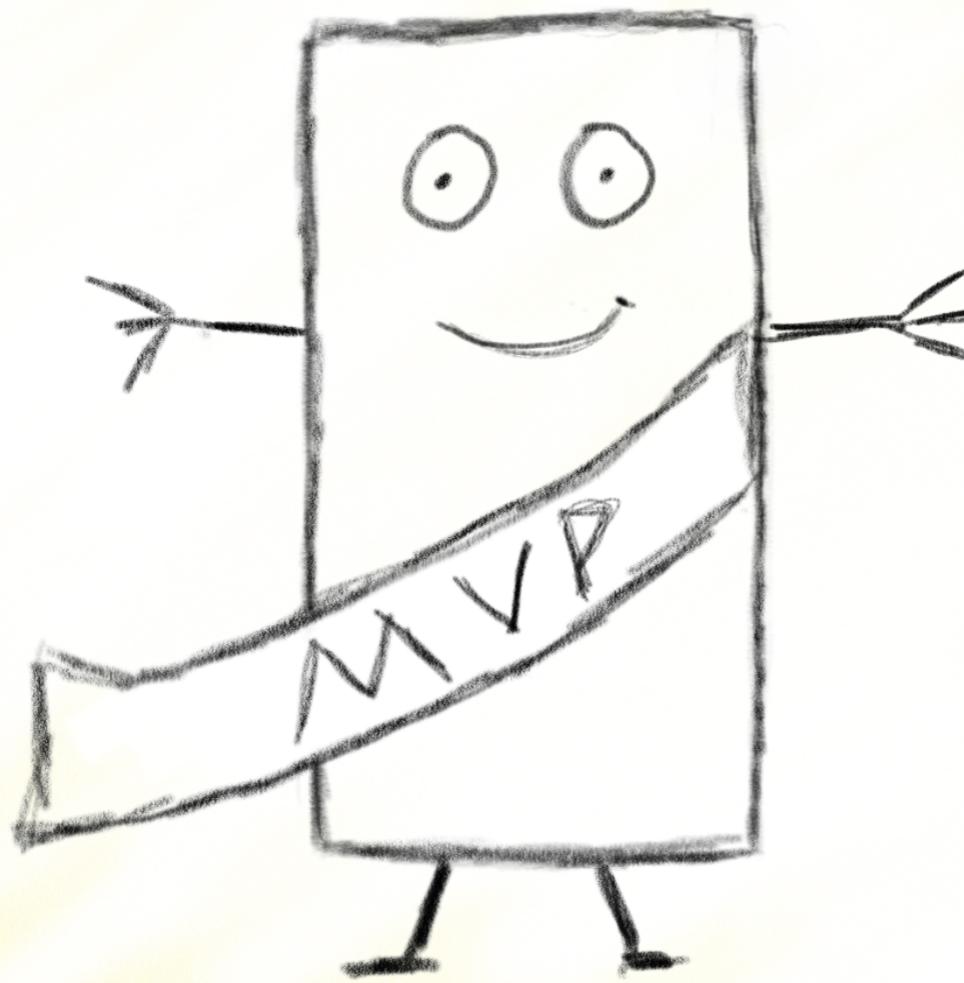
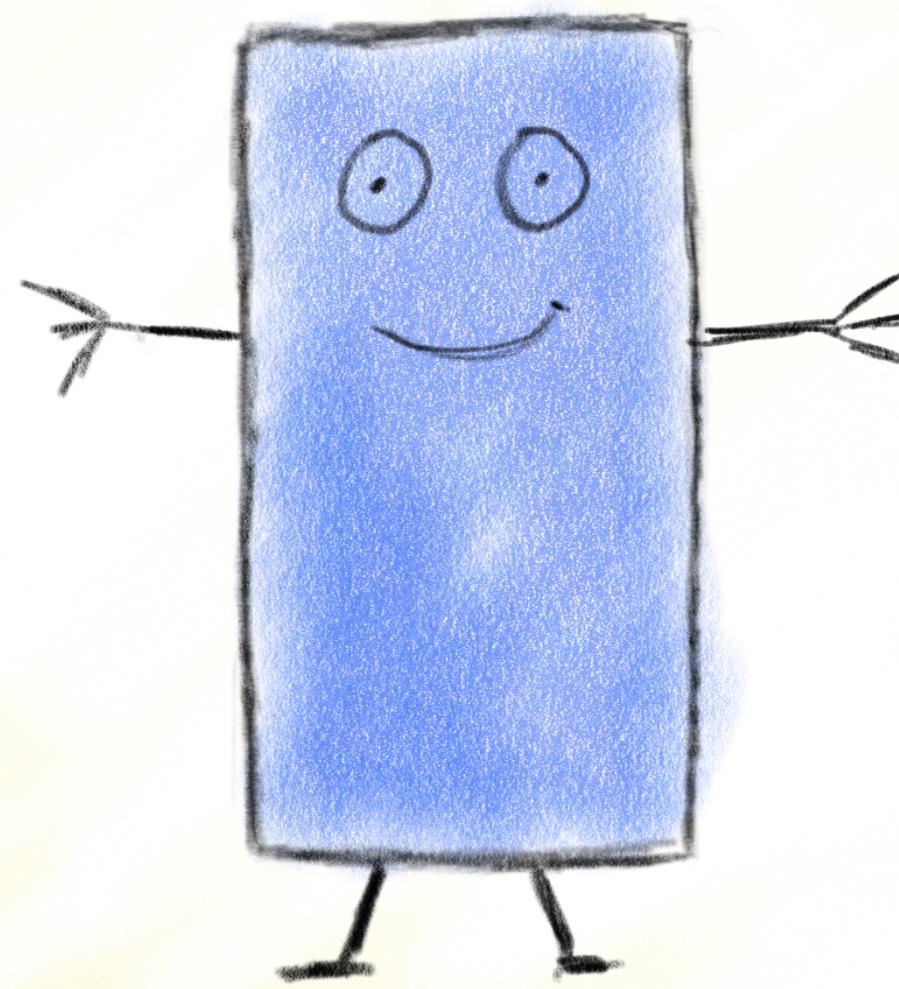


Enforcing Design Systems With ESLint

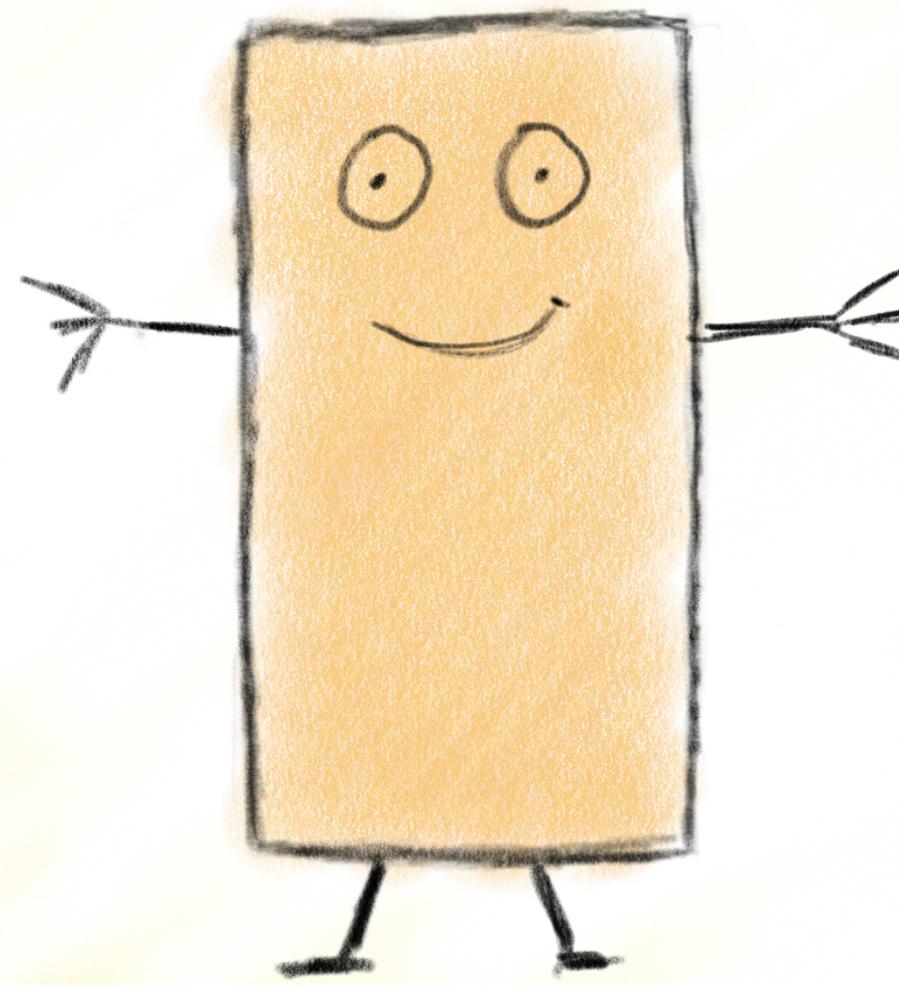
Rafael Mendiola
@groundcontrol <http://raf.dev>



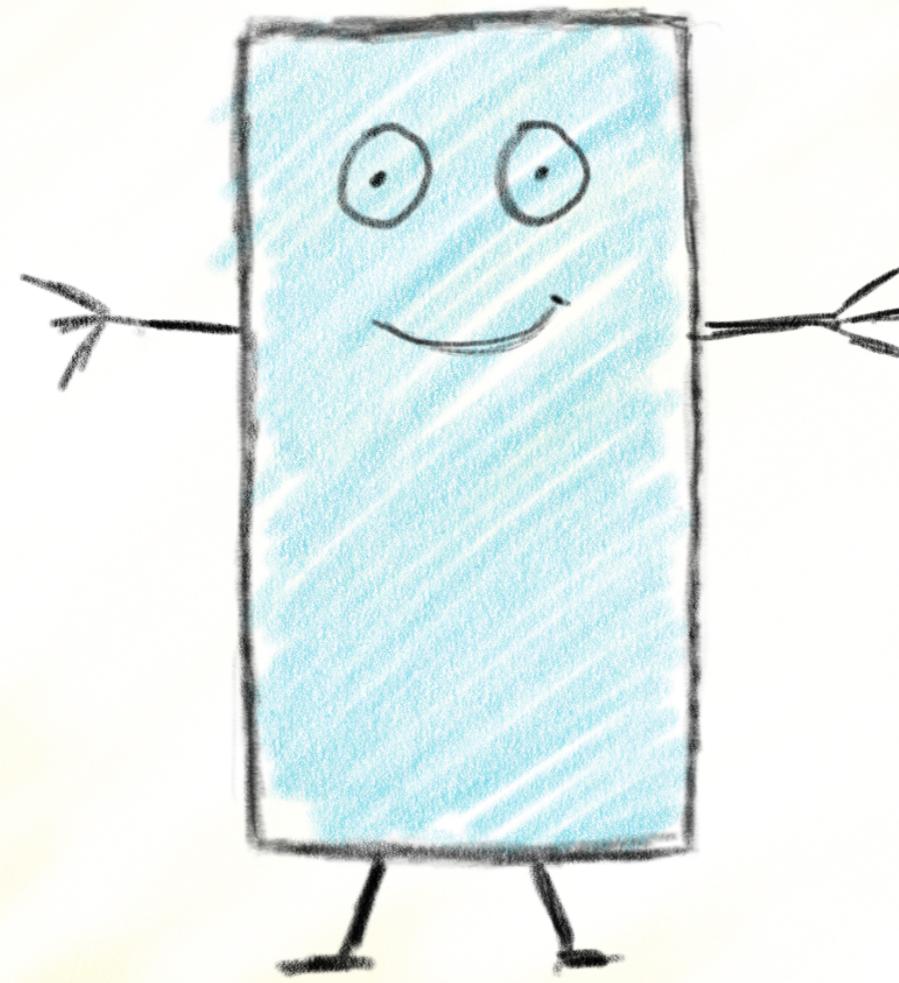
Shippy the App



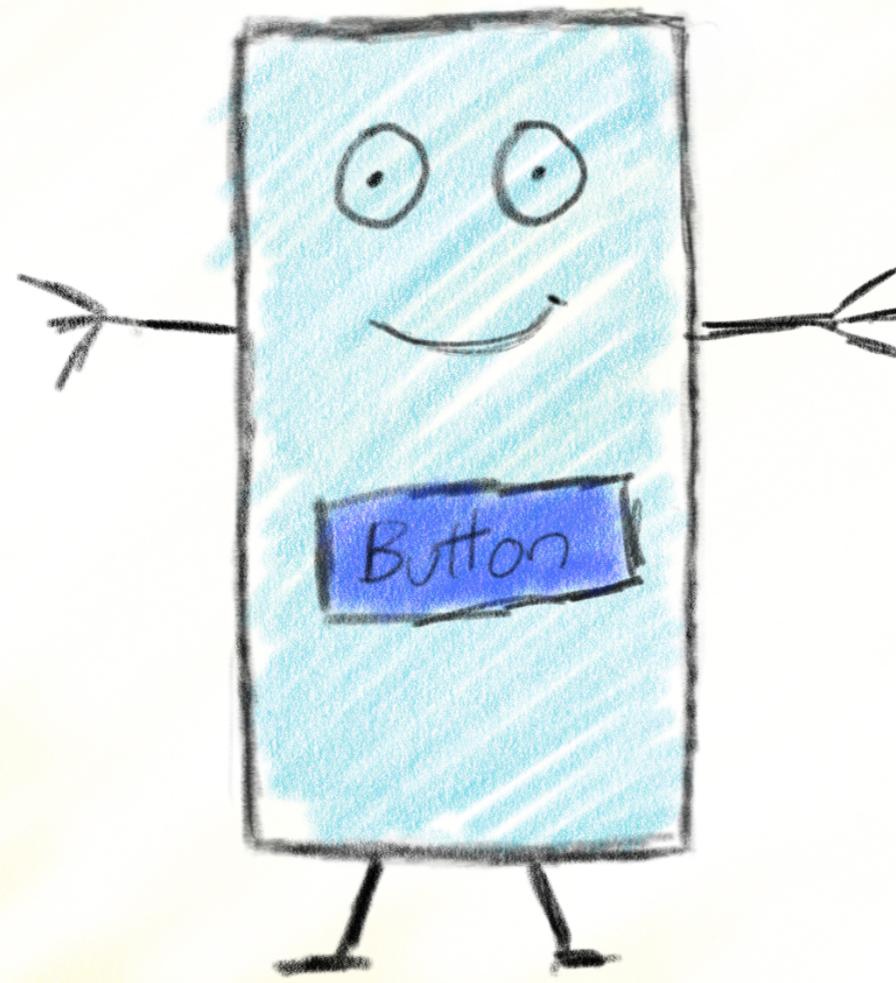
Shippy the App



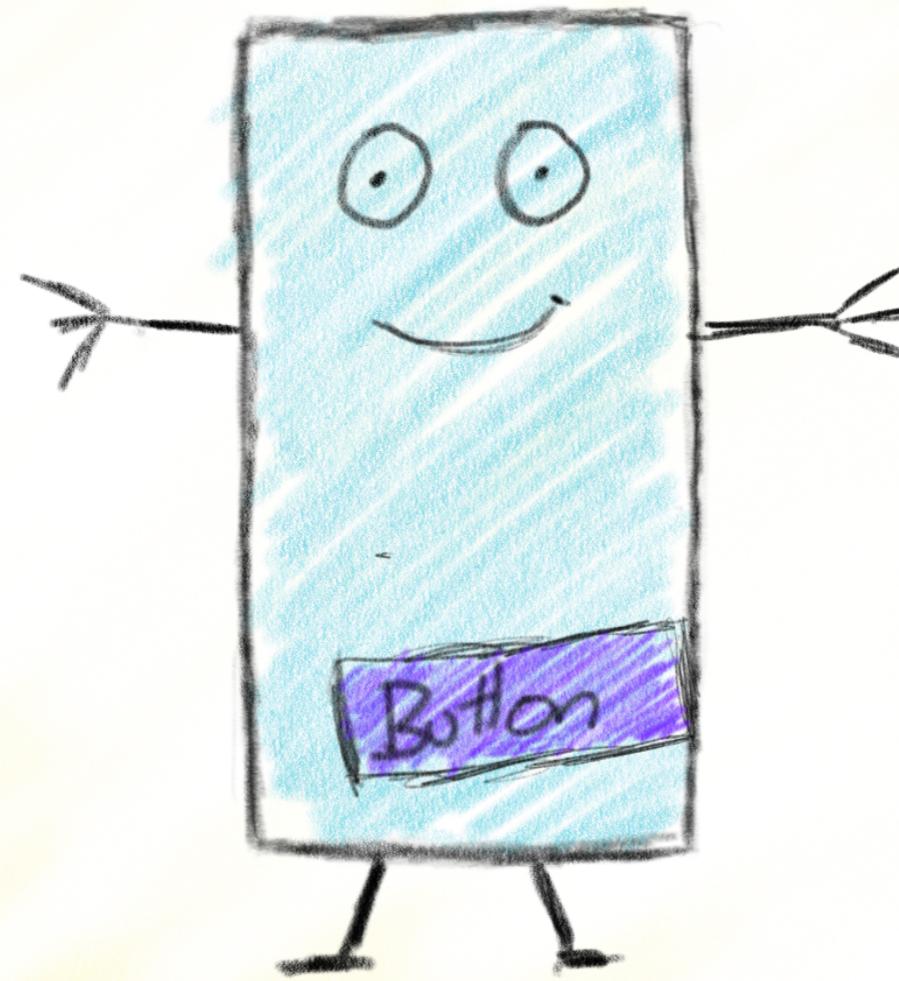
Shippy the App



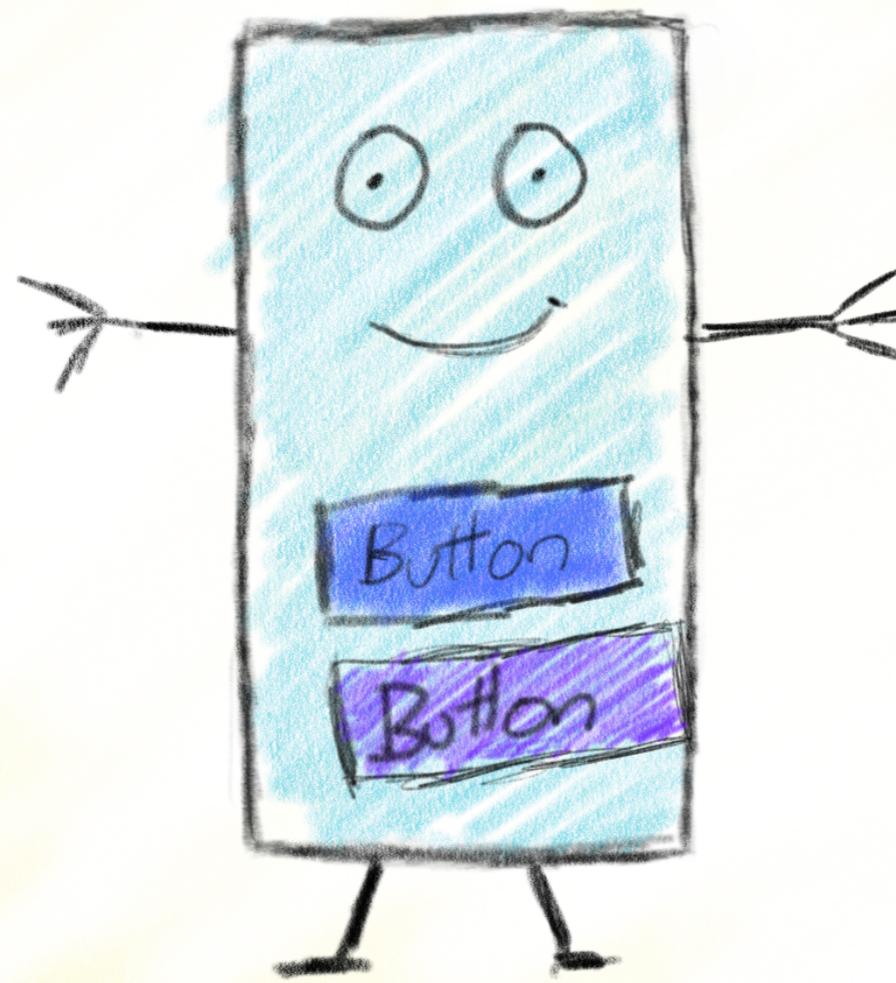
Shippy the App



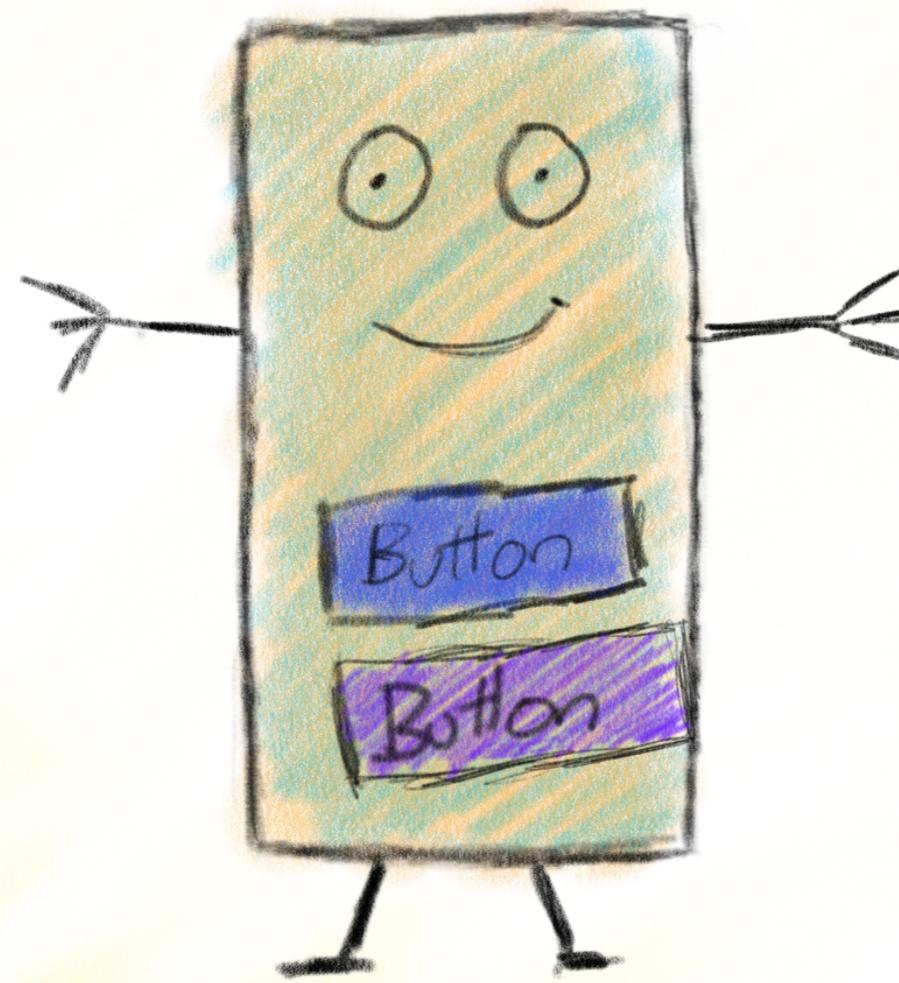
Shippy the App



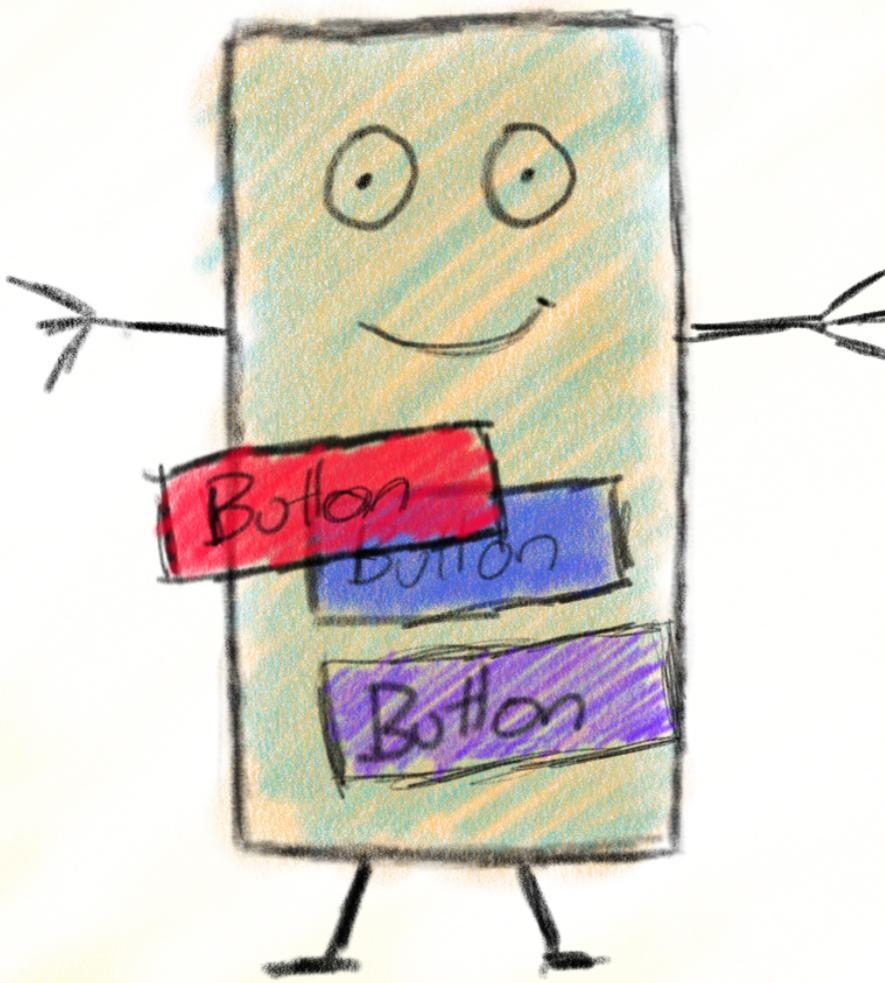
Shippy the App



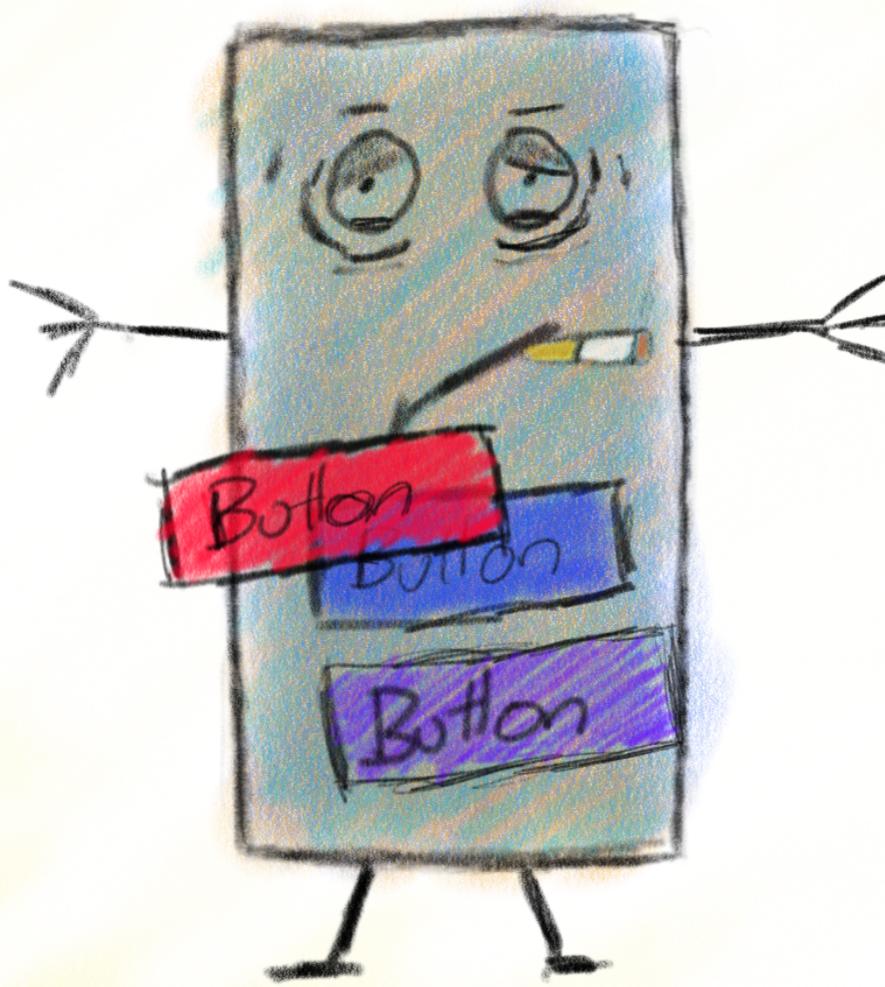
Shippy the App



Shippy the App



Shippy the App



Shippy the App

Most orgs leave safeguards until later:

"once we have more engineers..."

"once we have more bandwidth..."

Ironically, disciplining your codebase can help you the most *at the beginning*. If you wait until you're big enough, it's kinda too late no?

We use ESLint to set coding standards.

Can we use ESLint to set
Product Standards?



Writing Custom Lint Rules for Your Picky Developers



William Humphreys-Cloutier

[Follow](#)

May 9, 2017 · 5 min read

Engineers at Flexport are opinionated about the code we write. Instead of arguing about these opinions over and over again, we recently created a custom ESLint plugin to enforce these custom rules. It has been a bit over two months since we've had our custom ESLint plugin infrastructure in place. In that time a handful of engineers have created about 10 custom rules.

This article is meant as a practical guide to get a custom ESLint plugin off the ground in your organization, but could also be used to help write a rule for an existing open-source plugin.

WRITE MY OWN ESLINT PLUGIN?

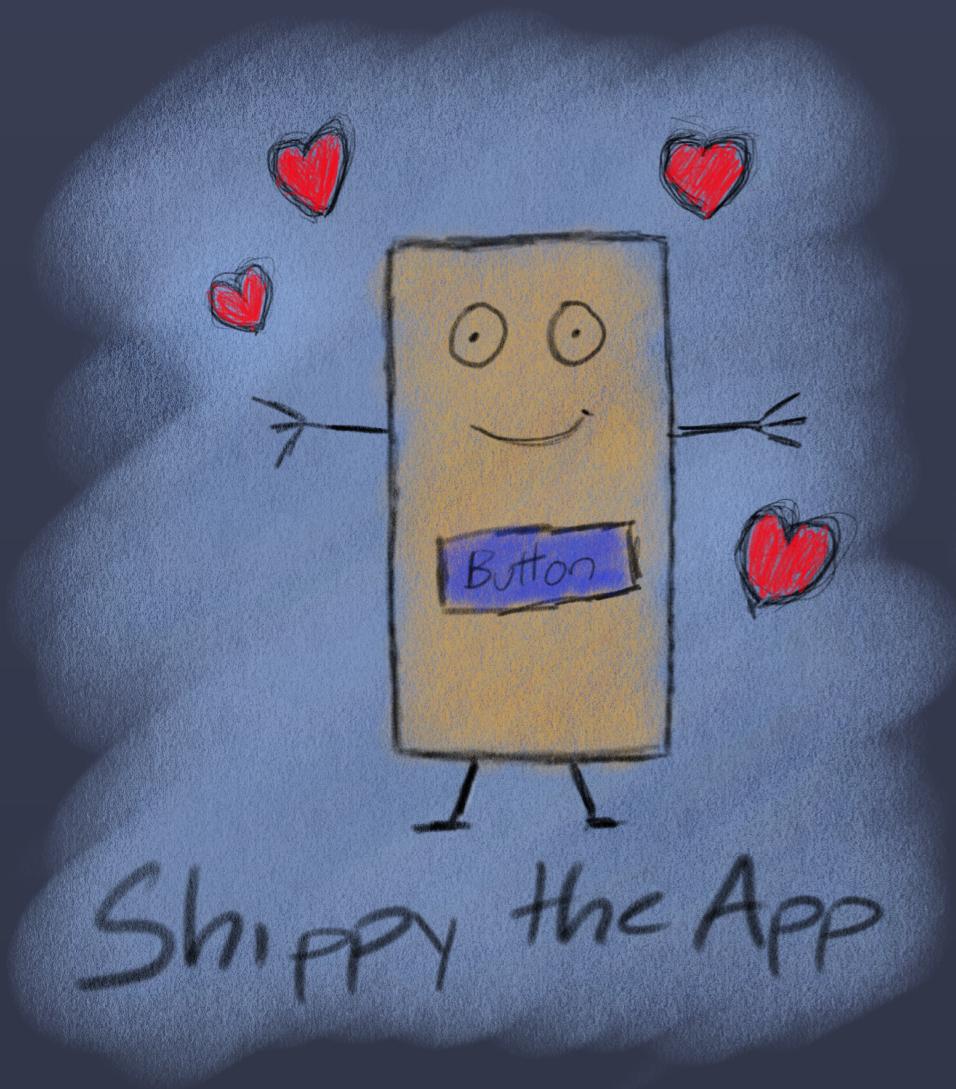


imgflip.com

I made you a plugin

```
$ yarn add -D eslint-plugin-design-system
```

Enforcing Design Systems With ESLint



Design systems aren't just for big companies.

Discipline your codebase early. Make sure that Shippy grows up happy and healthy.

You can create an ESLint plugin for your org. ASTs are fun!

Or you can use my plugin
`eslint-plugin-design-system`

Rafael Mendiola
@groundcontrol <http://raf.dev>