MFEs and Generators

Kisha Delain

Outline

- Code generators:
    - What are they?
    - Why would you use one?
    - Examples from different languages
    - Example:
        - nx code generator
        - How to customize
- MFEs
    - What is a micro front end?
    - What are use cases?
    - Example: nx → mfe
- This talk is NOT about using LLMs to generate or autocomplete code

Code generators

- Templates: What?
  - Tailored service templates
  - Create from basic input parameters
- Templates: Why?
  - Consistency across developers
  - Reduce repetitive coding
    - Initial project setup (think create-react-app – but there are better tools like vite or nx!)
    - Data access/schemas
    - Class setup, components, forms – anything that has a standard setup
    - Add & Update
  - Focus on that business logic

Code generators

- Templates: When?
  - Similar projects
  - Rewriting into a new framework
  - Meeting code standards
  - Micro frontends! Make a bunch!
  - Unit tests!

- Templates: Words of Caution
  - You still need to know what you’re doing
  - Templates can only get you so far – still need to write that business logic
  - Repository bloat – do you really need all those files?
  - Bug fixes – now you need to update all the things
    - Can you do this in a way that’s automatic? MAYBE (but also maybe not – be careful!)

Some Code Generators

Languages

- Python: Cog
- Ruby on Rails
- TypeScript (nx, vite, etc)
- .Net: Custom templates for dotnet, T4

API/Data access

- Swagger codegen
- Open API generator
- GraphQL codegen
- Apollo codegen

Not necessarily recommendations

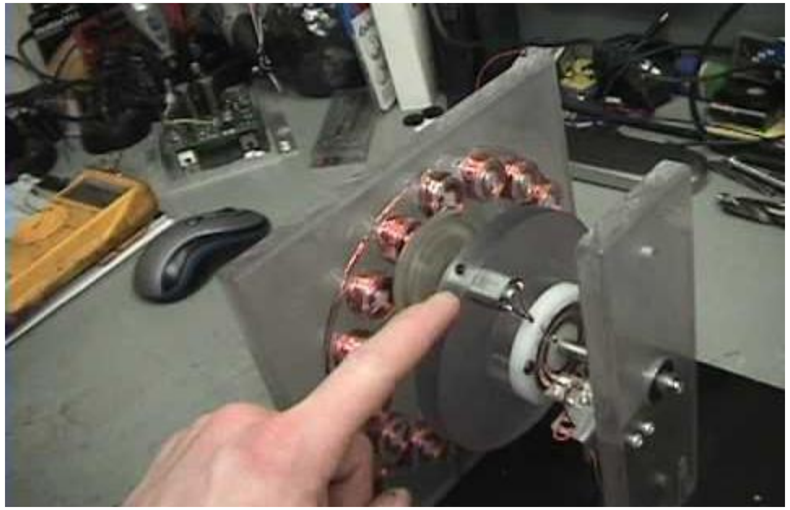
What code generators do you know about?
https://PollEv.com/free_text_polls/VqvrCgEgfKPrxshNrT36I/respond

Example: customizing Nx

- Pre-existing nx generators:
  - Lint with eslint
  - Unit tests with jest
  - Buildable or not

- Let’s take a look at some customization!
  - Write a Simple Generator (nx documentation)

Example: Make A Generator



- Setup:
  - Must be in an nx workspace
  - Must have installed @ng/plugin (for angular)

- Create plugin: nx generate @nx/plugin:plugin <my-plugin>
- Scaffold out:

```
npx nx generate @nx/plugin:generator <folder> --project=<my-plugin>
```

- Generate some stuff!
  - Nx plugin
  - Command line: nx generate @myorg/my-plugin:<folder>   <my-plugin>

Example: Our Generator

- Interface
  - schema.d.ts
- Schema
  - schema.json
- Generator
  - generator.ts
  - Note the “name” function
- Templates
  - Files/src/*.ts.template

Anker solar generator

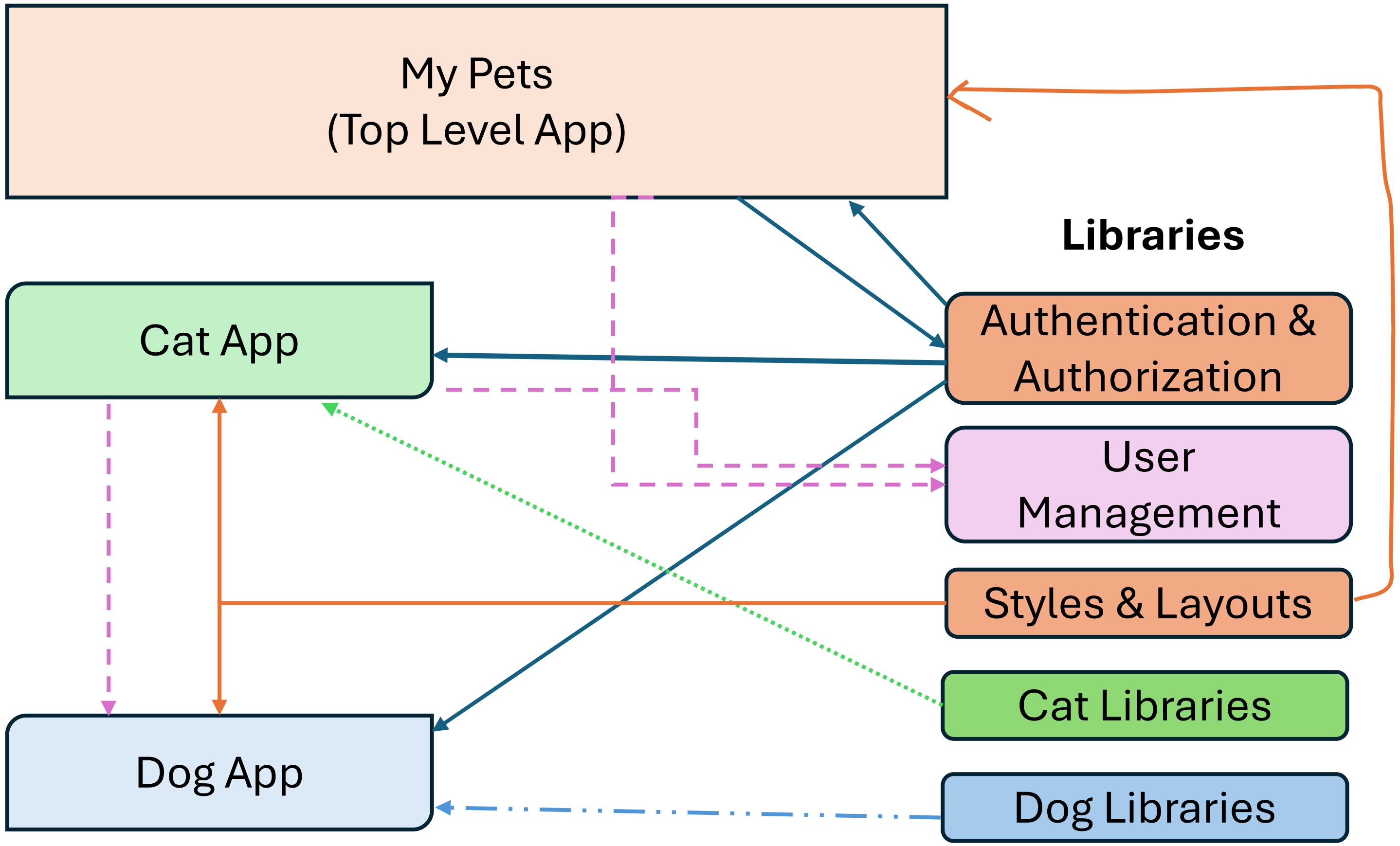
Micro Front Ends

Roderick Cox leads Minnesota Orchestra
(Matt Blewett, photographer)

Setup:

- One repo
- One “orchestrating” workspace
- Many “front end” apps
- Shared libraries across apps
  - Can be consumed by external people if published

Micro Front Ends



My Pets
(Top Level App)

Libraries

Cat App

Authentication & Authorization

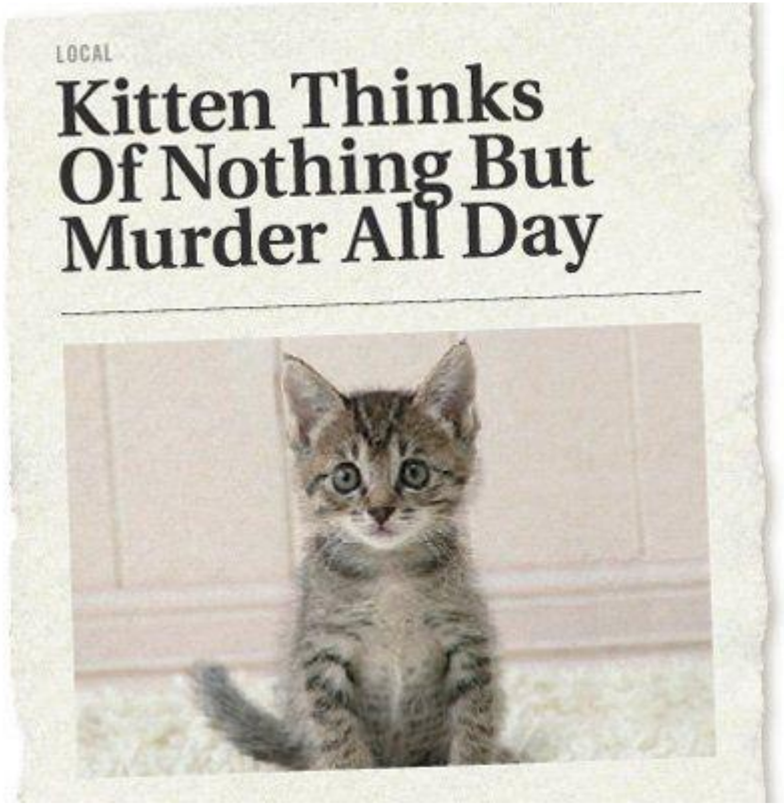
User Management

Styles & Layouts

Cat Libraries

Dog App

Dog Libraries



LOCAL

Kitten Thinks Of Nothing But Murder All Day

© The Onion