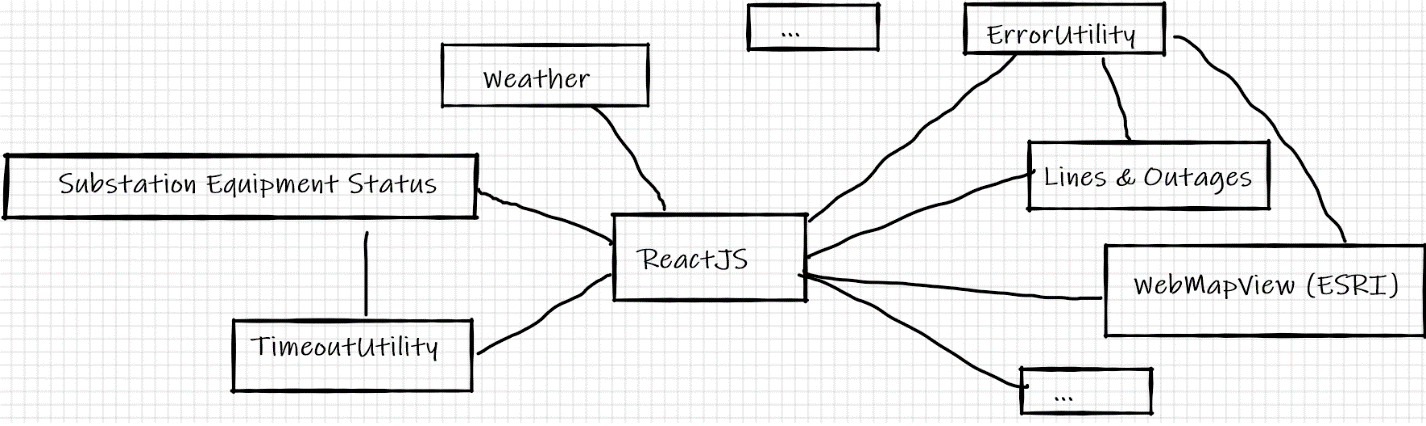
**ReactJS repository with organization modules**



**Pros:**

* Re-usable components
* Easy import/export of modules
* Can use each module internal to the package, or outside the package (ref: ErrorUtility)
* If a change is to be made, such as a bug fix, it only needs to be done
* Follows D.R.Y. practices
* Generics, allows for \*some\* customization using properties

**Cons:**

* Two sources of truth GitHub repository vs. npm package publishing
  + Can be band aided by enforcing common practices and using script files
* Breaking changes will affect all apps importing module
  + This is the adverse of only making change in one spot
* Not necessarily a con, but modules need to be well thought out before publishing to generalize as much as possible

**Problems we need to solve:**

* Should each module a React component or a class component?
* Two sources of truth for the package
* Caching, we can use caching to speed up loading (using localStorage is an option, but need to see if PJM supports caching for security, as well as a max cache size issue)
* Testing! Test environments, packages (Jest?), internal test personnel
  + We should ask PJM to bring in people to help test in a pre-prod environment
* Best way to access Oracle DB’s (GraphQL?)

**Getting set up:**

1. Clone the following repositories we should create a batch file to do this automatically
   1. git clone [git url app] | npm install
   2. git clone [git url package] | npm install