Best Practices

Decoupled CMS Architecture in Government

September 2020

amazee.io

About me

Justin Winter

Senior Software Engineer

Justin is a programmer, technologist, and Certified Drupal Developer. He works across the technology ecosystem on projects ranging from Drupal to React, Nodejs, and DevOps.

Email justin.winter@amazee.io

Agenda

- Government challenges
- Proposed solution
- Potential results
- Risks and concerns
- Hosting and deployment
- Where to go from here?
- QA

What’s the problem?

Drupal is complex, hard to learn, complicated to deploy and scale.

September 2020

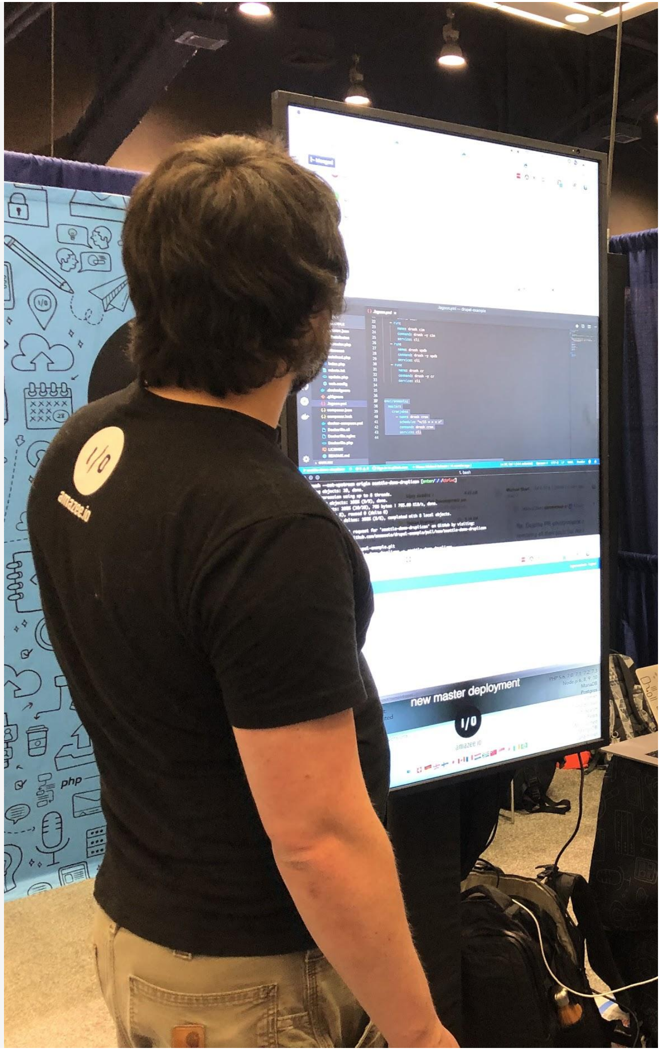
I/O

amazee.io

DRUPAL GOV CON

Government has a unique set of challenges

- Resource Allocation
- Training and Developing Skill Sets
- Open Source Community Integration
- Project Quality and Speed/Velocity
- Hosting/Scaling
- Budget Constraints



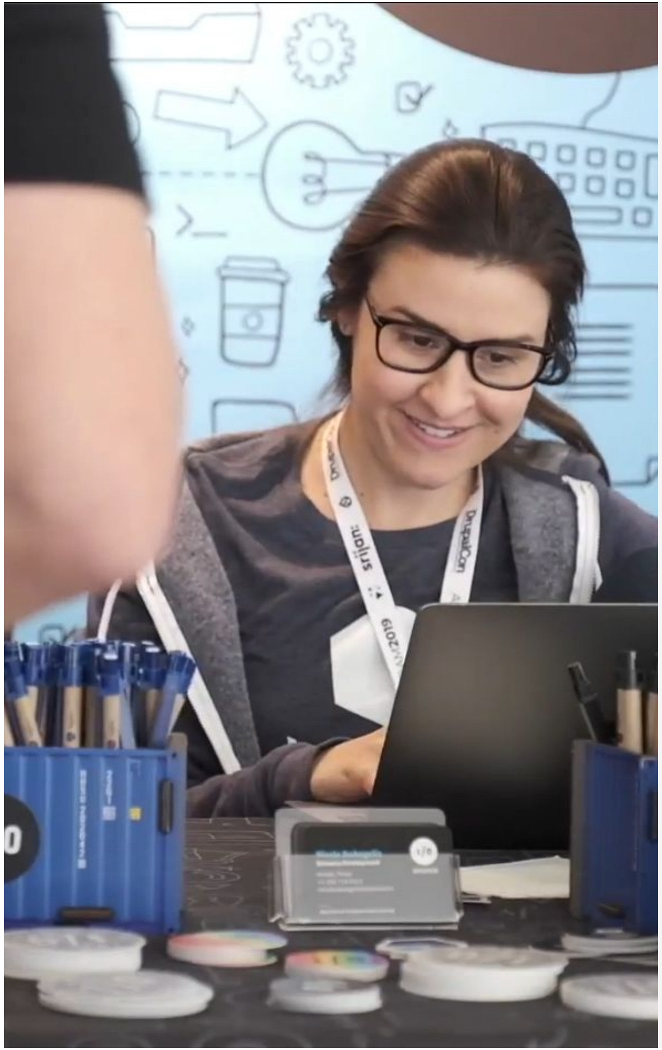
What's the solution?

Use Drupal for it's strengths, leverage front-end frameworks via an API, and deploy via containerized hosting.

September 2020

I/O
amazee.io

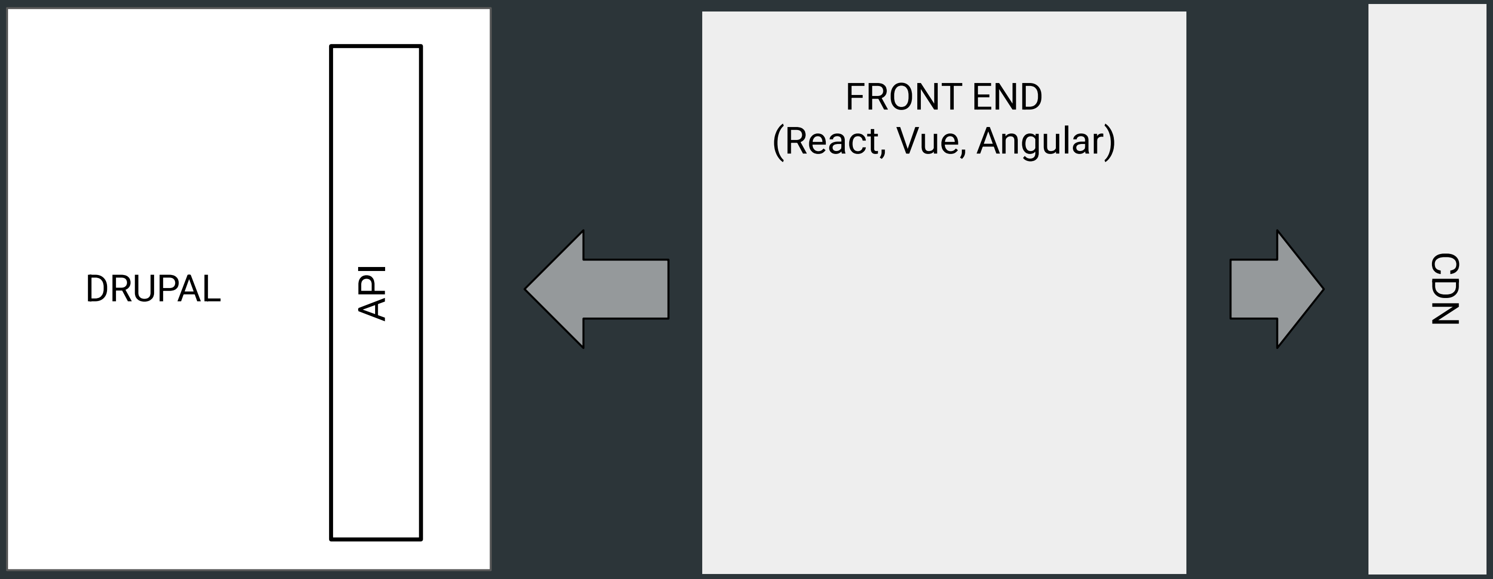
DRUPAL
GOV CON



Solution

- Drupal Strengths
  - Complex role based authoring,
  - Working with large authenticated user base,
  - Data architecture/modeling,
  - Configuration management,
  - Security,
  - Complex integrations,
- API First
  - GraphQL
  - JSON API
- Frontend Frameworks
  - React, Vue, Angular
- Containerized Development, Testing, and Hosting
  - Lando
  - Lagoon

Decoupled FTW!



DRUPAL

API

FRONT END
(React, Vue, Angular)

CDN

What are the benefits?

When you separate out the component pieces of a system, each piece of that system can be better managed, trained, scaled, tested, and swapped out as necessary.

Team

- Promotes Skillset Specialization
- Scaling
- Training

Technical

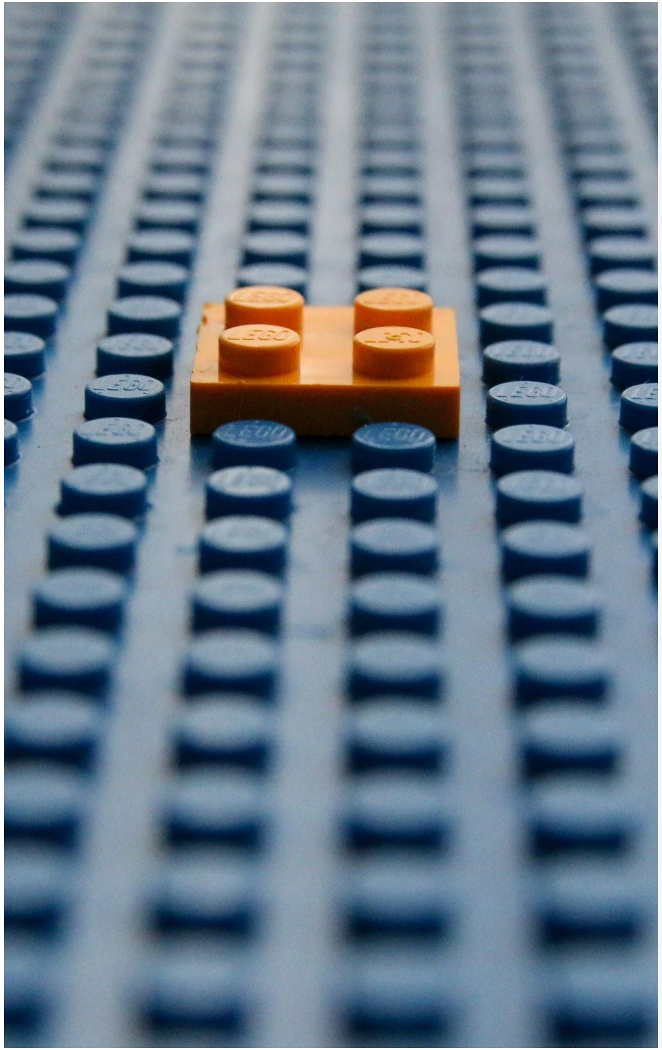
- Future compatible
- Scaling infrastructure
- Testing
- Local Development
- Architecture

Risks/Concerns?

- Ecosystem Maturity
- Fear of Change
- Increased Technology Stack
- Cross Team Communication
- Accessibility
- SEO/SSR
- Caching
- Hosting
- Technical Leadership

Development, Testing, Hosting, and Deployment

- Containers, Containers, Containers
- Development
  - Lando
  - Predefined Recipes (LAMP + NodeJS)
- Testing
  - Jest, Cypress
    - Including API tests
- Hosting/Deployment
  - Containerized hosting (i.e. Lagoon)



Where to go next?

- Give it a try!
- Druxt.js - Great working example and code
  - Drupal Backend with Vue front-end
- Amazee.io for more info on cloud and managed hosting solutions

Thank you

Justin Winter

Senior Software Engineer

justin.winter@amazee.io

amazee.io