



Über den Tellerrand blicken: Oder wie funktioniert eigentlich dieses Vue?

Marc Teufel



Unsere Agenda

- Kurzes Recap: Was ist eigentlich dieses Vue ? ✓
- Erste Vue-Anwendnungen schreiben ✓
- Geschmacksrichtungen ✓
 - Verschiedene Builds von Vue ✓
 - Scaffolding mit Vite ✓
- Komponenten und Hierarchien
- Kommunikation und Datenfluss

Komponenten und Hierarchien

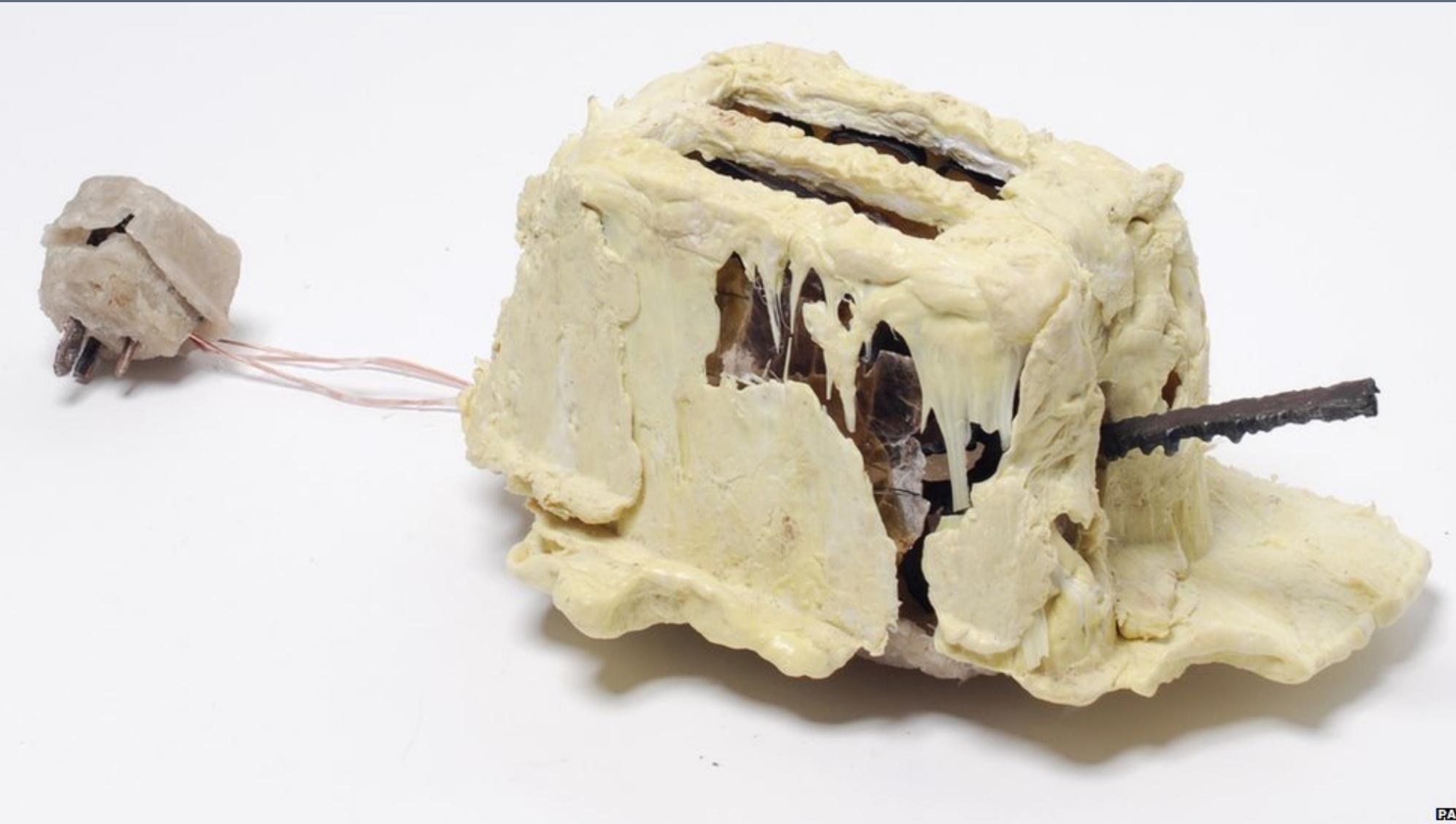




Komponenten

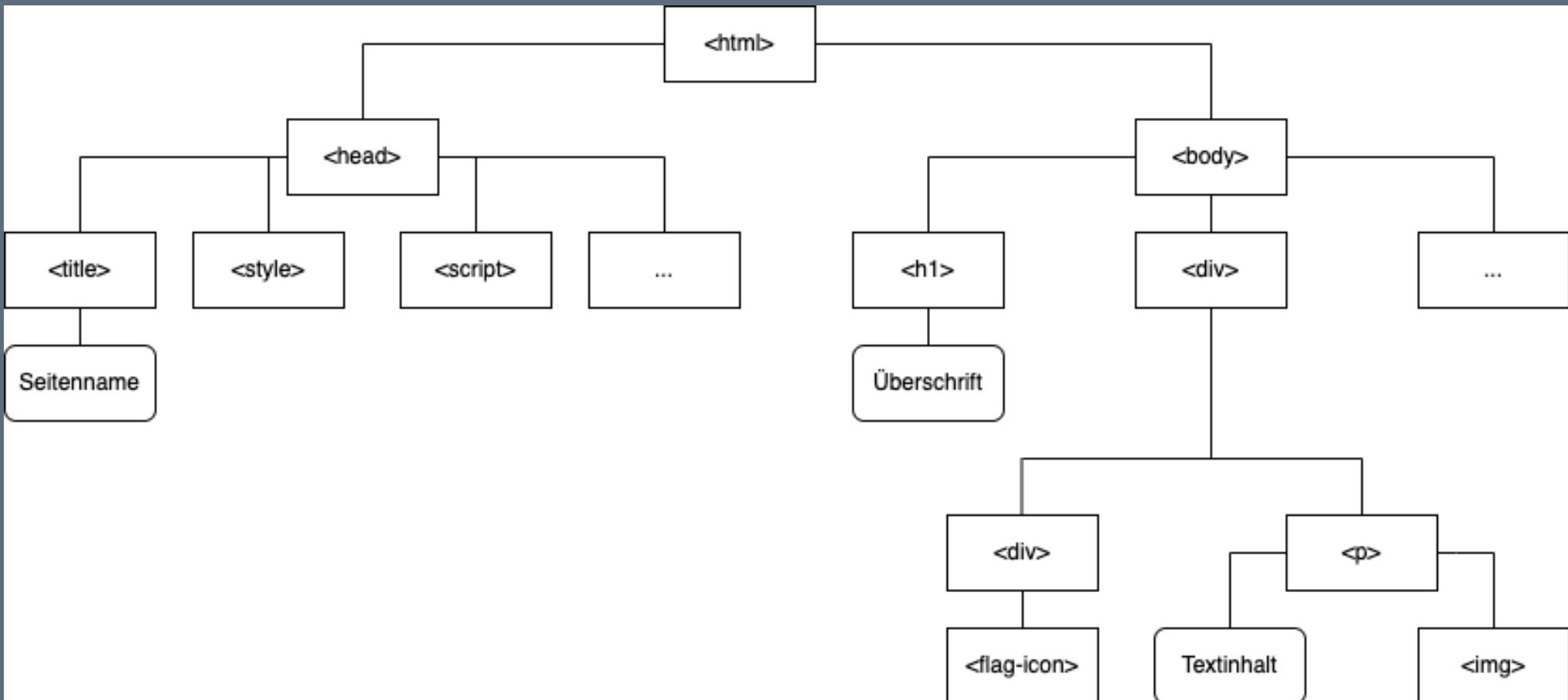
- Komponente kommt aus dem Lateinischen “compere” und bedeutet “zusammensetzend”
- Auch außerhalb der IT ist Komponentenorientierung ein gebräuchliches Muster
- Wir Entwickler glauben viel zu oft:
 - Wir können selbst machen
 - Wir können alles auf einmal machen
 - Struktur oder Architektur denken wir uns selbst aus!
 - Frameworks ? Zu was?
 - Pustekuchen!

Schiefgegangene Komponentenorientierung



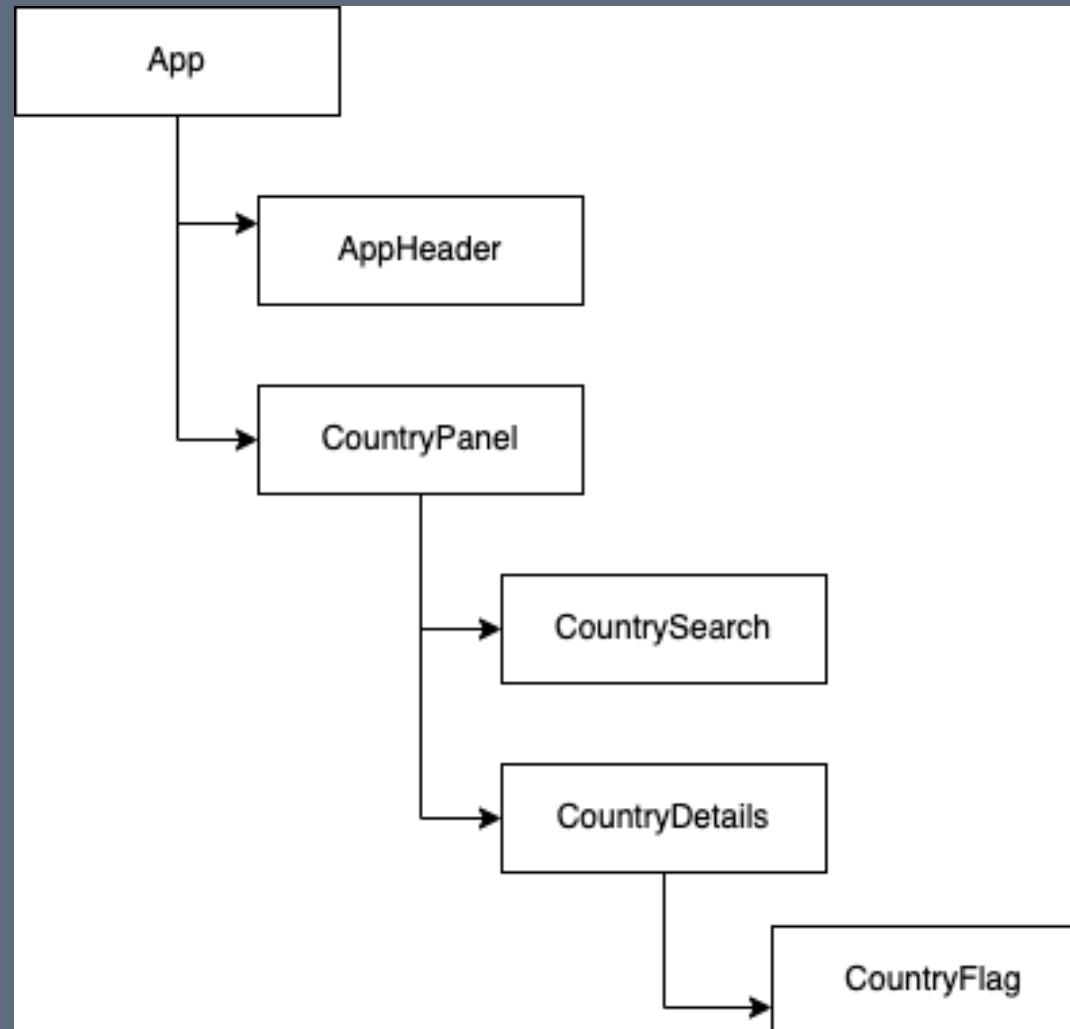
Komponenten sind in Hierarchien geordnet

- Eigentlich genauso wie der DOM:



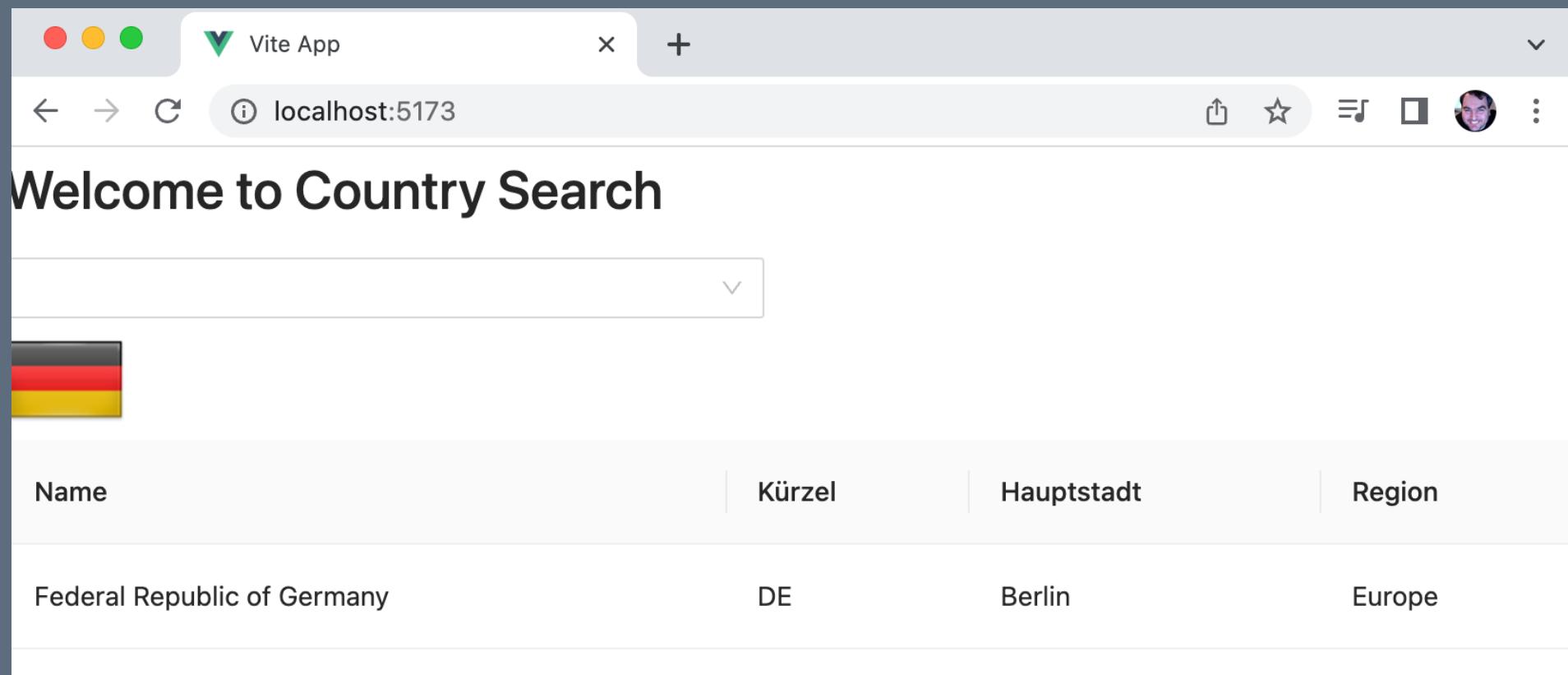
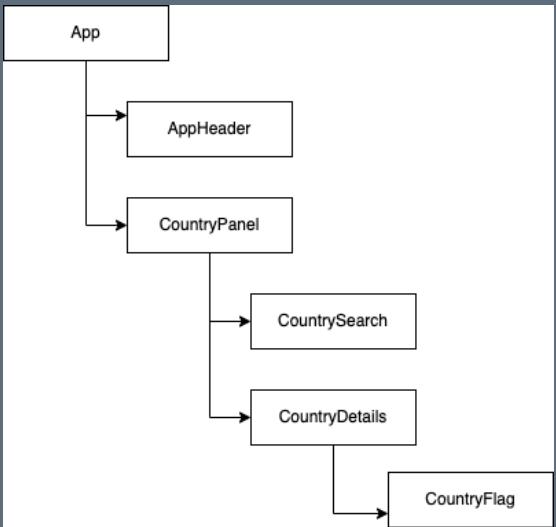
Komponenten sind in Hierarchien geordnet

- So könnte eine Komponentenhierarchie aussehen:



Komponenten sind in Hierarchien geordnet

- Komponentenhierarchie und die zugehörige Beispielanwendung:

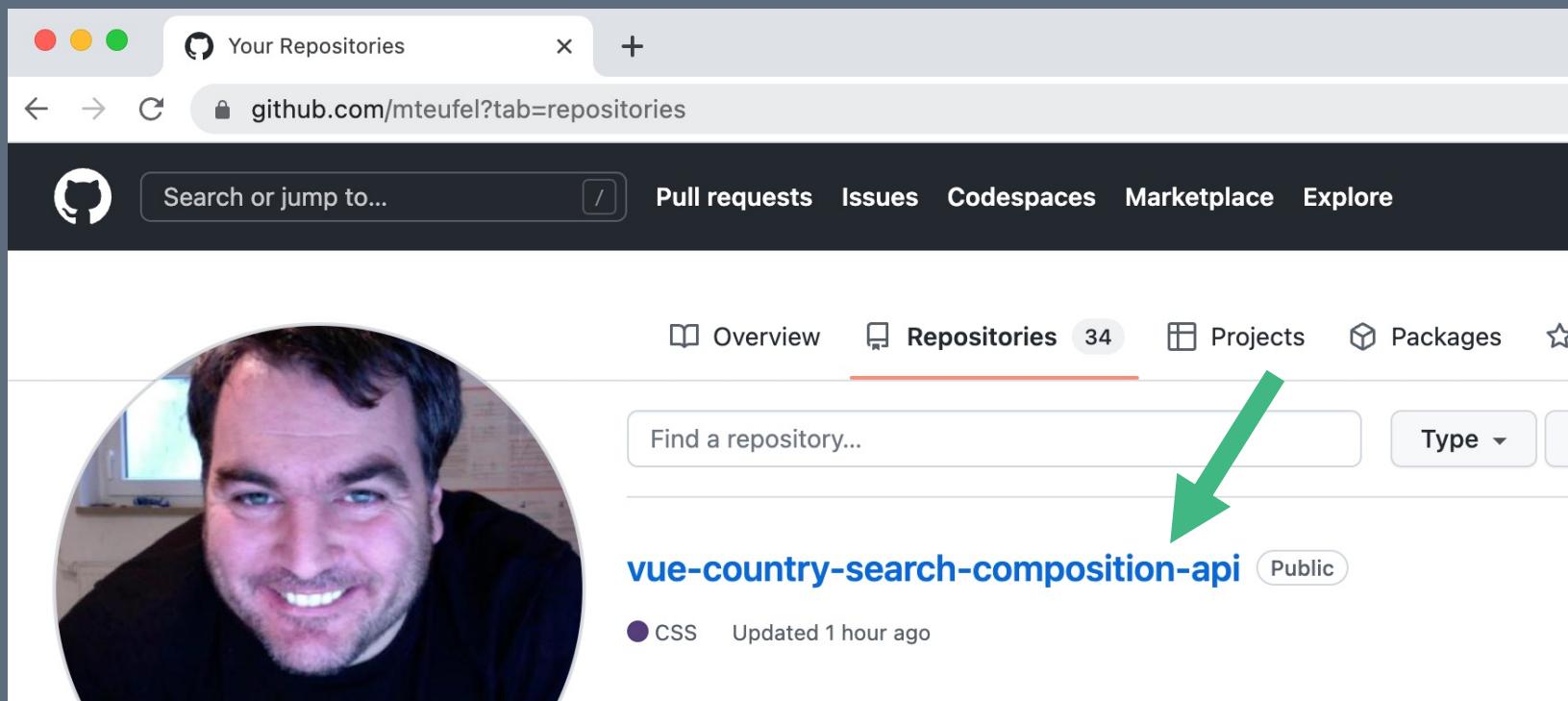


**It's
workshop-
Time!**



[git clone https://github.com/mteufel/vue-country-search-composition-api.git](https://github.com/mteufel/vue-country-search-composition-api.git)

oder



Kommunikation und Datenfluss



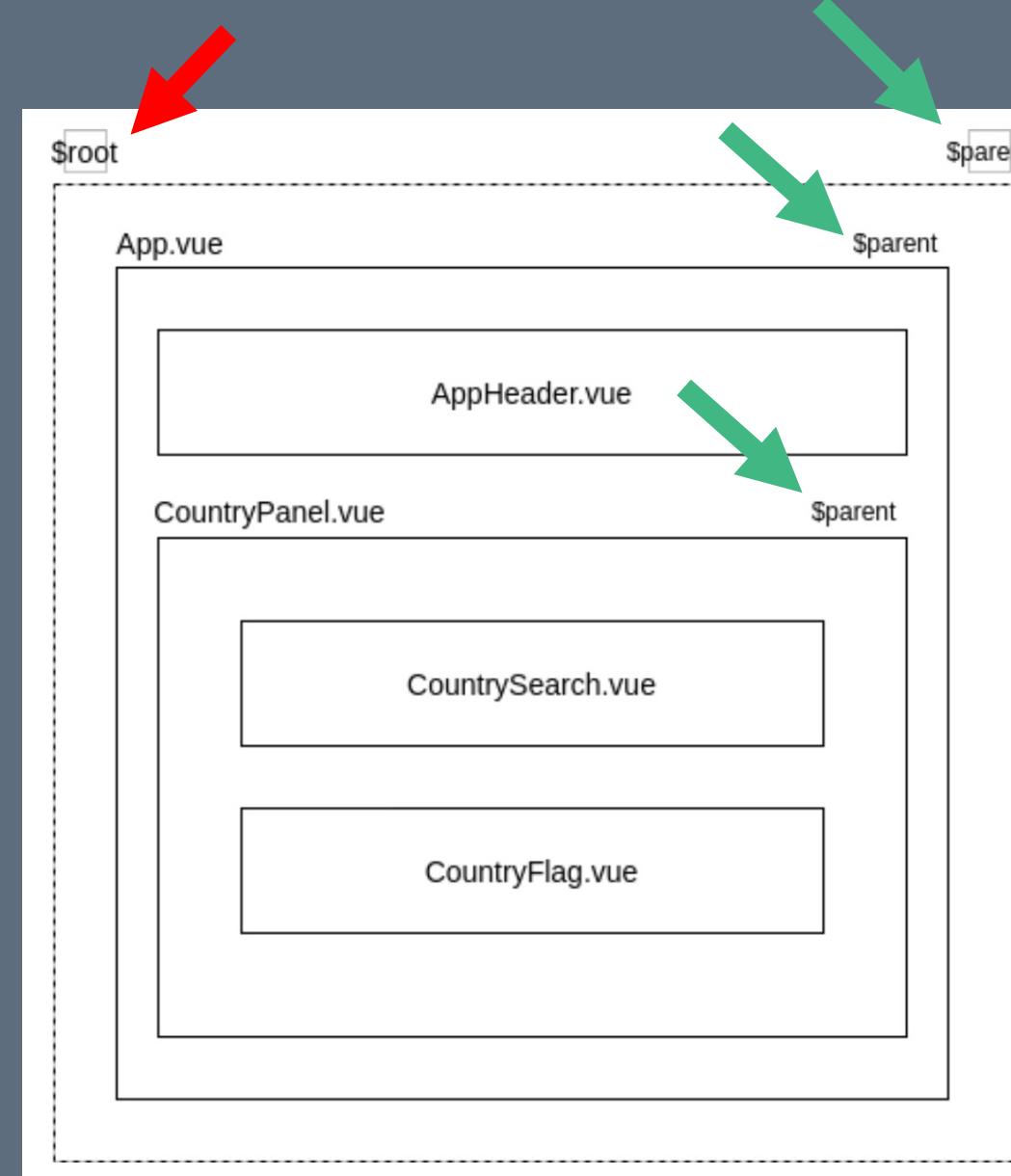
Aller guten Dinge sind DREI (und nicht vier!)



- \$root/\$parent
- Durchreichen → Props/Emit
- Rumreichen → Globale Stores, State Management
- Reinreichen → Provide/Inject



\$root/\$parent



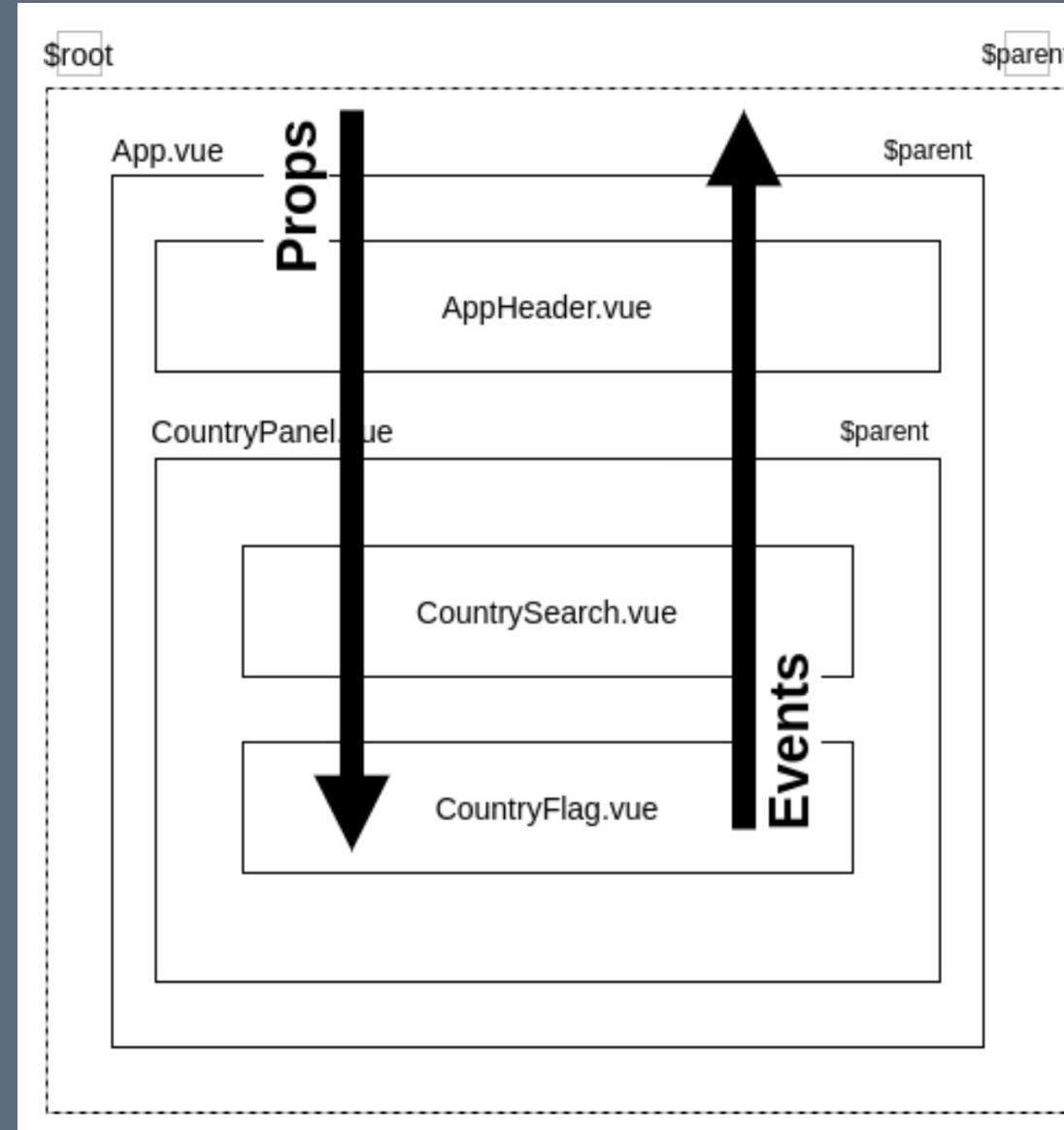


\$root/\$parent

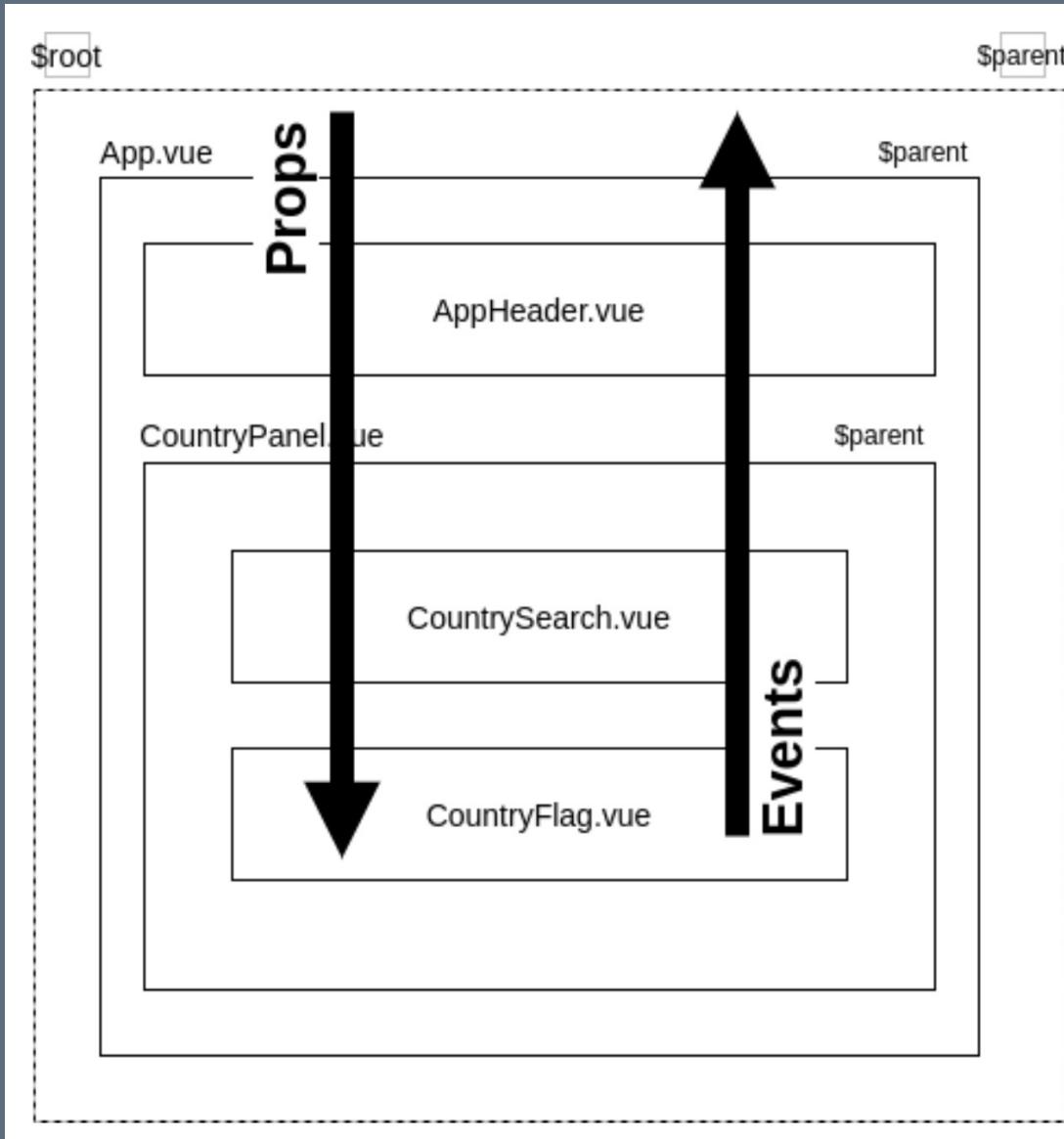




Daten durchreichen mit dem Standard-Weg: Props/Emit



Daten durchreichen mit dem Standard-Weg: Props/Emit



```
▼ App.vue M X
src > ▼ App.vue > {} style
<script setup>
1
2
3 import { ref } from 'vue'
4 import AppHeader from './components/AppHeader.vue'
5 import CountryPanel from './components/CountryPanel.vue'
6 import CountrySearch from './components/CountrySearch.vue'
7 import CountryDetail from './components/CountryDetail.vue';
8
9 const country = ref('DE')
10
11 function countryHasBeenChanged(selectedCountry) {
12   country.value = selectedCountry
13 }
14
15 </script>
16
17 <template id="app">
18   <div>
19     <AppHeader title="Welcome to Country Search"></AppHeader>
20     <CountryPanel>
21       <CountrySearch @country-changed="countryHasBeenChanged"/>
22       <CountryDetail :country="country" />
23     </CountryPanel>
24   </div>
25 </template>
```

Two green arrows highlight specific parts of the code:

- An arrow points to the `countryHasBeenChanged` function definition.
- An arrow points to the `@country-changed` event binding in the template.

Daten durchreichen mit dem Standard-Weg: Props/Emit

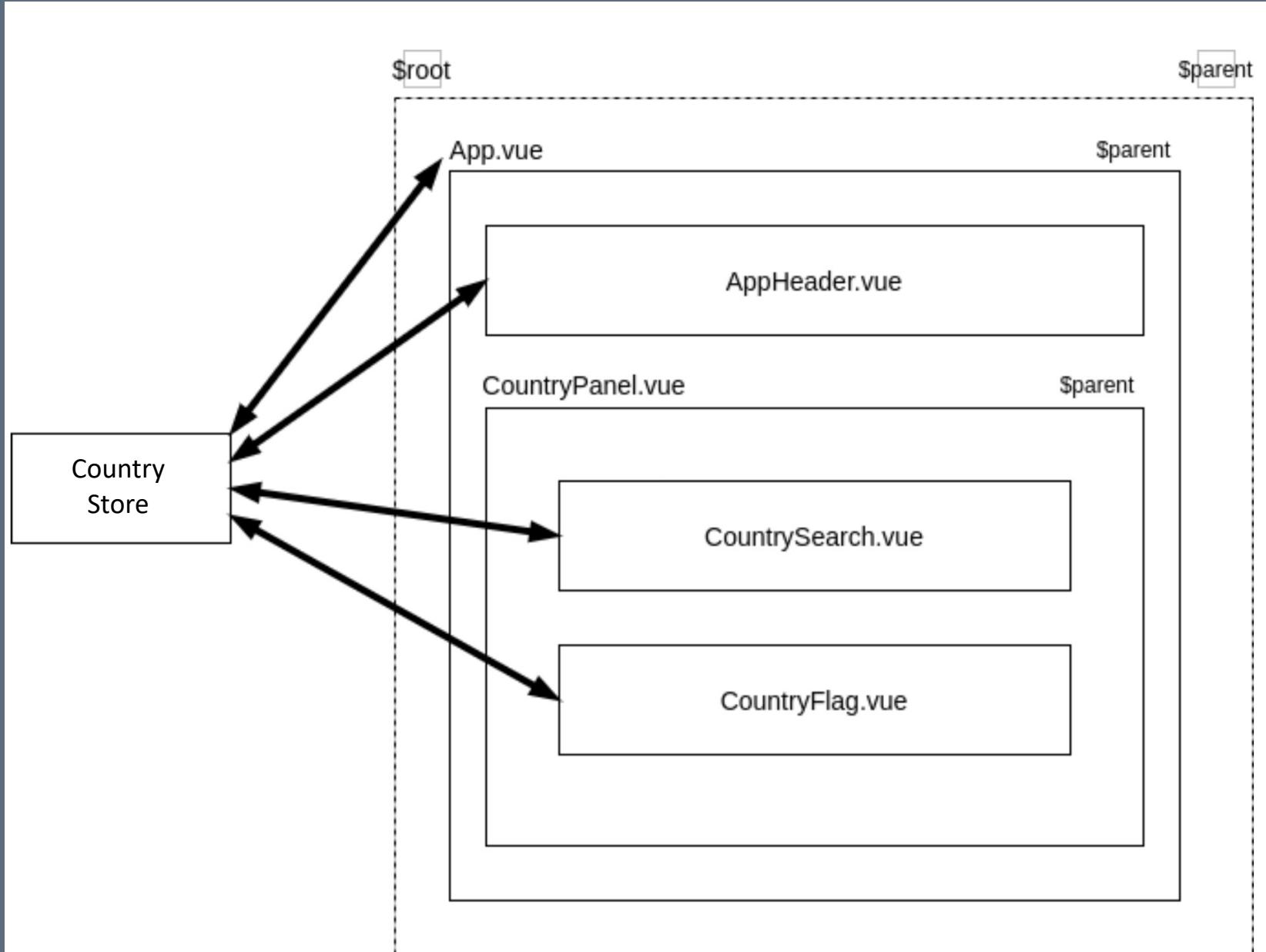
App.vue M X

```
src > App.vue > {} style
1  <script setup>
2
3    import { ref } from 'vue'
4    import AppHeader from './components/AppHeader.vue'
5    import CountryPanel from './components/CountryPanel.vue'
6    import CountrySearch from './components/CountrySearch.vue'
7    import CountryDetail from './components/CountryDetail.vue';
8
9    const country = ref('DE')
10
11   function countryHasBeenChanged(selectedCountry) {
12     country.value = selectedCountry
13   }
14
15 </script>
16
17 <template id="app">
18   <div>
19     <AppHeader title="Welcome to Country Search"></AppHeader>
20     <CountryPanel>
21       <CountrySearch @country-changed="countryHasBeenChanged"/>
22       <CountryDetail :country="country" />
23     </CountryPanel>
24   </div>
25 </template>
```

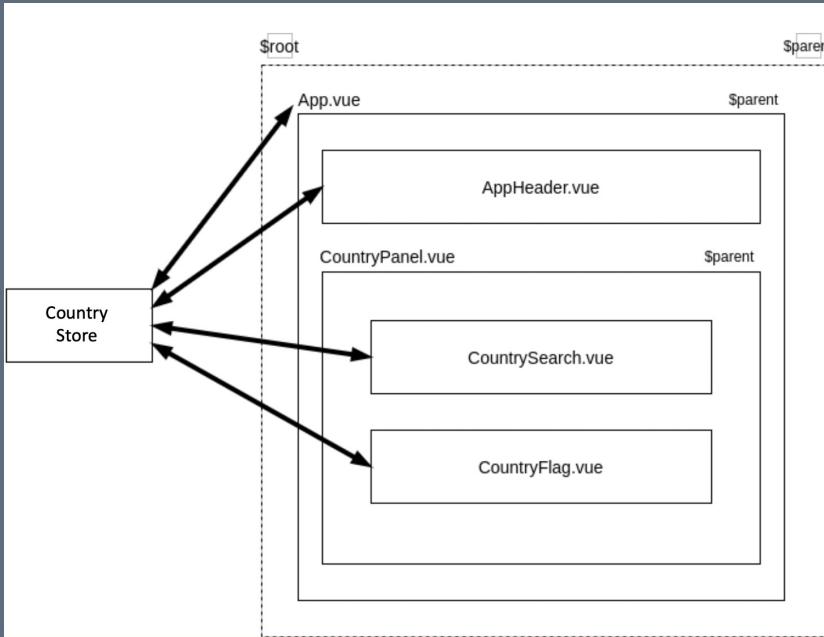
CountrySearch.vue M X

```
src > components > CountrySearch.vue > ...
1
2
3
4
5
6
7  <script setup>
8
9    import { ref, defineEmits } from 'vue'
10   import { Select } from 'ant-design-vue'
11   import { searchForCountries, firstRows } from '../data/countries'
12
13   const countries = ref( [ ] )
14
15   const emit = defineEmits(['country-changed'])
16
17   function handleChange(search) {
18     countries.value = []
19     if (search === "") {
20       return
21     }
22     // search the countries and convert to an array that is understandable
23     let result = searchForCountries(search).map( c => {
24       return { value: c.cca2, label: c.name.official }
25     })
26     // shrink result down to 10 countries max
27     countries.value = firstRows(result, 10)
28   }
29
30   function countrySelected(cca2) {
31     console.log('Country selected: ', cca2)
32     emit('country-changed', cca2)
33 }
```

Daten rumreichen mit Stores



Einen einfachen Store implementieren



```
import { searchForCountries, firstRows } from './countries'

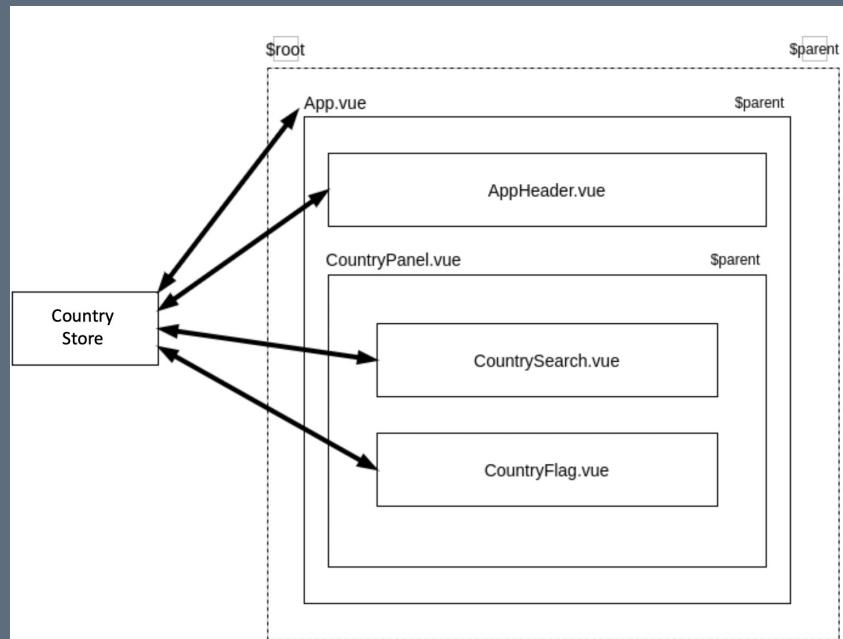
const createCountryStore = () => {

  let selectedCountry = 'IT'
  let options = []
  return {
    getOptions: () => options,
    getSelectedCountry: () => selectedCountry,
    doSearch: (search) => {
      if (search === "") {
        return
      }
      // search the countries and convert to an array that is understandable for ant design
      let result = searchForCountries(search).map( c => {
        return { value: c.cca2, label: c.name.official }
      })
      // shrink result down to 10 countries max
      options = firstRows(result, 10)
    }
  }
}

const countryStore = createCountryStore()
export default countryStore
```



Den Store einbauen



CountrySearch.vue

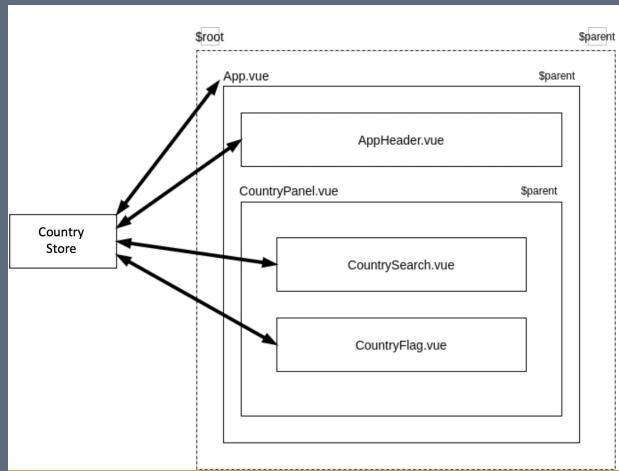
```
src > components > CountrySearch.vue > {} style > .country-search
```

```
1  <script>
2      export default {
3          name: "CountrySearch"
4      };
5  </script>
6
7  <script setup>
8
9      import { ref } from 'vue'
10     import { Select } from 'ant-design-vue'
11     import countryStore from '../data/countryStore'
12
13     const countries = ref( [ ] )
14
15     function handleChange(search) {
16         countryStore.doSearch(search)
17         countries.value = countryStore.getOptions()
18     }
19
20     function countrySelected(cca2) {
21         console.log('Country selected: ', cca2)
22         countryStore.setCountry(cca2)
23     }
24
25 </script>
26
```

A green arrow points to the line `countryStore.setCountry(cca2)` in the code, and another green arrow points to the line `countryStore.getOptions()`.



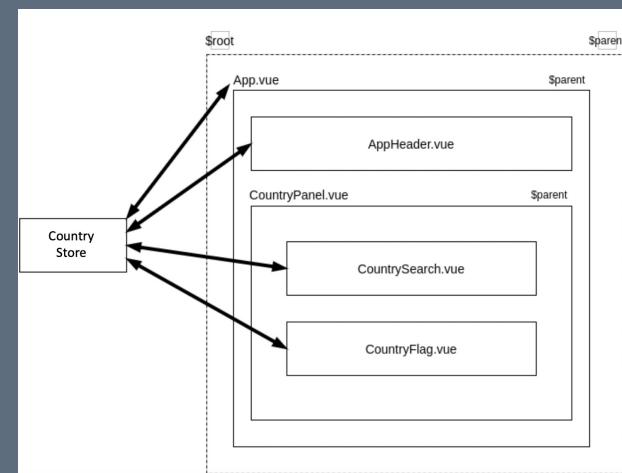
Observer Pattern hilft!



JS countryStore.js •

```
src > data > JS countryStore.js > ...
1 import { searchForCountries, firstRows, getCountryByCCAA } from './countries'
2
3 const createCountryStore = () => {
4
5     let selectedCountry = 'IT'
6     let options = []
7     let subscribers = []
8
9     return {
10        getOptions: () => options,
11        getSelectedCountry: () => selectedCountry,
12
13        doSearch: (search) => {
14            if (search === "") {
15                return
16            }
17            // search the countries and convert to an array that is understandable
18            let result = searchForCountries(search).map( c => {
19                return { value: c.cca2, label: c.name.official }
20            })
21            // shrink result down to 10 countries max
22            options = firstRows(result, 10)
23        },
24
25        subscribe: (fn) => {
26            subscribers.push(fn)
27        },
28        setCountry: (country) => {
29            selectedCountry = country;
30            subscribers.forEach( s => s() )
31        }
32    }
33}
34
35
36 const countryStore = createCountryStore()
37 export default countryStore
```

Observer Pattern hilft! – Den Client „anmelden“



```
App.vue ●  
src > App.vue > ...  
1 <script setup>  
2  
3 import AppHeader  
4 import AppHeader from './components/AppHeader.vue'  
5 import CountryPanel from './components/CountryPanel.vue'  
6 import CountrySearch from './components/CountrySearch.vue'  
7 import CountryDetail from './components/CountryDetail.vue';  
8  
9 import countryStore from './data/countryStore'  
10  
11 const country = ref(countryStore.getSelectedCountry())  
12  
13 countryStore.subscribe( () => country.value = countryStore.getSelectedCountry() )  
14  
15 </script>  
16  
17 <template id="app">  
18 <div>  
19 <AppHeader title="Welcome to Country Search"></AppHeader>  
20 <CountryPanel>  
21 <CountrySearch />  
22 <CountryDetail :country="country" />  
23 </CountryPanel>  
24 </div>  
25 </template>
```

A green arrow points from the line "countryStore.subscribe(() => country.value = countryStore.getSelectedCountry())" in the script block to the "country" prop in the CountryDetail component in the template block.



Dependency Injection mit Provide/Inject

Normalerweise reichen wir Daten mit Props durch die Komponentenhierarchie

Was ist wenn die Hierarchie sehr tief ist? Wirklich überall durchreichen?

Hier kann Provide/Inject helfen!

→Daten von einem beliebigen Punkt in der Komponentenhierarchie nach oben oder unten reinreichen.

Dependency Injection mit Provide/Inject

App.vue M X

```
src > App.vue > ...
1  <script setup>
2
3  import { ref, provide, getCurrentInstance } from 'vue'
4  import AppHeader from './components/AppHeader.vue'
5  import CountryPanel from './components/CountryPanel.vue'
6  import CountrySearch from './components/CountrySearch.vue'
7  import CountryDetail from './components/CountryDetail.vue';
8
9  const country = ref('US')
10 provide('country', country)
11 console.log(getCurrentInstance())
12
13 </script>
14
15 <template id="app">
16   <div>
17     <AppHeader title="Welcome to Country Search"></AppHeader>
18     <CountryPanel>
19       <CountrySearch />
20       <CountryDetail />
21     </CountryPanel>
22   </div>
23 </template>
```

Welcome to Country Search



Name

United States of America

Kürzel

US

Hauptstad

Washington

Elements Console Sources Network Performance Memory Application Security Lighthouse Recorder

top Filter

```
> effect: ReactiveEffect {active: true, deps: Array(0), parent: undefined, fn: f, scheduler: f, ...}
> emit: f()
  emitsOptions: null
  emitted: null
> exposeProxy: Proxy(Object) {__v_skip: true}
> exposed: {__v_skip: true}
  inheritAttrs: undefined
  isDeactivated: false
  isMounted: true
  isUnmounted: false
  m: null
  next: null
  parent: null
> props: Proxy(Object) {}
> propsDefaults: {}
> propsOptions: []
> provides:
  > country: RefImpl {__v_isShallow: false, dep: Set(3), __v_isRef: true, _rawValue: 'US', _value: 'US'}
  > [[Prototype]]: Object
```

Dependency Injection durchführen

▼ App.vue M X

```
src > ▼ App.vue > ...
1  <script setup>
2
3  import { ref, provide, getCurrentInstance } from 'vue'
4  import AppHeader from './components/AppHeader.vue'
5  import CountryPanel from './components/CountryPanel.vue'
6  import CountrySearch from './components/CountrySearch.vue'
7  import CountryDetail from './components/CountryDetail.vue';
8
9  const country = ref('US')
10 provide('country', country)
11 console.log(getCurrentInstance())
12
13 </script>
14
15 <template id="app">
16   <div>
17     <AppHeader title="Welcome to Country Search"></AppHeader>
18     <CountryPanel>
19       <CountrySearch />
20       <CountryDetail />
21     </CountryPanel>
22   </div>
23 </template>
```



Dependency Injection durchführen

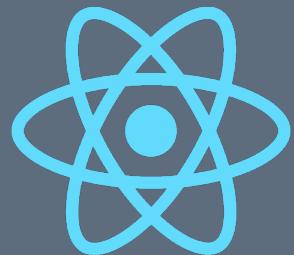
```
App.vue M
src > ▼ App.vue > ...
1  <script setup>
2
3  import { ref, provide
4  import AppHeader from
5  import CountryPanel
6  import CountrySearch
7  import CountryDetail
8
9  const country = ref('')
10 provide('country', co
11 console.log(getCurren
12
13 </script>
14
15 <template id="app">
16   <div>
17     <AppHeader title=...
18     <CountryPanel>
19       <CountrySearch>
20         <CountryDetail
21       </CountryPanel>
22     </div>
23   </template>
```

```
CountrySearch.vue M
src > components > ▼ CountrySearch.vue > ...
1  <script>
2    export default {
3      name: "CountrySearch"
4    };
5  </script>
6
7  <script setup>
8
9    import { ref, inject, provide } from 'vue'
10   import { Select } from 'ant-design-vue'
11   import { searchForCountries, firstRows } from '../data/countries'
12
13   const countries = ref( [ ] )
14   const country = inject('country') ↑
15
16   function handleChange(search) {
17     countries.value = []
18     if (search === "") {
19       return
20     }
21     // search the countries and convert to an array that is understandable
22     let result = searchForCountries(search).map( c => {
23       return { value: c.cca2, label: c.name.official }
24     })
25     // shrink result down to 10 countries max
26     countries.value = firstRows(result, 10)
27   }
28
29   function countrySelected(cca2) {
30     console.log('Country selected: ', cca2)
31     country.value = cca2
32   }
```

Dependency Injection durchführen

```
▼ App.vue M X
src > ▼ App.vue > ...
1  <script setup>
2
3  import { ref, provide, getCurrentInstance } from 'vue'
4  import AppHeader from './components/AppHeader.vue'
5  import CountryPanel from './components/CountryPanel.vue'
6  import CountrySearch from './components/CountrySearch.vue'
7  import CountryDetail from './components/CountryDetail.vue'
8
9  const country = ref('US') ▶
10 provide('country', country)      const countries = ref( [ ] )
11 console.log(getCurrentInstance()) const country = inject('country')
12
13 </script>
14
15 <template id="app">
16   <div>
17     <AppHeader title="Welcome to Country Search" />
18     <CountryPanel>
19       <CountrySearch />23
20       <CountryDetail />24
21     </CountryPanel>
22   </div>
23 </template>
24
25
26
27
28
29 function countrySelected(cca2) {
30   console.log('Country selected: ', cca2)
31   country.value = cca2
32 }
```

```
▼ CountryDetail.vue X
src > components > ▼ CountryDetail.vue > {} template
1  <script>
2  export default {
3    name: "CountryDetail",
4    components: { CountryFlag }
5  };
6  </script>
7
8  <script setup>
9  import { ref, computed, inject } from 'vue'
10 import { Table } from 'ant-design-vue'
11 import { getCountryByCCA2 } from '../data/countries'
12 import CountryFlag from './CountryFlag.vue';
13
14 const country = inject('country') ▶
15
16 const cols = ref( [ { title: 'Name', dataIndex: 'name' },
17                     { title: 'Kürzel', dataIndex: 'kuerzel' },
18                     { title: 'Hauptstadt', dataIndex: 'hauptstadt' },
19                     { title: 'Region', dataIndex: 'region' }
20                   ] )
21
22 const daten = computed( () => {
23   let c = getCountryByCCA2(country.value)
24   return [ { name: c.name.official, kuerzel: c.cca2, hauptstadt: c.capita
25 } )
26
27 </script>
28
29 <template>
30   <div>
31     <CountryFlag :country = "country" />
32   </div>
33   <div>
34     <Table :columns="cols" :data-source="daten" :pagination="false" />
35   </div>
36 </template>
```



Sie suchen einen auf Sie zugeschnittenen Workshop
oder Hilfe und Unterstützung in Ihren
Vue/React/JavaScript-Projekten ?

Ich unterstütze Sie gern!

Kontaktieren Sie mich:



Mail:

teufel.marc@gmail.com



Twitter: [@marcteufel](https://twitter.com/marcteufel)



Ant Design