Reliability | Velocity | Innovation

Your DXP, delivering results like never before!

Slow updates aren't an option. Next.js alone can hold you back—Spark delivers real-time content updates, from your CMS to your user effortlessly.

Spark

Challenges with Publishing
ISR and On-Demand ISR

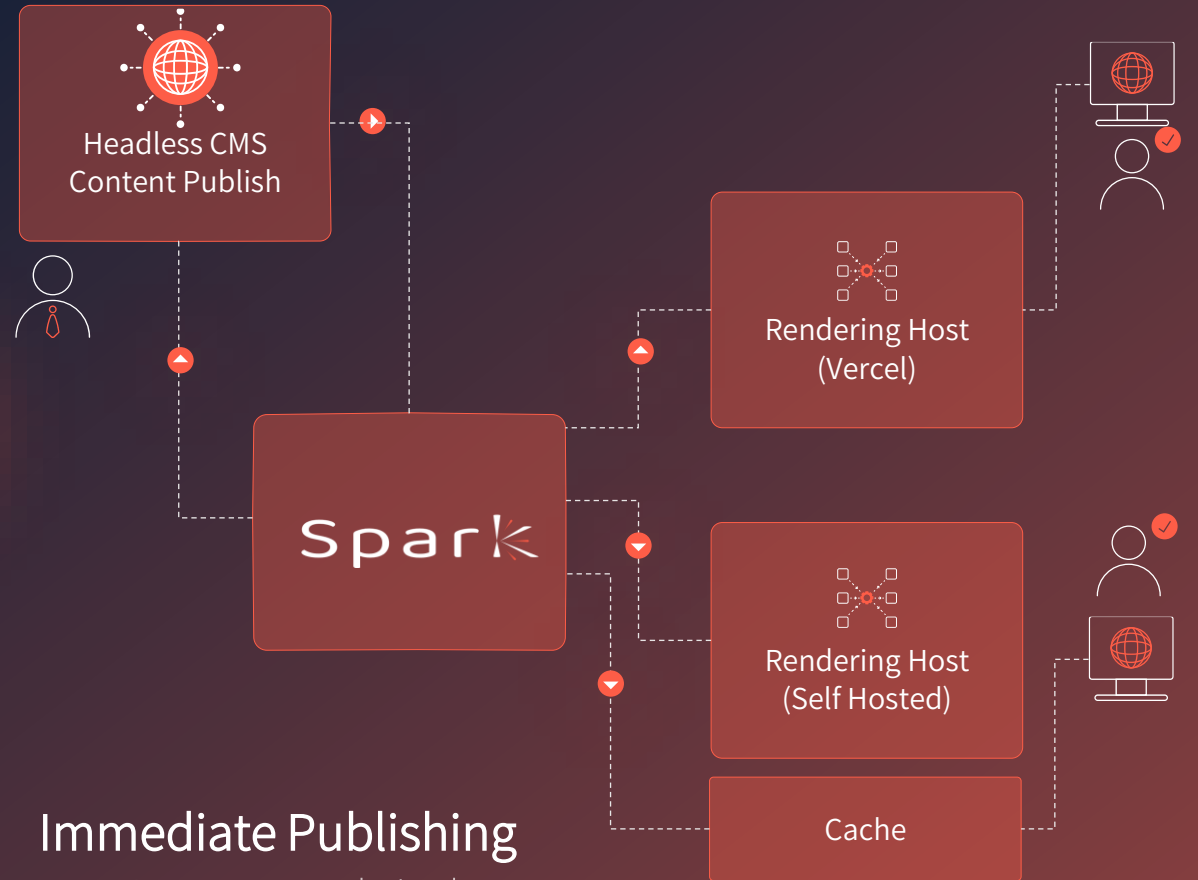- Multi-site
- Throttling Limits
- Dependency Mapping
- Un-publish and Page Removal
- Default Page List
- Make it live NOW!!

Headless CMS
Content Publish

Spark

Immediate Publishing
Across any rendering host

Rendering Host
(Vercel)

Rendering Host
(Self Hosted)

Cache

What is On-Demand ISR?

- Traditionally, Static Site Generation (SSG) creates pages during the build process. When a Next.js application is built, it pulls data from headless CMS platforms like Sitecore XM Cloud, generating pages based on published content at that time. This method works well for static content that rarely changes but falls short for dynamic CMS-driven websites where content updates are frequent.

- Incremental Static Regeneration (ISR) partially solves this by revalidating pages based on a set timeout (e.g. 30mins or even hours). When a page request exceeds this limit, Next.js serves the cached version while refreshing the content in the background. The next visitor then sees the updated version. While helpful, this doesn't offer real-time control, leaving gaps when content must be live immediately, such as for news articles, product launches, or urgent announcements.

Spark

Why On-Demand ISR?

- On-Demand ISR gives content editors the power to trigger updates instantly. Instead of waiting for a timeout, the CMS sends an event (via a webhook) to revalidate specific pages. This ensures that:

- New content is visible immediately, even to the first visitor or search engine crawlers like Google

- Editors control exactly when pages go live

- Time-critical content reaches the audience without delay

Spark

Challenges in Enterprise Environments

- While powerful, implementing On-Demand ISR at scale can be complex. Challenges include:

- Avoiding API throttling when rebuilding multiple pages

- Identifying and revalidating dependent pages

- Managing page deletions or unpublishing

- Providing fallbacks for key pages during large-scale updates

- Dealing with multi-site support

- Managing cache clearance and running post re-validation events

How Dataweavers **Spark** Simplifies On-Demand ISR

- At Dataweavers, we've productized an On-Demand ISR (ODISR) service that handles this complexity for you. Our solution ensures instant publishing with minimal effort:

- Register our API as the publish webhook in your CMS

- Our service manages the logic, deciding which pages need revalidation

- Use our provided NPM package to connect with your Next.js app, ensuring pages update instantly

- With Dataweavers ODISR, content goes live the moment it's published, delivering fast, reliable updates without the technical headaches.