



# 围绕 Vite 的前端统一工具链

尤雨溪  
D2 2025

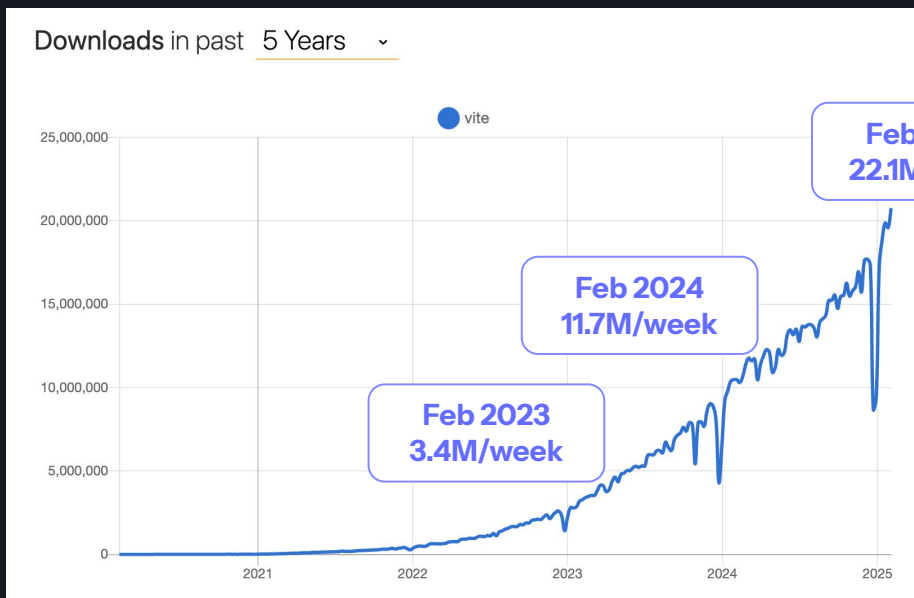
# Vite 的一点历史

- 2020 年作为实验性质的原型开始开发
  - 一开始只是为了 Vue 的用例设计
- 2021 年 Vite 2.0 转型成通用构建工具
- 2022 年开始被新兴框架广泛采用为底层构建工具
  - Nuxt 3, SvelteKit, Astro, SolidStart
- 2023: React 元框架也开始迁移到 Vite: Redwood, Remix
- 2024: 更多基于 Vite 的方案: TanStack Start, One, Ember
- 2025: create-react-app 被官方废弃后的事实继承者

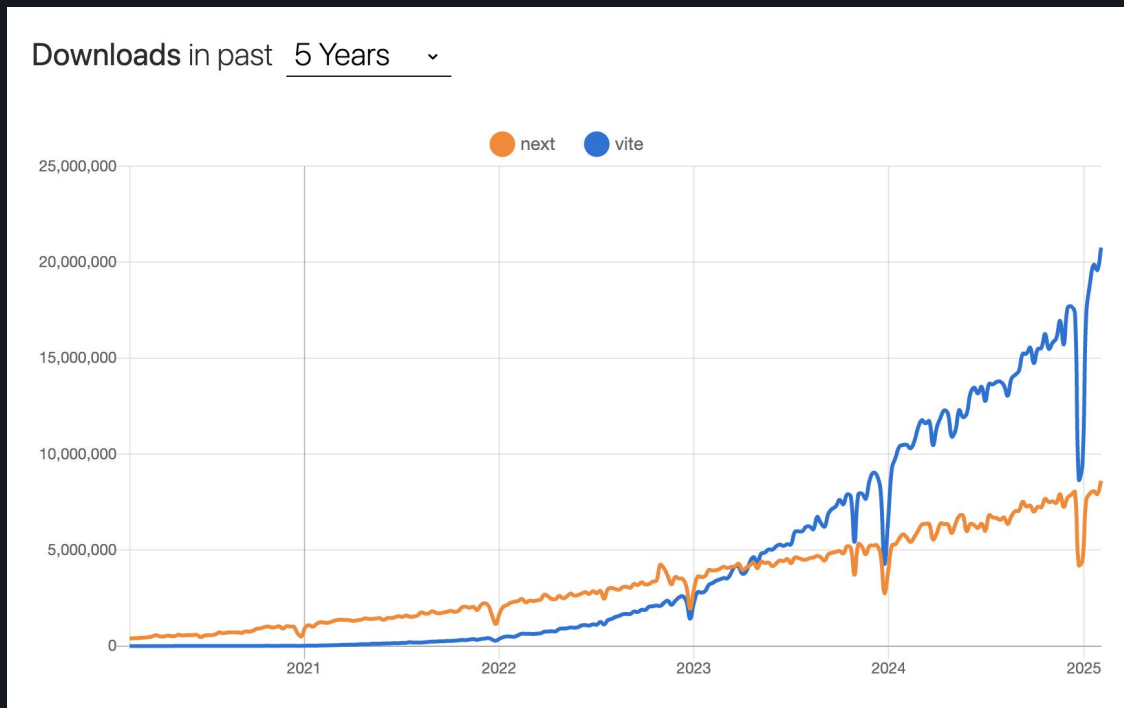
# Vite 的现状



npm 周下载量 2200+ 万次



## 和 Next.js 下载量的比较



# State of JS 2024 开发者问卷调查

<https://2024.stateofjs.com/en-US/awards/>



## Vite

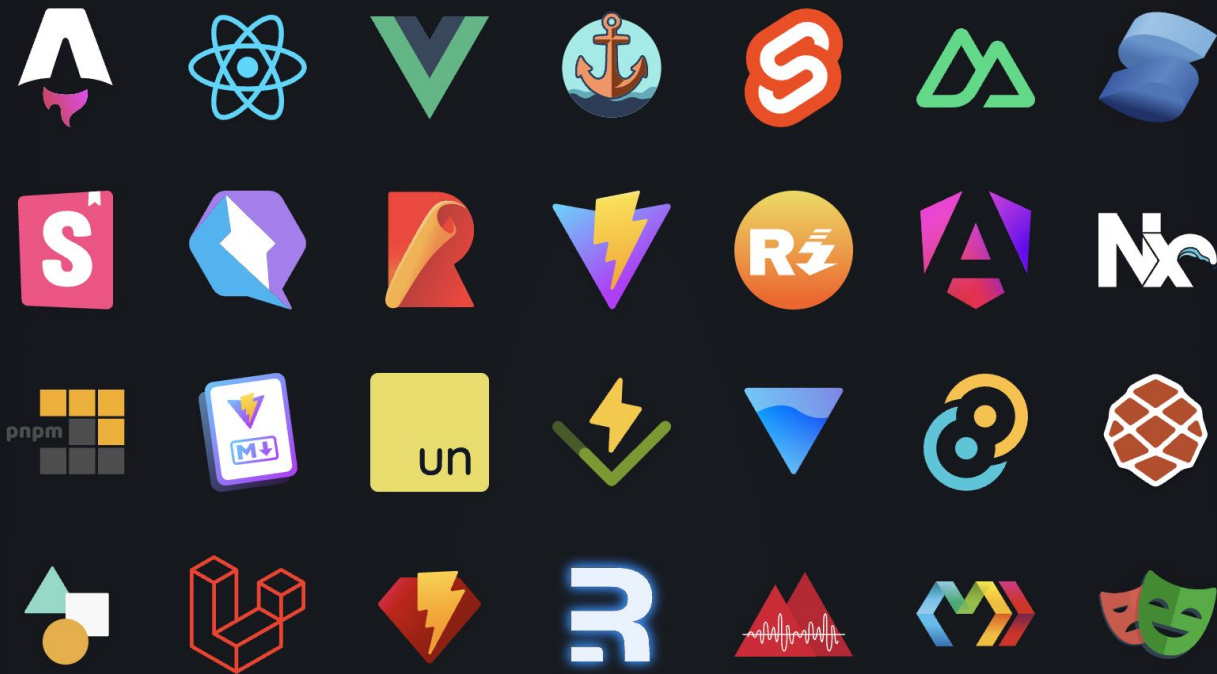
- No.1 增长 (+30%)
- No.2 满意度 (98%)
- No.3 感兴趣程度
- 全类别最受欢迎的库/工具



## Vitest

- No.2 增长 (+26%)
- No.1 满意度 (98%)
- No.2 感兴趣程度

# 庞大, 活跃, 多元化的生态



## 在生产环境中使用 Vite 或基于 Vite 的框架的公司



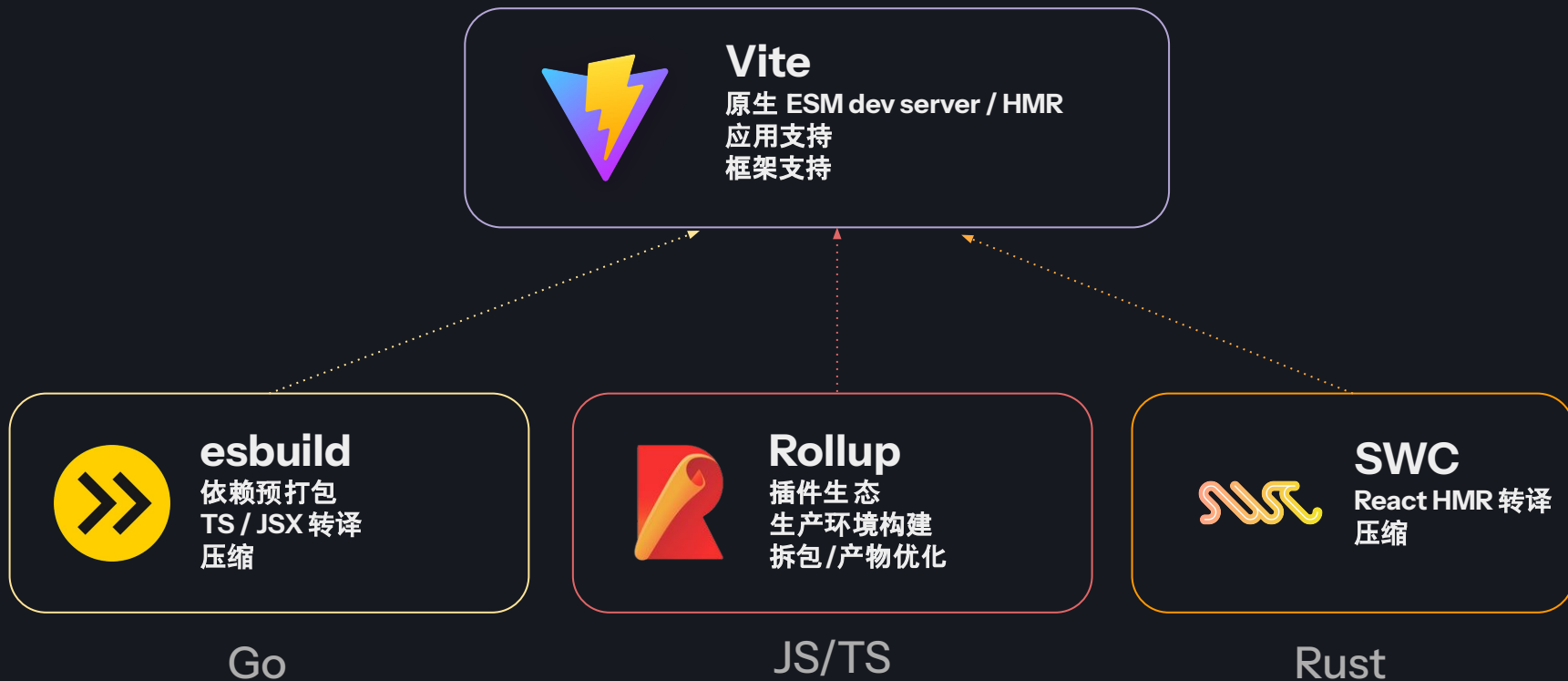
[...and many more](#)



Vite 已经成为了  
下一代 web 应用的  
共享基建层

**但, Vite 还并不完美。**

## Vite 底层依然依赖多个职责重复的第三方工具





## Vite

原生 ESM dev server / HMR  
应用支持  
框架支持



## esbuild

依赖预打包  
TS / JSX 转译  
压缩



## Rollup

插件生态  
生产环境构建  
拆包/产物优化



## SWC

React HMR 转译  
压缩

- 产物控制太弱(尤其是拆包)
- 插件系统局限性大
- 没有 HMR, 也不支持 React fast refresh 转译



## Vite

Native ESM dev server / HMR  
Application support  
Framework support



## esbuild

依赖预打包  
TS / JSX 转译  
压缩



## Rollup

插件生态  
生产环境构建  
拆包/产物优化



## SWC

React HMR 转译  
压缩

- 比起原生语言写的 bundler 来说太慢了



## Vite

Native ESM dev server / HMR  
Application support  
Framework support



## esbuild

依赖预打包  
TS / JSX 转译  
压缩



## Rollup

插件生态  
生产环境构建  
拆包 / 产物优化



## SWC

React HMR 转译  
压缩

- Binary 太重
- Bundler 不可用



## Vite

Native ESM dev server / HMR  
Application support  
Framework support



## esbuild

依赖预打包  
TS / JSX 转译  
压缩



## Rollup

插件生态  
生产环境构建  
拆包/产物优化



## SWC

React HMR 转译  
压缩

打包行为的微妙不一致  
(主要是 ESM/CJS interop 的处理)



## Vite

Native ESM dev server / HMR  
Application support  
Framework support



## esbuild

依赖预打包  
TS / JSX 转译  
压缩



## Rollup

插件生态  
生产环境构建  
拆包 / 产物优化



## SWC

React HMR 转译  
压缩

效率问题

多次重复 parse / transform / codegen  
且在 js / native 之间传递字符串



## Unbundled ESM 在请求数量巨大 时的性能瓶颈



## 总结一下: Vite 当前架构的问题

- 职责重叠的依赖
- 生产构建速度
- Unbundled ESM 在大型项目中的性能瓶颈
- 高阶功能, 比如极致拆包, 增量构建, **Module Federation**

于是, 我们开始开发 Rolldown



# Rolldown

- 兼容 Rollup 的插件 API 和生态
- Rust 提升生产构建性能(对标 esbuild)
- 添加 Rollup / esbuild 不具备的高阶功能(对标 webpack)
- Full bundle mode 解决 Vite unbundled ESM 的性能瓶颈

再挖深一点

**Vite 面对的问题其实是  
整个 JS 工具生态一直存在的问题的缩影**

# JS 工具生态在每一个层面 都非常的碎片化

- **Parsers:** babel / acorn / esbuild / espree / flow / TS / swc / uglify-js
- **Transformer:** babel / esbuild / swc / sucrose / TS
- **Test runner:** mocha / jasmine / tape / ava / jest
- **Linters:** jshint / eslint / biome
- **Formatter:** prettier / eslint
- **Bundlers:** webpack / rollup / esbuild / parcel
- **Monorepo tools:** nx / turborepo / rush / lerna / lage / wireit

# 碎片化导致的种种问题

- 选择困难 + 复杂度爆炸
- 一些工具互相根本不兼容
- 不同工具反复 **parse** / 编译导致的浪费
- 重复却又各不相同的配置
- 转译/模块解析等行为不一致
- 大量时间浪费在搭配和调试工具链上



想解决这些问题，  
我们需要一个统一的工具链。

## 为什么是现在？

1. 生态中各个层面的需求在过去十年中已经逐渐清晰，只是方案难以被统一。
2. 统一需要共识，而 Vite 展现出了 web 生态中前所未见的共识达成速度，提供了一个很有希望的基础。
3. Rust for JS 的生态地基有前人探路，也已经趋于成熟。

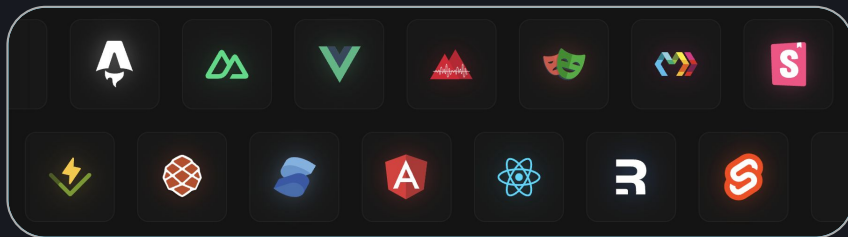
# void (∅)

下一代的 JS 统一工具链

[voidzero.dev](https://voidzero.dev)

# 架构一 览

上游框架 / 工具生态



void(0)

Vite Plus

一站式工具链解决方案

计划中



Vite

Dev server / HMR  
应用支持  
框架支持

生产



Vitest

Vite-native 测试工具  
Node.js + 浏览器

生产



Rolldown

打包工具

beta



Oxc

语言工具链

Composable NPM Packages & Rust Crates

Linters

Formatter

WIP

Transformer

Resolver

Minifier

beta

Parser

Semantic Analysis

# 为什么是 Rust?

- 极限性能和内存安全之间的平衡
- 已经有一个相对成熟的 for JS 开发者生态
  - 更容易找到贡献者
  - 成熟的 JS bindings (napi-rs)
  - 成熟的 Web Assembly 支持
- 注意: 我们并不打算锈化所有东西
  - CLI / API 等需要灵活迭代的东西还是用 TS
  - 需求明确稳定, 性能关键的部分用 Rust

## 目前的进展



## OXC 进度



### Parser

100% ECMA 262 spec compliance + TypeScript, JSX  
Semantic Analysis + Control Flow Graph



### Lint\*

400+ ESLint compatible rules  
IDE extensions / WIP linter plugin design in collab w/ Deno team



### Resolver

Node.js compatible CJS + ESM path resolution  
Fully customizable



### Transformer

TypeScript, React JSX, React Fast Refresh, isolatedDeclarations DTS,  
Syntax-lowering



### Minifier

Beta



### Formatter

Prototype



## OXC - 性能



### 最快的 parser

3x faster than SWC / 5x faster than Biome

[Benchmark](#)



### 最快的 linter

50~100x faster than ESLint and scales with number of CPU cores

[Benchmark](#)



### 最快的 resolver

28x faster than webpack/enhanced-resolve

[Benchmark](#)



### 最快的 transformer

TS / TSX: 4x faster than SWC / 40x faster than Babel

React Refresh: 6x faster than SWC / 70x faster than Babel

Isolated Declarations DTS Emit: 20~45x faster than tsc

[Benchmark](#)





# Oxc - Transformer Metrics



## 更少内存占用

Uses **24%** less memory than SWC and **70%** less than Babel

oxc-transform*	51MB
@swc/core*	67MB
@babel/core + presets	172MB



## 更小的 binary 文件大小

**95%** smaller than SWC / **90%** smaller than Babel

oxc-transform*	~2MB
@swc/core*	37.5MB
@babel/core + presets	21MB / 170 packages



Rolldown - 进度



## 基础打包

CJS + ESM interop / Source maps / Output formats  
Basic code splitting / CLI / Config file support



## 内置 OXC 转译 + 模块解析

Built-in transforms / Node resolver



## 高阶功能

Tree-shaking / define / inject  
高级拆包选项 / Module Federation



## 行为对齐

80%+ esbuild / Rollup 测试覆盖率  
90%+ Rollup 插件兼容



## 1.0 Beta

Rollup / esbuild 功能对齐



## 1.0 Beta

Rollup / esbuild 功能对齐



## Vite 整合 (unbundled)

兼容性 / 稳定性调整

Ecosystem CI (95%)



## Vite full bundle mode

HMR / Incremental build



Rolldown - 进度



## Rolldown - 性能

Benchmark 打包 19k 模块  
10k React JSX 组件 + 9k iconify JS 文件

rolldown	0.75s	<div></div>
esbuild	1.33s	<div></div>
rspack	3.08s	<div></div>
rollup	26.23s	<div></div>



## Rolldown - WASM 性能

[Benchmark](#) 打包 2.5k 模块  
环境: StackBlitz (web containers)

rolldown	0.61s	
rollup	4.52s	
esbuild	22.19s	

## 一些小细节

- OXC 围绕 arena allocator 做的内存管理优化
- Rolldown 基于 Rollup 插件 API 增加 filter, 避免 JS 调用开销
- Rolldown 在 JS 侧对插件 hooks 进行合并
- OXC 研发中的 0-cost AST transfer
- 基于 0-cost AST transfer 的轻量级 JS transform / lint API

**What's Next?**

# Vite Now (v6)



**Vite JavaScript core**

Native ESM Dev server

Module Graph

Plugin Container

Environment API

Dep Optimizer

Module Runner



**esbuild**

依赖预打包  
TS / JSX 转译  
压缩



**Rollup**

插件生态  
生产环境构建  
拆包 / 产物优化

via  
[@vitejs/plugin-react-swc](https://github.com/vitejs/plugin-react-swc)



**React  
HMR 转译  
压缩**



# Vite Next (Q1 2025)



**Vite JavaScript core**

Native ESM Dev server

Module Graph

Plugin Container

Environment API

Dep Optimizer

Module Runner



**Rollup**

依赖预打包  
生产构建  
拆包/产物优化



**Oxc**

模块解析  
TS / JSX 转译  
压缩

# Vite Future (Q2 2025)



**Vite JavaScript core**

▶ Bundled Dev server

Environment API / Module Runner ◀



**Rollup**

开发打包 + HMR  
生产打包 + 产物优化



**Oxc**

模块解析  
TS / JSX / Module runner 转译  
压缩

# Vite Plus

“cargo for JavaScript”

- Dev / Build (Vite / Rolldown)
- Test / Bench (Vitest)
- Lint / Format (OXC)
- 文档 (VitePress 多框架支持)
- Run TypeScript in Node.js (with tsconfig paths support)
- 项目脚手架 / 代码生成
- Monorepo 任务调度 / 缓存

**Coming in 2025.**

谢谢！  
[voidzero.dev](https://voidzero.dev)