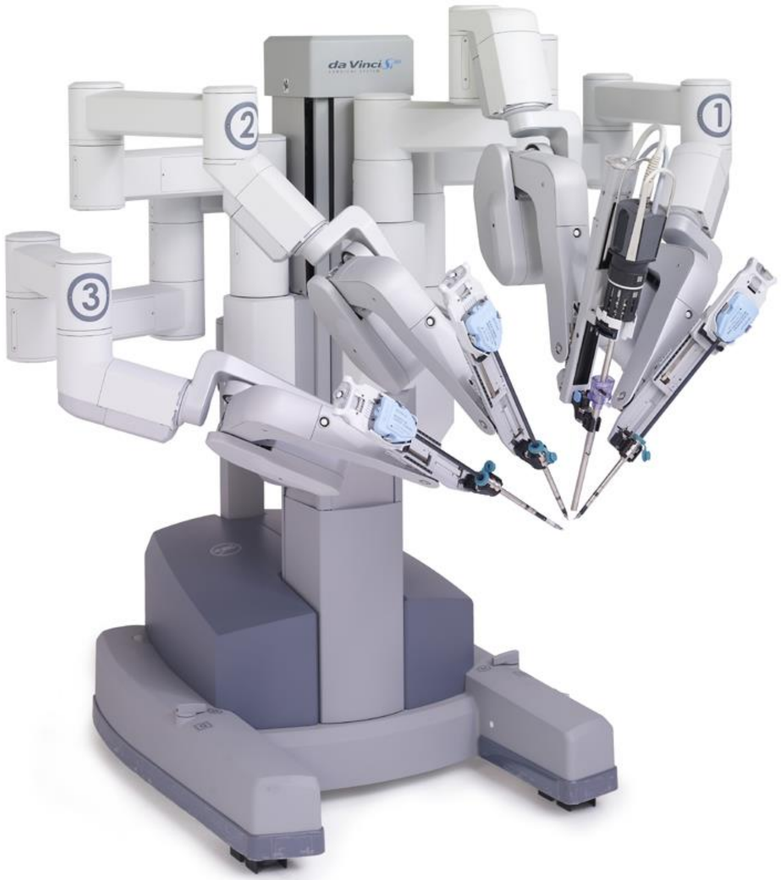
Control Add-in development supercharged

Vjekoslav Babic

When you are passionate about
Microsoft Dynamics NAV/365 Business Central


da Vinci S
1
2
3

This session is about…

NAV TECH DAYS 2019
mibuso.com

Making control add-in development work

Practices that work
… and practices that don’t
… and why they work and why they don’t

Developing for web like web developers do
… not like accountants do

Practices that don't work

NAV TECH DAYS 2019
mibuso.com

Hoping to do
36 demos
in a single session

Building control add-ins is easy... in AL

NAV TECH DAYS 2019
mibuso.com

- All-in-one environment
- Manifest and interface in one place
- Familiar language and type system
- Free folder structure

```
controladdin Simpler
{
    Scripts =
        'src/add-in/scripts/numeral.min.js',
        'src/add-in/scripts/simpler.js';

    StartupScript = 'src/add-in/scripts/start.js';

    StyleSheets = 'src/add-in/styles/simpler.css';

    RequestedHeight = 400;
    RequestedWidth = 600;
    MaximumHeight = 1200;
    MinimumHeight = 200;

    HorizontalShrink = false;
    HorizontalStretch = true;
    VerticalShrink = true;
    VerticalStretch = true;

    1 reference
    event OnControlReady();

    1 reference
    procedure SendData(Data: JsonArray);
}
```

Manifest

Interface

easy is tricky

Global scope pollution

Performance

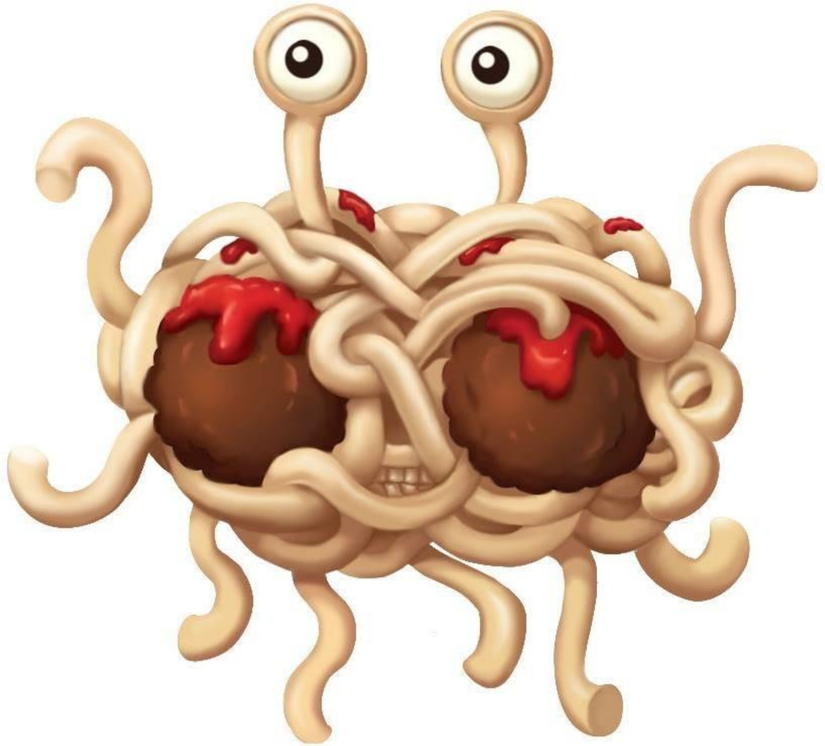
Debugging

Browser compatibility

Separation of concerns

Practices that don’t work

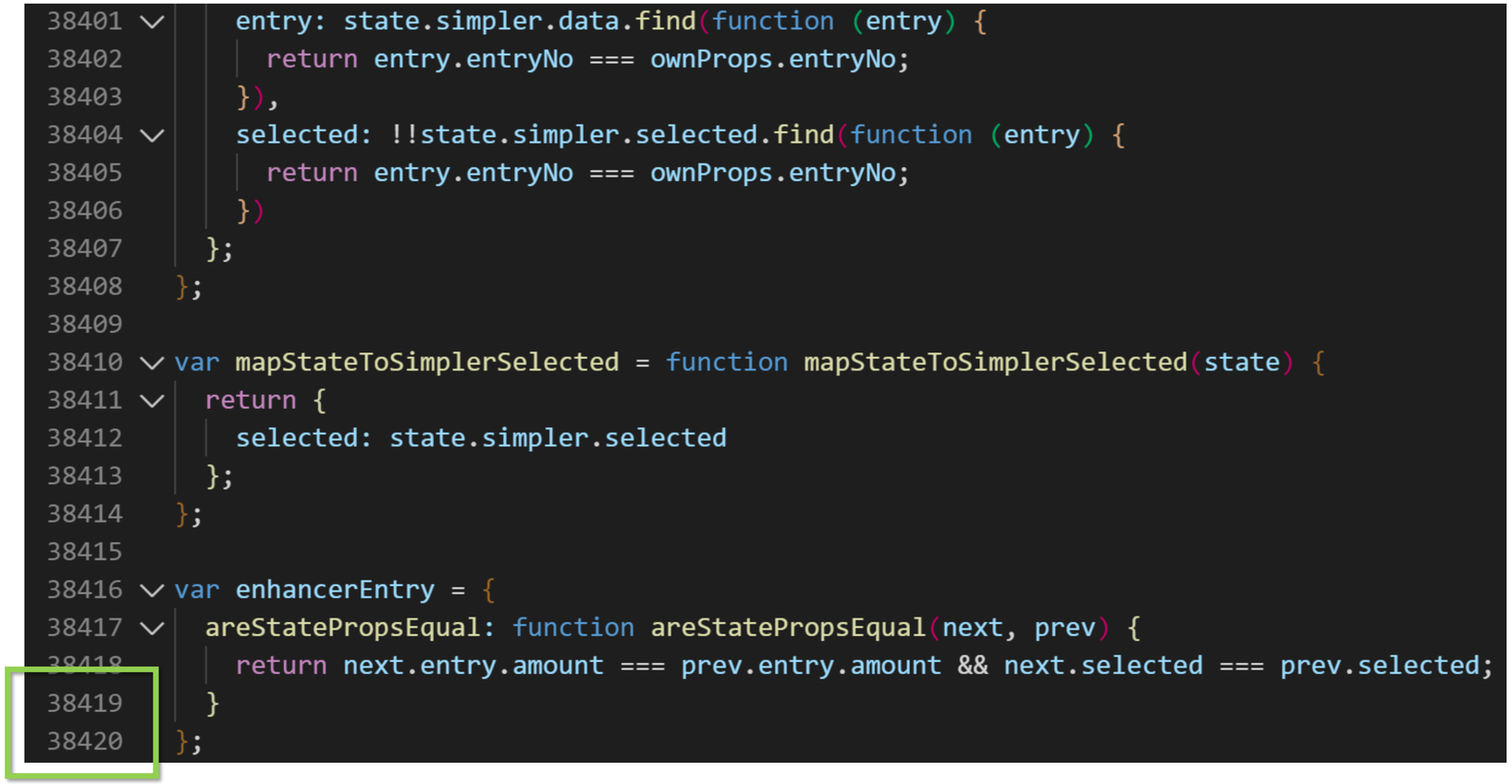
Developing in
one file



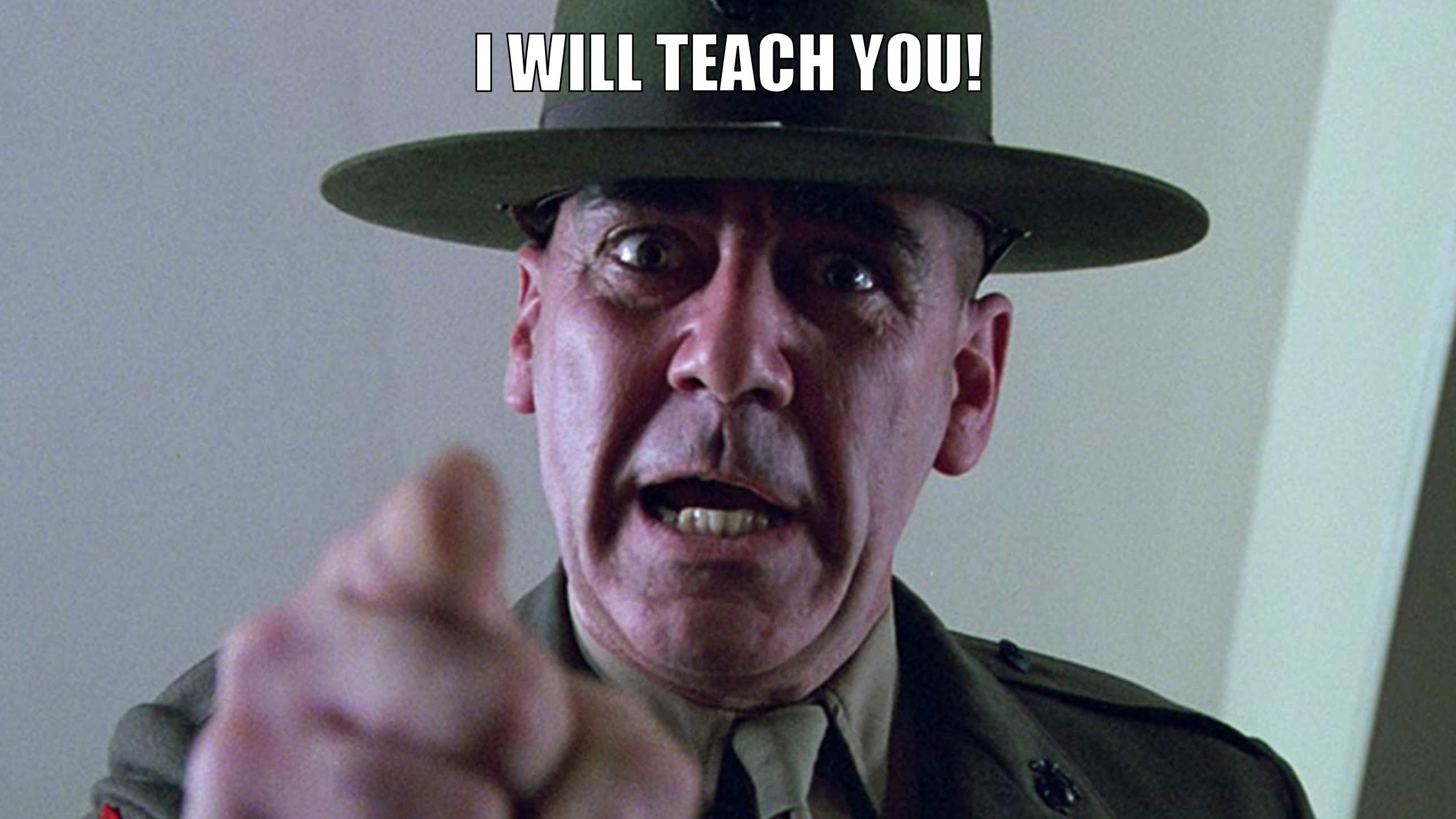
Practices that don’t work

Good for browser.

Not good for you.



```
38401      entry: state.simpler.data.find(function (entry) {
38402        return entry.entryNo === ownProps.entryNo;
38403      }),
38404      selected: !!state.simpler.selected.find(function (entry) {
38405        return entry.entryNo === ownProps.entryNo;
38406      })
38407    };
38408  };
38409
38410  var mapStateToSimplerSelected = function mapStateToSimplerSelected(state) {
38411    return {
38412      selected: state.simpler.selected
38413    };
38414  };
38415
38416  var enhancerEntry = {
38417    areStatePropsEqual: function areStatePropsEqual(next, prev) {
38418      return next.entry.amount === prev.entry.amount && next.selected === prev.selected;
38419    }
38420  };
```


I WILL TEACH YOU!


YOU BUILT A NEW JAVASCRIPT FRAMEWORK?
I WILL FIND YOU, AND I WILL KILL YOU!

Front-end frameworks

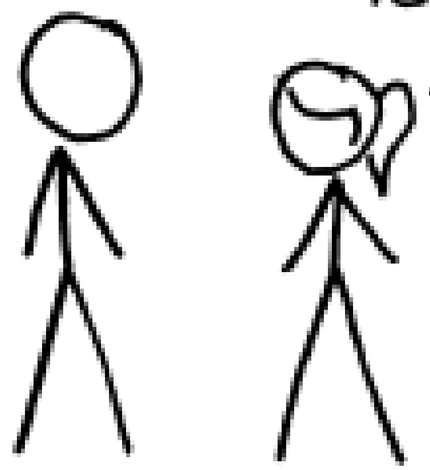

polymer
Mithril
aurelia
slim.js
deku
RIOT
canjs
BACKBONE.JS
marko
Knockout.
UIkit
PREACT
Ext JS
SVELTE
RJS
ember

HOW STANDARDS PROLIFERATE:

(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC.)

SITUATION:
THERE ARE
14 COMPETING
STANDARDS.

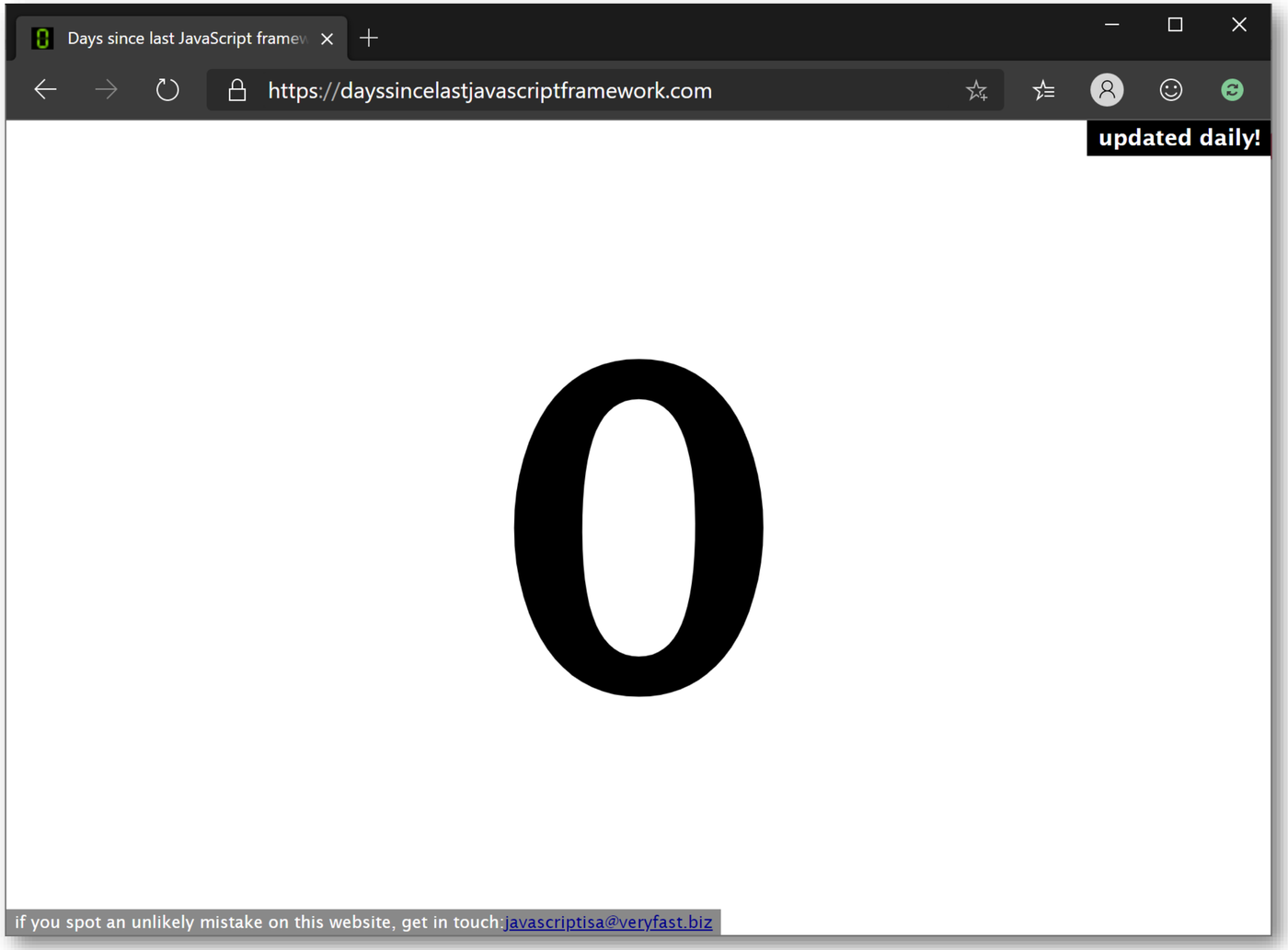
14?! RIDICULOUS!
WE NEED TO DEVELOP
ONE UNIVERSAL STANDARD
THAT COVERS EVERYONE'S
USE CASES.



SOON:

SITUATION:
THERE ARE
15 COMPETING
STANDARDS.

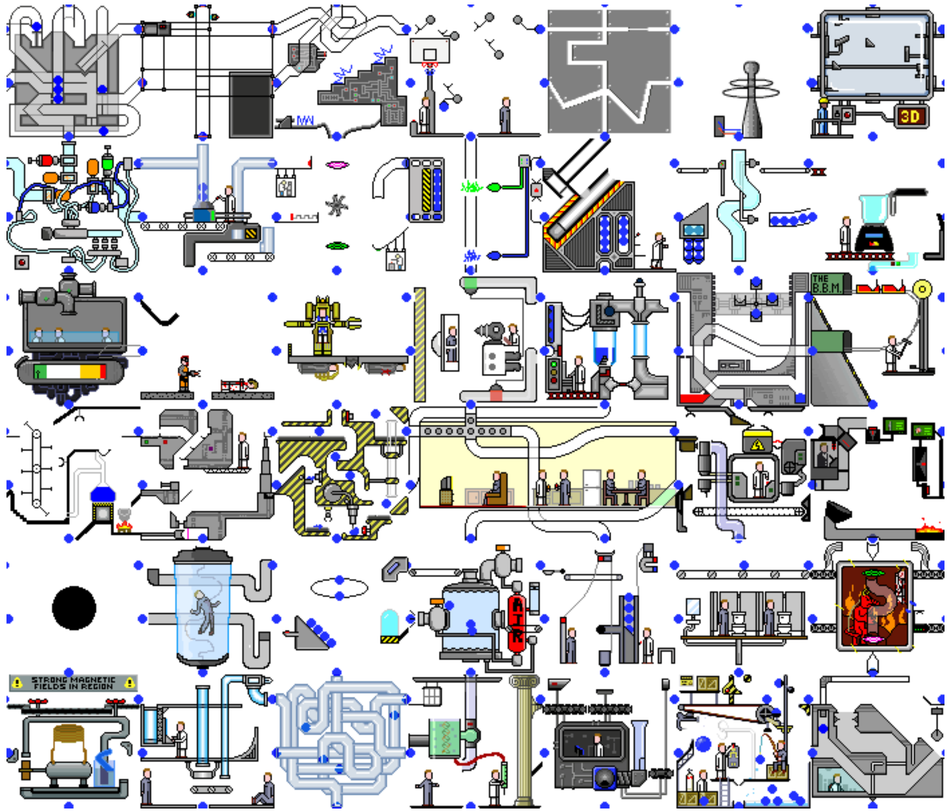
DaysSinceLastJavaScriptFramework.com



Practices that don't work

NAV TECH DAYS 2019
mibuso.com

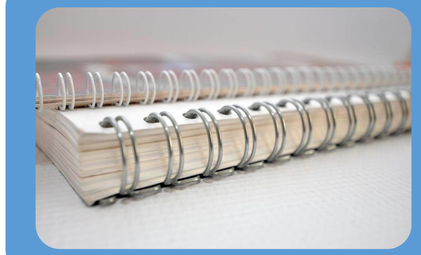
Developing a
yet another
JavaScript framework



Let’s forget about framework
(for a while)

Let’s talk bundles

JavaScript bundles



Group separate files into one

Reduce HTTP traffic

Improve page loading and processing

What is gulp

- https://gulpjs.com/

- “The streaming build system”

- Toolkit for automating development tasks
- Based on JavaScript and Node.js
- Integrates with VS Code

What can you use gulp for?

- 12,307 gulp-related packages on npm
- 4,074 gulp plugins

- Bundling
- Automating transpilation
- Minifying JavaScript, CSS, HTML, even images
- Linting
- Automating git tasks (especially complex ones)
- Compressing / decompressing files
- Run shell commands (including PowerShell)

What is Babel

- https://babeljs.io/

- Babel is a JavaScript compiler
- Turns next (or any) generation JavaScript into compatible code

- Things it does:
  - Transforms syntax
  - Polyfills missing features
  - Optimizes code
  - And more…

- You can integrate Babel with gulp

What (else) can gulp do for AL and BC?

NAV TECH DAYS 2019
mibuso.com

- Build Control Add-ins for pre-AL and pre-BC
  - Bundle
  - Zip
  - Deploy through PowerShell

- Build documentation from source AL files
- Extract Control Add-in resource file from *.app files
- Automatically maintain `controladdin` definition

- Probably a lot more, but you got the idea


ROAD
CLOSED
DETOUR

Practices that don’t work

Keeping Node.js inside the AL workspace

Practices that work

Use Multi-root Workspaces

Multi-root Workspaces

NAV TECH DAYS 2019
mibuso.com

- Combine multiple workspaces into one
- Useful for related projects
  (especially when different technologies are involved)

- Each workspace can have its own:
  - Debug/launch configurations
  - Compilers and setup
  - Extensions
  - Even git

AL with Multi-root Workspaces

NAV TECH DAYS 2019
mibuso.com

- AL doesn't play nice with VS Code

- Hijacks F5
- Replaces build task with its own
- "Detects" if project type is AL, and if it is…
  … *treats every loaded workspace as AL*
  … even injects its artifacts into other workspaces

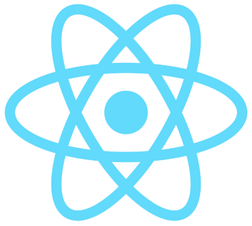
Remember frameworks?

Front-end frameworks


polymer
Mithril
aurelia


slim.js
deku


RIOT



canjs

marko

BACKBONE.JS

Knockout.

UIkit

PREACT

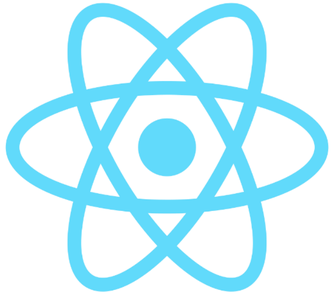
Ext JS

SVELTE

RJS

ember

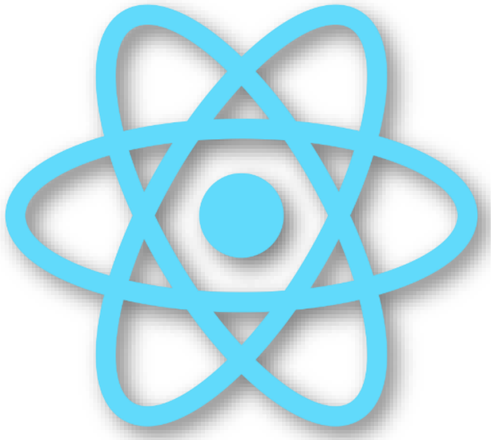
The big three



Angular
(from Google)

Superheroic JavaScript
Model-View-Whatever
Framework

The big three

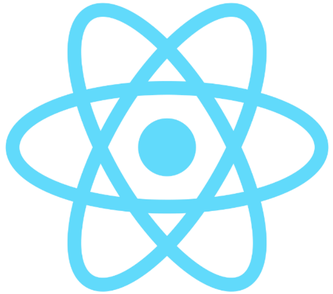


React

(from Facebook)

A JavaScript library for building user interfaces

The big three



Vue

(from Evan You)

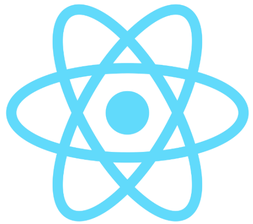
Reactive component-oriented view layer for modern web interfaces

What do these three have in common

NAV TECH DAYS 2019
mibuso.com

- Don’t just work in browsers (kind of)
- Require pre-compilation (for full functionality)
- Depend on toolchains (for best productivity)

- Huge developer base
- Deep knowledge base
- Wide learning resources availability

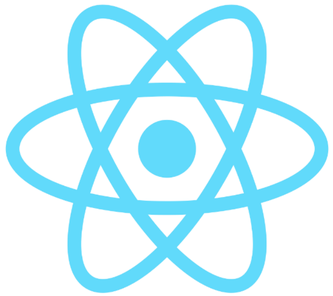

- Great support in VS Code

Which one to choose?

NAV TECH DAYS 2019
mibuso.com

- All three are great
- All three use bundles
- All three will work with Business Central

- Choose whatever suits your development style/philosophy the best

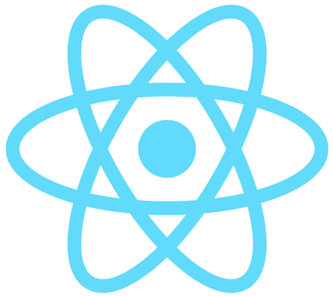




Which one to choose?

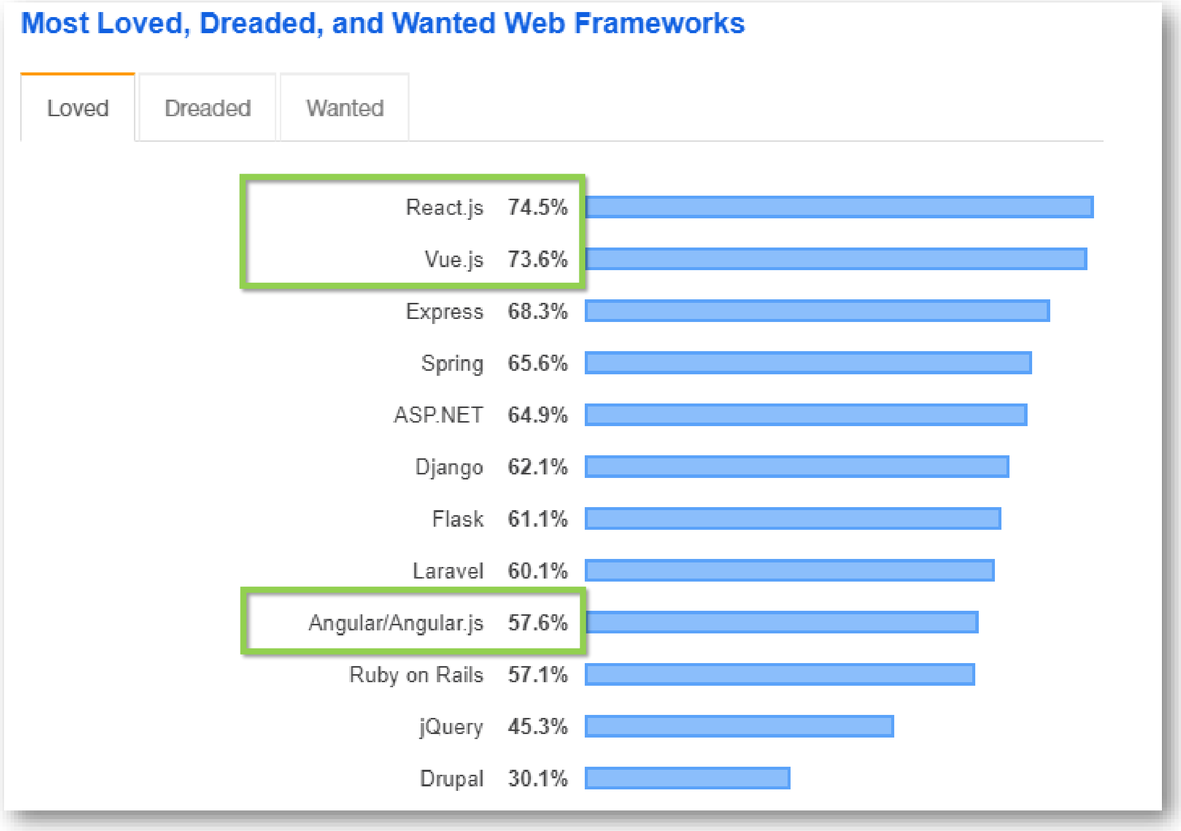
NAV TECH DAYS 2019
mibuso.com

- All three are great
- All three use bundles
- All three will work with Business Central
  (as a matter of fact, you can make all JavaScript frameworks work with Business Central)

- Choose whatever suits your development style/philosophy the best

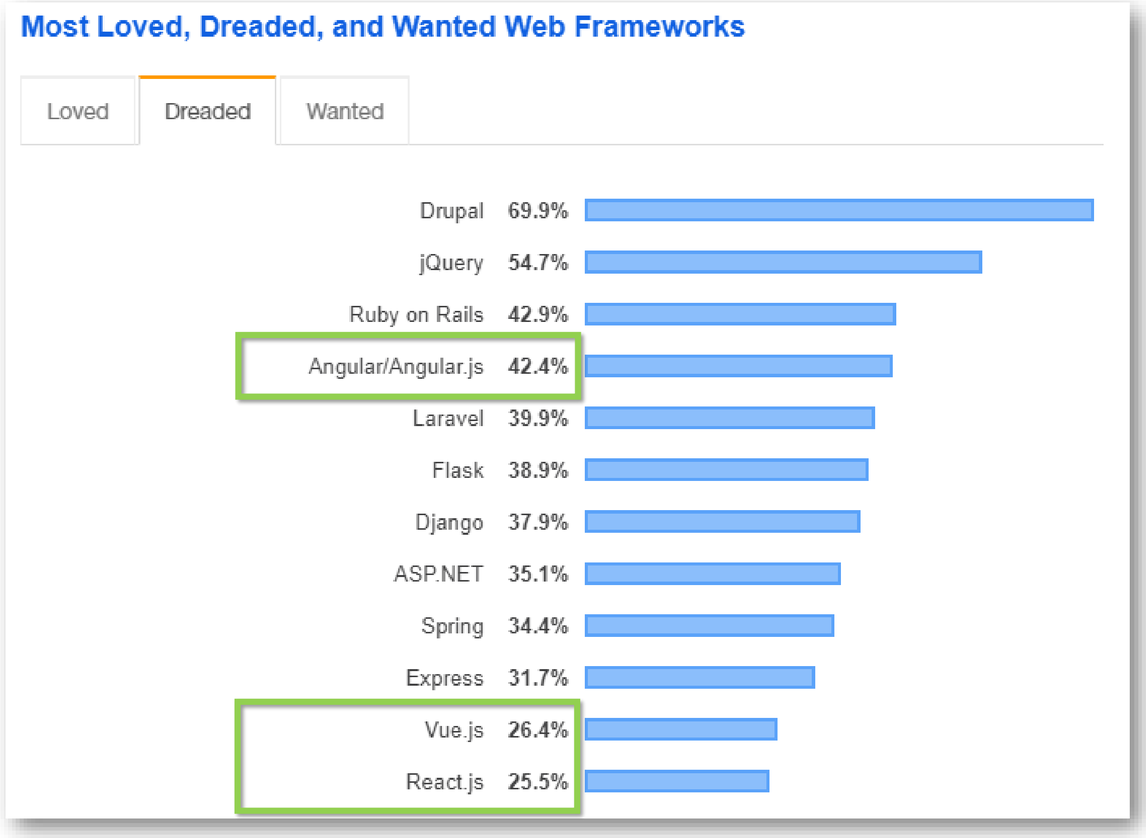




Which one to choose?



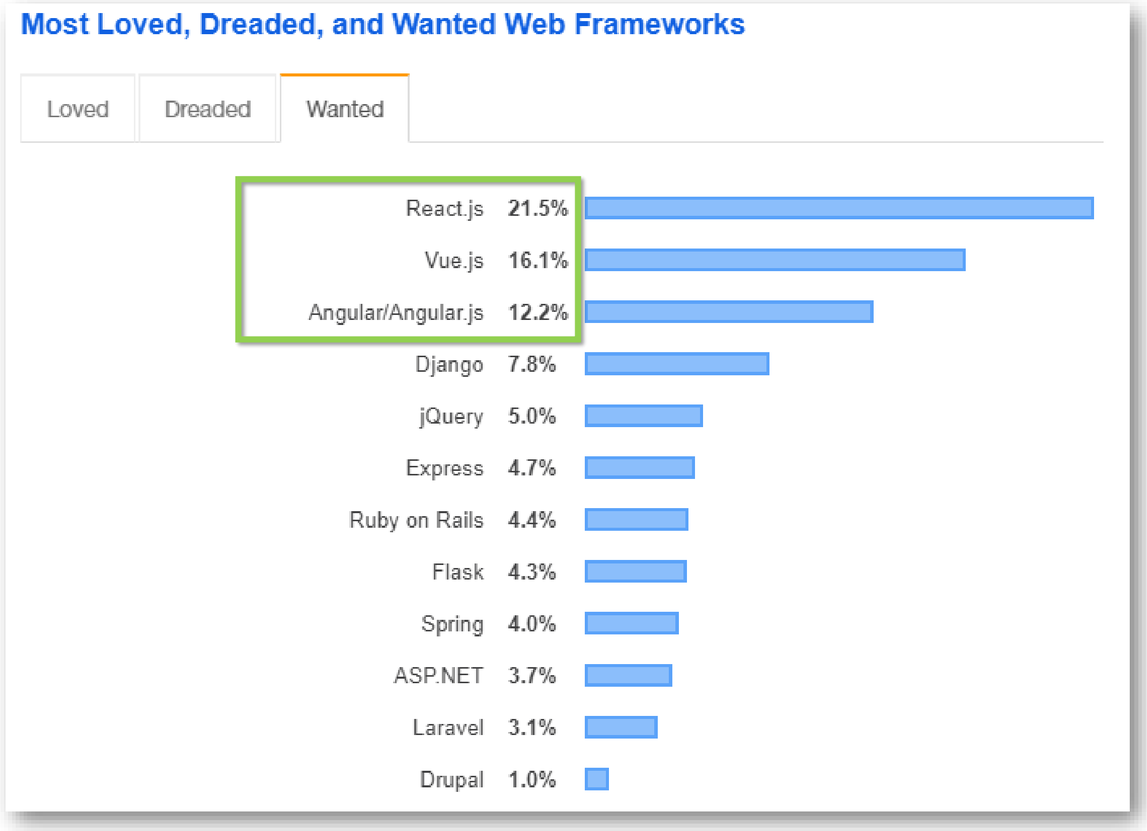
stackoverflow

Which one to choose?



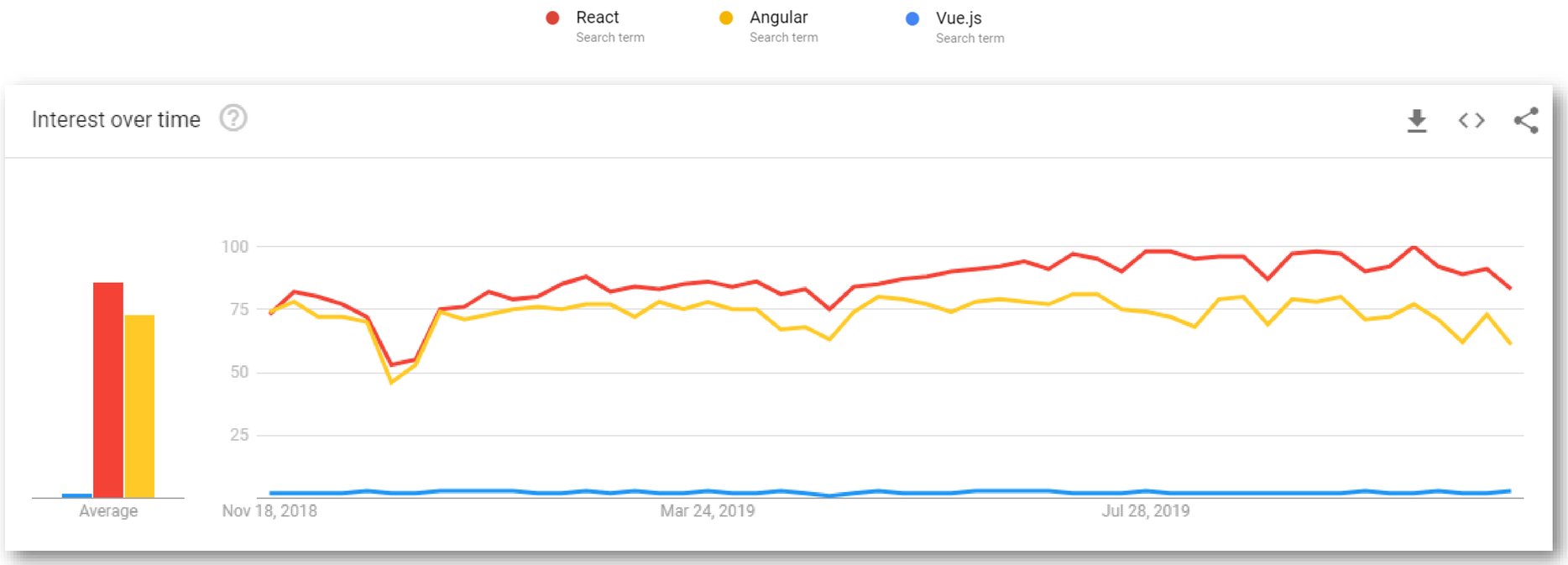
stackoverflow

Which one to choose?



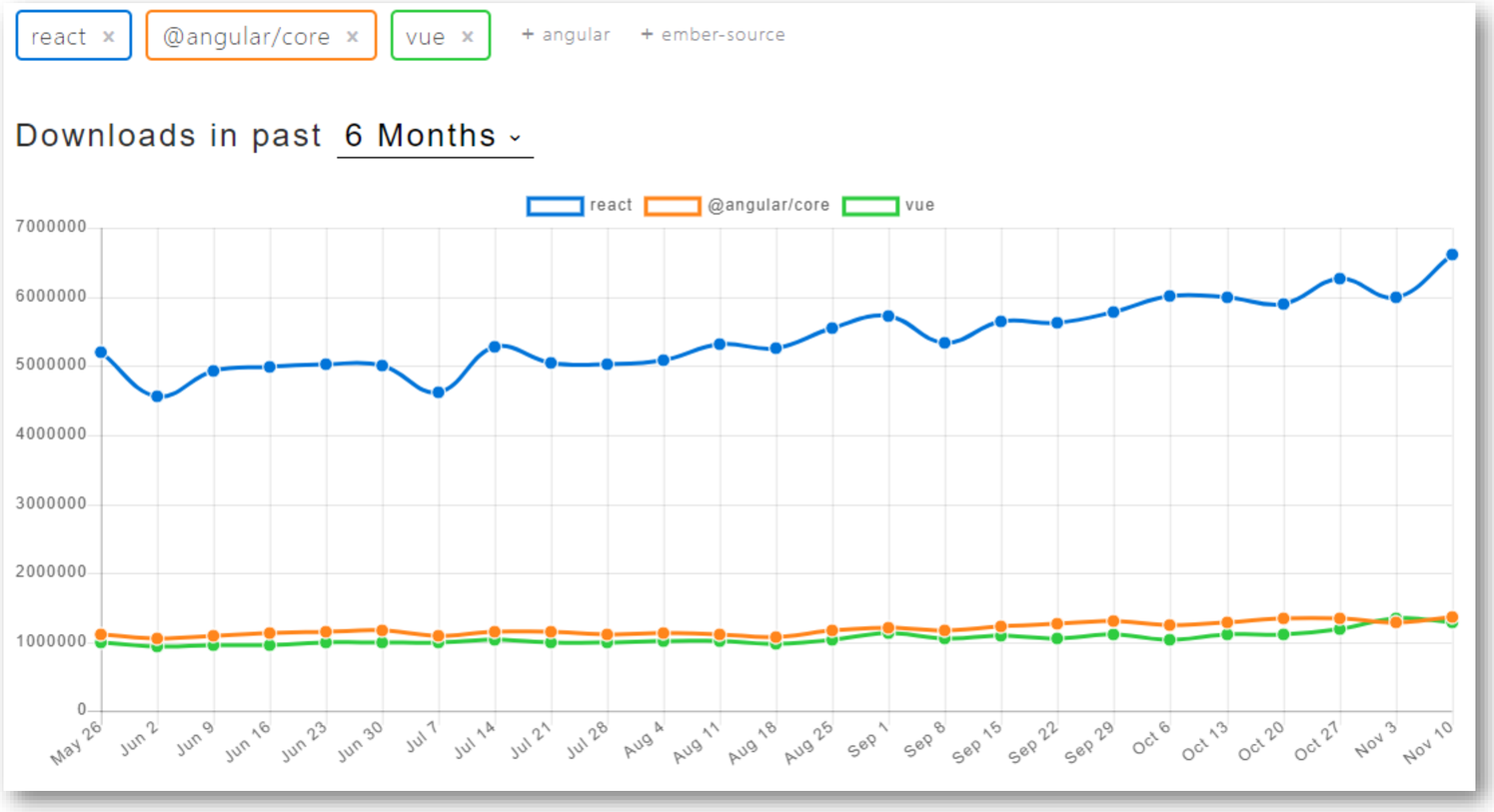
stackoverflow

Which one to choose?



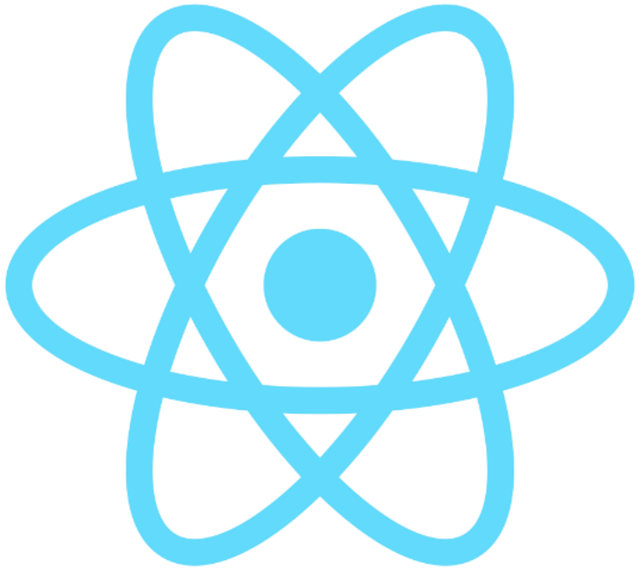
Google

Which one to choose?



Which one did I choose, and why?

I choose React

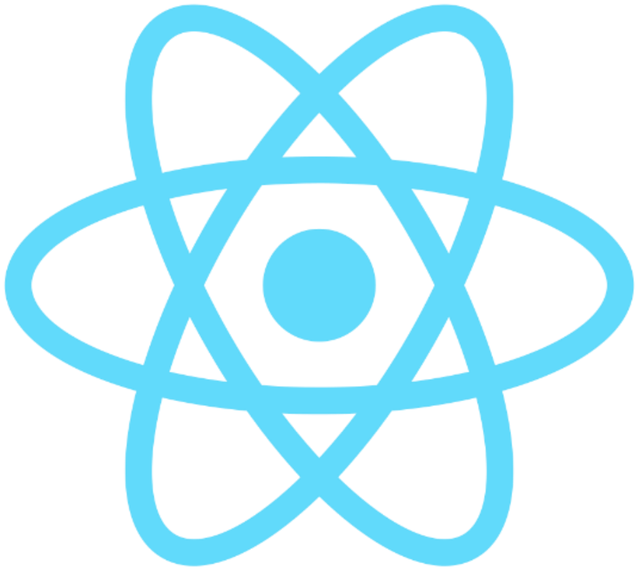


Why React?

- Straightforward and simple
- Component based (vs. template based)
- Imperative (vs. declarative)
- Flexible about architecture

- Fits best with how AL and JavaScript work together

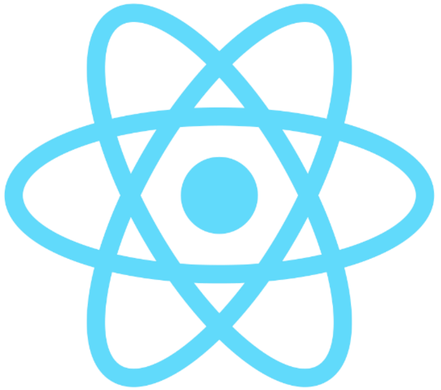
- Numbers do matter. React is #1.



Getting started with React

NAV TECH DAYS 2019
mibuso.com

Create-react-app
https://code.visualstudio.com/docs/nodejs/reactjs-tutorial



React Starter Kit
https://reactstarter.com/

Yeoman + Cheetah
https://bitbucket.org/MarcAdlington1/cheetah/src/master/

Yeoman + generator-react-webpack
https://github.com/react-webpack-generators/generator-react-webpack

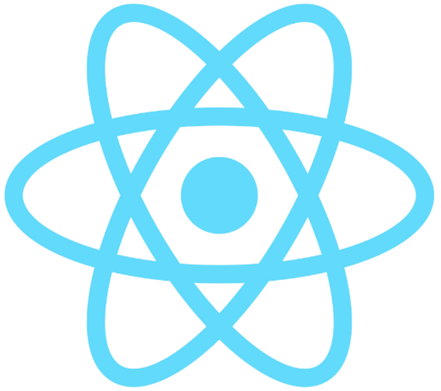
Manually from scratch
https://blog.usejournal.com/creating-a-react-app-from-scratch-f3c693b84658

My preference: Manually from scratch

NAV TECH DAYS 2019
mibuso.com

Takes 2 minutes to set up

Complete control over the content and setup



Completely unopinionated, especially about:
- Package manager: yarn or npm, both okay
- Version of dependencies
- Test framework
- State management

Minimum dependencies installed
- No unnecessary dependencies
- Maybe hot loader, but you can simply skip the step

What is webpack

NAV TECH DAYS 2019
mibuso.com

https://webpack.js.org/

- Static module bundler
- Processes all application source code
- Generates a bundle file
  (or more, depending on your configuration)

- Heavily configurable
  (through loaders and plugins)

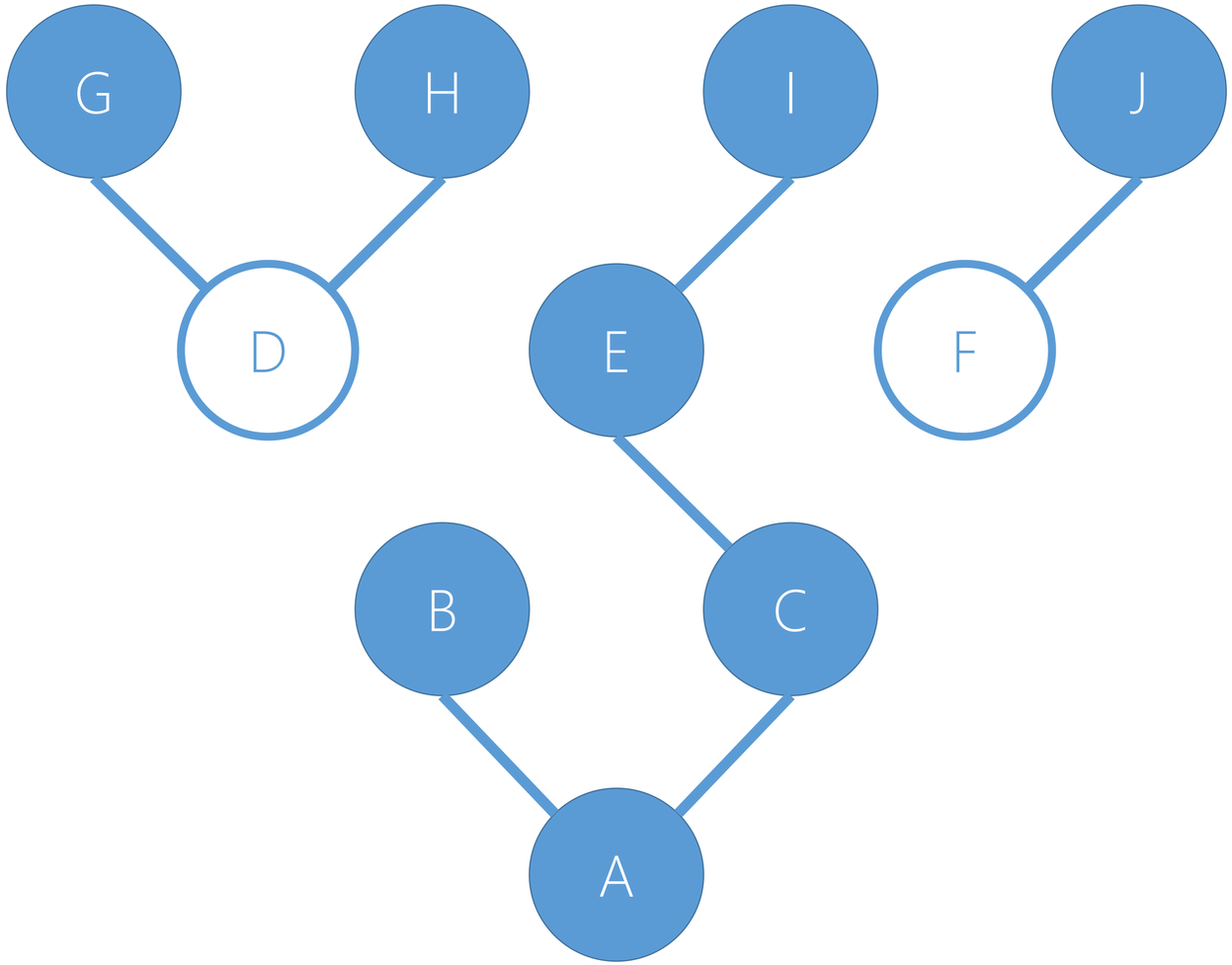
webpack output bundle

NAV TECH DAYS 2019
mibuso.com

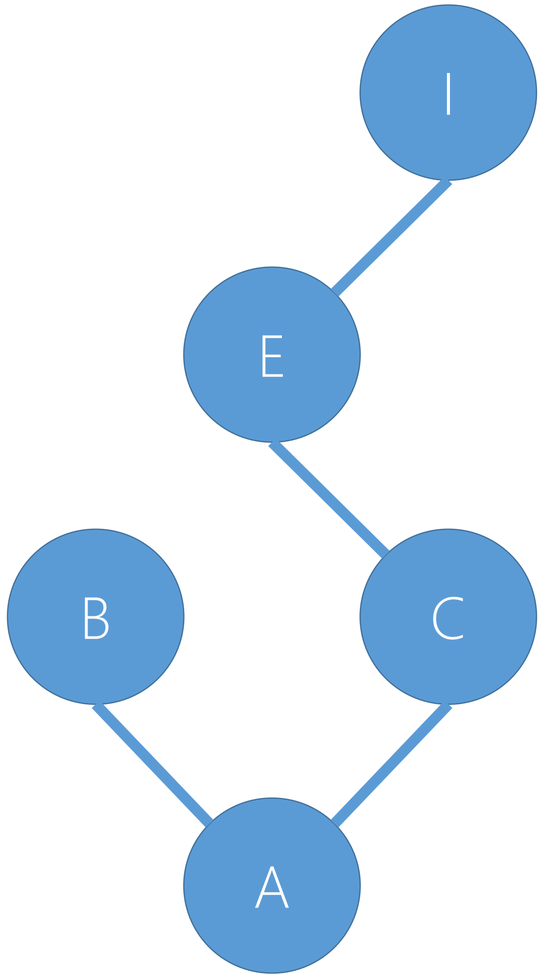
- A single file containing all code
- Built from dependency graph
- Code is transformed and optimized

- Not limited only to code. It can contain:
  - Stylesheets
  - Image resources
  - Other kinds of assets

webpack treeshaking optimization



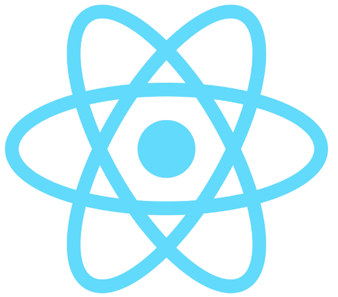
webpack treeshaking optimization



webpack with React

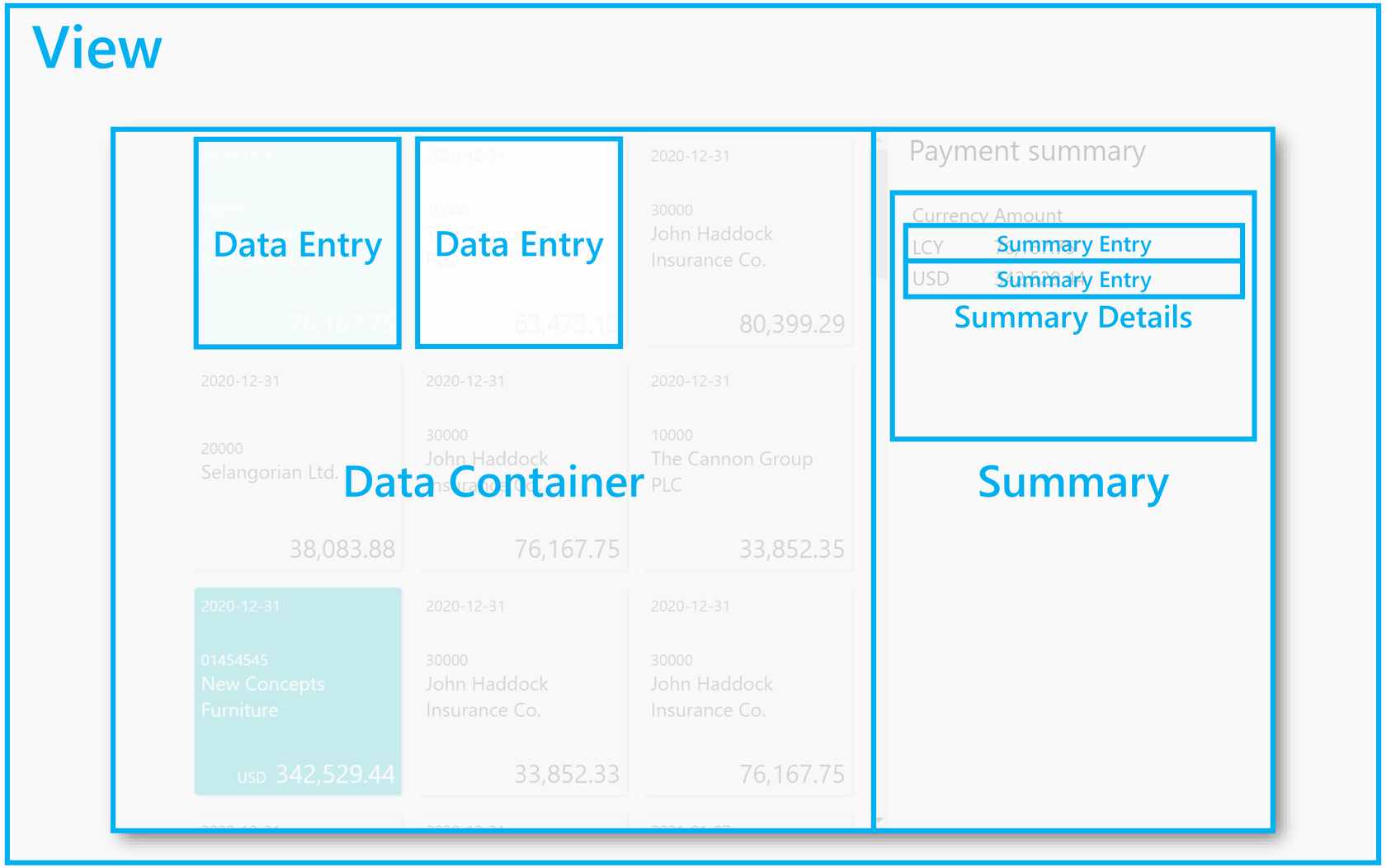
- React uses specific JavaScript dialect: JSX
- Most React code is either ES6 or TypeScript
- Browsers don’t natively process any of this

- Uses babel loader to transpile JSX
- Bundles all transpiled code (including dependency code) into a single bundle file



```
class HelloWorld extends Component {
    render() {
        return <h1>Hello, World!</h1>;
    }
}
```

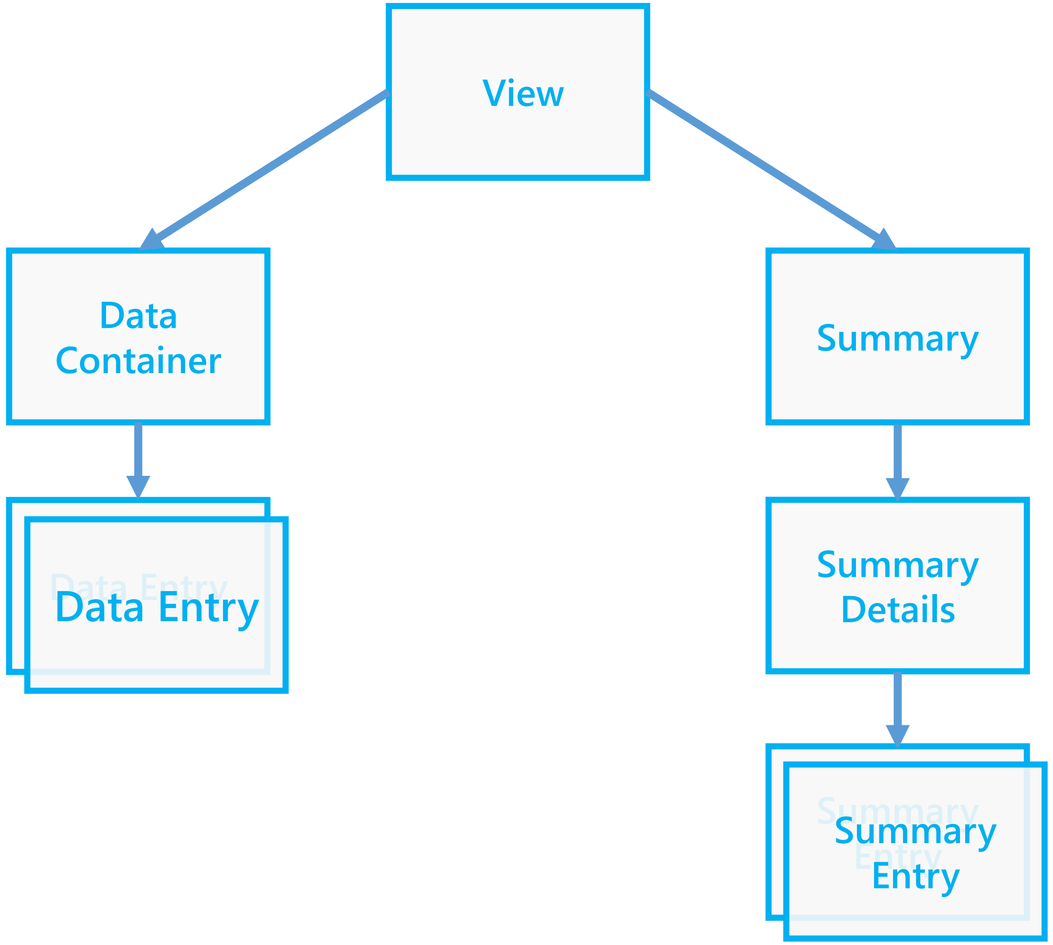
Simpler in React



Simpler in React

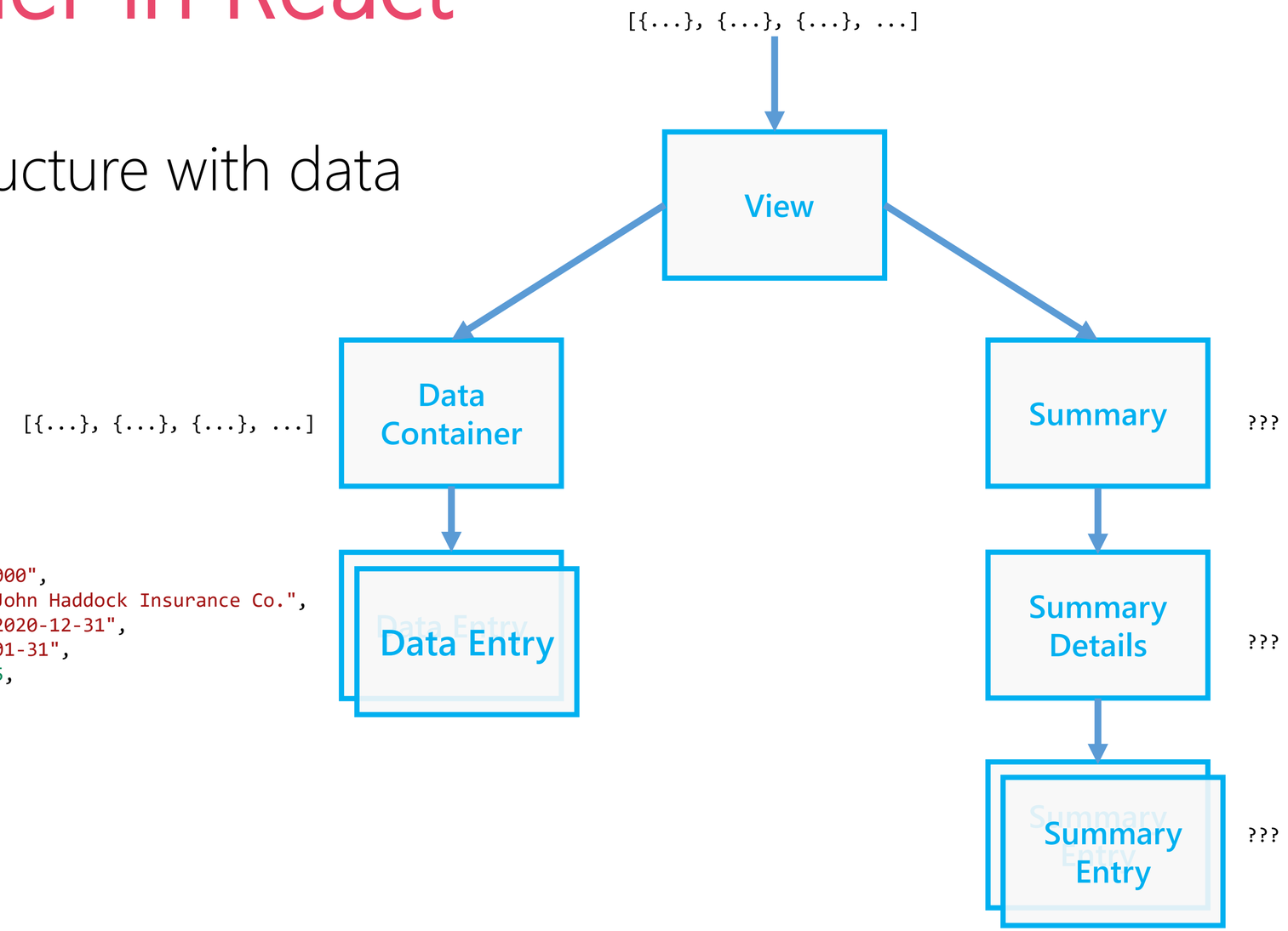
NAV TECH DAYS 2019
mibuso.com

Basic structure



Simpler in React

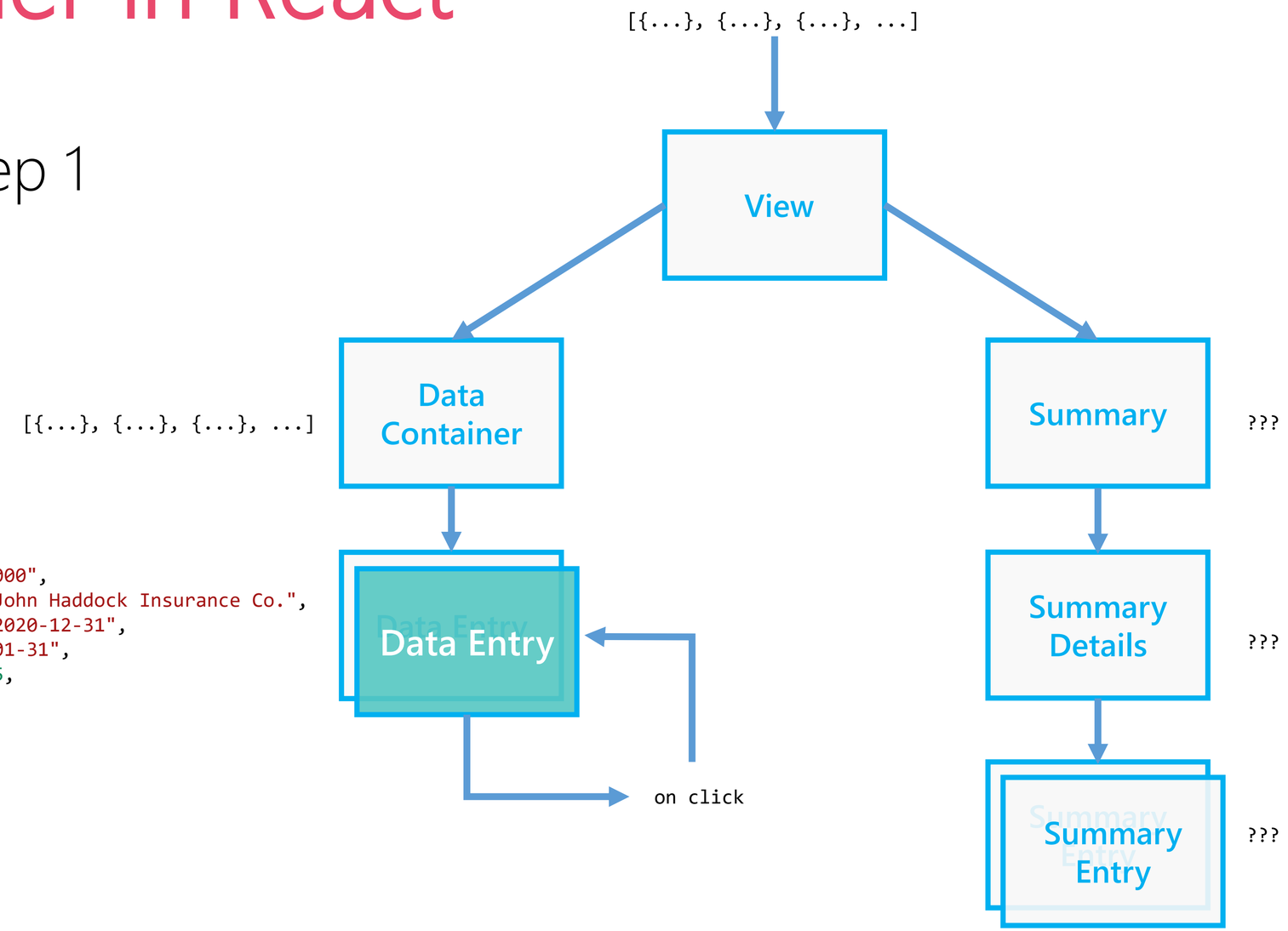
Basic structure with data




```
{
  "entryNo": 2433,
  "customerNo": "30000",
  "customerName": "John Haddock Insurance Co.",
  "documentDate": "2020-12-31",
  "dueDate": "2021-01-31",
  "amount": 76167.75,
  "currency": ""
}
```

Simpler in React

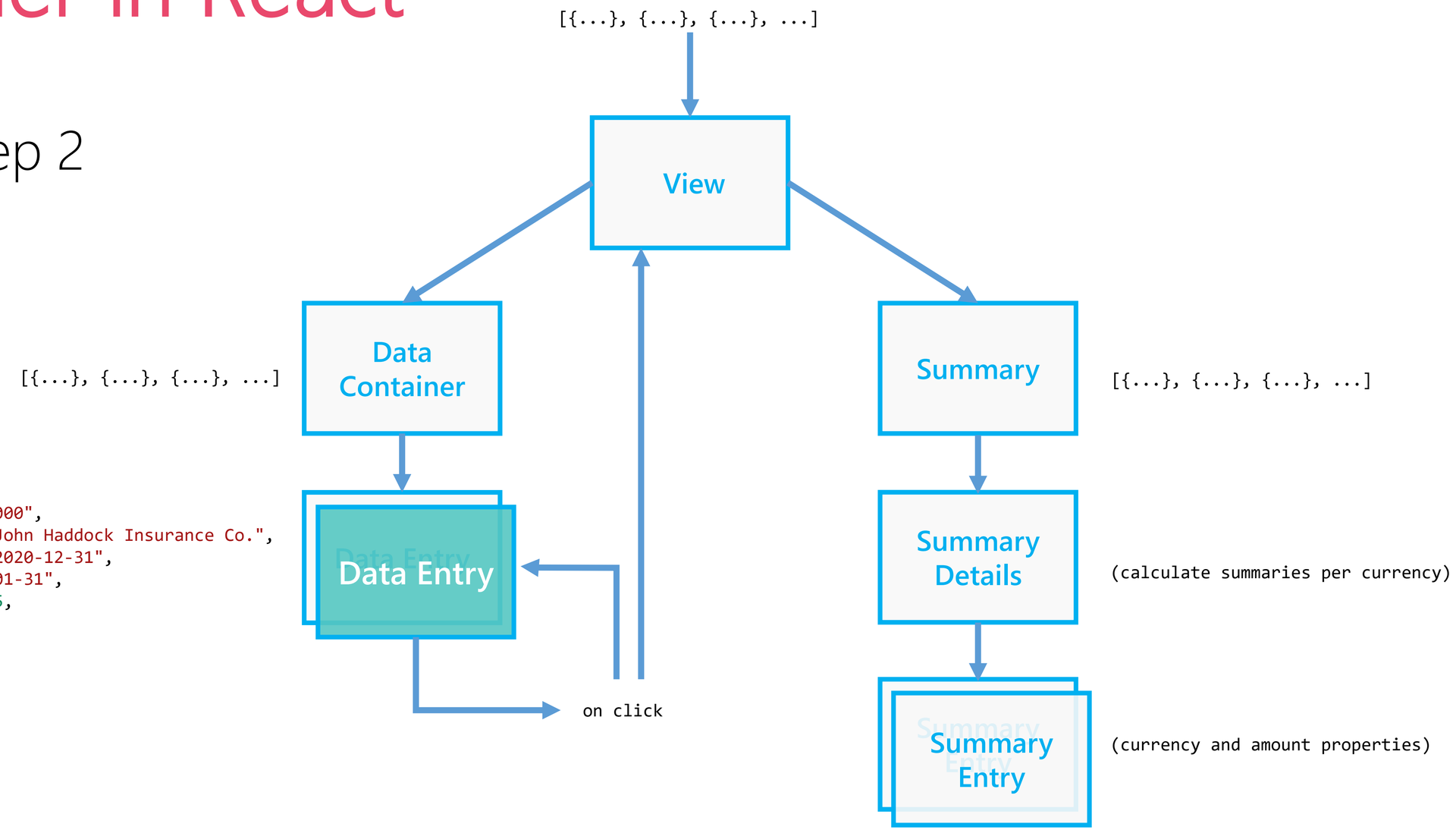
State, step 1




```
{
  "entryNo": 2433,
  "customerNo": "30000",
  "customerName": "John Haddock Insurance Co.",
  "documentDate": "2020-12-31",
  "dueDate": "2021-01-31",
  "amount": 76167.75,
  "currency": ""
}
```

Simpler in React

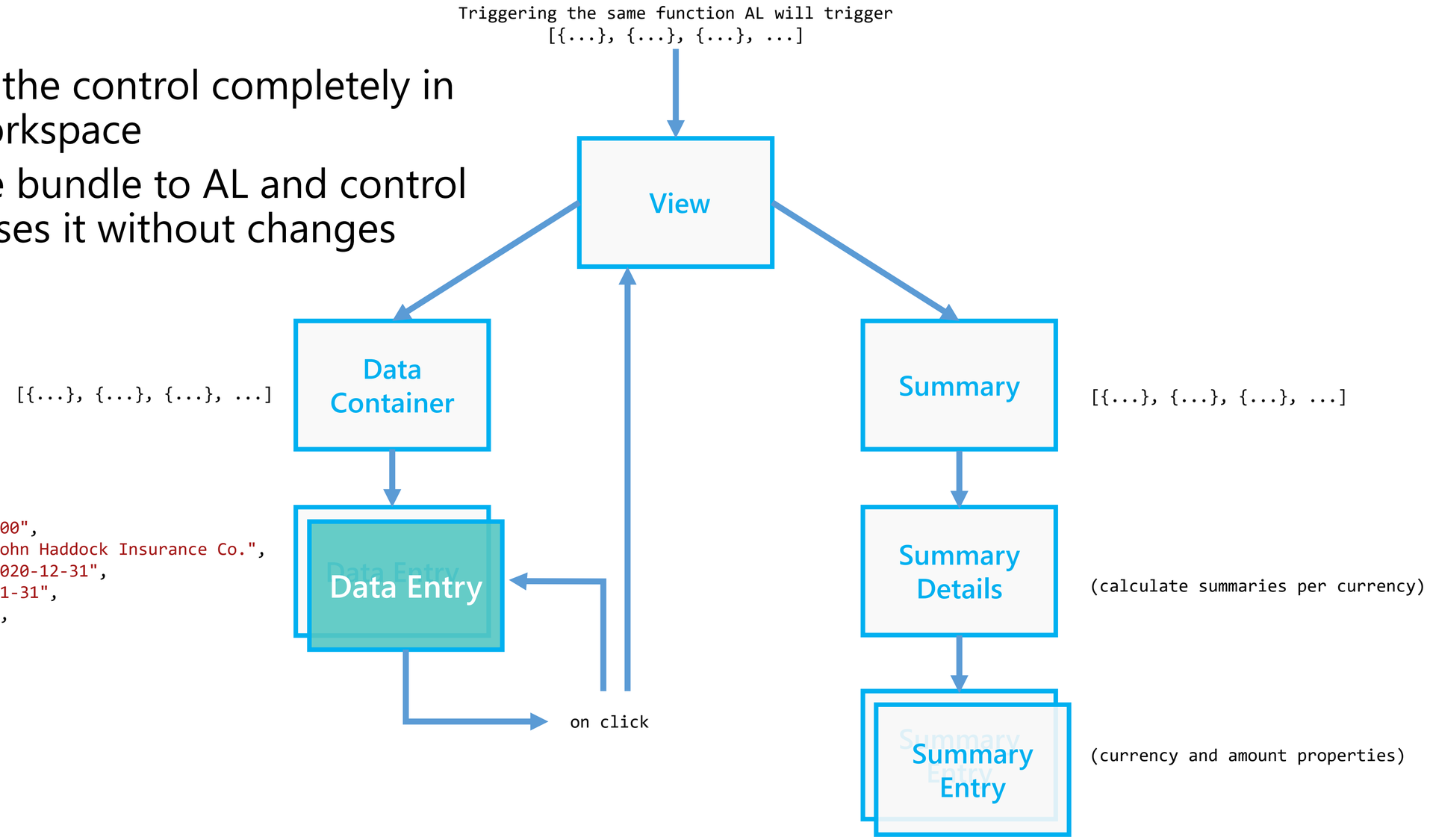
State, step 2




```
{
  "entryNo": 2433,
  "customerNo": "30000",
  "customerName": "John Haddock Insurance Co.",
  "documentDate": "2020-12-31",
  "dueDate": "2021-01-31",
  "amount": 76167.75,
  "currency": ""
}
```

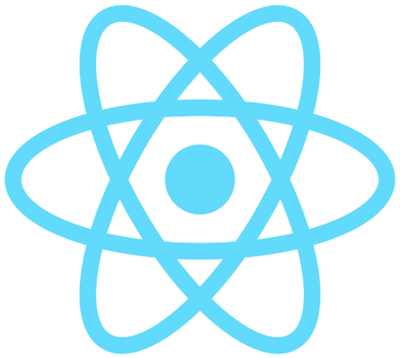
Simulating BC

- Develop the control completely in React workspace
- Copy the bundle to AL and control add-in uses it without changes



NAV TECH DAYS 2019
mibuso.com

React + Business Central



Architecture: what is React?

NAV TECH DAYS 2019
mibuso.com

MVC

MVP

MVVM

MVW

Architecture: what is React?

NAV TECH DAYS 2019
mibuso.com

None
of these

React Architecture

It all starts with state

Lessons learned so far

1. `this.state` object contains component state
2. You cannot manipulate state directly
3. … or rather: you **should not** manipulate state directly
4. You manipulate state through `this.setState` method
5. React refreshes the UI automatically on state change
6. It’s good to manipulate state through dedicated methods

What do we have so far?

NAV TECH DAYS 2019
mibuso.com

- All-in-one component:
  - Model = State
  - View = JSX
  - “controller” = onClick logic
- Can we do better?
- Let’s separate concerns a bit more:
  - Moving state out of component
  - Moving “controller” out of component

What is Redux

NAV TECH DAYS 2019
mibuso.com

https://redux.js.org/

- Library for managing application state
- Extremely popular and wide-spread
  (12.094 redux-related packages on npm)

The only correct way to manage state in React
… but it is not limited to React

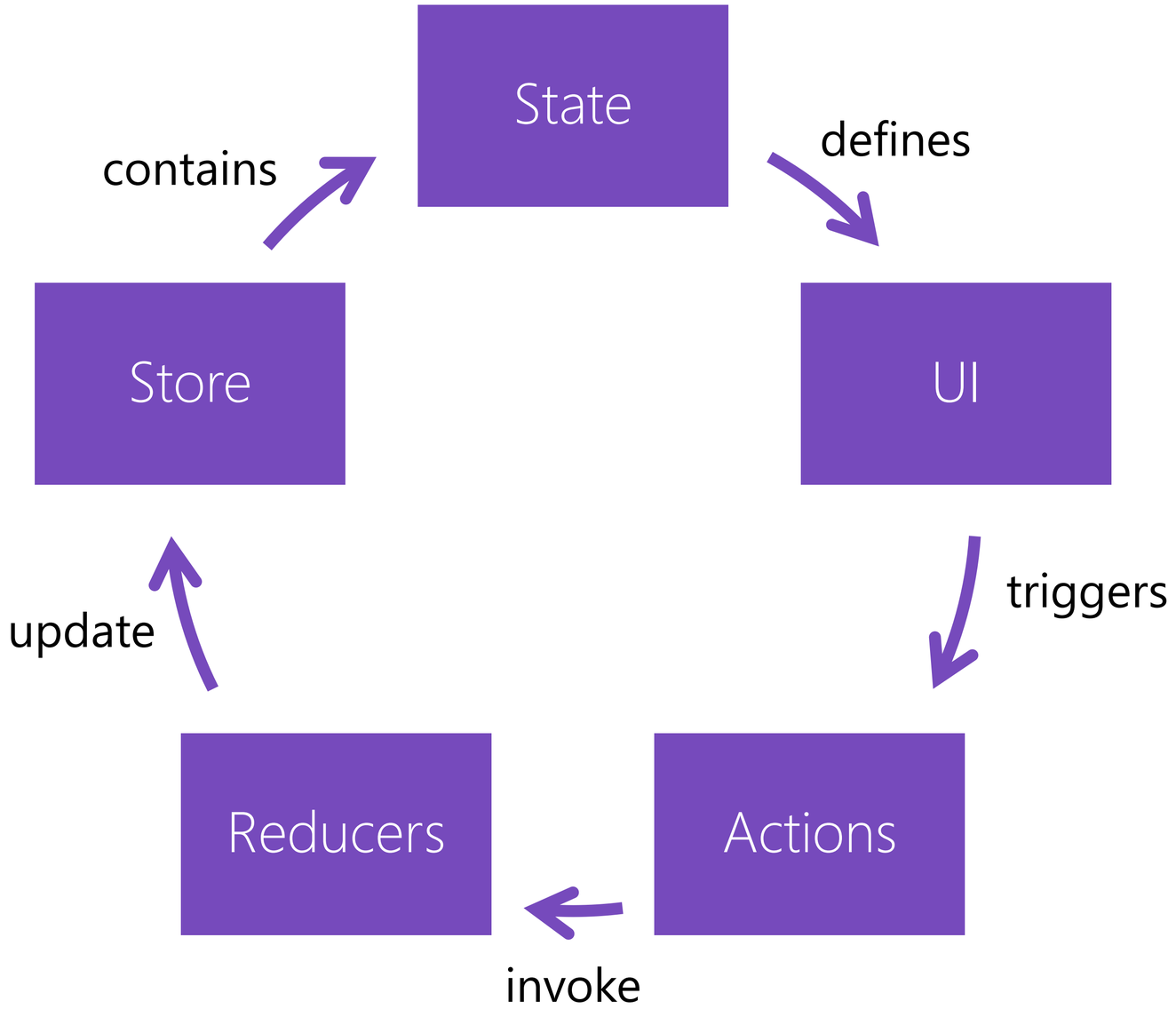
Redux principles

- State is centralized
- State is immutable (directly)
- State changes are only possible through *reducer* functions
- Data flow is strictly unidirectional

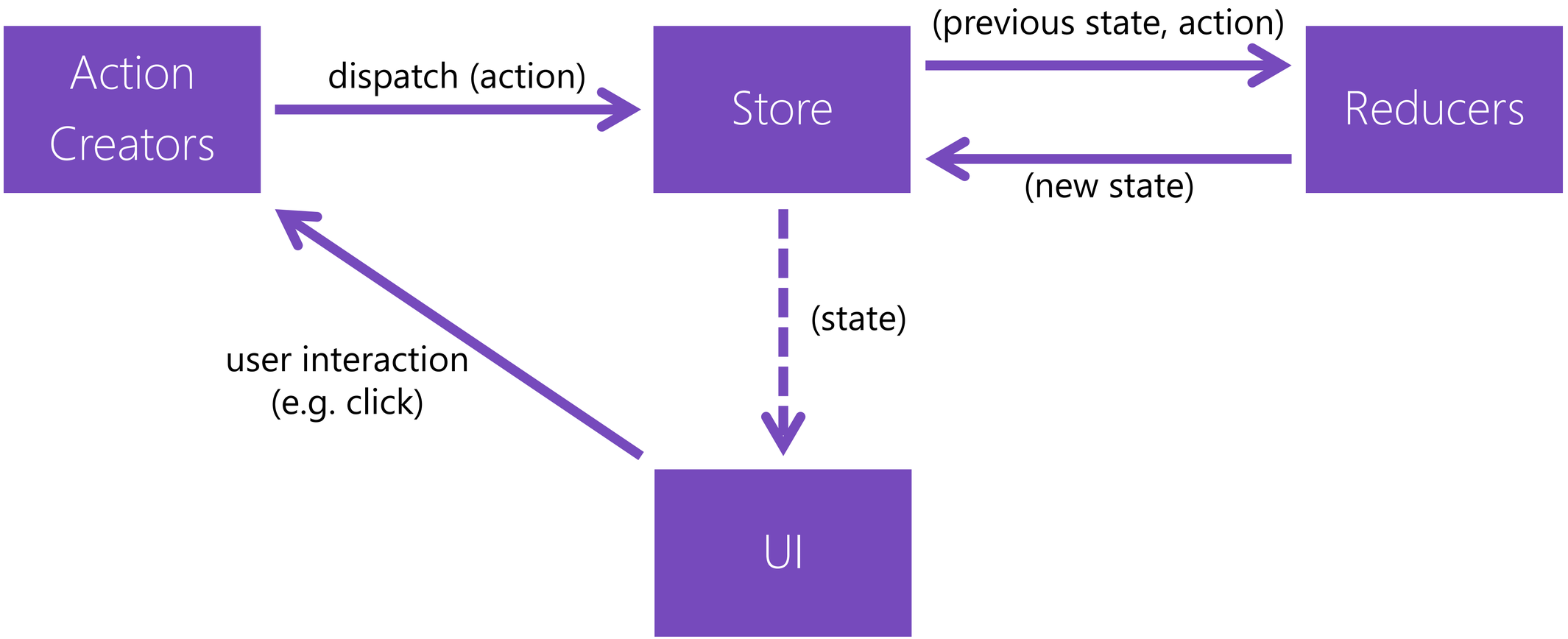
Redux terminology

- State = Application data
- Store = state container
- Reducers = functions that manipulate state
- Actions = functions that indicate intent to change state

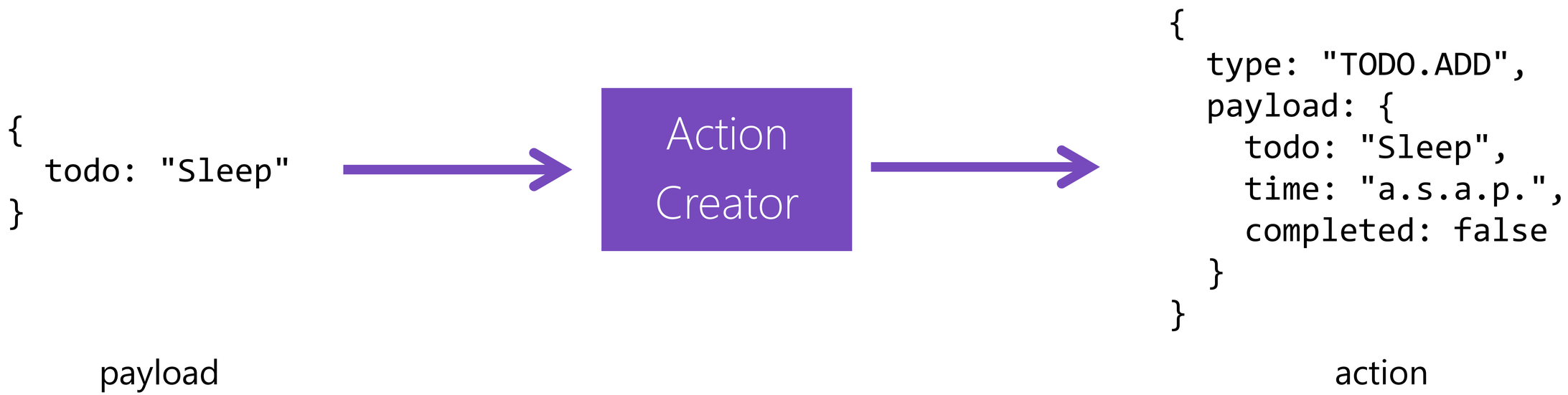
Redux unidirectionality



Actual Redux flow



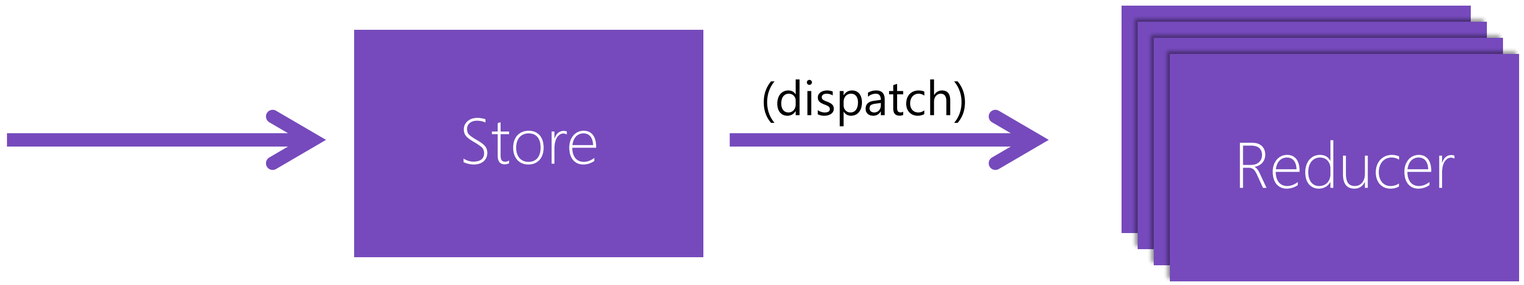
Inside an action



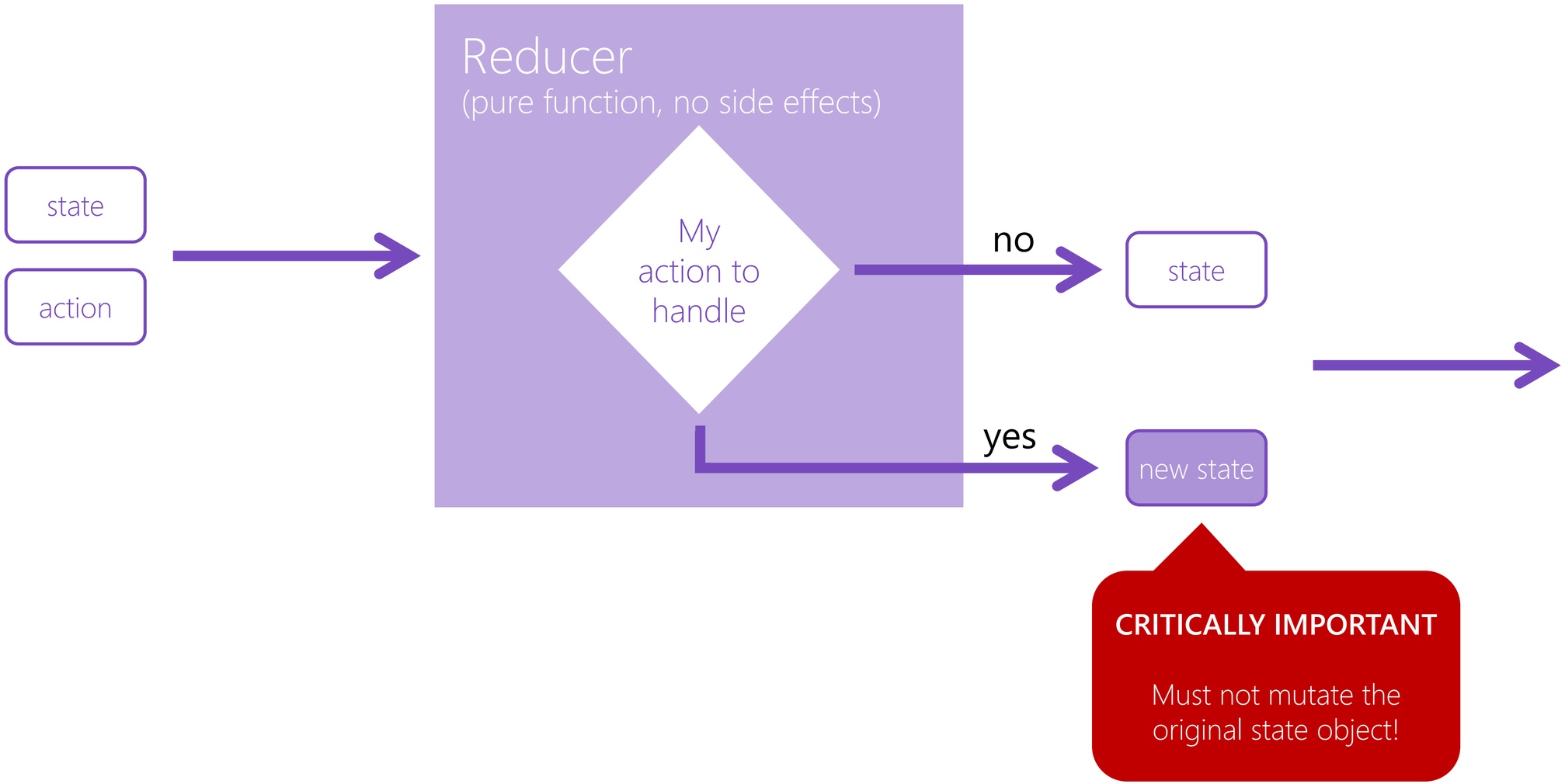
Inside an action

```
{
  type: "TODO.ADD",
  payload: {
    todo: "Sleep",
    time: "a.s.a.p.",
    completed: false
  }
}
```

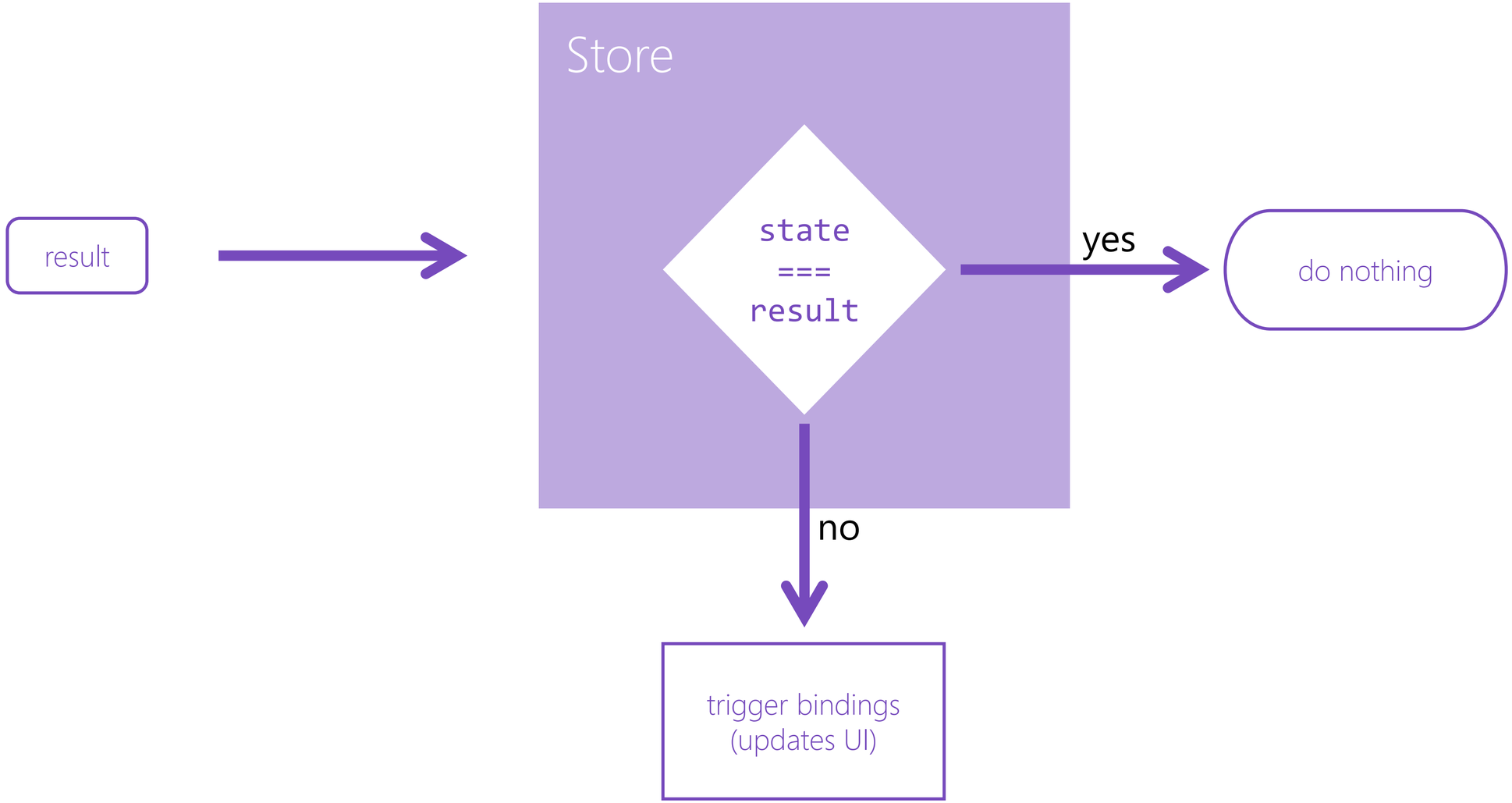
action



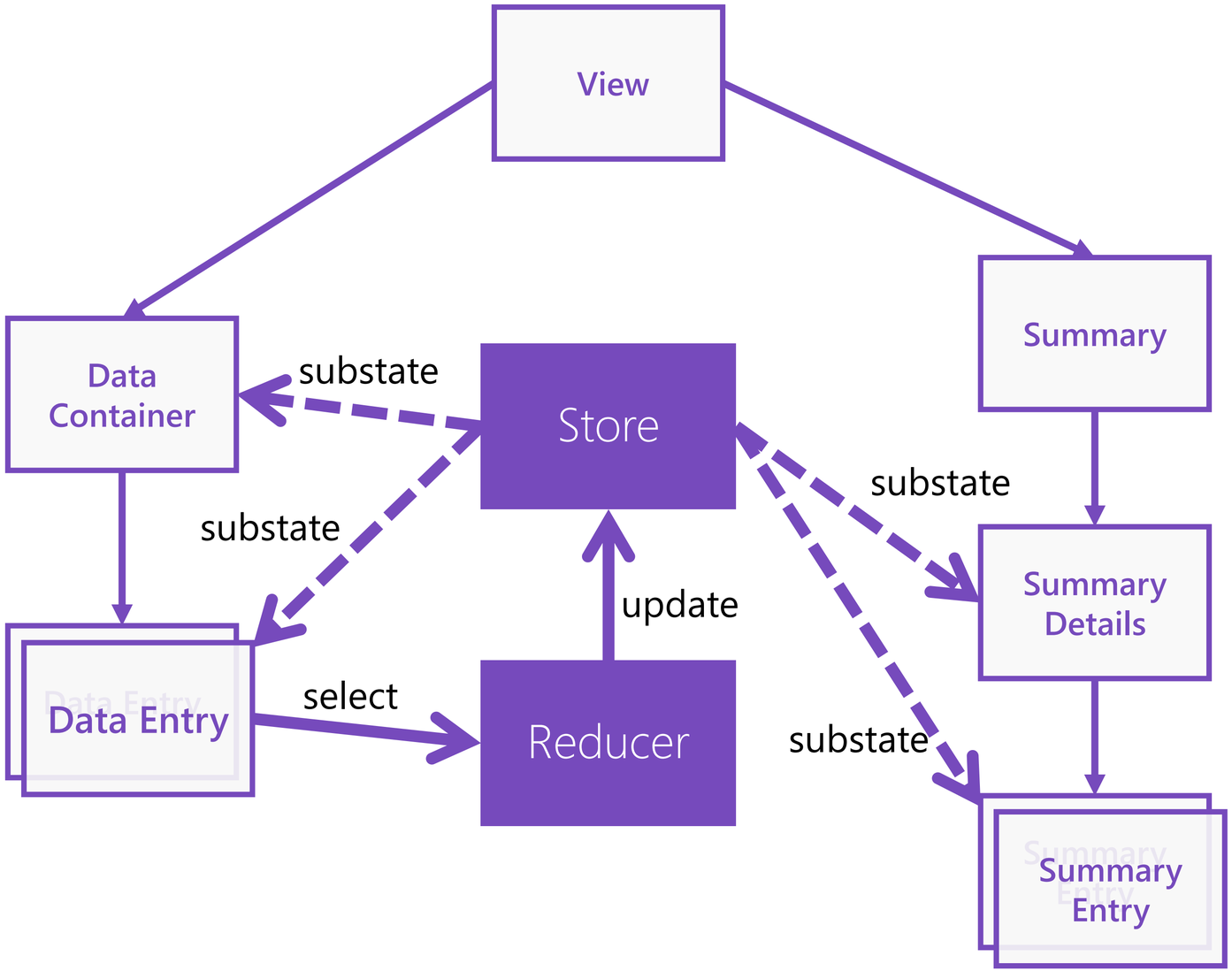
Inside a reducer



Inside the store



Our demo with Redux



A final touch

Mocking
Business Central
web client

Tomorrow 11am

NAV TECH DAYS 2019
mibuso.com

{Connect app}²

LEVEL: 400 - EXPERT | SESSION | ROOM 8

What if you have to create an iOS or Android app, and safely connect it to NAV, with your own tables, in the cloud or OnPrem. We will show you the ins and outs to do just that, and even enable it with the prime of technology like AI and Cognitive Services to enable your app with what is needed to be ready for the future.

You know connect apps? When you hear Microsoft talk about those, they are always payroll, expense management, yadda-yadda, boring stuff. Bear with us – and forget about those! Connect apps can be fun, and we are about to convince you of that. And that's the first ten minutes of our session. The remaining eighty will be sheer fun, for us two, at least.

In this session we are taking you on a rollercoaster of technology integrations, including React Native, Azure functions, machine learning, cognitive services, and yes, there will be some Business Central, too. Our goal is to show you how to build a full-blown sexy (oh yeah!) mobile connect app that uses Business Central APIs in, well, let's settle for "unconventional" ways, shall we? We are pretty sure that what we build in front of you won't get past Microsoft's App Source marketing validation, but from the technical angle, it'll blow your socks off.

This session will be heavy on code, demos, and technology. Just keep count of how many languages we'll use to glue all that stuff together (Dutch and Croatian excluded).

Wauters Eric (BE)
waldo.be

Babic Vjekoslav (HR)
Vjeko.com

NAV TECH DAYS 2019
mibuso.com

Q&A

Any Questions?

NAV TECH DAYS 2019
mibuso.com

Thank You!