


Benefits and pitfalls of integrating Drupal into a decoupled, static site build process for VA.gov

Presenters:

Elijah Lynn, VA.gov CMS DevOps Lead @ 

Robbie Holmes, U.S. Digital Service Alumni & VA.gov CMS Technical Product Owner @ 

Introduction | History | Why and Assumptions | Benefits and Pitfalls

Introduction - Session Details

In this session we will discuss a bit about why and how we implemented the VA.gov CMS (Drupal 8) as a backend Content API to feed a decoupled, Metalsmith static HTML build and deployment system for the frontend.

We will also reveal some notable challenges that we faced in implementing this piece of the overall architecture of which some were expected and some which weren't. We'll discuss how we overcame these challenges, including discussion of tradeoffs that our solutions entailed. Finally, we will share some lessons learned and guidance for other teams who may be considering implementing a decoupled, static HTML frontend + Drupal CMS at other agencies.

History: How we got here (part 1)

In 2015, Vets.gov was created as a greenfield project initiated by a U.S. Digital Service team working within the VA to allow veterans to track, manage and apply the benefits that they had earned.

1. Technology stack for vets.gov
 - a. Initially built with Jekyll (static site generator)
 - b. Refactored to Metalsmith + Liquid templating
 - c. React & Redux for dynamic interactions
 - d. APIs were developed behind vets.gov in Ruby on Rails (React apps consumed)
2. Frontend build process for vets.gov
 - a. Started with bash scripts and Travis CI
 - b. With the move to React we required NodeJS & Webpack
 - c. We then refactored to Metalsmith + Liquid templating replacing Jekyll
 - d. Jenkins replaced Travis CI as orchestrator of the build, test, release and deployment to S3

History: How we got here (part 2)

In 2018, Vets.gov and VA.gov went through a brand merger that brought forward the human centered design and modern engineering best practices from vets.gov to va.gov, the home of all initial interactions that veterans have with the VA.

1. Existing feature set and functionality from Vets.gov moved into VA's digital front door
2. This created the need for a unified system and content strategy, taking the existing systems from both vets.gov and va.gov and create a single editorial experience and workflow
3. Coming out of that was a decision to integrate an open source CMS into this new, unified system

Procurement

1. USDS partnered with the technology acquisition center with the VA to create a unique procurement opportunity in late spring of 2018.
2. Contract was awarded and kicked off in October of 2018

History: Where we are! (part 3)

In March of 2019, VA.gov CMS launched and this included

1. Launched the Pittsburgh Veterans Affairs Medical Center (VAMC)
2. Migration of large portion of the GitHub Content repository to Drupal
3. Refactored and integrated the GraphQL JSON data into the MetalSmith build process.

Current State

1. Migrating from GraphQL to Drupal Tome Sync (static JSON export)
2. Launching 17 additional VAMCs based the work done for Pittsburgh
3. Contributed 4 modules to Drupal.org and many other issues and bug fixes

Why: Problems we were trying to solve

1. Create a unified content editorial workflow aligned to a cohesive content strategy
 - a. Remove GitHub content repository from the editorial workflow
 - b. Replace an existing, inflexible CMS solution for VA.gov (TeamSite)
2. Contract requirement was that the new CMS was open source
3. Existing site at VA.gov needed to remain up and active throughout development lifecycle of new Drupal CMS
4. Existing editorial workflows needed to continue until the CMS was replaced or migrated
5. Existing static site build process could not be rewritten, Drupal had to plug into the current build process

Assumptions: Static Site / Decoupled Drupal

1. A fully rendered, static file website that is hosted on AWS S3 so that it will be secure, fast & reliable
2. Drupal only as Authoring Experience & not rendering/serving live traffic will be less complicated
3. CMS as only a Content Repository so that no veteran facing theming/UX work necessary
4. Authoring experience and website build process are discrete with two separate ownership models, thus scaling teams would be easier

Benefits & Pitfalls

Assumption: A fully rendered, static file website that is hosted on AWS S3 so that it will be secure, fast & reliable

Benefits:

- No Distributed Denial of Service (DDoS) attacks
- Static/Flat Files, so no server-side caching system needed
- Security updates for the frontend are handled as part of the build process not on a running system
- Performance (Time to first byte is solely based on S3 response time)
- Archiving/Auditing is simplified

Pitfalls:

- Could not instantly publish urgent content (breaking news, alerts etc)
- Now we had two distinct security review processes, one for CMS and one for Frontend build process
- Deployment times increased kept increasing as page count grew larger

Benefits & Pitfalls

Assumption: Drupal only as Authoring Experience & not rendering/serving live traffic will be less complicated

Benefits:

- We were able to have a highly customized authoring experience/workflow based on role/region within the VA
- Drupal's menu system improved editing and consistency of static site menu system
- Leveraging Drupal's redirect system simplified how redirects are created and managed

Pitfalls:

- No native Drupal preview meant we had to create our own preview process
- Validating internal and external links was complicated
- Utilizing Drupal's menu system to create a Mega Menu was complicated
- Redirects handled outside of the native Drupal menu system

Benefits & Pitfalls

Assumption: CMS as only a Content Repository, no veteran facing theming/UX work necessary

Benefits:

- Content is its own product
- Plugin to an existing build system seamlessly
- Authoring experience decoupled from output and rendering

Pitfalls:

- Drupal content output extremely tightly coupled to rendering in reality
- Conway's Law in action (teams, structure, contracts)
- Unclear who owns problems within the end-to-end system

Benefits & Pitfalls

Assumption: Authoring experience and website build process are discrete with two separate ownership models, thus scaling teams would be easier

Benefits:

- Frontend team did not need to know Drupal
- CMS team did not need to understand the frontend templating and build process
- Both teams could move independently and scale up easier
- Discrete ownership meant both could deploy independently

Pitfalls:

- Frontend team needed to know the data model and any subsequent changes to it
- Two separate repositories, one for CMS and one for FE, had different deployment cadences
- Deployments were required to be in sync between both repositories

Thanks, any questions?

Elijah Lynn - Engineer @ [agilesix](#)



[@ElijahLynn](#) | [@agilesix](#)

Robbie Holmes - VP of Product Development @ [STSI](#)



[@robbiethgeek](#) | [@stsiinc](#)

Appendix: Links to all the things

- **Drupal.org - Department of Veterans Affairs**
 - <https://www.drupal.org/departement-of-veterans-affairs>
- **Backend repository (Drupal 8 CMS)**
 - <https://github.com/departement-of-veterans-affairs/va.gov-cms/>
- **Frontend & Build Process repository (Metalsmith, Liquid Templates & React apps)**
 - <https://github.com/departement-of-veterans-affairs/vets-website/>
- **Drupal Tome (Tome Sync) Module**
 - <https://www.drupal.org/project/tome>
- **Federal Open Source Policy (not in recording)**
 - <https://obamawhitehouse.archives.gov/blog/2016/08/08/peoples-code>

Appendix: All VA Related Sessions

1. Lessons learned re-platforming VA.gov on Drupal in 12 months

Thursday September 24 2:30 pm ET

<https://www.drupalgovcon.org/2020/program/sessions/lessons-learned-re-platforming-vagov-drupal-12-months>

2. Benefits and pitfalls of integrating Drupal into a decoupled, static site build process for VA.gov (this session)

Friday September 25 10 am ET

<https://www.drupalgovcon.org/2020/program/sessions/benefits-and-pitfalls-integrating-drupal-decoupled-static-site-build-process>

3. Putting veterans first in the design of a pilot hospital site on VA.gov, the largest hospital system in the US

Friday September 25 1:30 pm ET

<https://www.drupalgovcon.org/2020/program/sessions/putting-veterans-first-design-pilot-hospital-site-vagov-largest-hospital>