Faster FE-stack in 2025

@Kode24-dagen 4.0

27th March

🧑‍💻 Bobby Westberg

🎙️ Faster FE-stack in 2025

Faster **stack** - we're **improving the DX** and **time-to-production**!

- Replacements for your standard tooling
  - 🔍 Focus on modern tools (**Rust** or **Go**)
  - 🕸️ What goes out? ❤️‍🔥 What comes in?
  - 💚 Pros & 💔 cons
  - 🔀 Experiences
- ⏬ Download-link at the end (✂️ Includes cut-outs!)

😮‍💨 Why replace?

- 🔨 **JavaScript** historically built our stack
  - with inherited limitations
- 🚧 **Rust/Go** "revolution"
  - (Rust, Go, Zig, C, C++)
  - not new, but **increased popularity** for webdev-tools
- 🛬 **JavaScript** and **CSS** evolution
  - browsers improve

Let’s rethink our habits!

🥸 But … why Rust/Go!?

This is **not** a talk **about** Rust/Go … but:

- **Compiles to native code**
- CPU-usage, threading, **memory**-control
- Communicates very good with JS
- And a lot more!
- = **Unbeatable speed**
- 🔫 PS: Not a silver bullet

Hi!

Bobby Westberg

- **Gjensidige**: 5 years *(and counting)*
  - Discipline Leader for front-end

How it began

- '94 `` `<FONT SIZE=2 COLOR=RED>` ``
- '97 ffuniverse.nu 🐤
- *Eventually: education and a job*
- Misc jobs, 11 as webdev
  - **Enonic** (open source): 7 years
  - Still their Swedish translator 🇸🇪

Gjensidige

- One of the **10 largest** on Oslo stock exchange
- *Cloud-platform*: **GAP** (gap.gjensidige.io) on **Azure**
- *Code, CI/CD*: **GitHub**
- *Communication*: **Slack** (us "Techies")
- *Front-end*: **React**
- *Back-end*: Java/Kotlin, ASP.NET, even COBOL

150
front-enders

600
in tech

4800+
employees

📦 Builders Core

- Gjensidige's design system
  - **Webapps** + **internal** + **mobile**
- 🧩 **React**-components
- 📚 Documentation with **Storybook**
- 🔒 Internal only *(but open docs)*
- 🎨 Designs in **Figma**

Builders Components

All | Actions | Containers | Forms | Inform | Navigate | Text

I am Action List
Link 1 →
Link 2 →
As button →

ActionList

Hello
We use the default alert in a task flow to notify users of the status of an action. They come in many forms.

Alert

AvtaleGiro

Badge

My name is Box. Just Box.
The box is just a box. You use it as a box. It's here to help you box stuff up, get a background colour, setting easy padding.
If you need a link

Box

Min oversikt / Forsikringer / Bilforsikring / Kasko

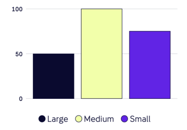
Breadcrumbs

I'm a button

Button

Card title
See all or change insurances

Card



Chart

Chevron

Copy

Clipboard

Label text

Combobox: Default

internal="small" (for internal applications only)
Option A

Combobox: Internal Small

internal="table-inline" (for internal applications only)

Heading 1	Heading 2
Content 1	O ×

Combobox: Internal Table Inline

Akkurat nå er det 20% rabatt
Insert relevant and important content which trigger interest in this text box. Try to keep it as short as possible.

CTA Banner

Datepicker

Datepicker: Default

Copy

Clipboard

Label text

Combobox: Default

internal="small" (for internal applications only)
Option A

Combobox: Internal Small

internal="table-inline" (for internal applications only)

Heading 1	Heading 2
Content 1	O ×

Combobox: Internal Table Inline

Akkurat nå er det 20% rabatt
Insert relevant and important content which trigger interest in this text box. Try to keep it as short as possible.

CTA Banner

Datepicker

Datepicker: Default

Month
Januar 2024

Datepicker: Type Month

internal="small" (for internal applications only)

Datepicker: Internal Small

Divider

Dropdown

Dropdown (deprecated)

Expandable Example

Expandable

Browse files

FileUploader

Filter

Filter (deprecated)

1 2 3 4 5 6

Flex

Label text
test

FormField

📊 Stats about Builders Core

- **78 000** lines of code
  - *(but, this includes a lot of `` `.mdx` ``)*
- Monorepo with **7 packages**
  - *(components, icons, fonts, tokens, ++)*
- Maintained by **Squad Design system** (my team ❤️)
  - Cross-disciplinary, **8 members**

42
components

400
webapps

2200+
web pages

🧑‍🔬 Gains for Builders Core

🧪 Tried all the talks' modernisations, **Builders Core** have:

- Fewer dependencies
- Faster installs
- Faster builds
- Faster testing
- Faster linting
- Smaller package sizes
- Easier-to-understand code
- Speed-ups in CI/CD

15min
2023

10 000
minutes

8-9min
2024

4-5min
2025

⚠️ Before we start

- Broad list of **varied suggestions**
  - New for someone, old for others - hopefully *something* for **you**
- Meta-frameworks (**Astro**, **Next.js**, **Remix**) not included
- **Builders Core**'s new speed is not *only* because of modern tools
  - **Vite**: replaced Rollup and Babel *(and almost 20 plugins)*
  - **Vitest** (and jsdom): replaced Jest and Cypress

Era of Rust & Go

Before we replace …

❤️‍🔥 TypeScript

Two weeks ago, Microsoft announced TypeScript Go-rewrite.

This will make it **10+ times faster**!

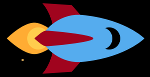
Maybe a beta for **summer 2025**.

husky, lint-staged

Interested in replacing two tools with one?

Husky, lint-staged

- Tooling for *git hooks*:
  - *Do extra stuff when you do something with git*
  - Example: on `` `git push` ``, run all tests
  - Example: on `` `git commit` ``, run prettier
- Husky is popular
  - But so is simple-git-hooks, and pre-commit
- To **only** lint staged files, Husky requires lint-staged
  - Prettier has pretty-quick
- Requires quite a lot for setup *(special folders and files, install script)*



❤️‍🔥 Lefthook

- By EvilMartians
- 💚 A lot faster!
- 💚 Can run commands in **parallel**
- 💚 Very good docs (and guides)
- 💚 More intuitive setup & config (`lefthook.yml`)
- 👶 Rather new in the git hooks-space
- 👻 Git hooks are still scary

Gjensidige

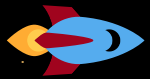
- Migrating was worth it, but a bit tricky
  - Mainly because of commit message-linting
  - Now only use `` `lefthook.yml` ``
  - Got to move some "git hooks"-code from `` `package.json` ``

⚰️ **eslint, prettier**

Go-to tools in almost any project, with a slew of extensions

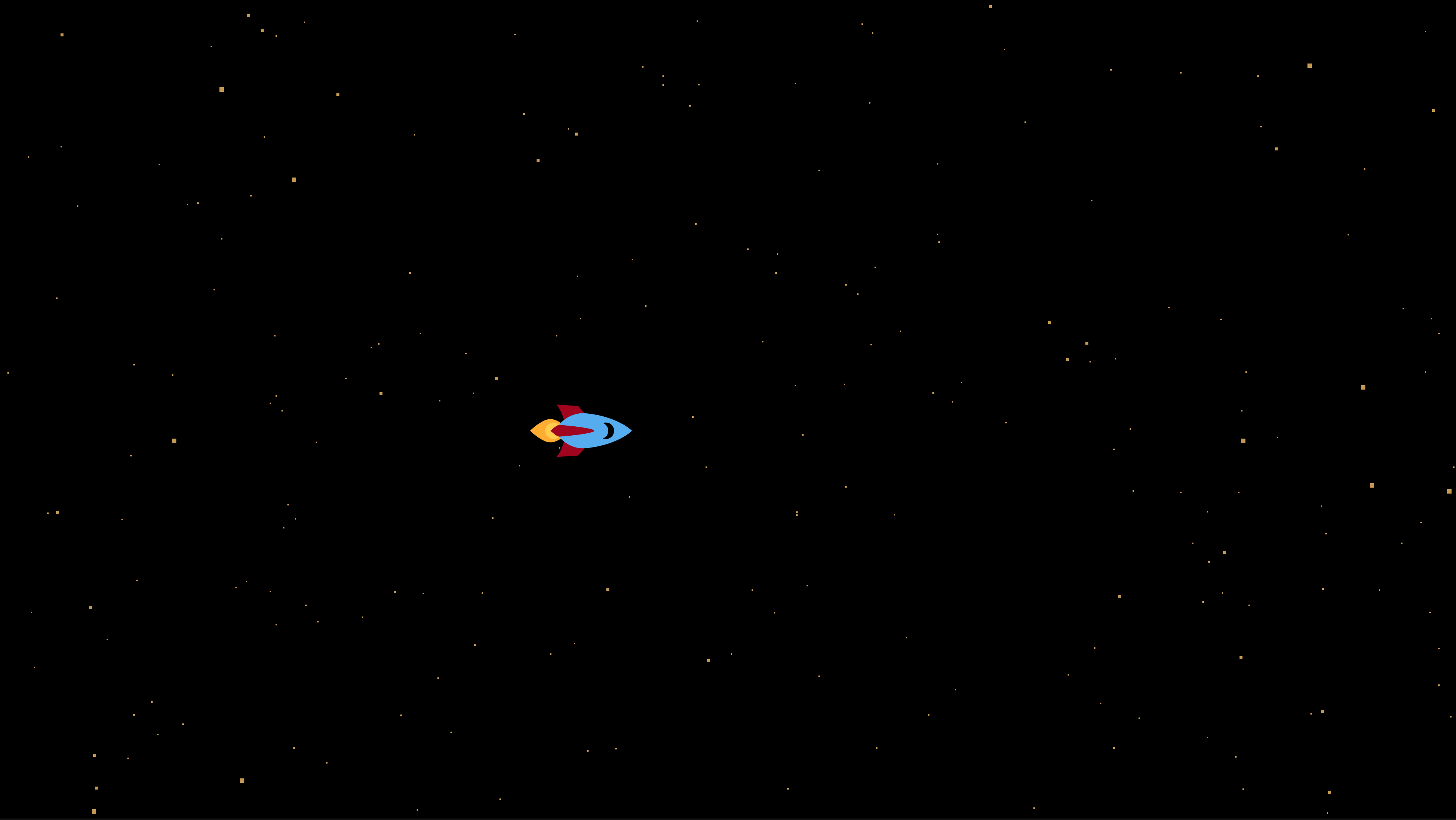
ES-lint, Prettier

- `` `eslint` `` helps us keep **code quality** high
- `` `prettier` `` helps us **format the code** the same way
- Usually needs **extra plugins/extensions**
- Go-to tools for almost any stack



❤️‍🔥 oxlint

- **oxc** - collection of JS-tools *(all Rust)*
  - Now Void(0) - *initiative by Evan You (inventor of Vue.JS, Vite, and more)*
- 💚 Up to **100 times** faster than eslint!
- 💚 Philosophy: **sane defaults**, **less plugins**
- 💔 ~~Doesn't fix lint-errors~~ … 💚 scratch that as of last week
- 💔 **Doesn't replace prettier** *(but another tool by oxc under development)*
- 💔 No type-rules (like `` `typescript-eslint` ``), or style-rules
- Docs warn: *"oxlint is not yet ready as a full eslint replacer"*
  - But depends on your project size and requirements



❤️‍🔥 Biome

- 💚 **35 times faster** than Prettier
- 💚 Biome replaces **both** eslint *(and typescript eslint)* and Prettier
- 💚 Comes with **migration-scripts** *(for eslint & prettier)*
- 💔 Not a 100% drop-in replacement …
  - yet (does 97% of Prettier)
- 💚 Like oxlint: defaults, batteries included *(no extensions)*
- 💔 Release-frequency
  - Last minor **1.9**: 7 months ago - last patch **1.9.4**: 5 months
  - 💚 2.0 to come this year

Gjensidige

- We started using **oxlint** in CICD for faster quality-check
- A couple teams are using **oxlint**, a handful **Biome**
- Rather simple to replace
  - Usually, only exception is **git hooks**
  - And that you might loose some feature

80 000
lines code

12s
eslint

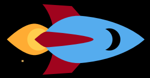
27ms
oxlint

⚰️ **lerna**

Go-to tool for doing monorepos since forever

Lerna

- All package managers now understand monorepos (`workspaces`)
- For `build`, `test`, `release`, we need monorepo tooling
- 💚 Lerna does a lot of things …
  - gives us Conventional Commits-versioning
  - 💔 but is very very big.
- 💔 Not fast
- 💔 Lacks cache
- Been around long, acquired by **Nrwl** a few years ago
  - ❤️‍🔥 Nrwl also develop **Nx** (Rust-based)
  - Claims to be the fastest tool - but also more paid service



❤️‍🔥 Turborepo

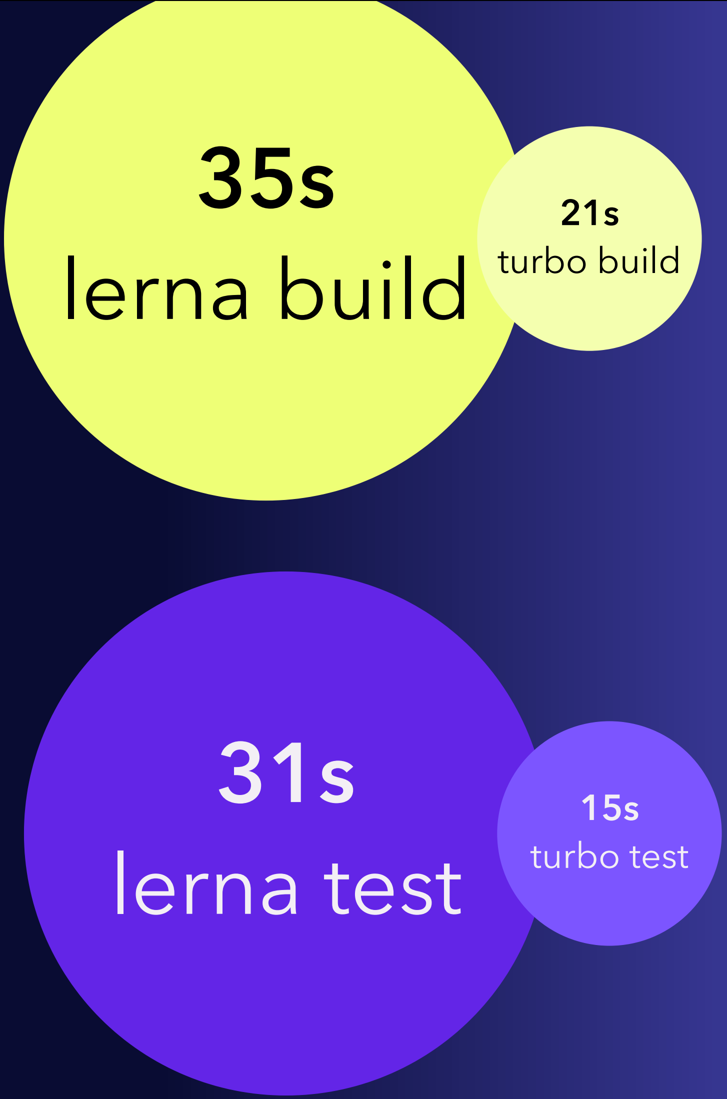
- By **Vercel** *(creators of Next.js)*
- 💚 A lot faster than Lerna
- 💚 Cache-mode called "turbo"
  - Only `build` / `test` / `lint` what is changed
  - 💔 Defaults to Vercel's cloud (usually paid)
  - 💚 … but you can configure this and use your own
- 💔 Cannot do versioning, git tag, publishing
  - Luckily, there's a tool for that:

❤️‍🔥 Changesets

- Changesets for tag, version, changelog, and releases
- 💔 Yet a package
- 💔 Not a Rust-tool
  - *but if you wanna replace Lerna you'll need something*
- 💔💚 Dialog-based releases using CLI
  - Feels odd, gets used to it
  - Auto-generated logs with **conventional commits** rewritten for Slack++
  - Changesets gave more control, flexibility

Gjensidige

- Cache gets us even faster!
- First month rocky, not anymore
  - *(turbo hanged sometimes during build)*
  - *(not 100% sure caused by turbo…)*
- Prob our most troublesome update
  - Still worth it



sass (or less)

We used Less, but not any more, and the replacement is NOT built with Rust or Go

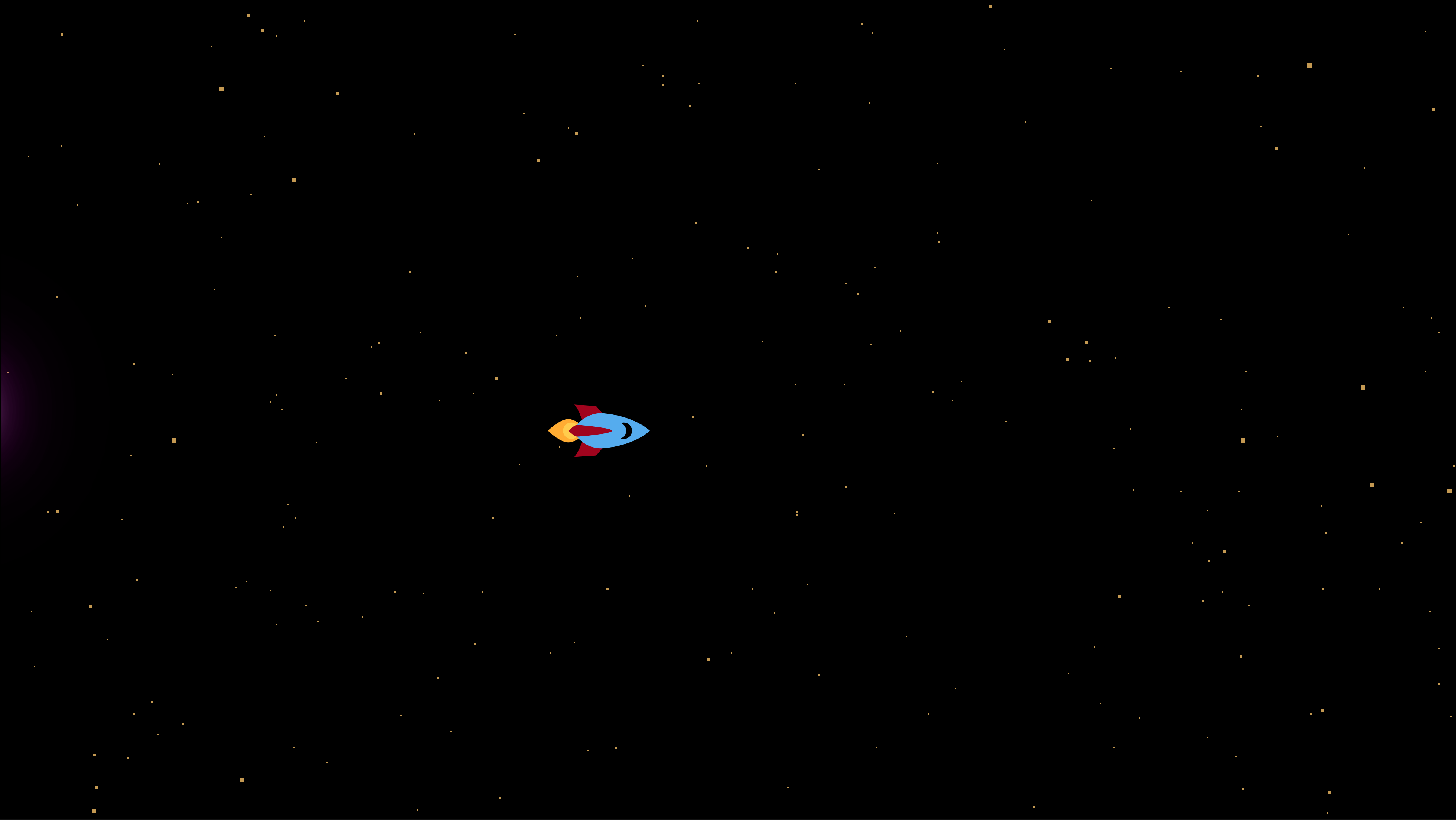
🕸 Sass/Less

- 💔 Built when CSS couldn’t do much
- 💔 Not that fast
- 💔 Packs some weight
- What are you really gaining?
  - Nesting? Variables? Mixins?

🍦 Just do vanilla!

- 💚 CSS evolves so fast
- 💚 Modern browsers too
- 💚 Variables works great
- Mixins mostly got in our way
- 💚 Nesting is improving
  - Install PostCSS nesting
  - Hook into your Vite-config
- **Builders Core** went this direction

Disappointed I didn't show a Rust-tool?



❤️‍🔥 LightningCSS

- Can't be a Rust/Go-talk without mentioning LightningCSS
- Haven't tried this (yet)
  - *But some of our teams have*
- 💚 Claims **over 100 times** faster than **CSSnano**
- 💚 All the features:
  - Vendor-prefixing, write latest CSS today, minify, modules, ++

🐘 The elephant

🪦 Node + NPM

Replacing small packages here and there, but what about the beast itself?!

Node, NPM

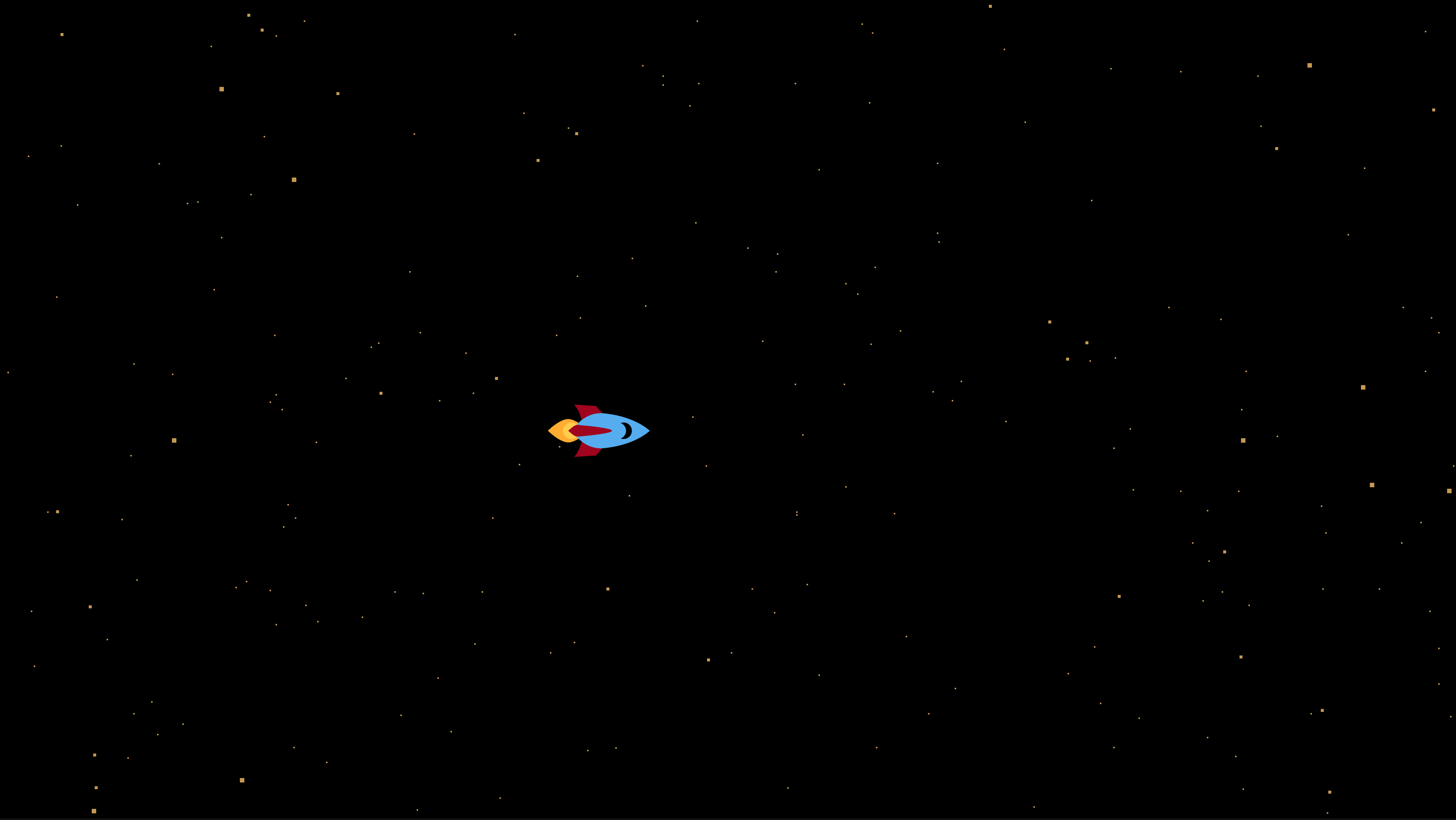
- Been around for a looooong time
- Architected in a completely different FE-world
- Supports a lot of legacy

We now have much better contenders

- They're **a lot faster**
- Can often replace a slew of other tools/packages
  - Like `dotenv`, `jsx` and `typescript`

❤️‍🔥 Deno

- By Node-creator
  - 2018: said **node/npm arcitechtual mismatch with modern JS**
- **Deno 1**: 2020
- **Deno 2**: 2024
  - 💚 **Faster, smarter, better DX, more secure**
  - 💚 "**batteries included**" - like OpenTelemetry, Linting
  - 💔 Heard **migration is harder** than other competitors
- Have not tested it … why?
  - Deno 1 was not an option
  - Deno 2 came when I was busy with …



🔥 Bun

> Bun is a new **JavaScript runtime** built from scratch to serve **the modern JavaScript ecosystem**.

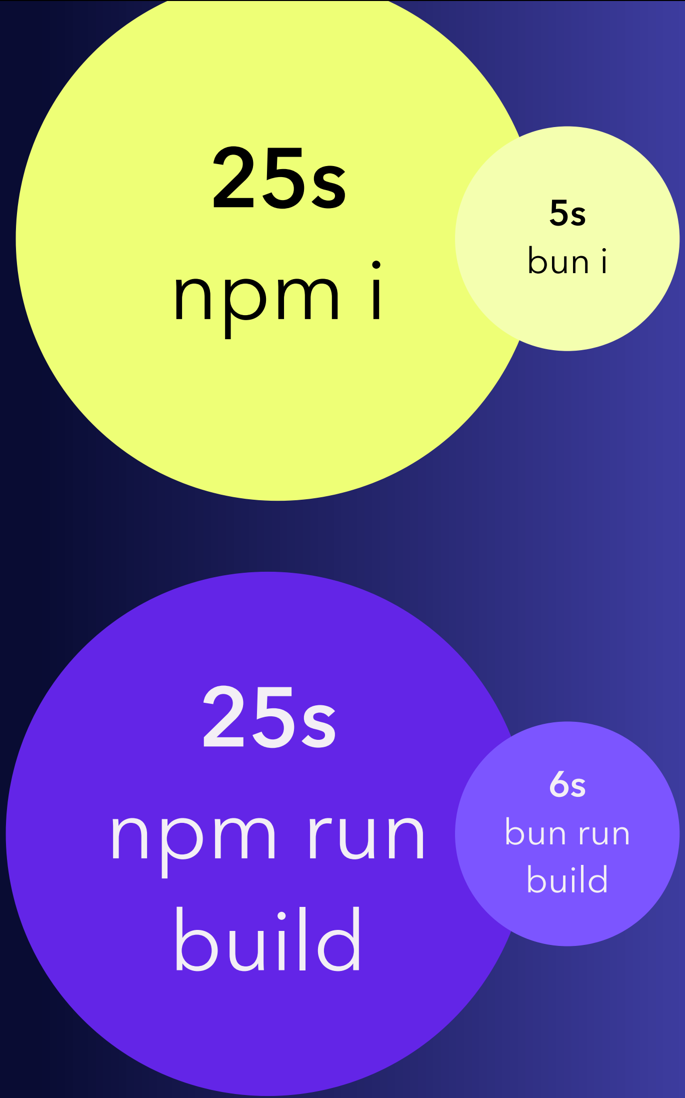
- Developed using **zig**
- `` `v1` `` 1.5 years ago
- 🗜️ Replaces Node **and** NPM (and more)
- 💚 Designed to start and run fast, "up to x4"
- 💚 "Batteries included":
  - Package-manager, native TypeScript-support, JSX-support, ++

❤️‍🔥 Bun (continued)

- 💚 Closest *"drop-in-replacement for Node and NPM"*
  - Even works side-by-side with node/npm
  - Uses `` `package.json` `` and `` `node_modules` ``
  - But custom `` `bun.lock` `` and `` `bunfig.toml` ``
- 💔 Lacks some edge-case node-features
  - 💚 Still 98% Node-compatible
- 💔 Using Node APIs in your code? Must re-map!
  - 💚 Identical APIs, change only the import:
  - Prepend with `` `node:` ``
  - `` `import * from "fs"` `` becomes `` `import * from "node:fs"` ``

Gjensidige

- **Builders Core**: 3 months in bun-land
- Our **biggest webapps** (and **mobile app**): Bun
- Hesitant? Just try `` `bun i` ``
  - keep npm for `` `npm run build` `` etc



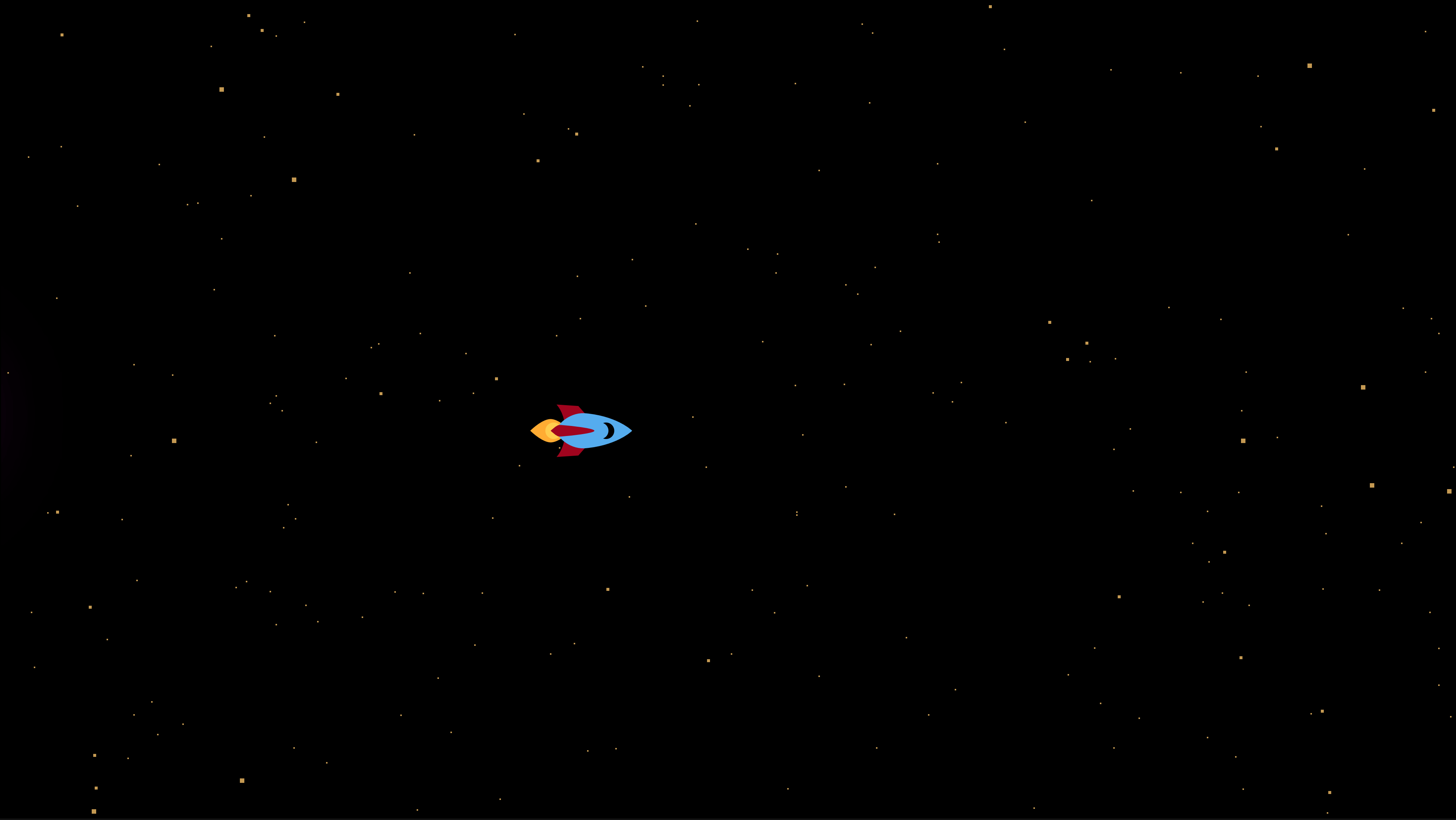
Stack recap

Builders Core - stack anno 2025

For now …

- `` `bun` `` replaced node/npm
- `` `oxlint` `` replaced eslint (in cicd)
- `` `turborepo` `` replaced lerna
- `` `lefthook` `` replaced husky
- Vanilla replaced less

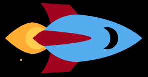
🔮 Future



❤️‍🔥 Rolldown

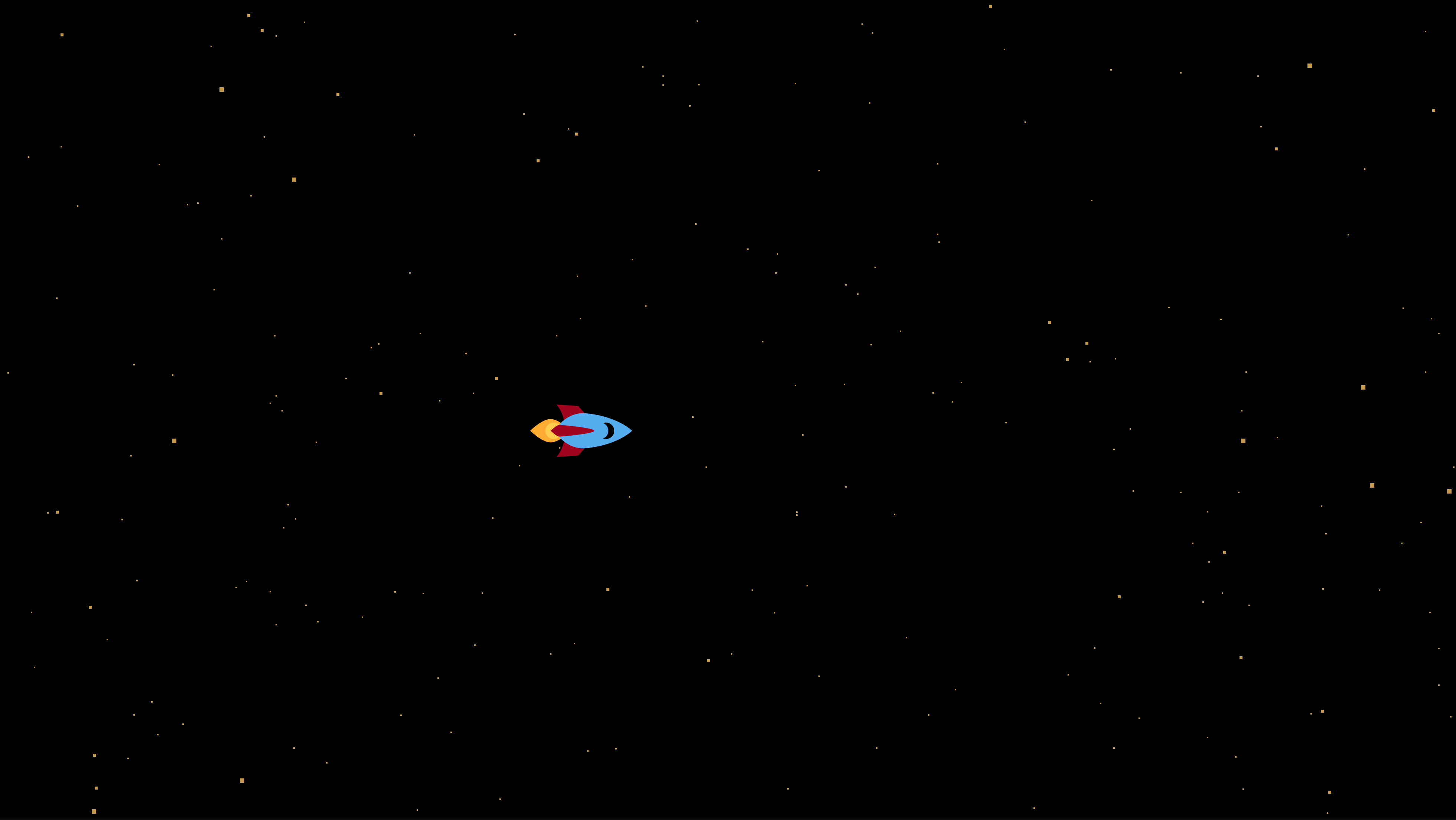
- By **Void(0)**
- Speed!
- Replacement for Rollup and esbuild in **Vite**
  - Also works *without* Vite
- "Vite is not perfect" - Evan You
  - Held back by **Rollup** and **esbuild**
  - Tools are used **sub-optimal**
- Maybe in Vite 7? 🤷‍♂️
  - Or sooner? - Evan: "Beta by end of March"

ck



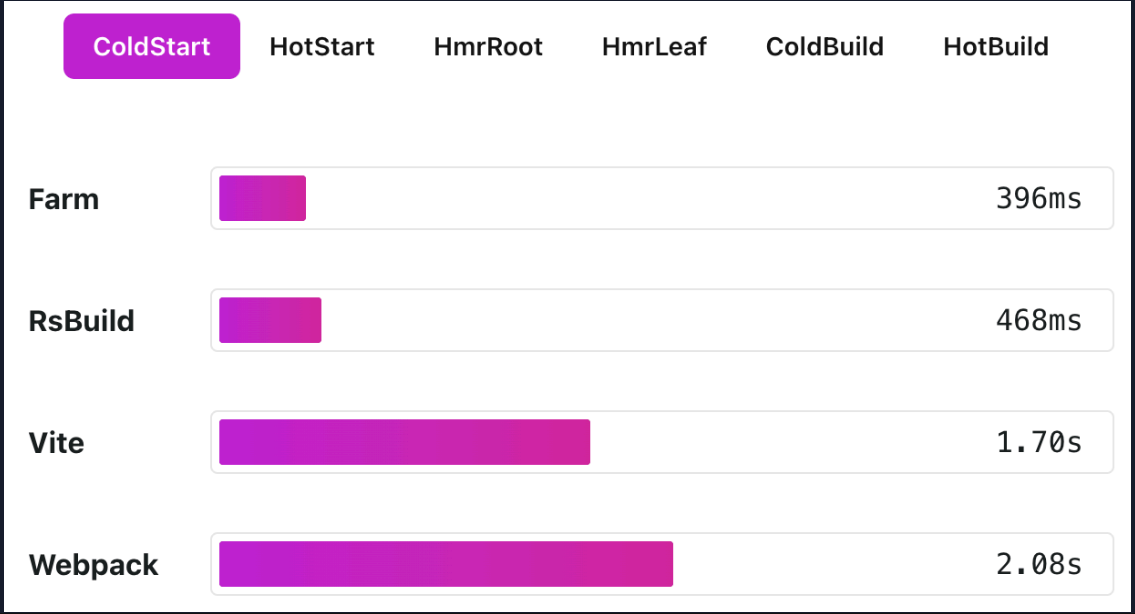
❤️‍🔥 The RS-stack

- 💚 OMG cutez logoz
- RSbuild - the faster Vite-option
- RSpack - the faster webpack/rollup-option
- Says "there’s nothing faster" …



Farm

- Rather new *(1.0)*
- A lot faster than **Vite**, even **RSbuild**!
- Against Vite's `` `dev ≠ prod` ``-philosophy (their "Why"-page)



Farm	396ms
RsBuild	468ms
Vite	1.70s
Webpack	2.08s

Gjensidige

- For now: **stay on Vite**
  - Hoping for a "free" boost with **Rolldown**
  - But **Farm** looks nice
- Also want to use more of **Bun**

Bare in mind

- 🐢 Speed is not everything
- **Don't go bananas 🍌**
  - One thing at a time
- There's always a faster car 😅

Do you have any other suggestions? Always eager to improve DX!

Thanks for listening!

I'm 👨‍💻 Bobby Westberg

⬅️ Download slides

at www.gjensidige.builders

(Contains a few bonus suggestions)

Gjensidige