Acquia

EXPERIENCE DIGITAL FREEDOM

THIRDANDGROVE

THE POWER OF DECOUPLED DRUPAL

More Than Just Websites

Introduction

The World Wide Web is changing. What used to be a collection of simple, static and often bulky webpages is now a sophisticated and diverse network of highly interconnected platforms communicating in real time. The Internet of Things, digital kiosks, smartwatches, smart TVs and even virtual reality are no longer relegated solely to the realm of tech-savvy geeks. Whether you're a preteen, a college student or an empty nester, you're more and more likely to be connected to more content than ever before—at home, in the classroom, at work and even at play.

In other words, websites are now just the starting point for a consumer's digital experience. Companies who want to create or enhance such an experience are faced with an ever-growing number of channels to provide services to, acquire data from, and stay in touch with their customers. The very notion of "content" as we know it is changing—now content is often off-the-screen. Movies and TV shows are still content, but so is virtual reality. So is biometric data. So is conversational content, such as a social media comment or a WhatsApp message. Content is changing and so is the way we deliver and manage it.

Usually, with more flexibility comes more complexity. But as the web becomes more powerful, agile and diverse, it is also becoming smarter. More ambitious digital experiences can be built leveraging modern technologies and front-end frameworks thanks to the power of tools like Decoupled Drupal. Decoupled Drupal is far from new, but it is poised to capitalize on the changing web in a big way.

This e-book will situate Decoupled Drupal within the larger Drupal conversation, offer thoughtful analysis of when Decoupled Drupal is the right option for an organization, and showcase some of the many ways that it can be used. **Readers will come away with a solid understanding of the versatility and power of Decoupled Drupal.**

Characteristics of a Modern CMS

A content management system (CMS) is a software application used to create and manage digital content. Drupal is a free, open-source content management system used by millions of people and organizations to do just that. Drupal is used to make many of the websites and applications you use every day, including some of the most influential around the world. The Drupal of today offers great standard features, like easy content authoring, reliable performance and excellent security.

With platforms like Drupal, the World Wide Web has come a long way since its early days, when the internet was largely a collection of static webpages composed of text and images. A modern CMS typically offers the following capabilities:

1.

Easy Content Creation: Authors should be able to create a new, ready-to-publish webpage with the click of a button—for both unique pages and those based on templates. Authors should also be able to easily revise content after publication, and if need be, sift through revisions of that content by various authors.

2.

Intuitive Indexing and Search Features: Users should be able to easily index, search and retrieve all data on a CMS-powered website. Furthermore, they should be able to sort and filter that search to browse content by author, publication date or type, or keywords.

3.

Central Administration: It should be pain-free to authors, editors, designers and administrators to access a CMS in just a few steps. A modern CMS offers a one-stop-shop for all members of a team to collaborate on the creation and management of content.

4.

Flexibility: Modern content management systems provide much more than the ability to write and publish content. A wealth of native and third-party plug-ins and APIs exist to extend the functionality of a CMS—countless of them ready to use immediately after installation.

The Advent of Drupal

Drupal was one of many open-source content management systems that arose in the early 2000s. Drupal, decoupled or not, is a strong choice for building future-proof digital experiences. When Drupal 8 was being created, devices like digital personal assistants and smartwatches didn't even exist. But, because Drupal endeavors to be API first, there is flexibility at the architectural level to tie into any current or future system.

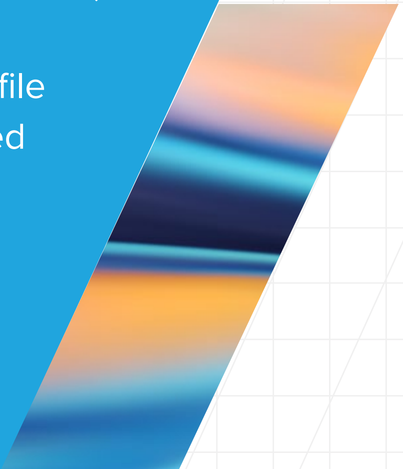
Today, approximately 2% percent of all websites are built with the Drupal framework, including the websites for major names such as the BBC, NBC News, Harvard University, the Government of Australia, the Grammy Awards and NBC Olympics. The flexibility, security and accountability of Drupal's source code are widely considered to be unmatched by most proprietary software options. Furthermore, the portability of Drupal code allows a web developer to set up a rather complex environment in a relatively short period of time.

Over time, as the needs of consumers have diversified and complexified, Drupal has stayed flexible to accommodate new technologies. Drupal's capabilities have been integrated with Backbone.js, CKEditor, Guzzle, Symfony2 and Twig. Drupal's capabilities support collaborative authoring, podcasts, image galleries, peer-to-peer networking, file uploads/downloads and more. Drupal developers have authored a wealth of tools that anyone may adopt to extend Drupal's functionality relatively quickly, which include more than **2,795 themes** and **43,831 modules**.

2,795
Drupal themes available

43,831
Drupal modules available

1,388,175
Drupal community members



The Drupal of Tomorrow

In light of such significant enhancement, IT decision-makers face an almost endless array of options at their fingertips for creating a Drupal-based digital experience. On top of that, there's the matter of how best to present your content—how should it look and how do you manage aesthetic presentation across multiple screens? Two guiding principles can offer a north star for IT decision-makers trying to navigate such a wild world:

FUTURE-READY

It's more important than ever to design applications today that will be relevant tomorrow. This isn't always easy. Even the most diligent architects won't always be successful. We have no idea what new products, platforms, and needs will come out tomorrow, or what we will need to integrate with. However, thanks to Drupal, there are multiple ways to ensure that you're building your architecture on a solid foundation, and the community is constantly responding to industry changes.

STREAMLINED

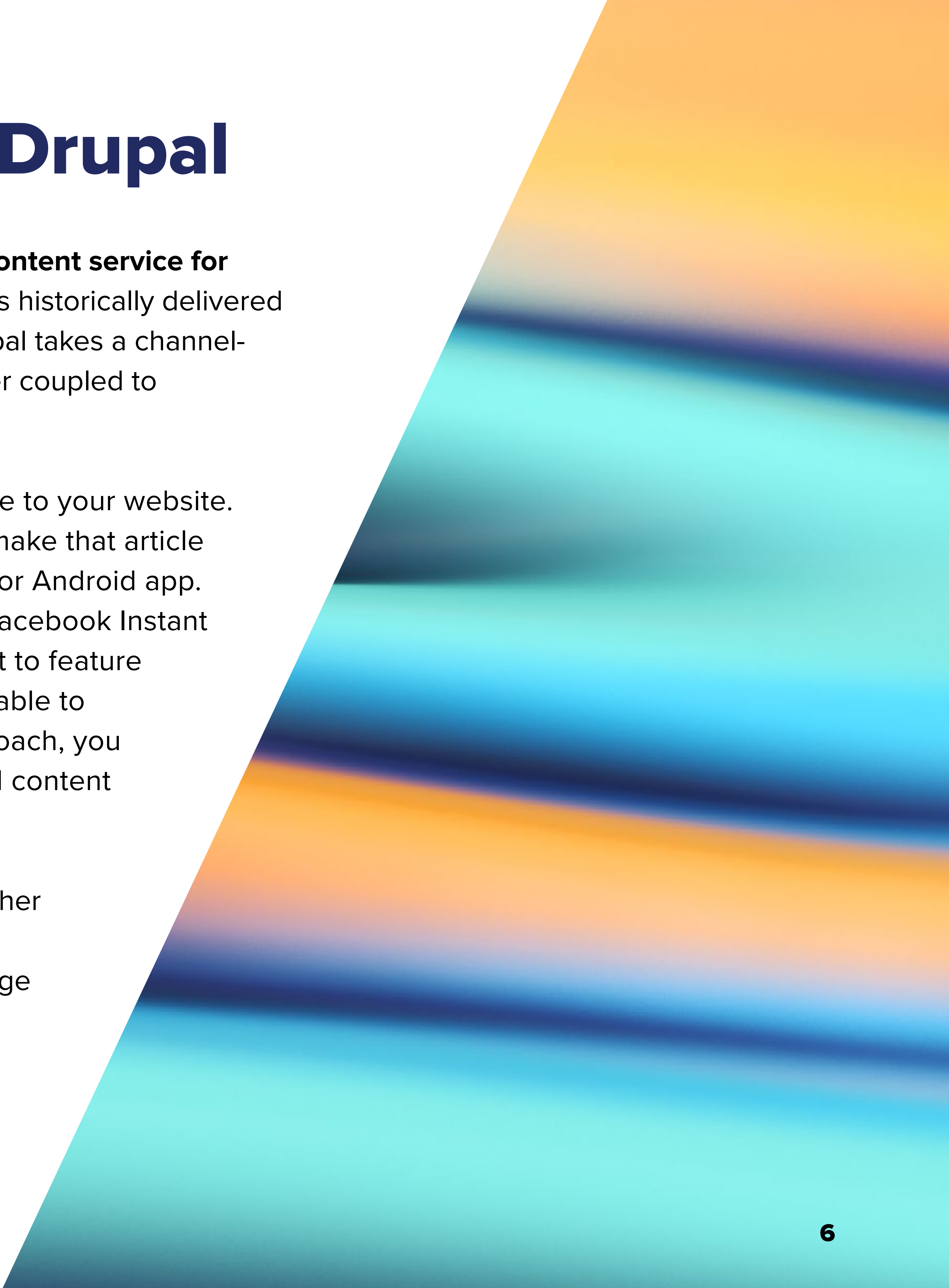
Your digital architecture may be totally all-encompassing for your business needs, but that doesn't mean it'll stay that way for long. You need to put your team into a position that enables them to embrace new channels and/or platforms as they arise. You never want to feel stuck, nor do you want to feel left behind. So it's important to ensure that the decisions you make today allow you to focus on quickly solving problems using best practices and industry standards.

Enter Decoupled Drupal

Decoupled Drupal means using Drupal as a content service for consumption by other applications. Drupal has historically delivered content for websites, however Decoupled Drupal takes a channel-agnostic view. Content, in this case, is no longer coupled to presentation.

For example, say you need to publish an article to your website. That's simple. But today, it's very common to make that article accessible to customers through your iPhone or Android app. You may then want to utilize Google AMP or Facebook Instant Articles for more distribution. Or you may want to feature that article on a digital display or make it available to users of digital kiosks. With a decoupled approach, you can achieve all of these goals with one central content management system.

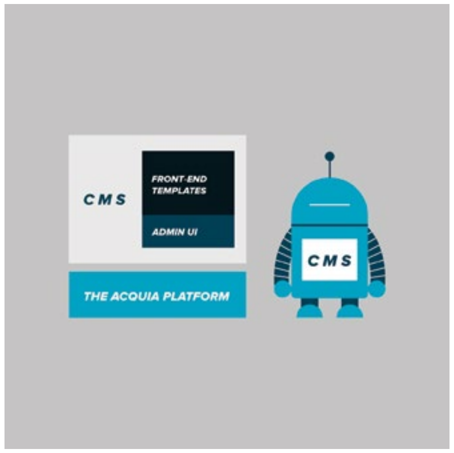
Decoupled Drupal can be utilized to support other server-side applications, applications for both native desktop and mobile platforms, single-page applications built with JavaScript and even IoT applications.



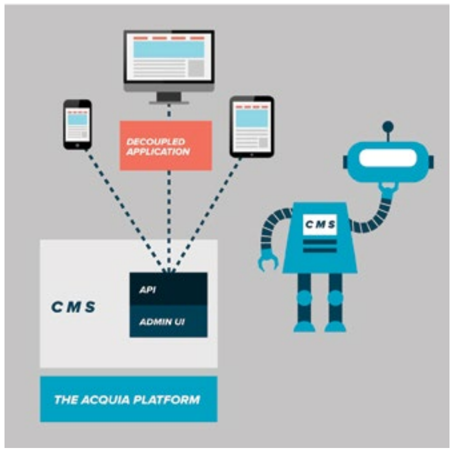
Different Kinds of Decoupling

As mentioned, Decoupled Drupal is the process of employing Drupal as a web service provider that exposes data for consumption by other applications. In “Coupled” Drupal, also known as traditional Drupal, the presentation (front-end) layer is part and parcel of the overarching Drupal framework. Drupal maintains complete control over all systems—it’s what you expect when you open an “out of the box” CMS.

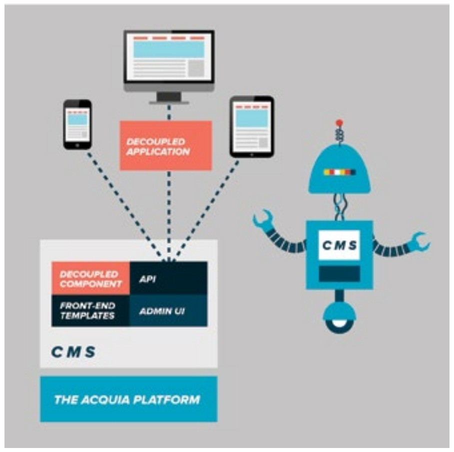
However in a decoupled architecture, a consumer application is written in an arbitrary language, such as JavaScript, Java or even PHP, and interacts with Drupal through a RESTful API present on the server side. And not surprisingly, there are multiple ways to decouple Drupal—specifically, three modes of developing with Drupal: Traditional, Decoupled, and Hybrid.



Traditional
(Fully coupled)



Decoupled
(Headless)



Hybrid
(Progressively decoupled)

Just as Drupal is constantly changing in response to enterprise needs and technological advancements, so is the decoupled movement. Whatever your needs, there is likely a decoupled solution for your project—or there soon will be.

DECOUPLED DRUPAL

A “fully” Decoupled Drupal installation serves as a RESTful data repository. There is a complete separation between the client side and the server side, resulting in differing experiences for Drupal users and developers. Fully Decoupled Drupal implementations typically do not expose a Drupal front end beyond the applications receiving data from Drupal.

Often, a Fully Decoupled Drupal implementation uses Drupal solely as a content repository. Many fully decoupled platforms consist of a single decoupled front end, such as one written in JavaScript, connected to a Drupal site. Some even choose to completely forego a Drupal-like interface for end users, opting instead for a more application-like experience. This approach gives the developer the most control at the expense of the content creator.

HYBRID (PROGRESSIVELY DECOUPLED) DRUPAL

In Hybrid mode, decoupled components can be embedded on top of the existing Drupal front end rather than replacing it. Hybrid Drupal is ideal when greater power is required on the front end than Drupal offers by default. JavaScript is a superb tool for a highly interactive experience, and Hybrid implementations enable builders to enjoy the novel benefits offered by Drupal front-end tools while reaping the rewards of decoupled components.

Think of these decoupled components as “mini applications.” They can be as small as a single field or a big as the entire content area of a page—needs dictate implementation. With this approach, front end developers can create these decoupled components as they normally would, and they will be available inside the CMS for use by site builders. Front end developers don’t need to learn PHP or Drupal to make their components available, and content creators retain power over the experience.

The Strengths of Decoupled Drupal

1 Create Once, Publish Anywhere

Decoupled Drupal means that the philosophy of “Write Once, Publish Anywhere” is more attainable than ever before. When decoupled, Drupal becomes the hub for an unlimited number of spokes representing unique customer experiences and applications. The versatility of Drupal can save you time when it comes to managing multiple channels.

2 Separation of Concerns

In a decoupled implementation, presentation and aesthetics become the domain of consumer applications. Drupal doesn’t need to showcase data to your end user. Instead, Drupal can do what it does best: back-end content management. Likewise, any third-party presentation layer can focus on its strengths, leading to a better user experience. For example, an interactive application requiring frequent re-renderings of content may be more effective in a JavaScript framework than in Drupal.

3 Pipelined Development

Separating concerns can often lead to revolutionary changes in workflow: for example: focused tooling, granular deployments when troubleshooting, and “pipelined development” (in which development for different technologies requiring different specialties can proceed at different velocities as needed). Front-end developers with specific knowledge need to know less about Drupal to get work done.

4 Future-Ready

Decoupled Drupal simplifies upgrade paths and maintenance. It collects and organizes content once, avoiding the task of editing the back end to support each new channel. Instead, each channel comes with its own “front end”—but these front ends don’t add to the complexity of the CMS. When the next new “instant” publishing format arrives, you’ll be in a position to adapt and adopt efficiently.

When Should You Decouple?

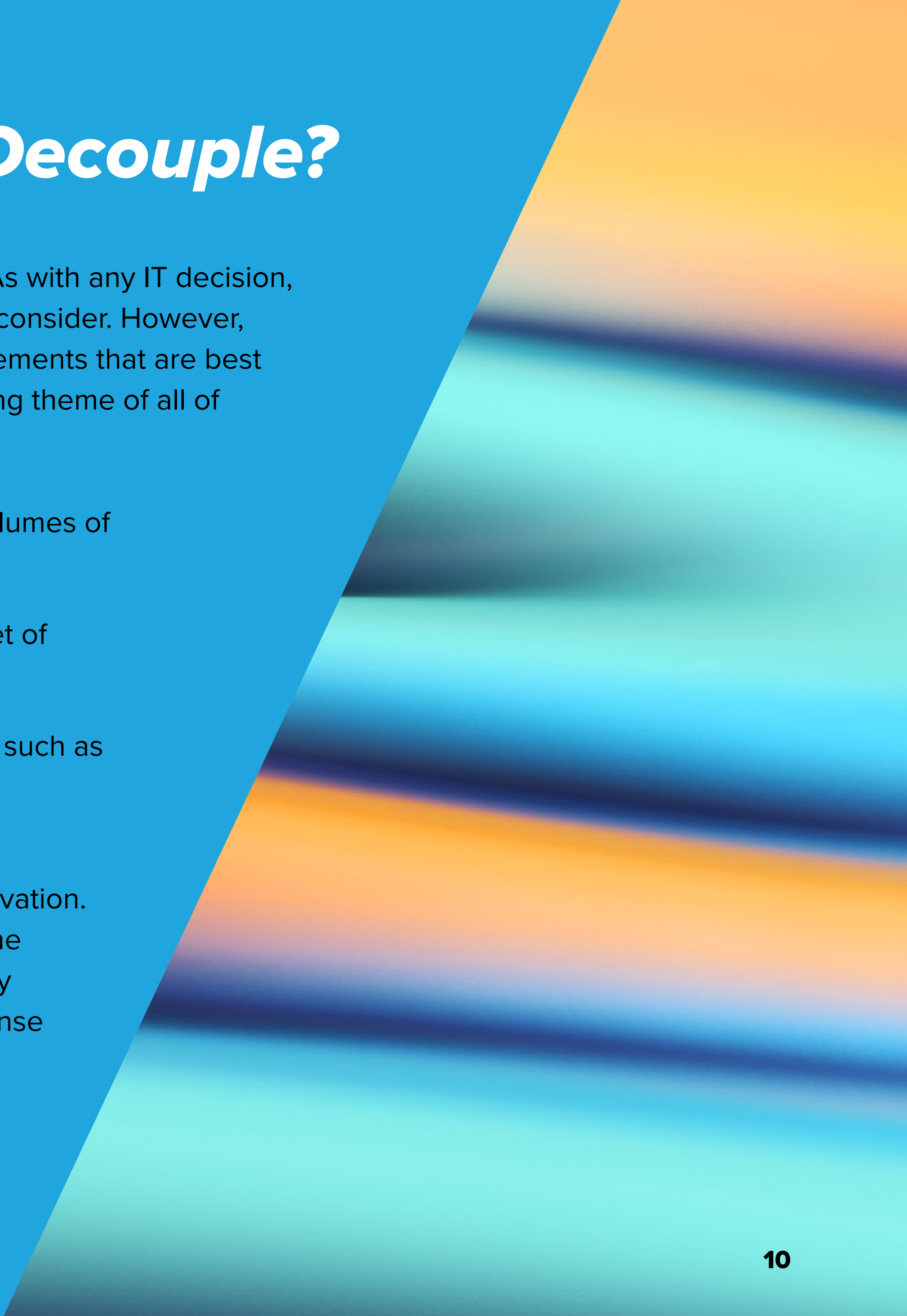
Not every project requires a decoupled approach. As with any IT decision, there are many elements —both pros and cons—to consider. However, there is a generally agreed upon checklist of requirements that are best facilitated with such an implementation. The recurring theme of all of them is a simple idea: flexibility.

— The flexibility to structure and/or display large volumes of content

— The flexibility to distribute content to a diverse set of platforms in addition to your basic website

— The flexibility to extract data from multiple feeds, such as social media or another CMS

Everyone likes flexibility, and it can also lead to innovation. But that doesn’t mean that the ends always justify the means. Any foray into a decoupled architecture (fully decoupled or hybrid decoupled) should be in response to business needs—both yours and that of your customer base.



How Drupal Empowers Headless Architecture

It’s no wonder that decoupling has become such a natural fit for so many Drupal-based digital experiences. After all, Drupal is renowned for two characteristics that tie directly into decoupling – agility and scalability. Decoupling is in many ways the natural next step for a content management system loved by both editors and developers alike for its ability to power organizations large and small, public and private, dynamic or static.

Furthermore, Drupal is more than API-friendly. It’s API-first. Site builders can utilize modules like Serialization and RESTAPI, among others, to easily create their own APIs for communicating with third-party web services. In other words, the ability to support external frameworks is built into Drupal’s DNA. And Drupal is working overtime to make decoupled architecture more accessible than ever.

For example, Reservoir is a Drupal-based content repository that boasts all the necessary web service APIs needed to build decoupled front-end applications. The possibilities include a React application, an Ember front end, a native application, an augmented reality application, a Java application or a .NET application (among others).

Reservoir maintains the workflows of Drupal, as well as keeps its editorial UI intact. Reservoir also returns control to web developers by emphasizing Drupal’s web service APIs. But with flexible content modeling and custom fields added to the equation, they also provide even more control to editors, so they may easily curate, combine and remix content for different channels.

What To Consider Before Decoupling

A comprehensive list of all of the things an IT decision-maker should take into account when considering a decoupling would likely exceed the length of this e-book. The most important factors are likely unique to your project. However, there are four overarching, high-level concerns to consider.

Your Resources: Decoupling is likely not the simplest solution at your disposal. Successful decoupling requires expertise. Expertise requires an investment. It will require time and money to build a front end and back end separately—you'll need two teams. And it likely will cost significantly more than designing a coupled asset. However, you can likely recover that time later when you want to redesign an asset or add a new channel.

Data: Decoupling your asset can result in unwieldy data management. For example, if you utilize third-party functionality on your website and decide to centralize all of your channels with a decoupled architecture, you may need to make complicated decisions about how to route the data from that third-party service when multiple channels interact with it. This requires a solid understanding of your data needs.

Security: Decoupled architecture requires careful scrutiny of your security measures. Drupal offers out-of-the-box form validation and text sanitization—but only in a traditional architecture. If you opt for a "vanilla" JavaScript without the aid of a framework, the security of user-generated input becomes a potentially massive risk. Instead of allowing Drupal to do the heavy lifting, your approach will require ample research to evaluate if you have taken satisfactory steps to ensure the security of users, the consumer applications and ultimately the entire architecture.

Hosting Architecture: Your current hosting architecture may be entirely inadequate for decoupling. It may not offer tools like caching that are necessary, or you may be beholden to contracts or restrictions on data migration or security that would make decoupling long and complicated.

Decoupling Tradeoffs

With flexibility comes complexity and responsibility. With all of the buzz surrounding Decoupled applications, it can be easy to sidestep the potential tradeoffs to decoupling, but they should be evaluated intensively. Choosing to use a CMS only for its web service capabilities and as a content repository can endanger your entire architecture if you need any of the critical functions that rely on the presence of the default front end. Below are some risks and disadvantages to consider.

Additional Point of Failure

Typically, traditional Drupal implementations are hosted on LAMP stacks. On the other hand, JavaScript consumer applications in a decoupled architecture obligate the use of Node.js stacks like MERN or MEAN. Other solutions entirely may be necessary for native mobile or IoT applications, which have different demands. Introducing an additional hosting stack into your organization's infrastructure may not only be difficult for those of more modest means; it also introduces an additional point of failure in your architecture.

Contextualized Editing and Administration

Some of a CMS's most compelling functions include in-place editing and configuration menus. During content preview in a coupled architecture, these interfaces permit editors to adjust content while simultaneously viewing its live result, or to access administration pages from the comfort of the visual preview. These contextualized tools, in a fully decoupled architecture, are no longer available.

Layout and Display Management

A CMS like Drupal offers a spectrum of options for variable content displays. Because they require significant control over Drupal's markup, these modules need to be tightly coupled to Drupal's presentation layer. Removing modules like Panels and Display Suite from the editorial equation means that layout management becomes a developer concern, not an editor's. This results in considerable challenges for teams without access to developers to assist in implementing layout changes.

Previewable Content Workflows

A CMS's functionality for content previews is typically lost when a front-end is no longer under the domain of that CMS. This can translate into considerable challenges if an editorial team desires a previewable content workflow, to which they may be accustomed after years of working within a traditional CMS.

System Notifications

A key feature of Drupal is its robust notification system, which displays information about any issues arising during a Drupal system process, especially severe system errors that demand immediate attention. A REST resource is available within Drupal to fetch watchdog logs, but these provide only a limited amount of the possible issues that administrators should scrutinize. Moreover, Drupal system messages frequently highlighted at the top of rendered pages are inaccessible.

Monolithic Performance Benefits

Cache tags allow developers to define dependencies on data managed by Drupal and permit cache invalidation of items that rely heavily on granular content contained within them. Such capabilities give Drupal the means to achieve significant performance improvements during the page load. This type of progressive loading dependent on cacheability metadata is not available to developers of fully decoupled implementations.

Accessibility and User Experience

Drupal's efforts on accessibility and user experience have included utmost consideration for markup and how it is presented to people living with disabilities and users of assistive technologies. For example, Drupal's use of ARIA roles and other techniques ensure that all Drupal content is available for users of screen readers. In the fully decoupled and progressively decoupled setting, Drupal no longer provides readymade front-end code or a roster of core interface components and interactions.

"Do the benefits outweigh the tradeoffs? Despite the drawbacks, for a variety of use cases going headless offers a compelling set of benefits that improve the customer experience layer. However, it is important to remember that no architectural decision in technology is free; everything has a cost. The most important consideration to ensure success is to dispassionately focus on what will drive real value to your business and what really helps you achieve your business objectives, and work backward from there."

Justin Emond, *Headless Rises: A Field Guide to the CMS Apocalypse*

The Power of Drupal

Not surprisingly, the versatility of Drupal has led to an explosion of uses across various platforms and channels. Thanks to decoupled architectures, Drupal is no longer simply a tool for powering websites. Indeed, the Drupal of today is now used to power everything from mobile applications to digital displays to smart TVs and smartwatches. Below are some notable stories of innovation that all share one ingredient: Drupal used in a decoupled mode.

Princess Cruises: Taking Mobile Apps and Chat Services to the Next Level

Princess Cruises has a rousing success story thanks to Drupal's decoupled architecture, which allowed the cruise line to take its Princess@Sea mobile application to a whole new level. What once was a simple application became an onboard digital ecosystem, providing passengers with personalized, real-time experiences, and Princess Cruises with an avenue to centrally manage all of their data.

Content is managed from a central Princess@Sea Drupal site and is then migrated out to 18 unique Drupal sites onboard the ships at sea. Guests can send messages to each other, plan their day's events, activities, and shore excursions, review the ship's itinerary and port guides, browse restaurant menus, and even access their stateroom accounts and book shore excursions from any screen. Princess can push this same information to all of their digital signage onboard to provide guests with the right experience, at the right place, at the right time.

MTA: Keeping NYC Moving with Digital Signage + AWS IoT

New York City's Metropolitan Transit Authority set out to improve the efficiency of its customer experience and create a cohesive user journey across all touchpoints and channels. To provide every commuter with the right experience on the right device at the right time, the MTA needed to extend the information accessed on mta.info into train stations and platforms. Enter Drupal, which empowers the MTA to use the same CMS that powers its website to push content and data to 1,800 digital signs in more than 400 stations in New York City, serving 11 million passengers daily.

This transformation required the ability to pull data from external feeds, push data to Amazon IoT Services and display the data in real time on the countdown clocks. Using Drupal as the brain behind the MTA's digital systems guarantees that content updates will be uniform across all platforms—and all devices will know what to do if there's an internet outage. By integrating resources across all touchpoints, the MTA can guarantee efficiency and establish an intuitive customer journey. Memorable and engaging experiences will not only increase customer satisfaction, but will enhance the value of a brand's goods and services.

Corelle: A Hybrid Approach to Commerce

Home goods retailer Corelle demonstrates the power of Hybrid Drupal to create a great shopping experience across its various brand websites, like Pyrex and CorningWare. The sites all use a presentation layer delivered via Drupal, in a mix of standard and decoupled components, which can call out to external or internal systems for data via API.

All of the components work together to create an optimized, relevant experience for the specific end user. Site builders and content authors control how things work, and front-end developers control what's going on inside components. With this approach, a site architect can make careful decisions about each individual component and where it is most appropriately served. Elements that are the same across all users can be heavily cached and served server side, while those that are deeply personalized are served client side. The flexibility of a hybrid approach to decoupling means Corelle can deliver both a rich website experience and high performance.

Presto: An Elevated Content Streaming Experience

As one of the largest media companies in Australia, Foxtel partnered with mega-giants like NBC and HBO and they utilized Drupal to create customer-centric experiences for Presto, their video-on-demand service similar to Netflix or Hulu (and one of the top streaming brands in Australia).

Foxtel found Drupal to be a perfect fit for their needs: vast amounts of content and personalized browsing experiences. In implementing the decoupled architecture, minimal disruption was paramount and efficient workflows to showcase new content was a high priority. The entire migration to Drupal only took three months.

With the new system, entire landing pages for new content can be created in a matter of minutes. Subscribers enjoy a wealth of personalized features and tools—they can see how much of a program they have left to watch and easily browse content they are entitled to access, without worrying about the rest.

Conclusion: A Brave New World

Until relatively recently, organizations have mostly selected Drupal due to its capabilities for building websites and providing editorial experiences with a rich array of features. But today, organizations are going further. They are serving entire digital ecosystems with content, centralizing all of their data into a single CMS.

So, the future is bright for the decoupled trend, which is built to last (by design). But that doesn't mean the Drupal community is resting on its laurels. There are many issues at play in the Drupal world about how to continue the onward and upward momentum of decoupled architecture. For example, as more and more marketers who have opted for a decoupled implementation seek the front-end tools of out-of-the-box Drupal, developers are working hard to bridge the gap between the coupled and decoupled extremes—exploring ways to bring more options to web administrators who desire flexibility and convenience.

But that's not all. Some influential voices in the Drupal community have expressed the opinion that Drupal should be decoupled by design. In other words, every single feature of Drupal possible today should be available through web services of remote procedure calls (RPCs), and Drupal should not attempt to integrate other technologies. The Drupal community is clearly thinking both granularly and broadly about how to ride the decoupled wave.

In any case, just as when IT decision-makers are deliberating with their teams about where to invest their time and energy, the next direction for Drupal will be in response to the market and what businesses require to best serve their customers. Drupal prides itself on its ability to serve organizations, platforms and ecosystems of all kinds. So if there's any doubt in your mind that a decoupled implementation can't provide a solution for your next big transition, all you have to do is wait.

The possibilities are endless, and so are the solutions.

About Acquia

Acquia is the open source digital experience company. We provide the world's most ambitious brands with technology that allows them to embrace innovation and create customer moments that matter. At Acquia, we believe in the power of community—giving our customers the freedom to build tomorrow on their terms.

CONTACT

THIRDANDGROVE

www.thirdandgrove.com

ACQUIA

www.acquia.com