The Book of CMS

Your guide to selecting the best content management system for your organization

viget

Foreword

Any well-executed digital project requires a connection between business strategy, interface design, and software engineering. When redesigning a site and re-platforming it onto a content management system, many organizations believe that they can separate these three critical areas of work. They'll define their strategy in-house, hire a design studio to think about content and create the visual design, then work with a development shop to integrate a CMS.

At best, this approach misses out on key opportunities such as prototype-driven user research, data-driven A|B testing, and a process of iterations to validate assumptions, course-correct the strategy, and produce the best possible solution. At worst, it results in a lack of vision; a bloated, unusable interface; a flawed CMS implementation that won't scale; and, ultimately, a failed project.

Since 1999, Viget has worked on thousands of web projects using myriad CMS options, from simple off-the-shelf platforms like WordPress to highly customized, complex solutions that go beyond just content publishing to integrate with third-party data sources and e-commerce tools or otherwise solve specific business requirements.

In all cases, we've worked to educate ourselves and our clients about the various options available.

This book brings together the collective experience we've gained -- and shared -- over the years into a single source.

Note that this book is not intended to reflect a comprehensive picture of today's CMS landscape. Rather, it reflects our unique experience, preferences, and recommendations based on the specific challenges we've tackled for unique organizations.

The solutions we discuss in this book have served our clients well, but we acknowledge that there's no one-size-fits-all solution for everyone. Instead, we hope you'll come away with a more informed understanding of content management systems and how to choose the most appropriate solution for your specific objectives.

I'm excited for you to learn more about one of our favorite areas of work in which strategy, design, and engineering come together to produce great results.

Brian Williams
Founder & CEO, Viget

Introduction

Who are we?

Founded in 1999, Viget is an industry-leading independent digital agency headquartered in Falls Church, VA (just outside of Washington, DC).

Led by two brothers, Brian Williams and Andy Rankin, the company has a full-time staff of nearly 70, all U.S.-based with additional offices in Durham, NC and Boulder, CO.

Viget provides a broad array of digital services, including strategy, design, development, and marketing of digital products and platforms.

Our clients include familiar names such as National Geographic, Stanley|Black|Decker, World Wildlife Fund, PUMA, Politico, and Dick's Sporting Goods.

For more information, visit viget.com.

Why did we write this ebook?

Content management systems (CMSs) are integral components of so many web-based solutions and are one of the aspects of website development clients ask us about most often.

As a result, we thought it would be helpful to consolidate our thoughts and experience on CMS options in one central place:

The Viget Book of CMS.

What will you learn?

You will learn:

- What a CMS is and how it works
- How to prepare for a new CMS
- What options you have to choose from
- Whether custom or off-the-shelf is best for you
- How to choose the right CMS
- Insights, tips, and hacks to optimize your CMS
- About CMS security, ecommerce, and the future of content
- Lessons from case study examples

Who is this ebook for?

This ebook is written in non-technical language for decision-makers, creative directors, junior developers, strategists, and project managers — folks who may not have a technical background but need a crash course in the latest CMS technology and best practices. That said, some familiarity with the web wouldn't hurt! Enjoy!

SECTION 1

What is a CMS?

A brief background

A content management system (CMS) is a web interface that allows administrators to modify digital elements on a website.

They were first created in the 1990s when organizations needed simpler, faster, and easier ways to manage their digital content.

Instead of submitting request tickets to the development team or an outside vendor, a CMS gives marketers, strategists, and editors the freedom to configure the site without touching the codebase.

While a CMS gives teams increased access to their website, it also safeguards the site's content with roles, permissions, and governance to protect the site from unapproved changes.

CMSs range widely in features and complexity, depending on the needs of the individual or the organization. A free off-the-shelf CMS can power a simple personal blog, while a custom-built CMS can provide the user interfaces for a multi-site, global content publishing hub.

How a CMS Works

At a minimum, a CMS lets you create and edit content. In most cases, the CMS is also responsible for rendering the public site. Many CMSs also let users define the types of content that they'll manage as well as the templates that the CMS uses to turn content into web pages, though in some cases, developers handle these tasks.

The CMS provides an admin interface to manage content, which is stored in a database (MySQL or PostgreSQL). When a user visits the site, the CMS retrieves the correct template, pulls the necessary content from the database, and puts them together to create the HTML that the user sees.

SECTION 2

Do you even need CMS?

Not all organizations need a CMS.

Find out if yours does.

A content management system is a great tool, but you may not even need one. If your work office is next door to your home, why would you drive when you could just walk? In the same way, your web needs may not require a CMS. For example, if your site contains static information that does not need to be updated or updates are infrequent, you may not need a CMS. But, in any scenario, it's best to be certain.

Here is a simple checklist to determine if you need a CMS:

- □ I need to frequently update/change the content on my website(s)
- □ I need a team of different roles to help manage my website(s)
- □ I need to manage all of my company's websites from one central place
- □ I need a non-technical way to have more control of my website(s)

If you checked any of the boxes above, you are probably ready for a CMS. A modern CMS will enable you to keep your website content current and empower you to update it easily and simply.

SECTION 3

What kind of CMS is most appropriate?

In this section, we discuss the various types of content management systems. First, you'll read about how to evaluate your CMS needs and the options you have. Then, we'll dive deeper to compare proprietary vs. open source CMSs. And then we'll finish this section with a comparison between off-the-shelf vs. custom CMSs. Are you ready?

Working With the Right CMS

Jeremy Fields

Front-End Development Director

Evaluating and working with content management systems (CMSs) is a large part of what we do at Viget. Our capabilities range from implementing of off-the-shelf (OTS) solutions to building customized applications with Ruby on Rails. This means we're able to work with our clients to identify the most appropriate technological solution based on their particular needs, and not force anyone into a one-size-fits-all solution.

Oftentimes, we get to work with a client to determine the best option, but occasionally they've already done some homework and come to us with an idea of what they need. Even in those situations, we attempt to complement the client's ideas with our own research and experience to help make the best possible decision for the long term.

Things We Consider When Evaluating a CMS

When choosing a CMS, the first decision we have to make is whether to pick an OTS or custom system. If the choice is OTS, we help clients select the best solution from all available options. At their most basic level, CMSs do three very simple things: store, manage, and present content. We consider each of these CMS features during the discovery phase (or sometimes sooner) of a project.

Storing Content

Content is one of the most precious resources of a website, and it needs to be treated with care. Shoehorn it into the wrong database structure and it'll be a nightmare to work with, display, and migrate. Place it in the right structure, and it becomes a breeze to slice and dice in any way you want. More importantly, it'll be future-proof when the inevitable migration to a new system happens.

Content that is ripe for OTS systems is content that is structured in pages and posts. Systems like Craft, WordPress, and ExpressionEngine are perfect for this kind of straightforward content. For e-commerce, we've become big fans of Shopify because of its easy Twig-based template creation and full style control; it even includes all-in-one order, payment, and inventory management processes.

All these systems, however, are opinionated* about how to store content and what kind of content really works best. We've found lots of ways to push, pull, and contort these CMSs to work with more unique data, but when it looks like the content demands are going to push the limits of an OTS CMS, we start looking into building a custom system.

Building custom systems allows us to create unique solutions that perfectly suit a client's needs. Creating a custom CMS necessitates a careful consideration of data relationships and integrity. We recommend custom systems when content is more than presentational (posts and pages) and may have complex relationships or taxonomies.

Managing Content

The administration (admin) interface —the place where content is managed — is typically seen and used by only a few people on the client's side, but it can have a huge impact on the quality and consistency of content delivered to visitors.

While OTS CMS admin interfaces are highly polished, they too have an opinion about content structure, as well as workflow. If the content structure matches the CMS, it's great. However, the further an OTS CMS is pushed and customized, the more confusing the workflow can become. WordPress can be especially vulnerable to this problem, as its admin is based around managing posts and pages, but can be highly customized and pushed way beyond its

*Roughly, opinionated software means: "The framework imposes a certain way of working on you [...] there's clearly one right way of using the framework which is nice and easy, and any other way of using the framework makes your life difficult." (source) core capability.

With a custom system, we're able to tailor the admin workflow to a client's specific needs and requests. Need to bulk import your proprietary list from a customized 10-year-old Excel spreadsheet? A custom CMS makes that easy. The development of our custom CMS Colonel Kurtz came out of our experience building custom admins and our desire to remove the interface as a barrier to content management.

Presenting Content

Database systems may seem arcane and confusing, but templating (content display) is the fun part! Or it can be, as long as we've chosen our CMS solution well.

Like content storage and admin interfaces, OTS CMSs generally have some opinion or structure around content display. Craft is the least opinionated, and lets us start from scratch and craft (hence the name) the site from the ground up with an easy-to-use templating language. WordPress, on the other hand, is built on a theming system, allowing users to easily install and swap themes on a whim. This is generally a good thing, but means building a theme for WordPress has a specific structure and workflow that's not always ideal and can occasionally impose restrictions. Additionally, templating languages such as Twig and WordPress's loop make getting straightforward content into templates … straightforward … but accessing more complex relationships can be very difficult.

Choosing a custom CMS opens up the possibilities for content display, brings the full expertise of our developers into the mix, and allows unique and complex mashups of data. Even when a heavy query can work in a OTS CMS, the resulting database work can be costly in server performance and result in slower loading pages. Custom development allows fine-tuning of queries to ensure fast server responses.

Getting Around the Pitfalls

Not all projects are ideal. Sometimes we're tasked to work with a CMS that isn't great for the application. There are many legitimate reasons for a less-than-ideal match:

Budget: Comparing the costs of OTS to custom CMSs can be deceiving. It's important to consider the upfront cost of a CMS, as well as the cost of customizations and long-term maintenance:

- **Upfront cost:** OTS CMSs get you off the ground quickly and with little ($299 for Craft) to no (free for WordPress) cost. Custom development has a much higher upfront cost, but can pay off in the long run.
- **Configuration cost:** An OTS CMS may seem cheaper, but if it has to be highly customized to work, its cost can ramp up quickly, sometimes making it comparable to a custom solution.
- **Maintenance:** Over the long term, the cost of maintaining the server and CMS needs to be considered. Good OTS systems are regularly being upgraded and come with easy "one-click" updates that allow even non-technical admin users to keep them up-to-date. Custom systems are built to be robust and may better fit long-term needs; however, additions and upgrades do require additional development expense.

External needs: CMS choice can be driven by factors beyond content and templating. Ease of use, ubiquity, and familiarity can all factor into a decision.

Internal preferences or technological comfort level: Sometimes simple comfort level with one system, wariness with another, or a key stakeholder's bias can drive a choice.

Our approach in these cases is to take a careful look at the situation and gain a better understanding of the client's motivations. Sometimes we're able to reframe our recommendations so the client is able to understand our rationale. Other times, we get a deeper understanding of the client's needs outside of databases and templates.

In either case, we make sure we're able to stand behind the CMS that's chosen and build a site that we're proud to deliver.

Wrap Up

Choosing and working with a CMS is a unique process for every client and project. At Viget, rather than putting every project into a one-size-fits-all solution, we've developed expertise in a variety of options so that we're able to help our clients to choose the best technology that fits their needs.

Proprietary vs. Open Source Content

Zach Robbins

Senior Digital Strategist

The land of Content Management Systems has changed greatly over the years, from first only managing content in hard-coded HTML files, to Dreamweaver, to hundreds of full-blown systems created for the sole purpose of adding, editing, and removing content. We often help clients decide what system makes the most sense for their organizational and user needs, as the decision is nothing less than intimidating.

A question that often initiates this process is categorical comparison, between the Proprietary and Open Source routes. In this case, I'm defining Proprietary as a system built, owned, licensed, and supported by one company, normally requiring significant financial investment. Open Source, in this case, is the development or creation of a system using open source technologies and frameworks, like Ruby on Rails or Python. There is a third category that I'm leaving out, for the time being, including off-the-shelf (OTS) options, that may technically be either Open Source or Proprietary, but muddles the lines between the categories (like Wordpress, ExpressionEngine, Craft, or Drupal).

There certainly is no clear "winner" for everyone, as the tool should enable organizations to fulfill business objectives, and the makeup of organizations, those objectives, and end users, vary greatly. But to start, here are the pros and cons of Proprietary and Open Source Content Management Systems that can aid in making those decisions.

Proprietary

PROS

- **Predictability** - features are documented and can generally be demoed; pricing is consistent
- **Options** - there are plenty of proprietary CMSs out there
- **Robust** - generally come packed with a lot of features
- **Cost to implement** - since the system is already built, there are often less upfront costs

CONS

- **Less current** - proprietary CMSes are generally not using the newest or best technology, because there's a lot invested in a large legacy system
- **Licensing fees** - you pay to use the software, you do not own it Handcuffed to one or just a couple provider(s) - supported only by the company that sells it or limited implementation partners–if you're unhappy with service, support, flexibility, there are no other options
- **Lack of customization** - "what you see is what you get"–the system is not created for your unique needs, but generalized to meet the needs of all their clients
- **Lack of flexibility** - do you want to do something the system doesn't do? It is often very difficult to get new features developed for a proprietary system, because of all the generalized dependencies across their clients

Open Source

PROS

- **Customizations** - an open source system can be built and customized to your specific needs, both in initial setup and in the future; as those needs change, so can the system
- **Flexibility** - built to easily integrate with other technologies and systems
- **Supported by a community, not a company** - open source developers are able to pick up and learn a system to maintain and extend it; you would not be locked into one provider for support or further customization
- **Existing frameworks** - reusability of code assets, including features and systems that can easily be dropped in to create a complete system, so you're not starting from scratch every time
- **New technology** - often use the newest technologies, staying up to date on bugs, fixes, and new tech advances
- **You own it** - when it is built for you, you own that instance of the software, thus you do not have to pay to use it, only to maintain it (in-house or by a provider)

CONS

- **Upfront cost** - since they are highly customized, there is more upfront effort to get it off the ground
- **Less "out of the box" features** - some features that come with proprietary systems might be expensive to create with open source
- **Less predictable support** - support and maintenance only happens as needed, instead of regularly, and is generally less predictable

CMS Solutions: Custom-Built on a Framework vs. Off-the-Shelf

Ryan Stenberg

Developer

I want a content management system. What are my options?

When it comes to acquiring a content management system (CMS), there are two common avenues:

1. Use pre-built, off-the-shelf software.
2. Custom-build it.

It's important to understand the differences -- what each approach simplifies and complicates and what that means for budget, timeline, and longevity.

Let's compare.

An off-the-shelf (OTS) CMS is a complete software product. It has a generic set of functionality and a pre-defined user interface for managing content. Typically, OTS CMSs have some kind of plugin-system to allow for extensibility and customization as well.

A custom-built CMS does not mean built from scratch. Most commonly, it is a web application built on top of an open source web framework, often leveraging other open source libraries as building blocks.

The underlying data model and the user interface are tailored to the business domain. All pieces are crafted together with the core requirements and business needs in mind, which drive the shape of a custom CMS.

Let’s get a little technical.

All CMS solutions are built for the web. They’re all written using a programming language like PHP, Java, Ruby, or Python. They’re all built with a web framework that supplies central functionality to power web applications. The top-most piece is the user interface (also called a UI) that defines the way users interact with and manage content inside of an admin.

While frameworks do a lot of work, they don’t do it all -- they are, after all, just frameworks for the rest of the application. They only go as far as to give you the tools necessary to build a web application, regardless of specific CMS or domain.
Off-the-shelf CMSs are built on top of web frameworks -- some open source, some proprietary -- with a standard, domain-agnostic interface for managing content. This means that the work of building CMS-specific functionality is already done for you. On the other hand, with a custom-built CMS, you’ll need to build your own interface for content management on top of the framework.

What does that mean for me?

Most importantly, the decision to custom-build versus buy an off-the-shelf solution hinges on two things: your business needs and the constraints of the project.

There are many cases where well-established, off-the-shelf CMSs are a perfect fit for a business’ needs. Many popular off-the-shelf CMSs have interfaces that allow you to point-and-click your way through CMS configuration. They’re both easy to use and streamlined. Many are relatively inexpensive, and some are even free. When the CMS has a plugin system, it’s possible to customize and extend the CMS functionality by writing code that interacts with the CMS plugin interface.

Using an OTS CMS forces you to rely on the software’s builders to have designed the solution that accommodates your business needs. These one-size-fits-all solutions can’t possibly cover every imaginable need, but in many case they create something that adequately covers requirements and can be modified through add-ons.

With a custom-built CMS, you have more ownership over what’s being built -- giving you more control and influence over the end-product. You’re in a position to tailor the solution to your specific business needs. Not only are you involved in shaping the solution, but you’re involved in the shaping the build process, too.

So, what do I need?

It really comes down to the core requirements and constraints for your CMS. Given unlimited budget and time, a custom CMS will always be better tailored for your business needs. OTS CMSs save you the budget of building the CMS interface, but there is more generalization and less customization.

Further, custom-built CMSs offer greater extensibility than an OTS CMS, where you're limited to pre-established interfaces by which you can customize the software. If you need to add domain-specific functionality with a custom CMS, you build internal features.

With an OTS CMS, you have to build custom functionality through CMS plugins. This means that you're limited to the options they provide and the flexibility of the interface.

Let's talk cost.

There's a higher price tag for a custom-built CMS with an open source framework than an OTS CMS that's already built. However, the real cost lies in the functionality gap between fundamental CMS capabilities and what your business needs and domain requires.
If you need a great amount of customization, it may be more cost efficient to choose a custom CMS due to the higher level of friction when extending OTS softwares to fit your specific needs. There are definite cost-savings in OTS CMSs, as long as your needs fit within the capabilities of the CMS.

“Most importantly, the decision to custom-build versus buy an off-the-shelf solution hinges on two things: your business needs and the constraints of the project.”

Appendix

This section is a cursory overview on how to choose a custom-built versus off-the-shelf CMS. Want to learn more? Here are some great resources:

Web Frameworks

- Rails
- Ruby Gems - A cloud-hosted repository of Ruby libraries (Ruby)
- Ruby Toolbox - A search tool that finds the right Ruby gem or the job
- Bundler - A Ruby dependency management tool
- Laravel (PHP)
- Django (Python)

Popular Off-The-Shelf CMSs

- Wordpress (PHP)
- ExpressionEngine (PHP)
- Craft (PHP)
  - Straight Up Craft - Plug-ins for the Craft CMS
- Drupal (PHP)
- Joomla (PHP)

Off-The-Shelf

- Good for when there are few requirements besides content management
- Smaller, contained scope
- Generally smaller budgets
- Extensibility mostly relies on existing plug-ins
- Only requires front-end developers

Custom Built

- Good for business-specific processes
- Larger scope
- Generally requires larger budgets
- Can use third-party integrations
- Requires both back- and front-end developers
- Good for when you need high-performance or advanced search

SECTION 4

What CMS should you use?

We've covered several approaches to CMSs, including proprietary vs. open source, and off-the-shelf vs. custom. Now it's time to take a closer look at the specific CMS options to choose from. In this section, we're going to compare the most popular off-the-shelf CMSs, tell you about our favorite, and also our preferred solution for custom CMS development.

Going Off-the-Shelf: WordPress vs. Craft vs. Drupal vs. Shopify

Megan Zlock

Front-End Developer

You've made it this far, but there are certainly a lot of options to choose from. And each choice for a CMS can have some direct influences on your website and your business. So, let's take the most popular products out there and break this down a little:

	Wordpress	Craft	Drupal	Shopify
Stability, Scalability and security[1]	Good	Great	Excellent	Excellent
Content Flexibility	Great	Excellent	Great	OK
Support, Community and Notoriety	Excellent	Good	Great	Great
Features & Add Ons	Excellent	Great	Excellent	Good
Ease of Use	Great	Excellent	OK	Good

[1] The stability and security metric assumes that you keep your CMS updated to the newest version

WordPress

You've probably heard of WordPress already, as it has a large community and is used for a staggering 25% of websites. Yes, that's a lot of sites. But is WordPress right for your site?

WordPress's power comes from its ability to create a site with a specific structure very quickly. If you need a website for your business that is 5 pages and a blog that needs to be up tomorrow, WordPress is your go-to. There are also an incredible number of plugins and add-ons, so you can add things like integration with your CRM platform in just a few clicks. You can also pick from a huge array of WordPress "themes" (some free, many you have to pay for) to determine the look and feel of your website. The better ones include some great customization options.

WordPress Multisite is also a great product for many small sites that need some level of connection and shared administration. I've seen it used well for multi-location events like travelling conferences. Each site (e.g. conference location) has the same pages and structure; they just need a little tying together.

Where WordPress gets a little shaky is sites with more unique content and data relationships. Many, many sites on WordPress fit into that 5-pages-and-a-blog model, so that is what out-of-the-box WordPress is optimized for. From there, WordPress can be customized and molded to an incredible degree serving enterprise and small businesses alike, but it does take effort to do so. Be prepared to spend smartly and get a team who knows how to work with the platform well and utilize its best features. There are many "WordPress experts" out there who can build you a site, but results (and maintainability of your site thereafter) will vary.

On one last note, I must say: beware of overusing plugins with WordPress. While powerful and one of the best selling features of WordPress, they can also be your worst nightmare. Using too many at once can kill the maintainability and performance of your site. Each plugin added is a failure point. The plugin developer could stop supporting the plugin, the plugin could stop working with a WordPress update, it could have shipped with a security flaw, etc. If you need 20-plus plugins to run your website, then a different platform (and maybe a custom solution) may serve you better.

Maintainability: WordPress offers regular updates approximately once or twice a month. Remember to update as often as possible to avoid security vulnerabilities. Make sure your plugins still work with every update.

Stability: WordPress is very reliable. Look out for hacking attempts which can cause some problems. Since WordPress is used so much, it is a high-profile target for hacking.

Adoption and Development Effort: Low - Medium Effort. If you don't have a tech team and are fine with the look-and-feel of an existing WordPress Theme, then you can create your website yourself. WordPress is well-documented, and a small development team can learn how to implement a custom theme very quickly. PHP knowledge is recommended, but not required.

Craft

Unlike WordPress, you may not have heard of Craft. And that, I think, is highly unfortunate. As you'll read below, we love Craft here at Viget and tend to prefer it over WordPress.

Craft is really amazing when it comes to ingesting, displaying, and relating whatever kind of content you can throw at it. It doesn't matter if you have just a simple blog, curated portfolio pages with custom designs, or complex relationships between your organization's mission pages and the rest of your content. Craft can handle that beautifully.

Added bonuses for content editors are Craft's Live Preview and Matrix features which are included in the core platform. The Live Preview is exactly what you'd expect – you can see your page change live as you edit the content. No more opening a preview in a new tab; Craft shows you your changes with a minor 1-2 second delay in the same window. I won't go into Matrix fields except to say that we've used them to great effect to make page builders for our clients. You can start with a completely blank page and use pre-created building blocks to build up what you need and only what you need on a page. Need a new block? A new page builder block can be added with far less effort than re-designing a whole page AND you can then use that new block on any other page. Granted, other platforms can achieve this flexibility as well, but only with a plugin or add-on.

However, you cannot spin up Craft nearly as quickly as WordPress. Craft is really meant for one-off custom websites and takes some tailoring during set-up. This is usually a plus since this model gets around some of the pitfalls of WordPress development, but it does mean you will not have a new site within 24 hours. Complex functionality might also take a little while since Craft doesn't have the plugin pool that WordPress does (although this is growing).
If you go with Craft, get a development team, give them a few weeks, and you'll have something amazing.

Maintainability: Craft updates are near-constant and are very small and safe unless it's a big version change. As with WordPress, update as often as you can while checking your plugins.

Stability: Craft is a newer platform, so there is the occasional hiccup with the system. Usually bugs are fixed quickly in their fast update cycle. Also, get hosting with higher memory allocation (minimum 2GB RAM) since Craft's behind-the-scenes tasks can hog memory.

Adoption and Development Effort: Medium Effort. Craft is incredibly easy to learn and requires no PHP knowledge (unless you need to write a plugin). A front-end developer or two can easily manage development for a Craft site with little training, but will need time to build from scratch.

Cost: It is worth noting that, unlike WordPress and Drupal, Craft is not free. Well, you can go with the Personal License for free, but that limits you to only one user account. More practically, you'd most likely end up with the Pro plan which is a one-time $299 license fee that covers you for life. If you want to read more about Craft's licenses, check out their pricing page.

ExpressionEngine

I'm including ExpressionEngine (EE) on this list, but only as a footnote to Craft, unfortunately. Craft was built by former ExpressionEngine add-on developers, so Craft borrows a lot from how EE works. In fact, Craft covers pretty much any use-case you would have for EE and is much easier to use. For those who are already on EE, the platform has been continuing to update and improve, but I wouldn't recommend ExpressionEngine on a new site build whenever Craft may be an option.

Drupal

Drupal is a complete beast when compared to the other content management systems on this list. It is highly customizable and very powerful when you know what you're doing. You can do a lot with Drupal, but working with it can be an art.

Like WordPress, Drupal comes with a lot of plugins and add-ons, which they call "modules". Modules mean you can drop-in sections of pages that are pre-built and ready to go; they just might need some re-styling. This is great when you want to add content and features quickly, and Drupal allows for some pretty flexible set-up, especially with the Page Manager Module.

You may not be able to use a page-builder as a content manager like we've done with Craft, but your developer can easily move things around and do a lot of customization quickly. Drupal also has designed and built its content controls so that they exist on top of your templates. The publicly viewable page, when logged in, is the same place where you edit content. Just click the block of text you want to change and edit away within the design.

Drupal is powerful but has a really steep learning curve. Drupal Core (the minimum feature set) does only so much on its own, so knowledge of available modules is key. If you choose to go with Drupal for your website, you'll want to find an experienced team who is well-versed in Drupal development. Training your own staff to work with it might take some time. This becomes especially important when you approach the point in time where you may need to update Drupal. Small updates are released often, but larger version updates (like the current conversion from Drupal 7 to Drupal 8) can take a lot of time and effort. All those essential modules need to get updated as well and it can become a real nightmare in the wrong hands.

Note: WordPress and Drupal have similar issues with updating and using too many modules/plugins. However, in Drupal, modules are vital to build your site, so you'd be doing yourself a disservice to restrict your module usage. Try to pick popular, well-maintained modules when you can to make things easier later. Or build your own and maintain them yourself. Luckily, Drupal doesn't hit the same performance and bloat issues as WordPress the more modules/plugins you pile on.

Maintainability: Drupal updates are somewhat sporadic and cover several versions. Small updates are fine, but be ready to invest a lot into whole version updates (like moving from Drupal 7 to 8). New Drupal versions are only released every few years, so you'll likely need a redesign when the next one comes up anyway.

Stability: Drupal is very stable with caching set-up correctly and is built to scale for very large, heavy-traffic sites. Drupal is also very transparent and vocal about any security updates.

Adoption and Development Effort: High Effort. Drupal has a very high learning curve which surfaces to most everyone who uses it, from developers to content managers. Get a team that specializes in Drupal development to succeed with this platform. Their community has been in somewhat of a decline, but Drupal 8 which released late 2015 could bring it back in full-force.

Shopify

When thinking about content management systems, Shopify might not cross your mind at first. In fact, it is a little bit of an oddball to compare to these other systems, but I think it's worth a mention. If the core of your business is products and sales, then Shopify is a great place to start. It acts as a great shopping platform with all of the tools you need as a business owner built in: order fulfillment, optimized shopping cart experiences, reimbursements and coupons, payment processing, etc.

Some think of Shopify as an add-on. Of course, that reputation is justified: you can use Shopify for just your Cart experience and then connect to any of the previous platforms with plugins and modules. But Shopify can do a lot on its own too. Like WordPress, you can choose from existing themes with some customization options or have a developer make a custom theme for you to personalize the look-and-feel.

Where Shopify gets a little funny is if you want to have a larger site with more custom functionality. If your site is just products, categories, and a few one-off pages, then you're all set. But add unique content and you'll have to hunt for a plugin that suits your needs, which you may or may not find. Shopify's templating is also very rigid. Themes dictate what content displays on a page and there is little you can do as an administrator to add or remove sections of your pages.

Note: The monthly price of Shopify might be scary (you'd probably need at least the Basic $29/month plan at minimum), but keep in mind that includes hosting fees and support; costs that are not included with the other platforms. There is also an Enterprise version, Shopify Plus, which can range into the thousands of dollars per month. Now that number really is scary, but it does include dedicated support, scaling for high traffic, and an Account Manager.

Maintainability: Shopify is constantly making improvements and, because their platform is Software as a Service (SaaS), updates are pushed to your site as soon as they're available. However, you may want to keep up with their blog so you're aware of what updates are coming. They have on occasion updated the Shopping Cart experience (over-writing any custom styles you may have put in place). Overall, those changes are for the better, since they put a lot of research into the optimal cart experience.

Stability: As a Software as a Service, Shopify is in charge of keeping your site going. They claim a 99.99% uptime as one of their features.

Adoption and Development Effort: Low - Medium Effort. One administrator can easily set up a Shopify website if s/he is happy with an existing theme. Once you get into custom themes, then picking up a front-end developer will do. Setting up the development and deployment structure with your tech team is a big hassle (and only one developer can actively work on it at once), but actual theme development is not difficult and easily learned.

Cost: Unlike the other options on this list, Shopify is a SaaS and requires a monthly fee. Check out their pricing tiers to see what might be right for you.

WooCommerce, Craft Commerce, Drupal Commerce

As a footnote to Shopify, be aware that WordPress, Craft, and Drupal all have their own e-commerce solutions. WooCommerce is one of the most popular e-commerce plugins for WordPress, Craft Commerce was just released for Craft, and Drupal has Drupal Commerce as its big e-commerce option. So don't feel tied to Shopify if e-commerce is your thing.

Try to decide based on the needs of your whole web presence and which has the best tools for your business needs.

Hopefully this run-down was helpful! Of course, there's only so much I can convey in one blog post, so you'll have to weigh the options on your own to figure out what might work for your specific needs. You can always give us a shout if you need some help deciding.

Remember to factor in hosting and maintenance costs which will be a factor on any platform. And think about your website's future and everything you might want it to do three years from now rather than just your immediate needs. Platform decisions will stick with you for a long time to come and can shape a lot of how customers see your business or organization.

If nothing seems to quite fit your unique problems, then maybe a custom solution is the right choice and will pay for the added effort over time.

Craft vs. WordPress: The Good, the Bad, and the Ugly Data

Megan Zlock

Front-End Developer

Any web developer who has been working in the industry for more than a few days has probably heard of WordPress. Stay for a couple more months and there's a good chance you've worked on a WordPress site — it's a popular platform since it's well-known, easy-to-use, and free.

You may have also heard of ExpressionEngine, Drupal, Joomla, and a few other CMS heavy-hitters. They all have their benefits and flaws, which is a topic for another time. I'd rather talk about the next up-and-comer and my new favorite, Craft.

Craft is a small CMS that was developed fairly recently by ExpressionEngine add-on developers Pixel & Tonic. Having worked with ExpressionEngine for a while, it's obvious these guys really know the pain-points in any client-facing CMS. Everything they've built into Craft solves a problem I've had on almost every site I ever made with WordPress. If I had to make a CMS myself, it would probably resemble Craft pretty closely.

So, how does Craft stack up against the industry go-to?

The Good

Craft is like WordPress if it was stripped naked and then clothed in Advanced Custom Fields.

WordPress strives to give its users as much as possible out-of-the-box, whether that user is a novice blogger or a talented developer who needs a good admin panel. The result is often too much functionality, which forces developers to strip out or disable features to meet the needs of your very custom website (like Comments or the Links Manager, pre WordPress 3.5).

Craft, on the other hand, starts with just the basic building blocks and minimal defaults. The Sections and Fields it does provide let you build up your content types and inputs to get to the custom dashboard you want. I would even say it takes less time to build Craft up to WordPress status than it does to fight with WordPress settings to take out all of those Blog-centric things you don't need.

Another win for Craft is that its Fields build your content interface in much the same way as WordPress's Advanced Custom Fields plugin (a must-have tool with WordPress in my book), but without downloading and installing another piece.

Building instead of Manipulating

This difference in the platforms' philosophies is apparent in their templating tools. WordPress supplies and spits out a lot of its own default HTML that requires manipulation of the API to change. Craft comes with nothing. No HTML at all. Which is glorious.

There is no API to coerce into the markup that you want and there is no lazy settling for the default because there is no default. As the O.C.D., semantic-obsessed front-end developer that I am, this is perfect. I set all my own HTML (with Twig templates in Craft), styles, and attributes from scratch. <3

Relationships are Hard

One thing that is a recurring pain to work around in WordPress is its lack of relationships between Post Types. If one Post Type needs to be related to another Post Type, you have to make some middle-man taxonomy or category to relate them, do some PHP magic to make your own custom inputs in the post editing screen, or find a plugin that meets your need. With Craft, Entries (the Crafty brother to Posts) are easily related with a simple Field type. Drag, drop, done. Hallelujah!

A Common Example: Say a non-profit site has a Section of "Social Causes" pages and they want all of their News and information to be categorized by and related to these Causes. Creating those relationships is far more difficult in WordPress.

Welcome to the Matrix

One other great client-serving feature of Craft is its handy Matrices. With Matrices, you can set up your interface in Blocks, which your client can then use to build their page — it's a win-win-win for clients, designers, and developers. Clients can control the order of their well-designed content without hacks or careful content input into a catch-all Editor; designers can rest assured that their designs won't be fouled-up by user error; and developers have complete control over the mark-up which is generated by these blocks.

In comparison, WordPress can do repeating blocks of the same content in a row. Not too bad. The catch is that you need Advanced Custom Fields, plus their Repeater Field addition which, unlike the main plugin, does cost a few bucks.

Less PHP! And More PHP! Wat?

This last "Good" point is relative to the type of developer you are, so I admit this could easily go in the "Bad" section. Although Craft is built off of PHP like WordPress, Craft uses Twig templates. This is great for front-end developers already familiar with other templating languages like Handlebars or Liquid, but may not be for all you PHP gurus. I personally like the change, since loops feel a little less clunky and the data syntax is closer to JavaScript.

You will find PHP in Craft's plugins. Since Craft does not have Themes, there really isn't anywhere to put your common template helpers and useful functions. Instead, be prepared to write your own plugin to add what you need. Gone are the days of plopping in a random PHP function into functions.php. This is great since your site is then based off of good modular code and proper PHP Classes, but you may need to read up a bit to get there.

The Bad

Google Maps vs. Apple Maps

As Apple found out the hard way, it can be tough to beat the big dog in the market. WordPress has been around longer, has more resources, and has more developers actively contributing to it. If you hit an issue with Craft, resources are few. I'm sure the Craft community will catch up, but in the mean time I recommend making a few new Twitter friends. @Craftcms, the folks from Pixel & Tonic themselves, and Viget's own Trevor Davis are good follows. Those passionate about Craft are happy to answer questions.

Craft? Crafts? Kraft? Minecraft?

Craft picked a pretty tough name in the Googleverse. Searching for common problems becomes a real chore, simply because you have to sort through 50 articles about Minecraft before getting to the few sources that are available. Compare that to WordPress results, which will stretch for pages and probably include at least five well-written solutions to your problem on Stack Overflow.

I search "Craft CMS" for the best results and include "Twig" if it's a templating problem.

The Deal-breaker

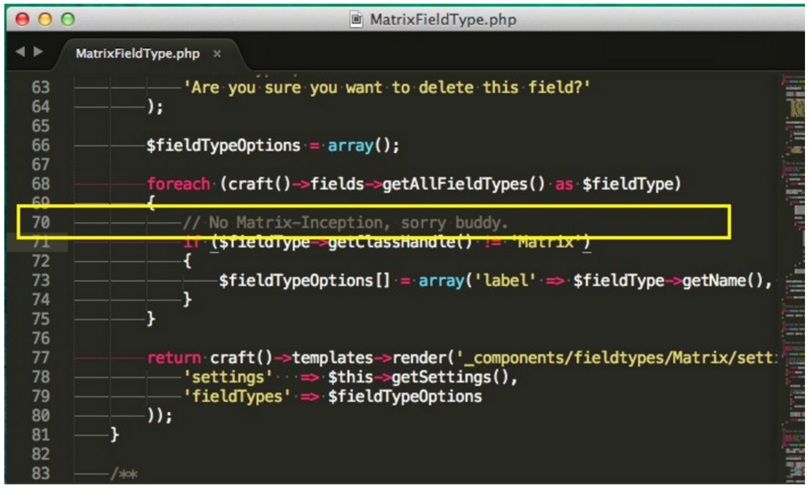
One obstacle for some clients when it comes to Craft is the price. One can't help but second guess the choice to throw down $299 when the usual go-to CMS is FREE. It's not such a tough sell on the agency level since clients have usually come prepared to spend much larger sums, but freelancers might have a harder time justifying the cost. Even so, I recommend you try — it's a one-time fee that then unlocks all of Craft's best features and goes toward the support and further development of the system.

Can I get both Pills? The Matrix + Inception

For good reason, Pixel & Tonic have restricted Craft's Matrix a bit. Some things I would love to see in future releases:

Re-using Blocks: I'd like to use already made Fields or sets of Fields inside Matrix blocks. If the same "module" exists both inside and outside of my Matrix, keeping the data the same requires careful duplication.

Matrix Inception: Please?



```
            'Are you sure you want to delete this field?'
        );

        $fieldTypeOptions = array();

        foreach (craft()->fields->getAllFieldTypes() as $fieldType)
        {
            // No Matrix-Inception, sorry buddy.
            if ($fieldType->getClassHandle() != 'Matrix')
            {
                $fieldTypeOptions[] = array('label' => $fieldType->getName(),
            }
        }

        return craft()->templates->render('_components/fieldtypes/Matrix/sett
            'settings'   => $this->getSettings(),
            'fieldTypes' => $fieldTypeOptions
        ));
    }

    /**
```

Craft leaves an easter-egg comment in their code:

"No Matrix Inception, sorry buddy."

I've hit data that could really use one more level inside of Craft's Matrices. (A page builder matrix with a repeating section inside of it does not compute.) It sounds like Pixel & Tonic are working on this, so my wishes should be fulfilled soon.

have a harder time justifying the cost. Even so, I recommend you try — it's a one-time fee that then unlocks all of Craft's best features and goes toward the support and further development of the system.

The Ugly Data

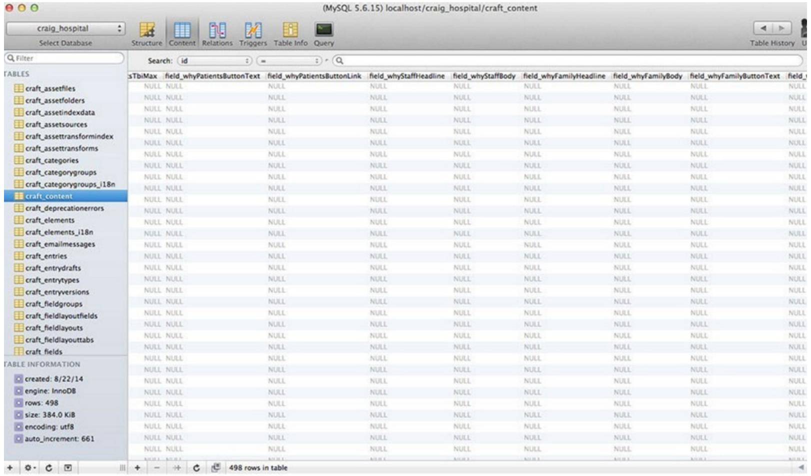
I am not a database-taming type of dev. I cannot whip up SQL queries and the like to properly clean and clear up my database. I may not be the right person to comment on this particular area, but I will say that the data for both of these platforms is rather Ugly.

WordPress

The biggest headache when working with WordPress in multiple environments is, of course, its data. Of all things to store in your database, WordPress stores the root URL of your website. If you've worked with WordPress, you know exactly what I'm talking about. If a developer ever has to do a manual find-and-replace within a SQL file, something is very, very wrong. Other undesirables include excessive settings management and a single data model for any kind of content.

Craft

Craft data is beautiful until you get to the "craft_content" table. And then you realize where all your great Fields went. After careful curation of Fields, Field Groups, and to which Entry Types they are assigned, each Field becomes a new column in this table and you're hit with a wall of NULL data.



- craft_assetfiles
- craft_assetfolders
- craft_assetindexdata
- craft_assetsources
- craft_assettransformindex
- craft_assettransforms
- craft_categories
- craft_categorygroups
- craft_categorygroups_i18n
- craft_content
- craft_deprecationerrors
- craft_elements
- craft_elements_i18n
- craft_emailmessages
- craft_entries
- craft_entrydrafts
- craft_entrytypes
- craft_entryversions
- craft_fieldgroups
- craft_fieldlayoutfields
- craft_fieldlayouts
- craft_fieldlayouttabs
- craft_fields

created: 8/22/14
engine: InnoDB
rows: 498
size: 384.0 KiB
encoding: utf8
auto_increment: 661

sTbiMax	field_whyPatientsButtonText	field_whyPatientsButtonLink	field_whyStaffHeadline	field_whyStaffBody	field_whyFamilyHeadline	field_whyFamilyBody	field_whyFamilyButtonText	field_
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

498 rows in table

Sometimes you hit areas of the database where it's just column after column of NULL data.

Such NULL. It seems like such a waste when each Entry Type could be broken into its own table to greatly reduce the NULL.

Aside from that, I would say Craft stores too much of its structure as data only. Again like WordPress's Advanced Custom Fields, Craft stores the new Fields you add to your admin panel as data. This is a big drawback when juggling environments.

Advanced Custom Fields later solved this problem by adding an export tool to spit out and store your Fields as a PHP object. I hope Craft will soon follow suit so I can stop adding “To Be Used Later” fields to production sites in order to preserve the flow of data.

TL;DR: Craft is better than WordPress for more custom websites because developers can build instead of manipulate. This philosophy applies to creating the admin, content entry, and templating. Prominent downsides include difficulty finding solutions, the price, and storing all Fields as data only.

Why We Love Craft CMS

Trevor Davis

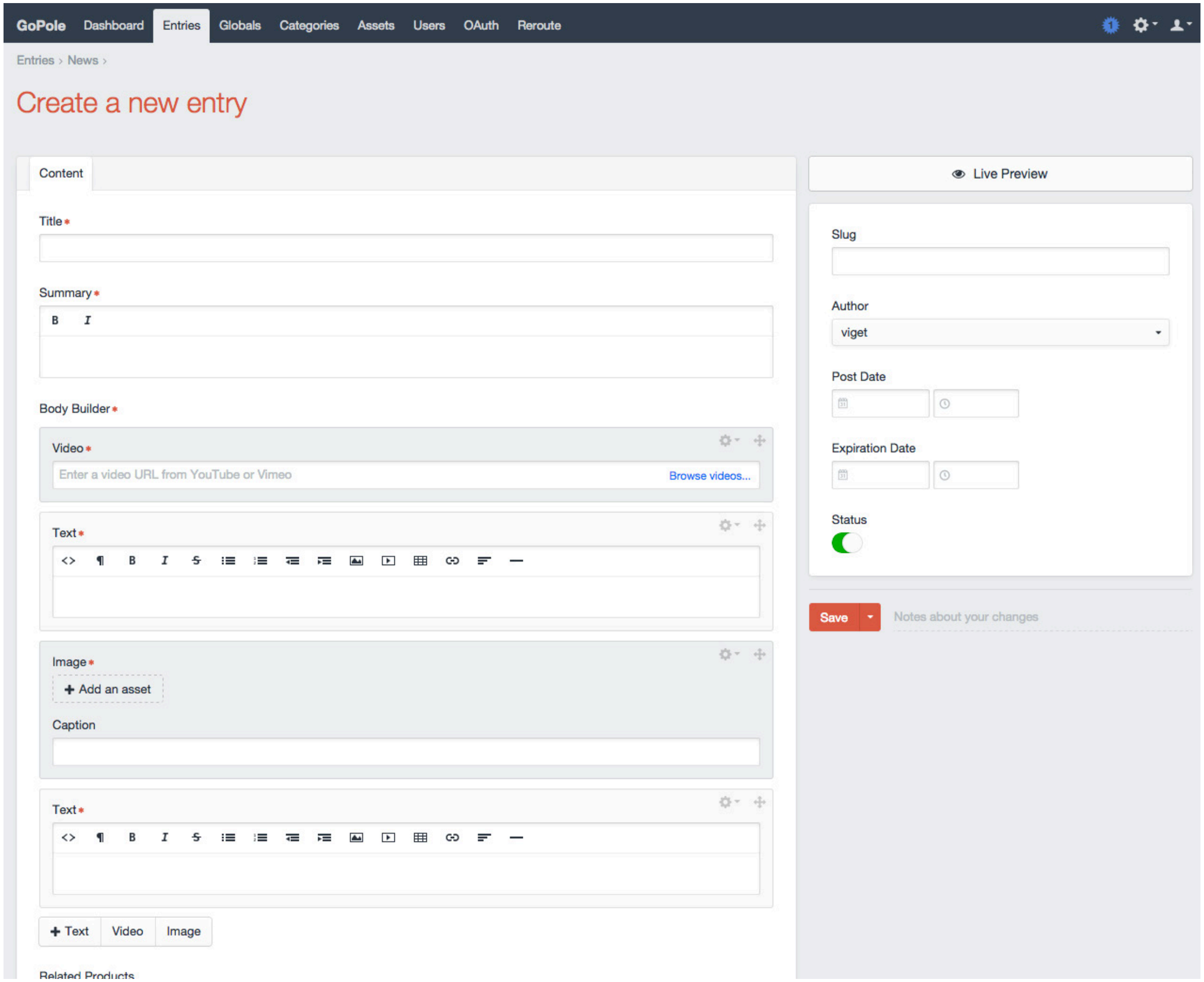
Senior Front-End Developer

Megan wrote the post above discussing Craft and WordPress that sparked an internal discussion about the difficulty we've encountered selling less popular content management systems—basically, anything other than WordPress or Drupal—because so many clients have only heard of the big boys. Viget has dealt with this issue for quite a while since we have long been a proponent of ExpressionEngine. Having had extensive CMS experience, including experience building my own, I have to say Craft is the best CMS that I have ever used. Hopefully, this post will help explain to both developers and clients why I've come to this conclusion.

It's natural to want to compare Craft to other CMSs, but I think it's like comparing apples to oranges; each CMS does certain things well. For example, WordPress is a blogging platform that can be tweaked (a lot) to manage non-blog content. Craft is designed specifically to be flexible enough to manage all types of content without a lot of hassle. If someone told me that they wanted a simple blog, I would recommend WordPress because it is really good for bloggers. It's also worth mentioning that Craft is an off-the-shelf (OTS) CMS and not a framework for building custom software applications; we use Ruby on Rails for those projects. So, with that in mind, the rest of this post is just going to highlight the benefits of using Craft.

Beautiful, Responsive Control Panel

The Craft control panel is beautifully simple. The design gets out of the way and makes it easy to edit and manage your content.



Create a new entry

Title*

Summary*

Body Builder*

Video*

Enter a video URL from YouTube or Vimeo

Text*

Image*

Caption

Text*

Related Products

Slug

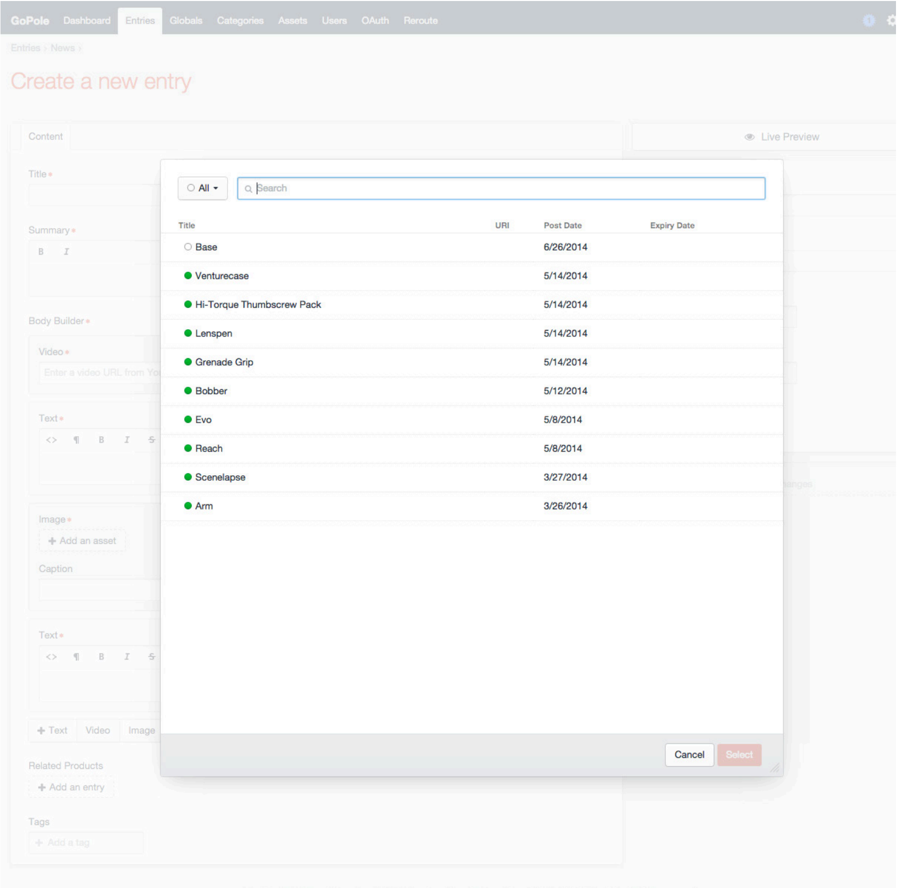
Author: viget

Post Date

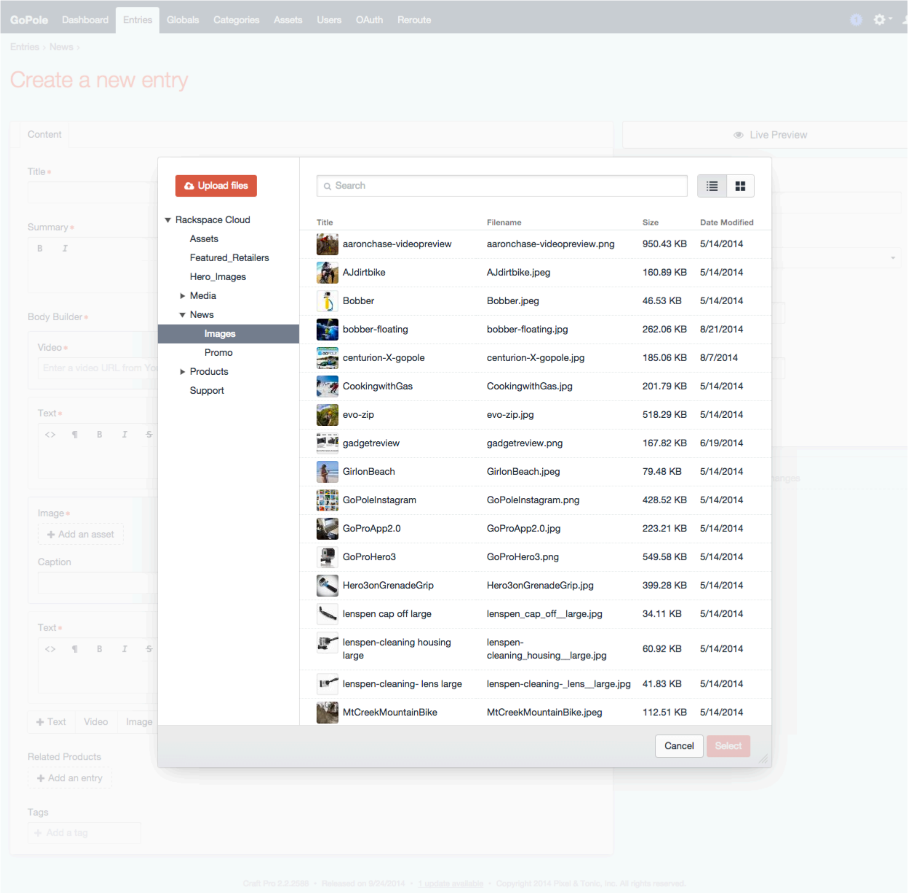
Expiration Date

Status

> Craft is insanely flexible and super user friendly. We trained our whole team how to add and edit content in less than 15 minutes.
>
> **— Ryan Vosburg, GoPole**



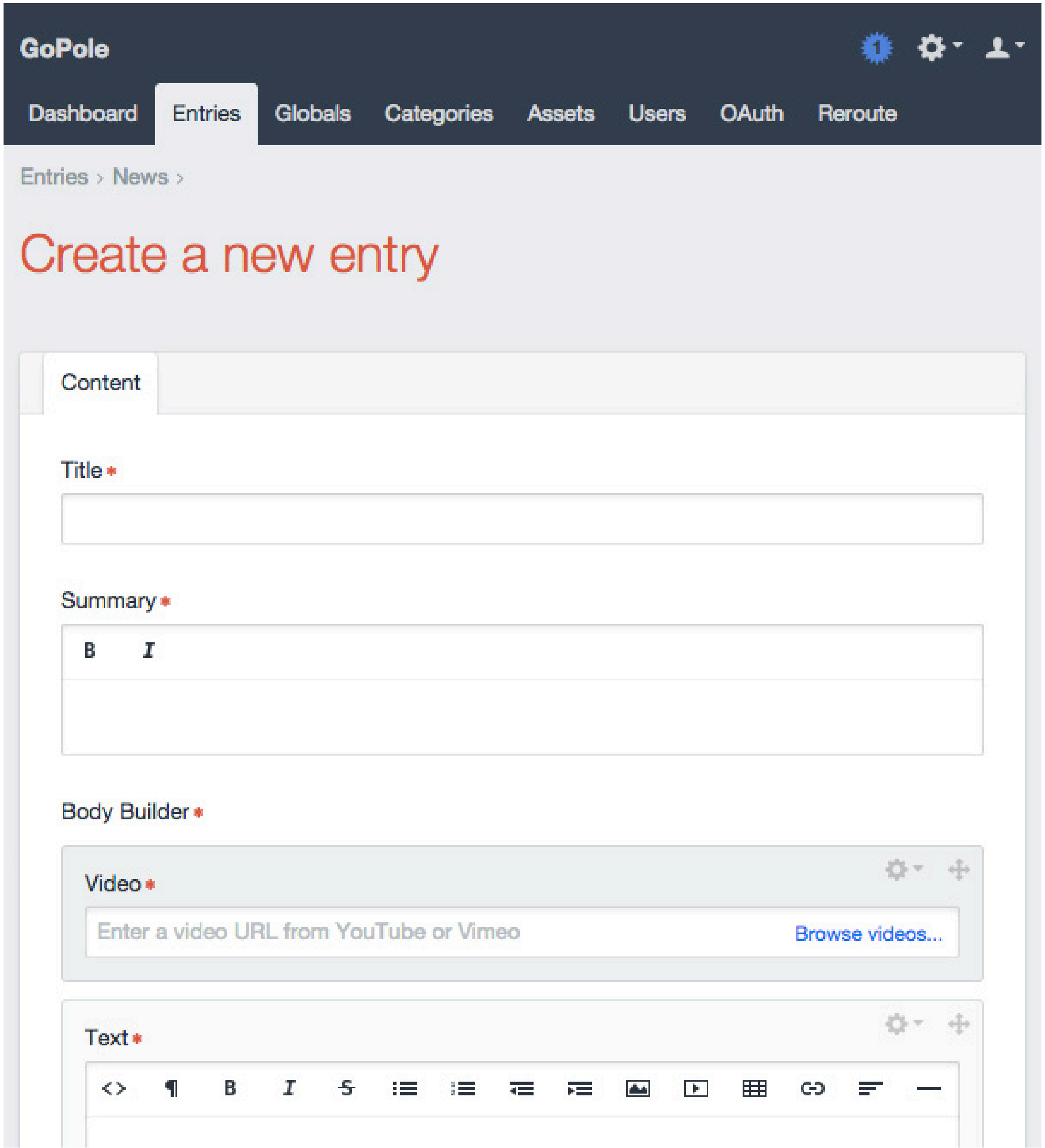
Title	URI	Post Date	Expiry Date
○ Base		6/26/2014	
● Venturecase		5/14/2014	
● Hi-Torque Thumbscrew Pack		5/14/2014	
● Lenspen		5/14/2014	
● Grenade Grip		5/14/2014	
● Bobber		5/12/2014	
● Evo		5/8/2014	
● Reach		5/8/2014	
● Scenelapse		3/27/2014	
● Arm		3/26/2014	



Title	Filename	Size	Date Modified
aaronchase-videopreview	aaronchase-videopreview.png	950.43 KB	5/14/2014
AJdirtbike	AJdirtbike.jpeg	160.89 KB	5/14/2014
Bobber	Bobber.jpeg	46.53 KB	5/14/2014
bobber-floating	bobber-floating.jpg	262.06 KB	8/21/2014
centurion-X-gopole	centurion-X-gopole.jpg	185.06 KB	8/7/2014
CookingwithGas	CookingwithGas.jpg	201.79 KB	5/14/2014
evo-zip	evo-zip.jpg	518.29 KB	5/14/2014
gadgetreview	gadgetreview.png	167.82 KB	6/19/2014
GirlonBeach	GirlonBeach.jpeg	79.48 KB	5/14/2014
GoPoleInstagram	GoPoleInstagram.png	428.52 KB	5/14/2014
GoProApp2.0	GoProApp2.0.jpg	223.21 KB	5/14/2014
GoProHero3	GoProHero3.png	549.58 KB	5/14/2014
Hero3onGrenadeGrip	Hero3onGrenadeGrip.jpg	399.28 KB	5/14/2014
lenspen cap off large	lenspen_cap_off__large.jpg	34.11 KB	5/14/2014
lenspen-cleaning housing large	lenspen-cleaning_housing__large.jpg	60.92 KB	5/14/2014
lenspen-cleaning- lens large	lenspen-cleaning-_lens__large.jpg	41.83 KB	5/14/2014
MtCreekMountainBike	MtCreekMountainBike.jpeg	112.51 KB	5/14/2014

Easily search, organize, and upload your content

Oh, and it’s responsive, so you can use it across all of your devices.



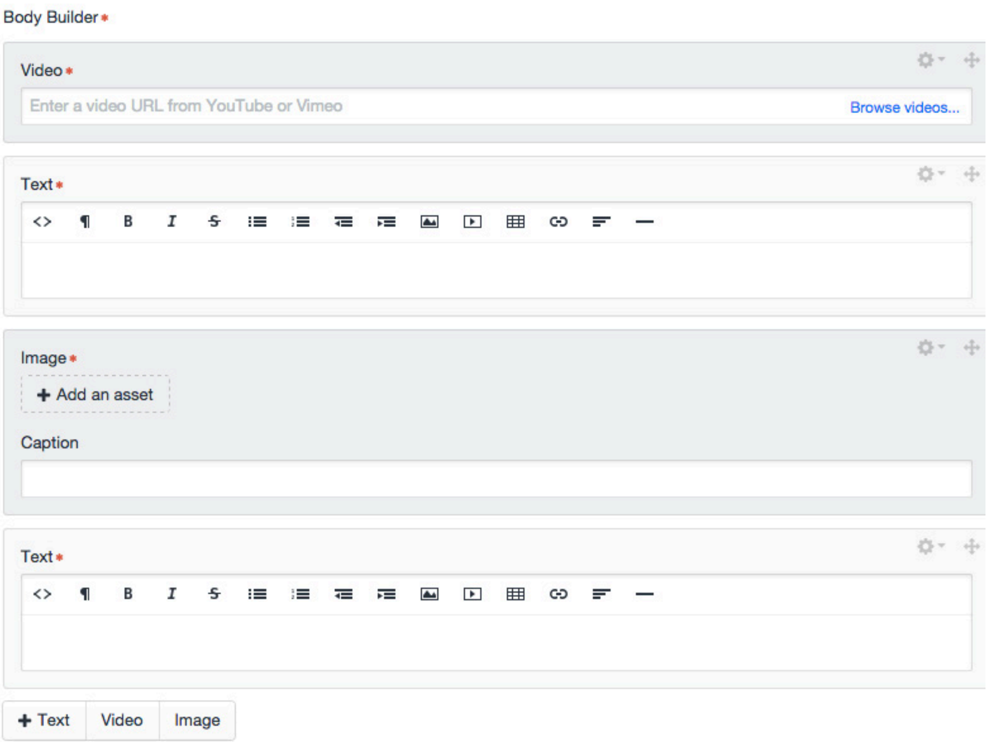
Live Preview

Craft handles previewing better than any other CMS I have used. When you configure your Sections in Craft, you select the template that acts as your entry template. Then, while you are creating an entry, you can click on Live Preview to bring up a split-screen view that lets you edit and view your changes as you type.

You can also generate shareable URLs so others can view the entry as it would appear on the page before publishing it.

Matrix

Pixel & Tonic built the Matrix add-on for ExpressionEngine way back in February 2010, and it was a game changer. Matrix for Craft is even better than you could imagine. When you configure the field, you define blocks that contain custom fields. Then, while you are creating an entry, you can add as many blocks as you want in whatever order you want.



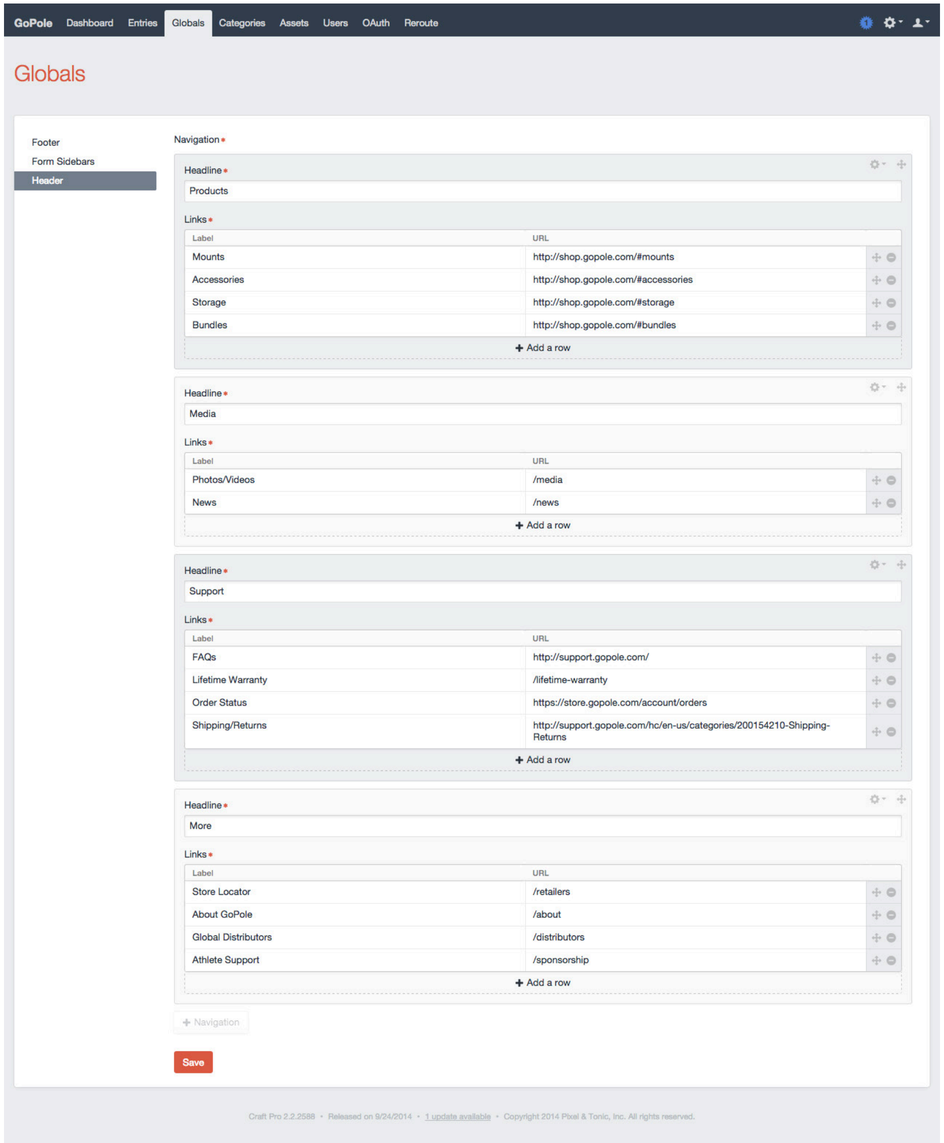
The organized nature of Craft allows us to easily find and modify content as well as work in real-time with the live preview feature. It also has empowered our team to use our creativity to build new pages and experiment with different page layouts.

— Alison Pilsner, Craig Hospital

Globals

For those little pieces of content that don’t deserve a full entry, Craft provides something called Globals. You can control the grouping and fields that are displayed within each Global.

In this example, we used Globals to give our client control over the entire navigation. This means that they can add, remove, and reorder links however they needed, without needing our assistance.



Globals

Navigation *

Headline * Products

Links *

Label	URL
Mounts	http://shop.gopole.com/#mounts
Accessories	http://shop.gopole.com/#accessories
Storage	http://shop.gopole.com/#storage
Bundles	http://shop.gopole.com/#bundles

Headline * Media

Links *

Label	URL
Photos/Videos	/media
News	/news

Headline * Support

Links *

Label	URL
FAQs	http://support.gopole.com/
Lifetime Warranty	/lifetime-warranty
Order Status	https://store.gopole.com/account/orders
Shipping/Returns	http://support.gopole.com/hc/en-us/categories/200154210-Shipping-Returns

Headline * More

Links *

Label	URL
Store Locator	/retailers
About GoPole	/about
Global Distributors	/distributors
Athlete Support	/sponsorship

Craft Pro 2.2.2588 • Released on 9/24/2014 • 1 update available • Copyright 2014 Pixel & Tonic, Inc. All rights reserved.

Shared Custom Fields

Though it may seem small, it's really nice that you can share custom fields across everything in Craft. You can use the same fields that you use in Sections in Globals, Users, Assets, Tags, and Categories. Craft also comes with 18 different custom field types right out of the box.

Core is so Powerful

Compared to many off-the-shelf CMSs that I have used, you actually don't need a ton of plugins for Craft. That's not to say there isn't a good collection of third-party plugins that exist—you just don't need them for every site you build. Here is a quick list of features that come native to Craft that sometimes require plugins with other CMSs:

- Versioning and drafts
- Localization
- Live preview
- Relationships
- Image resizing
- Using a CDN for assets
- Categories
- Tags
- User management

Ease of Updating Craft

Another benefit of less reliance on third-party plugins is that it makes updating the site even easier. It also doesn't hurt that Craft requires only one click to update the core.

As a developer, it also excites me that once you update the core files, you just have to visit the control panel to run migrations. This makes updating the core a breeze across multiple environments.

Support

In addition to having community support and a Stack Exchange site, there is an actual company behind Craft, Pixel & Tonic. That means that you can talk directly to the people who you paid for the product. Have a bug? Use the form on the dashboard of the control panel to send them a message along with error logs, database backup, and template files. Have a random question? Tweet it out and they will respond to you quickly (along with other Craft community members).

Twig

It’s not really something that clients care about, but developers should definitely care about the templating engine. Craft uses Twig. It’s really clean, simple to use, and it’s not proprietary. Find something that Twig doesn’t do out of the box? It’s easy to extend. My favorite part of Twig is that it is 100% bring your own HTML. You don’t have to work around HTML that is automatically generated—you are in control of it all.

Want to retrieve a list of blog posts?

```
{% set entries = craft.entries({ section: ‘blog’ }) %}

{% for entry in entries %}
    <div class=”post”>
        <h2><a href=”{{ entry.url }}”>{{ entry.title }}</a></h2>

        {{ entry.summary }}
    </div>
{% endfor %}
```

Twig syntax is very familiar, elegant, and easy to read. It also doesn’t hurt that, if you do have a typo, you get an error message that tells you exactly which line has the error.

So What?

That list doesn't convince you to give it a try? Well, you can try it for free, so why not?! The Craft community may be new and small, but it is thriving — here at Viget several of the Front-End Developers (myself included) are active members of the community and have contributed plugins as we've encountered the need. I have total confidence that Pixel & Tonic will continue to push Craft in a positive direction, and you will begin to see other agencies adopt Craft as their preferred CMS. We're at the early stages of major clients requesting Craft and expect to see many notable sites make the switch in the years to come.

A quick note: up to this point, we've discussed mostly OTS solutions in this section. These packages do the trick for a lot of organizations, especially smaller ones, but sometimes an organization needs something completely unique, something tailored to its own workflow and content types. This is where a custom-built CMS becomes practical. To develop a custom CMS, we use Rails.

Why Clients Choose a Rails CMS

Kevin Powers

Senior Digital Strategist

When it comes to selecting a CMS, we know one size doesn't always fit all. There's more to content management than just writing, tagging, and posting. You can't forget about the not-so-obvious pieces such as licensing fees, backend technology, and user permissions.

In section 3 of this ebook, we touched a little bit on custom CMSs and how they can be a better solution depending on the the needs of the business.

While we are continually experimenting with the newest technology, we have found Rails to be the best framework for building custom CMSs.

Viget is not the first: Rails is a popular choice for this type of work. Viget has standardized its tools and processes over the years and has found it flexible and robust for a broad range of project sizes, including World Wildlife Fund, National Trust of Historic Preservation, Volunteers of America, Shure, and PUMA.

4 Reasons Why Rails Is Our Top Pick for Custom CMSs

1. Rails is Open Source, Widely Adopted

Ruby on Rails is an open source web application framework that has been widely used since its initial release more than a decade ago. Major sites like Airbnb, Kickstarter, and Bloomberg News are built on Rails. In building a CMS, we pull heavily from a strong community of Rails developers to employ existing modules and functionality (e.g., Active Admin) so you're not spending time and money reinventing the wheel. In turn, we give back to the community by open-sourcing our own code.

- No software licensing fees
- Proven technology
- Available pre-existing modules and functionality
- Active community (and vendors) for support

2. Rails is Fast and Flexible

A Rails CMS respects your data. Meaning, your content lives in a well-defined relational database making access to and manipulation of that material efficient and clear. When building the CMS, we're able to purposefully architect the database and application components with an eye toward performance and speed, both for your admins and site visitors.

- Purposeful caching and optimization of assets (e.g., images, CSS)
- Relational data enables advanced search capabilities for users
- Structured data can be exposed via APIs and reused in other experiences (e.g., mobile/iOS apps)

3. Rails is Built for You

While pre-built and off-the-shelf content management systems may come with a deep catalog of features and configurations, these options can often become a burden and obstacle to publishing your content in an efficient. You spend increased time learning the rules of the system and need to navigate around options that don't apply to your work. The administrative experience of our Rails CMS is tailored to the way you work and insights from our user experience design team, and includes key features that you will actually use.

- Intuitive, tailored admin makes learning the system and content publishing easy
- No time spent hiding or working around features of existing systems that aren't relevant or useful

4. Rails is Future-Minded

Content management doesn't mean the same for every organization. Often unique needs arise that are hard to satisfy by bolting features onto an existing system or changing the way that system works. A Rails CMS opens the door to endless customizations, including:

- Personalization
- Multi-site management
- In-admin analytics
- Custom workflow
- Third-party integrations
- API to power content in an iPad app

SECTION 5

Miscellaneous and special cases

Up to this point, you've read about the different approaches and types of CMSs. But regardless of the platform you choose, it's critical to be aware of security threats, e-commerce options, and the editorial perspective. In this section, we'd like to share our thoughts and experience in these three important areas.

Security: Your CMS is Probably Vulnerable to Privilege Escalation Attacks

Lawson Kurtz

Senior Developer

We recently discovered an easily exploitable, vertical privilege escalation vulnerability in every popular, off-the-shelf CMS that we tested.

The lesson: CMSs either need stronger security around user permission updates, or to backtrack away from the convenience afforded by allowing raw HTML editing and publication from non-admin users.

What is a privilege escalation attack?

A privilege escalation attack is the process of exploiting a bug, insecurity, or poor configuration to increase your level of access within a system. Through such an attack, a user who already has a limited degree of access to a CMS can assign themselves the unrestricted access.

Which CMSs are affected?

Just about every CMS that provides unfiltered HTML editing capabilities to non-admin users is vulnerable to XSS-based vertical privilege escalation attacks, even when CSRF protection is in place. We personally validated the attack against Craft, WordPress, and Drupal*.

The vulnerability was responsibly disclosed with exploit samples to each security team in April 2016. The Craft security team immediately responded by releasing a fix for the exploit as a critical update, and is no longer vulnerable in recent versions.

**Note that in its default configuration, Drupal is not vulnerable to this attack as the administrator user role is the only role able to author unfiltered HTML, however nearly all organizations we've seen use a more complex system of user roles and permissions which increase their susceptibility to this attack.*

What is the outcome of the attack?

Non-admin CMS users can update their user account's role to admin status. This typically provides the attacker with a large number of new exploitable vectors. (e.g., download a database backup to crack contained hashed passwords, deploy a broader XSS attack within a CMS theme template file, deface the public website, etc.)

Technical Attack Summary

The attack is a straightforward XSS exploit.

I: The Exploit Script

The exploit revolves around exploiting the trust that the CMS has in the browser of an admin user. An attacker needs only an elementary understanding of JavaScript and HTML to create a script that silently issues a request to the CMS to update their user account with increased permissions.

It is a common misconception that CSRF-protection mechanisms are useful against these sorts of XSS attacks, however CSRF-protection merely adds one additional, trivial step to the attack process.

II: Exploit Deployment

With the exploit script in hand, an attacker can drop the script anywhere in the CMS where unescaped, unfiltered HTML is displayed. (This is almost everywhere in most popular CMS configurations.)

Some doubt that attackers would be so bold as to place obviously malicious code directly into their own blog posts or page updates. Keen attackers, however, will obfuscate the intention of the exploit script before deployment to avoid detection, perhaps through minification and encoding.

```
<script>

eval(window.atob(‘RGlkHlvdSByZWFsbHk
gdGhpbmsgS SdkIGp1c3QgZ2l2ZSB5b3UgYW4gYX
R0YWNrIHNjcmlwdD8gVGhlIGxhdyBpcyBu
b3QgcGVyZmVjdGx5IGNsZWFyIG9uIHRoaXMgbW
F0dGVyLCBidXQgaXQgY2FuIGJlIGFyZ3VlZCB0
aGF0IH B1Ymxpc2hpbmcgYXR0YWNrcyBpcy
BhIGNyaW1pbmFsIGFjdGl2aXR5LiBJJ20
gbm90IGEgZmFuIG9mIGdvdmVybm1lbnQg
ZmluZXMgb3IgamFpbCwgc28gSSdtIHB1Y
mxpc2hpbmcgdGhpcyBzaWxseSBtZXNzYW
dlIGluc3RlYWQuIEFuZCBpZiB5b3UgYWN0dWFs
bHkgdG9vayB0aGUgd GltZSB0byBkZWNvZGUgd
Ghpcywgd2UnZCBwcm9iYWJseSBlbmpveSB3b3Jr
aW5nIHdp dGggeW91LiBIaXQgdXMgdXAgYXQgam9i
c0B2aWdldC5jb20gYW5kIHRlbGwgdGhlbSBMYXd
zb24gc2Vu dCB5b3UuIENoZWVycy4=’));

</script>
```

Still too obvious? how about...

```
<script>
// Omniture Tracking Snippet
var omTrack =

eval(window.atob('RGlklJlvdSByZWFsbHkgdGhpbmsgSSdklGp1c3QgZ2l2ZSB5b3UgYW4gYX

R0YWNrIHNjcmlwdD8gVGhlIGxhdyBpcyBub3QgcGVyZmVjdGx5IGNsZWFyIG9uIHRoaXMgbW

F0dGVyLCBidXQgaXQgY2FuIGJlIGFyZ3VlZCB0aGF0IHB1Ymxpc2hpbmcgYXR0YWNrcyBpcy

BhIGNyaW1pbmFsIGFjdGl2aXR5LiBJJ20gbm90IGEgZmFuIG9mIGdvdmVybm1lbnQgZmluZX

Mgb3IgamFpbCwgc28gSSdtIHB1Ymxpc2hpbmcgdGhpcyBzaWxseSBtZXNzYWdlIGluc3RlYWQ

uIEFuZCBpZiB5b3UgYWN0dWFsbHkgdG9vayB0aGUgdGltZSB0byBkZWNvZGUgdGhpcywgd

2UnZCBwcm9iYWJseSBlbmpveSB3b3JraW5nIHdpdGggeW91LiBIaXQgdXMgdXAgYXQgam9i

c0B2aWdldC5jb20gYW5kIHRlbGwgdGhlbSBMYXdzb24gc2VudCB5b3UuIENoZWVycy4=')); 

omTrack.linkTrackVars="eVar45,events";

omTrack.linkTrackEvents="event23";

omTrack.eVar45="MaturingEquityLine";
</script>

<!-- That's right. Nobody at your company actually knows how Omniture works. -->
```

III: Phising Around

After the exploit has been deployed, the attacker now needs only to guide an admin to the deployed exploit. Unfortunately, common CMS workflow practices make this insanely easy.

Once the target hits the deployment, the script executes in the context of the admin's session. The CMS trusts the target's session, so it fulfills the permission escalation request without hesitation.

The attacker is now a system admin and the target is none the wiser.

Defending Against The Attack

Being attacked sucks. Fortunately there are steps each of us can take to protect ourselves and our teammates from this sort of attack.

For Everyone

- Use strong and unique passwords. CMS privilege escalation can often provide access to database backups which contain encrypted user passwords. It is much more difficult for an attacker to decrypt your password if you have a long password (that isn't just an obvious word with some numbers tagged on to the end). If your password is decrypted, and you reuse the same password elsewhere, this limited scope attack can become considerably more damaging.

- Be skeptical. Don't even click on links sent to you from people you don't trust (especially not links found in sketchy emails). Contact your security team if anything ever feels weird or fishy.

For Site Developers/Administors

- Keep your CMS updated to the latest version. Releases often contain security fixes that may patch vulnerabilities such as the one described above. These releases can often actually show attackers where vulnerabilities existed in older versions, so your site is extra vulnerable if you choose not to update.

- Consider applying a content security policy to your site that prohibits execution of untrusted scripts to remove the XSS vector that CMS WYSIWYG editors typically provide. (Unfortunately these types of CSPs are extremely onerous. Get ready to do some serious lifting.)

- Don't provide CMS access (particularly access to post raw HTML) to anybody that you would not trust as a CMS admin. Have others draft content in a Google Doc, and have a trusted user copy it over.

- Stay logged out of sites unless you need to be logged in to complete your current task. (Yep, actually use that log out button.)

For CMS Developers/Maintainers

- Require reauthentication for all requests attempting to modify user permissions. (This is the approach with which Craft has mitigated this exploit.)

- Limit the scope of CMS actions that can be performed from a web portal (e.g. not providing an online file editor… looking at you WordPress).

- Store database backups securely, and keep them inaccessible from the CMS's web interface.

Choosing a CMS and E-Commerce Combination

Heather Muety

Senior Project Manager

Rich, creative content is an increasingly important part of online marketing strategies. As inbound-marketing giant Hubspot notes:

> "Big brands realize they need so much more than a picture and description to sell products. They're all about selling a lifestyle, and that means bigger, better content is needed to engage consumers."

Unfortunately, the two most prominent e-commerce platforms — Magento and Shopify — have limited capabilities for creating content. So what should savvy brands do if they want both a powerful e-commerce solution and a feature-rich content management system?

Beautiful, Responsive Control Panel

The Craft control panel is beautifully simple. The design gets out of the way and makes it easy to edit and manage your content.

	Craft Commerce	**ShopifyPlus**	**Magento Community**	**Magento Enterprise**
Cost	$999/ one-time fee + hosting + PCI compliance	$995 per month + extensions	Free + hosting + PCI compliance + extensions	Starts at $18,000 per year + hosting + PCI compliance + extensions
Build	Developer needed	Developer recommended	Developer needed	Developer needed
Manage	No support	Dedicated 24/7 live support	No support	24/7 support
Host	Client must purchase and maintain hosting, SSL certificate, and PCI compliance	ShopifyPlus fully hosts your site and provides Certified Level 1 PCI DSS compliance free SSL certificate, and two-factor authentication	Client must purchase and maintain hosting, SSL certificate, and PCI compliance	Client must purchase and maintain hosting, SSL certificate, and PCI compliance

	Craft Commerce	ShopifyPlus	Magento Community	Magento Enterprise
Features	Fully integrated CMS and e-commerce; powerful content marketing capabilities; live preview; endless upsell; cross-sell potential; puts you in control of everything	Customer tags; blog; additional sales channels such as Shopify Buy Buttons; Facebook, Pinterest & Twitter; over 70+ payment gateways; built-in integration with fulfillment vendors Amazon, Shipwire and Rakuten	Multiple stores within one admin backend; Google shopping available; credit card info can be saved on file; send wishlist via email; hundreds of extensions available (free + paid)	Community features + gift registry; private sales; loyalty rewards; wishlists; customer-initiated returns; hundreds of extensions available (free + paid)
Deal Breaker?	New this year so the extensions (plugins) available are limited; client must manage hosting and PCI compliance	Limited CMS functionality makes customized content pages difficult; credit cards can not be saved on file (sometimes deal breaker for subscription payments); client must host site with Shopify	Limited payment gateways without extension purchase; extension required for any fulfillment integration; dynamically generated URLs not SEO-friendly and extension required to update; you may need to purchase a lot of extensions to get the features you need	Same as Magento Community

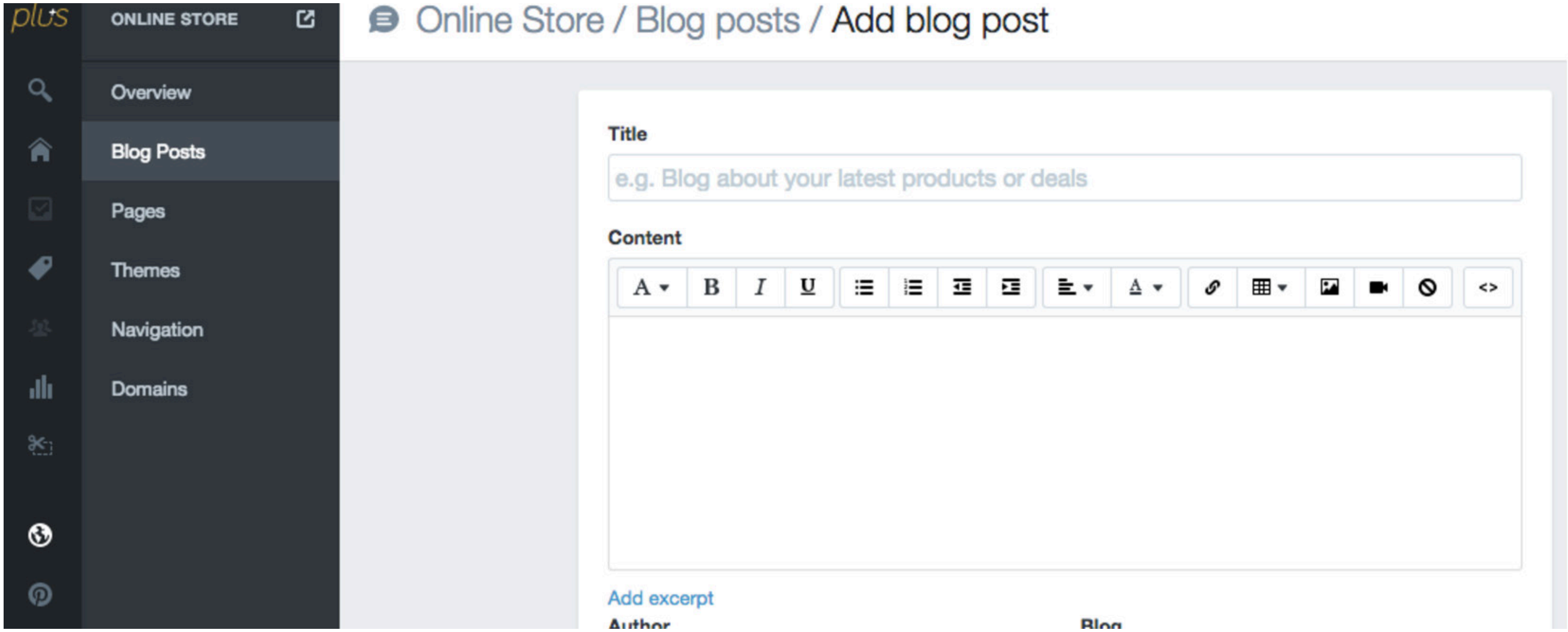
Your specific needs, priorities, and budget will ultimately inform your decision. But generally, if you already use Magento Community/Enterprise or ShopifyPlus, Craft is a great choice, and WordPress also integrates well. If you don't already use (or can move away from) Magento or Shopify, we recommend considering Craft and its Craft Commerce platform.

Here's a look at those CMS + e-commerce approaches. management system?

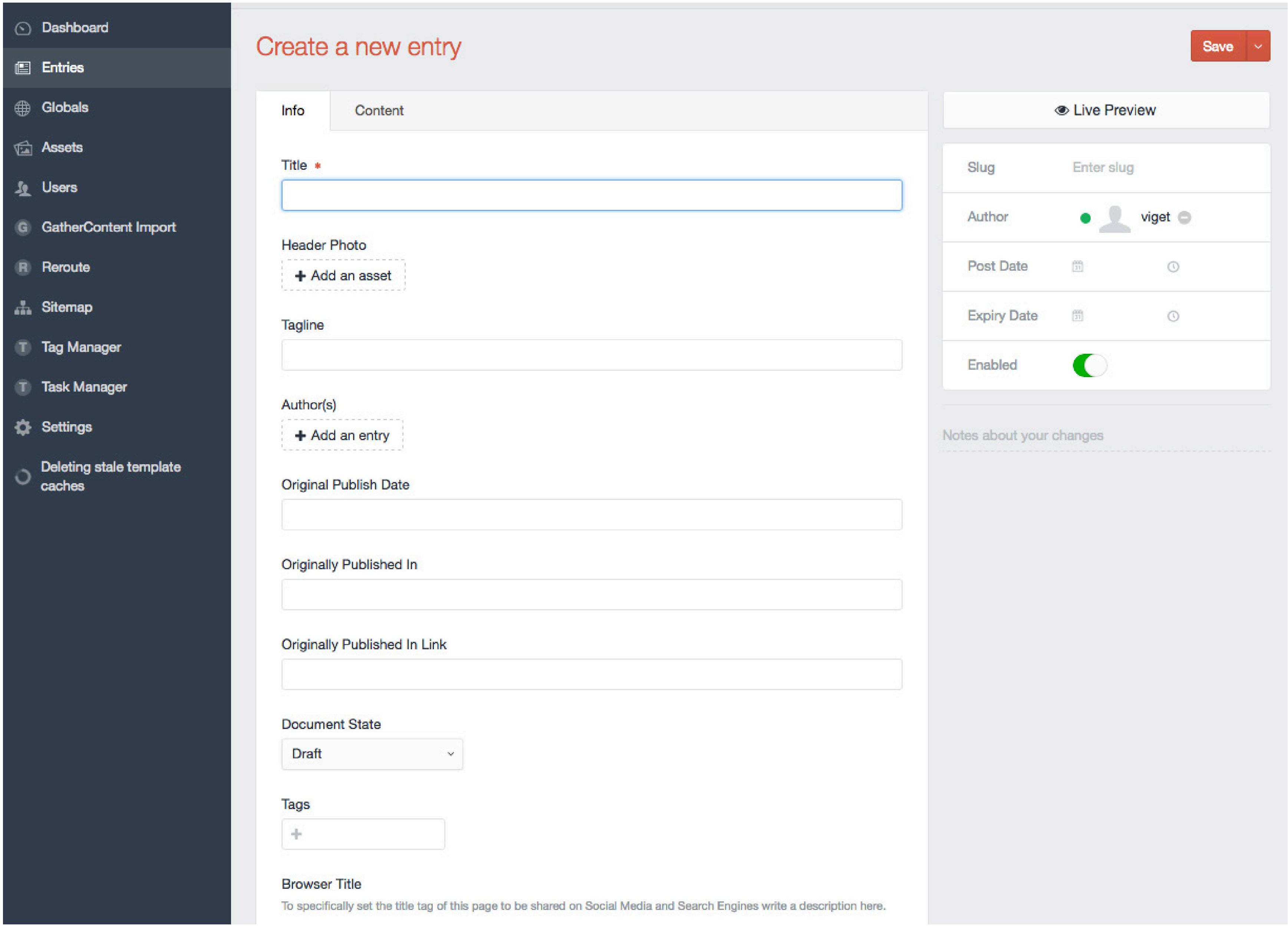
Craft + Magento or Shopify

Magento and Shopify focus on e-commerce features such as product variances, pricing and shipping models, tax rules, return management, and security. Both list CMS capabilities, but neither offers intuitive or easy ways to build and manage the large amounts of rich and engaging content that many brands want to create.

For example, Shopify lets you create product pages, a one-off page (think legal policy), and a blog page. The blog and page templates only offer a title and a text editor — that's all, folks:



In contrast, Craft offers unlimited content types (e.g., blog posts, articles, pages, press releases, people, events) and unlimited, re-orderable content blocks (e.g., text, photo, photo carousel, quote):



Create a new entry

Title *

Header Photo

+ Add an asset

Tagline

Author(s)

+ Add an entry

Original Publish Date

Originally Published In

Originally Published In Link

Document State

Draft

Tags

Browser Title

To specifically set the title tag of this page to be shared on Social Media and Search Engines write a description here.

Slug: Enter slug

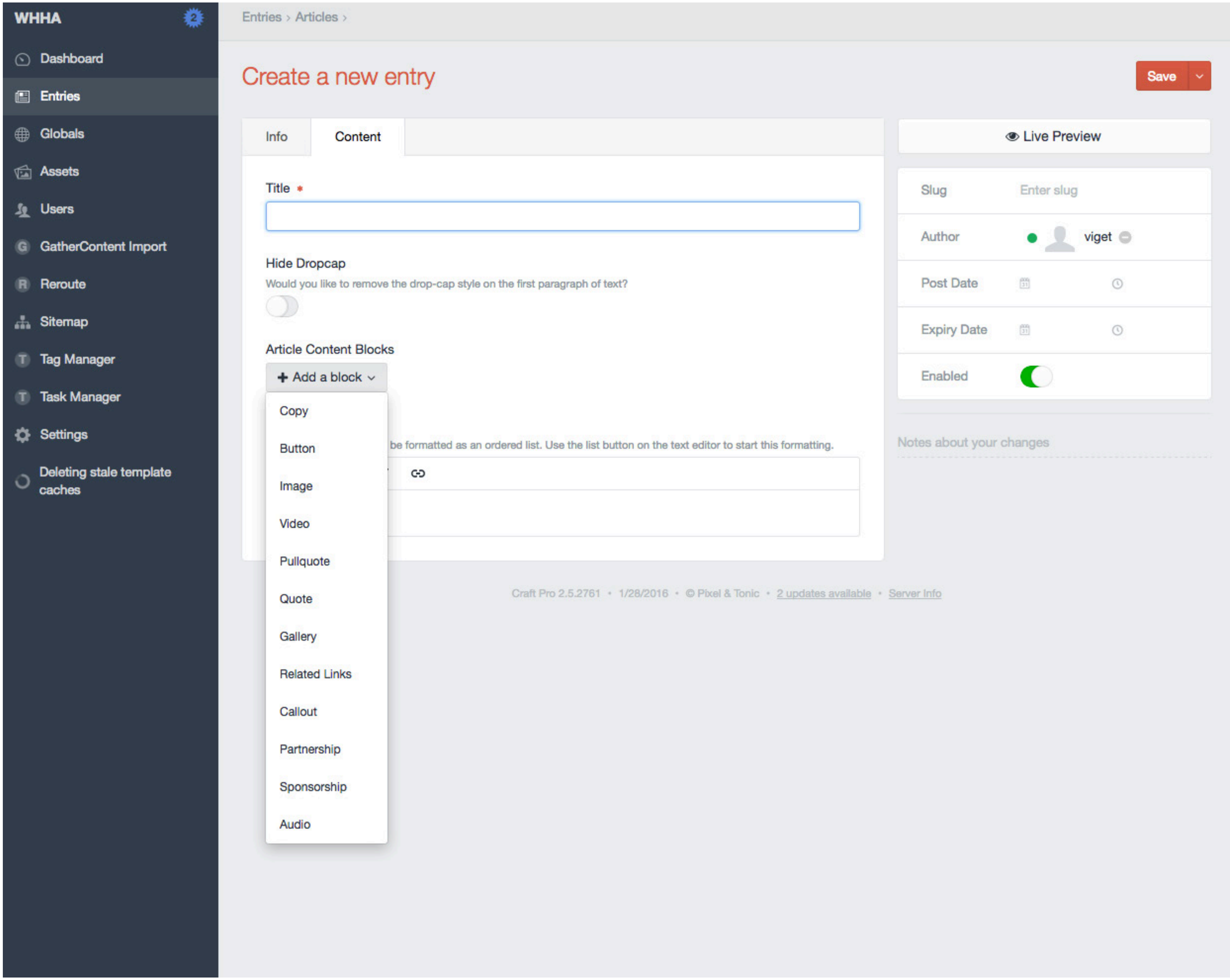
Author: viget

Post Date

Expiry Date

Enabled

Notes about your changes



Would you like to remove the drop-cap style on the first paragraph of text?

be formatted as an ordered list. Use the list button on the text editor to start this formatting.

Craft Pro 2.5.2761 • 1/28/2016 • © Pixel & Tonic • 2 updates available • Server Info

Other Craft highlights include:

- Easy-to-create content relationships
- Drafts/versioning
- Live preview
- Categories and tags
- Automated image resizing

In short, Craft lets you create a fully customized marketing site that can tell your story and display rich content exactly the way you imagined it.

Things to consider:

- Your e-commerce pages will be on a separate domain from your content marketing pages.
- You'll need to log into two admin sites: one to manage e-commerce data and one to manage marketing content.

For Craft + Magento:

- You will be hosting two sites.
- You'll need to develop a plugin to display Magento product data on your Craft marketing site

For Craft + ShopifyPlus:

- You will be hosting one site and paying Shopify to host your other site.
- Your Shopify product data can be displayed on your Craft marketing site, if your developers use Viget's free Craft + Shopify plugin.
- Cross-sell is possible but will require development and manual upkeep.
- Checkout will be dictated by Magento/Shopify patterns, with limited customization.
- Magento and Shopify require multiple extensions (or plugins or apps).

Wordpress + Magento

The Magento WordPress Integration Plugin by FishPig integrates your WordPress site with your Magento Community or Magento Enterprise site.

Things to consider:

- Free.
- 1-click login to your WordPress admin from your Magento admin.
- No additional development needed to style WordPress pages in your Magento theme.
- Create relationships between your WordPress content pages and Magento product pages.
- Full translations in over 10 languages.
- 12 extensions for your extension (sigh) including:
  - Multi-site integration
  - Custom types and taxono mies
  - Advanced custom fields
  - Email subscriptions tied to MailChimp
  - Contact forms
  - Single sign-on to both WordPress and Magento customer accounts

Wordpress + Shopify

The Shopify WordPress Plugin lets you add your Shopify products (or collections) and Buy buttons to any page on your WordPress site from the WordPress admin. The Buy button connects directly with your Shopify secure shopping cart.

Things to consider:

- Free.
- Easy-to-generate embed code
- Available in English, French, and Spanish
- Customize button and cart colors
- Customize button target
(cart, checkout, Shopify product page)

CraftPro + Craft Commerce

If you aren't wedded to Magento or Shopify, Craft Pro + Craft Commerce offers an integrated feature-rich CMS and e-commerce platform. Unlike Shopify/Magento, you have control over basically everything — from content blocks and page design to the checkout flow.

Things to consider:

- $999 one-time-fee ($700 if you've already purchased Craft Pro). One fully integrated site: one domain, one admin interface for both content and product data, one hosting environment.
- Fully customized design; no rigid templates
- Endless ability to upsell, cross-sell
- Craft Commerce does not come with some out-of-the-box features you'll find with Shopify or Magento (e.g., gift registry, private sales, customer reviews, persistent shopping cart). However, these features can be developed.
- Newest to market, limited (but growing) availability of plugins.

If you need guidance on choosing a CMS, check out our comparison of off-the-shelf systems and our closer look at Craft vs. WordPress.

You can always give us a shout if you need some help deciding.

Editors: Beyond Content Management

Lawson Kurtz

Senior Developer

Nobody is excited by content. The word itself implies a sterility and blandness that is antithetical to our clients' work. And when you build systems to create and manage content, the resultant experience feels like, well, content. So we don't.

Each of our clients has a unique and compelling story to tell. Battles to save vanishing species. Initiatives to preserve our history. Products to change people's lives. Our job isn't merely to build apps that hold these organizations' content. It's to help them tell their stories in the digital medium. We build systems to create and manage stories.

But what's the difference?

Content is a flat and singular notion, so content management experiences almost always fall into one of two categories:

Contrived (aka Too Rigid)

In systems designed to manage content, stories are treated as a series of data points. A title. A picture. Body text.
That's it (deal with it).

Editors are forced to break down their stories into contrived pieces to fit them into a predefined set of fields. The editor loses control of the stories, and the stories lose their character.

Oftentimes, to make up for the constraints, the system will introduce an endless list of options to provide the illusion of flexibility. These efforts typically serve only to render the system bafflingly complex and unintuitive.

Unguided [aka Too Free]

In response to the shackles of contrived content management experiences, many instead favor what-you-see-is-what-you-get (WYSIWYG) tools for managing content. These tools give the editor more control, but they are complex, and difficult to use. Every possible decision is left to the editor. Layout. Colors. Font. Sizing. Everything.

So to a certain extent, the tool gets in the way of telling the story. And it's quite often difficult to achieve the consistency necessary to communicate a story in the context of a larger brand or initiative.

Beyond Content Management

Stories are more complicated than content. They are neither flat nor singular. They are structured, but unique. Free-flowing, but nuanced. Their components have relationships. They have context. They are not content themselves, but are instead composed of content. Story management demands composition.

Composition is completely neglected by most content management systems, yet is the key to effective story telling and experience building.

Building a Better Editor

So drawing from our vast experience building and using content management systems, we set off to build something fundamentally better. Something compositional.

The result, Colonel Kurtz, is an advanced compositional editor, that allows editors to intuitively build complex stories and interactive experiences.

Colonel Kurtz allows editors to compose stories from any number of content components (called blocks). And unlike existing block-style editors like Sir Trevor, Colonel Kurtz is truly compositional;

Blocks can nest within each other, allowing for the intuitive management of complex experiences (e.g. slide shows).

Different types of blocks can be added to Colonel Kurtz easily, and their display is custom-tailored to the specific property to which they belong. This removes the onus of visual design decisions as the editor creates their story. They compose their story as they wish, and it's guaranteed to look beautiful every time.

And the best parts: We've open-sourced Colonel Kurtz, so it's completely free for your business to use and extend. And you don't have to replace your existing CMS to use it. Colonel Kurtz is a tool that can be integrated with almost any existing CMS.

In the Wild

Colonel Kurtz has quickly become a staple in much of our current work. It already powers the content of beautiful properties like http://wcs.org/ and https://savingplaces.org/, and it's getting better all the time.

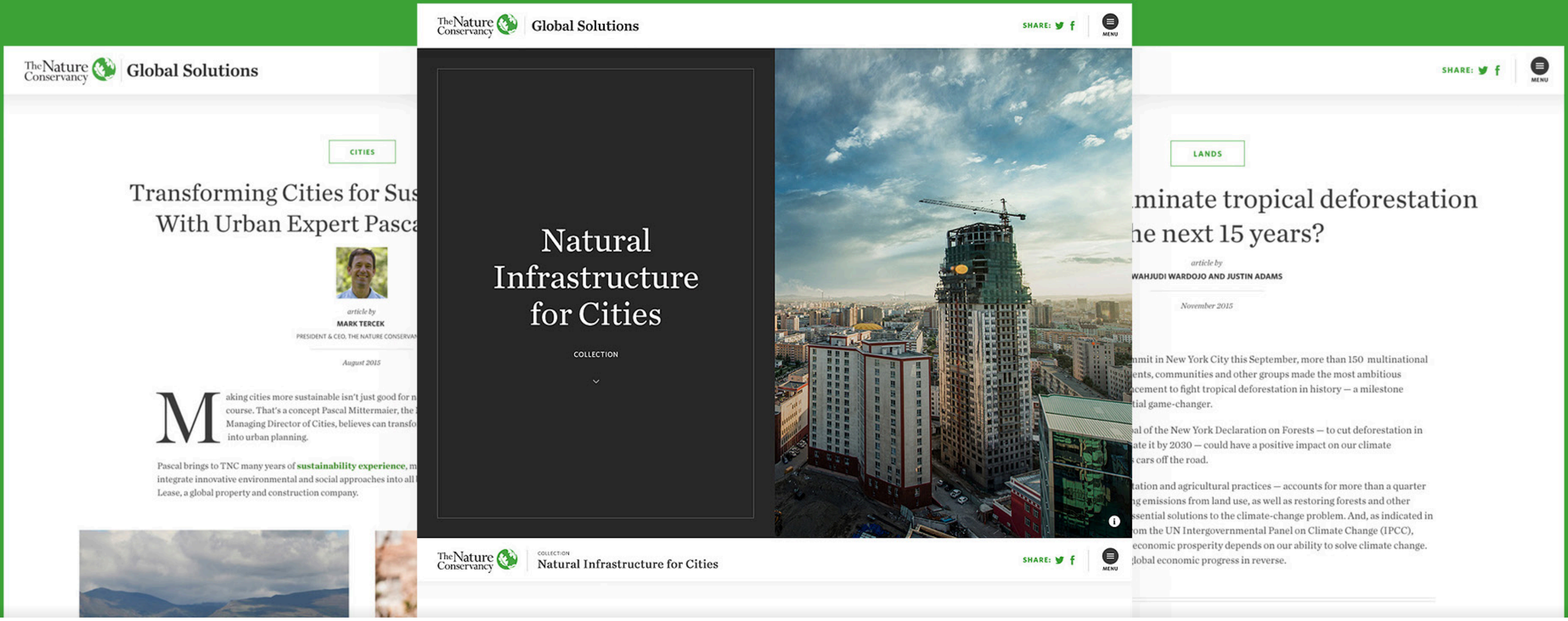
We're extremely excited about Colonel Kurtz, and have some really big plans for its future. So give it a try: we're confident you'll love it.

If you'd like to learn more about how you can use Colonel Kurtz, get in touch or head over to the project's home on GitHub.

SECTION 6

See it in action: Case Studies

To illustrate our thoughts in sections 1-5 of this ebook, we have included seven case studies. We chose each of these case studies to highlight the benefits of various CMS solutions. You'll notice the wide range of client needs and how we tailored the CMS to accomplish their objectives. You'll find examples of custom and off-the-shelf CMSs, as well as CMSs built for specific functionality such as photo management and ecommerce.



CITIES

Transforming Cities for Sus
With Urban Expert Pasca

article by
MARK TERCEK
PRESIDENT & CEO, THE NATURE CONSERVAN

August 2015

Making cities more sustainable isn't just good for n
course. That's a concept Pascal Mittermaier, the
Managing Director of Cities, believes can transfo
into urban planning.

Pascal brings to TNC many years of **sustainability experience**, m
integrate innovative environmental and social approaches into all
Lease, a global property and construction company.

Natural Infrastructure for Cities

COLLECTION

COLLECTION
Natural Infrastructure for Cities

LANDS

minate tropical deforestation
he next 15 years?

article by
WAHJUDI WARDOJO AND JUSTIN ADAMS

November 2015

mmit in New York City this September, more than 150 multinational
ents, communities and other groups made the most ambitious
cement to fight tropical deforestation in history — a milestone
tial game-changer.

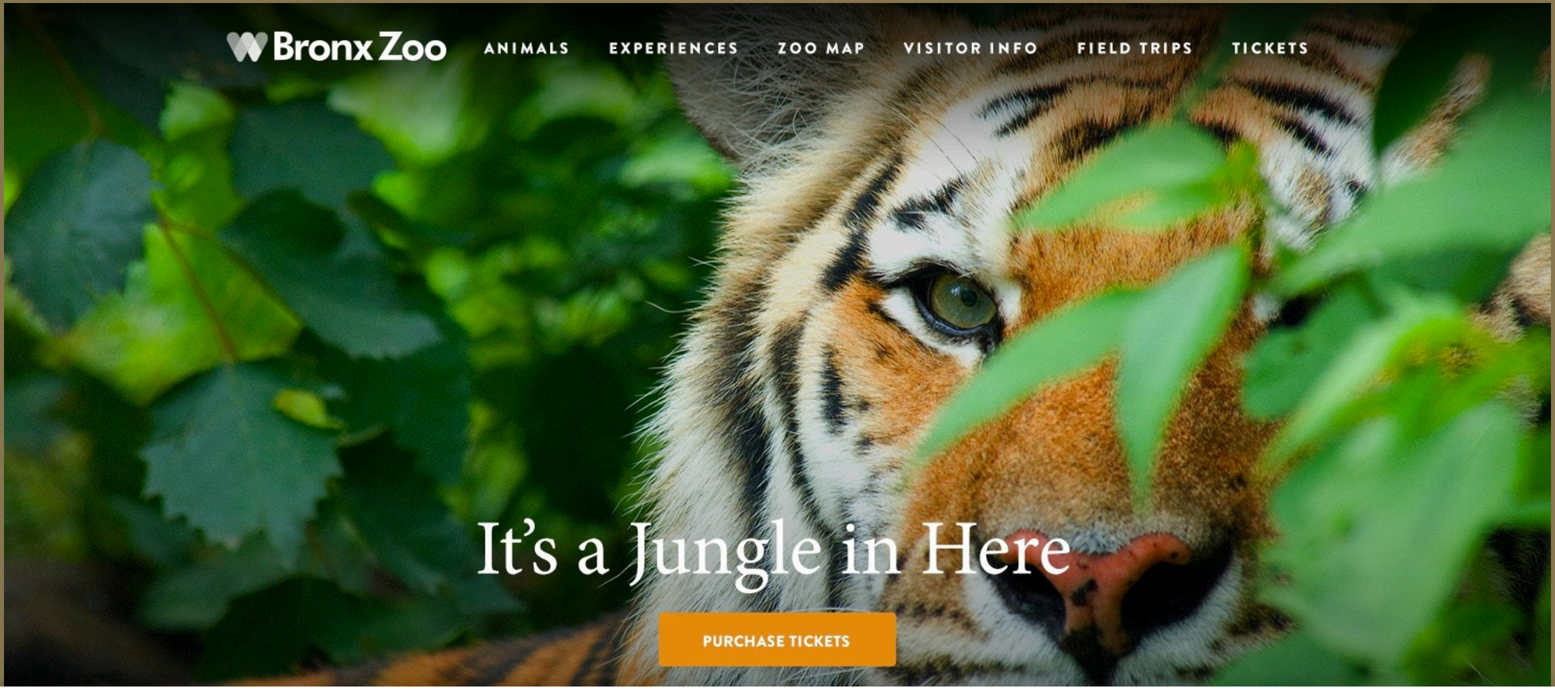
al of the New York Declaration on Forests — to cut deforestation in
ate it by 2030 — could have a positive impact on our climate
cars off the road.

tation and agricultural practices — accounts for more than a quarter
g emissions from land use, as well as restoring forests and other
ssential solutions to the climate-change problem. And, as indicated in
om the UN Intergovernmental Panel on Climate Change (IPCC),
economic prosperity depends on our ability to solve climate change.
lobal economic progress in reverse.

1. Flexibility: The Nature Conservancy

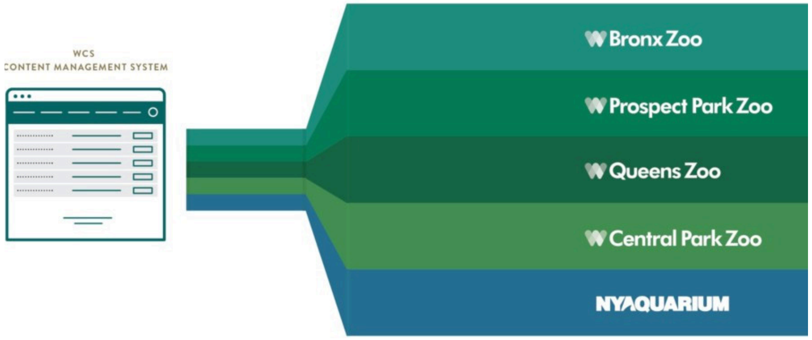
The Nature Conservancy (TNC), a global nonprofit that protects animals and preserves land, needed to create new content and aggregate content from elsewhere on the Internet. Knowing this, we designed and built a flexible CMS written in Ruby on Rails. Customized to solve their unique challenges, such as spinning up curated collections, we gave TNC ultimate flexibility in creating and managing content by using our own editor which allows for easy addition of photos, videos, styled text, file attachments, buttons, and more.

View the case study online

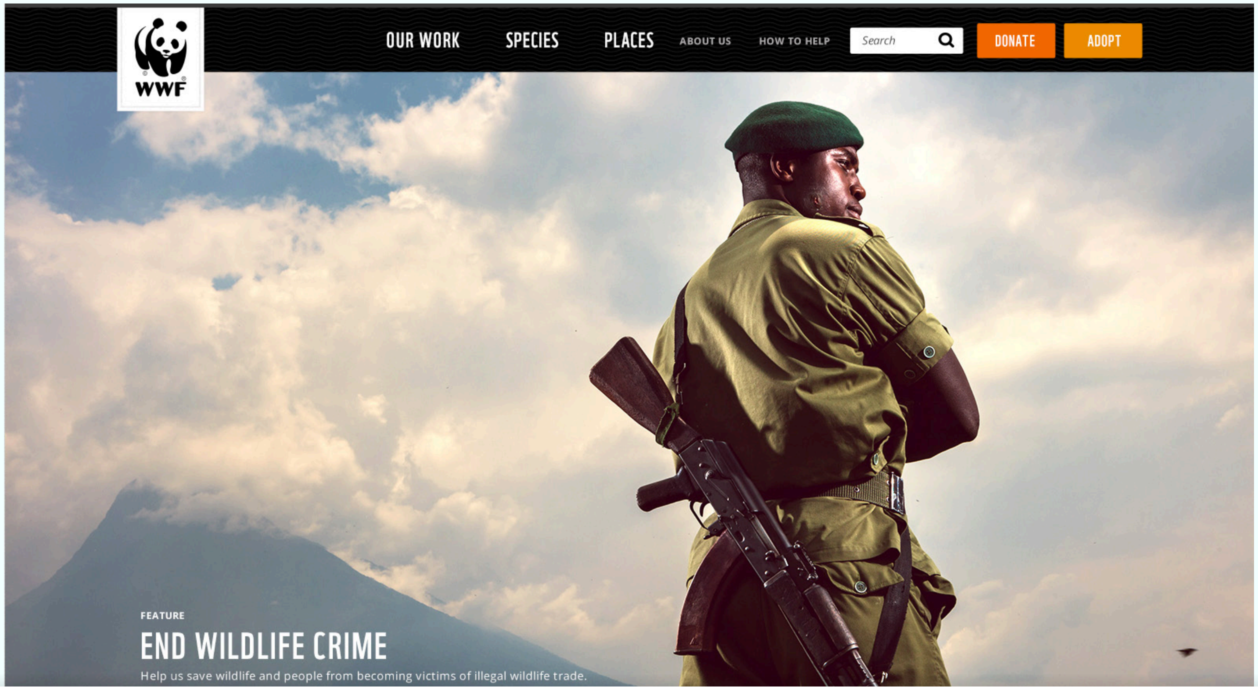


2: Multi-Site — Wildlife Conservation Society

The Wildlife Conservation Society, an international organization that protects wildlife, needed a central content management system to not only manage their five facility sites Bronx Zoo, Prospect Park Zoo, Queens , Zoo, Central Park Zoo, and the NY Aquarium), but also plan for future sites. Viget built a Ruby on Rails-based CMS platform that gives site administrators the ability to share content and assets between all five facility sites as well as the ability to spin up new sites easily.



View the case study online



FEATURE

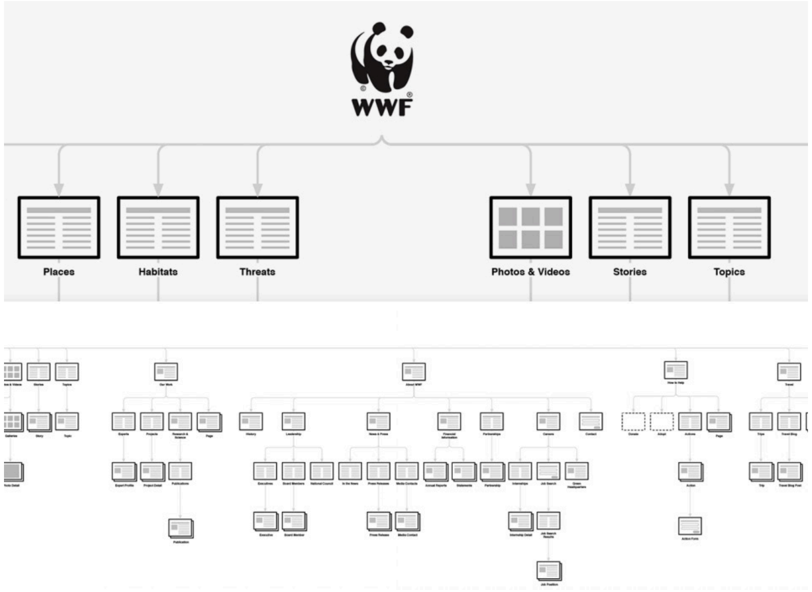
END WILDLIFE CRIME

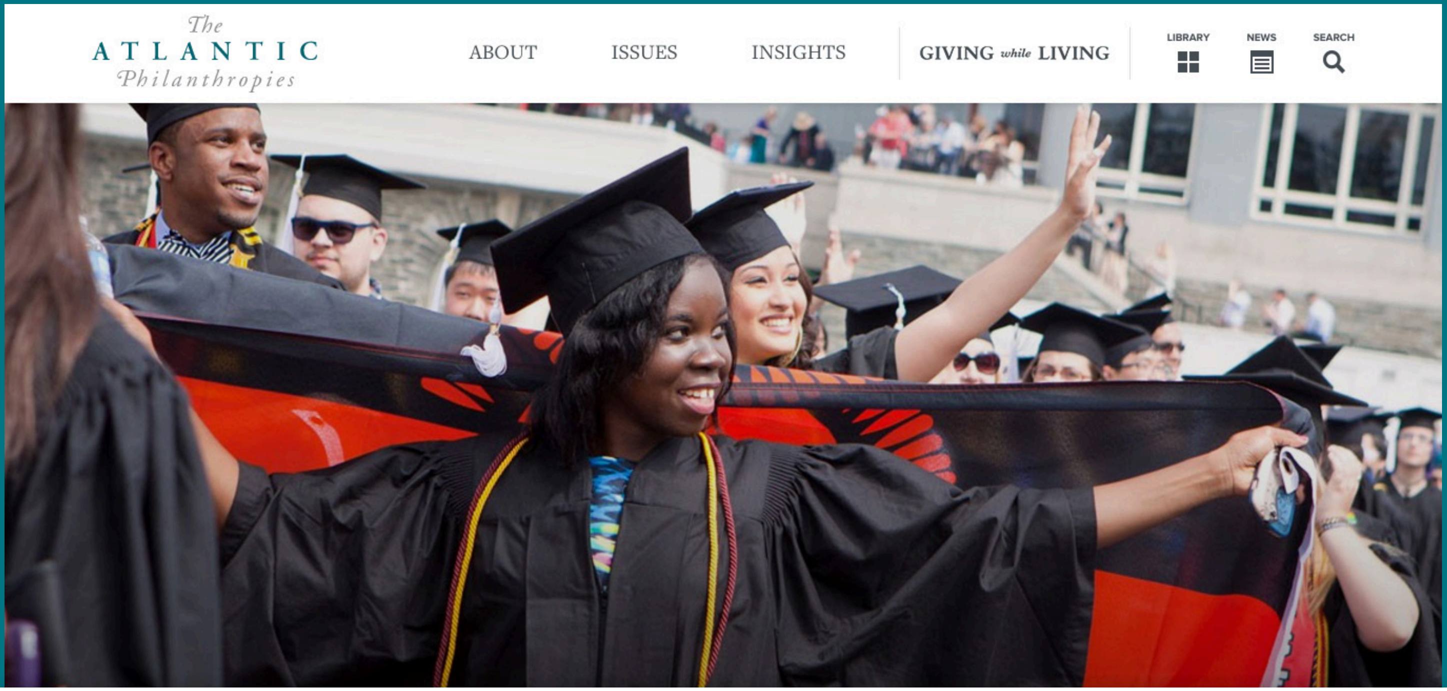
Help us save wildlife and people from becoming victims of illegal wildlife trade.

3: Photo management — World Wildlife Fund

The World Wildlife Fund (WWF) is the largest independent conservation organization on the planet. Photography was instrumental to WWF's storytelling, so we built a variety of photo management tools into the CMS, including a feature that automatically scales and crops an uploaded image into various sizes. While our primary goal was image optimization for mobile users, streamlined photo management was an added benefit. By starting with in-depth user testing and site analytics, we let WWF's own audience help us develop a site that was intuitive, engaging, and effective — increasing revenue per visit by 22%.

View the case study online

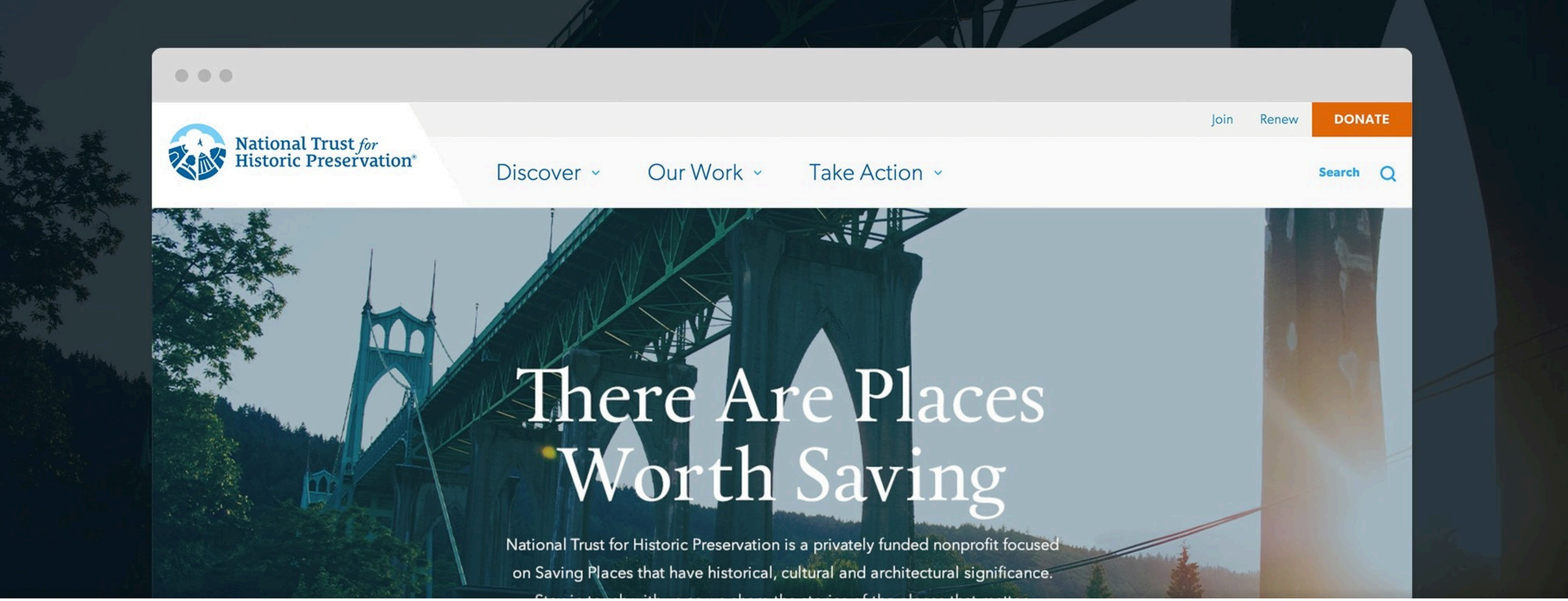




4: WordPress - Atlantic Philanthropies

Atlantic Philanthropies, a foundation dedicated to the service of humanity, needed a photo-rich and inspiring site to promote a new book, *Laying Foundations for Change*, which centered around capital projects funded by the foundation. Atlantic asked Viget for help translating the book into a digital medium and building out the WordPress site that could be updated to share progress on Atlantic's work.

View the case study online

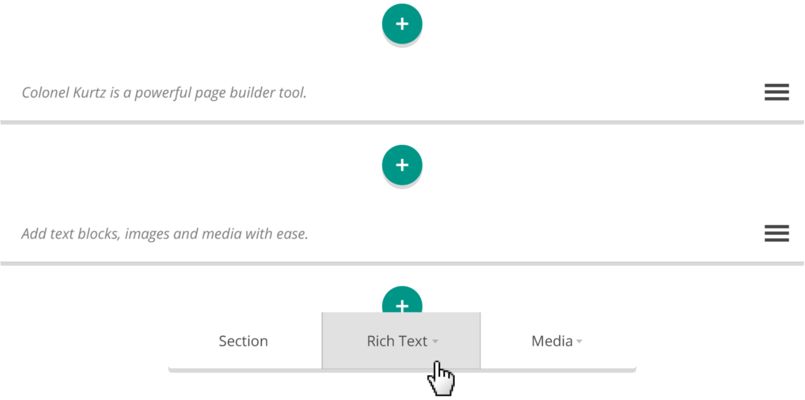


There Are Places Worth Saving

National Trust for Historic Preservation is a privately funded nonprofit focused on Saving Places that have historical, cultural and architectural significance.

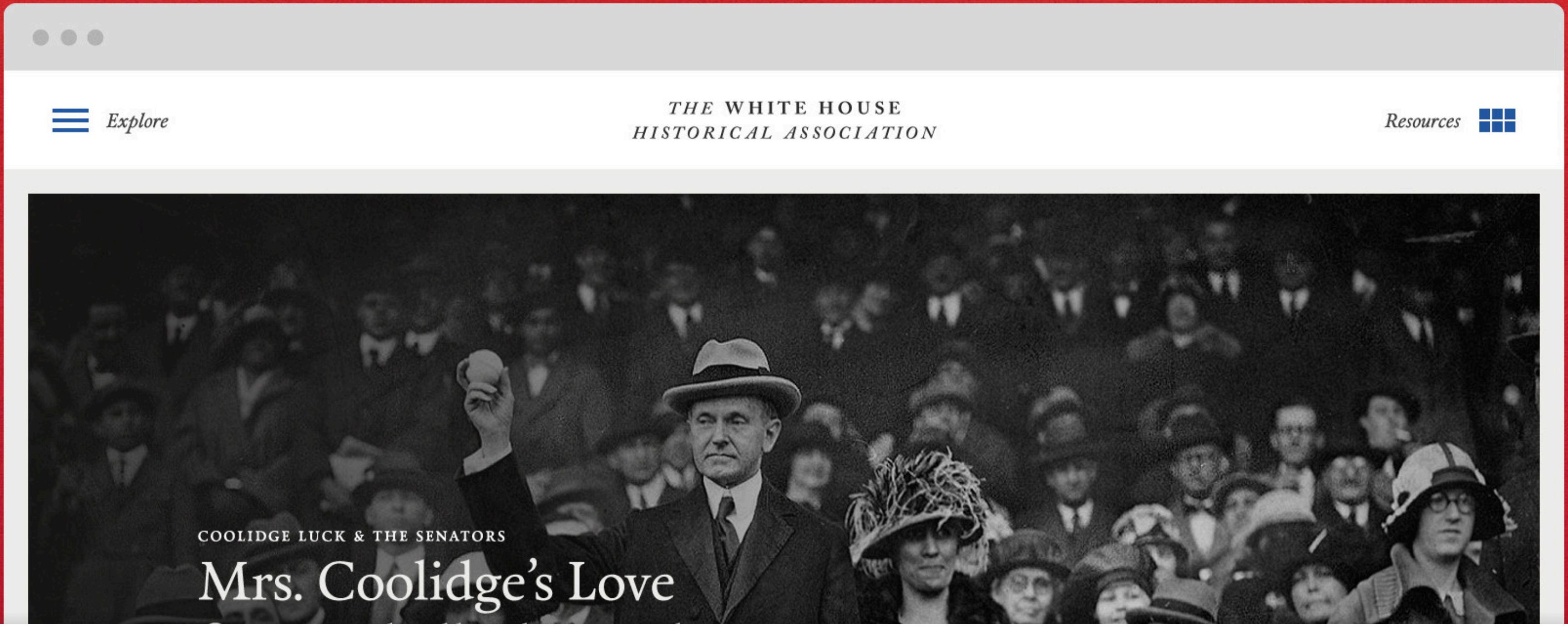
5: Colonel Kurtz — National Trust for Historic Preservation

An effort for the National Trust for Historic Preservation required a CMS that would strike a balance between structure and flexibility. The CMS itself was written in Ruby on Rails, which baked domain modeling and structured data into its development conventions, but made it difficult to introduce custom content. So, we used Colonel Kurtz, a JS library, to make it simple for multiple admins to compose beautiful, one-of-a-kind layouts, while still modeling the domain appropriately.


Colonel Kurtz is a powerful page builder tool.

Add text blocks, images and media with ease.

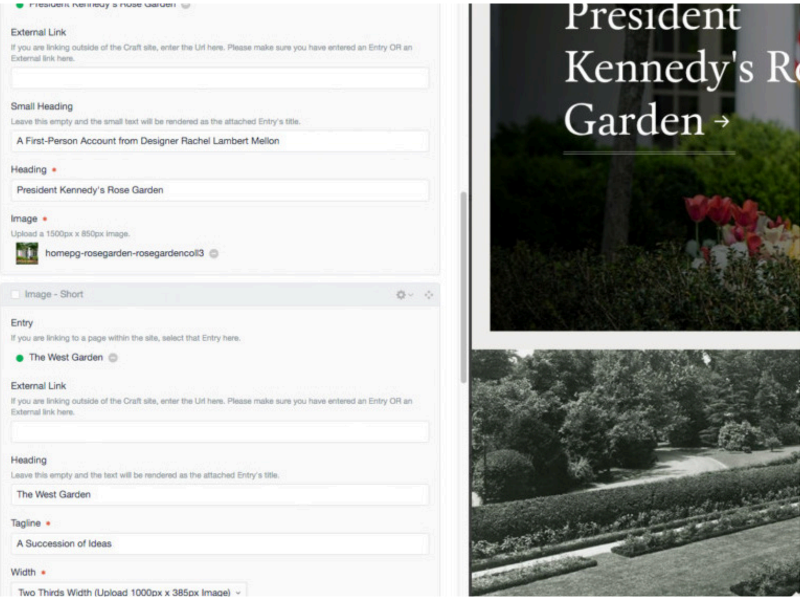
View the case study online



6: Craft — White House Historical Association

The White House Historical Association strives to protect, preserve, and provide public access to the rich history of America's Executive Mansion. The Association needed a better way to tell historical and organizational stories in the digital age. Viget implemented Craft over other CMS options because it fit the Association's publishing needs and developers could build new features — one of Craft's greatest benefits — instead of manipulate a limited library of plug-ins.

View the case study online



● President Kennedy's Rose Garden

External Link
If you are linking outside of the Craft site, enter the Url here. Please make sure you have entered an Entry OR an External link here.

Small Heading
Leave this empty and the small text will be rendered as the attached Entry's title.
A First-Person Account from Designer Rachel Lambert Mellon

Heading •
President Kennedy's Rose Garden

Image •
Upload a 1500px x 850px image.
homepg-rosegarden-rosegardencoll3

□ Image - Short

Entry
If you are linking to a page within the site, select that Entry here.
● The West Garden

External Link
If you are linking outside of the Craft site, enter the Url here. Please make sure you have entered an Entry OR an External link here.

Heading
Leave this empty and the text will be rendered as the attached Entry's title.
The West Garden

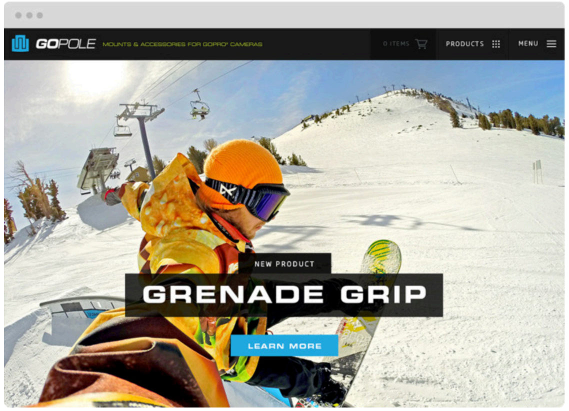
Tagline •
A Succession of Ideas

Width •
Two Thirds Width (Upload 1000px x 385px Image)

7: Ecommerce — GoPole.com

GoPole provides custom GoPro camera accessories for adventurers and athletes. GoPole needed to leverage amazing photography, as well as an engaging and demonstrative shopping experience. Viget wrote its own Craft/Shopify plug-in to give GoPole the best of both worlds: Craft's CMS and Shopify's ecommerce functionality.

View the case study online



CONCLUSION

We hope that you have found this book insightful.

If you have any questions about CMSs or about Viget, feel free to reach out directly to any of the authors in this ebook or just say hi at hello@viget.com. We'd love to hear from you.

We want your feedback! Please let us know your comments on this ebook so we can improve it. Email us at ebookfeedback@viget.com.

Thanks for reading.

Follow @viget on Twitter or send questions to hello@viget.com

viget