
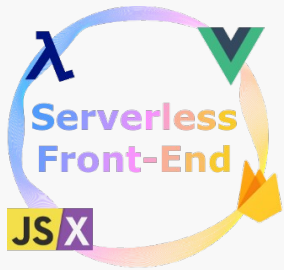


Vue with JSX

(but not really )

Cristóvão Trevisan



Componentization using Vue with JSX

(but not really )

Cristóvão Trevisan

About me (About this presentation)

- Electronic Engineering - UTFPR
- Computer Engineering Exchange (1 year) - Syracuse University (NY)
- GitHub: [cristovao-trevisan](#)
- Ponto Ativo (~1 year) - Cordova -> React -> React Native
- Contabilizei (~7 months) - Vue

This presentation **objective** is to convince you why **Component Based Architecture** is a good idea, and then show the general idea of how to use it.

I'll use **Vue with JSX** as a **tool**, that's it.

About me (About this presentation)

- Electronic Engineering
- Computer Engineering
- GitHub: [cristovao](#)
- Ponto Ativo (~1 year)
- Contabilizei (~7 years)



This presentation is about
Component Based

show the general idea of how to use it.

I'll use **Vue with JSX** as a **tool**, that's it.

University (NY)

e

u why

idea, and then

Introduction

- Front-end history brief
- DOM and the Virtual DOM
- CSS at scale
- Components Architecture
- Modern Flux-Component Architecture
- Componentization
- Vue with JSX

Frontend is a crazy thing...

Javascript

1995: Netscape and DOM

1997: ECMAScript

1999: AJAX

2006: jQuery

2013: React

2014: Vue

2015: React Native

Complete change
of paradigm

Tooling

2010: 

2011:  (deprecated)

2012: Webpack 

2015: 



*“The characteristic we found the most consistent among those we interviewed was that a Senior Developer should have X years of experience minimum. “X” could really be any number, but most developers and hiring managers we spoke to agreed that **10** was a good number.”*

What Defines a “Senior Developer?” - Stack Overflow

Frontend is a crazy thing...

Javascript

1995: Netscape a

1997: ECMAScrip

1999: AJAX

2006: jQuery

2013: React ←

2014: Vue

2015: React Nativ



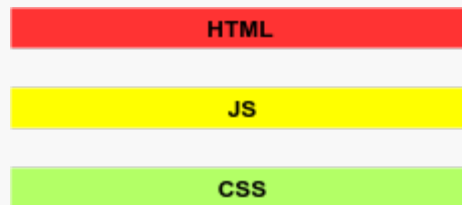
(ated)

interviewed was
m. "X" could
spoke to

Frontend Evolution

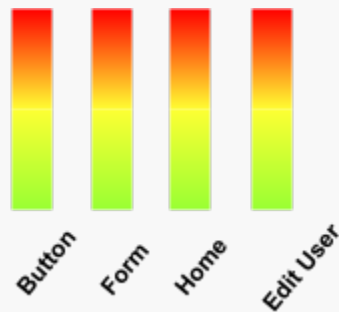
Separation by Technology:

1. Server Rendered Page
2. Dynamic Content (JS)
3. MVC



Separation by Concerns:

- React
- Vue
- Angular (MV*)



HTML at Scale

- No strict contract

```
<div class="could-do-anything">  
  I might not even show.  
</div>
```

- Insufficient Abstractions

```
<input type="wtf" id="I-come-with-pre-defined-css" />
```

- DOM

DOM vs Virtual-DOM

DOM is like CISC:

- Complicated
- Large
- Low Level
- Hard to make performant
- Powerfull

You need to update it for every data change

Virtual-DOM is like RISC:

- Simple
- Small
- High Level
- Easy to make performant
- Easy to Understand

You tell how your visual behaves based on the data and let the virtual-dom do the rest

DOM vs Virtual-DOM

DOM is like

- Complex
- Large
- Low Level
- Hard to
- Powerful

You need to
change



ant

has
the virtual-

CSS at Scale

| | CSS | Sass / SCSS | Vue | Vue JSX |
|--------------------------|-----|-------------|-------|---------|
| Local Variables | X | ✓ | ✓ | ✓ |
| Sharing of constants | X | X | X | ✓ |
| Deterministic Resolution | X | X | X | ✓ |
| Dead code elimination | X | X | - | + |
| Code minification | X | X | X | ✓ |
| Isolation | X | X | scope | module |

CSS Tooling

Vue Template

- post-css (build time) - fast and light

Vue JSX

- **Inline-style-prefixer** (run time) - “slow” and “heavy” (but now much)
- Postcss-js (build time) - “fast” and “light”

What are Components (or NOT...)?

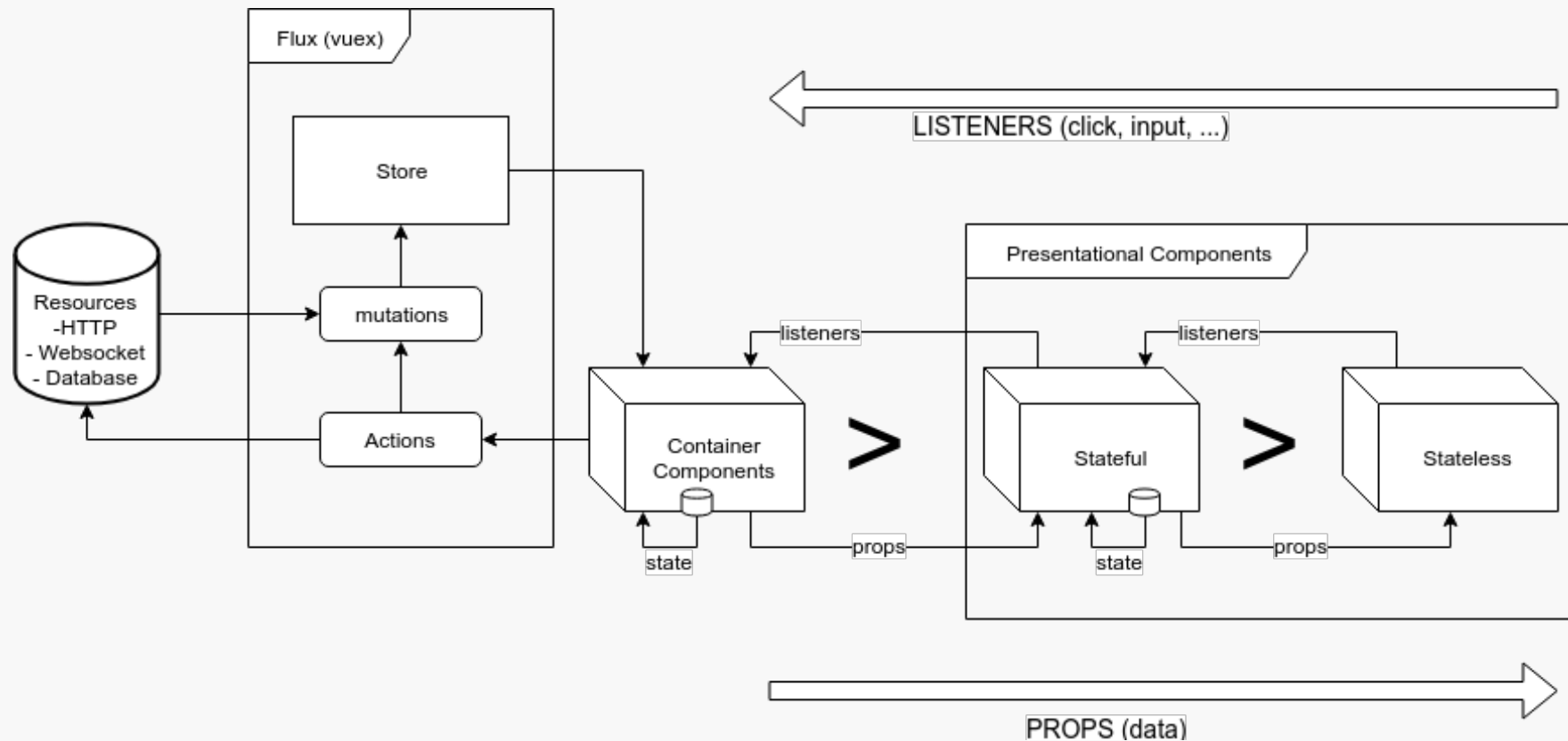
Components are NOT:

- The WHOLE page
- Auto-sufficient
- With side effects

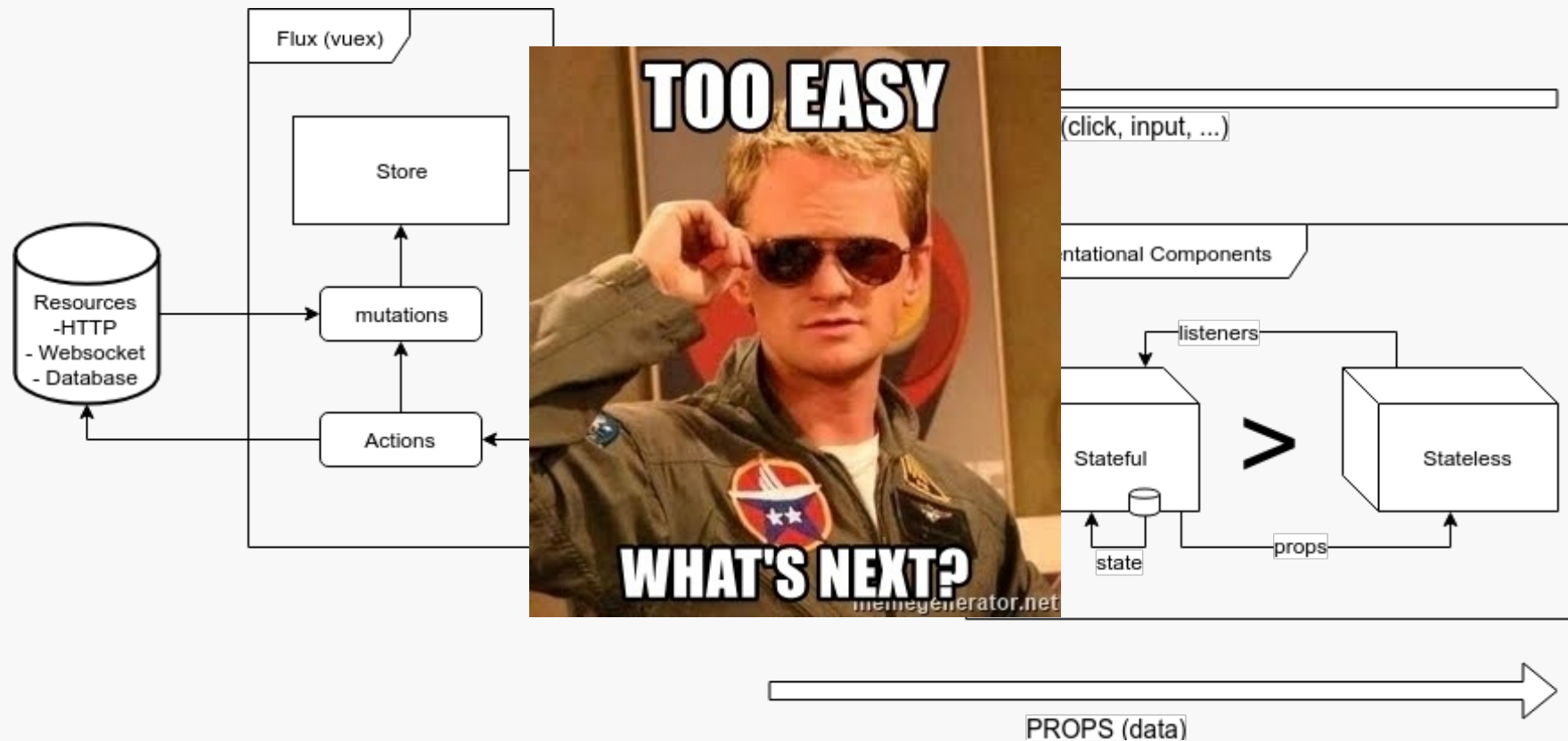
Components ARE:

- Scoped
- Small
- Expose an predictable API
- Avoid repeated code
- Expressive (Quick to read)

Components with Flux Architecture (in Vue)

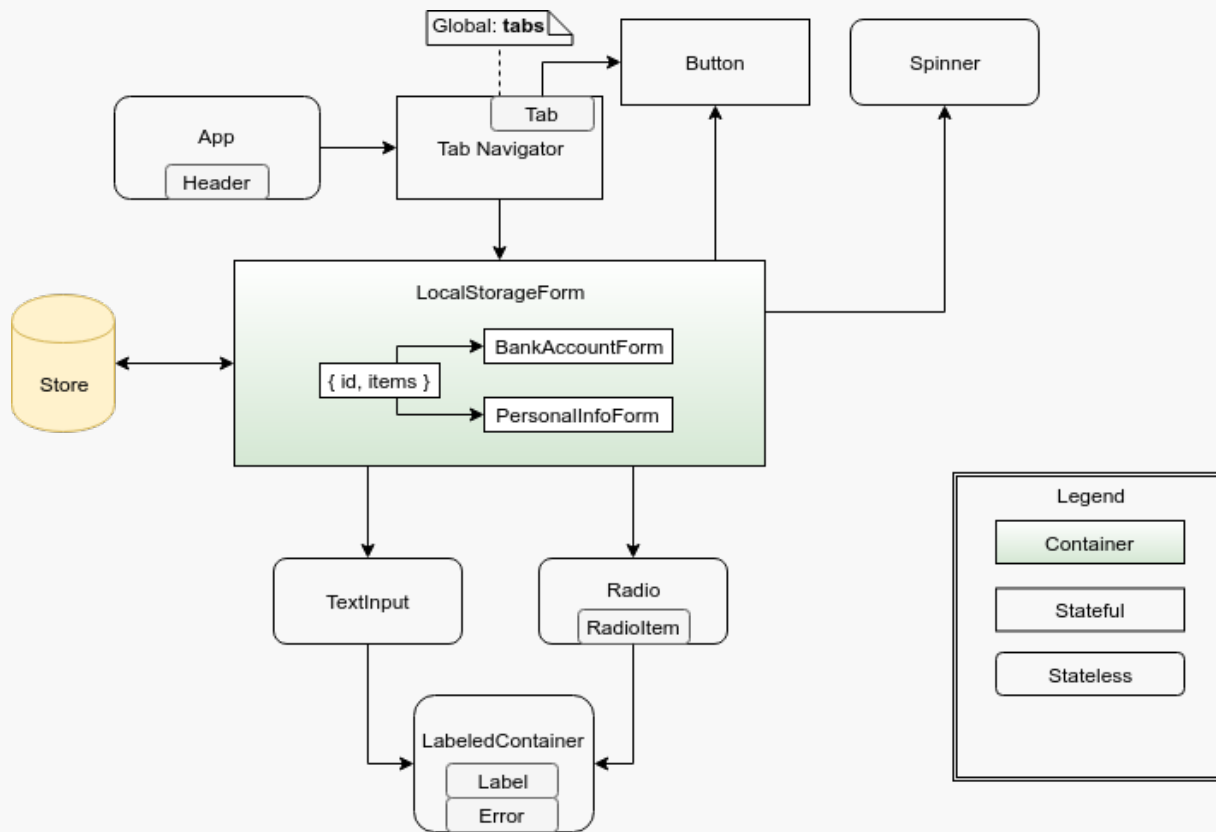
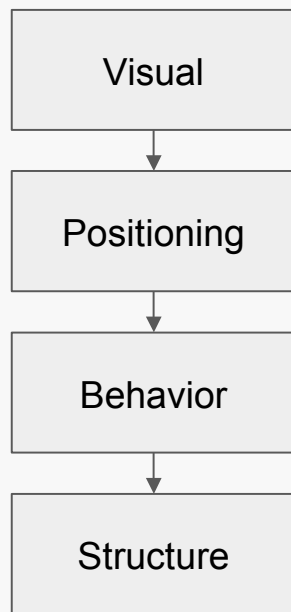


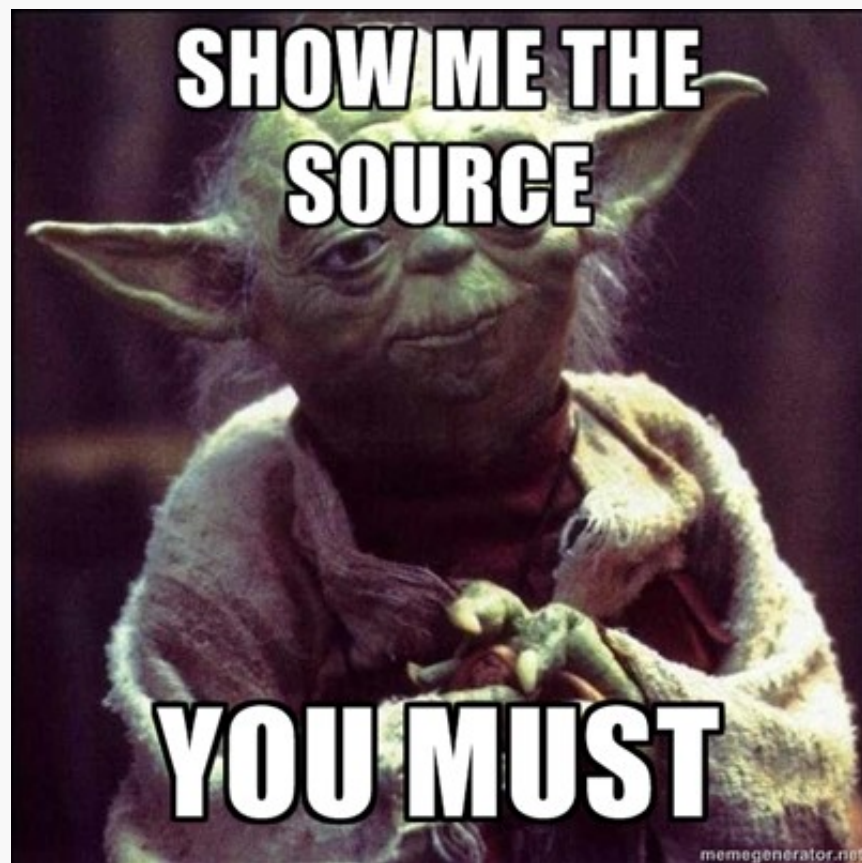
Components with Flux Architecture (in Vue)



Componentization: Form App

The Formula





References

- Stack overflow research - <https://www.stackoverflowbusiness.com/blog/what-defines-a-senior-developer>
- Twitter talk about React Native - <https://youtu.be/tFFn39ILO-U>

Repository: <https://github.com/cristovao-trevisan/vue-with-jsx>