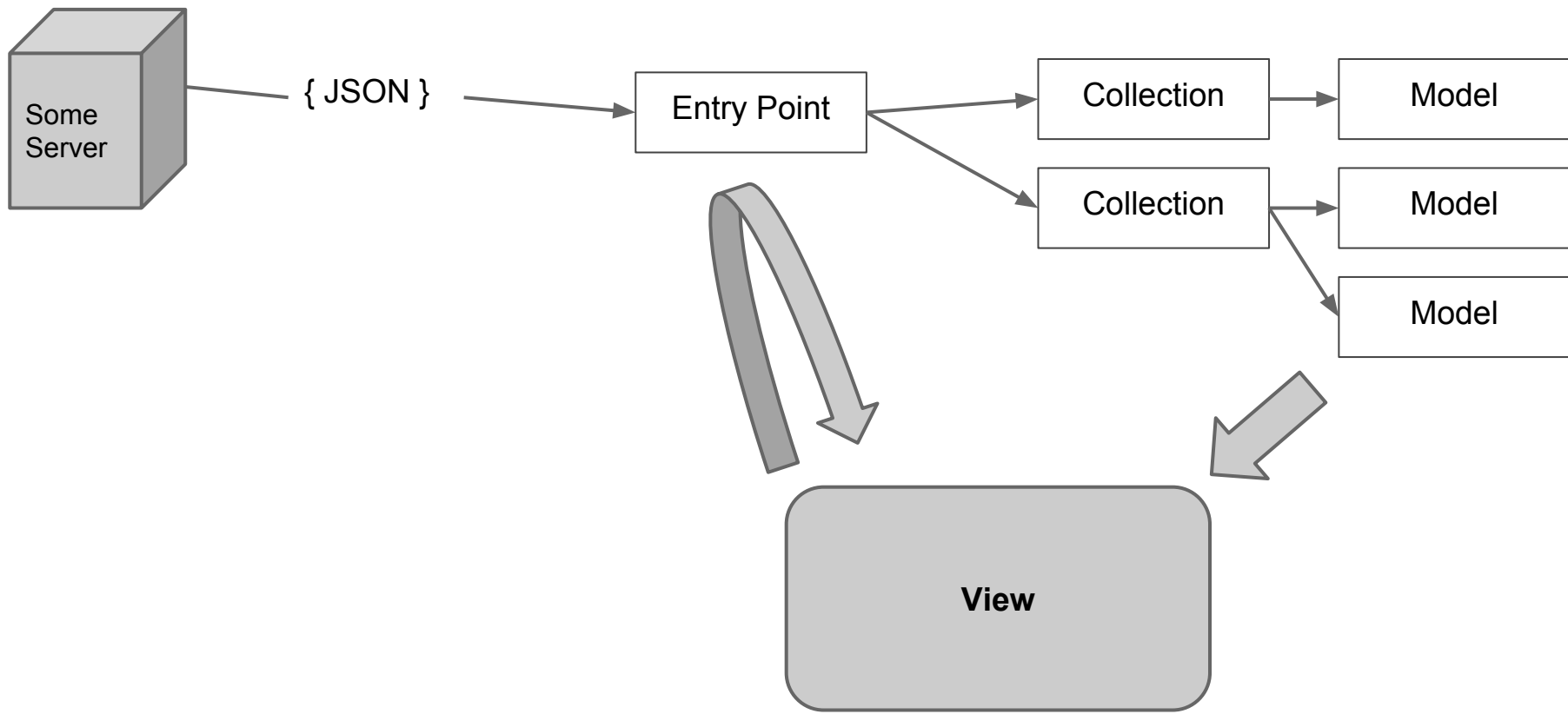
- They define the structure of the data.
- They have properties
- Properties can be primitives, other *models* or *collections*.

Collection (*[arrays in json]*)

- They contain and maintain set of models.
- There are pre-built functions to manipulate, or get models.
- You are also free to define your own

Views

- They render DOM elements and retain references to them.
- They generally correspond to one model. (e. g. ShapeModel -> ShapeView)
- They need to have an hierarchy of the layout. (superviews/subviews)



The JSON

Description of the views that we're trying to render.

Can be nested arrays and objects.

```
var initialBoard = {  
  "shapes": [  
    {  
      "xCoord": 192.5,  
      "yCoord": 130,  
      "type": "circle"  
    },  
    {  
      "xCoord": 420.5,  
      "yCoord": 125,  
      "type": "circle"  
    }  
  ]  
}
```

Entry Point - Main Model

```
var BoardModel = Backbone.Model.extend({

  initialize: function(bone) {
    var collection = new ShapeCollection(bone.shapes||[]);
    this.set('shapes', collection);
  },

  toJSON: function() {
    var json = _.clone(this.attributes);
    // need to call toJSON on nested collections and models
    json.shapes = json.shapes.toJSON();
    return json;
  }

});
```

Views

BoardView

```
renderShapes: function() {  
    this.model.get('shapes').each(function(shapeModel) {  
        this.renderShape(shapeModel);  
    }, this);  
},
```

CircleView

className: 'shape-widget
circle',

ShapeView

className: 'shape-widget
square',

```
render: function() {  
    this.el.style.left = this.model.get('xCoord') + 'px';  
    this.el.style.top = this.model.get('yCoord') + 'px';  
  
    return this;  
},
```

Templating

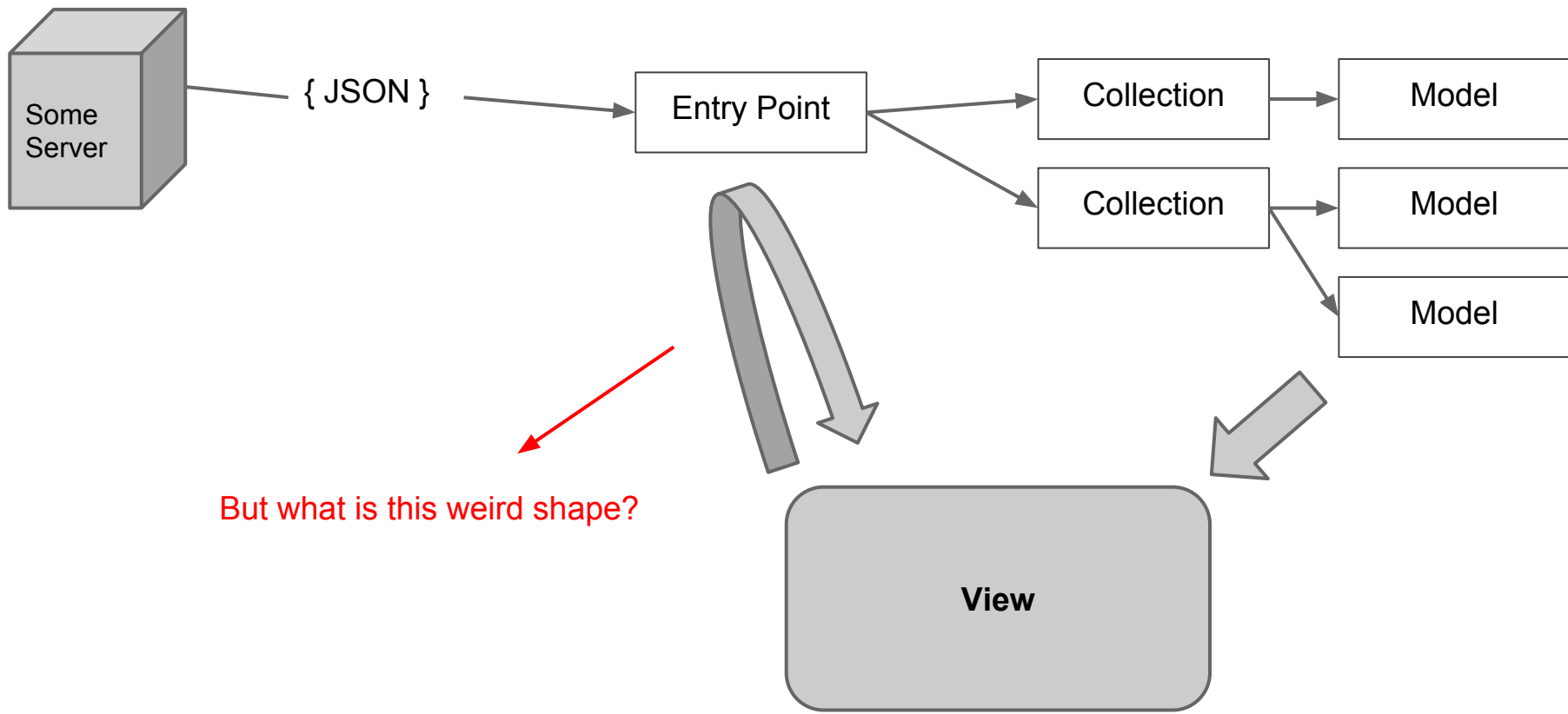
_.template is all you need.

```
var templateStr = [  
  '<% _.each(shapes, function(shapeStr) { %>',  
  '<div class="<%= shapeStr %> shape"><%= shapeStr %></div>',  
  '<% }); %>',  
  '<div class="save btn">Save</div>',  
  '<div class="load btn">Load</div>',  
].join('\n');  
  
var availableShapes = ['circle', 'square'];  
  
this.el.innerHTML = _.template(templateStr, { shapes: availableShapes});
```

How does jQuery help?

```
this.$el.find( ".shape" ).draggable({  
  helper: "clone",  
  stop: this.shapeDropped  
});
```

```
shapeDropped: function(e) {  
  var type = e.target.dataset.type;  
  var y = this.findTop(e);  
  var x = this.findLeft(e);  
  
  this.model.addShape(type, x, y);  
},
```

Observer Pattern = Less Coupling

BackboneDemo / js / views / BoardView.js

```
initialize: function() {  
  _.bindAll(this);  
  
  this.model = new BoardModel({});  
  
  // listeners (read(or ask me) more about observer pattern)  
  this.listenTo(this.model.get('shapes'), 'add', this.renderShape);  
  
}
```

I can just remove this BoardView module, and nothing in the codebase will break.

How to put these modules together?

```
<script src="js/libs/module1.js"></script>
```

```
<script src="js/libs/module2.js"></script>
```

```
<script src="js/libs/module3.js"></script>
```

```
.
```

```
.
```

```
.
```

```
.
```

```
.
```

```
.
```

```
.
```

```
.
```

```
.
```

```
.
```

```
.
```

```
<script src="js/libs/module99.js"></script>
```

```
<script src="js/libs/module100.js"></script>
```



Do I know if module60.js is loaded?

File Tree from Appcubator code-base

```
|— AnalyticsView.js
|— AppInfoView.js
|— AppRouter.js
|— CollaboratorsView.js
|— DeployView.js
|— DownloadModalView.js
|— GarageView.js
|— InvitationsView.js
|— OverviewPageView.js
|— PluginsView.js
|— QuickTour.js
|— RedoController.js
|— RouteLogger.js
|— ShareModalView.js
|— SimpleTwitterTour.js
|— SoftErrorView.js
|— Striper.js
|— ThemeDisplayView.js
|— ThemesGalleryView.js
|— TwitterTour.js
|— WorldView.js

|— dicts
| |— constant-containers.js
| |— default-UIelements.js
| |— page-templates.js
|— editor
| |— CustomWidgetEditorModal.js
| |— EditorGallerySectionView.js
| |— EditorGalleryView.js
| |— EditorView.js
| |— FacebookShareEditor.js
| |— FooterEditorView.js
| |— FooterView.js
| |— GuideView.js
| |— ImageSliderEditorView.js
| |— KeyDispatcher.js
| |— LinkEditorView.js
| |— MarqueeView.js
| |— MouseDispatcher.js
| |— MultiSelectorView.js
| |— NavbarEditorView.js
| |— NavbarView.js

| |— PageTemplatePicker.js
| |— PickCreateFormEntityView.js
| |— QueryEditorView.js
| |— SearchEditorView.js
| |— ToolbarView.js
| |— VideoEmbedEditor.js
| |— WidgetClassPickerView.js
| |— WidgetContainerView.js
| |— WidgetContentEditorView.js
| |— WidgetCustomView.js
| |— WidgetEditorView.js
| |— WidgetEditorViewProxy.js
| |— WidgetFormView.js
| |— WidgetLayoutEditorView.js
| |— WidgetListView.js
| |— WidgetSelectorView.js
| |— WidgetView.js
| |— WidgetsManagerView.js
| |— editor-templates.js
| |— form-editor
```

.... Could not fit all

Require.js

```
define(function(require, exports, module) {
```

```
    'use strict';
```

```
    require('backbone');
```

```
    var ShapeModel = require('data/ShapeModel');
```

```
    var CircleView = require('views/CircleView');
```

```
    var SquareView = require('views/SquareView');
```

Dealing with the Memory Leaks

Checkout Backbone.LayoutManager: <https://github.com/tbranyen/backbone.layoutmanager>

BONUS SLIDE

Why Backbone

Modularity - AMD - Require

Entry Point Module - Controller

Inheritance in Backbone - Mixins

Views - Backbone.LayoutManager

Templating - Benchmarking

jQuery-UI