



VueJS



a simple Framework for simple people



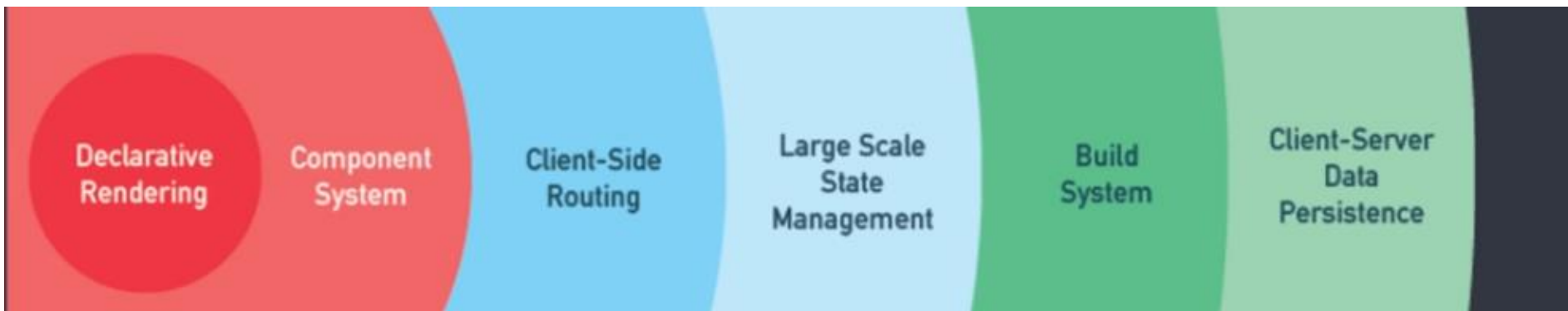
What's VueJS

- progressive framework for building user interfaces.
- Easy to pick up and integrate with other libraries or existing projects.
- Capable of powering modern progressive web applications
 - Here is an [Hello World fiddle](#).



Why VueJS

- Its amazingly simple
- It can be effectively used to build simple to large applications
 - It is used by large companies such as Alibaba and Baidu
 - Its built in within Laravel PHP
 - Its small – 19KB! minified and gzipped





Feature Rich

Computed Properties **Syncing**

Events **Outputting Data**

Developer Tools **Filters**

Watchers **Components**

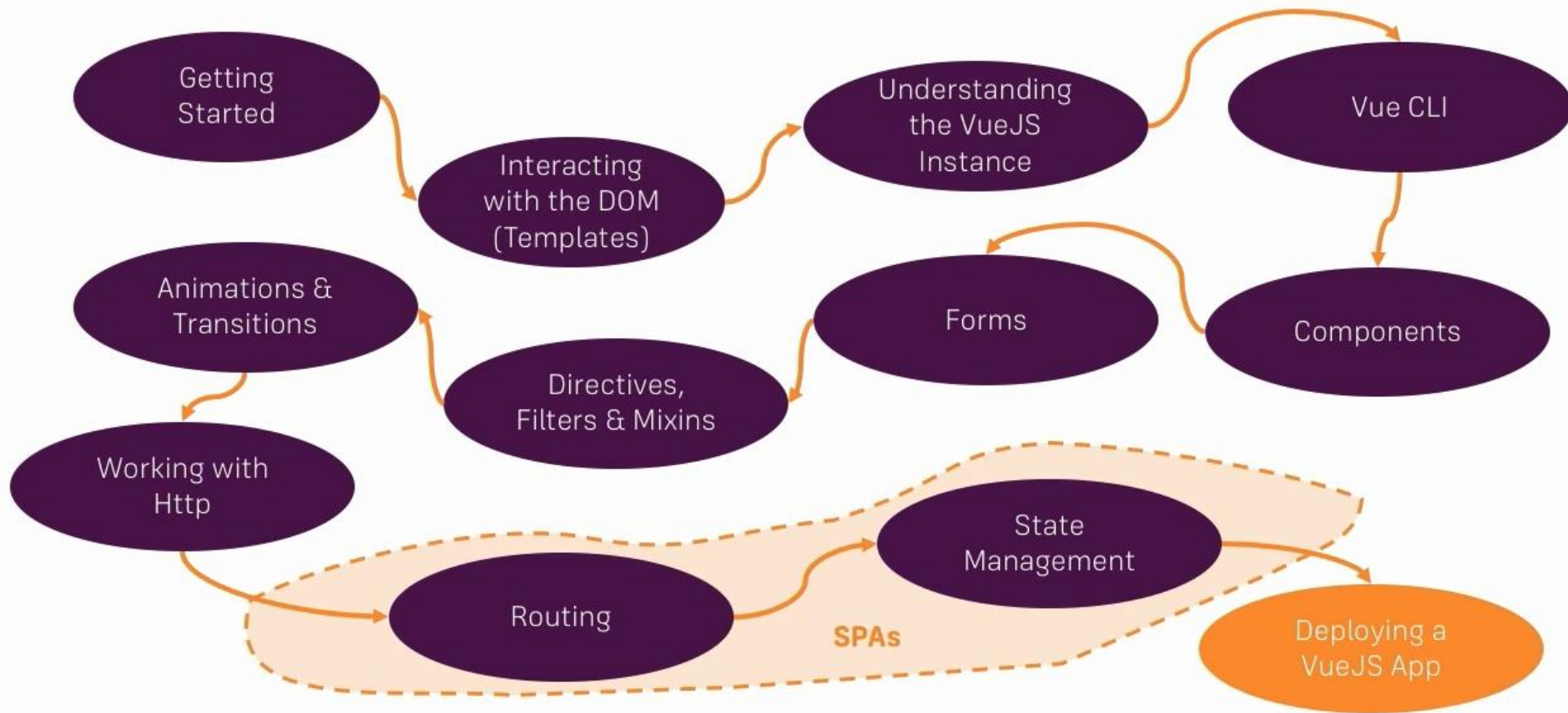
Lists **Mixins**

Forms **Conditionals**

Reactivity **Directives** **Dynamic Styles**



The Plan





Declarative Rendering

Render data to the DOM using template syntax:

```
<div id="app">  
  <span v-bind:title="msg">  
    {{ msg }}  
  </span>  
</div>
```

```
var app = new Vue({  
  el: '#app',  
  data: {  
    msg: 'Hello Vue!' + Date.now()  
  }  
})
```



Conditions & Loops

Toggle the presence of an element

```
<p v-if="seen">Now you see me</p>
```

```
<ol>
```

```
  <li v-for="pet in pets">
```

```
    {{ pet.name }}
  </li>
```

```
</ol>
```

```
</ol>
```



Handling Events

Use the v-on directive to attach event listeners that invoke methods on our Vue instances:

```
<div id="app">
  <span :class="{editable: editMode}"
        :contenteditable="editMode"
        v-on:blur="editMode=false">
    Hello {{userName}}
  </span>
  <button v-show="!editMode"
        v-on:click="toggleEdit">⌵</button>
</div>
```

Lets see it in [action](#)



Two way data binding

The v-model directive makes two-way binding between form input and our model

```
<p>{{ message }}</p>
```

```
<input v-model="message">
```



Vue devtools

Install vue-devtools extension

- See selected: \$vm0



Vue.js devtools

offered by <https://vuejs.org>

★★★★★ (695)

[Developer Tools](#)

262,130 users



Exercise

- Create a new project – **eventos**
- Build a service - **EventosService**
 - The entity should be called **evento {id, name}**
(to avoid confusion with dom events)
 - This service has a method **query** that returns a promise for a list of eventos
- In your root component, render the list, if empty show msg: “No events”
- Allow adding an event