

Redux

State Management for React

Why do we need it?

- Components have local state, but apps have global state.
- Sometimes state needs to be shared.
- We need to save the shared state somewhere.
- Redux = State Container

When you don't need Redux

Building a Blog



http://

Q

`<BlogContainer />`

`getPosts()` (Calls an API)

renders `<Post />` Component with 'title' and 'content' props

`.map(post => <Post title={post.title} content={post.content`

`<Post />`

`<Post />`

`<Post />`

`<Post />`

`<Post />`

`<Post />`

`.map(post => <Post title={post`

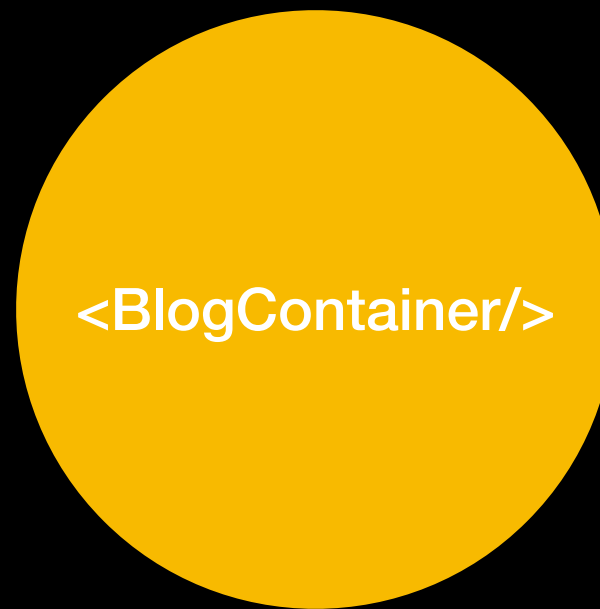
`<Post />`

`getComments(postId)`

renders

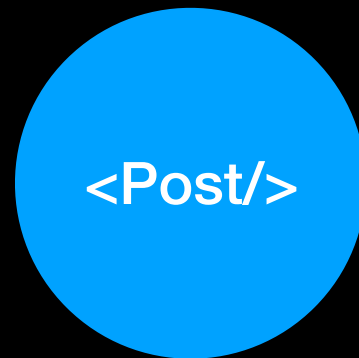
`<Comment />`

`<Post /`

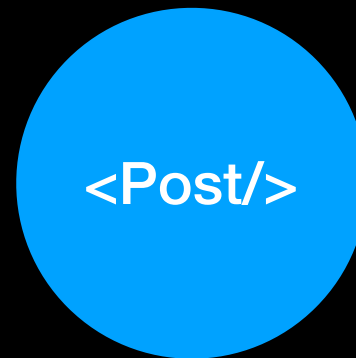


`<BlogContainer/>`

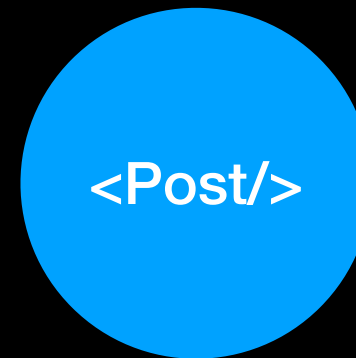
State:
blogPosts



`<Post/>`

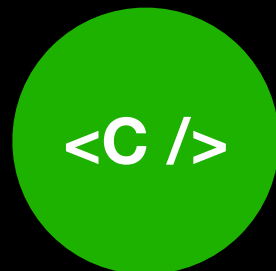


`<Post/>`

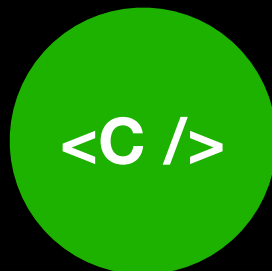


`<Post/>`

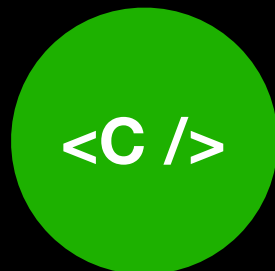
Props:
title
content



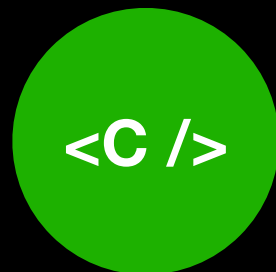
`<C />`



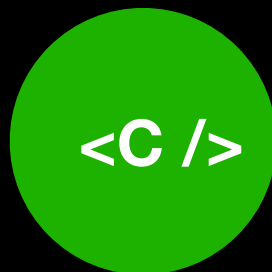
`<C />`



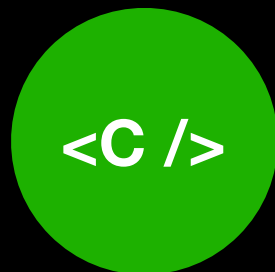
`<C />`



`<C />`



`<C />`



`<C />`

Props:
postId
comment

No Redux.
No problem.



When do you need Redux?

Commenting on a post

Replying on a Post

- The `<Comment />` Component has a `sendComment()` function.
- The `sendComment()` function needs to know who is sending the comment (author), the comment ID and the post ID.

Where is the author?

Where do we get it from?



http://



`<BlogContainer />`

`getPosts()` (Calls an API)

`getUser()` (Calls an API)

renders `<Post />` Component with 'title',
'content' and 'loggedInUser' props

`<Post />`

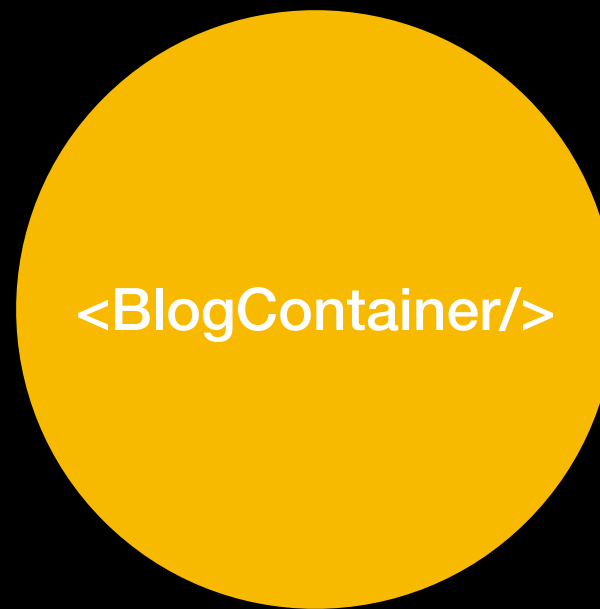
`getComments(postId)`

renders

`<Comment />`

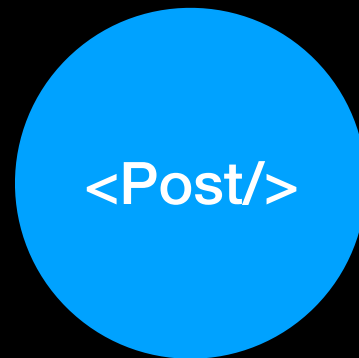
`<Post />`

`<Post />`

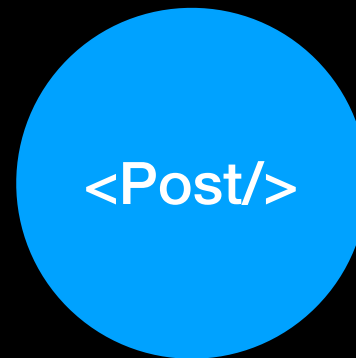


`<BlogContainer/>`

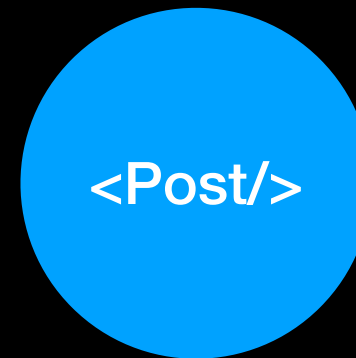
State:
blogPosts
loggedUser



`<Post/>`

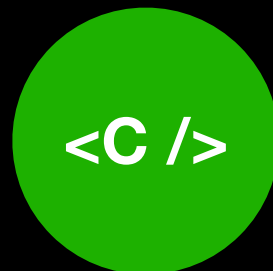


`<Post/>`

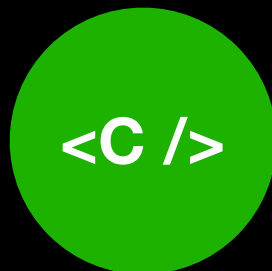


`<Post/>`

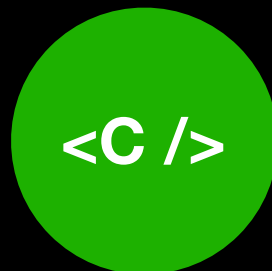
Props:
title
content
+ loggedUser



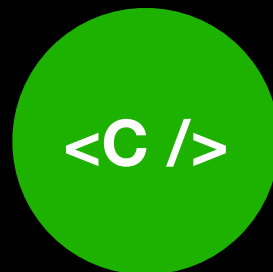
`<C />`



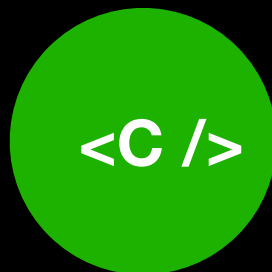
`<C />`



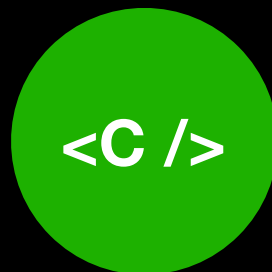
`<C />`



`<C />`



`<C />`



`<C />`

Props:
postId
comment
+ loggedUserId

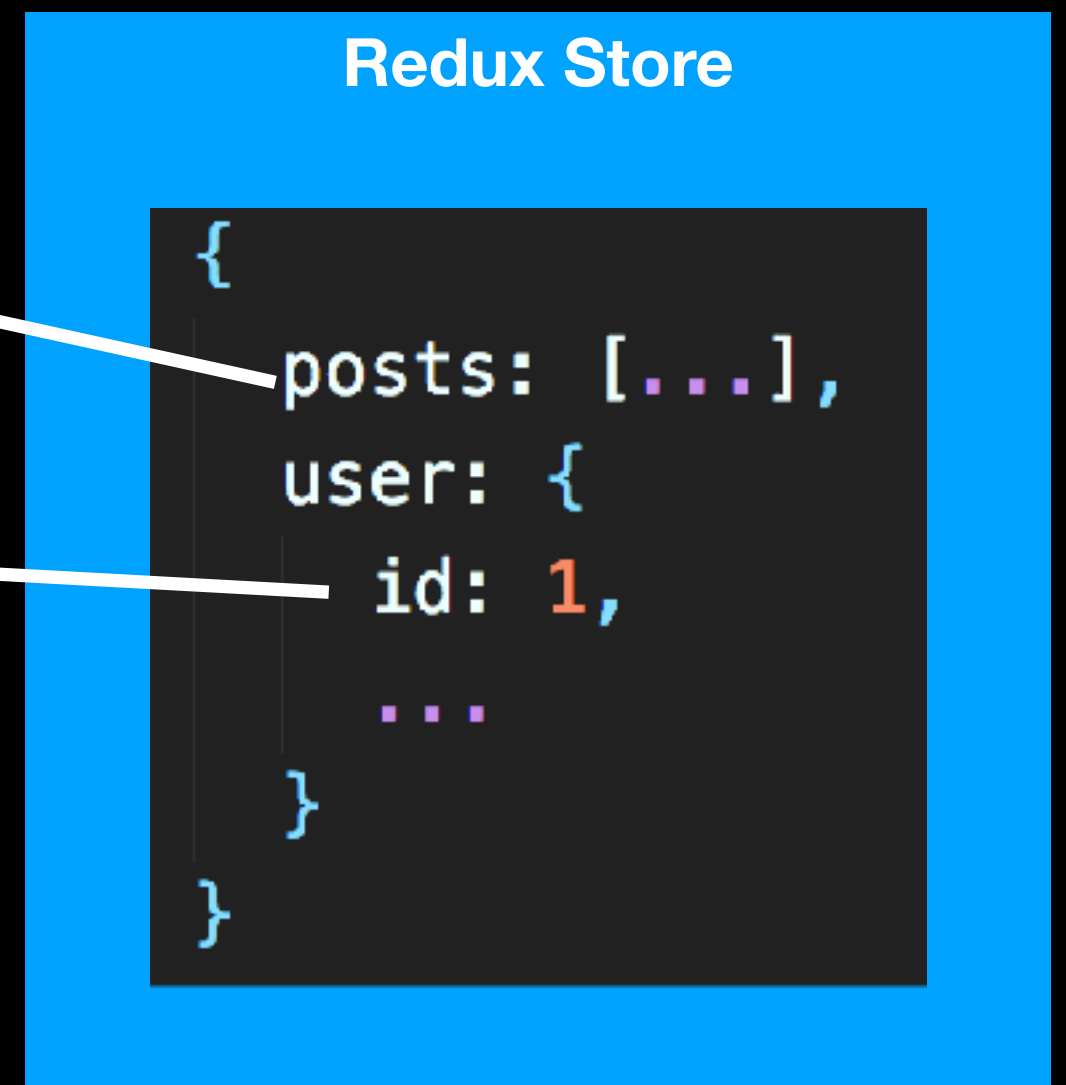
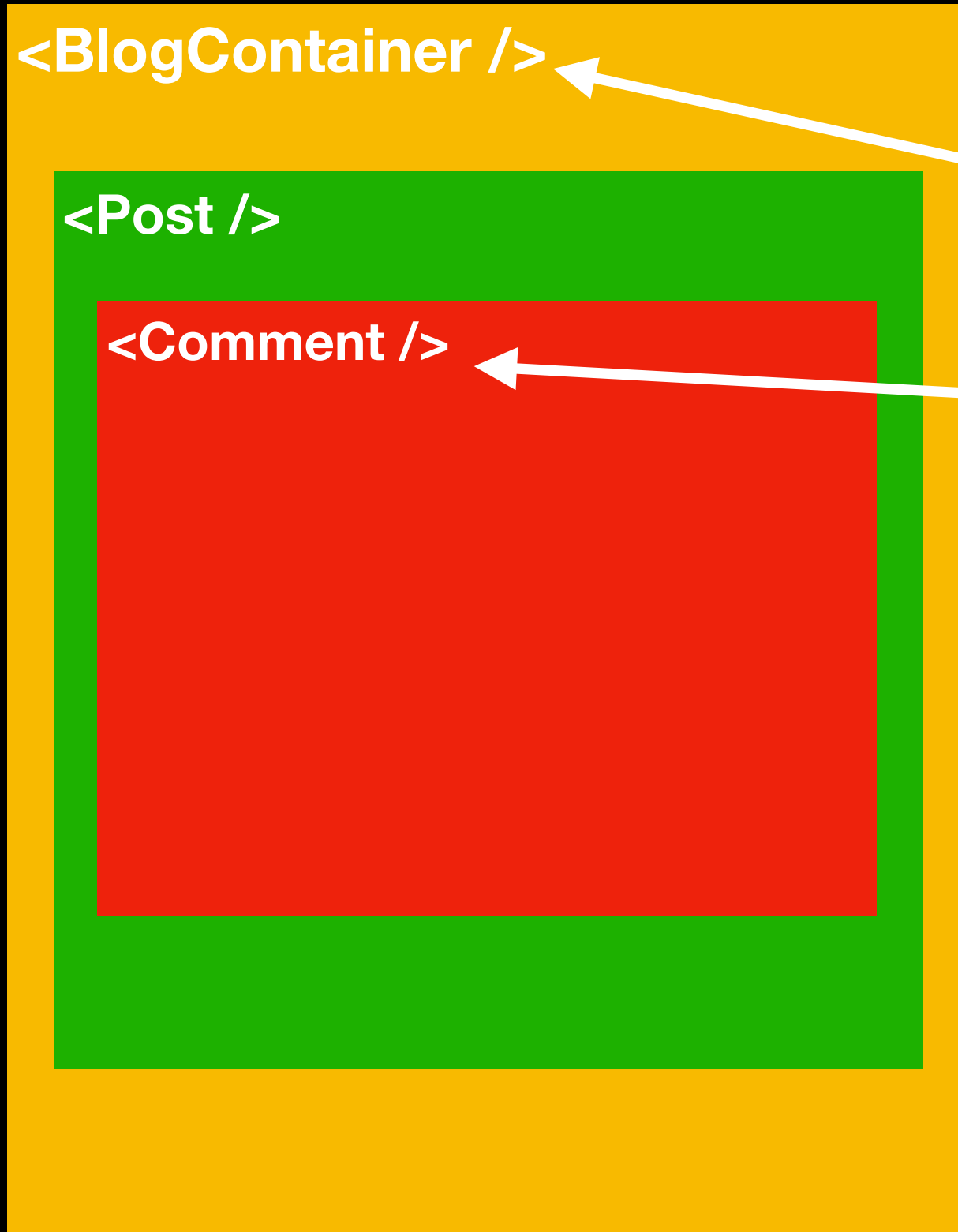
The flying prop

Useless Props

- We are sending the 'loggedUser' prop to <Post /> so later we can send it to <Comment /> so then we can use it to reply to a comment.
- Everything has to be inside of <BlogContainer /> and that's not good news.



Enter: Redux



No flying props



Redux Store

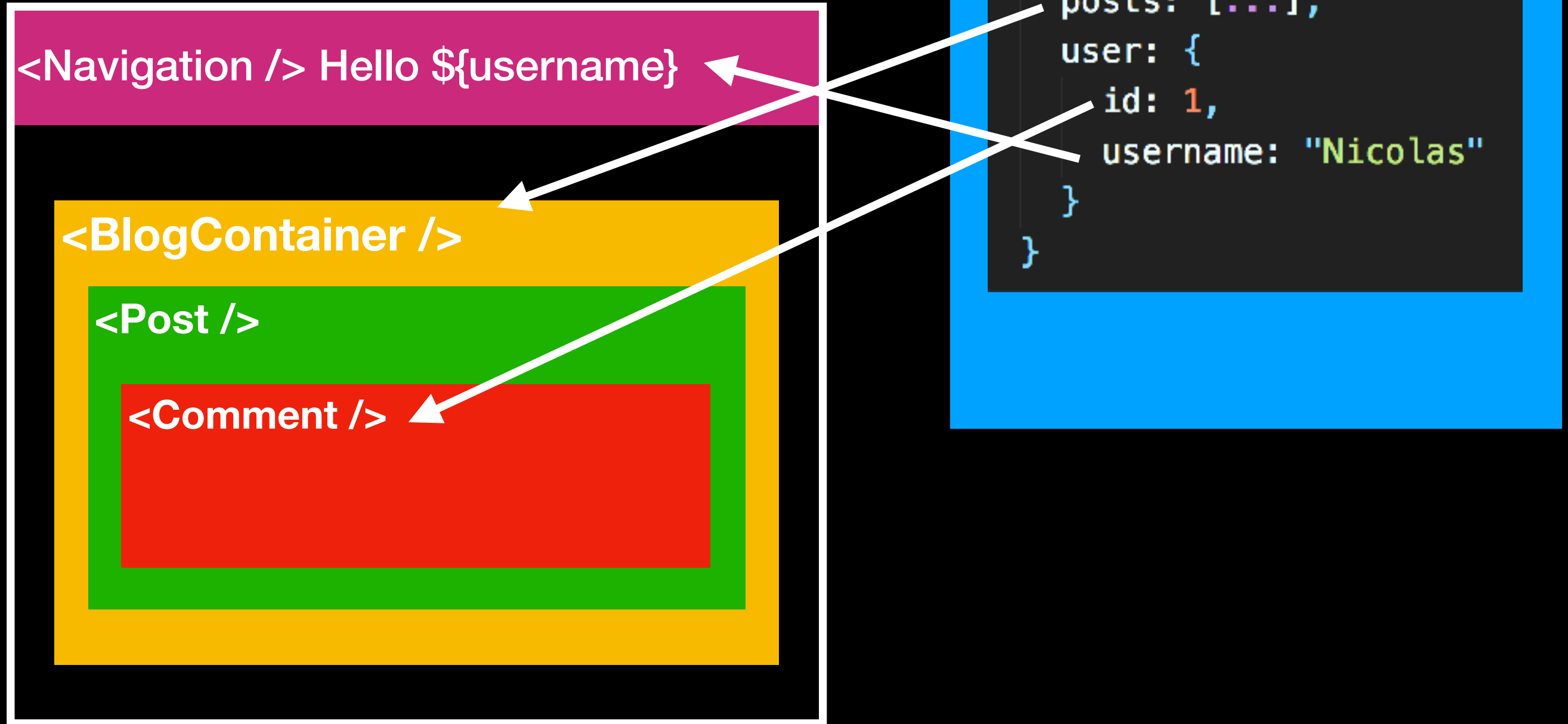
```
{  
  posts: [...],  
  user: {  
    id: 1,  
    username: "Nicolas"  
  }  
}
```

<Navigation /> Hello \${username}

<BlogContainer />

<Post />

<Comment />



Redux is a global state container

- The Redux Store is like a box
- We go and grab what we need for the container.
- For example, from `<Navigation />` we only grab the username.

Stuff to remember

- The whole state of your app is stored in an object (store)
- If you want to change the data inside of this object you need to 'dispatch' (send) an action
- You will send this actions to a reducer and this reducer will change the object for you.