Applications with Splunk UI and React Visualizations

Forward-Looking Statements

During the course of this presentation, we may make forward-looking statements regarding future events or plans of the company. We caution you that such statements reflect our current expectations and estimates based on factors currently known to us and that actual events or results may differ materially. The forward-looking statements made in the this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, it may not contain current or accurate information. We do not assume any obligation to update any forward-looking statements made herein.

In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only, and shall not be incorporated into any contract or other commitment. Splunk undertakes no obligation either to develop the features or functionalities described or to include any such feature or functionality in a future release.

Patrick Wied
Sr. Software Engineer | Splunk

Ziyan Wang
Software Engineer | Splunk

splunk> .conf19

Overview

- What we're going to build / The End Result: 1min
- How this would have been done in the old world: 1min
- Quick reminder about React & styled-components: 1min
- The three primary building blocks that we use for our app: 5-10min
- Splunk UI intro
- Dashboard Framework
- React Visualizations
- UI Deconstruction: 1min
- Roughly associate the responsibilities of packages to deconstructed UI: 1min
- Setup: 1min
- classic folder structure
- The Main View: 5-10min

Action Plan

1. Demo
2. The 3 building blocks we will use
   - Splunk UI
   - Dashboard Framework
   - React Visualizations
3. How we built the demo
4. Bonus

Demo

splunk> .conf19

Overview

Enterprise

- SimpleXML Dashboard
- Custom Javascript Extension
- Hide the Chrome
- SplunkJS
- Packaged as Splunk App

The New Way

- React Single Page App
- Packaged as web app
- Component libraries that make it easy to use Splunk services

The 3 building blocks

Splunk UI	@splunk/react-ui …
Dashboard Framework	@splunk/dashboard-core …
React Visualizations	@splunk/react-visualizations …

Splunk UI

What's Splunk UI

- A front-end toolkit

Why Splunk UI

- Documented
- Isolated
- Accessible
- Tested
- Versioned

Where to use

Dashboard Framework

- A set of React components that make it easy to show dashboards
- Dashboards are powered by dashboard definition

```
{
  dataSources: {...},
  visualizations:{...},
  layout: {...}
}
```

DDISDO

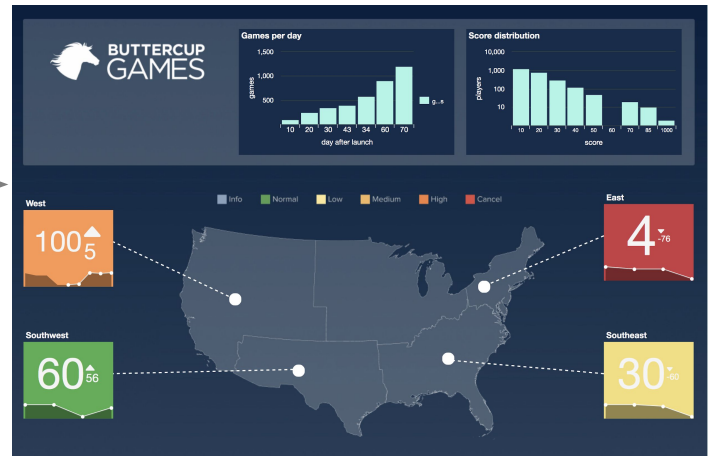
Dashboard Definition In Shiny Dashboard Out

Dashboard Definition

```
{
  dataSources: {...},
  visualizations:{...},
  layout: {...}
}
```

dashboard framework

Shiny Dashboard

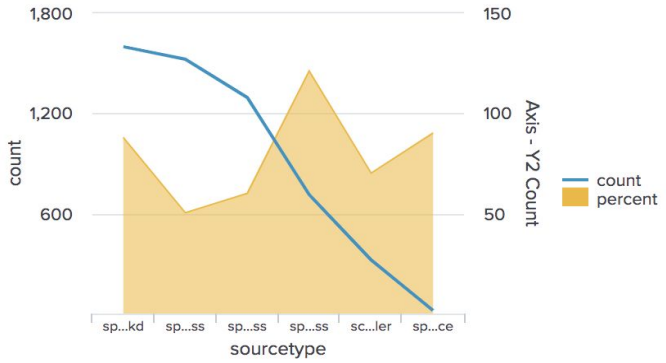


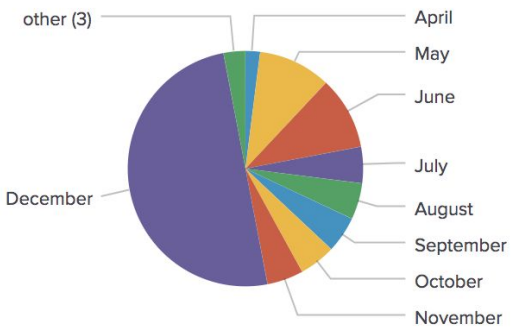
<DashboardCore>

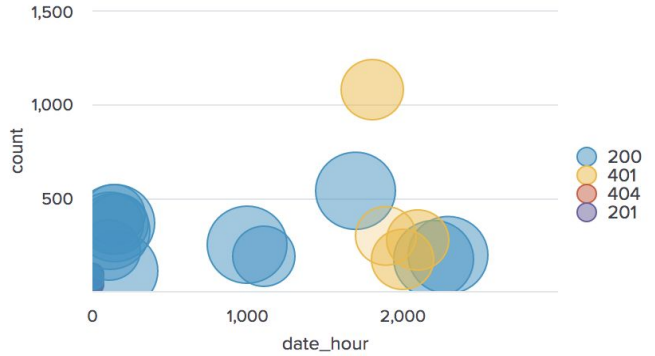
- A React component to render dashboards
- Takes a dashboard definition as prop
- Exposes Dashboard Lifecycle Events via dashboardCorePlugin

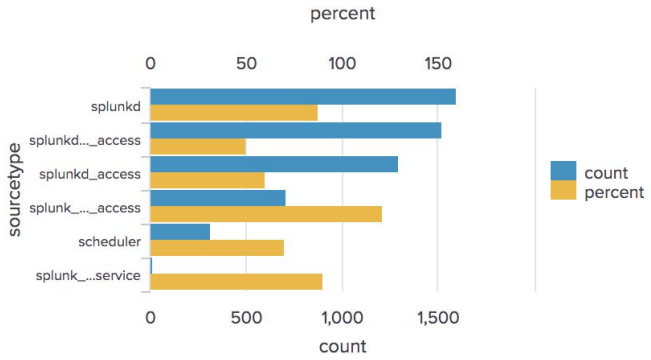
```
<DashboardCore
    width="100%"
    height="100%"
    preset={DefaultPreset}
    definition={dashboardDefinition}
    dashboardCorePlugin={{
        onInitialized: console.log('I am ready!'),
    }}
/>
```

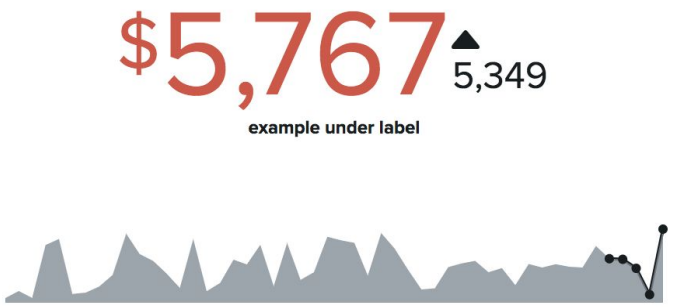
React Visualizations

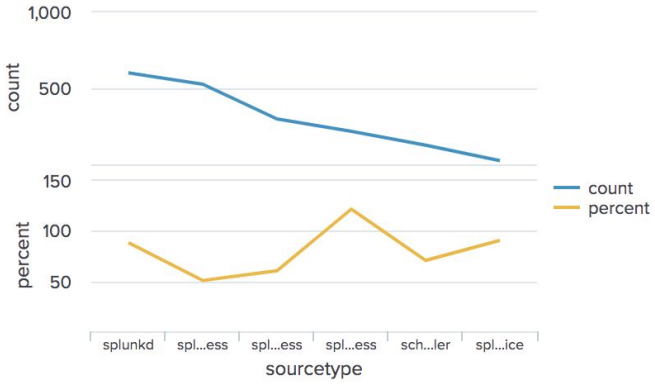








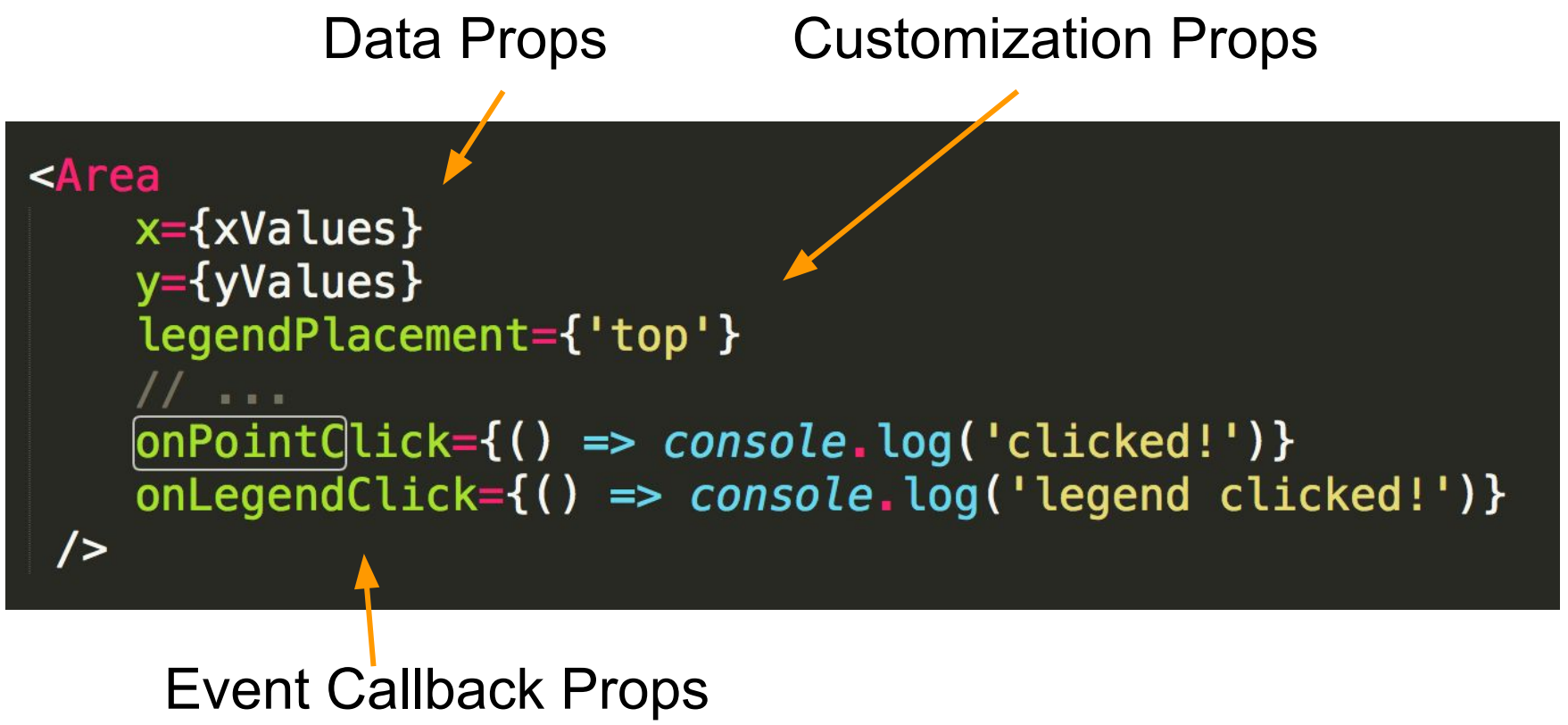




and many more!

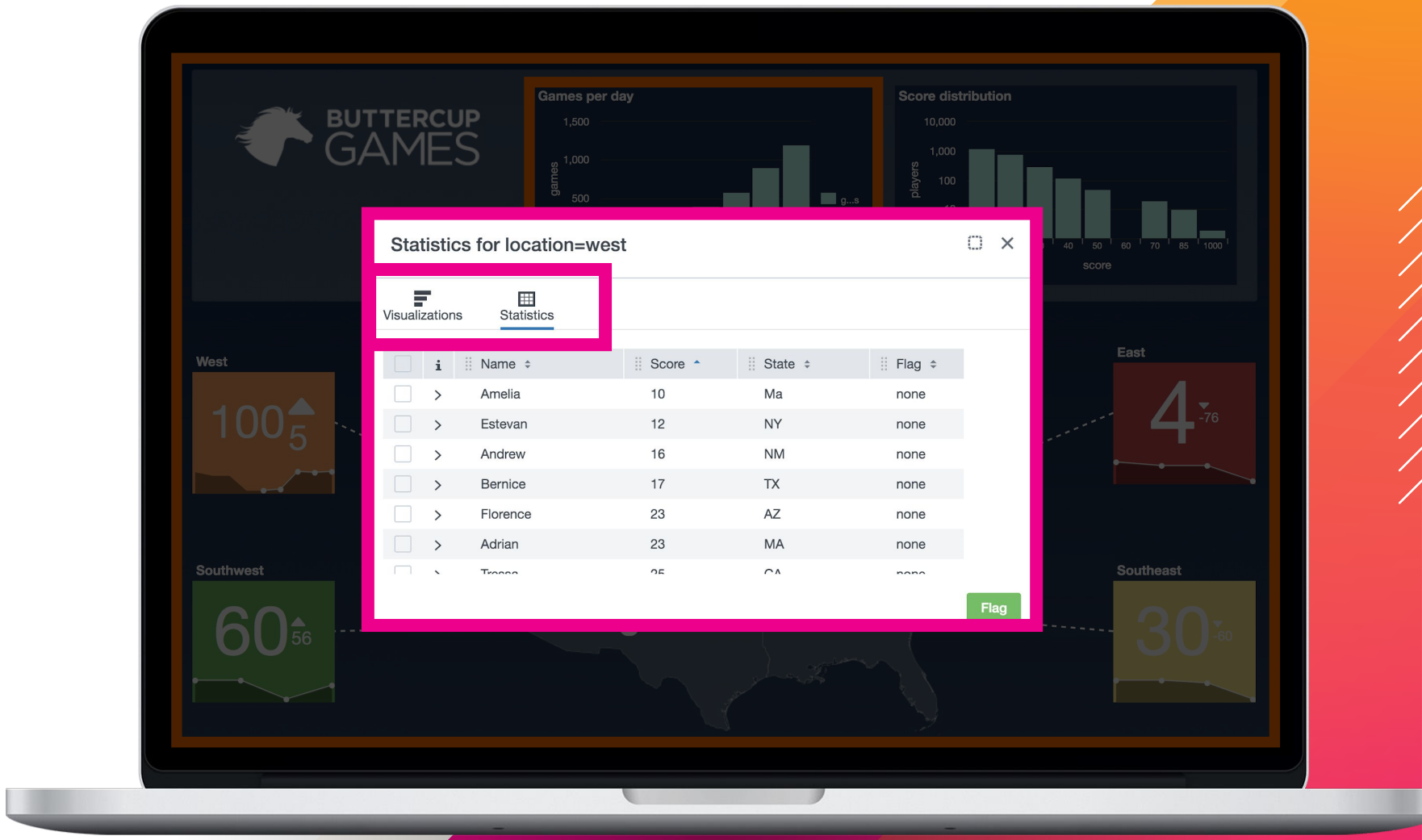
splunk> .conf19

React Visualizations



Let's deconstruct the User Interface

- Dashboard
- Visualizations
- User Interactions
- Modal Window
- Tab Bar
- Table
- Visualization



The 3 building blocks

Splunk UI

- provides UI components (Table, Tab, Modal)
- animates our Modal

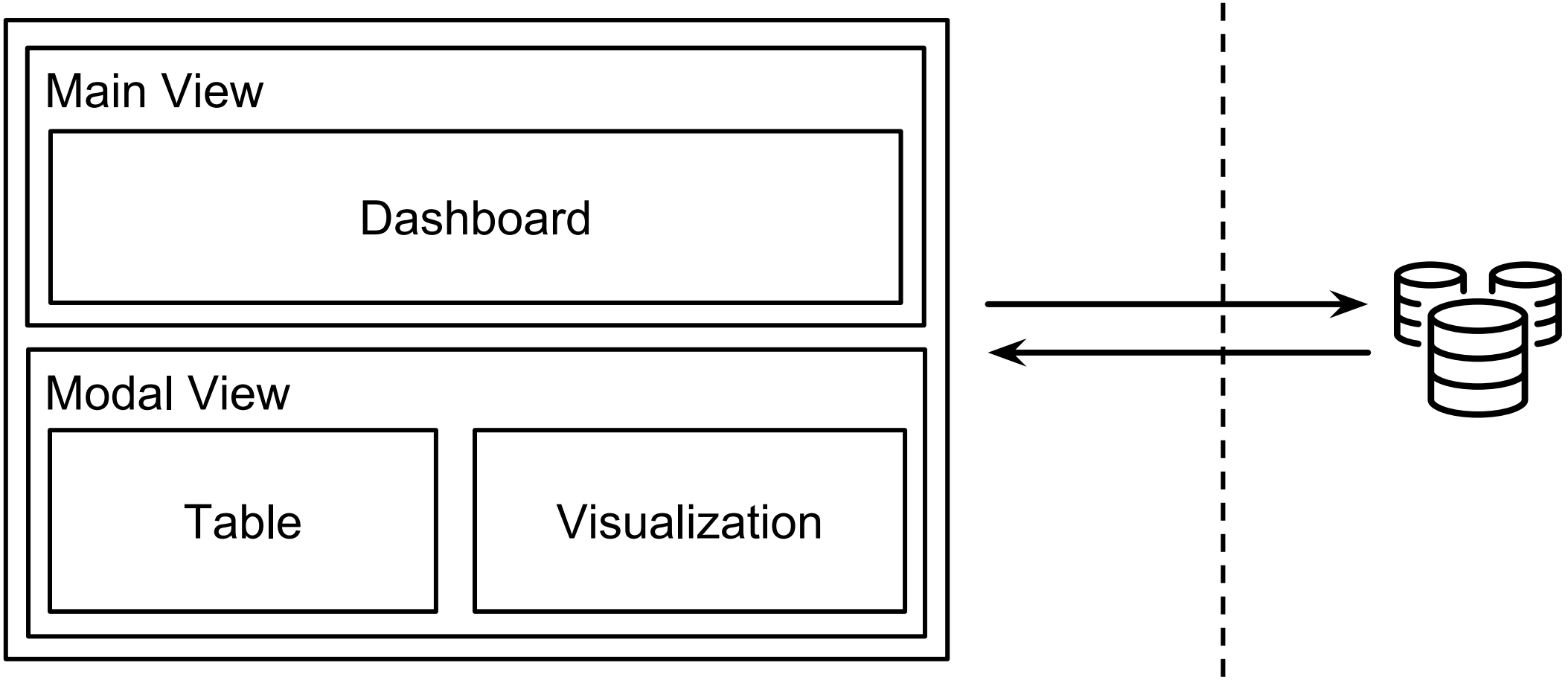
Dashboard Framework

- renders the dashboard
- reacts to click events

React Visualizations

- provides the Visualization component used in the Modal

App Architecture

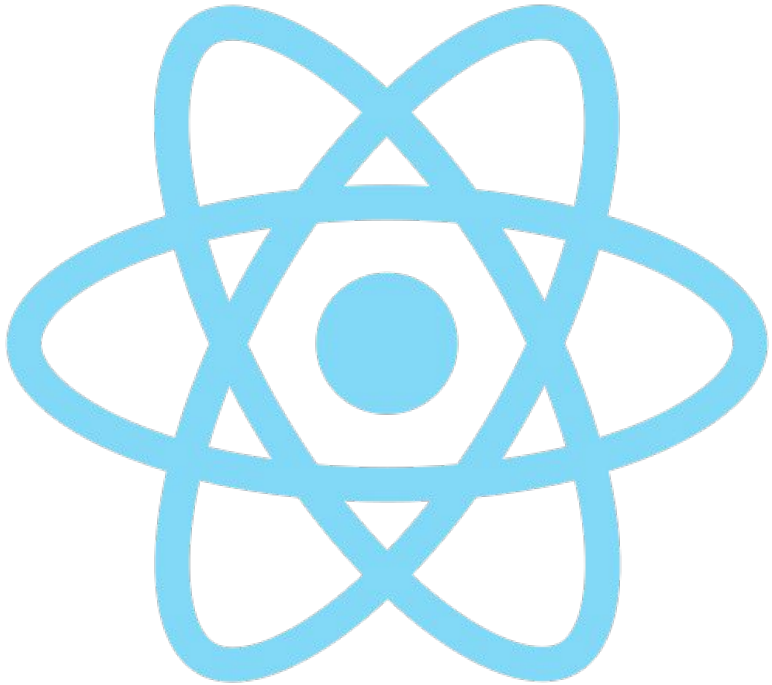


Setup

splunk> .conf19

Set-up

- React Single Page Application
- npx create-react-app buttercup-games-app
- yarn add / npm install to add dependencies



Recap

What did we do?

1. Used create-react-app to create the app
2. Added library dependencies
   - @splunk/dashboard-core
   - @splunk/react-ui
   - @splunk/react-visualizations
3. Added peer dependencies
   - styled-components
   - babel-polyfill
   - @splunk/dashboard-presets

The Main View

splunk> .conf19

Create the Dashboard Definition

- Splunk Investigate has a drag and drop editor to build dashboards
- Use the source, Luke!

Recap

What did we do?

1. Imported the DashboardCore
2. Created a dashboard in Splunk Investigate
3. Copied the dashboard definition from source mode
4. Passed the dashboard definition to DashboardCore

The Modal View

splunk> .conf19

The Modal View

What Component we are using

- Modal
- TabLayout
- Table
- Area Visualization

```
<Modal>
    <TabLayout>
            <TabLayout.Panel>
             <Area Visualization>
        </TabLayout.Panel>
            <TabLayout.Panel>
                    <Table>
        </TabLayout.Panel>
    <TabLayout>
</Modal>
```

Recap

What did we do?

1. Imported the DashboardCore
2. Created a dashboard in Splunk Investigate
3. Copied the dashboard definition from source mode
4. Passed the dashboard definition to DashboardCore

Connect the Components

splunk> .conf19

Recap

What did we do?

1. Added event handlers to dashboard definition
2. Passed a dashboard core plugin with event callback functions
3. Set state when dashboard emits event

Bonus: Dark Mode

splunk> .conf19

Recap

What did we do?

1. Imported theme objects from our packages
2. Passed theme to ThemeProvider
3.

What did we learn?

- How to create a dashboard that can be embedded in a web app
- How to use UI components from Splunk UI
- How to react to dashboard events
- How to use themes

Source Code:

https://github.com/splunk/customized-app-conf19

Dashboard Examples:

https://github.com/splunk/dashboard-conf19-examples

Talk to us at the Visualizations Booth!

Patrick Wied | Senior Software Engineer
Ziyan Wang | Software Engineer

splunk> .conf19

.conf19

splunk>

Thank You !

Go to the .conf19 mobile app to

RATE THIS SESSION