

# Flux Architecture with Alt



Hendrik Swanepoel

tagtree.tv | @hendrikswan

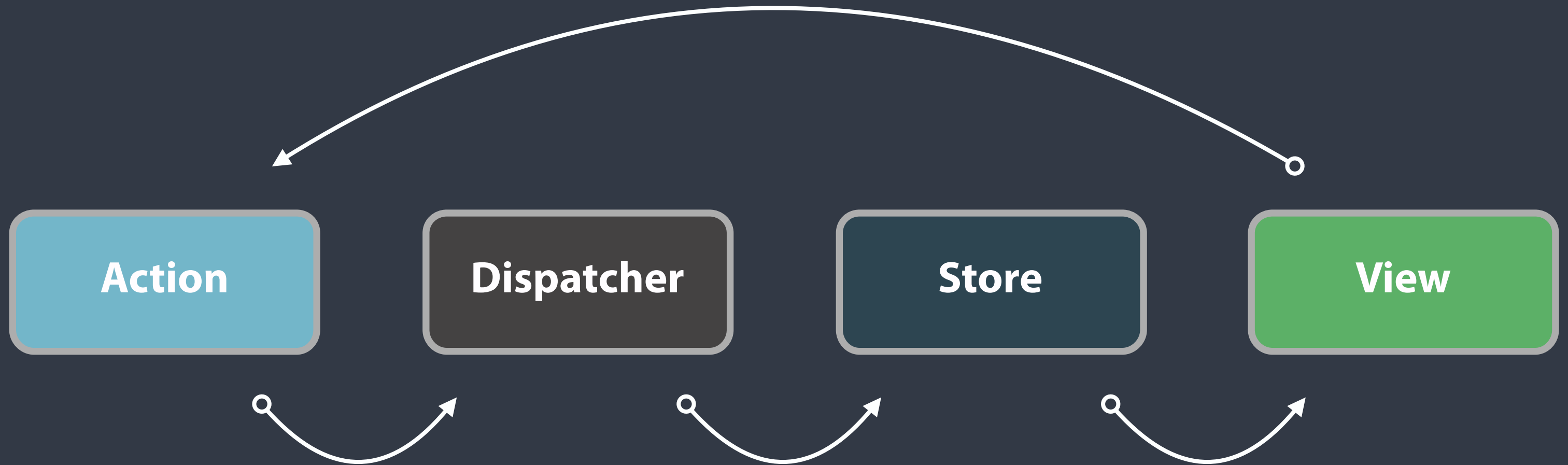
---

# Overview

- What is Flux?
- Why use Alt?
- Install and configure
- Add login functionality
- Refactor the channel list
- Refactor the message list



Facebook Architecture to  
Structure React Applications  
<http://facebook.github.io/flux/>





# *The Benefits of Using Alt*

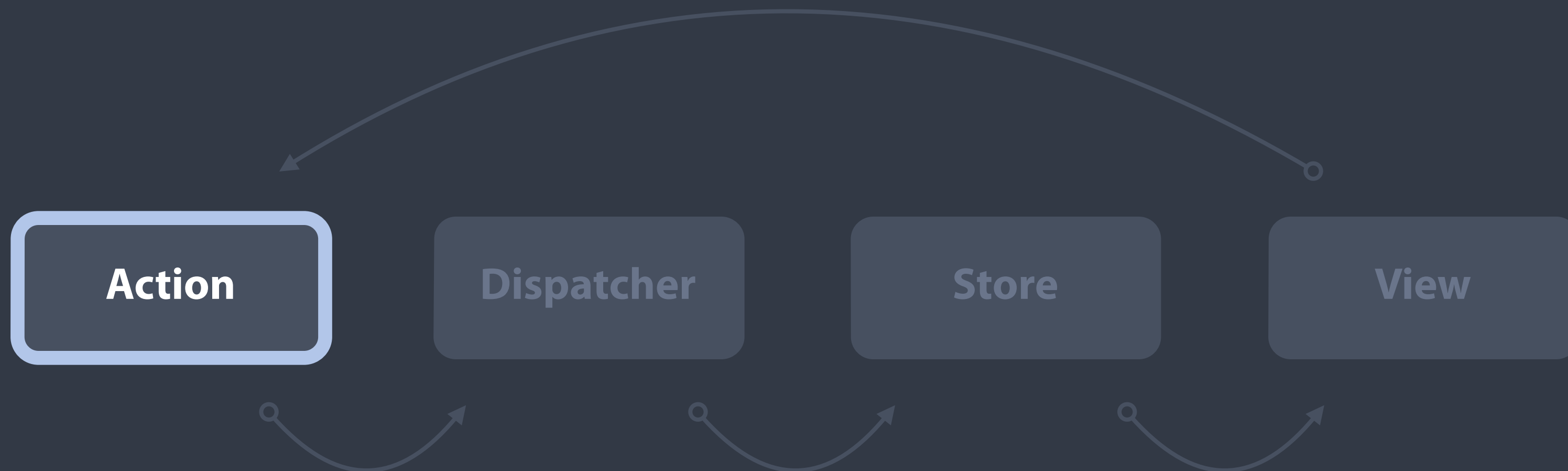


A Developer Friendly Implementation  
of the Flux Architecture

<https://github.com/goatslacker/alt>



VS



```
let constants = {  
  SAY_HELLO: 'SAY_HELLO',  
  SAY_BYE: 'SAY_BYE'  
}  
  
export default constants;
```



```
var AppDispatcher = require('../dispatcher');
var AppConstants = require('../constants');

class Actions {
  sayHello(message){
    AppDispatcher.dispatch({
      actionType: AppConstants.SAY_HELLO,
      message: message
    });
  }

  sayBye(message){
    AppDispatcher.dispatch({
      actionType: AppConstants.SAY_BYE,
      message: message
    });
  }
}

export default new Actions();
```

```
import alt from '../alt';

class Actions {
  constructor(){
    this.generateActions(
      'sayHello',
      'sayBye'
    );
  }
}

export default alt.createActions(Actions);
```



VS





VS



```
import alt from '../alt';  
import Actions from '../actions';  
import {decorate, bind} from 'alt/utils/decorators';
```

```
@decorate(alt)
```

```
class ChatStore {
```

```
  constructor(){
```

```
    super();
```

```
    this.state = {
```

```
      helloMessage: null,
```

```
      byeMessage: null
```

```
    };
```

```
  }
```

```
@bind(Actions.sayHello)
```

```
sayHello(){
```

```
  this.setState({
```

```
    helloMessage: true
```

```
  });
```

```
}
```

```
@bind(Actions.sayBye)
```

```
sayBye(){
```

```
  this.setState({
```

```
    helloMessage: true
```

```
  });
```

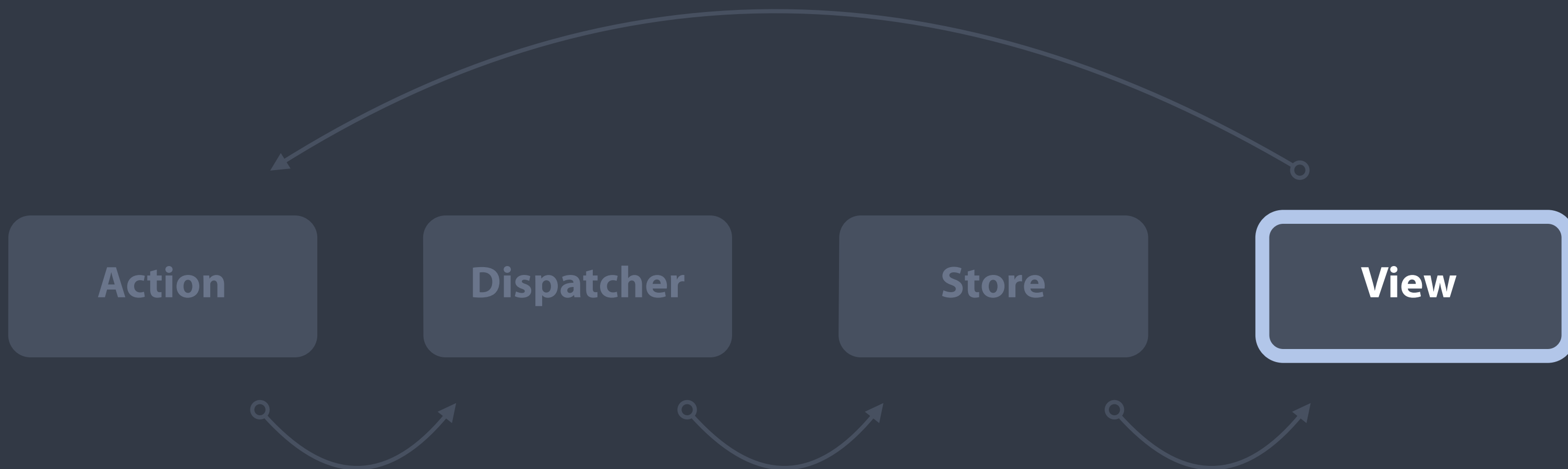
```
}
```

```
}
```

```
export default alt.createStore(ChatStore);
```



VS



```
import React from 'react';
import ChatStore from '../ChatStore';

class MessageComponent extends React.Component {
  constructor(props){
    super(props);
    this.state = {
      message: null
    };
    this.changeListener = this.onChange.bind(this);
    ChatStore.addChangeListener(this.changeListener);
  }

  componentWillUnmount() {
    ChatStore.removeChangeListener(this.changeListener);
  }

  onChange(){
    this.setState(ChatStore.getMessages());
  }

  render(){
    ...
  }
}

export default MessageComponent;
```

```
import React from 'react';
import connectToStores from 'alt/utils/connectToStores';
import ChatStore from './ChatStore';
```

@connectToStores

```
class MessageComponent extends React.Component {
  constructor(props){
    super(props);
    this.state = {
      message: null
    };
  }

  static getStores(){
    return [ChatStore];
  }

  static getPropsFromStores(){
    return ChatStore.getState();
  }

  render(){
    ...
  }
}
```

```
export default MessageComponent;
```

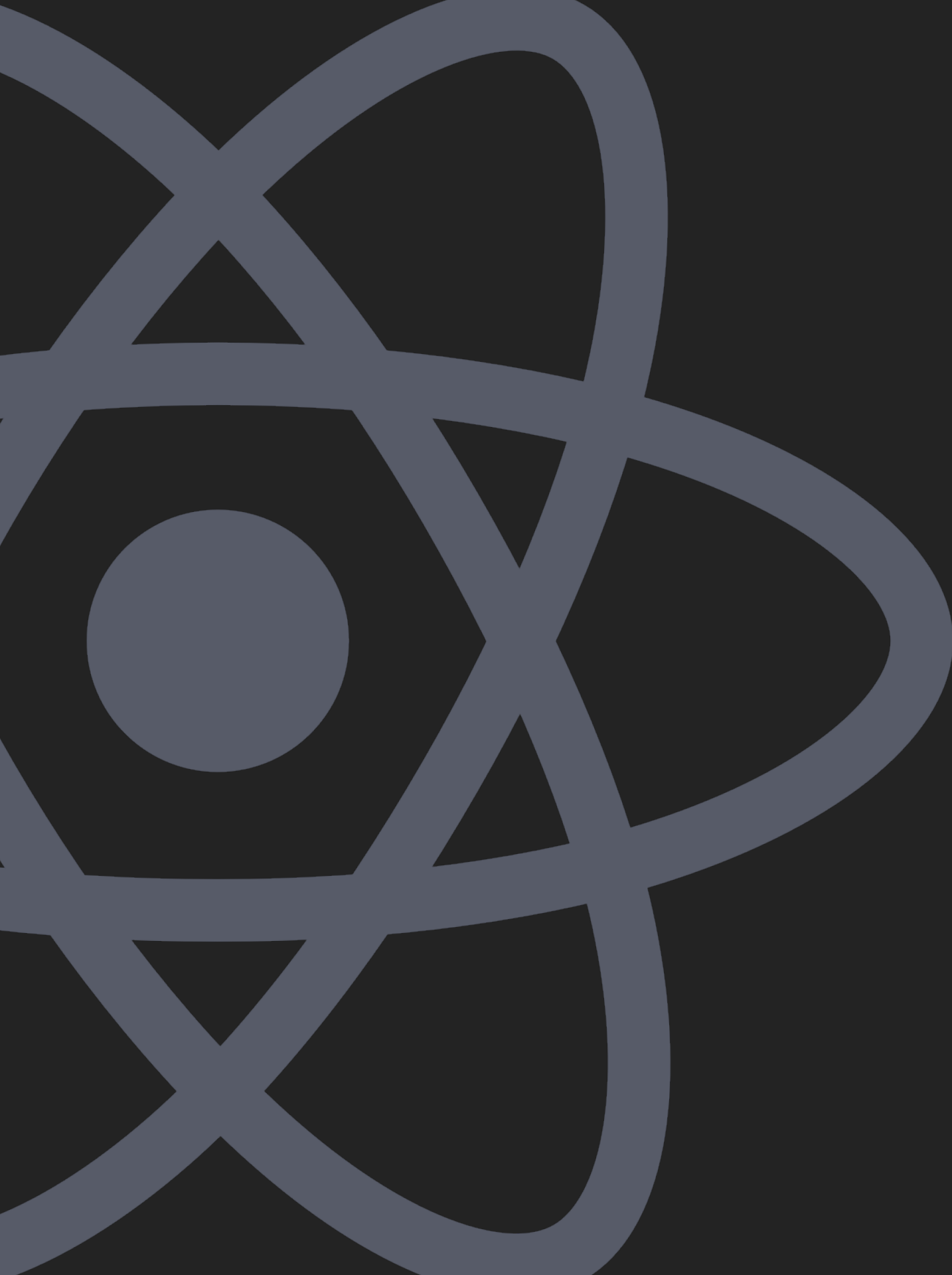




# *Getting Started with the Alt Library*



*Refactor the Channel  
List Component to Flux*



# *Refactor the Message List Component to Flux*



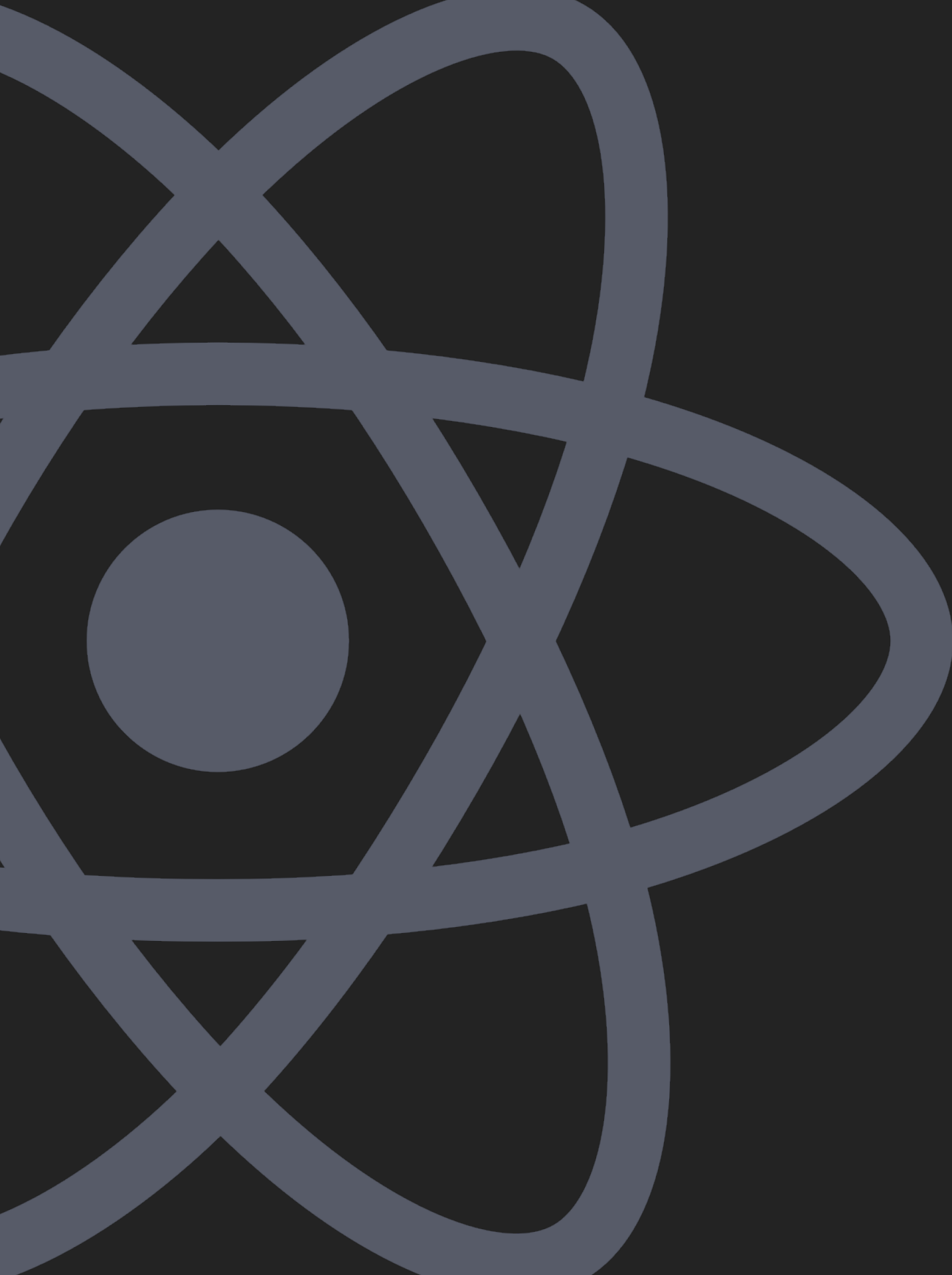
# *Responding to a Channel Selection*



*Showing Message  
Loading Progress*



# *The Flux Way to Add a New Message*



*Adding a New Message  
to the Flux Store*



*Adding a New Message  
to the Flux Store*





*Recap*

# Recap

- Actions
  - `generateActions()`
- Stores
  - Sources - `@datasource`
  - Binding to actions - `@bind`
- Component binding to store updates
  - `@connectToStores`
  - `getStores()`
  - `getPropsFromStores()`



**REACT/ROUTER**