netlify

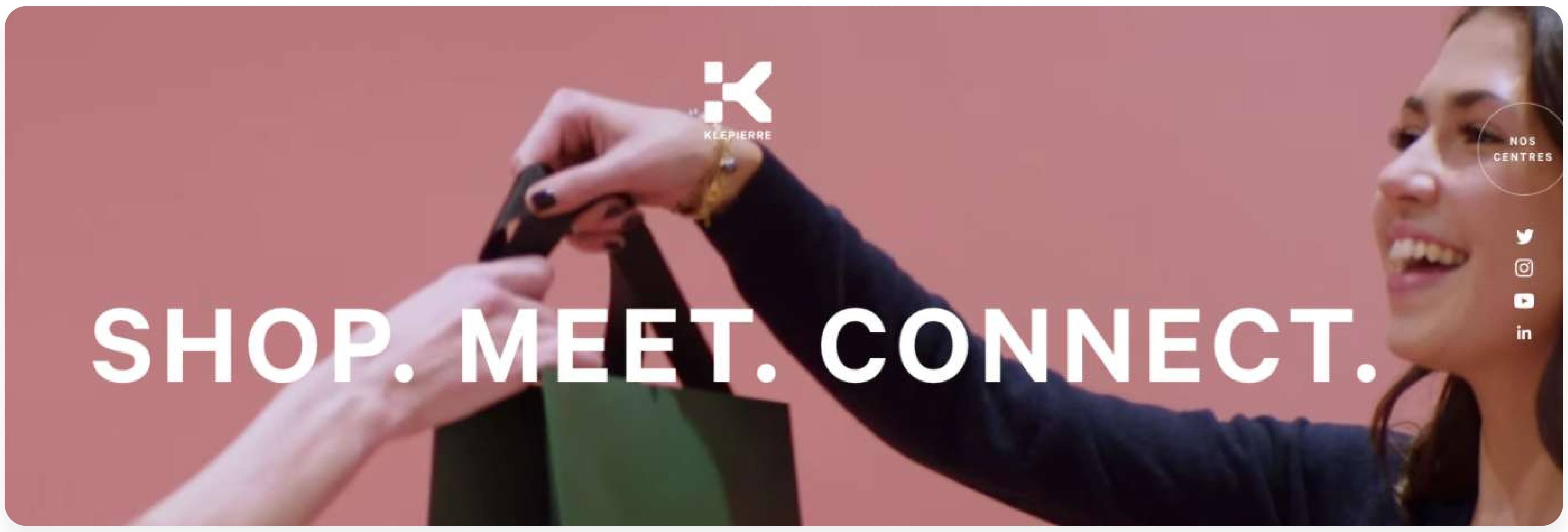
+

KLEPIERRE

Case Study: Klépierre's Journey to the Jamstack

European shopping mall leader embraces lightning-fast Headless Sitecore *(with predictable cloud spend)* on Netlify

Klépierre's Journey to the Jamstack



Faced with the challenges of the digital revolution and transformative shifts to the shopping experience, European mall operator Klépierre decided to undertake a massive overhaul of their web presence. Klépierre operates over 150 malls in 14 European countries, and has hundreds of content authors and stakeholders across their 98 web properties. Their existing Sitecore architecture worked well for content authoring, but Sitecore's heavy infrastructure didn't allow for the lightning-fast experience their customers expected.

Klépierre engaged the creative and technology agency **Proximity** to help undergo a global platform redesign, and develop **98 sites in just six months**. Sites needed to be **fast** while still image-heavy, **easy to use** for Klépierre's many stakeholders and content authors, and the entire project needed to happen within the agreed **budget and timeframe**.

After encountering performance and price barriers with their planned Sitecore redesign, Proximity made a discerning shift to their planned technical approach. With only 3 months left in the project, they rearchitected on the **Jamstack** model to dramatically cut their Azure costs. With a headless Sitecore CMS and Netlify as the distributed edge network (an advanced CDN), they massively reduced cloud spend while still delivering performant sites on the client's timeline.

While the initial impetus for the move to the Jamstack was cost-savings, the Proximity team soon realized additional benefits of this decoupled Sitecore architecture. Moving to Netlify also meant improved workflows, simpler cloud management as well as great website performance. This case study explores the technical decision-making behind the change in the architecture, and what the Jamstack has enabled for Klépierre so far.

Scaling Sitecore without scaling the cost thanks to the Jamstack

Proximity is a global agency with extensive knowledge of enterprise websites and data platforms. Their **portfolio** exhibits a proven ability to undergo massive content migrations, as well as deep expertise architecting multilingual, multi-country, and multitenant sites. They won Klépierre's RFP by showcasing their deep understanding of Sitecore as well as how to couple it with a customer data platform (Tealium), a CIAM (customer identity and access management) solution (Azure) and marketing automation (Selligent).

In the initial pitch, Proximity proposed a heavy Sitecore build that would allow them to:

- **Build 98 sites, fast.** The project needed to deliver 98 multilingual sites for 14 European markets in a matter of months. Klépierre needed a technical approach that would ensure on-time delivery while working with many stakeholders.
- **Keep content entry easy**. With hundreds of stores across their properties, Klépierre's sites needed to be editable by lots of stakeholders. Proximity's proposed Sitecore development allowed Klépierre to benefit from Sitecore content editing experience and its multi-tenancy features.
- **Deliver performant sites**. Klépierre's emphasis on an excellent digital experience mandated that sites be performant while still image-heavy and providing content localized to the customer.

After encountering performance and price barriers with their planned Sitecore redesign, Proximity made a discerning shift to their planned technical approach. With only 3 months left in the project, they rearchitected on the **Jamstack** model to dramatically cut their Azure costs. With a headless Sitecore CMS and Netlify as the distributed edge network (an advanced CDN), they massively reduced cloud spend while still delivering performant sites on the client's timeline.

While the initial impetus for the move to the Jamstack was cost-savings, the Proximity team soon realized additional benefits of this decoupled Sitecore architecture. Moving to Netlify also meant improved workflows, simpler cloud management as well as great website performance. This case study explores the technical decision-making behind the change in the architecture, and what the Jamstack has enabled for Klépierre so far.

Proximity's initial approach was similar to other enterprise projects they'd worked on involving Sitecore. They would hit these three goals by undergoing a major platform re-development using Sitecore's SXA component-based architecture. The infrastructure side of the team hit the ground running modernizing Sitecore, focusing on building a Identity Management system, and pipelines between the website to the CRM to the CDP.

But Proximity soon encountered a problem: cost. While this approach worked for Proximity's many clients with similarly scaled platforms, in the context of Klepierre, they also had to worry about the total cost of ownership on the IT side on behalf of Klepierre. With that in mind, considering Klépierre's desired 98 sites, the perspectives on the infrastructure cost over time was a concern. Scaling Sitecore to address the expected level of traffic would necessarily be an issue on the budget end. Sitecore was expected to be the content platform that feeds the websites but also many other touchpoints (Loyalty apps, chatbots, emails, in-mall digital terminals). On top of that, personalization was also awaited. If Proximity were to proceed with the regular approach, the Azure bills would grow significantly above the amount budgeted into their existing Sitecore package. Expensive overage costs would have to be expected.

The question soon became: how could Proximity scale Sitecore without scaling the cost? This is where the Jamstack came in.

> **"Scale changes everything. Performance, the way parallelization works, the way you handle your team, the way you deploy... Netlify delivers scale, redirections, is blazingly fast for speed of builds and deployment, and gives us 0 downtime.**
>
> **— Zyad Rujeedawa,** global VP of engineering and technical director, Proximity

Reality Check:
Moving Sitecore to the Jamstack

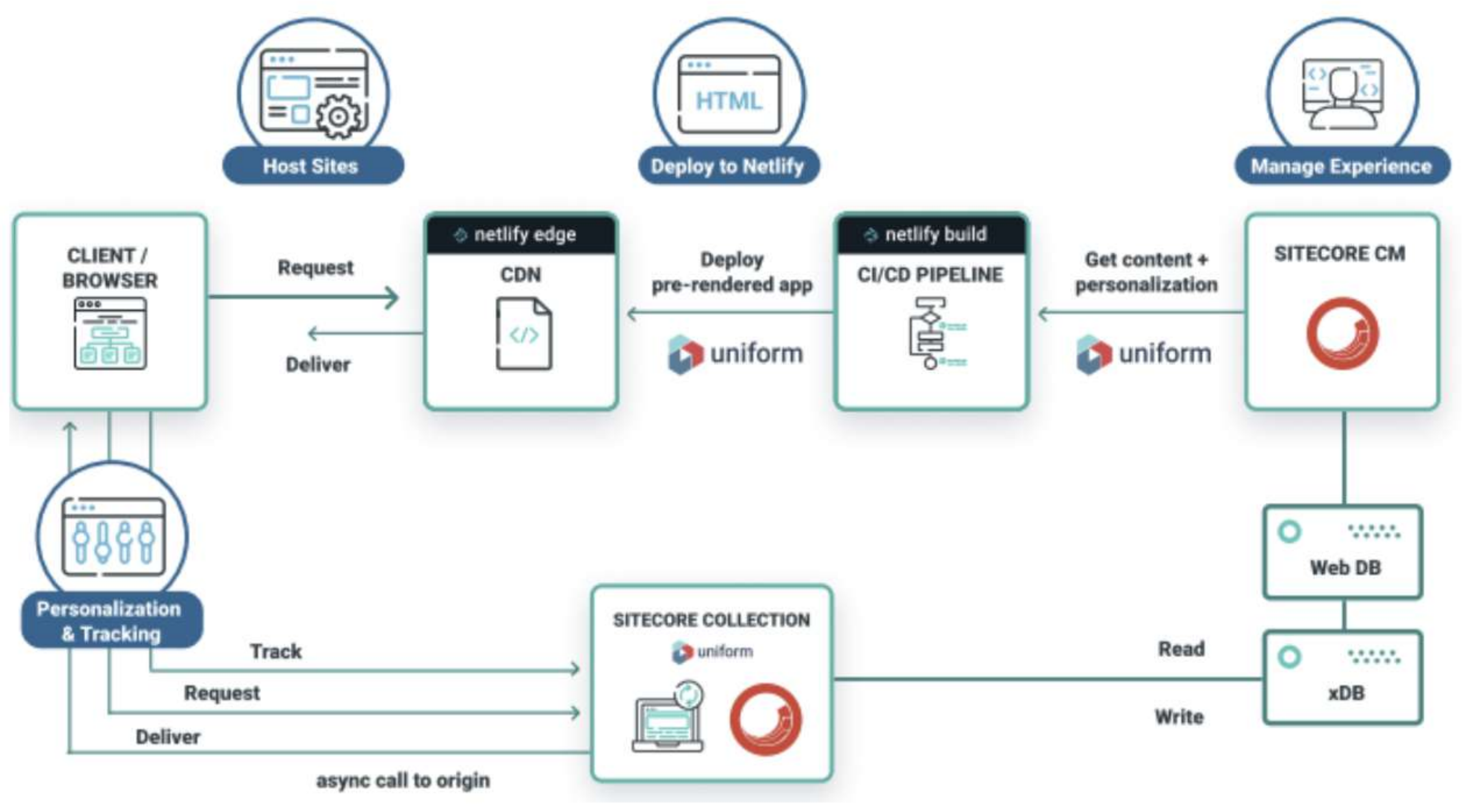
Proximity realized that a Jamstack model would offer Klépierre the cost-savings of a static site, while maintaining the rich features Klépierre needed. Whereas a monolithic Sitecore on Azure method would require purchasing--and managing--multiple servers, a Jamstack approach could instead pre-render files and serve them directly from Netlify's CDN.

> **We realized: you could cut your hosting costs in half if you take a Jamstack approach.**
>
> **— Jonathan Bobo,** CTO, Proximity

Now, they were faced with a new question: the Jamstack approach of a decoupled CMS was certainly the right step from a performance and cost perspective. But how could Proximity fully decouple the frontend delivery from Sitecore CMS?

Uniform, a Netlify Partner, made that a seamless experience. By using Uniform's out-of-the-box Sitecore hooks, Proximity could decouple the front from the back end, while still maintaining the existing Sitecore CMS experience for content authors. Because Uniform is built by ex-Sitecore developers, this functionality works from the jump, so there was no need for Proximity to develop the Sitecore to Jamstack integration.

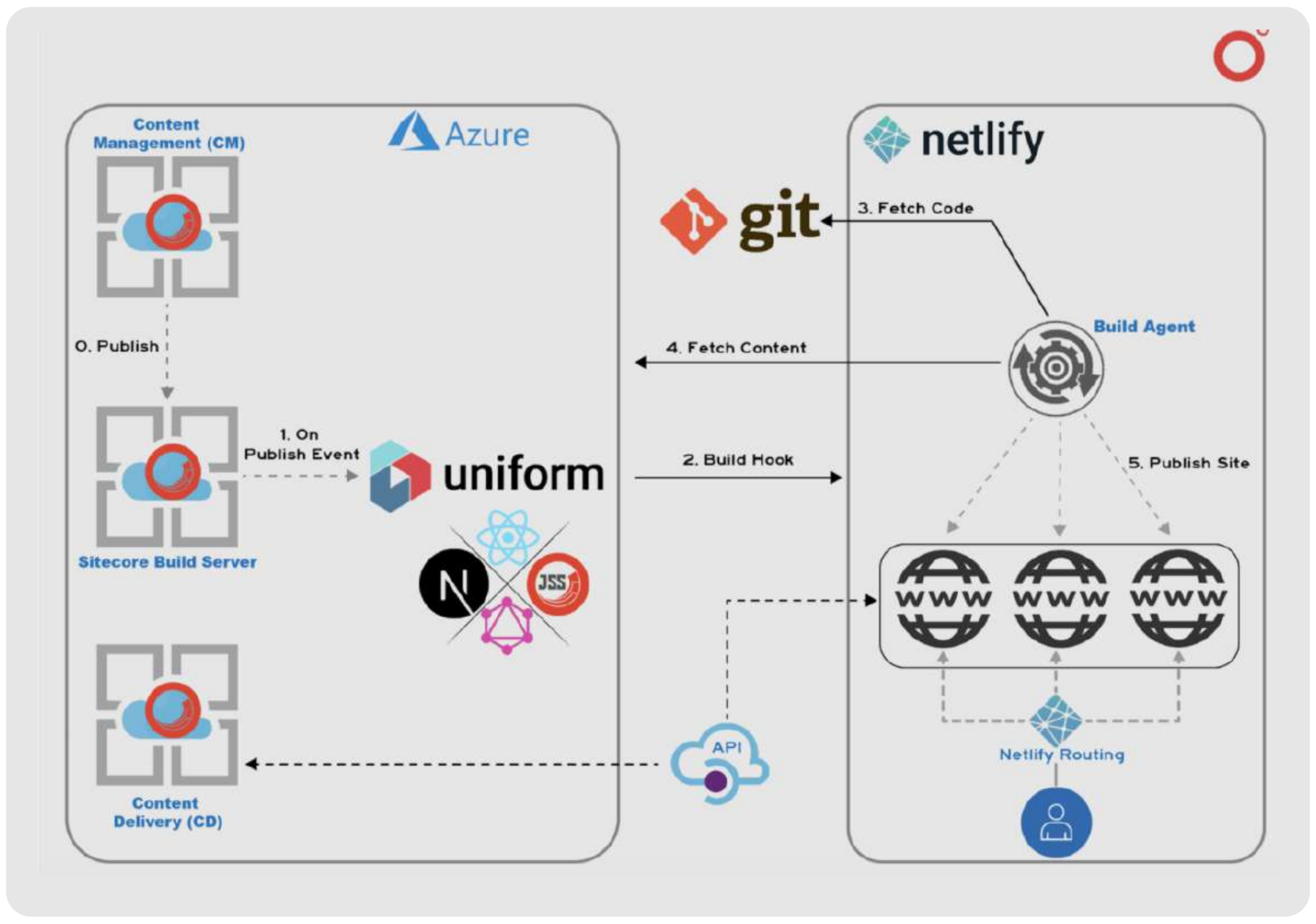
With Uniform, Netlify builds and deploys each Sitecore site to the Netlify Edge network. Sites are cached and pre-rendered so they load instantly on the user's device. For personalization and tracking, Uniform pre renders personalization logic at client side so that personalization is significantly faster.



Overview: How Klépierre Runs Sitecore as a Headless CMS on Netlify

- A multi-tenant Sitecore deployment runs on a dedicated content management server. Website builds and deployments are triggered on a regular drumbeat. Before a build is executed, there is first a check on the content to confirm whether or not there was a content change.

- Upon registering content changes, Netlify runs CI/CD hitting on a dedicated SItecore build server, then the entire site is rendered statically via NextJS and Uniform. The resulting sites are deployed across Netlify Edge.



Klépierre's Sitecore on Jamstack Architecture, using Sitecore 9.2 as CMS with JSS, ReactJS for frontend, NodeJS and Sitecore JSS for server-side rendering, Netlify as CDN, and the Uniform connector to seamlessly hook Sitecore and Netlify

Delivering Dynamic Features with Microservices

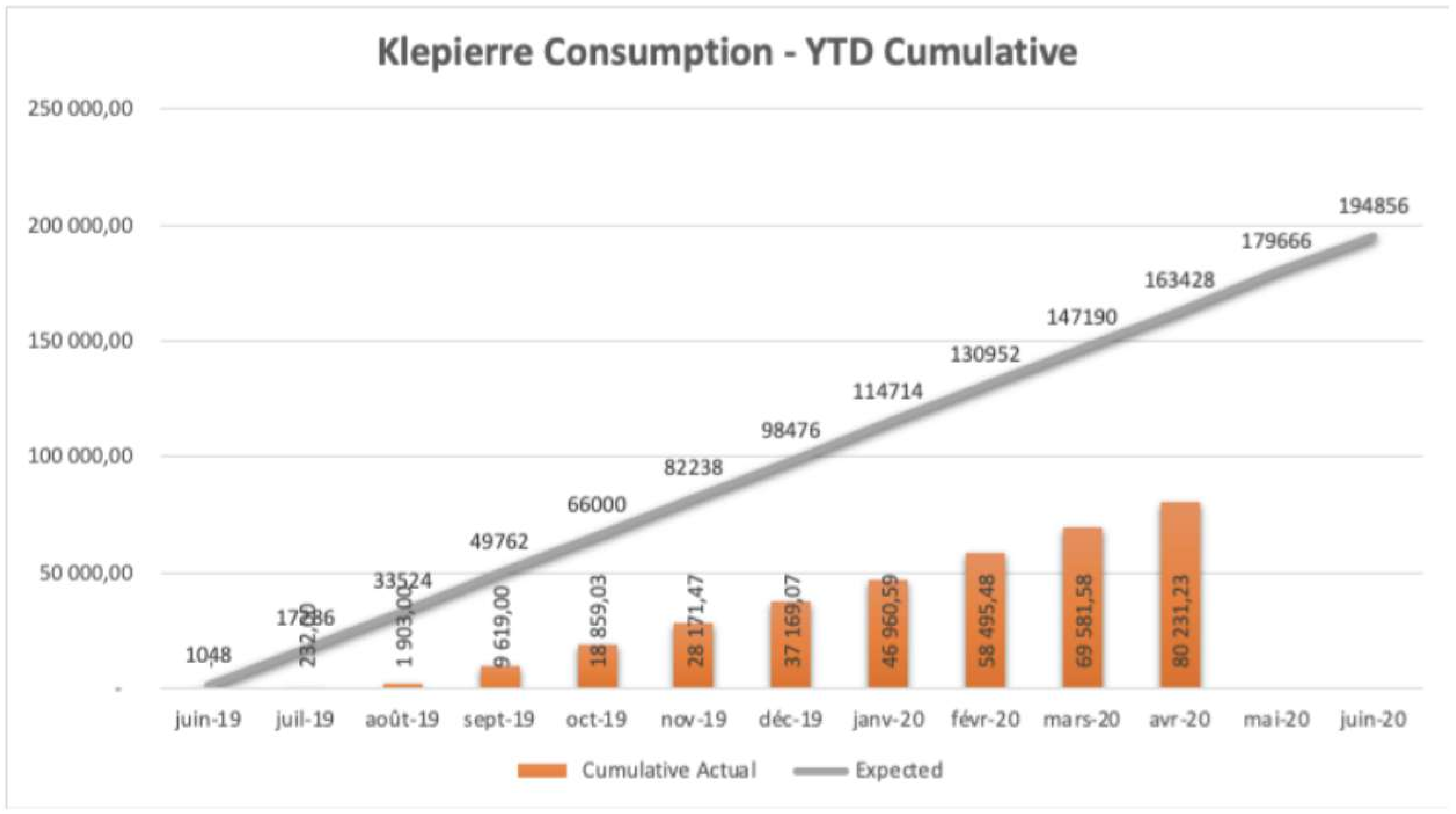
Of course, not all elements of the sites can be static, which is why Proximity used APIs to enrich the frontend localization and personalization experience. Proximity used some third-party tools, and built some APIs themselves.

Since Klépierre manages real estate for malls, store opening and closing times need to be front and center on the sites, in order for the end consumer to know when the stores are open. That type of information shouldn't be cached in HTML: it should be real-time. For this kind of information, Proximity was able to deliver real-time connectivity via microservices, alongside the static cache of the rest of the site.

The Result: Cost-Savings, Uptime, Greater Performance and Page Speed, and Faster Content

4x Cheaper Azure Spend on Jamstack

With the Jamstack approach in hand, Proximity could confidently approach Klépierre with the spend forecast for the Azure and Sitecore proposal, and the option using Netlify and Uniform to mitigate the cost risk. In the end, even while running two servers (one for content management, the other for builds) the total **Azure spend is way cheaper with the Jamstack model.**



Klépierre's Azure Consumption: Expected vs. Actual

Beyond Cost-Savings: Additional Benefits of Headless Sitecore on the Jamstack

The cost savings inherent in a Jamstack approach made the decision to decouple the CMS an easy one. It wasn't until after the decision to go headless was made that Proximity was able to realize the many *additional* benefits of this approach, while experiencing ease with unified developer workflows on Netlify's distributed edge network.

Improved Site Performance

The simplicity of the Jamstack makes sites performant by default. Since more of the application is pushed to the edge, Klépierre customers get a better experience. Six months after the release, Klépierre has had zero outages since the launch, and has achieved an average of 99.999% availability across the 98 sites; an unprecedented for such an image and feature-rich commerce site.

> **When you have this type of scale, everything breaks at some point… the fact there there were no issues with Netlify is big.**
>
> **— Jonathan Bobo,** CTO, Proximity

Page speed has also performed well especially for these content-rich sites, with an average score of 90+ according to Google PageSpeed Insights.

Performance and Personalization

To enhance the experience and personalization for its clients, Klépierre wanted to list shopping malls that a web visitor would be most interested in. The web team was able to personalize specific components on a page while also maintaining the super-fast page speed that was already gained using the Jamstack. Proximity took a modular approach to personalization and did a loose coupling of different building blocks, made possible with Uniform. At build time, the list of malls were generated in a JSON format, and kept on the edge network. They split the traffic and only delivered personalization to a portion of it, personalizing the web experience while keeping up the website performance.

Better Development Efficiency

With a decoupled approach, only the infrastructure team on the Proximity side needed to work directly with Netlify, leaving the frontend developers to continue working in React as if nothing had changed. And by using Uniform, managing the connection between Netlify and Sitecore was easy, which meant the infrastructure team could focus on optimizing build times and minimizing costs. With Netlify's git-generated CI/CD pipeline, the Proximity team benefitted from a single **workflow** on a fully managed global platform without having to manage any servers.

Additionally, other Netlify features, like the redirects API, made building 98 sites and migrating content much easier than anticipated.

Enhanced Security

The Jamstack approach also provided Klépierre with additional security, since a headless model inherently provides fewer attack vectors. A statically generated site only provides connectivity to microservices, not the database, leaving important information secure from attacks.

Empowered, Connected Teams

What's more, the CMS experience has been praised by clients across the Klépierre organization. All in all, content creation time went down from 8-10 minutes per content edit to 1-5 minutes per content edit. All in all, Klépierre has seen a 300% improvement in content creation time.

> **We decided to deliver a Jamstack model on Netlify as a means to better manage costs and mitigate risk on significant licensing from more traditional models. Our client is pleased that their sites load fast, and we were able to cut their costs significantly.**
>
> **— Jonathan Bobo,** CTO, Proximity

In the end, the model that Proximity developed allows the Klépierre team to deliver fast experiences across Europe, on sites that are both user-friendly and author-friendly. Not only does it provide speed, security, and scalability, it provides predictable costs as Klépierre continues their digital transformation. And because they went with a decoupled CMS, if Klépierre ever decides to replatform again and use a different frontend for their content authors, they don't have to re architect everything on the backend.

The Proximity technical team behind Klépierre's digital transformation

By delivering a modern dev experience while maintaining their investment in Sitecore, Klépierre now has the benefits of an enterprise-grade content management system of Sitecore alongside the high performance global delivery, and the elastic scale and cost efficiency of the Jamstack with Netlify.

Engage further

- **Watch a webinar with Proximity and Netlify**
- **Learn about running the fastest Sitecore sites**
- **Have a project in mind? One of our experts would love to talk with you about the use-case and requirements.**

netlify