



VueJS



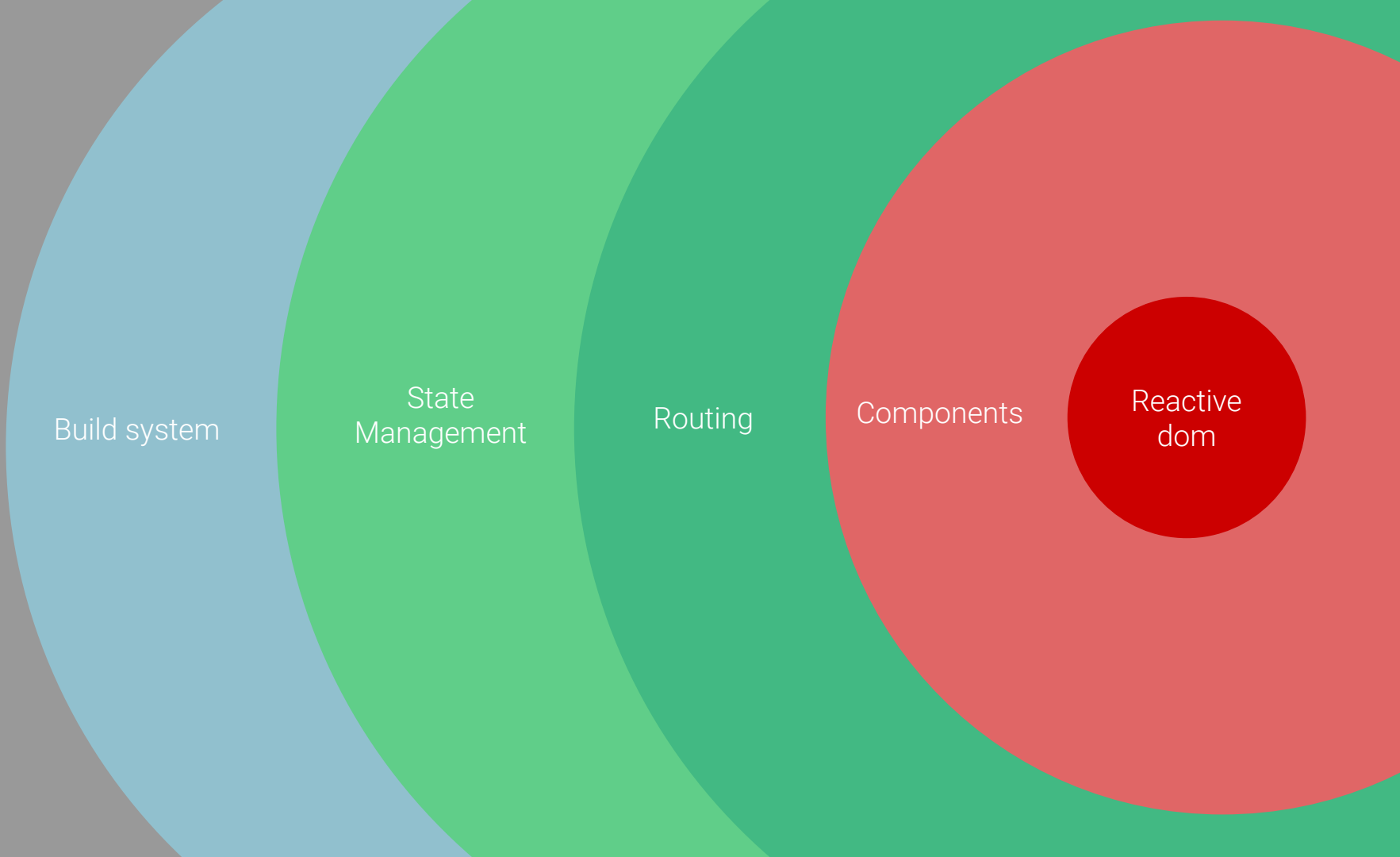
Lasse Fisker



IT MINDS
YOUNG MINDS - EXCELLENT SOLUTIONS

- IKT from IHA
- Intern at IT-Minds
- Part time -> Full time
- Frontender
- User Experience

What are we going to talk about?



Reaktiv dom

- Virtuel dom
- Free choice in template lang
- All the directives you expect

```
<template>
  <div>
    <h1>Hello!</h1>
  </div>
</template>

<script type="text/x-template" id="template">
  <div class="hello">
    <h1>Hello!</h1>
  </div>
</script>
```

```
<template lang="pug">
  .hello
    h1 Hello!
</template>

1 <template lang="pug">
2   include helloworld.pug
3 </template>
```

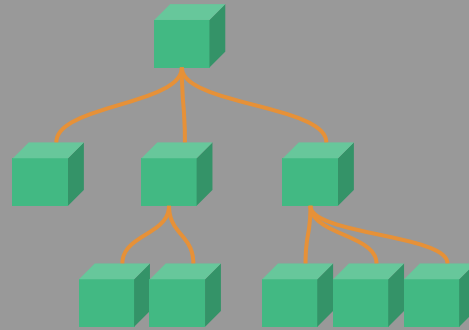
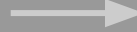
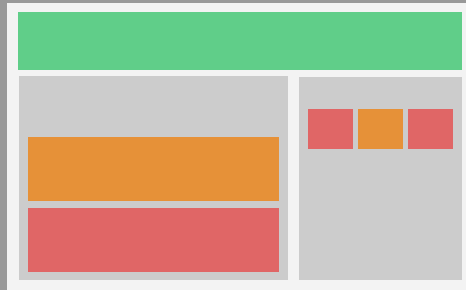
```
render: function (createElement) {
  return createElement()
},
```

Template syntax

```
<template>
  <div>
    <div class="full-syntax" v-bind:prop="val" v-on:event="handlerFunc"></div>
    <div class="shorthand" :prop="val" @event="handlerFunc"></div>

    <div class="loop" v-for="item in collection" :key="item">{{ item }}</div>
    <div class="control-statements" v-if="boolean"></div>
    <div class="dynamic-css-class" :class="'some-class': boolean"></div>
    <div class="dynamic-css-style" :style="{styleProp: styleValue, otherStyleProp: othervalue}"></div>
  </div>
</template>
```

Components



Components

```
1 <template lang="pug">
2
3 import Vue, { ComponentOptions } from 'vue'
4
5 // Declare the component's type
6 interface MyComponentOptions extends ComponentOptions {}
7
8 // The @Component decorator indicates the class is a Vue component
9
10 // All component options are allowed in here
11
12 export @Component({
13   // All component options are allowed in here
14   data: {
15     template: '<button @click="onClick">Click!</button>'
16   },
17   // Initial data can be declared as instance properties
18   message: string = 'Hello!'
19 }, {
20   // Component methods can be declared as instance methods
21   onClick (): void {
22     window.alert(this.message)
23   }
24 })
25
26 // We can use the MyComponent type
27 // with the MyComponent type
28 } as ComponentOptions<MyComponent>
```

- Single file components
- Script - template - styles
- One object
- Or as typescript
- With class decorator

Components

```
computed: {  
  computedProp () {  
    return this.message.split('').reverse().join('')  
  }  
},
```

```
computedProp: {  
  get: function() {  
    return this.msg.split('').reverse().join('')  
  },  
  set: function(newVal) {  
    this.msg = newVal  
  }  
}
```

```
watch: {  
  question: function (newQuestion) {  
    // runs every time question changes  
  }  
},
```

- Computed properties
- Reactive
- Explicit setter
- Watch

Komponenter

```
props: {  
  parentMsg: String,  
  required: true  
},
```

```
this.$emit('stuffHappened', {  
  eventParam: 'Im a parameter!'  
})
```

```
<child  
  :parentMsg="messageFromParent"  
  v-on:stuffHappened="handleStuffHappened">  
</child>
```

```
methods: {  
  whateverFunc (param) {  
    // ...code  
  }  
}
```

- Props - from parent to child
- Emit - from child to parent
- Methods - component level functions

Komponenten

```
<style>
/* global styles */
</style>
```

```
<style scoped>
/* local styles */
</style>
```

```
<style scoped>
.a >>> .b { /* ... */ }
</style>
```

- Styles - optionally scoped
- CSS, SASS, SCSS, LESS, STYLUS
- Deep selector

Routing

```
const routes = [{
  path: "/foo",
  component: Foo,
  name: 'namedRoute',
  children: [{ path: "/subfoo", component: bar }]
}];
const router = new VueRouter({
  routes // shorthand for `routes: routes`
});
const app = new Vue({
  router
}).$mount("#app");
```

```
<template>
  <router-link to="/foo">Go to Foo</router-link>
  <router-link to="{name: 'namedRoute'}">Go to Bar</router-link>
</template>
```

```
router.push('home')
router.push({ path: 'home' })
router.push({ name: 'user', params: { userId: 123 } })
router.push({ path: 'register', query: { plan: 'private' } })
```

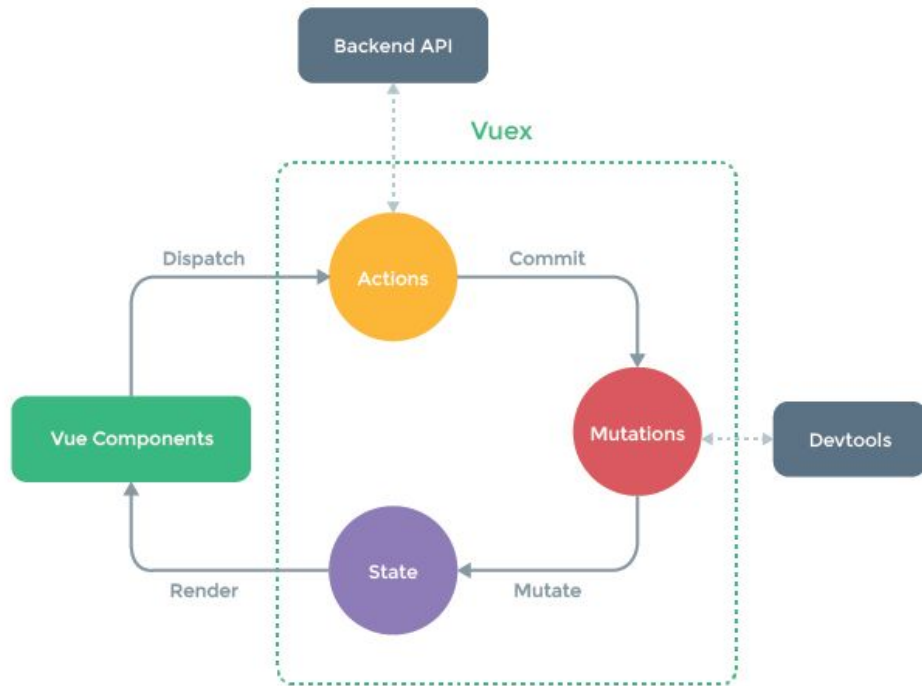
- Emulate multipage in SPA
- Define routes
- Inject router
- Navigate to route

State management

State management

Vuex

- Flux pattern
- Decoupled components
- Cleaner components
- Sane separation of concerns
- Time travel & debugging



Vuex

```
state: {  
  someArray: [],  
  someProp: {  
    name: 'Hej!'  
  }  
},
```

```
mutations: {  
  setSomeArray: (state, val) => state.someArray = val  
},
```

```
getters: {  
  getSomeArraySorted: (state, getters) =>  
    [...state.someArray].sort((a, b) => a < b ? 1 : -1)  
},
```

```
actions: {  
  async doSlowStuff({commit}, {param1, param2}) {  
    serverCall(param1, param2).then(res =>  
      Promise.resolve(res.body))  
  
    commit('setSomeArray', await serverCall)  
  }  
}
```

- App State
- Mutations
- Getters
- Actions

Vuex

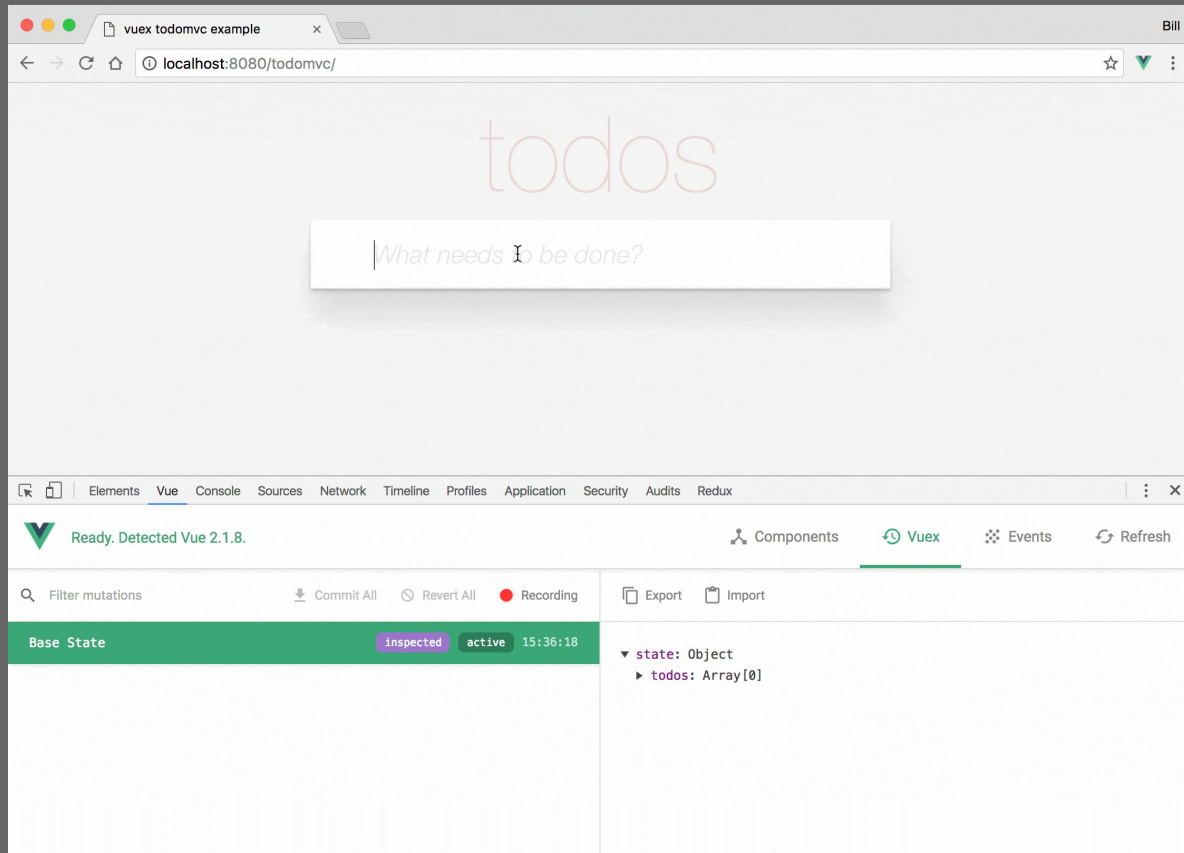
```
toggleTodo() {  
  this.$store.dispatch("toggleTodoAsync", this.todo);  
},
```

```
actions: {  
  async toggleTodoAsync({ commit }, todo: Todo) {  
    commit("setLoading", true);  
    await new Promise((resolve, reject) => setTimeout(resolve, 500)).then(  
      () => {  
        commit("toggleTodo", todo);  
        commit("setLoading", false);  
      }  
    );  
  }  
}
```

```
computed: {  
  todos(): Todo[] {  
    return this.$store.state.todos;  
  }  
},
```

- Dispatch actions
- Handle side effects
- Read state

Dev tools



Build system

- Webpack support
- Testing
- IDE's
- AOT or run time compile
- CLI

CLI

```
lassefisker-MBP-2:Desktop lassefisker$ vue create awesomeapp
```

```
(Press <space> to select, <a> to toggle all, <i> to invert selection)  
> ☒ TypeScript  
  ☐ Progressive Web App (PWA) Support  
  ☐ Router  
  ☐ Vuex  
  ☐ CSS Pre-processors  
  ☐ Linter / Formatter  
  ☐ Unit Testing  
  ☐ E2E Testing
```

- Project scaffolding
- Beta right now after major upgrade
- Zero config
- No lock in - configs can be added at any time

Lets code!



Questions?



Interested?

<https://it-minds.dk/karriere/stillinger/>

