



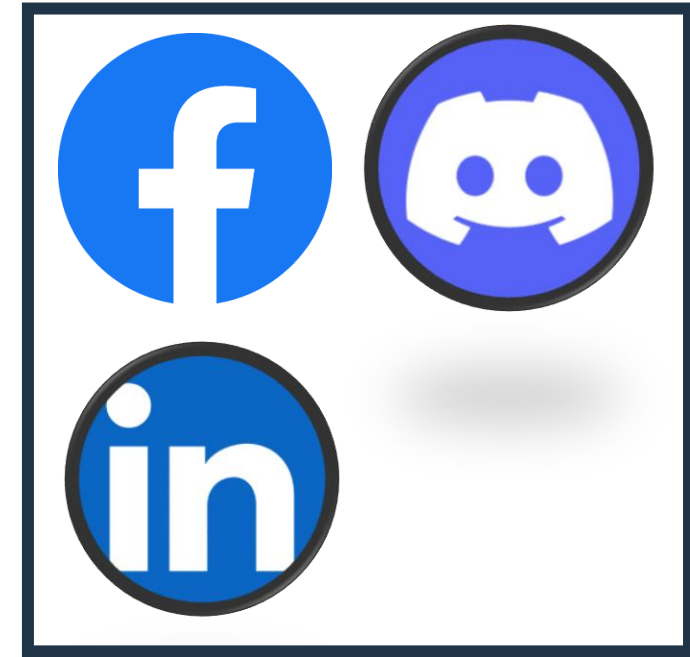
# Vue

```
<script setup>
import { onMounted } from 'vue'

onMounted(() => {
  console.log("Benja Vicente")
  console.log("Jose Antonio Castro")
  console.log("José Madriaza")
})
</script>
```

**Temática: Cercanía en Red Social**

# Redes Sociales con seguimiento bidireccional



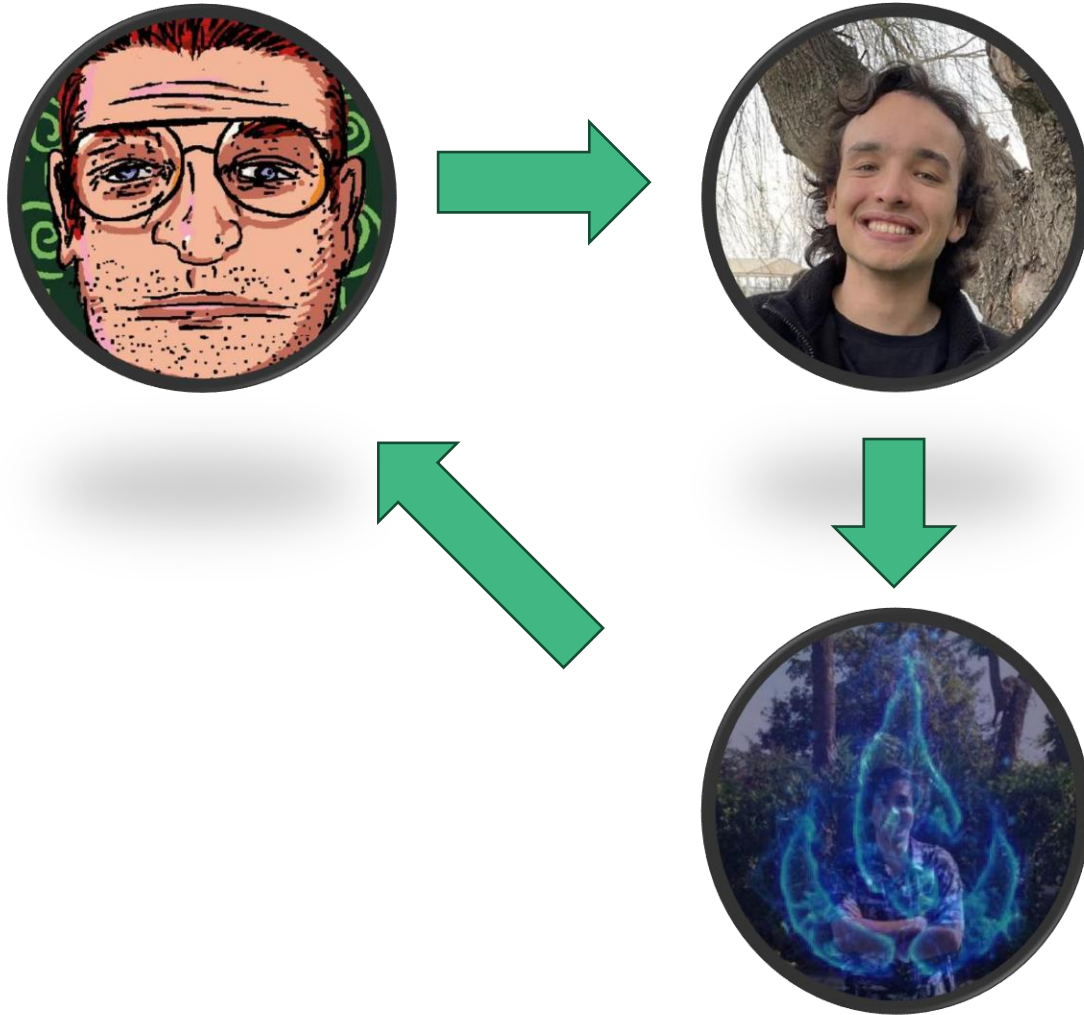
"Amigo"

# Redes Sociales con seguimiento unidireccional

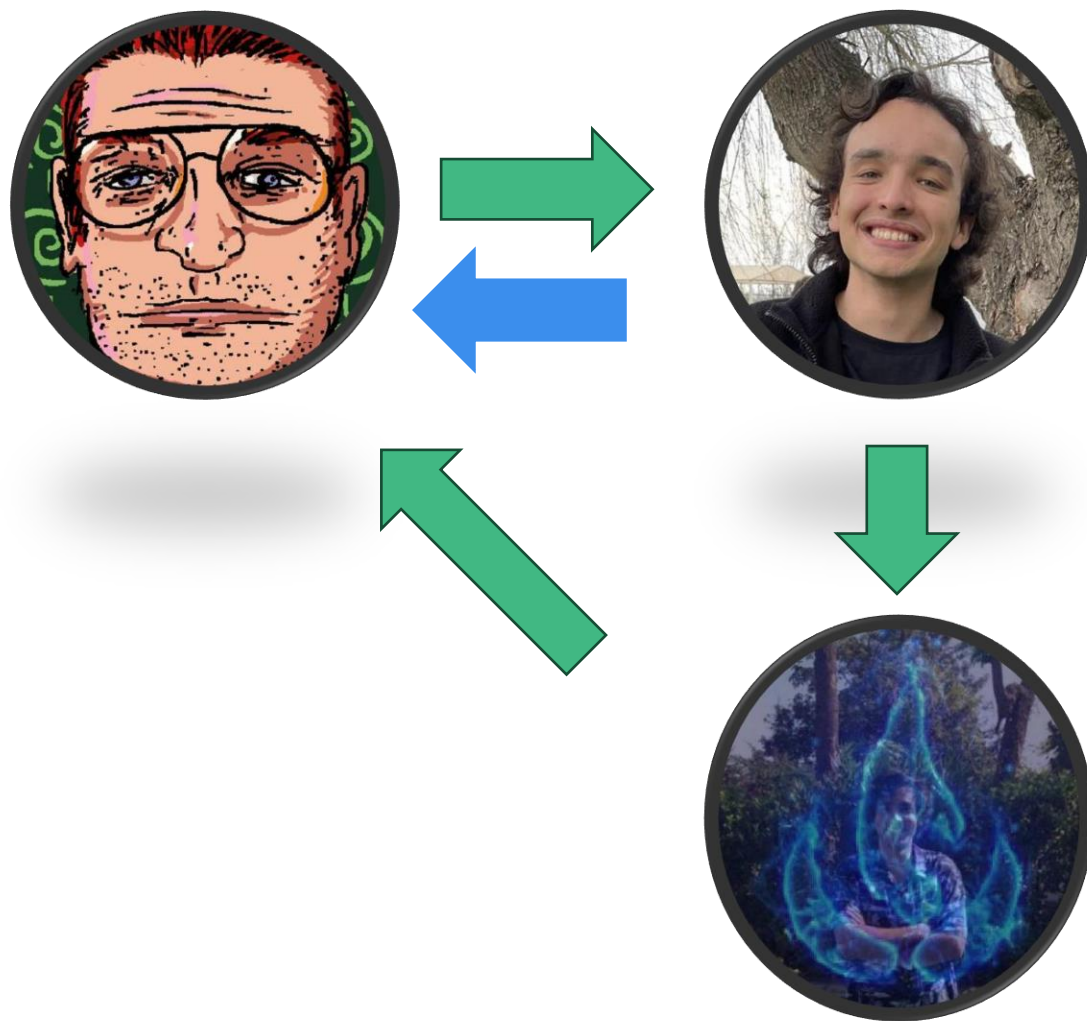


"Seguidor"

# Red de seguimiento unidireccional



# Red de seguimiento unidireccional



# Demo



<https://missing-github-followers.vercel.app/>

Missing followers

LOG IN

# GitHub Missing Followers: 0

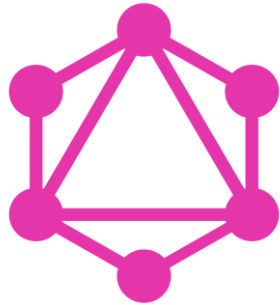


+1 TO COUNTER

You can now check your missing followers and following on GitHub

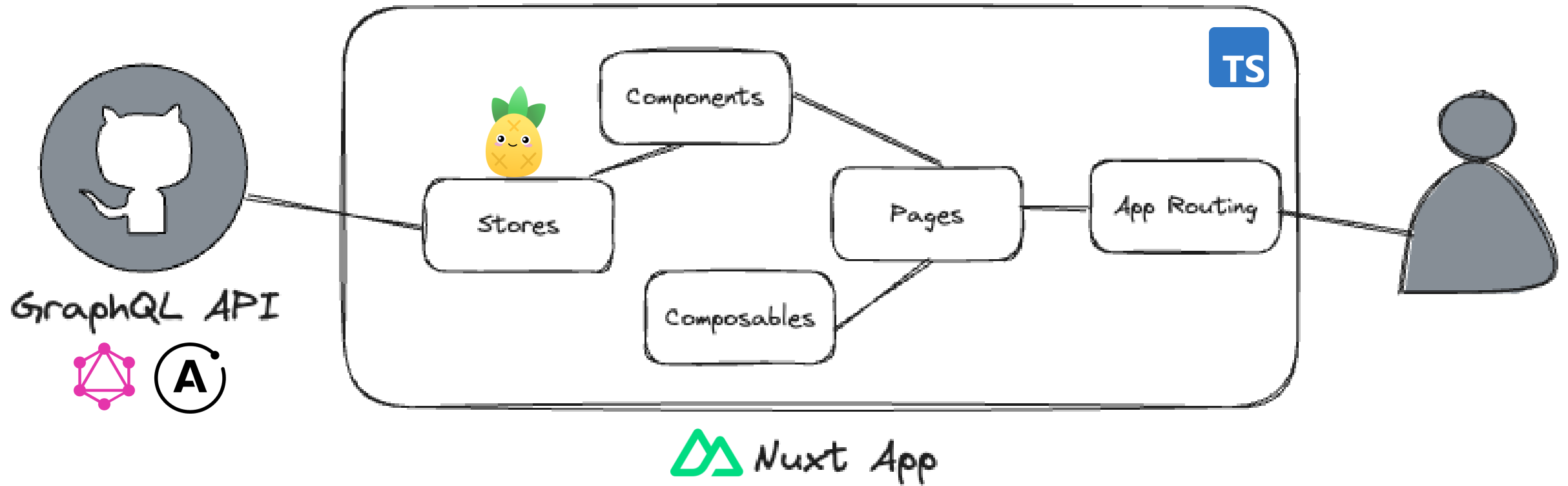


# Arquitectura de la App



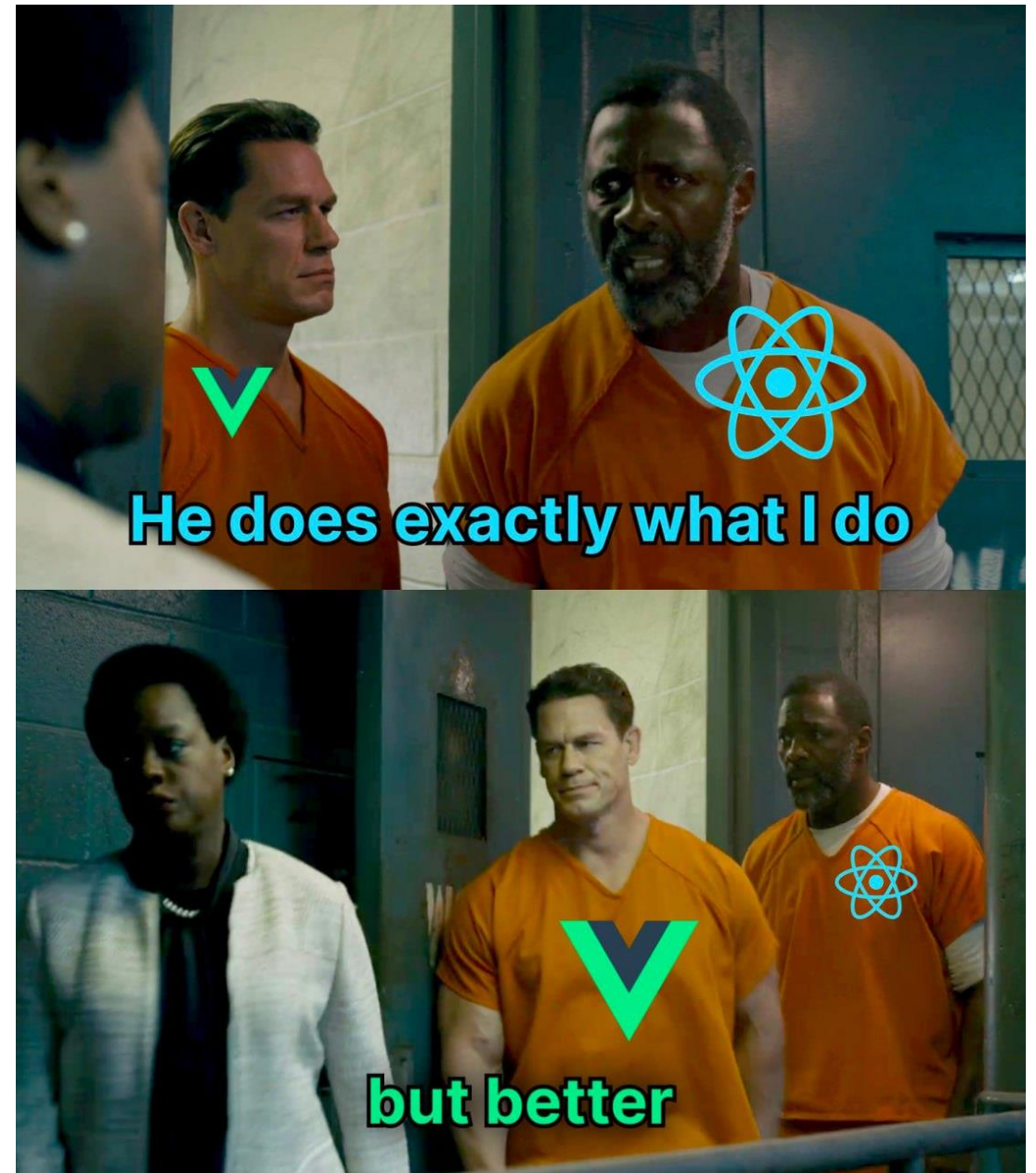


# Arquitectura de la App



# Implementación con Vue

# ¿Qué hace Vue?



# Reactividad en Vue

## Runtime-based

```
<script setup lang="ts">
  // Store
  const currentUserStore = useCurrentUserStore()
  // Composables
  const { isLoggedIn, token } = useAuth()
  // Computed properties
  const currentUserFollowing = computed(() => currentUserStore.following)
  const currentUserFollowers = computed(() => currentUserStore.followers)
  const loadingUserInfo = computed(() => currentUserStore.loadingUserInfo)
  // Refs
  const count = ref<number>(0)
  // Methods
  const increment = () => {
    count.value++
  }
</script>
```

# Reactividad en Vue

Runtime-based con proxies basado en un WeakMap global

```
function reactive(obj) {  
  return new Proxy(obj, {  
    get(target, key) {  
      track(target, key)  
      return target[key]  
    },  
    set(target, key, value) {  
      target[key] = value  
      trigger(target, key)  
    }  
  })  
}
```

```
function ref(value) {  
  const refObject = {  
    get value() {  
      track(refObject, 'value')  
      return value  
    },  
    set value(newValue) {  
      value = newValue  
      trigger(refObject, 'value')  
    }  
  }  
  return refObject  
}
```

```
<script setup>
import { reactive } from 'vue'

const state = reactive({ count: 0 })

function increment() {
  state.count++
}
</script>

<template>
  <button @click="increment">
    {{ state.count }}
  </button>
</template>
```

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

const count = ref(0)

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

<template>
  <button @click="increment">
    {{ count }}
  </button>
</template>
```

# Template Syntax

```
return (  
  <div>  
    {!isAuthenticated ? (  
      ...  
      <button onClick={() => login(user)}>Log in</button>  
    ) : (  
      ...  
      <button onClick={logout}>Log out</button>  
    )}  
  </div>  
) ;
```



```
<template>  
  <div v-if="!isAuthenticated">  
    ...  
    <button @click="login(user)">Log In</button>  
  </div>  
  <div v-else>  
    ...  
    <button @click="logout">Log Out</button>  
  </div>  
</template>
```





# Template Syntax: Directives

**v-on:submit.prevent="onSubmit"**

## Name

Starts with v-  
May be omitted when  
using shorthands

## Argument

Follows the colon  
or shorthand  
symbol

## Modifiers

Denoted by the  
leading dot

## Value

Interpreted as  
JavaScript expressions

# Render functions

`v-if`

Template:

```
<div>  
  <div v-if="ok">yes</div>  
  <span v-else>no</span>  
</div>
```

template

Equivalent render function / JSX:

```
h('div', [ok.value ? h('div', 'yes') : h('span', 'no')])
```

js

```
<div>{ok.value ? <div>yes</div> : <span>no</span>}</div>
```

jsx

# Single File Components

- JS, HTML y CSS
- Lo que cambia a la vez está en el mismo lugar
- Débilmente acoplado



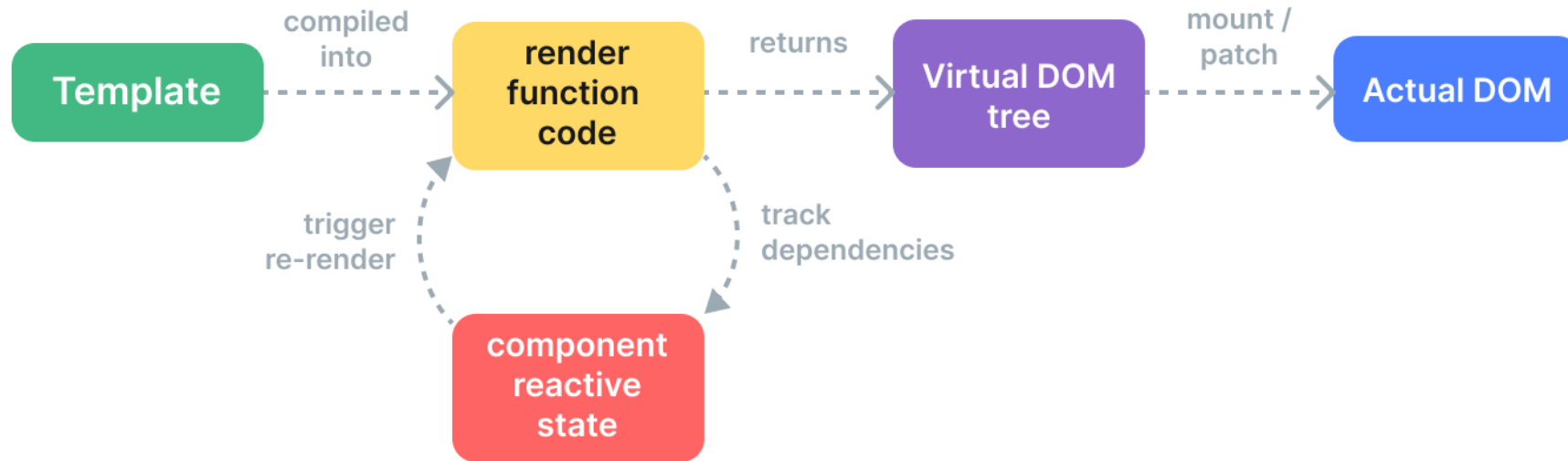
```
<script setup lang='ts'>
import { ref } from 'vue'
const greeting = ref<string>('Hello World!')
</script>

<template>
  <p class='greeting'>{{ greeting }}</p>
</template>

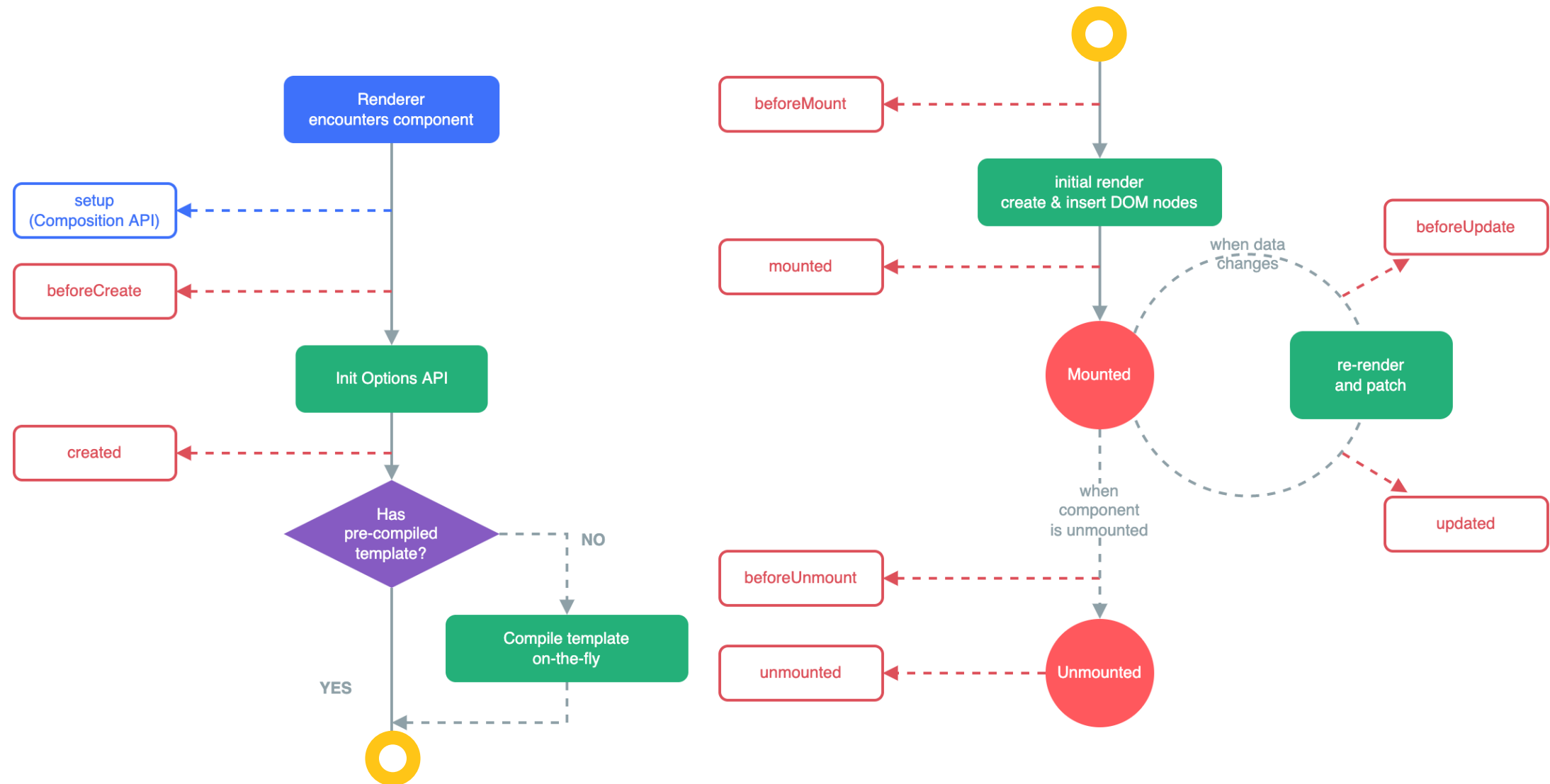
<style scoped>
.greeting {
  color: red;
  font-weight: bold;
}
</style>
```

# Mecanismo de renderizado

- Virtual DOM con paso de hidratación en SSR
- Compilación del template permite obtener optimizaciones



# Mecanismo de renderizado



# Conclusiones

- El estado en Vue es más intuitivo
- Template syntax es más limpio y óptimo que JSX
- Existe un gran ecosistema y comunidad
- Poco acoplamiento gracias a los SFC