uniform

Uniform whitepaper

Maximize the value of Sitecore plus a CDN

Enhance web or app performance without a major overhaul by adopting Uniform

Q2 2022

Why should your brand use a content delivery network (CDN) for your website or app? Performance is a key benefit: A CDN can significantly improve performance by delivering content from edge servers that are geographically close to users. A CDN can also help you scale quickly to handle spikes in demand. And it can enhance reliability and security so content remains available even if there are hardware issues or breaches elsewhere in your network.

But if you are using Sitecore to deliver and manage digital experiences, you might not be realizing the full benefits of a CDN. Many Sitecore customers use a CDN, but in a very limited way: The CDN is used only for caching static assets (like images) while Sitecore CD serves content. Others depend solely on Sitecore Content Delivery (CD) to serve assets and content, without a CDN.

While many customers depend entirely on Sitecore's CD architecture, there are challenges. Sitecore's origin-based request-handling architecture handles each visitor request individually with a Sitecore server (the "origin"), which then generates the page. This architecture can't provide the same levels of performance, scalability, security, or reliability as a modern CDN.

Uniform can help you take full advantage of the performance, scalability, security, and reliability benefits of a CDN when using Sitecore, without requiring a major overhaul of your environment. Uniform lets you modernize your Sitecore implementation and make more of your investments. With Uniform, get more out of your existing Sitecore infrastructure while taking advantage of true CDN performance and scalability.

Make the best of Sitecore and a CDN with Uniform

Uniform augments Sitecore and enables you to realize the full value of a CDN while keeping current environment and processes in place.

Maintain existing workflows: With Uniform, you can preserve your existing, familiar content management experience for marketers and other business users. Uniform decouples delivery from the Sitecore CD instances—the business user workflow is not changed in any way.

Reduce costs: Decoupling delivery of content from Sitecore CD instances enables you to use Sitecore servers more efficiently. You can use fewer servers even as the volume of visitors grows. Consequently, you can significantly reduce costs.

Streamline front-end development: Front-end frameworks such as Next.js, Gatsby, and Nuxt allow developers to build front ends faster than ever before. Uniform enables developers to use those tools to build front ends for Sitecore sites without having to upgrade your Sitecore version or convert the entire front end all at once. You can modernize your front end incrementally, one component at a time. And if your most efficient front-end development is done using Sitecore MVC or Sitecore Experience Accelerator (SXA), you can continue using your preferred tool - and Sitecore version - without losing any of the performance or scalability benefits that Uniform offers.

Maximize performance: The primary benefit of a CDN is the ability to deliver content to web or app users fast, regardless of where in the world they are located. Moving content delivery to the CDN lets you take advantage of a global network whose sole purpose is to do that well. At the same time, using a CDN eliminates the performance impact of resource-hungry processes, such as publishing, on the visitor experience, because the content delivery process is entirely disconnected from Sitecore.

Gain instant, global scalability: The Sitecore content delivery process involves numerous backend application servers and databases. As the load increases, Sitecore requires more servers. It takes time for these servers to come online, and until those servers are available, visitors experience diminished performance. Uniform eliminates the need for backend application servers and databases to handle the delivery process, avoiding these cold-start delays and enabling you to scale immediately.

Meet stringent SLAs, bolster resilience: With no backend application servers or databases required to power content delivery, uptime and resilience are dependent on the CDN alone. You can better maintain availability, even if there are outages elsewhere in your network. And a Sitecore server outage will never result in your site being unavailable.

Simplify the content delivery architecture: Removing application servers and databases from the delivery layer reduces architectural complexity. With fewer components in the architecture, there are fewer systems to buy and fewer places to look when things break.

Progress at your own pace: With Uniform, there is no need to upgrade your existing Sitecore implementation: Uniform supports version 8.2 and above. Similarly, there is no need to change your front end: Uniform supports sites built using MVC, SXA, and JSS. You can progress at your own pace, realizing benefits quickly and expanding when you are ready. You can use Uniform for a single page and add more pages later. Or modernize a single component and add more components down the road. Working incrementally, first with a limited scope, enables your team to learn a new approach and experience some early gains while controlling risks.

Customize your adoption: Uniform gives you flexible adoption choices. You can employ a traditional configuration where Sitecore handles page building. In that case, you can still minimize the size of the CD environment while increasing use of the CDN. Or you can use a Jamstack architecture - without losing tracking and personalization - and eliminate the CD environment entirely.

Boost performance using Jamstack for incremental static generation

Uniform enables you to employ Jamstack with Sitecore to take advantage of static site generation and eliminate the need for the Sitecore CD environment without losing the tracking and personalization capabilities you depend on. Using static site generation can improve performance and scalability, reduce system complexity, and lower the total cost of ownership.

Who is this for?

Using Jamstack and employing static site generation can work well if you are experiencing performance, scalability, or reliability issues with your Sitecore CD environment. To use this approach, your front end must have been built using MVC, SXA, or JSS. You could be using either a non-personalized site (Sitecore XM) or a personalized one (Sitecore XP).

How does it work?

With a Jamstack architecture, site pages are generated before a visitor views them. This process is called static site generation. Once an author edits and then publishes content, Uniform triggers the static site generation process, which involves reading content from Sitecore. The output from the process is a set of files, which Uniform deploys to the cloud or network storage, or CDN of your choice.

Even with the static site generation approach, publishing can be slow with large Sitecore sites. Uniform improves performance by optimizing the task of reading content from Sitecore. These optimization capabilities from Uniform are the result of many years of real-world experience helping organizations Jamstack-enable their large and complex Sitecore sites.

Supporting leading front-end technologies

Uniform generates a static site from Sitecore content using a static site generator, such as Next.js, Gatsby, or Nuxt. These generators are built on front-end technologies such as React (for Next.js and Gatsby) or Vue.js (for Nuxt).

If your Sitecore front end is built using JSS, you are already well prepared to use a static site generator: You are almost certainly building front-end components with a technology supported by at least one static site generator.

How will this affect your front end?

Sitecore offers almost endless flexibility for how front-end components are built and what they do. Uniform hides most of the technical details involved with Jamstack-enabling a Sitecore site, but Uniform might require changes to the front-end technology. Most front-end components are fully compatible with Uniform without any changes. But components that depend on server-side logic that must run for each page view could require changes.

Uniform’s Customer Success team is always available to help you determine how best to handle server-side logic. If you have a personalized site and you don’t want to make any changes whatsoever to the front end—or you have immediate performance, scalability, or reliability concerns—you can still use Uniform.

But if your front end is built using an ASP.NET-based front-end technology such as Sitecore MVC or Sitecore SXA, there is no static site generator that can use your front-end components directly. In that case, Uniform enables your MVC or SXA components to be used with a modern static site generator. With Uniform, you can preserve your existing front-end components while still realizing the benefits of a modern static site generator.

Using Uniform provides the added benefit of allowing you to incrementally modernize your front end, component by component. Since Uniform maps components defined in Sitecore to components used by the static site generator, you can create a new front-end component using a modern front-end technology (such as React) that replicates a Sitecore rendering (for example, an MVC view). Uniform will ensure the static site generator uses the new front-end component.

Support for leading CDNs

After the static site is generated, the resulting files must be deployed somewhere so visitors can access the site. Files are usually sent to cloud or network storage with a web server in front, or directly to a CDN. Uniform triggers this deployment as a part of the Sitecore publishing process.

Uniform supports virtually any CDN, or cloud or network storage, including:

- Akamai NetStorage
- Cloudflare Workers KV
- AWS S3
- Netlify
- Vercel
- Tencent Cloud

Is static site generation only for static sites?

The term “static site generation” can be a little misleading. It doesn’t mean that the site is static. It means that when the visitor views the page, no dynamic page generation logic needs to run on an application server and no data needs to be retrieved from a database.

Deliver up-to-date content from the edge with integrated cache purging

Uniform simplifies cache purging, helping to ensure that the CDN is delivering the most up-to-date content to users without substantially affecting performance.

Who is this for?

Cache purging is essential for delivering the latest content to visitors. Until the cache is purged, visitors will see only old content. If you are using a traditional CDN, such as Akamai Ion or Cloudflare CDN, integrated cache purging is required, because these solutions do not know when content changes unless you tell them. Solutions such as Netlify and Vercel, which can act as CDNs but also serve as the origin of content, do not have this strict requirement.

How does it work?

With Uniform, there are multiple ways to initiate the cache purging process. For example, you can automate the purge, having Uniform send purge instructions when publishing finishes. You can also manually trigger the cache purge from within the Sitecore Content Editor—Uniform adds a tab to the Content Editor ribbon that provides a button for clearing the cache. Or you can trigger the purge programmatically, using the Uniform API.

Support for leading CDNs

Uniform's integrated cache purging supports several leading CDNs, including:

- Akamai Ion
- Cloudflare CDN
- AWS CloudFront

Enhance Sitecore personalization at the edge

Personalization at the edge is beneficial if you are experiencing performance, scalability, or reliability issues with your Sitecore XP CD environment. Uniform can help you make the most of the Sitecore personalization that you've already implemented.

Who is this for?

You might have implemented Sitecore personalization in the past and were happy with the content authoring experience. But when you deployed personalization to production, you might have found your site suffering from performance, scalability and stability problems, which forced you to disable personalization. Using Uniform to move the execution of personalization configured in Sitecore to the CDN can help you overcome those issues. You can use this approach whether or not you are using a Jamstack architecture.

Accelerating publishing by eliminating Sitecore CD with Uniform

A tourism company wanted to improve the user experience and streamline the content delivery architecture used for a Jamstack website that is available to visitors in mainland China.

Challenges and goals

- Eliminate an expensive Sitecore CD and SQL Server infrastructure in Hong Kong that was serving mainland China
- Reduce lengthy publishing time caused by latency between a United Arab Emirates (UAE) data center and remote publishing target in Hong Kong
- Scale up to support high traffic in China, delivering subsecond load times
- Repurpose the China site for visitors in UAE, using distinct CDNs
- Customize site behavior to use cloud services available in China

Solution

The organization decoupled content delivery from Sitecore with Uniform. The full Sitecore JSS site was exported and rendered within the UAE data center and deployed to Tencent Blob Storage with a CDN front end.

Benefits

- Improved performance and efficiency by using local static site generation and bulk upload
- Ensured fresh content for visitors by automating CDN cache purging when new static assets and pages are uploaded
- Maintained workflows through full integration with the Sitecore publishing process
- Increased scalability of pre-rendered pages using Tencent CDN points of presence (PoPs)
- Reduced infrastructure costs, simplified IT by eliminating the need for Sitecore or SQL Server infrastructure to serve pages
- Dynamic functionality is still available, it just runs on the client or on the CDN

How does it work?

You configure personalization and tracking within Sitecore. Personalization instructions are converted into a format that can run on the CDN. Uniform then extracts personalization and tracking instructions and includes them in the pages that are generated. Personalization instructions are converted into a format that can run on the CDN. Tracking is handled on the client by the Uniform tracker script.

Tracking activity can be dispatched anywhere. You can dispatch tracking to one or many targets, including Google Analytics, Segment, Sitecore Experience Database (xDB), or virtually any other target. This allows you to capture visitor activity in the system – or systems – you depend on to understand visitor engagement. In addition, personalization is integrated with tracking so personalization activity is tracked.

Support for leading CDNs

In providing personalization at the edge, Uniform supports leading CDNs, including:

- Akamai ESI
- Cloudflare Workers

Implement Uniform personalization at the edge

Uniform can also help you experience the benefits of personalization at the edge without using Sitecore personalization capabilities.

Who is this for?

This approach enables you to add personalization that is distinct from Sitecore personalization. You might want to use intent-based personalization instead of the event-based personalization that Sitecore offers. Alternatively, you might want to personalize using attributes and activity from other systems without having to build and maintain custom integrations (like with Sitecore CDP). Or you might be planning to migrate away from Sitecore or at least want to avoid increasing your Sitecore investment.

How does it work?

With this approach, you configure personalization and tracking within Uniform. Personalization and tracking instructions are extracted from Uniform and included in the pages that are generated. Personalization instructions are converted into a format that can run on the CDN. Tracking is handled on the client.

Support for leading CDNs

Using this approach instead of existing Sitecore personalization, Uniform supports an even broader array of CDN services, including:

- Akamai EdgeWorkers
- Cloudflare Workers
- AWS Lambda@Edge
- Vercel Edge Functions

Make the most of Sitecore and your CDN with Uniform

Using a CDN can help you significantly improve performance, scalability, reliability, and security for your website or application. But for too many Sitecore customers, the potential benefits of a CDN have been out of reach because of Sitecore's origin-based request-handling architecture. Uniform enables you to overcome those limitations and maximize the benefits of a CDN with Sitecore.

Uniform can help you whether Sitecore will continue to be a central element in your site delivery architecture or whether it will play a more limited role. With Uniform, you can continue to derive value from your existing Sitecore investment even as you change your technical direction. If you want to preserve as much of your existing architecture as possible, Uniform enables you to offload significant amount of work off your CD instances. If you are considering a change in technical direction, Uniform can be a bridge to modernization. In both cases, Uniform's support for incremental adoption enables you to introduce changes at your own page - avoiding disruptive change or replatforming.

Ready to learn more?

Connect with us today to learn more about how Uniform can help you make the most of your Sitecore and CDN investments
- visit **uniform.dev/demo.**