Guide to Headless CMS & GraphQL

Deep Dive into Headless CMS Architecture and the GraphQL Query Language

BRIGHTSPOT®

Brightspot eBook
October 2020

BRIGHTSPOT®

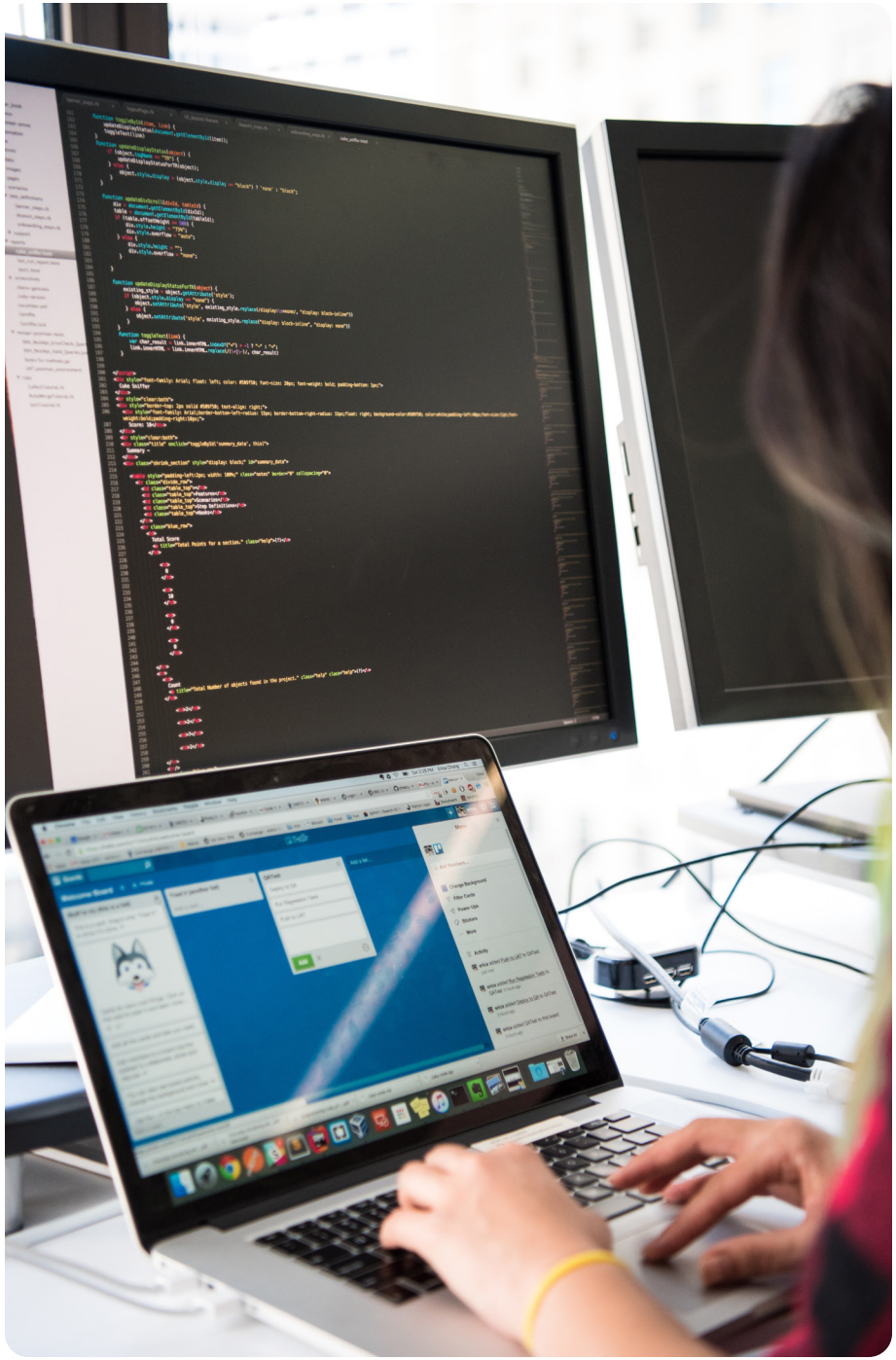
Table of Contents

BRIGHTSPOT®

Introduction

Today's technical buyers live in an ever-changing world—one that's permanently in flux as new technologies and methodologies emerge and disrupt what was previously the new thing in place to solve the next technical challenge. Pressure to always be one step ahead has been compounded by the need to rapidly evolve and digitally transform.

How exactly do technical teams play a role in successfully creating these digital-first environments and strategies that keep their organizations ahead of the game?



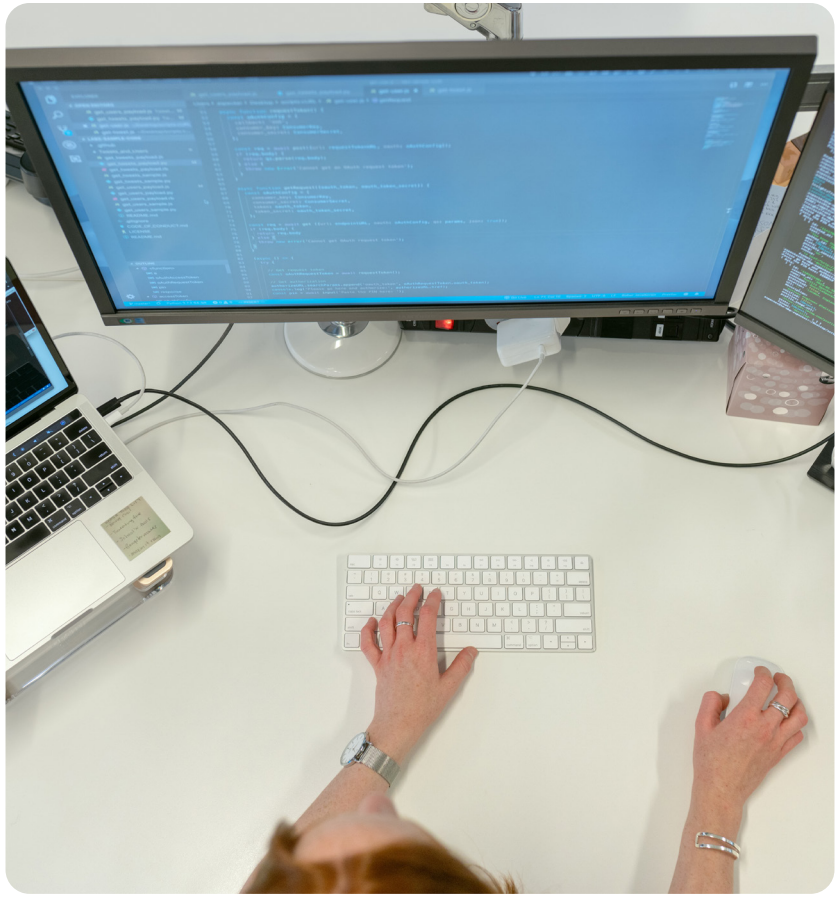
BRIGHTSPOT®

Technical teams are ultimately responsible for aligning technology to support the business mission, strategy and processes. They dictate technical standards—including code, tools and platforms—by taking into account cross-domain considerations, long-term objectives and business process and governance. Most importantly, these teams' northstar is always driving transformation towards a desired future state—building businesses not just for today, but for tomorrow.

Technical teams are, unsurprisingly, the driver of technological innovation, and tasked with enhancing their organization's resiliency. They're constantly thinking about how things could and should work for an optimal experience—and optimal results. To put it lightly, this is a tall order.

To meet today's demands to be digitally nimble and transformative, technical teams need a modern architecture that provides ultimate flexibility in allowing their businesses to transform for the future. This is where understanding headless CMS architecture and GraphQL can make an impact.

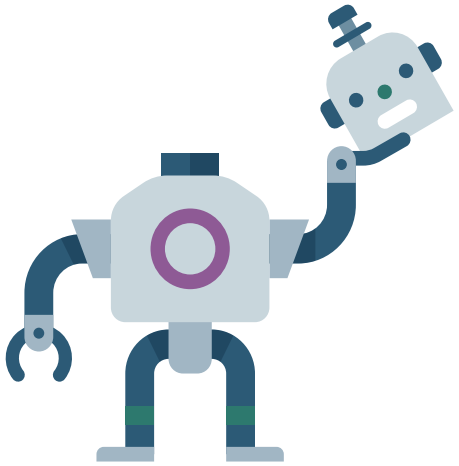
This eBook explains the origin of headless CMS and why this architecture is an increasingly appealing option for technical teams. We'll explain the pros and cons of the various architecture types, considerations to make before going headless, and what role GraphQL plays in supporting headless CMS. Throughout, we'll detail the benefits of headless for technical teams and provide guidance for how to select the best CMS for an organization's unique needs.



72% of strategists say their company's digital efforts are missing revenue expectations.*

*Gartner

The Origin of Headless CMS



Headless CMS applications are on the rise—but what is headless, and where did it come from? To help answer these questions, let's briefly walk through the different content management systems that are available to businesses today.

Traditional:

A traditional—or coupled—CMS tightly links the back end to the front end. Content is created, managed and stored, along with all associated digital assets, on the site's back end. The back end is also where website design and customization applications are stored. This content management back-end and database is bound within the same system that delivers and presents content to the end users' respective devices.

Headless:

Headless solutions are a subset of decoupled architecture. With a headless CMS platform, there is no fixed front end—instead, the solution acts as a content-only data source. This allows developers to use a combination of their favorite tools and frameworks to determine where and how content appears.

Decoupled:

In a decoupled environment, the back end and front end of a website are split—hence "decoupled"—into two unique systems that are managed separately. One system handles content creation and storage, while the other is responsible for taking that input and presenting it to the user through a chosen interface.

Hybrid:

Decoupled and headless architectures have paved the way for the hybrid model. With a hybrid CMS architecture, organizations and publishers have the ability to mix presentation or front-end choices. The hybrid approach offers an environment that allows users to deliver different experiences to a browser window or a device, where both decoupled and headless CMS architectures can be combined.

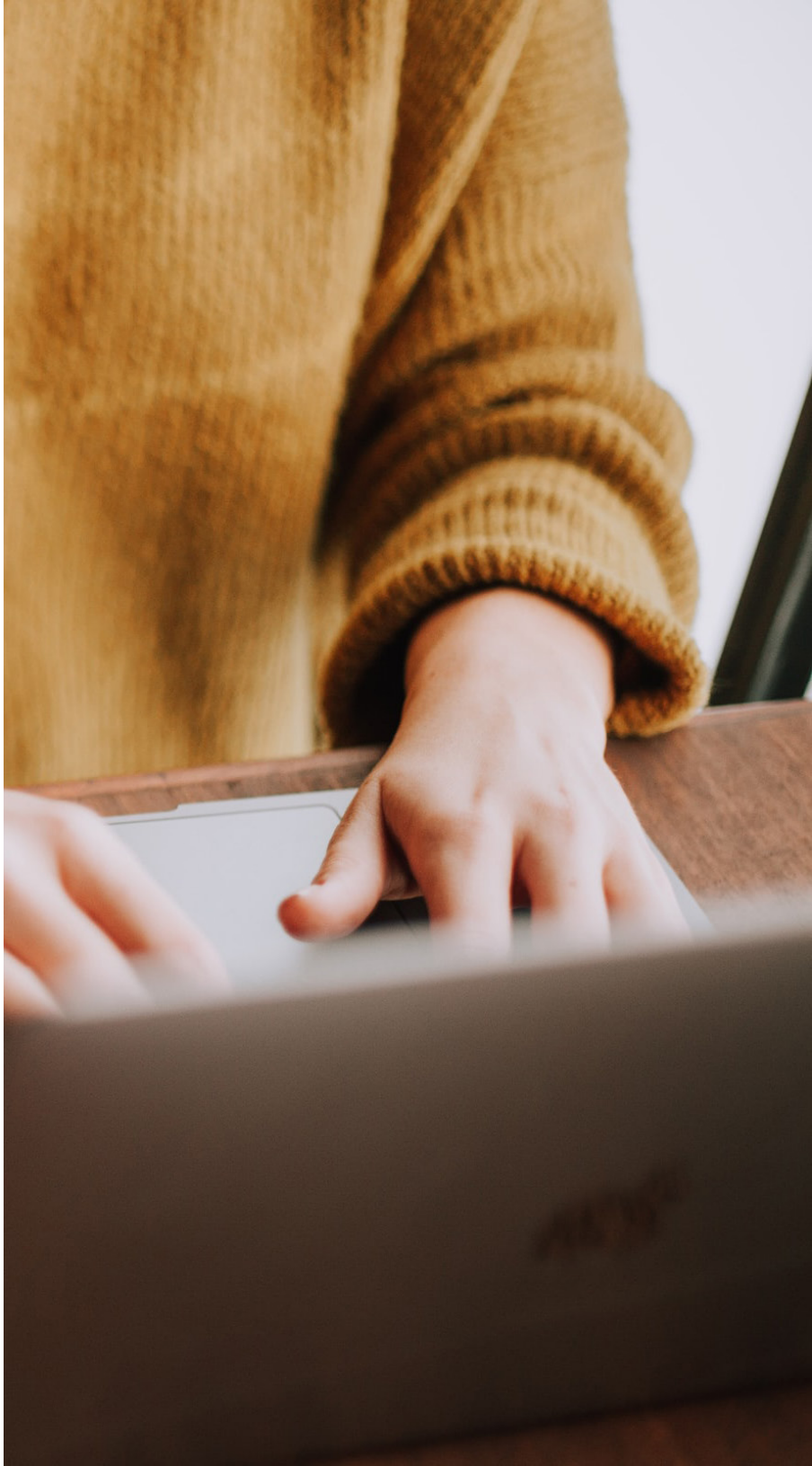
BRIGHTSPOT®

The upward incline of headless CMS adoption started in 2015: the year that the public GraphQL CMS specification was developed, opening up headless architecture to a wider audience. GraphQL has since become an open standard and is an important part of what makes headless such an attractive option for CMS applications today—more on GraphQL later!

From a broader industry standpoint, headless CMSs started to gain traction as businesses needed a better solution to engage people in personalized ways. Not only that, organizations and publishers sought to be able to reach these existing and prospective customers on multiple channels across the entire buyer journey—and they needed the flexibility to do it in the ways they wanted. From traditional web-based applications to emerging technologies like VR or smart-home devices, headless offers an adaptable solution for future-proofing a business' ability to deliver the best possible experience regardless of device

Today, across all of these channels, 71% of customers crave a consistent experience, yet only 29% say they actually get it*. These insights support the idea that organizations and publishers are increasingly in need of a modern CMS solution that allows them to reach people where they are engaging in the ways they expect—not just today, but into the future as new channels and technologies emerge.

*Source



Considerations for Choosing Headless CMS Architecture

Despite the popularity and front-end freedom of a headless CMS, it is not for everyone. **Let us explain.**

The best way to think about a headless approach is as a management system that solely looks after your data, and allows you to access that data. Other architecture types that are not headless involve a system that will also render a webpage—or front end.

This is explained well by looking at the four main components that make up headless CMS:

- A database where content and digital assets are stored (back end);
- A content management back end where content is created (back end);
- An API that connects the content management back end to any device or channel;
- The ability to connect to any publishing front end, allowing organizations to have the front-end technology of their choosing.

How technical teams approach headless architecture and which hurdles they'll run into depend on various factors. What type of web application is being built? What are the dynamics and skill sets on the team? What use cases and requirements need to be implemented? These are all important questions and

considerations each team needs to assess and address before taking the leap in choosing which architecture best suits their business needs.

If headless architecture is a fit, it needs to be implemented correctly (of course) to bring to life its intended benefits. This requires an experienced team to ensure the back end and front end are well planned from the beginning, so they can sync up seamlessly later. Keep in mind, the freedom that teams enjoy when using these systems means that they are responsible for writing, debugging and maintaining everything that their rendering systems require.

Comparing Modern Content Management System Architectures

PROS

Traditional:

- Simple; ideal for text-based content
- Built-in themes and templates
- Customize your front end

Decoupled:

- Fast and flexible content delivery with a specified delivery environment
- Rapid design iterations and simpler deployments
- Fewer dependencies on IT
- The best of both worlds in a CMS (structured back end & flexible front end)
- Future-proof (integrates easily with new technology and innovations)

Headless:

- Fast content delivery
- Provides complete control over how and where your content appears
- Allows developers to use their favorite tools and frameworks
- Future-proof (integrates easily with new technology and innovations)

Hybrid:

- Combines the benefits of decoupled and headless with ability to mix presentation or front-end choices, allowing for the most flexibility
- Flexible “content-first” approach encourages content reuse across different experiences and channels

CONS

Traditional:

- Content types and delivery channels are limited
- Limited programming framework
- More time and money required for customization, maintenance and enhancements

Decoupled:

- More complex than traditional to configure and deploy
- Front end development work required for design

Headless:

- No presentation functionality
- Live preview functionality requires both technical input and front-end coordination
- Reliant on additional technologies for its “head”

Hybrid:

- Success depends on vendor implementation and execution of feature set, capital, customer base, etc.

BRIGHTSPOT®

A headless-only approach tends to be the best option for organizations with robust development teams that know their way around additional technologies required to establish the front end.

With the right team in place—and with the architecture implemented correctly—organizations will quickly start to reap the benefits of a headless CMS.

What kind of benefits? With the decoupling of back-end and front-end needs, implementations can be quicker and development teams can accommodate changing business requirements more easily. The front end can change completely without impacting what's happening in the back end, making it simpler and faster to integrate new designs.

Businesses with multi-national sites or a network of multisites, for example, can benefit from the ability to centralize content management within a headless CMS, which is then published via APIs to back-end-agnostic sites, applications or distribution channels.

Developers can also tap into their favorite tools and frameworks to determine where and how content appears, providing freedom and flexibility to pave their own way forward.

3 Benefits of Headless CMS for Technical Teams

- ***Agility:*** Separation of the presentation layer from the platform lets teams move faster, while separation of content and presentation helps authors and developers work independently, accelerating time to market.

- ***Flexibility:*** Ability to mix and match front-end content offerings, meaning the best user experience can be delivered across every device, channel and touchpoint.

- ***Resiliency:*** Supports organizations in future-proofing their businesses by making it easy to continually evolve alongside technology, no matter what new device or platform emerges.

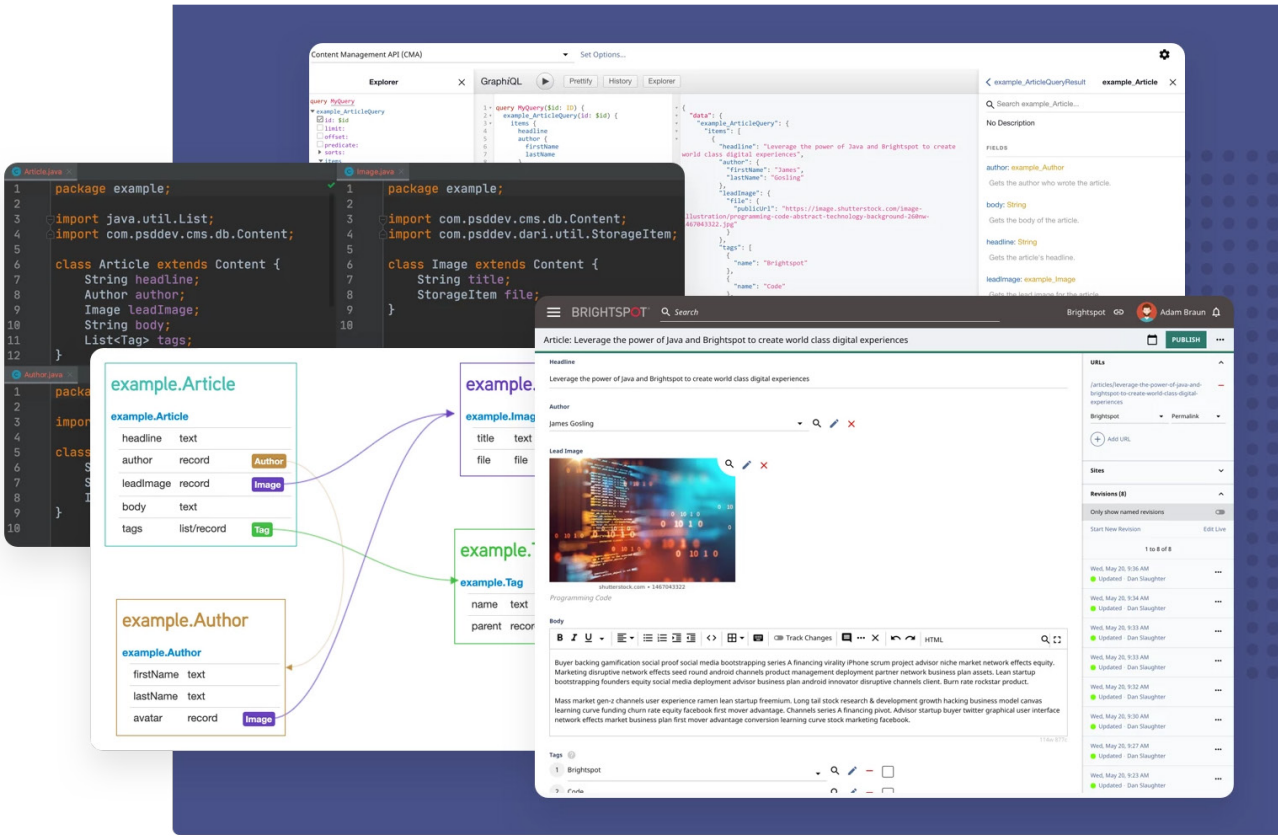
"Headless CMSs are ready to support technologies that will become popular in the future. Some companies are already pushing the limits of content delivery by incorporating more IoT devices, augmented reality, virtual reality and more. A headless CMS built upon powerful APIs will be more easily integrated with the newest technologies that come out, and companies will be poised for quickly taking advantage of new audience segments.'

- Kaya Ismail, CMSWire

BRIGHTSPOT®

Understanding GraphQL

An important part of understanding how headless CMS integrations work to provide ultimate flexibility for businesses is GraphQL. Known as the querying language that enables flexible connection with APIs to support headless CMS integrations, GraphQL enables content management and delivery to external systems, including third-party syndication.



BRIGHTSPOT®

What is GraphQL?

GraphQL is a query language for APIs and a runtime for fulfilling those queries with existing data, providing an alternative to the traditional RESTful style of web services. Whereas RESTful style tends to have looser guidelines for how an API should be structured, GraphQL strictly structures the communication between the client and the server, irrespective of the specific data models used. GraphQL's self-describing type system enables automation in the case of changes to the content data model.

→ What is GraphQL and why it matters for headless CMS

BRIGHTSPOT®

GraphQL Considerations

As mentioned, taking a fully headless approach comes with its own set of potential drawbacks. For each, GraphQL through Brightspot's Content Business Platform offers a solution, but ultimately it will be up to developers to determine if those solutions will make headless CMS the best option to solve their unique business challenges. Let's look at how the Brightspot Content Business Platform is configured to address specific considerations with respect to GraphQL.

Consideration	Resolution
GraphQL queries aren't small, requiring POST requests to fetch the data.	"Automatic Persisted Queries" replace large query text with a generated ID, to which an app can then map and query each time it's requested.
Since rendering of the application occurs outside the CMS, live preview is difficult to implement; this makes it challenging, especially for editors, to see how their publishing changes in one system affect the presentation of the site they're working on.	Brightspot's preview system is extensible, supporting custom preview implementations for print templates from InDesign as well as native apps. Similarly, you can specify the URL to your app, and Brightspot will iframe it into the preview pane and pass along a special preview ID. If the client uses that same preview ID when making its call to the GraphQL CMS API, then Brightspot will return data specific to the content the editor is modifying.
The front-end application that uses specific image sizes is separate from the CMS, meaning it's unable to access the data it needs to run rich image editing and cropping capabilities.	Users can register a configuration file of their standard image sizes, which will display their defined crops in the image editor and return image URLs from the GraphQL API.
Code running in the client's browser is completely open and visible to the public, and anyone can view the data transferred from APIs. GraphQL developers must be explicit about which fields they are fetching in order to mitigate risk, but a malicious user could still inspect the schema and make their own queries to fetch additional data not used by the app.	To prevent data, potentially sensitive, from being inspected, your app needs to run server side instead of client side. Or, you can leverage Brightspot's View System (which sits in between the raw Database Model and the ViewModel that's returned by the GraphQL API) to sanitize the data on the back end, thus allowing you to continue running your app client side if you desire.

BRIGHTSPOT®

GraphQL Advantages

The benefits of a GraphQL CMS come down to three things: simplicity, automation and flexibility.

At the same time, GraphQL CMS APIs simplify data gathering by enabling users to collect all data an app requires in a single request; REST APIs, on the other hand, require loading from multiple URLs.

In terms of automation, GraphQL has a self-describing type system that automatically reflects new fields added to a content type. This enables clients to more easily discover which data types and fields are accessible from the API, and which are being utilized. The self-describing type system also supports features like auto-complete.

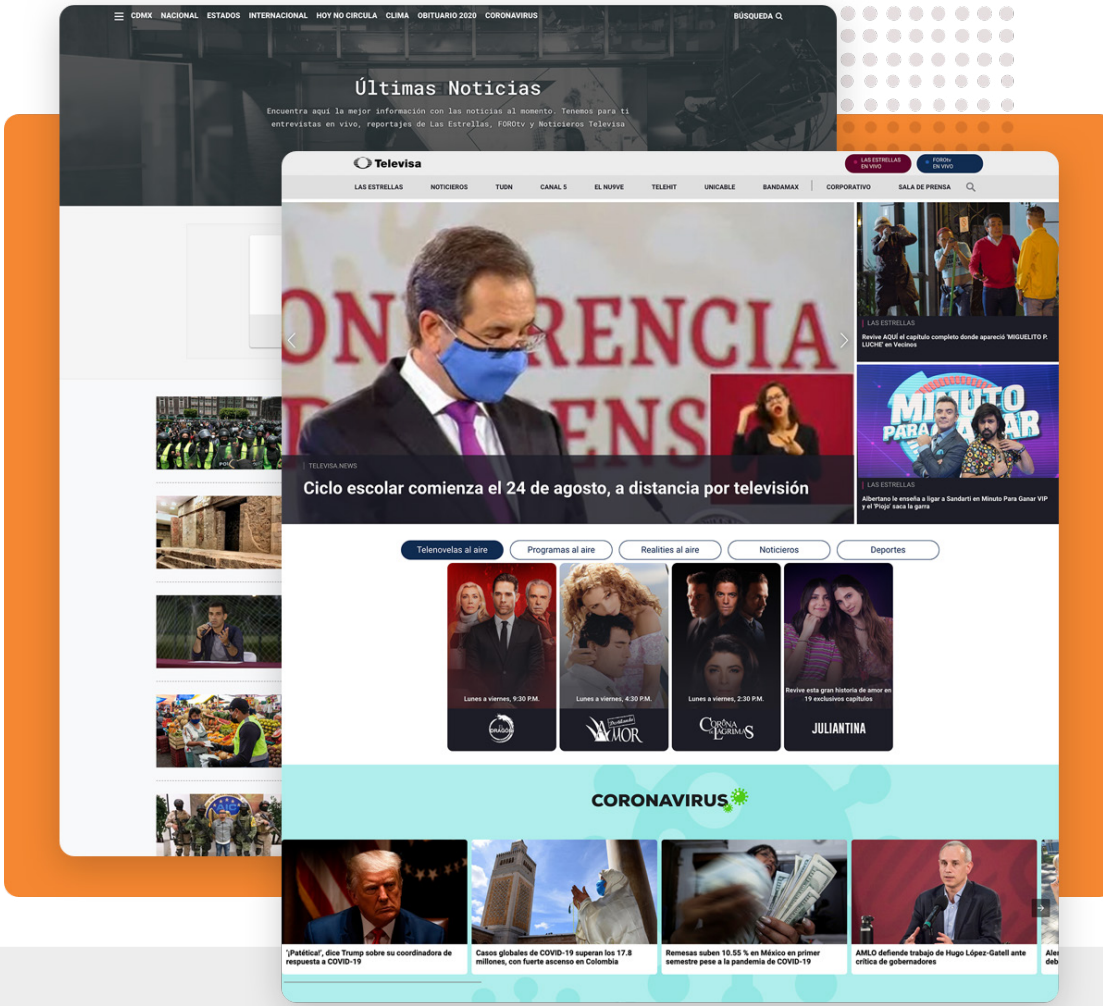
Finally, GraphQL APIs operate as one continuous evolving system, meaning adding new fields and types doesn't affect existing queries, and older field versions can be hidden. Through one single version, apps get continuous access to new features which makes for a cleaner and better-maintained server code. At the same time, GraphQL is flexible in that it is not limited by a specific storage engine, meaning users can create a uniform API across their entire application. Because they're available in many languages, GraphQL engines can also be used to write GraphQL APIs that leverage existing data and code.

We've outlined the key benefits of headless and GraphQL—**but what sets Brightspot apart?**

1. Developer-friendly for use with any front-end framework, including Brightspot's proprietary front end
2. Extensible architecture that offers ultimate flexibility to choose decoupled, headless or hybrid-headless CMS as your approach.
3. Integration-ready and migration-friendly architecture.
4. Custom workflows that act as an extension of an existing business logic.
5. Future-proofing for your content business platform.

Televisa

Televisa Customer Success Spotlight



Challenge:

After leveraging several CMS platforms to manage all of its digital properties, Televisa decided to replatform its sites onto one multisite, headless CMS. The ideal solution would streamline all back-end functions, give the developers full front-end control, and make it easy to create and launch future sites independently.

Solution:

Over the course of five months, Televisa's front-end developers and editorial teams worked side by side with the Brightspot team to learn Brightspot. After migrating all of its content and creating a model with the first headless site launch, Las Estrellas, Televisa replatformed a total of nine sites in just five months.

Results:

- 50% reduction in launch times, with nine sites launched in five months
- Nine CMS platforms consolidated to one
- 35 front-end styles powered by four simple fields

BRIGHTSPOT®

Getting Started: The Technical Buyer's Checklist

Every business is different—and there is no one-size-fits-all solution to choosing the CMS that's right for yours. To assess which CMS solution is best for your organization, technical buyers should consider the following factors.

- **Review your options**—Make sure you've done your research and understand the differences between traditional, decoupled, headless and hybrid CMS architectures.

- **Consider your team**—Understanding a headless approach requires deeper technical know-how, so determine if your team has the skill sets to successfully fulfill front-end requirements and maintain this approach.

- **Identify your content business goals**—Where does your organization hope to be in five or ten years? Will you be looking to scale to different regions and touchpoints? If so, having the front-end flexibility to easily deploy experiences on new channels is a plus.

- **Weigh business outcomes**—Select the architecture that ensures teams (technical, publishing, marketing, etc.) and the organization at large will experience the intended benefits of the chosen CMS.

At Brightspot, we believe in front-end freedom of choice—to be able to choose the architecture that best suits each individual organization's unique needs. That's why we've designed our Brightspot Content Business Platform to operate as a traditional, decoupled, headless or hybrid CMS solution.

We're here to answer your questions and guide you on your journey, whichever direction you may take.

BRIGHTSPOT®

Next Steps

Did you find our Guide to Understanding Headless CMS & GraphQL helpful? You are now familiar with the technical considerations to help you select the best CMS for your organization's unique needs. Move your business forward with a content business platform built with an extensible architecture that's built to solve your unique business challenges without having to compromise.

Is the Brightspot Content Business Platform the right solution for your business? **Request a demo** with a product manager to learn more about:

- **How Brightspot is designed to grow with your business and extend with your business logic**
- **Deploying a headless, decoupled or hybrid headless CMS—all from the same environment**
- **Our world-class Delivery Team and how you can start your POC**

Request a Demo

At Brightspot® we believe technology should enable content-focused teams to work smarter, faster and more seamlessly to move businesses forward. With decades of experience in publishing and media, we help companies transform their business content and digital experiences by creating enterprise applications at scale with astonishing speed.

@BrightspotTeam **www.Brightspot.com** **Demo@Brightspot.com**