



DRUPALCON

Portland

MAY 20-24 2013

Using Backbone.js with Drupal 7 and 8

VADIM MIRGOROD

Front-end, 05/22/2013

Vadim Mirgorod



- Lead developer in Trelon



- Writing Backbone.js CookBook for PACKT
- Email: dealancer@gmail.com
- Web: <http://vmirgorod.name>
- Twitter: @dealancer



**Remember the web
development in 90s?**

1995: JavaScript



2000: XMLHttpRequest



2006: jQuery



2013:



What's new in JS?

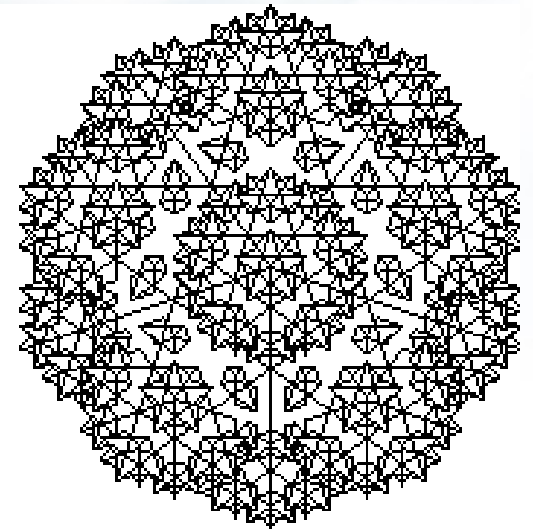
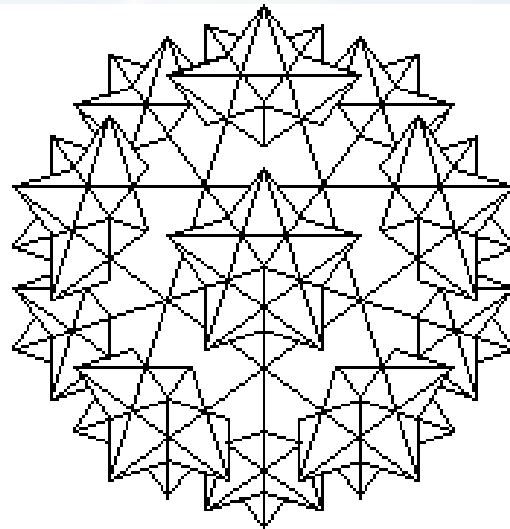
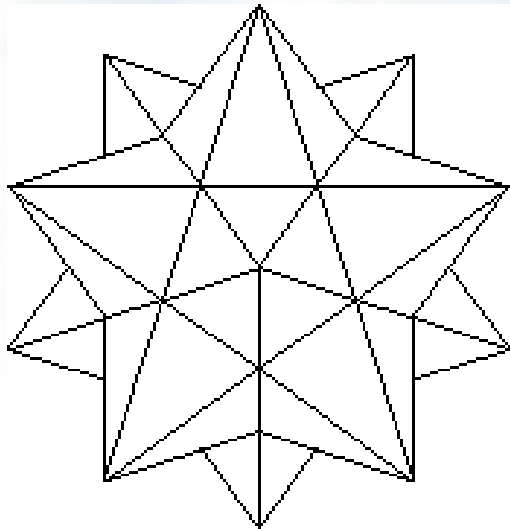


- HTML5:
 - Local Storage
 - pushState
- JS templating engines:
 - Mustache.js
 - Twig.js
- Representative State Transfer (REST)

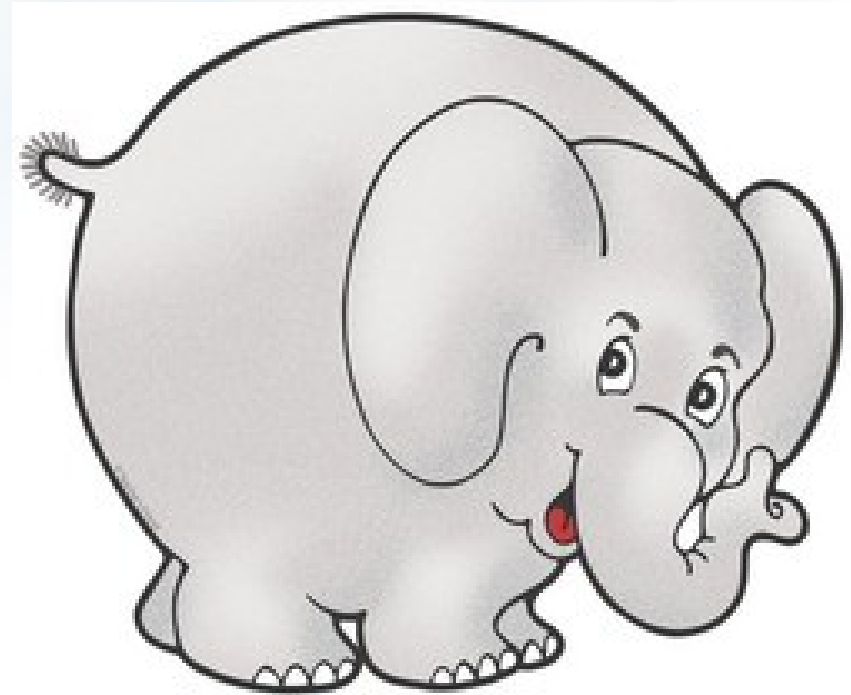
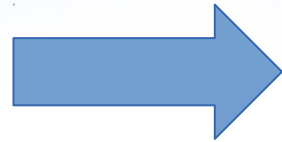


JavaScript client evolution

Complexity



Thin to thick transition



Performance





Problems!

Right now typical AJAX code looks like this



```
$(document).ready(function() {  
    $("#getData").click( function() {  
        $.getJSON("artistsRemote.cfc?method=getArtists&returnformat=json",  
            function(data) {  
                $("#artistsContent").empty();  
                $("#artistsTemplate").tmpl( data ).appendTo("#artistsContent");  
                var nowIs = new Date().toLocaleString();  
                $('#lastLoad').html( nowIs );  
            });  
    });  
});
```

AJAX in Drupal



Have you seen any JS?





Let's do things properly!



Meet...



BACKBONE.JS



Backbone.js



- URL: <http://backbonejs.org/>
- Created by Jeremy Ashkenas in 2010, an author of CoffeeScript
- Based on Underscore.js:
<http://backbonejs.org/>
- Requires jQuery or Zepto

Backbone.js features



- Minimalistic
- Modular
- Perfect OOP design
- Over 100 available extensions:

<https://github.com/documentcloud/backbone/wiki/Extensions,-Plugins,-Resources>

- Community

Backbone vs...



- Knockout
- AngularJS
- CanJS
- Spine
- Ember.js
- many others

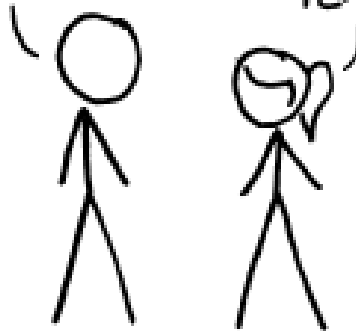
Backbone vs...



HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC)

SITUATION:
THERE ARE
14 COMPETING
STANDARDS.

14?! RIDICULOUS!
WE NEED TO DEVELOP
ONE UNIVERSAL STANDARD
THAT COVERS EVERYONE'S
USE CASES.



SOON:

SITUATION:
THERE ARE
15 COMPETING
STANDARDS.

Who uses Backbone.js?



- Groupon Now!
- Foursquare
- LinkedIn Mobile
- Airbnb



**Let's learn how to
backbone!**

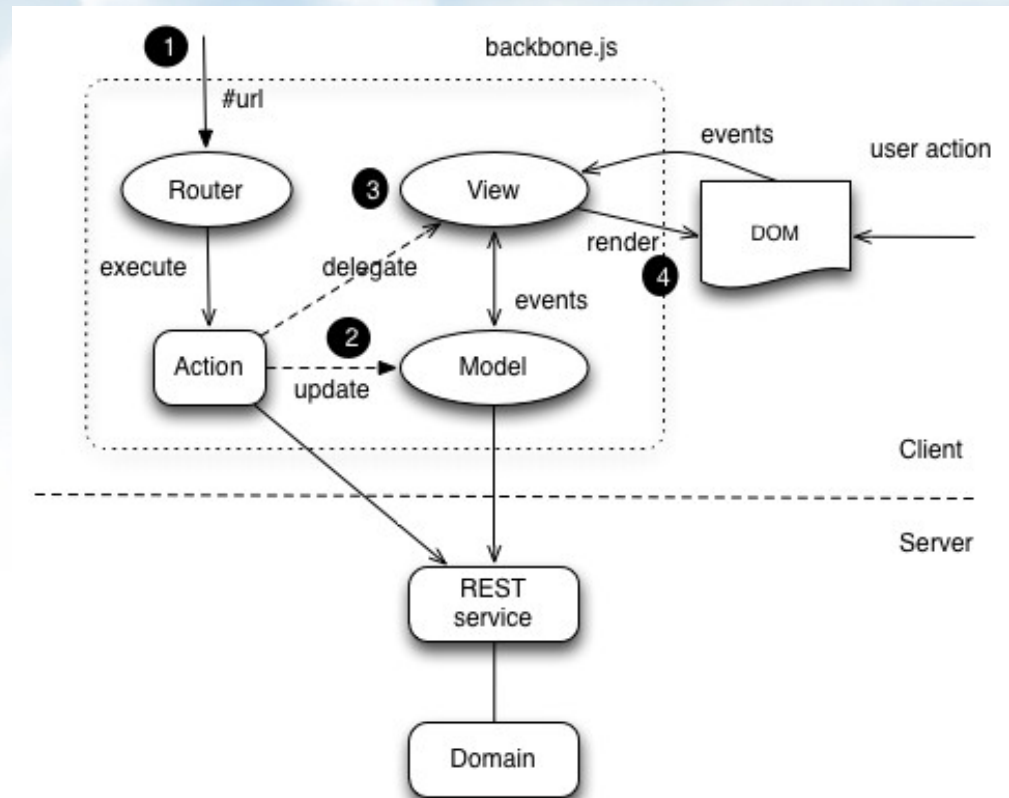
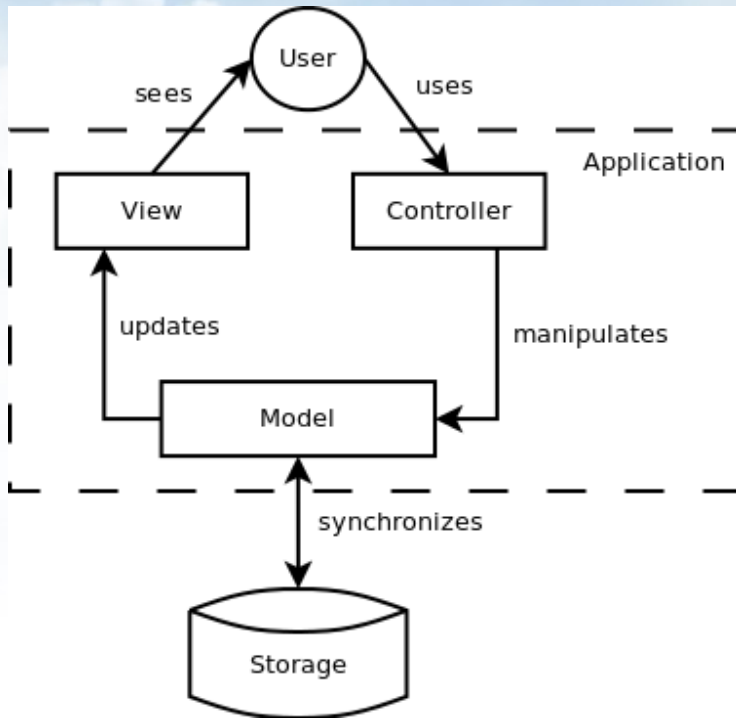
Main concepts



- Model
- Collection
- View
- Template
- Router
- History
- Events

MVC:

Backbone:





DEMO!

TUDO APP

Easy example



Quantity	Description	Price	Amount
3	Wooden Toy House	22	66
1	Farm Animal Set	17	17

Define the model and the collection



```
// Define new model.  
var InvoiceItemModel = Backbone.Model.extend({  
  calculateAmount: function() {  
    return this.get('price') * this.get('quantity');  
  }  
});  
  
// Define new collection object.  
var InvoiceItemCollection = Backbone.Collection.extend({  
  model: InvoiceItemModel  
});
```

Define the view to render the model



```
var InvoiceItemView = Backbone.View.extend({
  // Define element tag name.
  tagName: 'tr',

  // Define template.
  template: _.template($('#item-row-template').html()),

  // Render the view.
  render: function() {
    var data = this.model.toJSON();
    data.amount = this.model.calculateAmount();
    $(this.el).html(this.template(data));
    return this;
  },
});
```

Define the view to render the collection



```
var InvoiceItemListView = Backbone.View.extend({
  tagName: 'table',
  template: _.template($('#item-table-template').html()),

  // Render the view.
  render: function() {
    $(this.el).html(this.template());
    _.each(this.collection.models, function(model, key) { this.append(model); }, this);
    return this;
  },

  // Add an invoice item row to the table.
  append: function(model) {
    $(this.el).append(new InvoiceItemView({ model: model }).render().el);
  }
});
```

Add templates into index.html



```
<script type="text/html" class="template" id="item-table-template">
  <tr>
    <th>Quantity</th>
    <th>Description</th>
    <th>Price</th>
    <th>Amount</th>
  </tr>
</script>
```

```
<script type="text/html" class="template" id="item-row-template">
  <td><%= quantity %></td>
  <td><%= description %></td>
  <td><%= price %></td>
  <td><%= amount %></td>
</script>
```

Create collection and show the view



```
var invoiceItemCollection = new InvoiceItemCollection([
  { description: 'Wooden Toy House', price: 22, quantity: 3 },
  { description: 'Farm Animal Set', price: 17, quantity: 1 },
]);

$('body').html(new InvoiceItemListView({
  collection: invoiceItemCollection }).render().el
);
```


Some cool things



Model to view binding

Two way binding

Forms

Quantity	Description	Price	Amount
3	Wooden Toy House	22	66
1	Farm Animal Set	17	17

Layouts

Router

Templates

Model to view binding



```
var InvoiceItemView = Backbone.View.extend({
  // ...
  initialize: function() {
    this.listenTo(this.model, 'destroy', this.destroy, this);
    this.listenTo(this.model, 'change', this.render, this);
  },
  destroy: function() { this.remove(); }
});
```

```
var InvoiceItemListView = Backbone.View.extend({
  // ...
  initialize: function() {
    this.listenTo(this.collection, 'add', this.append, this);
  }
});
```

Forms



```
// Backbone-forms extension.
var UserModel = Backbone.Model.extend({
  schema: {
    title:      { type: 'Select', options: ['Mr', 'Mrs', 'Ms'] },
    name:       'Text',
    email:      { validators: ['required', 'email'] },
    password:   'Password',
    birthday:   'Date',
  }
});

userModel = new BuyerModel();

$('body').append(new Backbone.Form({ model: this.user }).form.render().el);
```

Two-way binding



```
// Backbone.stickit extension.  
var InvoiceItemFormView = Backbone.View.extend({  
  className: 'invoice-item-form-view',  
  bindings: {  
    '#description': 'description',  
    '#price': 'price',  
    '#quantity': 'quantity'  
  },  
  render: function() {  
    var html = '<label>Description:</label><input type="text" id="description"></input><br>' +  
      '<label>Price:</label><input type="text" id="price"></input><br>' +  
      '<label>Quantity:</label><input type="text" id="quantity"></input><br>';  
    $(this.el).html(html);  
    // Here binding occurs.  
    this.stickit();  
    return this;  
  }  
});
```

Forms



```
// Backbone-forms extension.
var UserModel = Backbone.Model.extend({
  schema: {
    title:      { type: 'Select', options: ['Mr', 'Mrs', 'Ms'] },
    name:       'Text',
    email:      { validators: ['required', 'email'] },
    password:   'Password',
    birthday:   'Date',
  }
});

userModel = new BuyerModel();

$('body').append(new Backbone.Form({ model: this.user }).form.render().el);
```

Router



```
var Workspace = Backbone.Router.extend({
  routes: {
    // Default path.
    '': 'invoiceList',
    // Usage of static path.
    'invoice': 'invoiceList',
    // Usage of fragment parameter.
    'invoice/:id': 'invoicePage',
    // Usage of fragment parameters.
    'help/:topic/page:page': 'helpPage',
    // Usage of splat parameter.
    'download/*path': 'downloadPage'
  },
});
```


Other cool things...



in Backbone.js Cookbook



Drupal™



Backbone and Drupal 7



- **Bootstrapping:**
 - Technique we saw above
 - Do it yourself
- **Representational State Transfer:**
 - Services module
 - RESTful webservices module
- **Backbone.js module**

REST



Resource URI	Collection:	Element:
	http://example.com/resources/items	http://example.com/resources/items/1
POST	Creates new item in the collection. Returns newly created item or it's URI.	Is not typically used. If used does the same as POST query for collection resource.
GET	Lists collection items or theirs URIs.	Retrieves collection item by it's URI.
PUT	Replaces the entire collection with another collection.	Replaces collection item or creates one if it does not exist.
DELETE	Deletes the entire collection.	Deletes item from the collection.

REST in Backbone



```
var PostModel = Backbone.Model.extend({
  // Override id attribute.
  idAttribute: '_id',
  // Define URL root to access model resource. Otherwise use
  // url() method to provide full path to the model resource.
  urlRoot: function() { return 'http://example.com/posts/'; }
});
```

```
var PostCollection = Backbone.Collection.extend({
  model: PostModel,
  // Define path to the collection resource.
  url: function() { return 'http://example.com/posts/'; }
});
```

REST in Backbone.js



```
// Fetches data into a model.  
model.fetch();  
  
// Saves a model.  
model.save();  
  
// Destroys a model.  
model.destroy();  
  
  
// Fetches data into a collection.  
collection.fetch();  
  
// Adds models to a collection.  
collection.add(models);  
  
// Removes specific models from collection.  
collection.remove(models);
```

Backbone module



- URL: <http://drupal.org/project/backbone>
- Provides models and collections for Drupal entities via REST:
 - Node: Node model
 - All node: NodeIndex collection
 - Arbitrary view: NodeView collection
- Works with both Services and RESTful Web Services modules.

Backbone module



```
// Create new NodeView collection.  
var viewCollection = new Drupal.Backbone.Collections.NodeView();  
  
// Set Drupal View name.  
viewCollection.viewName = 'backbone_example';  
  
// Fetch data from the collection.  
viewCollection.fetch({success: function() {  
    console.log(viewCollection.toJSON());  
}});
```

Backbone mdl roadmap



- Support for bootstrapping
- Better views support
- In-place content editing
- Drag and Drop
- D8 version?



<http://www.flickr.com/photos/gabblebee/3977912731/>

Backbone and Drupal 8



- It is in core!
- Used for In-place editing issue:
<http://drupal.org/node/1824500>
- Used for layouts issue:
<http://drupal.org/node/1841584>
- Used for toolbar issue:
<http://drupal.org/node/1860434>
- META issue: <http://drupal.org/node/1858368>



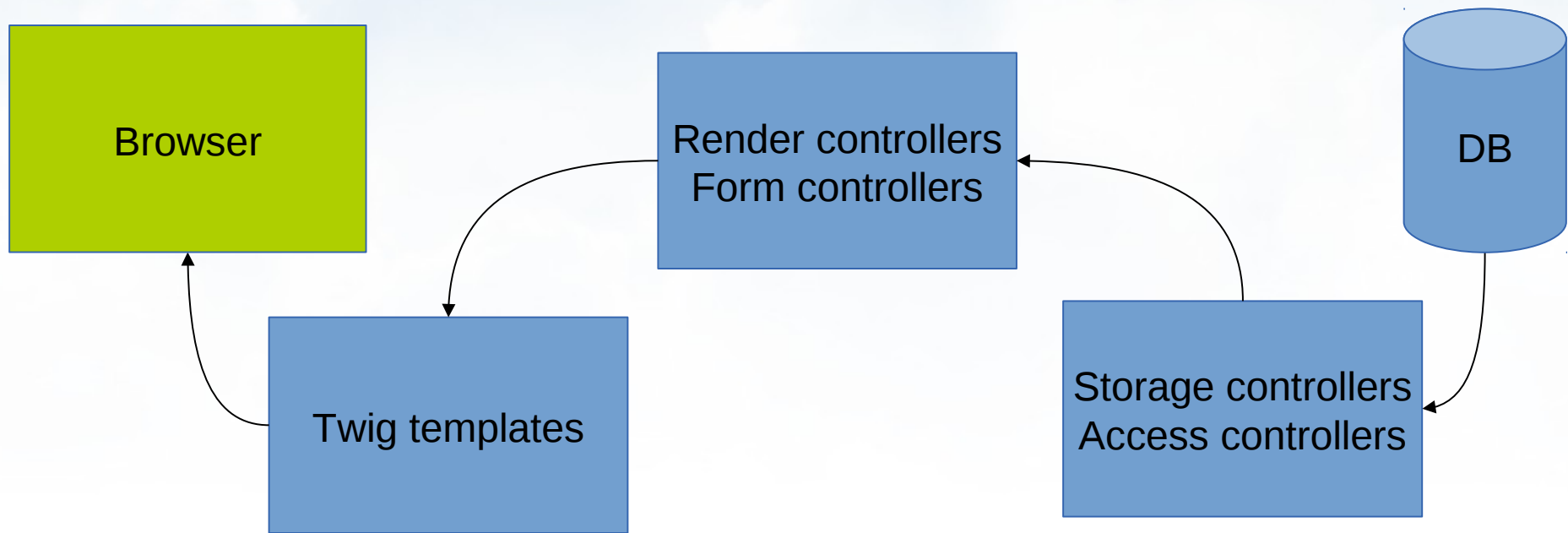
DEMO!

One more thing!

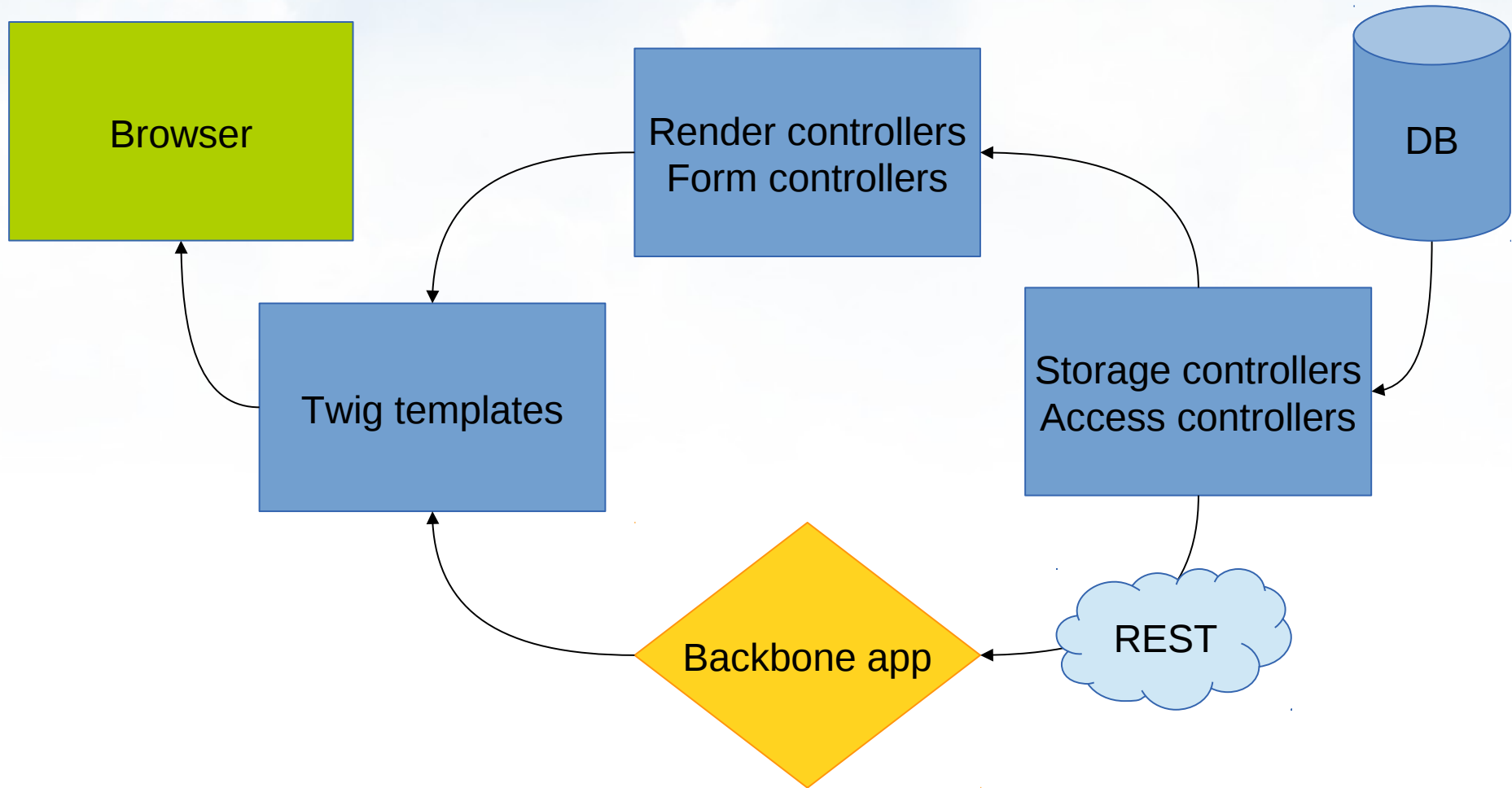


- Web services initiative
 - REST in core
 - Storage controllers
- Twig
 - Templating engine
 - Twig in core (Twig sandbox)
 - Works both for PHP and JS

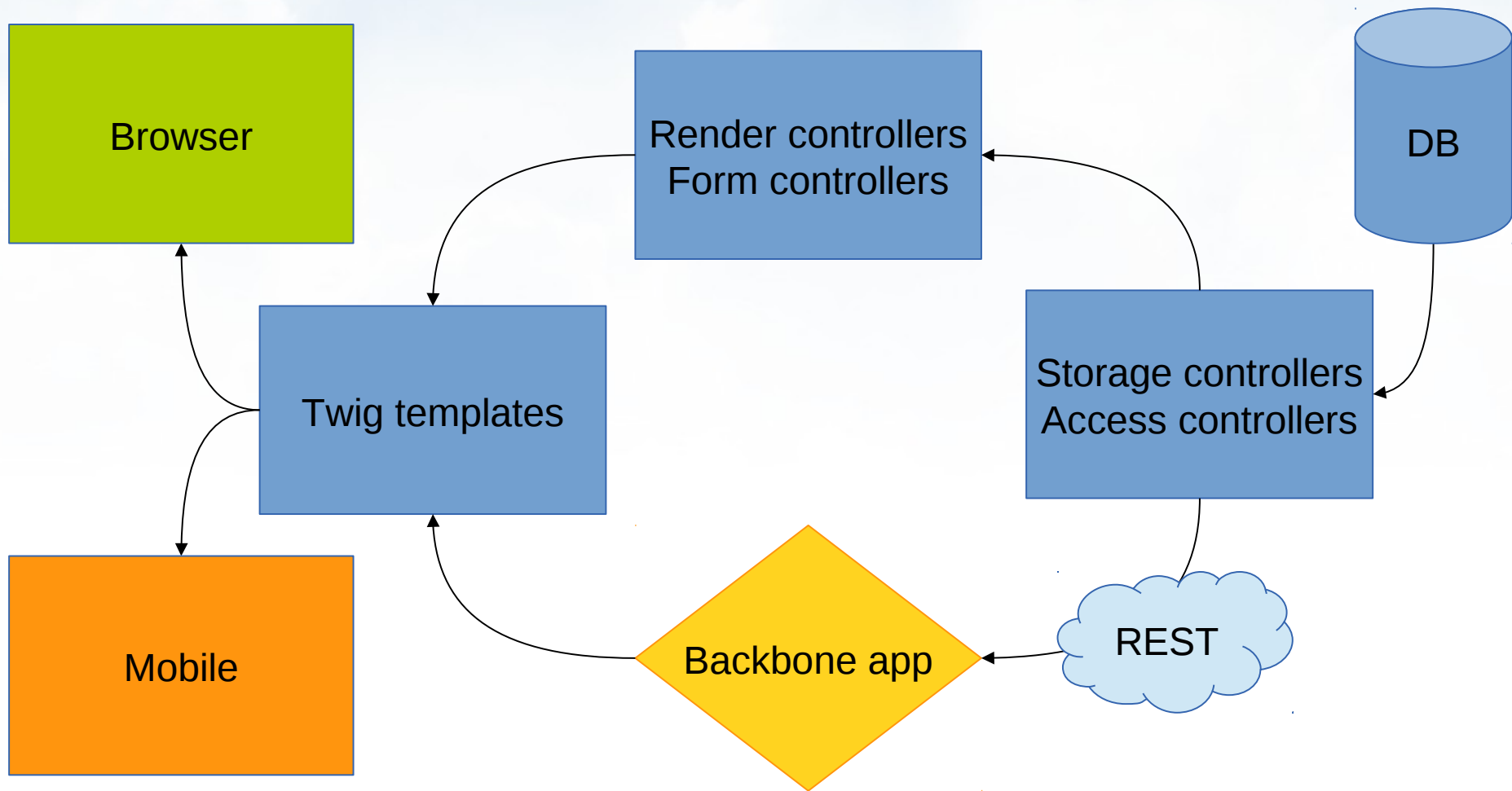
One more thing!



One more thing!



One more thing!



One more thing!



Mobile app built with PhoneGap or Trigger.io

Twig templates

Backbone app

Storage controllers
Access controllers

REST

DB



DRUPALCON

Portland

MAY 20-24 2013

What did you think?

Evaluate this session at:

[portland2013.drupal.org/node/1578.](http://portland2013.drupal.org/node/1578)

Thank you!