

# Seeing Beyond the Framework Illusion

*Sean C Davis*

# ~~Why Vue is better than React!~~

You should migrate from Next  
to TanStack immediately!

11ty is a full-stack framework!

Astro is the best choice  
for enterprises!

# Sean C Davis

Developer Education, Netlify



 @seancdavis

 @seancdavis.com

  @seancdavis29

# How do we choose a framework?



#SchittsCreek



CAN WE FEEL THAT?

# Developer Experience (DX)

Good DX means shipping faster  
from Day 1 to Day 1,000.

A framework's value comes  
from the engineering practices  
applied to its DX.



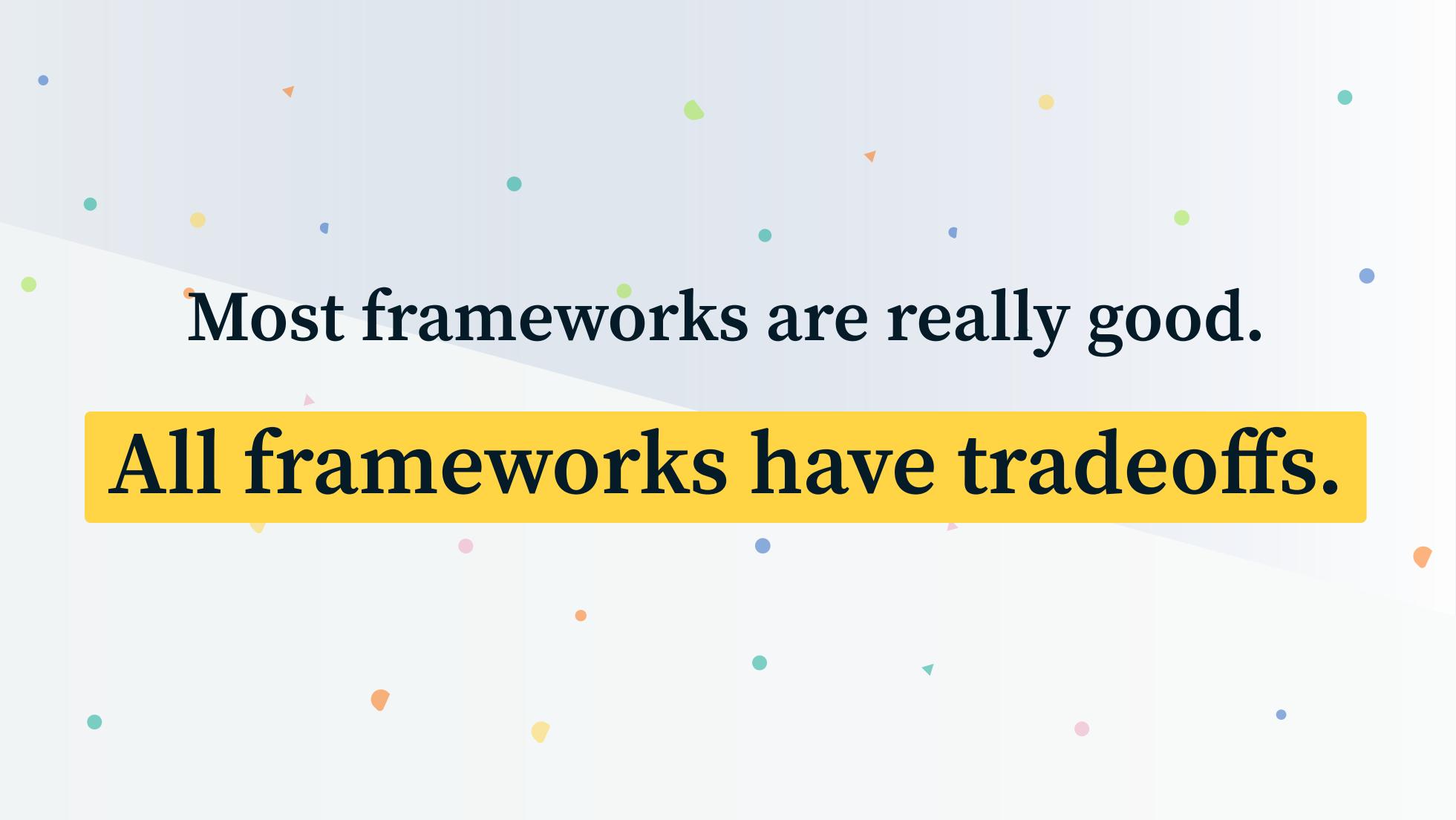
Stan.

# User Experience (UX)

**Users don't care about your  
framework choice.**

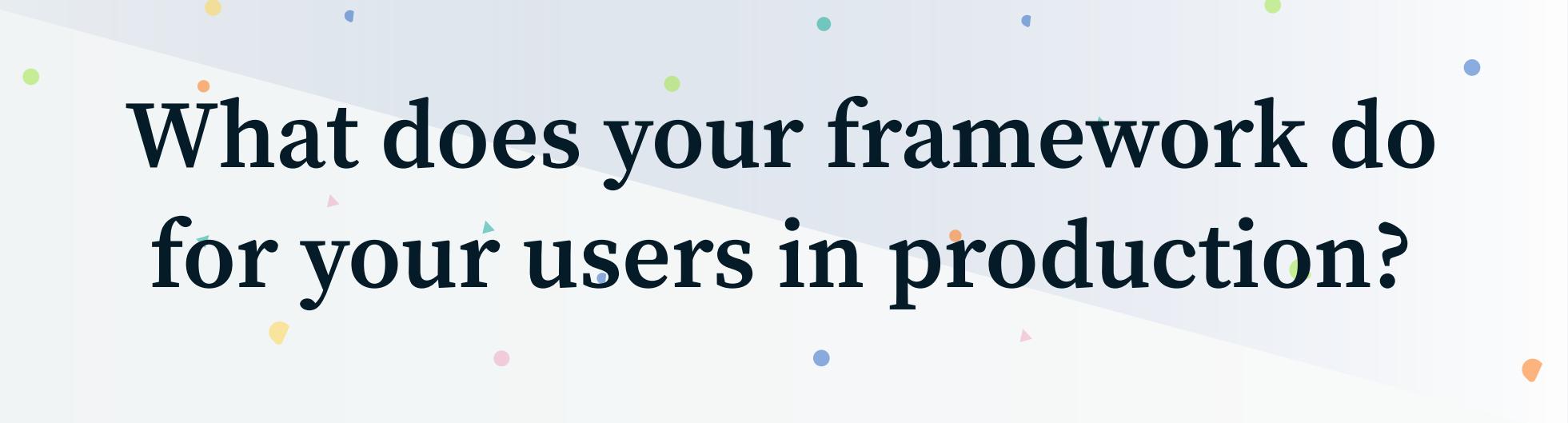
**They want to get things done.**

**FAST!**



Most frameworks are really good.

All frameworks have tradeoffs.



What does your framework do  
for your users in production?

# Example: Astro Dev vs Prod

## *Development*

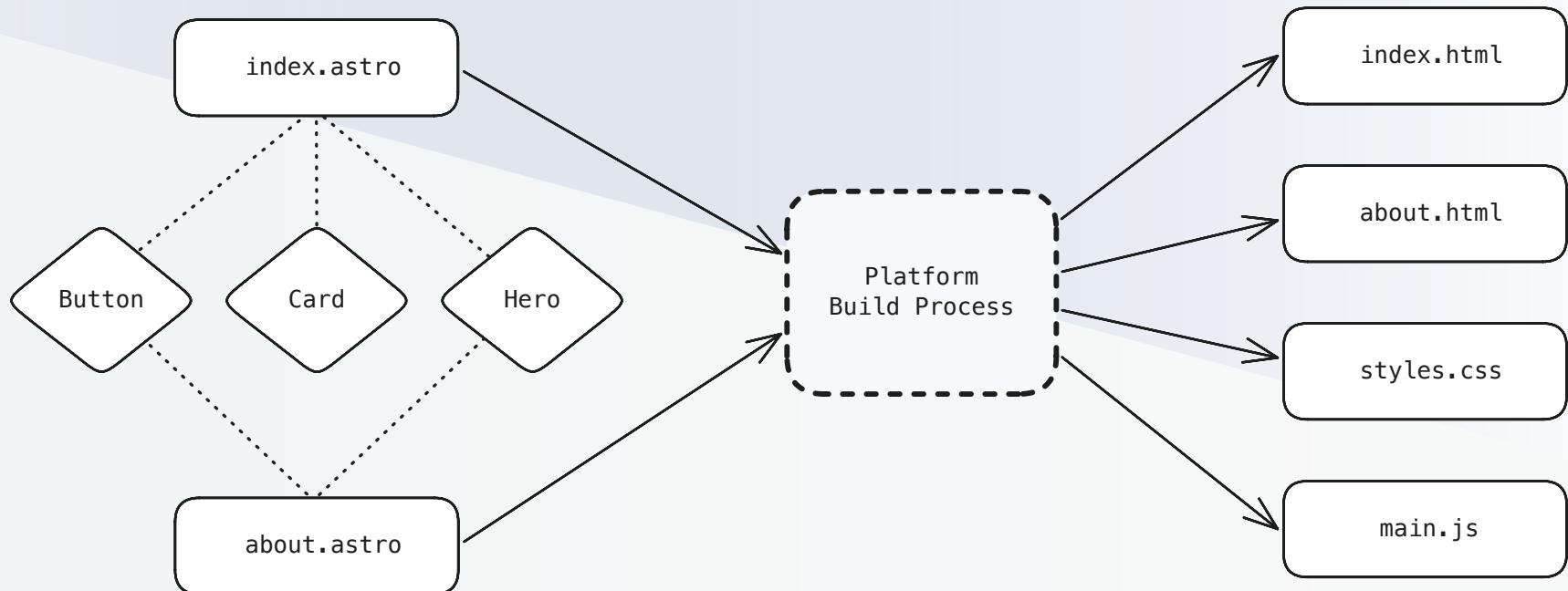
```
---  
import Welcome from "../components/Welcome.astro";  
import Layout from "../layouts/Layout.astro";  
---  
  
<Layout>  
  <Welcome />  
</Layout>
```

## *Production*

```
<!DOCTYPE html>  
<html lang="en" data-astro-cid-sckkx6r4>  
  <head>  
    <meta charset="UTF-8" />  
    <meta name="viewport" content="width=device-width" />  
    <link rel="icon" type="image/svg+xml" href="/favicon.svg" />  
    <meta name="generator" content="Astro v5.13.7" />  
    <title>Astro Basics</title>  
    <style>  
      /* ... */  
    </style>  
  </head>  
  <body data-astro-cid-sckkx6r4>  
    <div id="container" data-astro-cid-mmcc7otgs>  
      <!-- ... -->  
    </div>  
  </body>  
</html>
```



# Example: Astro Build Process (simplified)



# Example: Rails Dev vs Prod

## *Development*

```
class UsersController < ApplicationController
  authenticate! :user, except: %i[accept_invitation complete]
  set! :user, only: %i[show edit update destroy]

  def index
    @users = User.all.includes(:company)
    @companies = Company.by_name
    params[:sort] ||= { by: :last_name, type: :alpha, in: :asc }
    @users = sort_collection(@users)
    @users = Kaminari.paginate_array(@users).page(params[:page])
  end
end
```

## *Production*

```
class UsersController < ApplicationController
  authenticate! :user, except: %i[accept_invitation complete]
  set! :user, only: %i[show edit update destroy]

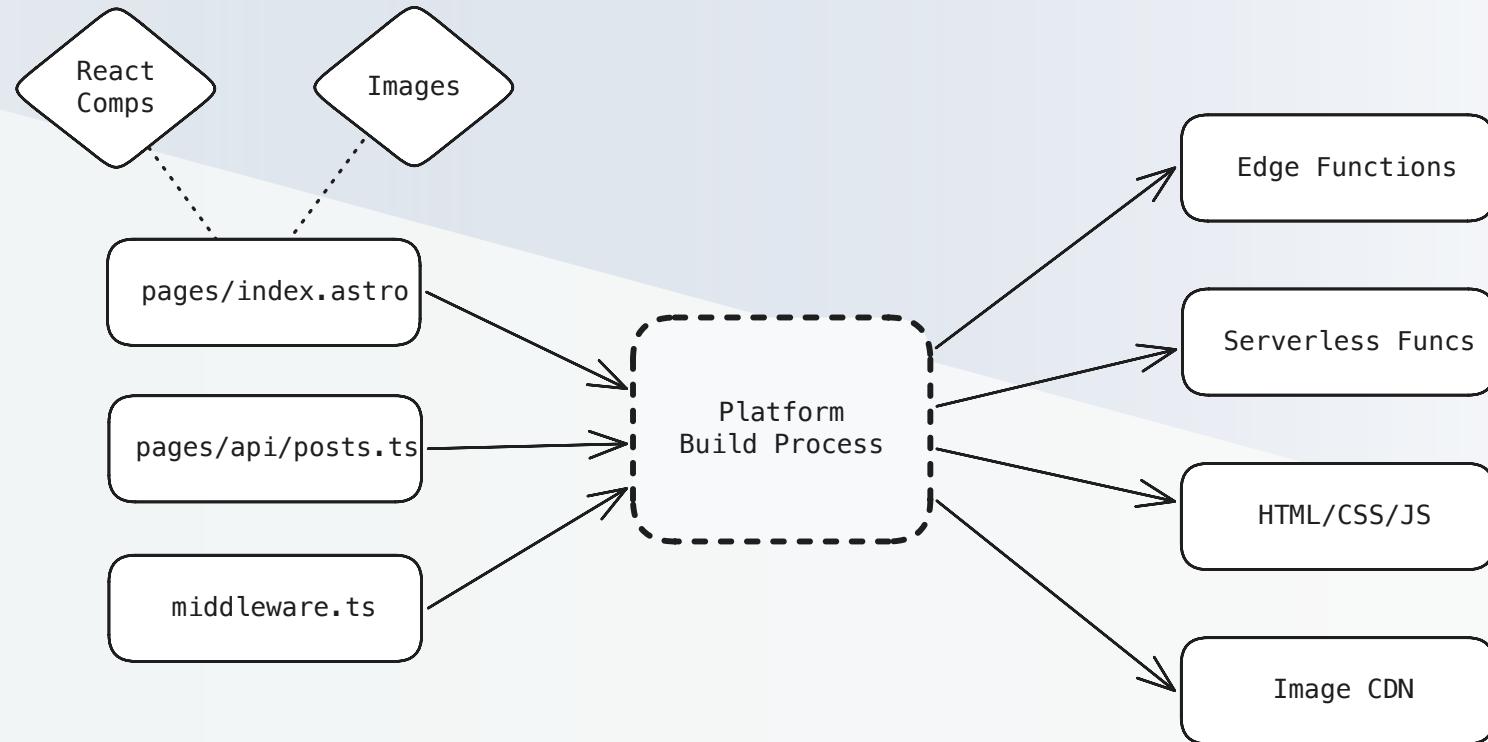
  def index
    @users = User.all.includes(:company)
    @companies = Company.by_name
    params[:sort] ||= { by: :last_name, type: :alpha, in: :asc }
    @users = sort_collection(@users)
    @users = Kaminari.paginate_array(@users).page(params[:page])
  end
end
```

# Example: SSR with Astro on Netlify

```
import netlify from "@astrojs/netlify";
import { defineConfig } from "astro/config";

export default defineConfig({
  output: "server",
  adapter: netlify(),
});
```

# SSR in JavaScript frameworks



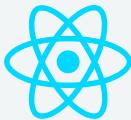
A platform adapter hooks into a framework's build process to transform framework features into platform primitives

**Every framework is only  
as powerful as  
the platform it runs on.**

• All these frameworks have the  
same set of capabilities on Netlify:



11ty



# Demo time!



Platform integration is what  
matters most for a framework.

# Introducing AX: Why Agent Experience Matters

Mathias Biilmann · January 28, 2025



We tend to focus on the generative aspect of large language models, the outputs they can create, the text, images and videos they produce. But the largest disruption from the current evolution of AI will come from bringing **agency** to computers.

Agent experience matters because  
DX and UX are directly impacted  
by the quality of solutions an  
AI agent can implement.

Pick what works for you.

We're all different.

*And ...*

**Don't forget to have  
a little fun along the way!**