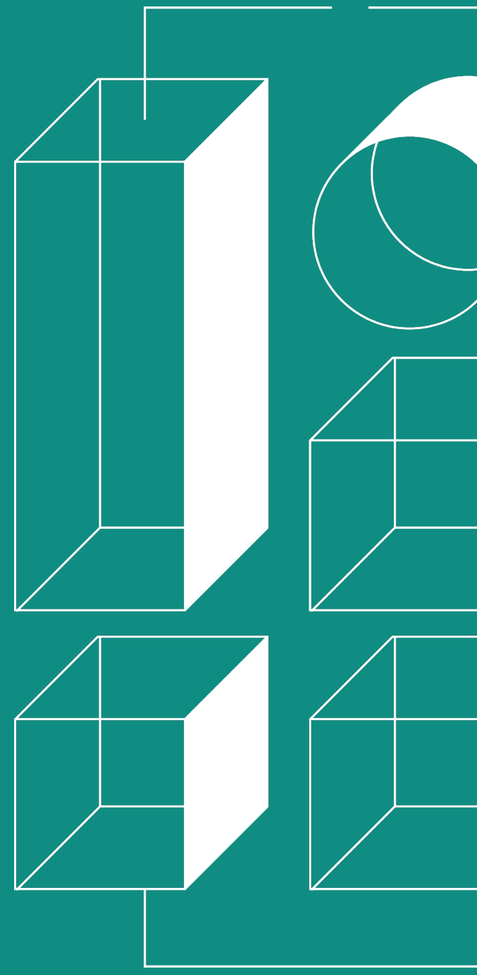
Skip the Overhead:

Lean Web Development with Django

Jochen Wersdörfer

- Django Developer since 2013
- Python-Podcast.de Host
- Creator of **django-resume**



What to Expect

"Developers are drawn to complexity like moths to a flame, often with the same result."

Neal Ford

1. What's up with modern web development?
2. Building interactive websites without the bloat
3. Live demo: Keeping it simple with django-resume

Modern Web Development

Backend

- JSON/REST APIs
- Or, heaven forbid, GraphQL

Frontend

- React/Vue/Angular + Store (Redux, Vuex, etc.)

CSS

- Tailwind or Bootstrap or another large framework

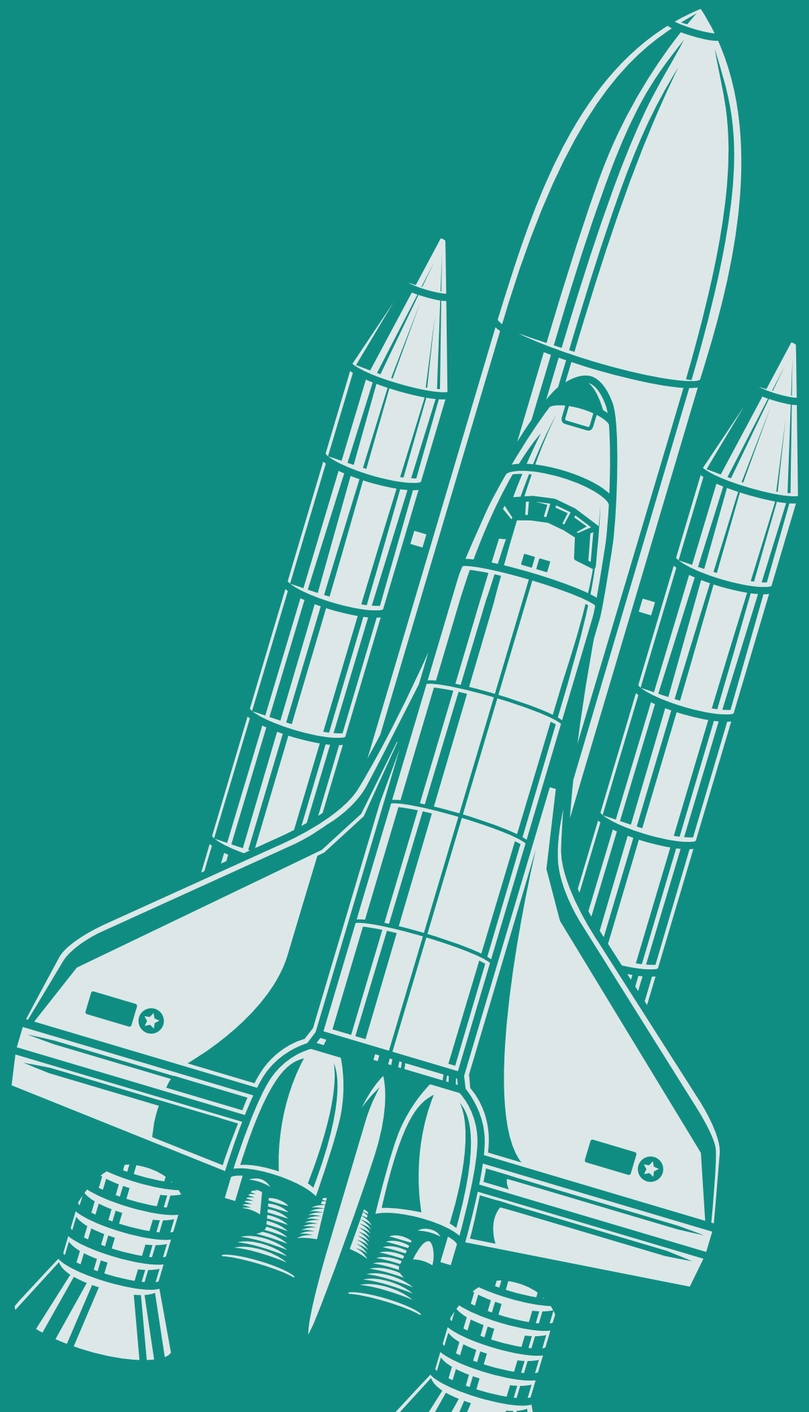
Build Pipeline

- TypeScript, Webpack/Vite, Babel

Modern Web Development

> "GOD: i have made a Single page webapp
>
> ANGELS: you fucked up a perfectly good website is what you did. look at it. it's got dependency injections."

Eric Meyer

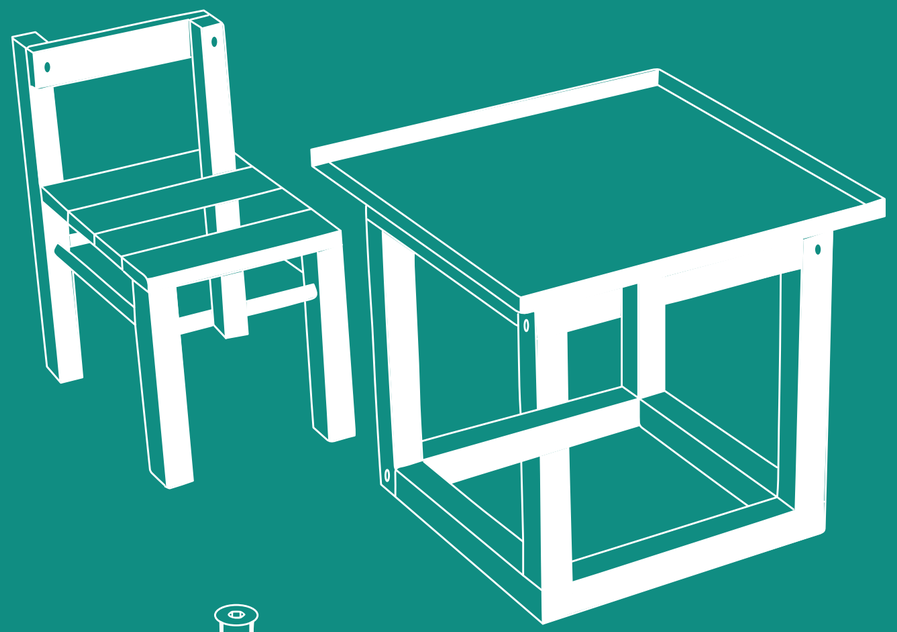


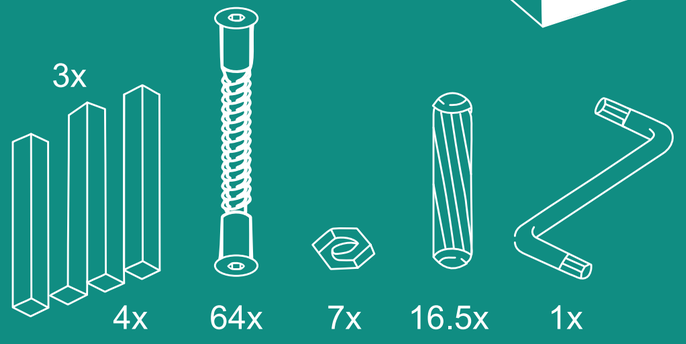
Shooting for the Moon

> "We choose to go to the moon in this decade and do the other things, not because they are easy, but because they are hard."

John F. Kennedy

PARADÖX
play table and chair





Shooting for the Moon

> "The Programmers' Credo: we do these things not because they are easy, but because we thought they were going to be easy."

Maciej Cegłowski

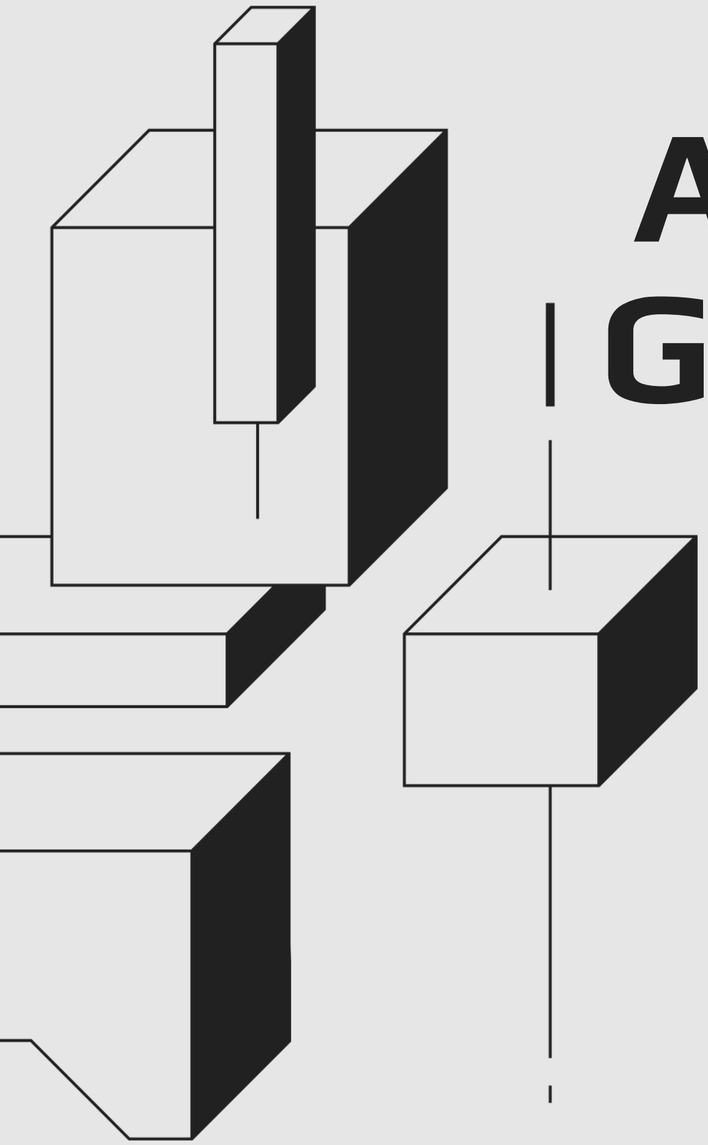
Non-Functional Requirements:

What We Really Want

Smooth user experience

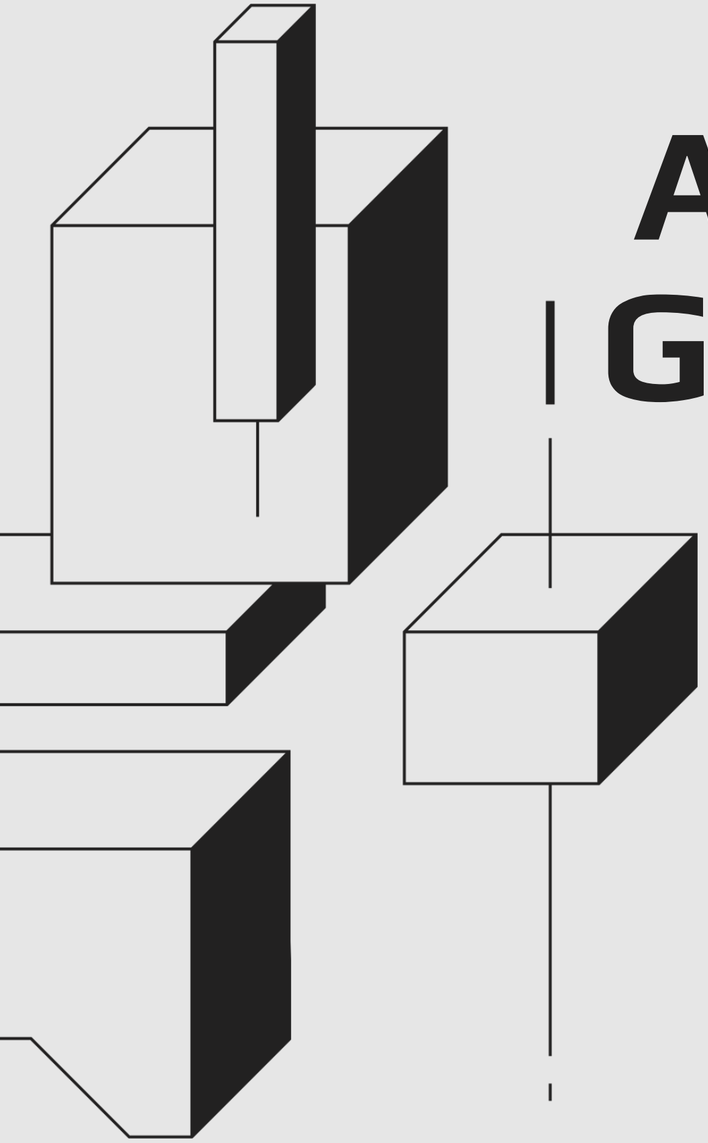
Appealing visuals

Developer experience



Achieving Modern Web Goals Without the Bloat

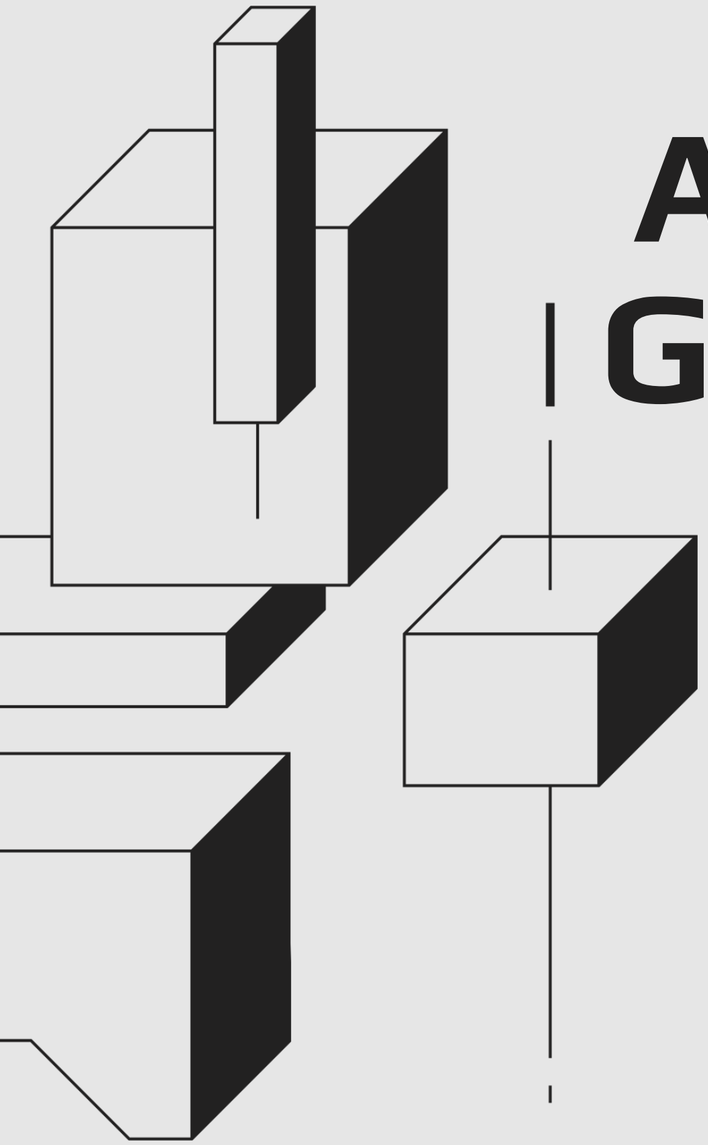
> "Einstein repeatedly argued that there must be simplified explanations of nature, because God is not capricious or arbitrary. No such faith comforts the software engineer."
>
> Fred Brooks



Achieving Modern Web Goals Without the Bloat

django

- Community Governed
- Boring Technology – Spend your Innovation Tokens Elsewhere
- Django is a Frontend Framework



Achieving Modern Web Goals Without the Bloat

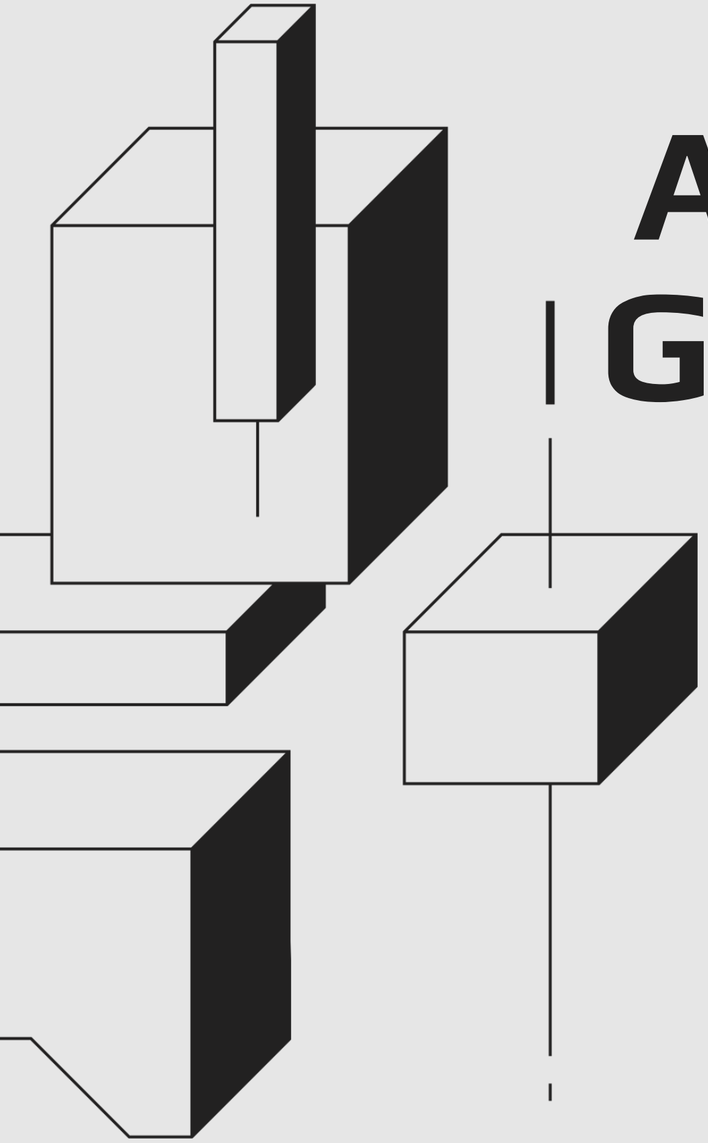
CSS

EVERY LAYOUT

Relearn CSS layout

every-layout.dev

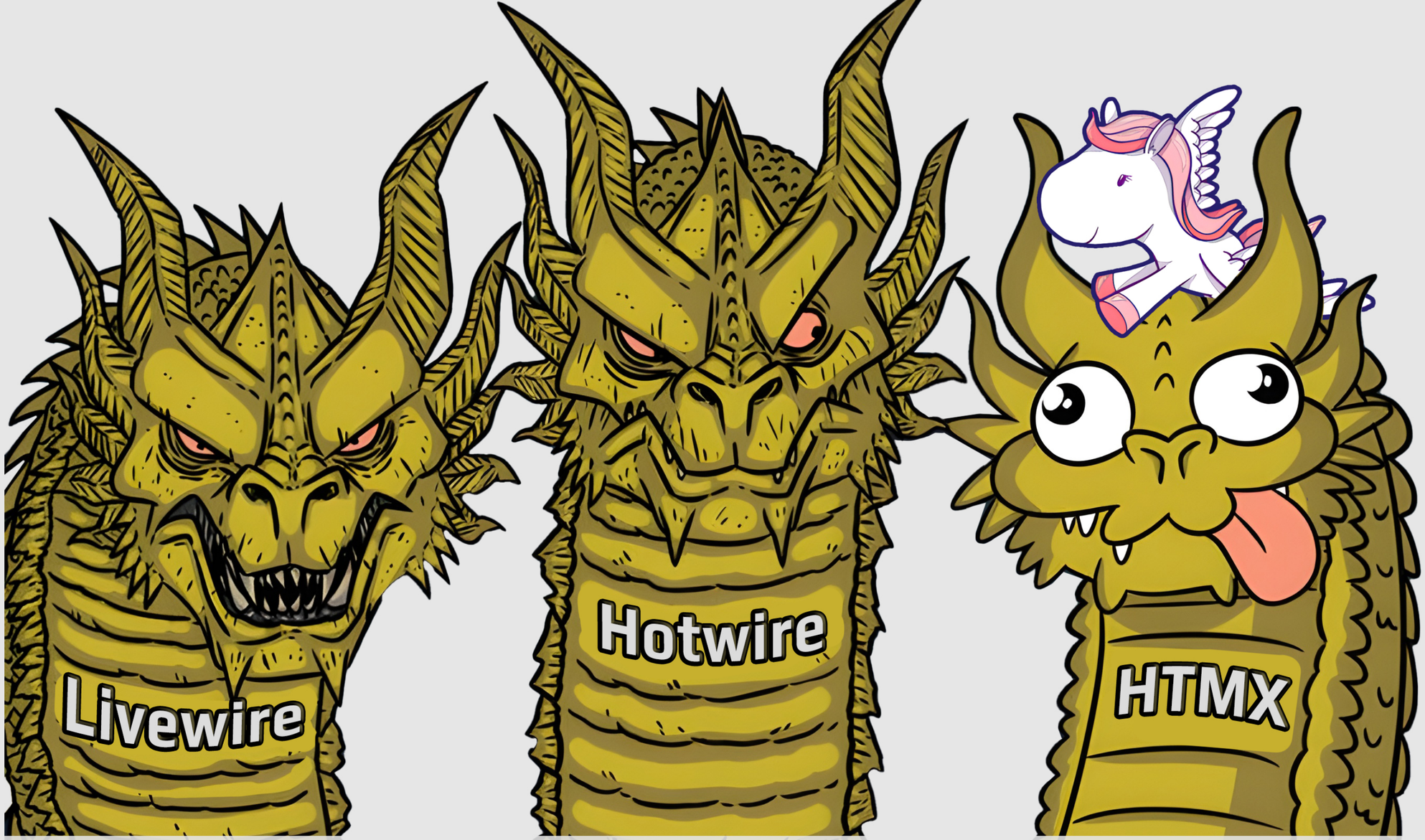
- Big Frameworks no Longer Essential
- Selectors and the Cascade are your Friends
- Layout Primitives Instead of Utility-First Components



Achieving Modern Web Goals Without the Bloat

</> htmx

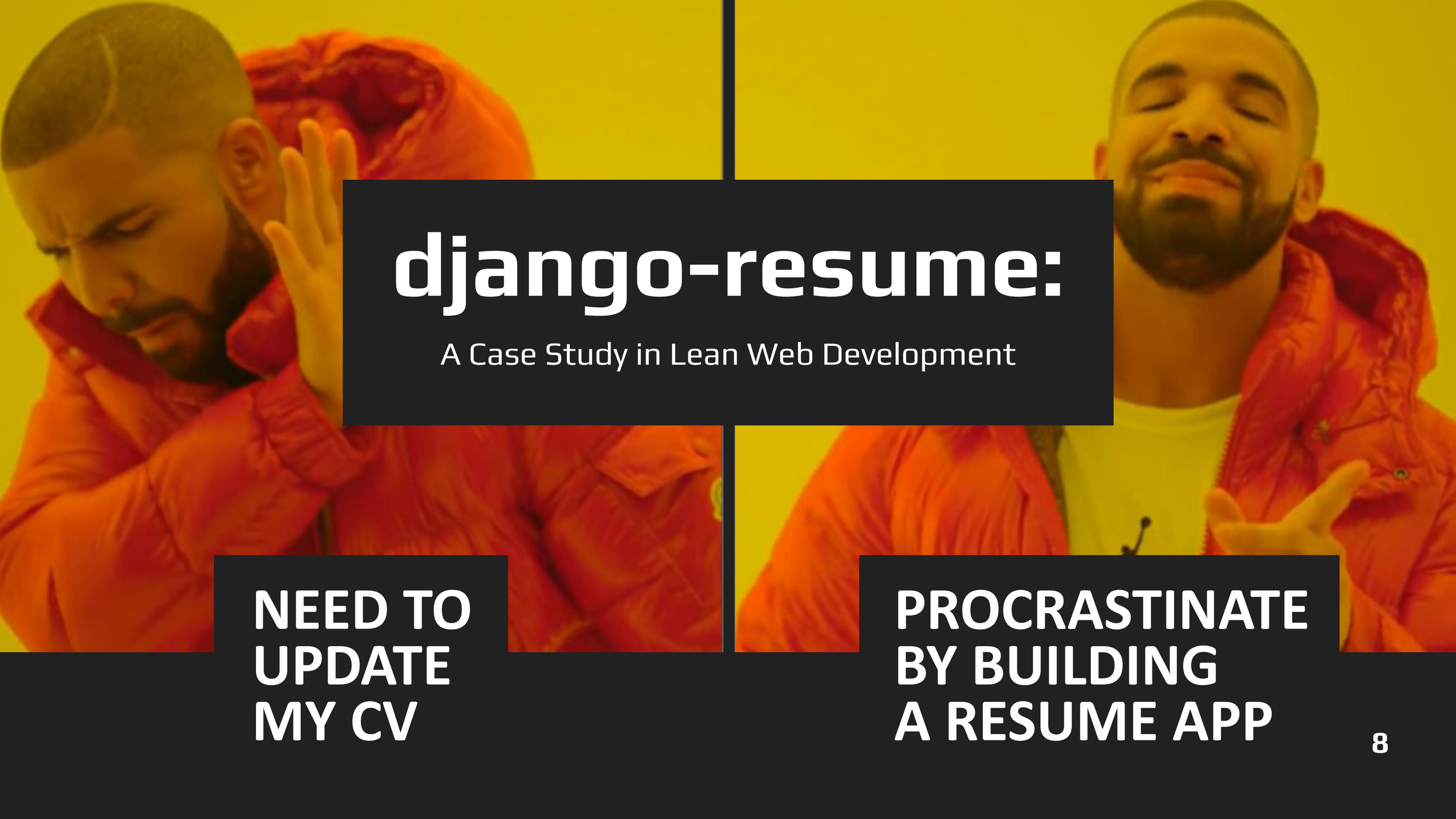
- The Next Evolution of HTML Pokémon
- SPA Experience in Multi-Page Apps
- HOWL – Hypermedia On Whatever you'd Like



Laravel

RAILS

django



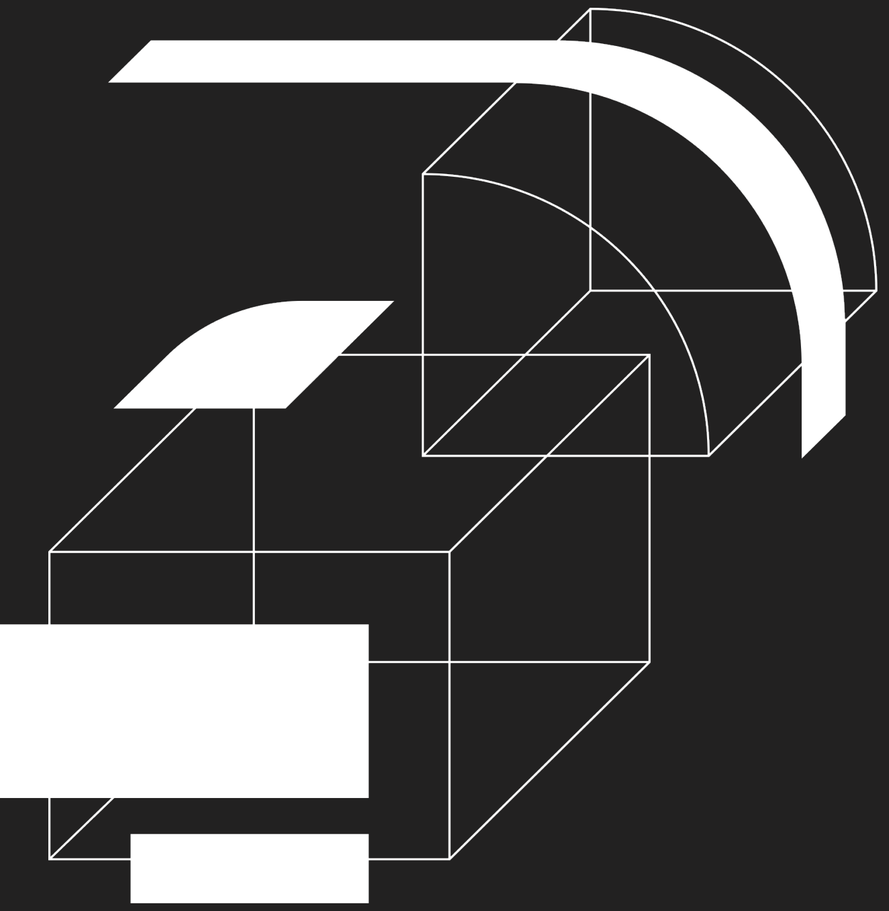
django-resume:

A Case Study in Lean Web Development

NEED TO UPDATE MY CV

PROCRASTINATE BY BUILDING A RESUME APP

Functional Requirements



> *"Technology is the art of arranging the world so we don't have to experience it.*
>
> Max Frisch

Single Source of Truth

- Manage all resume content in one centralized system.

Easy Customization

- Adaptable to different professional needs: Developers, academics, designers, and more.

Plugin Architecture

Central JSONField for Plugin Data

- No additional models or migrations required.
- Each plugin stores its data under its unique namespace in the JSONField.

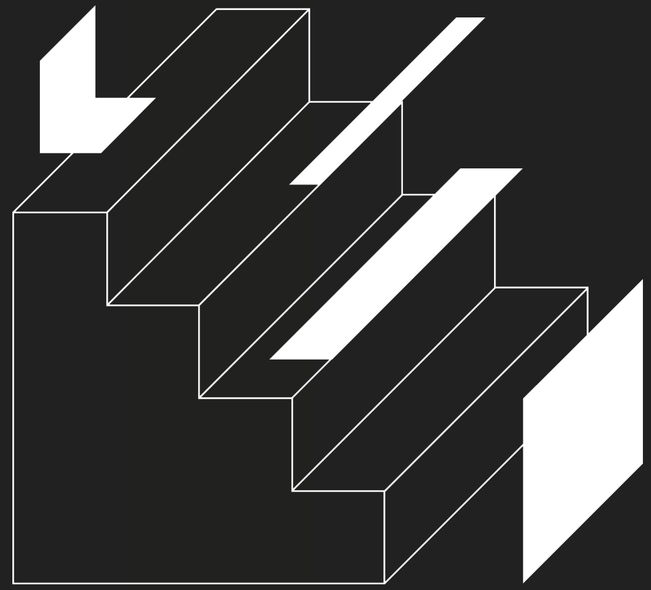
> "Anything worth doing is worth doing badly."
>
> G. K. Chesterton

Django Forms for Validation

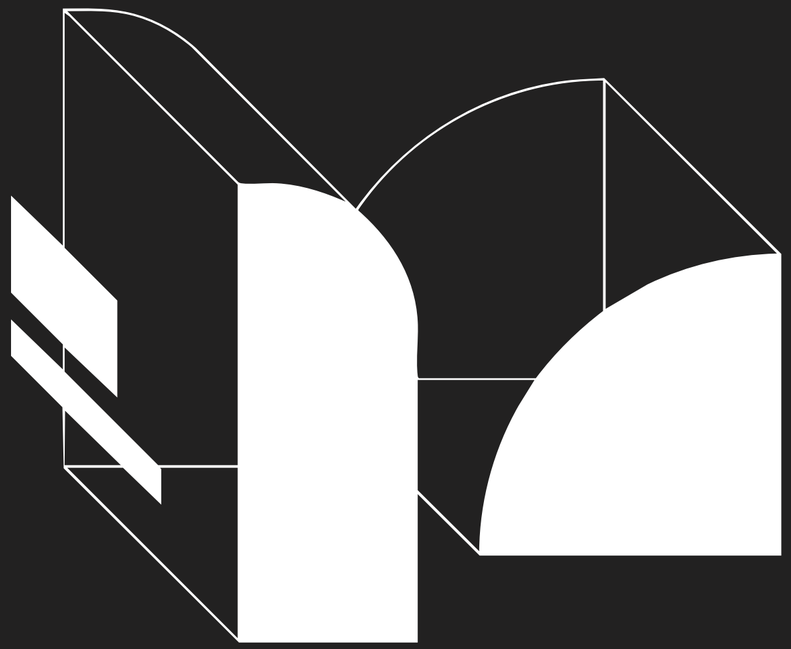
- Define and validate the structure of plugin-specific data.

Templates for Rendering and Editing

- Render plugin data as HTML for display.
- Provide user interfaces for editing plugin data.



Web Components for UI Edge Cases



> "UIs are big, messy, mutable, stateful bags of sadness."

Josh Abernathy

Examples in django-resume

<badge-editor>

- Handles dynamic lists (e.g., skills or tags)
- Updates locally → Single server request on "Save"

<editable-form>

- Links `contenteditable=true´ fields to inputs of a hidden form
- Enables inline editing without custom form templates

Why Not SPAs?

Web Components are:

- Built into the browser → No extra JavaScript framework
- Standards-based → Durable & future-proof

Leveraging LLMs for Automation

Structured JSON Generation

Streamlining Plugin Development

- Use LLMs to generate plugin boilerplate:
  - Forms, templates, and registry setup
  - Example prompt: "Create a plugin for certifications (name, issuer, date)"
- Saves time and lowers the barrier for customization

> "ZIZEK: that AI will be the death of learning and so on; to this, I say NO! My student brings me their essay, which has been written by AI, and I plug it into my grading AI, and we are free! While the 'learning' happens, our superego satisfied, we are free now to learn whatever we want."
>
> Slavoj Žižek

Demo: django-resume in Action

Overview of django-resume

- The admin interface for managing plugins and resume content.
- Example plugins: Projects, Certifications, Skills.

Web Components in Action

- Demonstrate `BadgeEditor´ for dynamic lists.
- Inline editing with `EditableForm´ and `contenteditable=true´.

Showcasing Interactivity

- Adding a new plugin via the registry.
- Using htmx for partial updates (e.g., editing a skill or certification inline).

Highlight Extensibility

- How new plugins fit seamlessly into the system.
- JSONField data structure in action.

Let's see how django-resume keeps it simple yet powerful!

Thank you!

Any Questions?

I'm happy to answer them!

Want to Get Involved?

Check out
django-resume
on GitHub

Got ideas? I'd love your feedback!

Find Me Online

GitHub: **ephes**

Fediverse: **@jochen@wersdoerfer.de**

Blog: **wersdoerfer.de/blogs/ephes_blog/**

Python Podcast