

Building sustainable
enterprise apps
with Vue

PaleoPuzzle



Overview

Overview

I. Defining enterprise

Overview

1. Defining enterprise
2. Is Vue enterprise-ready?

Overview

1. Defining enterprise
2. Is Vue enterprise-ready?
3. Starting out

Overview

1. Defining enterprise
2. Is Vue enterprise-ready?
3. Starting out
4. Maintaining

Overview

1. Defining enterprise
2. Is Vue enterprise-ready?
3. Starting out
4. Maintaining
5. Performance

Overview

1. Defining enterprise
2. Is Vue enterprise-ready?
3. Starting out
4. Maintaining
5. Performance
6. The future

Defining enterprise

Current Time

9 : 00 am



What's enterprise?

What's enterprise?



code(bases)

What's enterprise?



code(bases)



teams

What's enterprise?



code(bases)



teams



traffic

What's enterprise?



code(bases)



teams



traffic



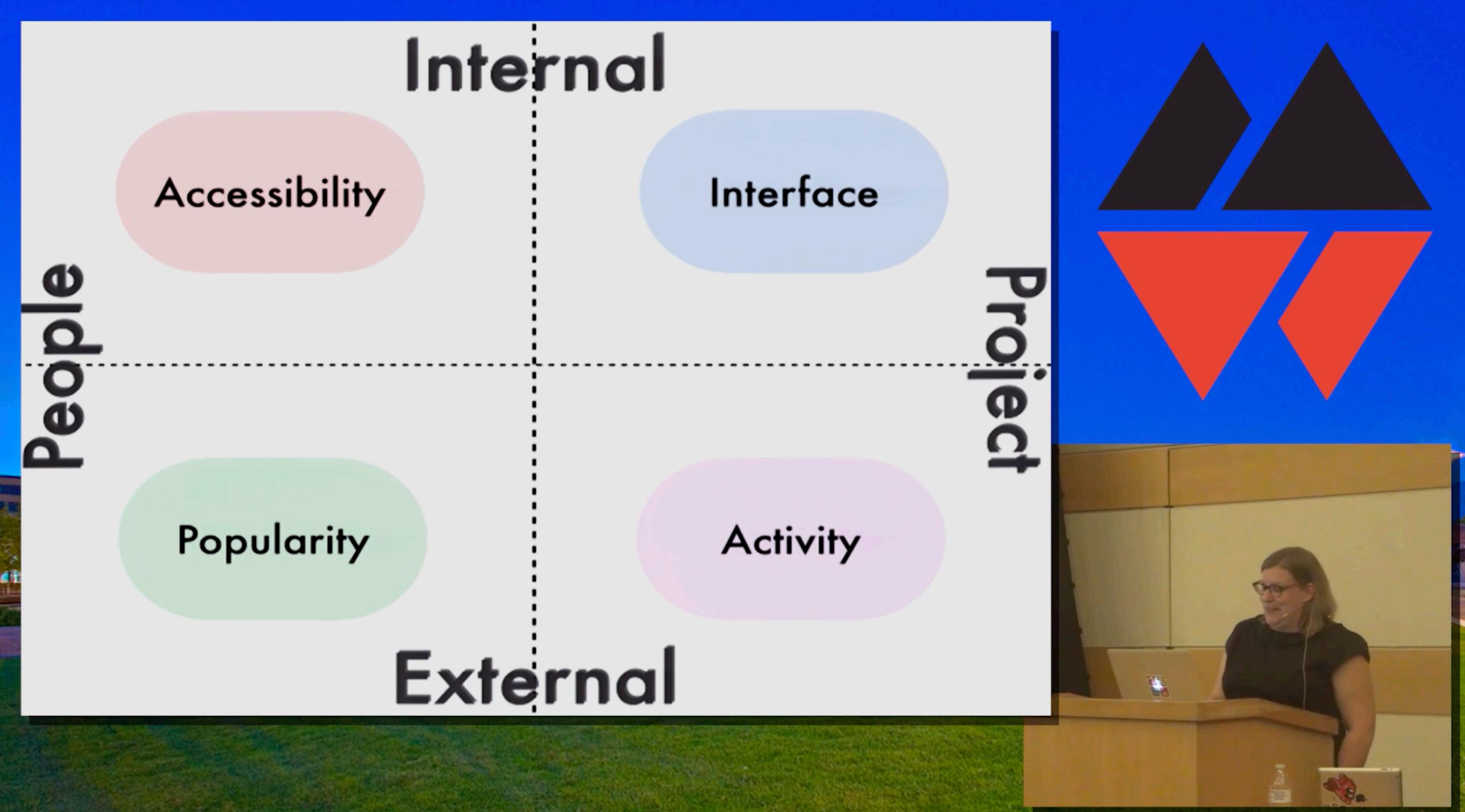
lifespan



What's *legacy* code?

Is Vue
enterprise-ready?

The Mei System



Vue is built for
developer happiness

Vue helps you...

Vue helps you...

I. Hire

Vue helps you...

1. Hire

2. Support

Vue helps you...

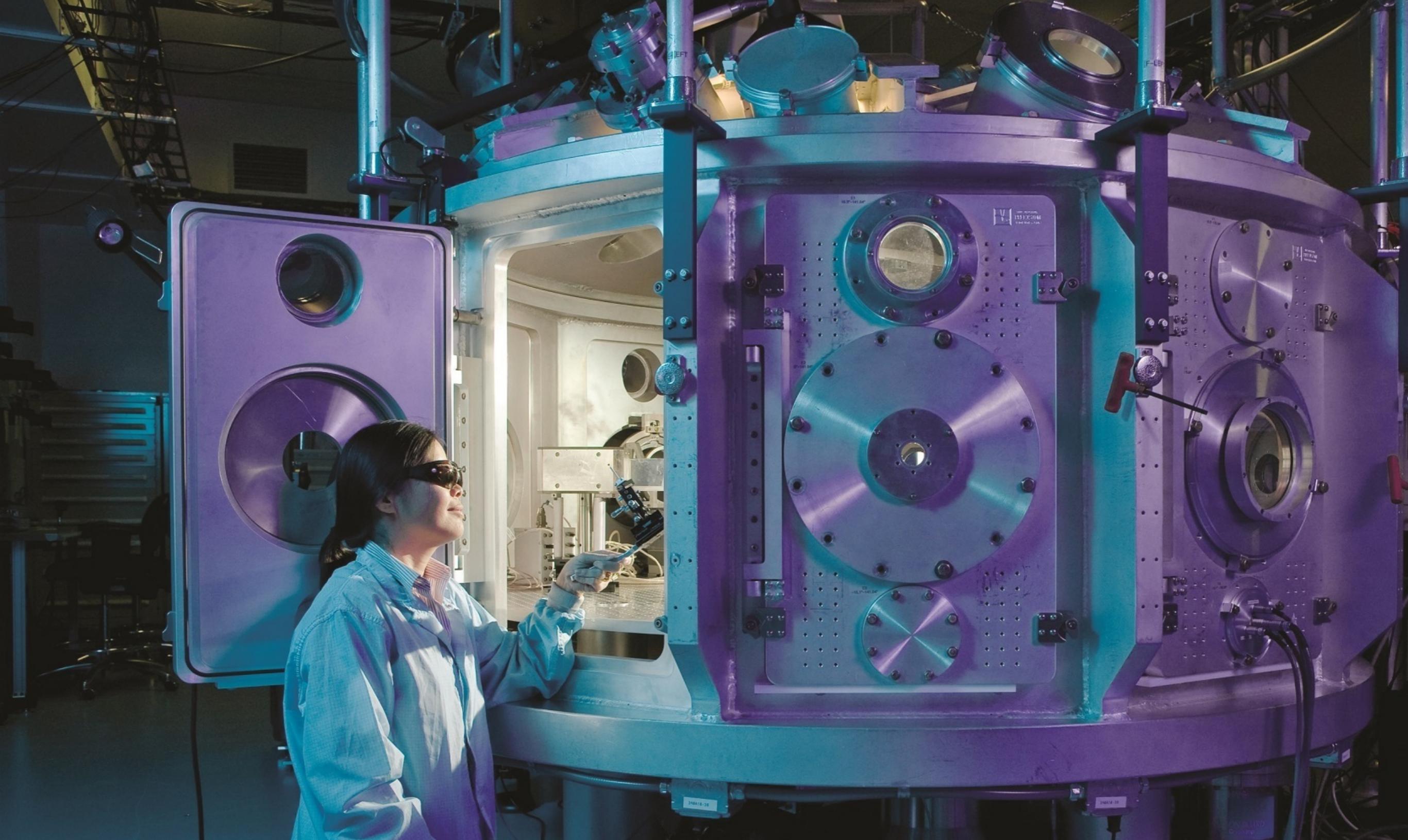
1. Hire
2. Support
3. Retain

Vue helps you
migrate gradually

Starting an enterprise app

Mindset

*Don't know
anything*



Build working hypotheses

Choose your base

Script

```
<script src="/vue.js"></script>
```

Vue CLI

```
vue create my-app
```

Nuxt

```
npx create-nuxt-app my-app
```

Gridsome

```
gridsome create my-app
```

Vue Enterprise Boilerplate

Maintaining an enterprise app



Code is like
comedy

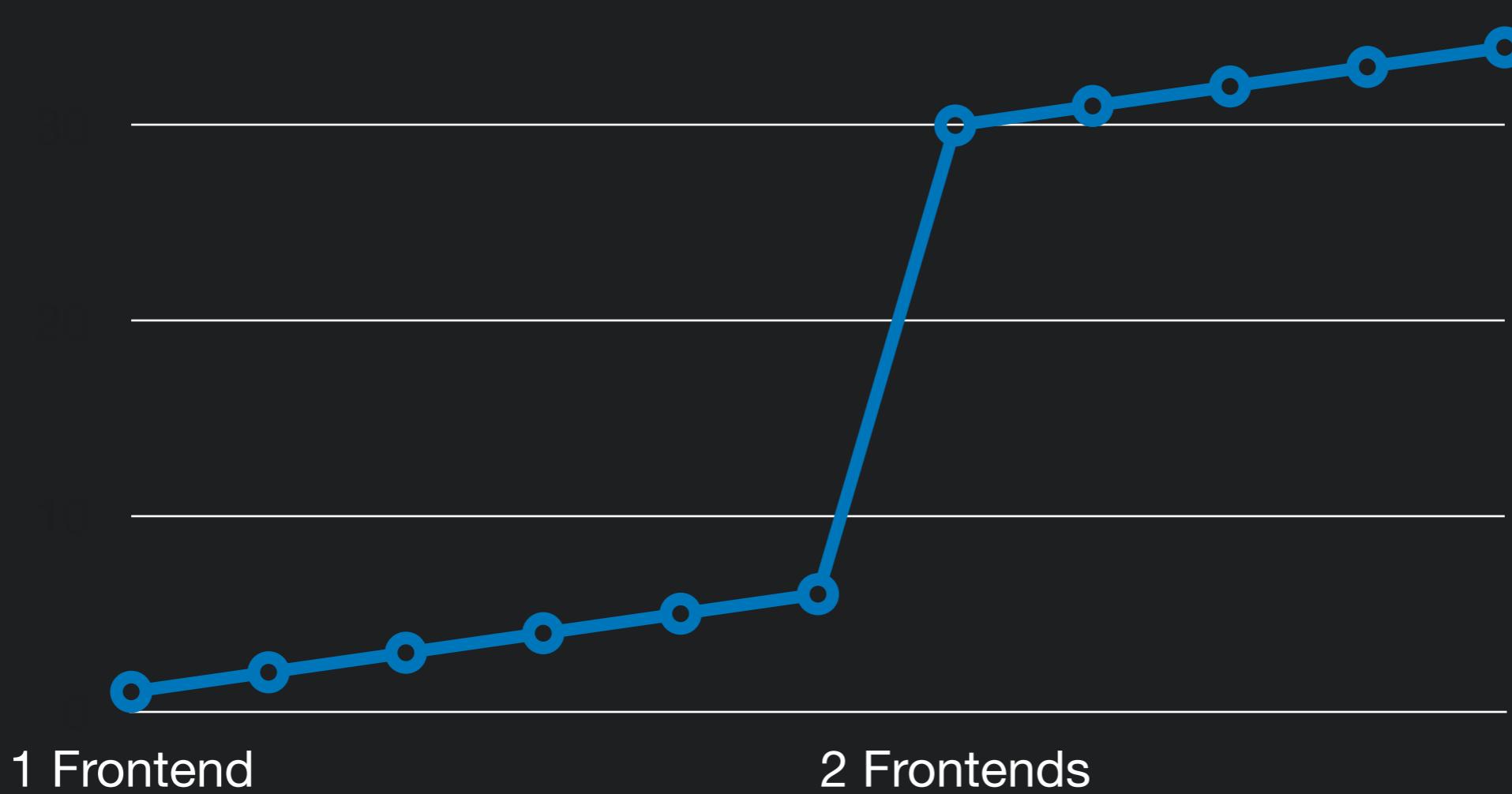
Code is
like stuff





Code is like garbage

Complexity leaps



Monorepos

Defining a monorepo

Defining a monorepo

A single codebase, with independently
deployable/publishable projects that
share some common resources

Monorepo != Monolith

Monorepo != Monolith







Monorepos make
sharing easier

Monorepos make
sharing easier

...while keeping
you productive

Monorepo tooling

- Yarn workspaces
- Rush
- Lerna
- Bolt

Component
organization

Nested

```
✓ components
  ✓ admin
    ✓ registration
      ▼ form.vue
  ✓ distributor
    ✓ registration
      ▼ form.vue
  ✓ manufacturer
    ✓ registration
      ▼ form.vue
  ✓ shared
    ✓ email
      ▼ input.vue
    ✓ password
      ▼ input.vue
    ✓ registration
      ✓ submission
        ✓ messages
          ▼ error.vue
          ▼ success.vue
          ▼ button.vue
          ▼ loader.vue
```

Flat

- ✓ components
 - ▼ input-email.vue
 - ▼ input-password.vue
 - ▼ registration-button.vue
 - ▼ registration-form-admin.vue
 - ▼ registration-form-distributor.vue
 - ▼ registration-form-manufacturer.vue
 - ▼ registration-loader.vue
 - ▼ registration-message-error.vue
 - ▼ registration-message-success.vue

Wrap vendor components

```
<script>
import { FontAwesomeIcon } from '@fortawesome/vue-fontawesome'
import fontAwesome from '@fortawesome/fontawesome-svg-core'

fontAwesome.library.add(
  require('@fortawesome/free-solid-svg-icons/faSync').definition,
  require('@fortawesome/free-solid-svg-icons/faUser').definition
)

export default {
  components: { FontAwesomeIcon },
  props: ['source', 'name']
}
</script>

<template>
  <FontAwesomeIcon v-if="source === 'font-awesome'" :icon="name" />
  <span v-else-if="source === 'custom'" :class="'icon-custom-' + name" />
</template>
```

Wrap vendor components

```
export default {  
  components: { FontAwesomeIcon },  
  props: {  
    source: {  
      type: String,  
      default: 'font-awesome',  
      validator(value) {  
        return ['font-awesome', 'custom'].indexOf(value) !== -1  
      }  
    },  
    name: {  
      type: String,  
      required: true,  
      validator(value) {  
        return ['sync', 'user'].indexOf(value) !== -1  
      }  
    }  
  }  
}
```

Making it easy to follow
best practices

Lint & format
everything

Lint & format everything

- github.com/vuejs/eslint-plugin-vue

Lint & format everything

- github.com/vuejs/eslint-plugin-vue
- vuejs.org/v2/style-guide

Custom generators & snippets

```
vue-enterprise-boilerplate master *
> yarn new component
yarn run v1.17.3
$ cross-env HYGEN_TMPLS=generators hygen new component
? Name: > whatevs
```

➤ Component names should contain at least two words to avoid conflicts with existing and future HTML elements.

Custom generators & snippets

```
vue-enterprise-boilerplate master *  
› yarn new component  
yarn run v1.17.3  
$ cross-env HYGEN_TMPLS=generators hygen new component  
✓ Name: · todo-list  
✓ Blocks: · script, template, style
```

```
Loaded templates: generators  
    added: src/components/todo-list.vue  
    added: src/components/todo-list.unit.js  
✨ Done in 2.46s.
```

Internal warnings

```
methods: {
  // Perform more complex prop validations than is possible
  // inside individual validator functions for each prop.
  validateProps() {
    if (process.env.NODE_ENV === 'production') return

    if (this.href) {
      // Check for insecure URL in href.
      if (!this.allowInsecure && !/^https|mailto|tel):/.test(this.href)) {
        return console.warn(
          `Insecure <BaseLink> href: ${this.href
            }.\nWhen linking to external sites, always prefer https URLs.`
        )
      }
    }
  }
}
```

Internal warnings

```
methods: {
  // Perform more complex prop validations than is possible
  // inside individual validator functions for each prop.
  validateProps() {
    if (process.env.NODE_ENV === 'production') return

    if (this.href) {
      // Check for insecure URL in href.
      if (!this.allowInsecure && !/^https|mailto|tel):/.test(this.href)) {
        return console.warn(
          `Insecure <BaseLink> href: ${this.href
            }.\nWhen linking to external sites, always prefer https URLs.`
        )
      }
    }
  }
}
```

Internal warnings

```
methods: {
  // Perform more complex prop validations than is possible
  // inside individual validator functions for each prop.
  validateProps() {
    if ('production' === 'production') return

    if (this.href) {
      // Check for insecure URL in href.
      if (!this.allowInsecure && !/^https|mailto|tel):/.test(this.href)) {
        return console.warn(
          `Insecure <BaseLink> href: ${this.href
            }.\nWhen linking to external sites, always prefer https URLs.`
        )
      }
    }
  }
}
```

Internal warnings

```
methods: {  
  // Perform more complex prop validations than is possible  
  // inside individual validator functions for each prop.  
  validateProps() {  
    if (true) return  
  
    if (this.href) {  
      // Check for insecure URL in href.  
      if (!this.allowInsecure && !/^https|mailto|tel):/.test(this.href)) {  
        return console.warn(  
          `Insecure <BaseLink> href: ${  
            this.href  
          }.\nWhen linking to external sites, always prefer https URLs.`  
      )  
    }  
  }  
}  
}
```

Internal warnings

```
methods: {  
  validateProps() {  
    if (true) return  
  }  
}
```

Tests



Two essential e2e tests

Two essential e2e tests

- Can users log in?

Two essential e2e tests

- Can users log in?
- Can they give us money?



cypress

Keeping an app
performant

Top 3 perf tips

Top 3 perf tips

unnecessary
code

Top 3 perf tips

unnecessary
code

```
{  
  path: '/',
  name: 'home',
  component: () =>
    import('@/views/home.vue'),
}
```

Top 3 perf tips

unnecessary
code

unnecessary
renders

```
{  
  path: '/',
  name: 'home',
  component: () =>
    import('@/views/home.vue'),  
}
```

Top 3 perf tips

unnecessary
code

unnecessary
renders

```
{  
  path: '/',
  name: 'home',
  component: () =>
    import('@/views/home.vue'),
}
```

[github.com/Akryum/
vue-virtual-scroller](https://github.com/Akryum/vue-virtual-scroller)

Top 3 perf tips

unnecessary
code

```
{  
  path: '/',
  name: 'home',
  component: () =>
    import('@/views/home.vue'),
}
```

unnecessary
renders

[github.com/Akryum/
vue-virtual-scroller](https://github.com/Akryum/vue-virtual-scroller)

unnecessary
re-renders

Top 3 perf tips

unnecessary
code

```
{  
  path: '/',
  name: 'home',
  component: () =>
    import('@/views/home.vue'),
}
```

unnecessary
renders

[github.com/Akryum/
vue-virtual-scroller](https://github.com/Akryum/vue-virtual-scroller)

unnecessary
re-renders

```
<keep-alive>  
  <component  
    :is="currentTab"  
    :key="currentTab.name"  
  />  
</keep-alive>
```

The future

Simpler and
more explicit

Composition API

Composition API

```
import { reactive } from 'vue'

export default function useSearch(getResults) {
  const state = reactive({
    query: '',
    results: [],
    run() {
      state.results = []
      return getResults(state.query).then(results => {
        state.results = results
      })
    }
  })
}

return state
}
```

Composition API

```
import useSearch from '@use/search'
```

Composition API

```
<script>
import useSearch from '@use/search'
import axios from 'axios'

export default {
  setup() {
    const productSearch = useSearch(query =>
      axios
        .get('/api/products', { params: { query } })
        .then(response => response.data.products)
    )

    return { productSearch }
  }
}
</script>
```

Composition API

```
<script>
import useSearch from '@use/search'
import axios from 'axios'

export default {
  setup() {
    const productSearch = useSearch(query =>
      axios
        .get('/api/products', { params: { query } })
        .then(response => response.data.products)
    )

    return { productSearch }
  }
}
</script>
```

Composition API

```
    return { productSearch }
}
</script>

<template>
  <input
    v-model="productSearch.query"
    @keydown.enter="productSearch.run"
  />
  <ProductList :products="productSearch.results" />
</template>
```

That's all!

Thanks to my Vuesionary sponsors!

Vuetify

Vue School

Vue Mastery

Ben Hong

patreon.com/chrisvuefritz

Thanks!

github.com/chrisvfritz/vue-enterprise-slides

@chrisvfritz