

Consideration Factor		Small Batch Publishing	Pro/Con	Fully Coupled Drupal	Pro/Con	Decoupled Drupal	Pro/Con	SSG with Incremental Build (e.g. Gatsby)	Pro/Con
Description		Small batches of pages are built and published to VA.gov immediately upon content updates.		Drupal directly serves VA.gov pages to visitors.		VA.gov pages are rendered by a front-end application querying Drupal for content in real time via Lighthouse.		VA.gov pages are rendered by a OOTB static site generator with incremental build capability.	
Documentation									
	Implementation Unknowns/Uncertainty	Low uncertainty (pending PoCs)	PRO	Few unknowns, as several members on the team have implemented high performance Drupal sites. Any uncertainty will lie in uncovering logic within the build for pages which do not exist within Drupal.	PRO	There are many unknowns with the Lighthouse API. Prototyping will be required to discover if its capabilities will fulfill our needs.	CON	Unknown in the absence of a PoC.	NEUTRAL
	LOE Uncertainty	Using a phased approach where incremental builds per content type and field type will reduce the uncertainty of initial implementation.	NEUTRAL	We did a POC to bring some liquid templates into Drupal to help reduce this risk. The primary uncertainty is with ATO approval.	NEUTRAL	The amount of effort required varies greatly depending on the capabilities of Lighthouse. We know we have to increase the capabilities of our infrastructure but the extent depends on Lighthouse.	CON	Unknown in the absence of a PoC.	NEUTRAL
	Security risks & mitigations	No change	PRO	Drupal Patches will become more urgent. A web application firewall and internet gateway will have to be managed and secured. Security practices will be created to mitigate the risk	CON	Exposing the Drupal API to the public involves lots of security risk.	CON	No change	PRO
	Reliability risks & mitigations	Risk: Introduces risk that a change to one piece of content may require rebuilding other content that is not detected and scheduled to be built Mitigation: PoC work to compare incremental builds with full builds, schedule daily off-hours full content build	CON	Requires running web, application, cache and database servers to be setup in a High Availability mode. These practices are well known.	NEUTRAL	Along with the items listed in the "Fully Coupled Drupal" column, Lighthouse is an extra layer of infrastructure to be managed.	CON	Mitigation: extensive testing and validation will be required.	CON
	Performance risks & mitigations	Work to incrementalize current build may be leveraged to parallelize full content builds. Risk remains that increasing numbers of users & published content will cause increasingly slowed builds. This should be viewed as a temporary stopgap solution.	CON	Serving content directly from Drupal can cause high database load, affecting performance. Mitigation: A well managed reverse proxy cache (e.g. Varnish) using React and/or ESIs (Edge Side Includes) to reduce the load on the database.	NEUTRAL	Serving content directly from Drupal can cause high database load, affecting performance. Mitigation: A well managed reverse proxy cache (e.g. Varnish) using React and/or ESIs (Edge Side Includes) to reduce the load on the database.	NEUTRAL	Risk remains that changes to any global content templates will incur full site builds. This could potentially be mitigated by componentizing frequently updated global site sections. (e.g. header, banner alerts)	CON
	ATO implications	No change	PRO	ATO approval will be required.	CON	No if using Lighthouse, Yes if using Drupal API	PRO	No change	PRO
	Infrastructure Cost Implications	Low risk - possibility of new usage of AWS Lambda functions	PRO	All services will need to be setup in a Highly Available architecture, significantly increasing hosting cost. New infrastructure including varnish and networking, security applications will also be added.	CON	All services will need to be setup in a Highly Available architecture, significantly increasing hosting costs. New infrastructure including varnish and networking, security applications will also be added.	CON	Likely to remain very similar	PRO
	Co-Development/Collaboration Impact on other teams	Collaboration will be required with Platform Ops and vets-website code owners	NEUTRAL	Ops would need to be involved to setup Internet gateway and make sure our services are highly available. Product teams will be developing Drupal templates rather than liquid templates. React widgets will need to be refactored to pull from Drupal API from dynamic content listings. How product teams work will change completely. Change management needs to be considered early and often during the project.	CON	There is a heavy dependency on the Lighthouse team for API development and the front end teams to implement changes to react widgets.	CON	Collaboration will be required from all involved teams, as a total overhaul of the build pipeline and templates would be required.	CON
	Long Term Maintenance Plan and Ownership	CMS team would own more of the incremental stage of the build process	NEUTRAL	The entire content publishing pipeline would be owned by the CMS. The team would need to be expanded to support the additional usage.	PRO	Initial ownership would not change. Eventually, a SPA would be put in place but this would be a different project.	NEUTRAL	Long term maintenance requirements should be reduced as a good bit of custom functionality would be replaced with community-supported software.	PRO
	Handling of markdown content	No change	PRO	All static content would be migrated to the CMS.	CON	All static content would be migrated to the CMS.	CON	Markdown content would almost certainly need to be replaced with content in the CMS.	CON
	Handling of dynamic content (react widgets)	No change, possibility to have react widgets pull from JSON files on S3 which can be generated by Drupal or a lambda/build layer.	NEUTRAL	Dynamic content would pull from the same APIs. Some of the widgets which work off of static JSON objects would change to using the Drupal API. The rendering of the react widgets would be handled by Drupal.	PRO	React Widgets would pull from Drupal directly and not require a build.	PRO	React widgets would likely need to be re-implemented and still require a non-direct path to query data from the CMS.	CON
	Skillset Availability and Requirements for Future Product Development on VA.gov	No change	PRO	Javascript experts would be added. Possible the FE tools team would help mitigate this new need.	NEUTRAL	No Change	PRO	TBD: required skillsets would depend on the product selected. (And likely inform product selection)	NEUTRAL
	Implementation Timeline	3-4 months for Phase 1	PRO	TBD, 9 month minimum.	CON	TBD, 9 month minimum.	CON	circa 9-18 months	CON
	Infrastructure Scaling Effort	Low	PRO	All Drupal infrastructure would have to be made highly available, along with implementing Varnish and Blue/Green deployments.	CON	All Drupal infrastructure would have to be made highly available, along with implementing Varnish and Blue/Green deployments.	CON	Low	PRO
	Instant Publishing	No, still a delay even if short	NEUTRAL	Yes	PRO	Yes	PRO	No, still a delay even if short	NEUTRAL
	Editorial Control over display	No Change	NEUTRAL	Yes, including the ability to use tools like layout builder	PRO	Unchanged	NEUTRAL	No Change	NEUTRAL
	Distinction between Content Publish and Content Deploy	Vastly reduces the time a deploy takes, leaving a distinction that is probably not important to editors	NEUTRAL	No distinction	PRO	No distinction	PRO	May significantly reduce the time a deploy takes, leaving a distinction that is probably not important to editors	NEUTRAL
	Preview content	Currently single page with the preview server. A holistic solution, while possible still needs to be discovered.	NEUTRAL	Pages may be previewed using production templates.	PRO	A new solution would be required for page preview.	CON	Dependent on the product selected, a preview server product exists for Gatsby.	NEUTRAL