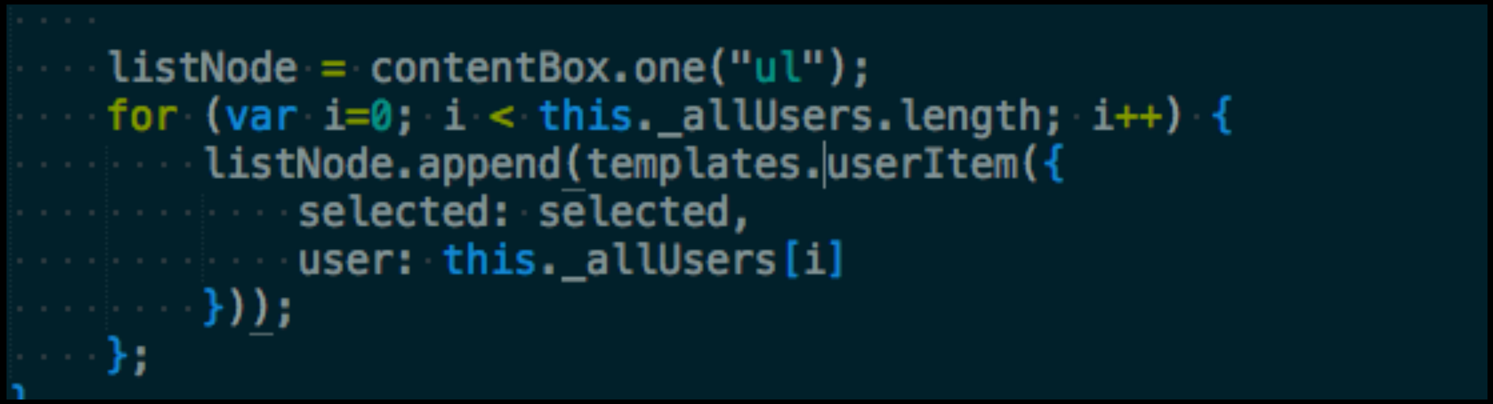
Flux + React

Rick Mak
May 2015

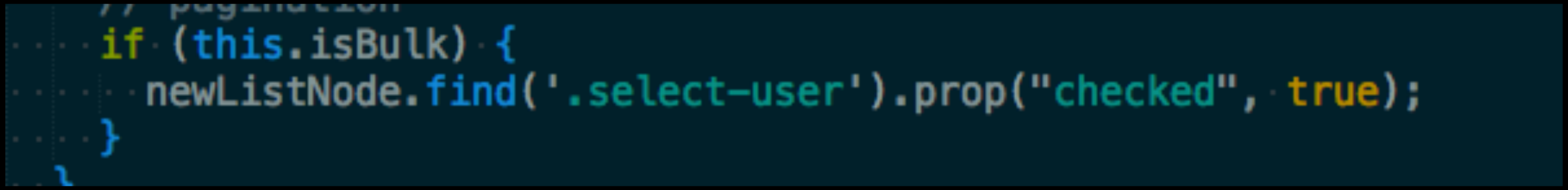
One of the pain

DOM

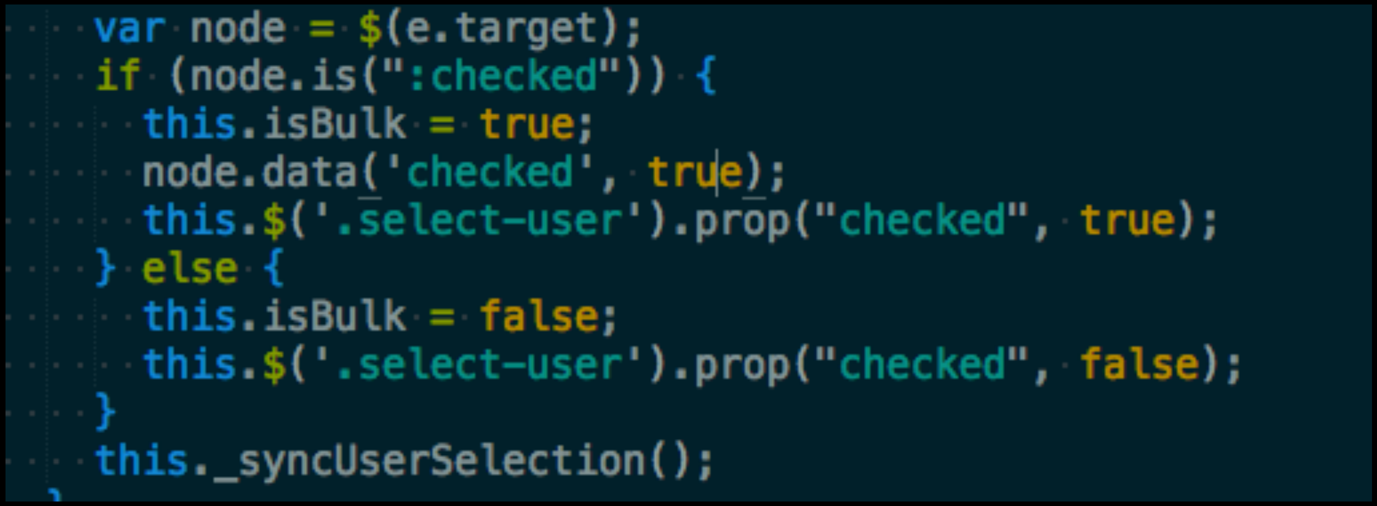
Direct DOM = painful = BUG



```
    listNode = contentBox.one("ul");
    for (var i=0; i < this._allUsers.length; i++) {
        listNode.append(templates.userItem({
            selected: selected,
            user: this._allUsers[i]
        }));
    };
```

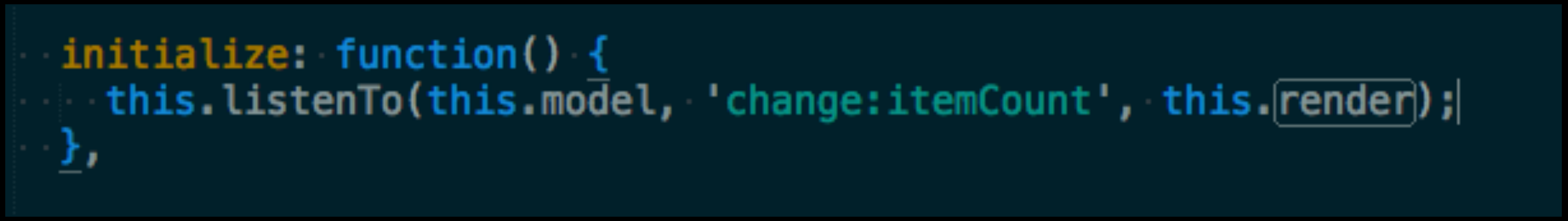


```
    // pagination
    if (this.isBulk) {
      newListNode.find('.select-user').prop("checked", true);
    }
```

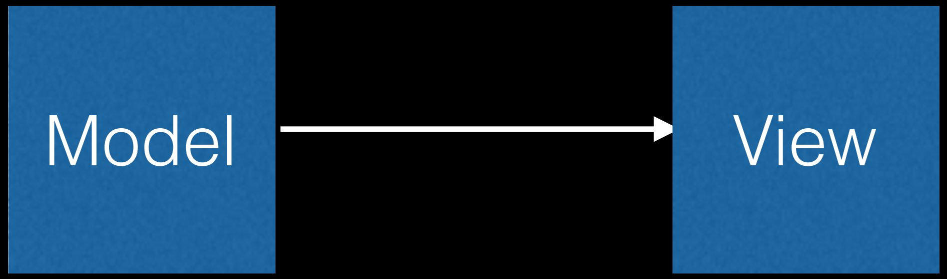


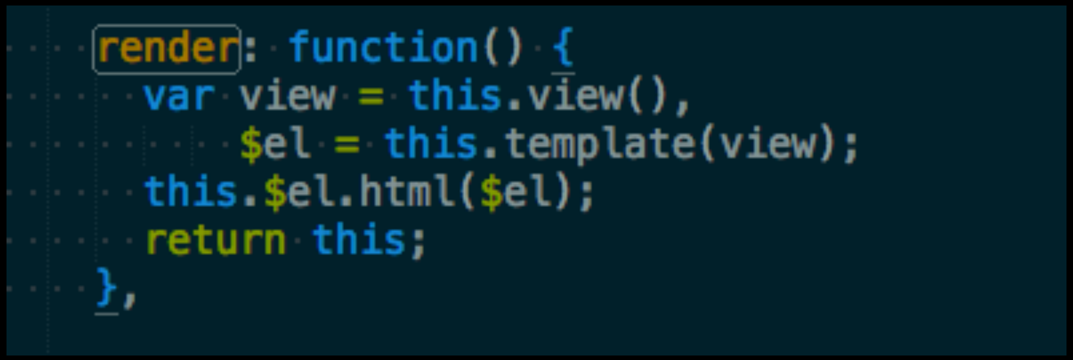
```
    var node = $(e.target);
    if (node.is(":checked")) {
      this.isBulk = true;
      node.data('checked', true);
      this.$('.select-user').prop("checked", true);
    } else {
      this.isBulk = false;
      this.$('.select-user').prop("checked", false);
    }
    this._syncUserSelection();
```

Age of Backbone



```
initialize: function() {
  this.listenTo(this.model, 'change:itemCount', this.render);
},
```

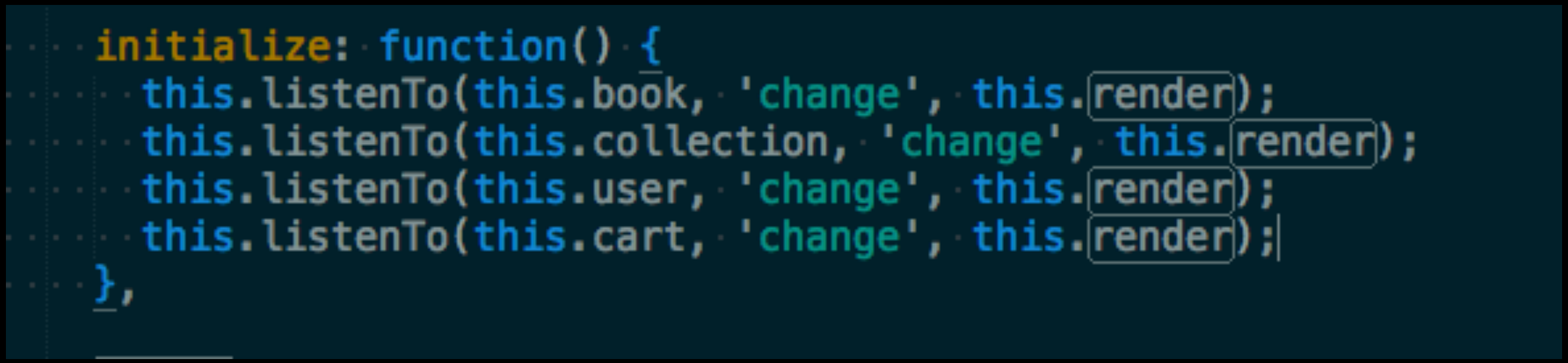




```
render: function() {
  var view = this.view(),
      $el = this.template(view);
  this.$el.html($el);
  return this;
},
```

Works Great in small scale

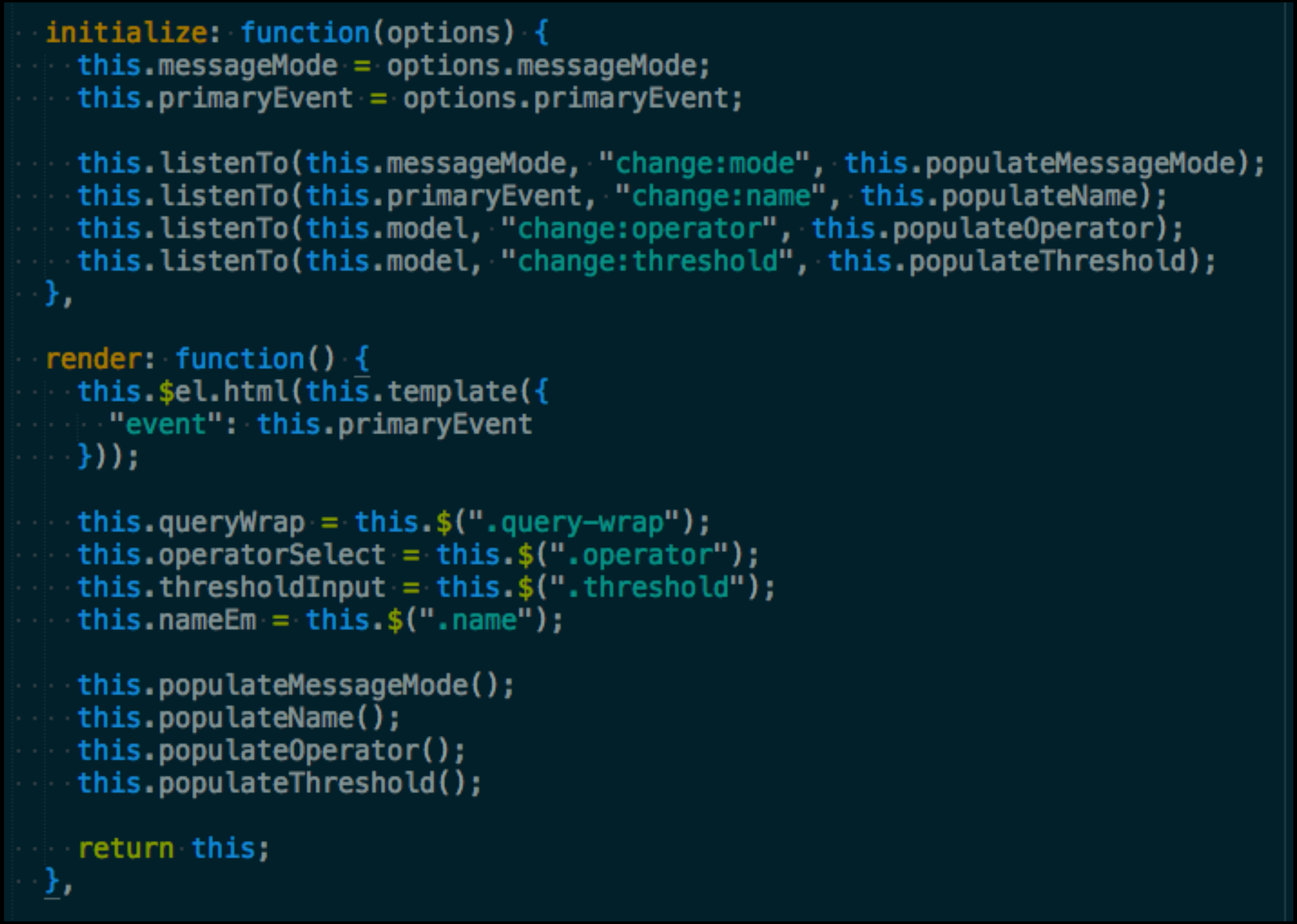
more Model; more View



```
  initialize: function() {
    this.listenTo(this.book, 'change', this.render);
    this.listenTo(this.collection, 'change', this.render);
    this.listenTo(this.user, 'change', this.render);
    this.listenTo(this.cart, 'change', this.render);
  },
```

It is SLOW

Let Optimise it

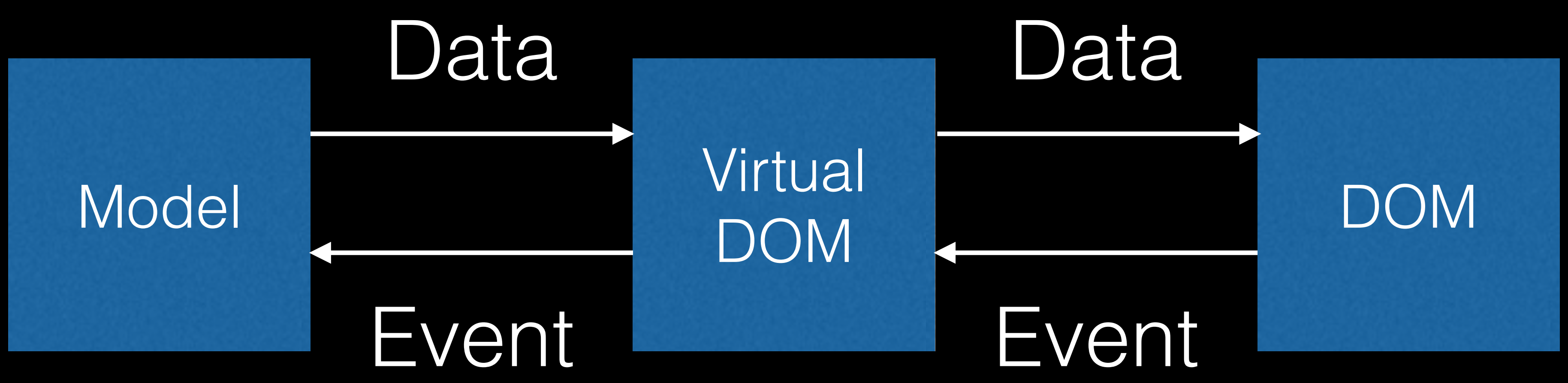


```
initialize: function(options) {
    this.messageMode = options.messageMode;
    this.primaryEvent = options.primaryEvent;

    this.listenTo(this.messageMode, "change:mode", this.populateMessageMode);
    this.listenTo(this.primaryEvent, "change:name", this.populateName);
    this.listenTo(this.model, "change:operator", this.populateOperator);
    this.listenTo(this.model, "change:threshold", this.populateThreshold);
},

render: function() {
    this.$el.html(this.template({
        "event": this.primaryEvent
    }));

    this.queryWrap = this.$(".query-wrap");
    this.operatorSelect = this.$(".operator");
    this.thresholdInput = this.$(".threshold");
    this.nameEm = this.$(".name");

    this.populateMessageMode();
    this.populateName();
    this.populateOperator();
    this.populateThreshold();

    return this;
},
```

evolve images

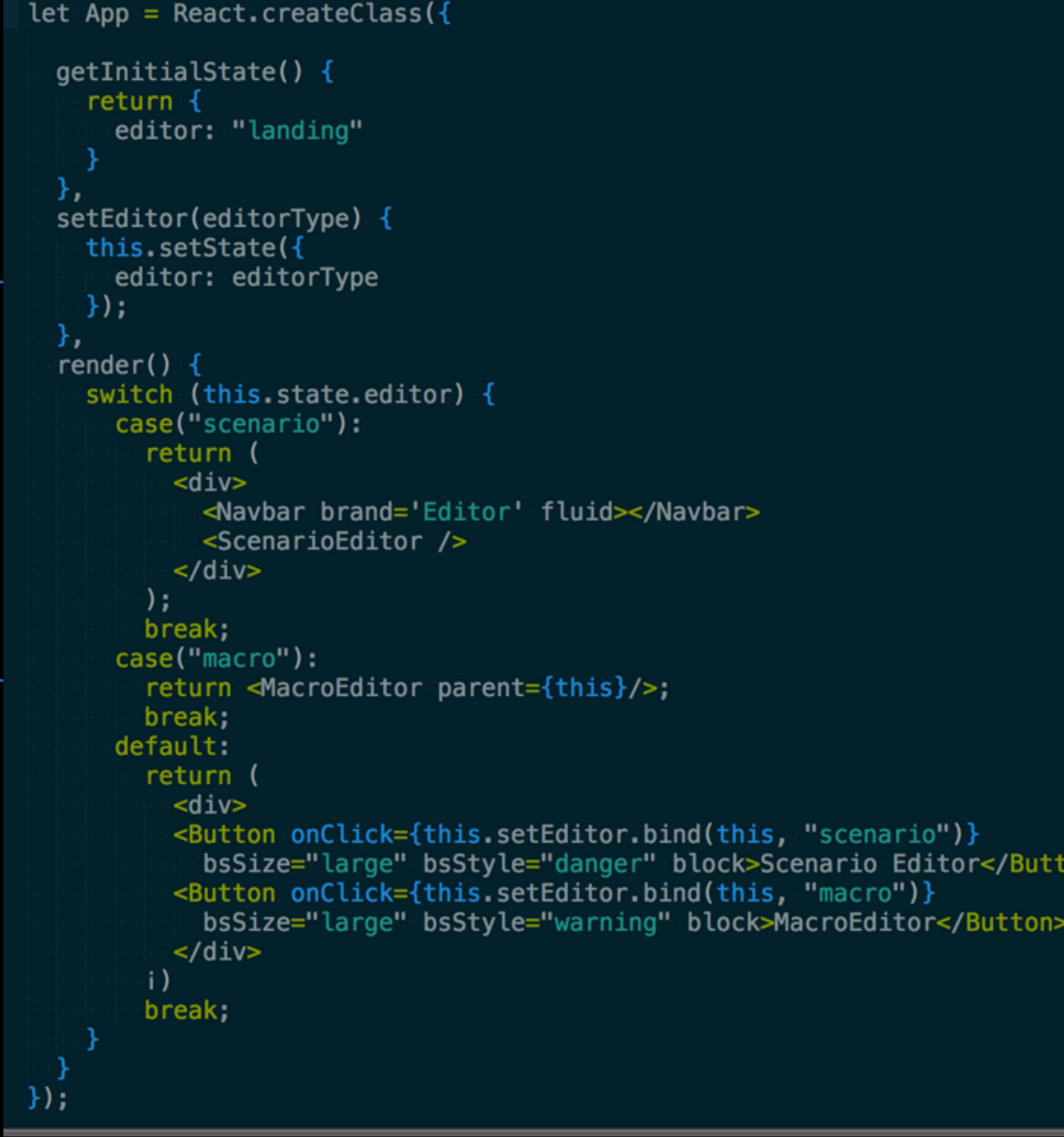
Say Hello to VirtualDOM

Interact with VirtualDOM



setState will trigger render

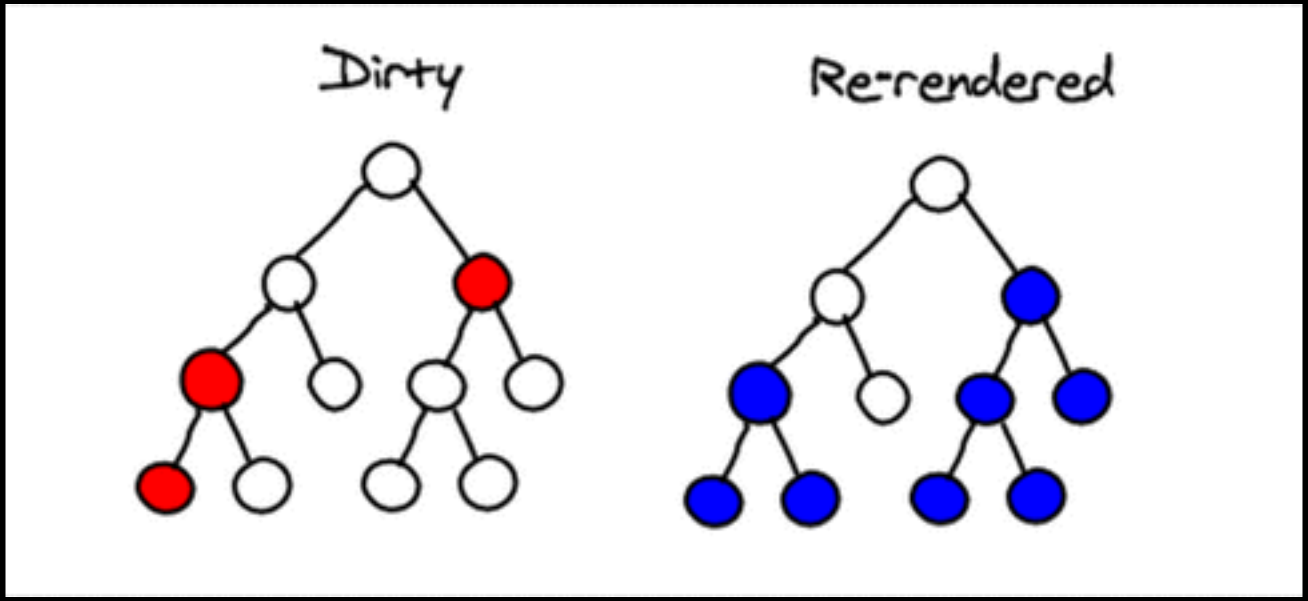
Define your view



```
let App = React.createClass({

  getInitialState() {
    return {
      editor: "landing"
    }
  },
  setEditor(editorType) {
    this.setState({
      editor: editorType
    });
  },
  render() {
    switch (this.state.editor) {
      case("scenario"):
        return (
          <div>
            <Navbar brand='Editor' fluid></Navbar>
            <ScenarioEditor />
          </div>
        );
        break;
      case("macro"):
        return <MacroEditor parent={this}/>;
        break;
      default:
        return (
          <div>
          <Button onClick={this.setEditor.bind(this, "scenario")}
            bsSize="large" bsStyle="danger" block>Scenario Editor</Butt
          <Button onClick={this.setEditor.bind(this, "macro")}
            bsSize="large" bsStyle="warning" block>MacroEditor</Button>
          </div>
        ;)
        break;
    }
  }
});
```

React calculate the diff

- Set state will mark the red dot
- React will find out the blue dots
- Re-render with only the modified dom



Source: https://facebook.github.io/react/img/blog/react-diff-tree.png

Fast without spaghetti

Let focus on Logic

State inconsistency

Very common in single page webapp

State inconsistency

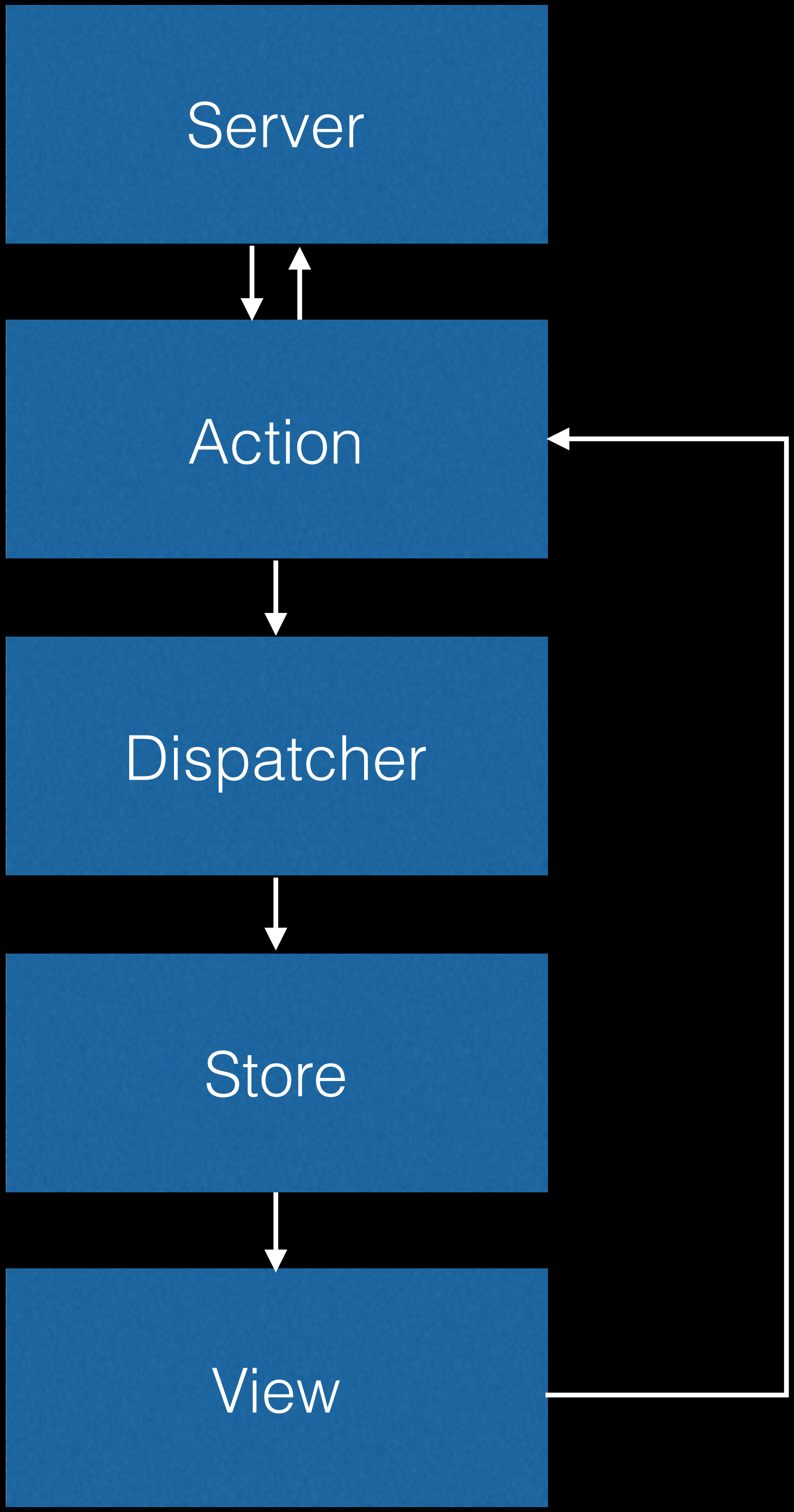
- Server sync not sync with client state
  - item missing
  - duplicate item
- State between client not sync
- State panic with poor internet

Flux fix that

Flux is not framework

It is a pattern

A pattern enforcing unidirectional data flow


Server
Action
Dispatcher
Store
View

Unidirectional data flow

- Faster debug
- Faster on boarding
- Faster iteration
- Less cascading effect

Thank you