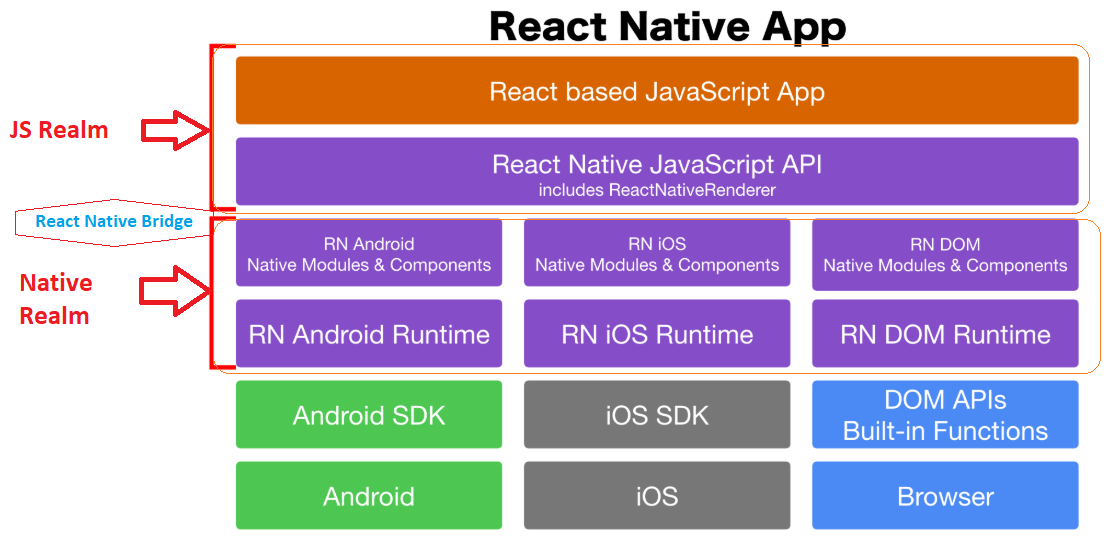
React Native Challenges –

**Performance issues with ReactNative bridge**



**Approach -**

To start with, we can implement all the functionality in JS for maximum productivity. We will not focus on optimization.

Have a separate phase after implementation for React-Native bridge optimizations -

* + After implementation, we can test on a real device to see where your app falters.
  + In areas that are prone for bridge-overuse, such as animations / interactions — prefer declarative libraries.
  + Many interactions can be expressed declaratively and even-though it may not be intuitive at first (see our fourth implementation), it is worth the effort.
  + If traditional React optimizations fail, surgically move the troublesome parts to native.

We can have couple of native developers in our teams for this purpose.

For some complex interactions that cannot be expressed declaratively or libraries will have direct implementation in native.

<https://facebook.github.io/react-native/docs/native-modules-ios>

<https://facebook.github.io/react-native/docs/communication-ios>

<https://facebook.github.io/react-native/docs/native-components-android>

Assumptions –

Socket.io will be supported by the server side API.

This will be used between react native app on the client side and server side components.

