

Vue Vapor: 重新发明



2024-07-06

Vue Vapor: 重新发明

🚀 探索无虚拟 DOM 的 Vue

Exploring Vue without Virtual DOM



2024-07-06

三咲智子 *Kevin Deng*

作者  Vue Vapor /  Vue Macros /  Elk

核心团队成员  Vue /  VueUse

维护者  Unplugin



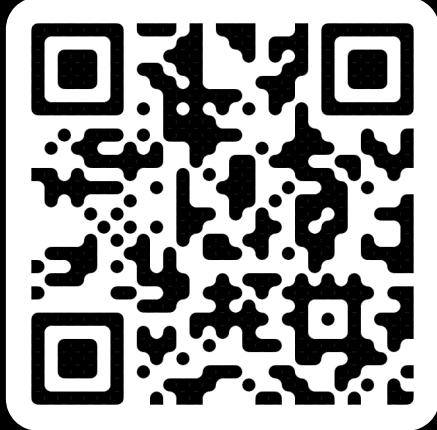
 sxzz  sanxiaozhizi  三咲智子



听过 Vue Vapor 吗?



听过 Vue Vapor 吗?





听过 Vue Vapor 吗？



没听过	听说过
0	0
使用过	参与开发过
0	0

🙋‍♂️ 为什么叫 Vapor ?

💁‍♀️ 为什么叫 Vapor ?



Vue Vapor 是什么？

What's Vue Vapor?

Vue Vapor 是什么？

全新的渲染机制

What's Vue Vapor?

Vue Vapor 是什么？

无虚拟 DOM

全新的渲染机制

What's Vue Vapor?

Vue Vapor 是什么？

无虚拟 DOM 提升性能

全新的渲染机制

What's Vue Vapor?

Vue Vapor 是什么？

无虚拟 DOM 提升性能 减少包体积

全新的渲染机制

What's Vue Vapor?

Vue Vapor 是什么？

无虚拟 DOM

提升性能

减少包体积

全新的渲染机制

基于响应性 @vue/reactivity

原理 mechanism

Vue Vapor 基于 `@vue/reactivity` 响应性系统，通过追踪数据的变化来更新 DOM。

原理 mechanism

Vue Vapor 基于 `@vue/reactivity` 响应性系统，通过追踪数据的变化来更新 DOM。

```
1 // 初始化
2 const container = document.createElement('div')
3 const label = document.createElement('h1')
4 const button = document.createElement('button')
5 button.textContent = 'Increase'
6
7 button.addEventListener('click', increase) // 注册事件
8 let count = 0
9 const render = () => {
10   label.textContent = `Count: ${count}`
11 }
12 function increase() {
13   count++
14   render() // 重新渲染
15 }
16 render() // 首次渲染
17
18 document.body.append(container)
19 container.append(label, button)
```

原理 mechanism

Vue Vapor 基于 `@vue/reactivity` 响应性系统，通过追踪数据的变化来更新 DOM。

```
1 // 初始化
2 const container = document.createElement('div')
3 const label = document.createElement('h1')
4 const button = document.createElement('button')
5 button.textContent = 'Increase'
6
7 button.addEventListener('click', increase) // 注册事件
8 let count = 0
9 const render = () => {
10   label.textContent = `Count: ${count}`
11 }
12 function increase() {
13   count++
14   render() // 重新渲染
15 }
16 render() // 首次渲染
17
18 document.body.append(container)
19 container.append(label, button)
```

原理 mechanism

Vue Vapor 基于 `@vue/reactivity` 响应性系统，通过追踪数据的变化来更新 DOM。

```
1 // 初始化
2 const container = document.createElement('div')
3 const label = document.createElement('h1')
4 const button = document.createElement('button')
5 button.textContent = 'Increase'
6
7 button.addEventListener('click', increase) // 注册事件
8 let count = 0
9 const render = () => {
10   label.textContent = `Count: ${count}`
11 }
12 function increase() {
13   count++
14   render() // 重新渲染
15 }
16 render() // 首次渲染
17
18 document.body.append(container)
19 container.append(label, button)
```

原理 mechanism

Vue Vapor 基于 `@vue/reactivity` 响应性系统，通过追踪数据的变化来更新 DOM。

```
1 // 初始化
2 const container = document.createElement('div')
3 const label = document.createElement('h1')
4 const button = document.createElement('button')
5 button.textContent = 'Increase'
6
7 button.addEventListener('click', increase) // 注册事件
8 let count = 0
9 const render = () => {
10   label.textContent = `Count: ${count}`
11 }
12 function increase() {
13   count++
14   render() // 重新渲染
15 }
16 render() // 首次渲染
17
18 document.body.append(container)
19 container.append(label, button)
```

原理 mechanism

Vue Vapor 基于 `@vue/reactivity` 响应性系统，通过追踪数据的变化来更新 DOM。

```
1 import { ref } from '@vue/reactivity'
2
3 // 初始化
4 const container = document.createElement('div')
5 const label = document.createElement('h1')
6 const button = document.createElement('button')
7 button.textContent = 'Increase'
8
9 button.addEventListener('click', increase) // 注册事件
10 const count = ref(0)
11 const render = () => {
12   label.textContent = `Count: ${count.value}`
13 }
14 function increase() {
15   count.value++
16   render() // 重新渲染
17 }
18 render() // 首次渲染
19
20 document.body.append(container)
21 container.append(label, button)
```

原理 mechanism

Vue Vapor 基于 `@vue/reactivity` 响应性系统，通过追踪数据的变化来更新 DOM。

```
1 import { effect, ref } from '@vue/reactivity'
2
3 // 初始化
4 const container = document.createElement('div')
5 const label = document.createElement('h1')
6 const button = document.createElement('button')
7 button.textContent = 'Increase'
8
9 button.addEventListener('click', increase) // 注册事件
10 const count = ref(0)
11 effect(() => {
12   label.textContent = `Count: ${count.value}`
13 })
14 function increase() {
15   count.value++
16 }
17
18 document.body.append(container)
19 container.append(label, button)
```

原理 mechanism

Vue Vapor 基于 `@vue/reactivity` 响应性系统，通过追踪数据的变化来更新 DOM。

```
1 import { effect, ref } from '@vue/reactivity'
2
3 // 初始化
4 const container = document.createElement('div')
5 const label = document.createElement('h1')
6 const button = document.createElement('button')
7 button.textContent = 'Increase'
8
9 button.addEventListener('click', increase) // 注册事件
10 const count = ref(0)
11 effect(() => {
12   label.textContent = `Count: ${count.value}`
13 })
14 function increase() {
15   count.value++
16 }
17
18 document.body.append(container)
19 container.append(label, button)
```

Vapor SFC 编译 compilation

```
<script setup vapor>
import { ref } from 'vue'

const count = ref(0)

function increase() {
  count.value++
}

</script>

<template>
  <h1>Count: {{ count }}</h1>
  <button @click="increase">Increase</button>
</template>
```

Vapor SFC 编译 compilation

```
<script setup vapor>
import { ref } from 'vue'

const count = ref(0)

function increase() {
  count.value++
}
</script>

<template>
  <h1>Count: {{ count }}</h1>
  <button @click="increase">Increase</button>
</template>
```

Vapor SFC 编译 compilation

```
<script setup vapor>
import { ref } from 'vue'

const count = ref(0)

function increase() {
  count.value++
}

</script>

<template>
  <h1>Count: {{ count }}</h1>
  <button @click="increase">Increase</button>
</template>
```

```
import { delegate, delegateEvents, renderEffect, setText,
        template } from 'vue/vapor'
import { ref } from 'vue'

const t0 = template('<h1>')
const t1 = template('<button>Increase')
delegateEvents('click')

export default {
  setup() {
    const count = ref(0)
    function increase() {
      count.value++
    }

    const n0 = t0()
    const n1 = t1()
    delegate(n1, 'click', () => increase)
    renderEffect(() => setText(n0, 'Count: ',
                                count.value))
    return [n0, n1]
  },
}
```

Vapor SFC 编译 compilation

```
<script setup vapor>
import { ref } from 'vue'

const count = ref(0)

function increase() {
  count.value++
}

</script>

<template>
  <h1>Count: {{ count }}</h1>
  <button @click="increase">Increase</button>
</template>
```

```
import { delegate, delegateEvents, renderEffect, setText,
        template } from 'vue/vapor'
import { ref } from 'vue'

const t0 = template('<h1>')
const t1 = template('<button>Increase')
delegateEvents('click')

export default {
  setup() {
    const count = ref(0)
    function increase() {
      count.value++
    }

    const n0 = t0()
    const n1 = t1()
    delegate(n1, 'click', () => increase)
    renderEffect(() => setText(n0, 'Count: ',
                                count.value))
    return [n0, n1]
  }
}
```

Vapor SFC 编译 compilation

```
<script setup vapor>
import { ref } from 'vue'

const count = ref(0)

function increase() {
  count.value++
}

</script>

<template>
  <h1>Count: {{ count }}</h1>
  <button @click="increase">Increase</button>
</template>
```

```
import { delegate, delegateEvents, renderEffect, setText,
        template } from 'vue/vapor'
import { ref } from 'vue'

const t0 = template('<h1>')
const t1 = template('<button>Increase')
delegateEvents('click')

export default {
  setup() {
    const count = ref(0)
    function increase() {
      count.value++
    }

    const n0 = t0()
    const n1 = t1()
    delegate(n1, 'click', () => increase)
    renderEffect(() => setText(n0, 'Count: ',
                                count.value))
    return [n0, n1]
  },
}
```

Vapor SFC 编译 compilation

```
<script setup vapor>
import { ref } from 'vue'

const count = ref(0)

function increase() {
  count.value++
}

</script>

<template>
  <h1>Count: {{ count }}</h1>
  <button @click="increase">Increase</button>
</template>
```

```
import { delegate, delegateEvents, renderEffect, setText,
        template } from 'vue/vapor'
import { ref } from 'vue'

const t0 = template('<h1>')
const t1 = template('<button>Increase')
delegateEvents('click')

export default {
  setup() {
    const count = ref(0)
    function increase() {
      count.value++
    }

    const n0 = t0()
    const n1 = t1()
    delegate(n1, 'click', () => increase)
    renderEffect(() => setText(n0, 'Count: ',
                                count.value))

    return [n0, n1]
  },
}
```

Vapor SFC 编译 compilation

```
<script setup vapor>
import { ref } from 'vue'

const count = ref(0)

function increase() {
  count.value++
}

</script>

<template>
  <h1>Count: {{ count }}</h1>
  <button @click="increase">Increase</button>
</template>
```

```
import { delegate, delegateEvents, renderEffect, setText,
        template } from 'vue/vapor'
import { ref } from 'vue'

const t0 = template('<h1>')
const t1 = template('<button>Increase')
delegateEvents('click')

export default {
  setup() {
    const count = ref(0)
    function increase() {
      count.value++
    }

    const n0 = t0()
    const n1 = t1()
    delegate(n1, 'click', () => increase)
    renderEffect(() => setText(n0, 'Count: ',
                                count.value))
    return [n0, n1]
  }
}
```

bundle size
包体积

* 比较 Vapor 与 vDOM 模式的核心

bundle size

包体积

33.6%

※ 比 vDOM 模式减少

* 比较 Vapor 与 vDOM 模式的核心

performance

性能

performance

性能

performance

性能

Name Duration for...	vanillajs-lite	solid-v1.8.15	svelte-v5.0.0-next.105	vue-vapor-v3.2024.0-761f785	vue-v3.4.23	react-jotai-v17.0.1 + 1.7.2
Implementation notes	772					
Implementation link	code	code	code	code	code	code
create rows creating 1,000 rows. (5 warmup runs).	35.8 ± 0.3 (1.00)	36.9 ± 0.2 (1.03)	37.4 ± 0.2 (1.04)	40.7 ± 0.3 (1.14)	45.0 ± 0.8 (1.26)	52.1 ± 0.4 (1.46)
replace all rows updating all 1,000 rows. (5 warmup runs).	39.2 ± 0.2 (1.00)	43.3 ± 0.2 (1.10)	43.4 ± 0.2 (1.11)	45.8 ± 0.4 (1.17)	52.6 ± 0.2 (1.34)	57.0 ± 0.3 (1.45)
partial update updating every 10th row for 1,000 row. (3 warmup runs). 4 x CPU slowdown.	16.5 ± 0.3 (1.04)	16.9 ± 0.4 (1.06)	17.4 ± 0.4 (1.09)	21.1 ± 0.3 (1.33)	21.1 ± 0.5 (1.33)	22.2 ± 0.5 (1.40)
select row highlighting a selected row. (5 warmup runs). 4 x CPU slowdown.	2.8 ± 0.1 (1.04)	3.3 ± 0.3 (1.22)	4.3 ± 0.1 (1.59)	7.7 ± 0.4 (2.85)	5.0 ± 0.2 (1.85)	5.1 ± 0.1 (1.89)

🤔 怎么变快? faster

怎么搞?



怎么搞?

- 打磨渲染逻辑



怎么搞?

- 打磨渲染逻辑
- 丢掉包袱，重装上阵！

破坏性变动

breaking changes

~~破坏性变动~~



No!

子集 subset

Vue Vapor 是 Vue vDOM 模式的子集。

子集 subset

Vue Vapor 是 Vue vDOM 模式的子集。



vDOM

- Composition API
- Options API
- 自定义指令
- Mixins...

子集 subset

Vue Vapor 是 Vue vDOM 模式的子集。

- 仅支持 Composition API
 - Options API 或许会作为第三方库?



子集 subset

Vue Vapor 是 Vue vDOM 模式的子集。

- 仅支持 Composition API
 - Options API 或许会作为第三方库?
- 仅构建工具编译 (Vite / Webpack...)



子集 subset

Vue Vapor 是 Vue vDOM 模式的子集。

- 仅支持 Composition API
 - Options API 或许会作为第三方库?
- 仅构建工具编译 (Vite / Webpack...)
- 仅支持 `<script setup>`



兼容性 compatibility

兼容性 compatibility

-  VueUse

兼容性 compatibility

-  VueUse
-  VitePress

兼容性 compatibility

-  VueUse
-  VitePress
-  Vue Router / Pinia

兼容性 compatibility

-  VueUse
-  VitePress
-  Vue Router / Pinia
-  Nuxt

兼容性 compatibility

-  VueUse
-  VitePress
-  Vue Router / Pinia
-  Nuxt
- UI 库
 - Naive UI / Element Plus / Vuetify...

兼容性 compatibility

-  VueUse
-  VitePress
-  Vue Router / Pinia
-  Nuxt
- UI 库
 -  Naive UI / Element Plus / Vuetify...
-  Elk

兼容性 compatibility

-  VueUse
-  VitePress
-  Vue Router / Pinia
-  Nuxt
- UI 库
 -  Naive UI / Element Plus / Vuetify...
-  Elk
- Vue 2

兼容性 compatibility

-  VueUse
-  VitePress
-  Vue Router / Pinia
-  Nuxt
- UI 库
 -  Naive UI / Element Plus / Vuetify...
-  Elk
- ~~Vue 2~~

进度 progress

- SFC 编译
- 基础组件能力
 - 内置指令
 - 生命周期钩子
 - ...
- 投票 App

进度 progress

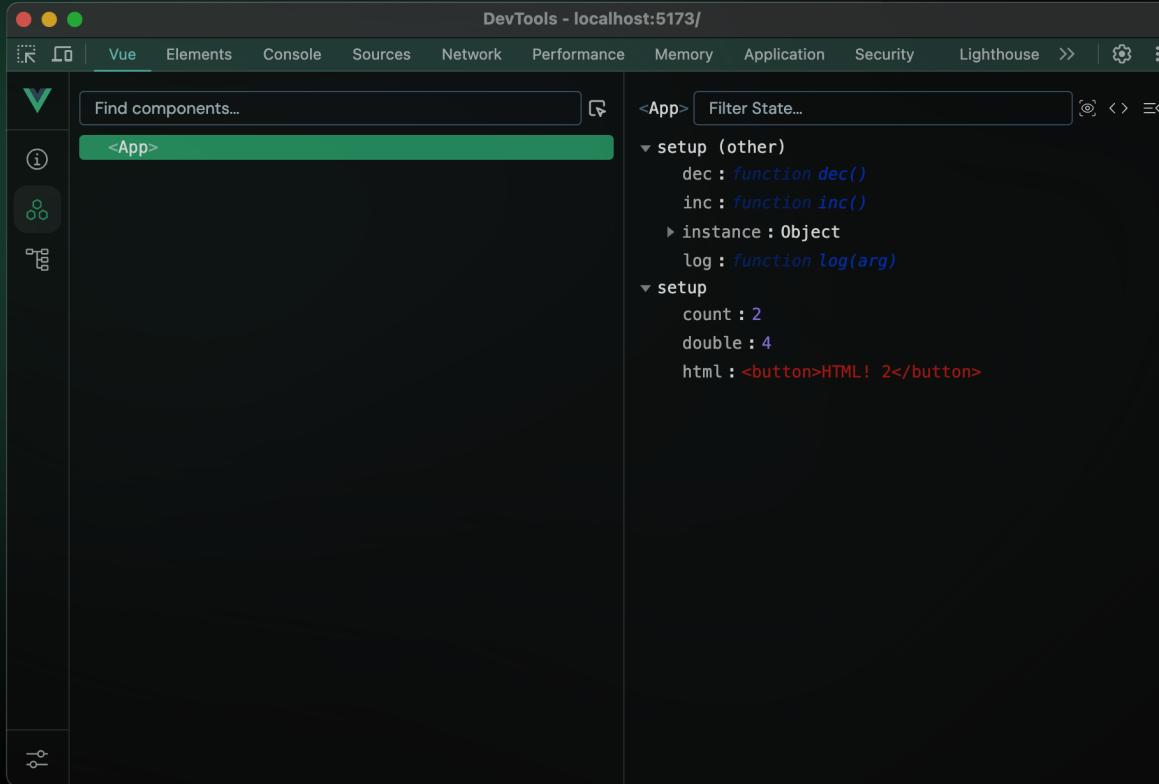
- SFC 编译
- 基础组件能力
 - 内置指令
 - 生命周期钩子
 - ...
- 投票 App
- 性能调优！性能调优！性能调优！
- SSR / 水合 (hydration)
- DevTools
- 异步组件
- 内置组件

进度 progress

- SFC 编译
- 基础组件能力
 - 内置指令
 - 生命周期钩子
 - ...
- 投票 App
- 性能调优！性能调优！性能调优！
- SSR / 水合 (hydration)
- DevTools
- 异步组件
- 内置组件
- VitePress 将会在今年底支持，并发布 beta 版

DevTools

感谢 Vue DevTools 团队  webfansplz



支持 JSX ?

支持 JSX !

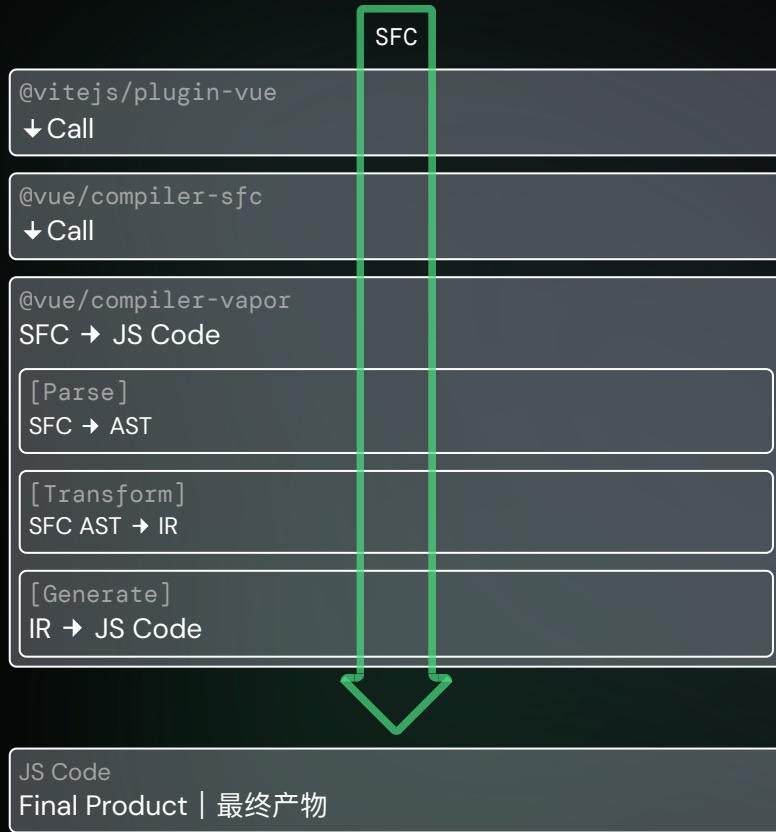
支持 JSX !

unplugin-vue-jsx-vapor@1.0.0

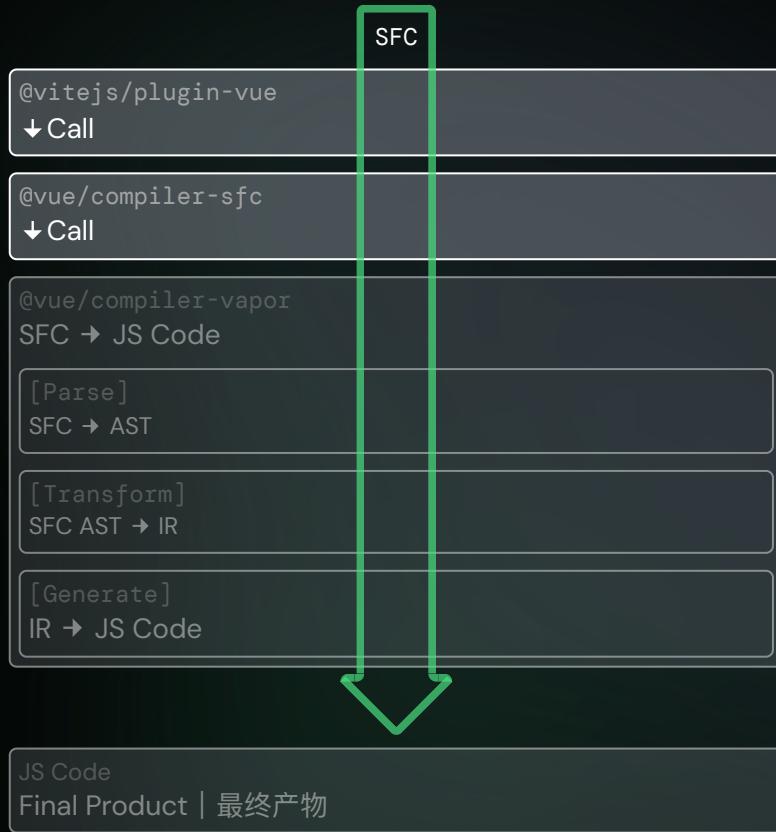
感谢  zhiyuanzmj

架构 architecture

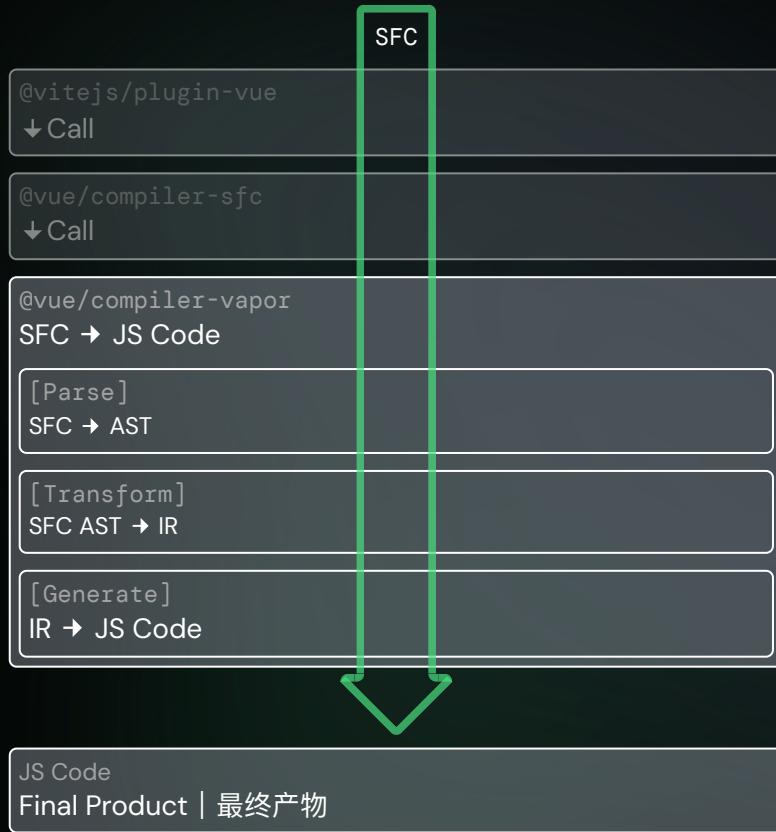
架构 architecture



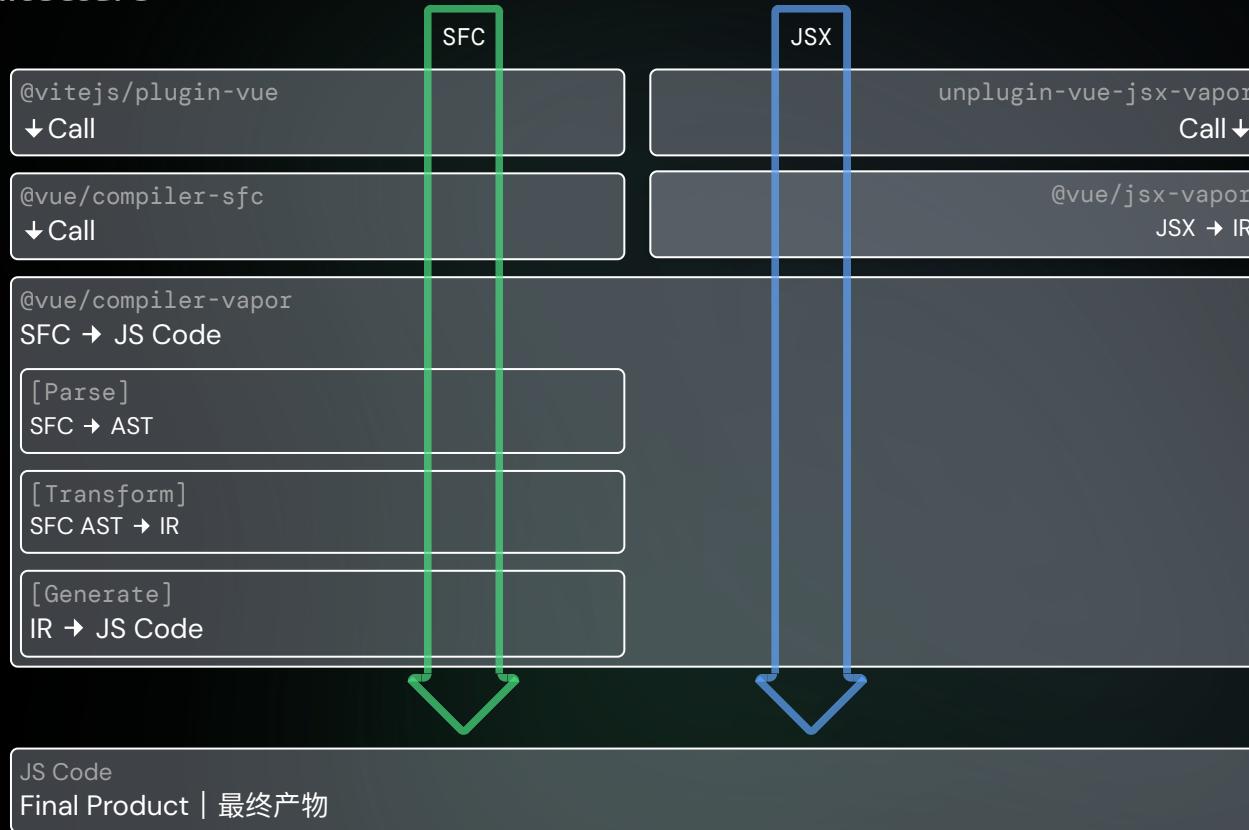
架构 architecture



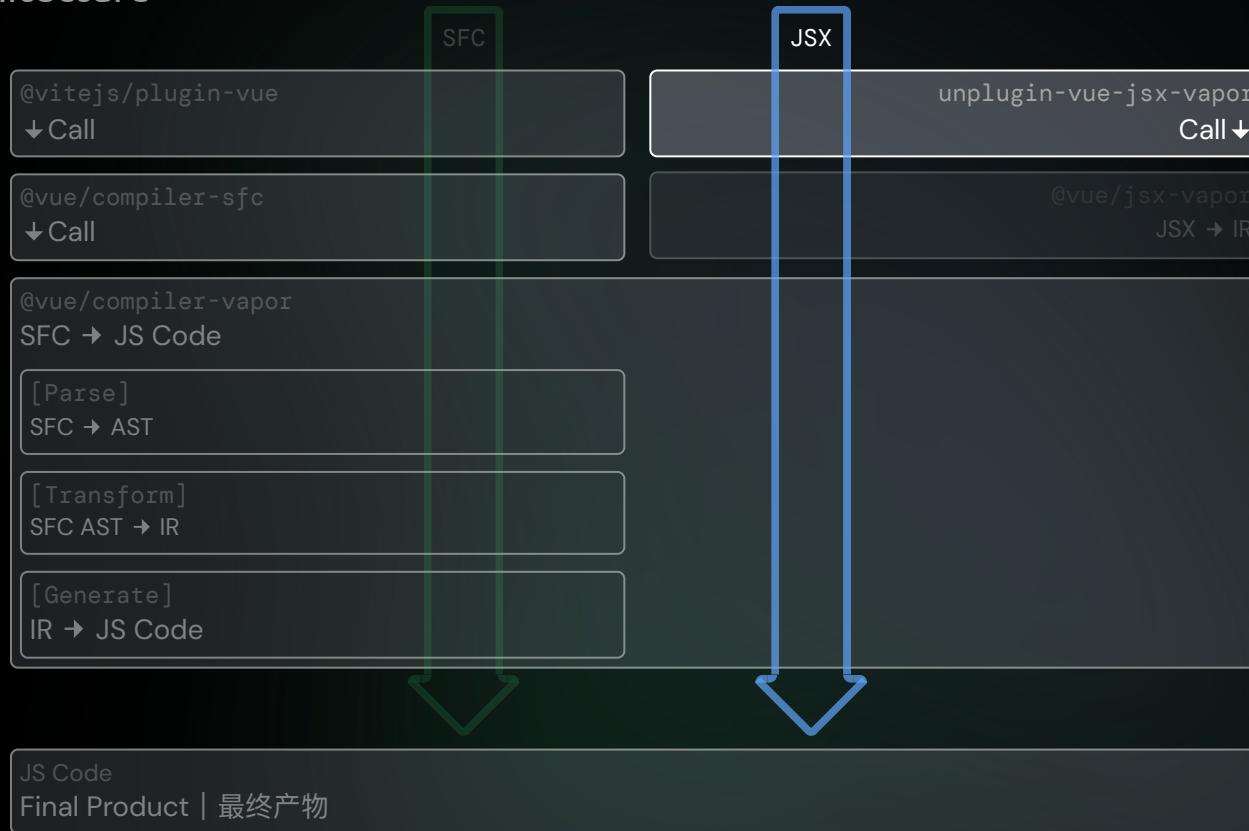
架构 architecture



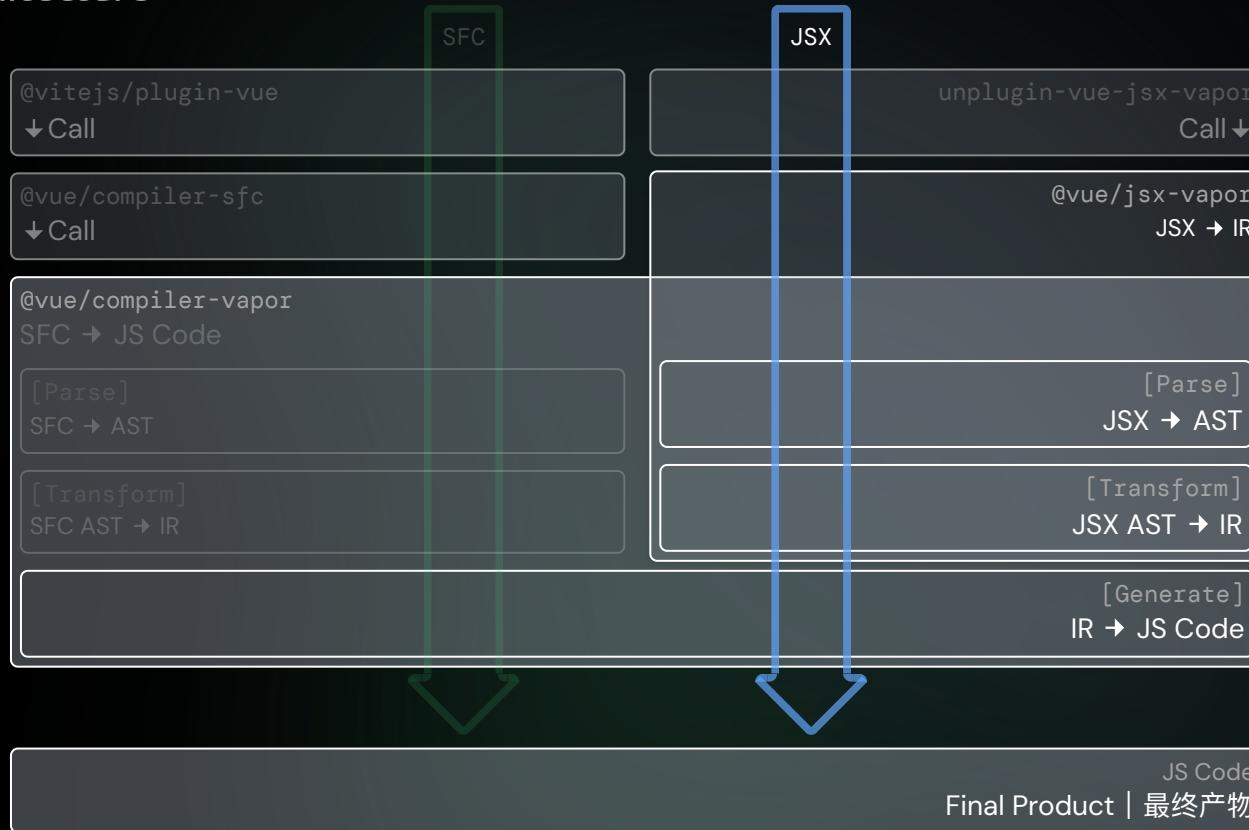
架构 architecture



架构 architecture



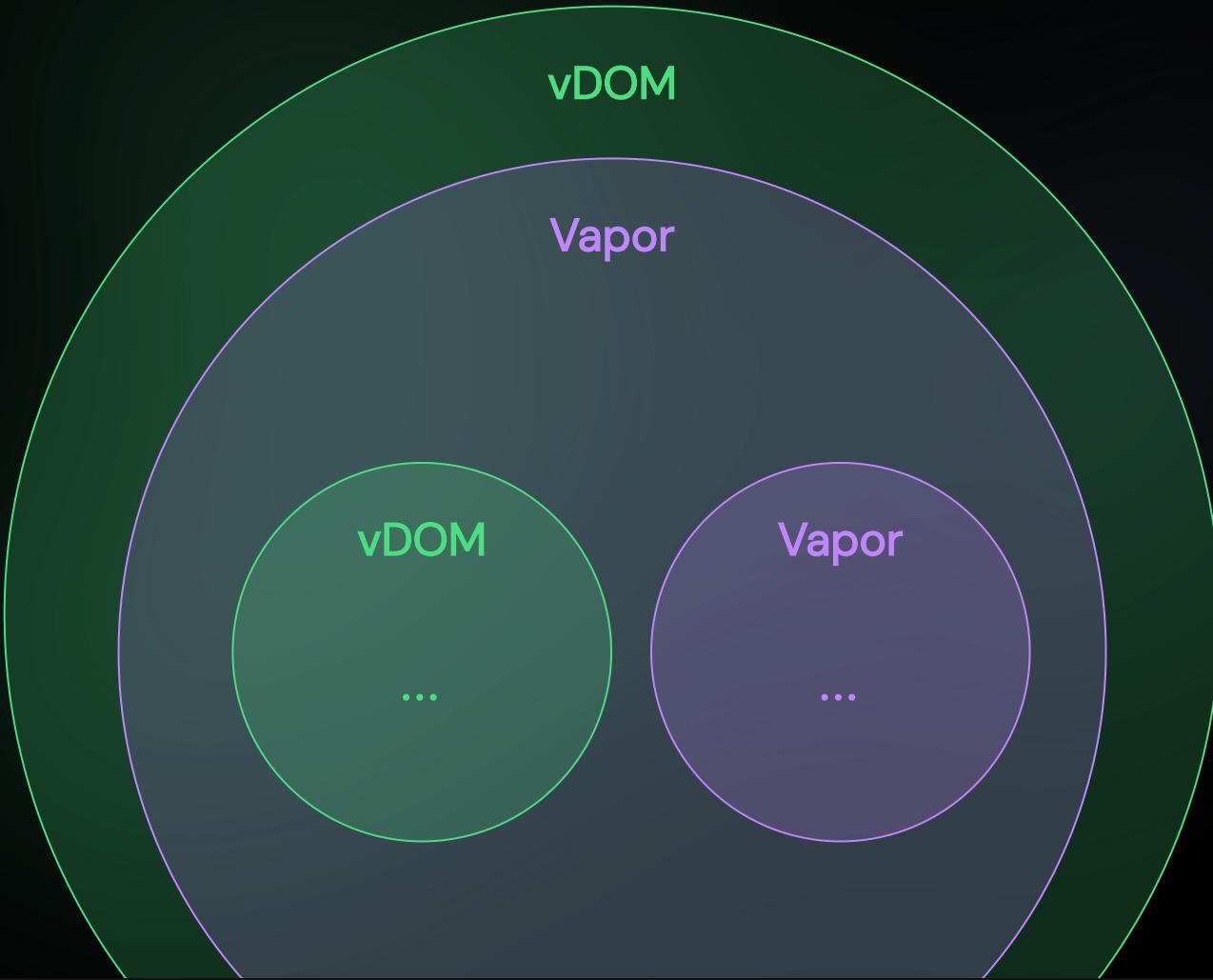
架构 architecture



融合 fusion

融合 fusion

- Vapor 组件支持 vDOM 模式
- vDOM 组件将支持 Vapor 模式
- 支持纯 Vapor 模式



功能规划 feature plan

功能规划 feature plan

RFC 意见征求稿

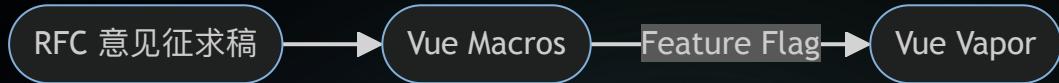
1. 提出 RFC 意见征求稿

功能规划 feature plan



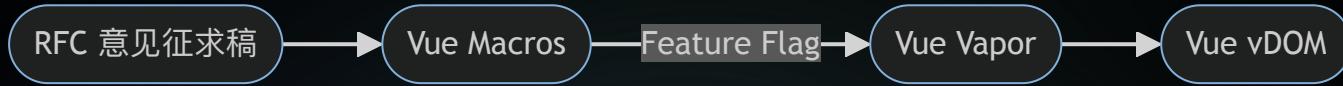
1. 提出 RFC 意见征求稿
2. Vue Macros 第三方项目先试验

功能规划 feature plan



1. 提出 RFC 意见征求稿
2. Vue Macros 第三方项目先试验
3. 以 Feature Flag 的形式引入 Vue Vapor

功能规划 feature plan



1. 提出 RFC 意见征求稿
2. Vue Macros 第三方项目先试验
3. 以 Feature Flag 的形式引入 Vue Vapor
4. 移植到 Vue vDOM

总结 recap

总结 recap

- Vapor 是 Vue 的一个新的开始

总结 recap

- Vapor 是 Vue 的一个新的开始
- Vapor 是 Vue vDOM 模式的子集，尝试做减法

总结 recap

- Vapor 是 Vue 的一个新的开始
- Vapor 是 Vue vDOM 模式的子集，尝试做减法
- Vapor 将持续优化性能和包体积

总结 recap

- Vapor 是 Vue 的一个新的开始
- Vapor 是 Vue vDOM 模式的子集，尝试做减法
- Vapor 将持续优化性能和包体积
- Vapor 将有更好的 JSX 支持

总结 recap

- Vapor 是 Vue 的一个新的开始
- Vapor 是 Vue vDOM 模式的子集，尝试做减法
- Vapor 将持续优化性能和包体积
- Vapor 将有更好的 JSX 支持
- Vapor 将在今年底发布 beta 版，并支持  VitePress

contributors

贡献者

 github.com/vuejs/core-vapor

contributors

贡献者

⌚ github.com/vuejs/core-vapor



Vue 作者

Evan You

contributors

贡献者

github.com/vuejs/core-vapor



Vue 作者

Evan You



Vue Vapor 作者

Kevin Deng

contributors

贡献者

 github.com/vuejs/core-vapor



Vue 作者

Evan You



Vue Vapor 作者

Kevin Deng



活跃开发者

Rizumu Ayaka



活跃开发者

UbugeeeI



活跃开发者

Doctor Wu

contributors

贡献者

 github.com/vuejs/core-vapor



Vue 作者

Evan You



Vue Vapor 作者

Kevin Deng



活跃开发者

Rizumu Ayaka



活跃开发者

UbugeeeI



活跃开发者

Doctor Wu



Vapor JSX 作者

高飞

sponsors

赞助

♥ github.com/sponsors/sxzz



sponsors

赞助

♥ github.com/sponsors/sxzz



感谢

幻灯片可稍后在  github.com/sxzz/talks 浏览

 感谢 Anthony Fu, 幻灯片由  Sliderv 强力驱动!

Q & A