Augmenting Features and Tooling

Roland Guijt
MVP | CONSULTANT | TRAINER | AUTHOR

@rolandguijt rolandguijt.com

Module Overview

- Separating state and UI
- Type checking
- Routing
- Context
- Flux & Redux
- Ejecting
- Unit testing
- Progressive React apps

Separating State and UI

Arguably a best practice

Separation of concerns

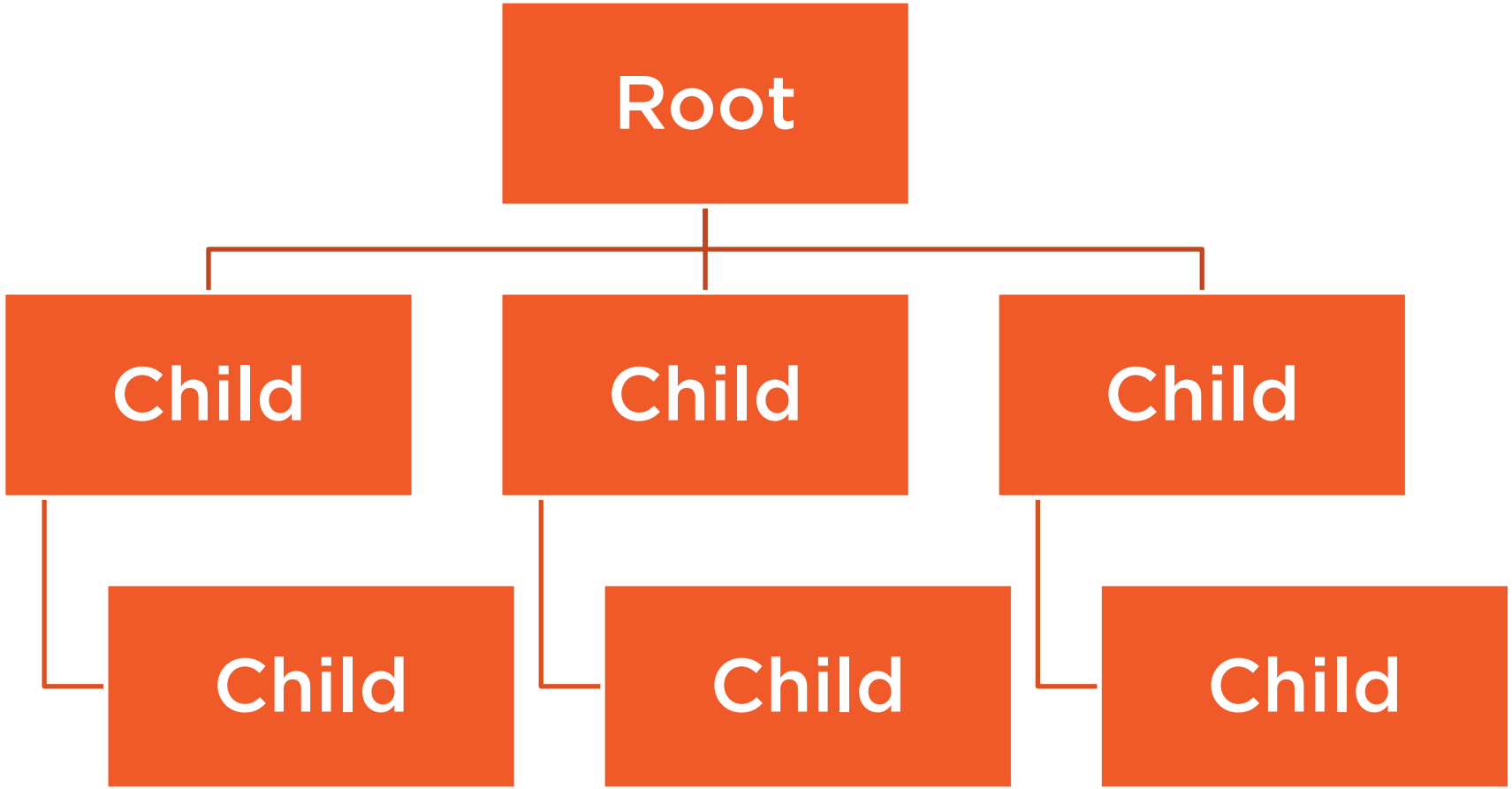
More complex

A Practical Start with TypeScript

React Router

Language

Context



Provider and Consumer

Support dynamic context

Automatic updating

Multiple consumers

Flux

Architectural pattern

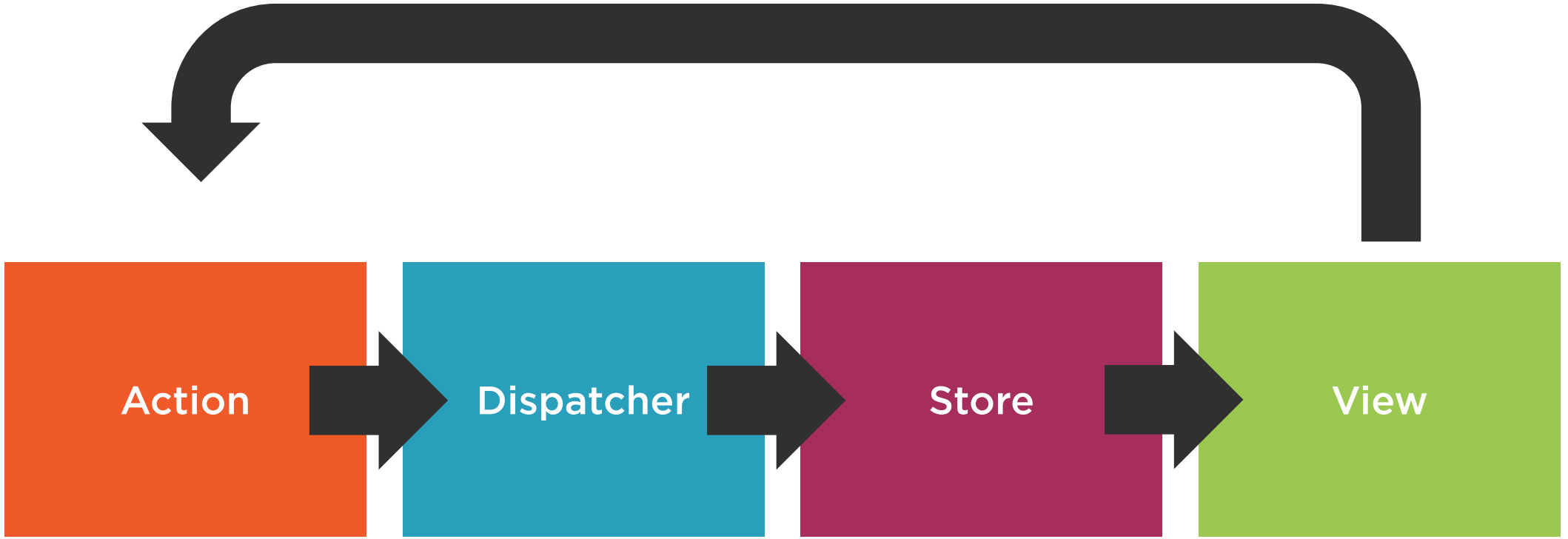
Redux is an implementation

You don't need it until you need it

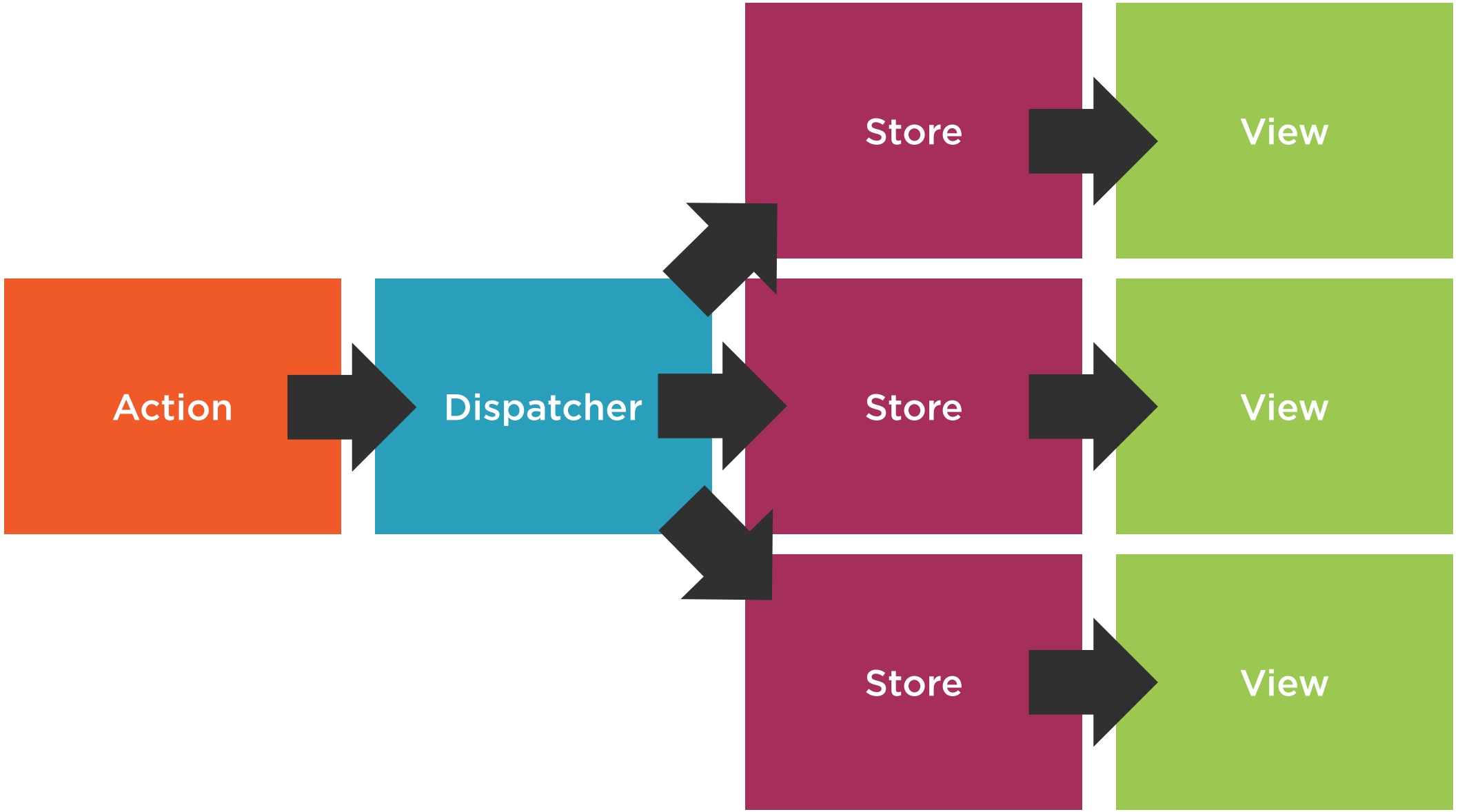
For complex applications

New concept around state and UI updates

Flux



Flux



https://redux.js.org

Ejecting: When you want to change the create-react-app default build configuration

https://4sh.nl/CustomReactScripts

https://4sh.nl/jest

Progressive Web App

Progressive Apps

Great for occasionally connected devices

Potentially extra work

Especially when adding server interaction

Summary

- Separation of concerns
- Easier debugging
- Routing URLs
- Sharing data
- Flux architecture
- Taking control of the build pipeline
- Unit testing
- Progressive web apps