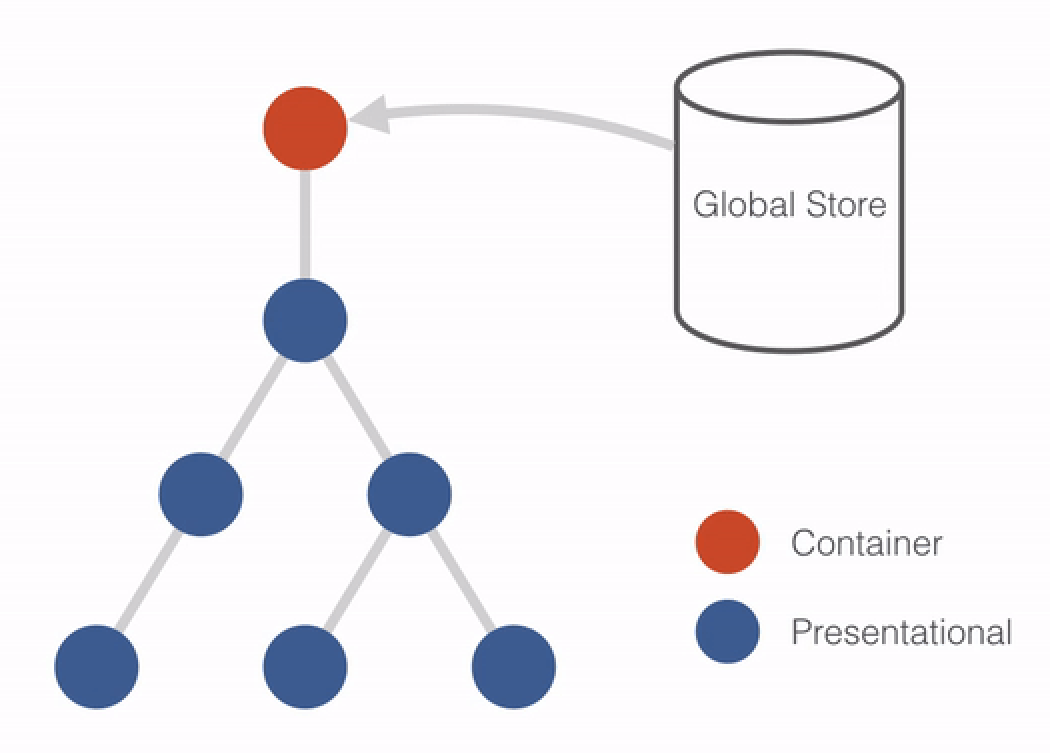
4 - Footer Container

Container / Presentation Arquitecture

If your component has a lot of functionality, often it helps to separate the content in two separate files, one with the logic (Container) and one with the template (Presentation).

In our app: - Container Component: FooterContainer.js - Presentation Component: FooterDisplay.jsx - index is usually used to export the container, since it will make a shorter reference in an import.



Containment / Children

Some components don’t know their children ahead of time. This is especially common for components represent generic “boxes”.

In our app:

```
import Footer from './FooterDisplay';

// Container
class FooterContainer extends Component {

const url = <a href="https://reactjs.org/docs/hello-world.html">React
Documentation</a>;

class FooterContainer extends Component {
  constructor(props) {
    super(props);
    this.state = {
      mail: 'mailto:a.de.la.cruz@accenture.com',
      name: 'Alejandro De la Cruz Tarin'
    }
  }

  render() {
    // You can put any other JSX object inside your component tag to pass it as children
    return (
      <Footer mail={this.state.mail} name={this.state.name}> {url} </Footer>
    );
  }
}
```

FooterDisplay

```
  const Footer = (props) => {
    return (
      <footer>
        <p>Based on:
          {
            /* Children passed as props */
            props.children
          }
        </p>
      </footer>
    )
```

Read More:

- Import Module
- Composition vs Inheritance

<< return