
CMSの新しいパラドックス コンテンツ管理での未来対立

プレストン・ソー • 2019年 12月 7日 • DrupalCamp DEN 名古屋 2019 キーノート

The new CMS paradox

Coming conflicts in content management

Preston So • December 7, 2019 • DrupalCamp DEN Nagoya 2019 keynote

はじめまして！

プレストン・ソーはイノベーションリーダー、研究者、スピーカー、そして「*Decoupled Drupal in Practice* (Apress 2018年)」の著者です。

Decoupled Drupalの世界的なリーダーであり、Decentralized Webや会話形デザインの専門家でもあります。プレストンは、Gatsby Inc.(ギャツビー社)でプリンシパル・プロダクト・マネージャーです。

過去3回にわたり、SXSWインタラクティブで講演し、大陸にて3つの言語で基調講演をおこなう人気スピーカーでもあります。

linkedin.com/in/prestonso • [@prestonso](https://twitter.com/prestonso) • [#NewCMSParadox](https://twitter.com/#NewCMSParadox)

Welcome!

Preston So is an innovation lead, researcher, speaker, and author of *Decoupled Drupal in Practice* (Apress, 2018).

A globally recognized voice on decoupled Drupal and subject matter expert in the decentralized web and conversational design, Preston is Principal Product Manager at Gatsby.

Having spoken three times at SXSW Interactive, Preston is a sought-after presenter with keynotes on five continents and in three languages.

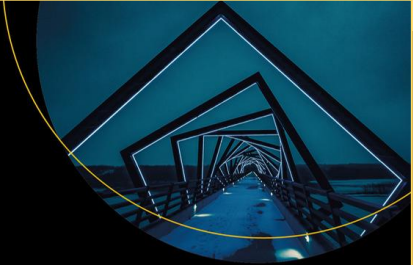
[linkedin.com/in/prestonso](https://www.linkedin.com/in/prestonso) • [@prestonso](https://twitter.com/prestonso) • [#NewCMSParadox](https://twitter.com/prestonso)

Decoupled Drupal in Practice

Available now!

tiny.cc/decoupled-in-practice

In the first and only comprehensive guide to decoupled Drupal across the stack, with a foreword by Drupal project lead Dries Buytaert, learn everything you need to know about decoupled Drupal—from building the back end and designing APIs to integrating with front-end technologies.



Decoupled Drupal in Practice

Architect and Implement Decoupled
Drupal Architectures Across the Stack

—
Preston So

Foreword by Dries Buytaert

apress®



DECOUPLED DAYS 2020

New York City
July 22–23, 2020

Registration opening soon

Visit decoupleddeays.com for more updates about tickets and our call for papers.

Sponsorships

For sponsorship information, reach out to our team at decoupleddeays@gmail.com.

@decoupleddeays

Follow us on Twitter to stay updated on important announcements.

[#decoupleddeays](https://twitter.com/decoupleddeays) • decoupleddeays.com • [@decoupleddeays](https://twitter.com/decoupleddeays)

What we'll cover

- The uneasy alliance between marketers and developers
- How decoupled CMSs have attempted to fill in the gaps
- The distributed CMS: From system to stack
- The new CMS paradox: Marketing and tech at odds
- Epilogue: Getting ready for the new CMS

1

The uneasy alliance between marketers and developers

**Drupal and other traditional CMSs
are experiencing divergent priorities**

**Marketers and content editors want
control over non-web experiences**



Cambrian explosion



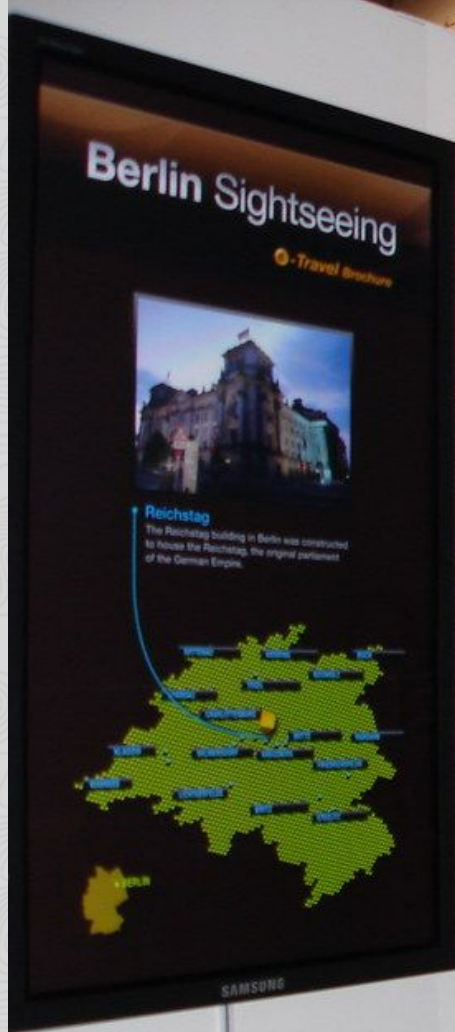
Mobile applications



Set-top boxes



Wearables



PDP Series



Samsung Network Solution
MX Series



Digital signage

**JAUME
ARENAS
I CLAVELL**

MATARO (1918-1983)
ABSTRACCIÓ, S/D
AQUAREL·LA SOBRE
PAPER

▶

DESCOBREIX MÉS
SOBRE L'AUTOR

Augmented reality

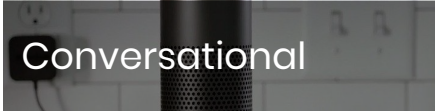


Conversational interfaces

**Marketers and content editors want
control over non-web experiences**

A close-up of a smartwatch with a blue screen displaying a call log with names like 'James', 'Express', 'Call', and 'Liam'.

Wearables

A black smart speaker with a mesh grille, sitting on a wooden surface.

Conversational

A digital signage display in a store, showing various advertisements and product information.

Digital signage

A digital screen displaying an augmented reality interface with text like 'TCLAVELL', 'MATARO (1918-1983)', 'ABSTRACTO', and 'DESCOBRIR MEC'.A red smartphone lying on a red surface, with a hand partially visible on the right.

Mobile

A close-up of several buttons for streaming services: 'NETFLIX', 'sling', and 'vudu'.A solid orange rectangular background.

Websites

A solid orange rectangular background.

Responsive

Developers

Marketers
Content editors

The incongruity of content management today

**Developers and stakeholders want
affordability, flexibility, performance**



Infrastructural affordability

Infrastructural affordability

- **TCO is increasing.** The total cost of ownership of a web infrastructure is increasing at an unprecedented rate, with many large Drupal sites now running more than \$1M per year.
- **Cloud solutions are expensive.** Serverless solutions such as Netlify and Zeit have proven to be much more affordable than established solutions like Acquia.



Resourcing affordability

Resourcing affordability

- **CMS developers are in high demand.** Demand is high for specialized skills like Drupal and WordPress, and developers command high prices that are increasingly unaffordable.
- **Resourcing is restrictive.** The amount of specialization in CMS development now results in a high barrier to entry for new developers and limited options for companies looking to hire.



Flexibility

Flexibility and interchangeability

- **Monolithic CMS is not flexible.** Service integrations in Drupal and WordPress require modules or plugins; new solutions cannot be substituted easily for Drupal features.
- **Interchangeable services are in vogue.** Companies today are interested in working with more flexible vendors with less onerous prospects for replatforming.



Performance

Performance and speed

- **Interactive experiences.** Today, users' expectations have far outpaced traditional CMS features, most of which lack link prefetching or image lazy-loading.
- **Performance optimizations.** Developers are increasingly seeking performance optimizations at every stage of the process, including both payload and delivery optimizations.



Daniel Case
CC BY-SA 3.0

Security

Security

- **Vulnerabilities are growing.** Security advisories are increasing in both Drupal and WordPress due to their popularity.
- **Decoupled aids security.** Decoupled architectures can help lock down CMS layers and prevent outside access to certain services.

**Is it possible to satisfy the needs of
both marketers and technologists?**



**The separation of concerns
between data and presentation
remains sacrosanct**

2

**How decoupled CMSs
attempt to fill in the gaps**

The first wave

1995–2006

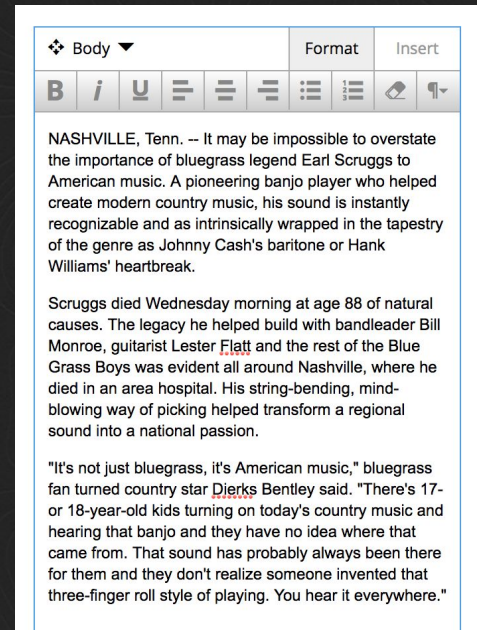
the **static** CMS

- Essentially glorified static site generators with rudimentary templating
- Application building was rudimentary; mostly leveraged to build content-rich websites
- No interactive client-side behavior

The second wave 2002–2014

- Dynamic server-side applications for the first time
- Extensive abstraction from the data layer to the templating layer
- Rudimentary interactive client-side behavior (see Drupal's Ajax API)

the dynamic CMS the monolithic CMS



The third wave 2012–2020

- Dynamic client-side applications for the first time, thanks to web services and universal JavaScript and owing to channel explosion
- Full **architectural** decoupling between structured data and presentation
- Complex interactive client-side behavior made easier and integrations with JavaScript frameworks (see Prismic edit button)

the **headless** CMS
the **decoupled** CMS

[Content model](#) > Article



Article

[Edit](#)

Actions ▾

Save

[+ Add field](#)

Fields (7)

JSON preview

Duplicate

Delete



Title Short text

Entry title

[Settings](#)



Body Long text

[Settings](#)



Media References, many

[Settings](#)



Author Reference

[Settings](#)









Tags Short text, list

[Settings](#)



Contentful






Work

Planned ▾


Archived


Search


Christmas release status 


Publish options


Private members



 This release will be published in 2 day (27 fevrier 2016) [Edit](#)






















 Publish now

 Unschedule

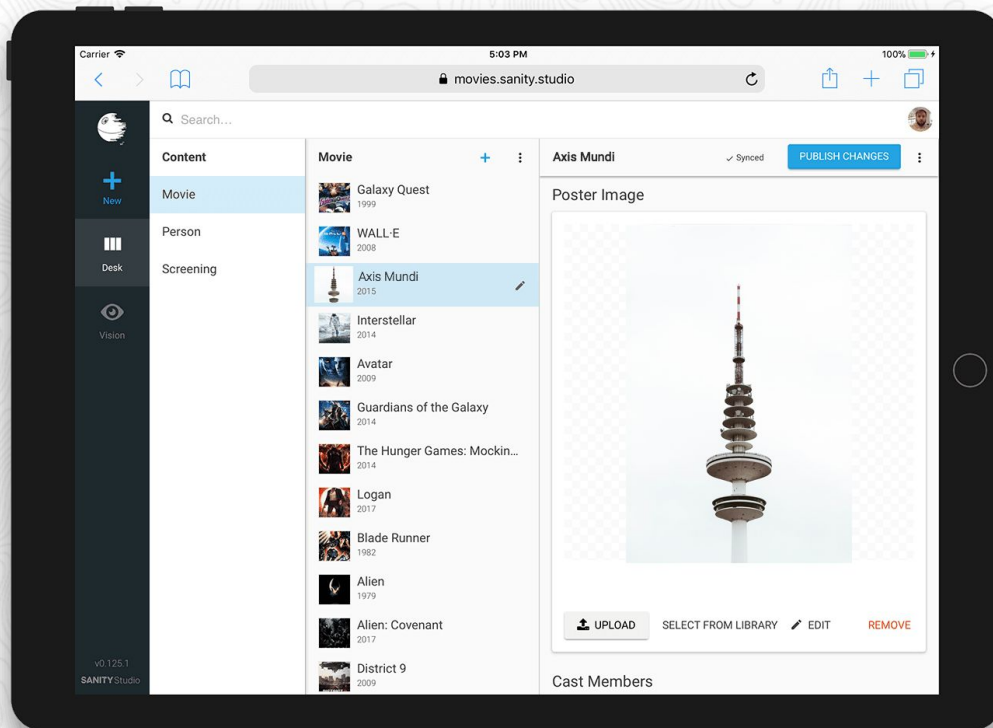
 Preview

 Delete this release

Type ▾ Authors ▾ Tags ▾ Status ▾  [Write](#) 

State	Name	Type	Last update	Author
	The Basics Of Buying A Telescope	Blogpost	10/18/2016 / 12:12	 
	The Basics Of Buying A Telescope TO TRANSLATE	Blogpost	12/23/2016	 
	How To Look Up TO TRANSLATE	Blogpost	09/25/2016	 
	The Basics Of Buying A Telescope	Blogpost	09/02/2016	 
	Asteroids	Blogpost	07/20/2016	 
	How To Look Up	Blogpost	07/01/2016	 
	Astronomy Or Astrology	Blogpost	07/19/2016	 

Prismic



Sanity.io

Developer Resources

Browse: [Home](#) / [REST API Handbook](#) / REST API Handbook



CHAPTERS

1 [REST API Handbook](#)

2 [Reference](#)

[Posts](#)
[Post Revisions](#)
[Categories](#)
[Tags](#)
[Pages](#)
[Comments](#)
[Taxonomies](#)
[Media](#)
[Users](#)
[Post Types](#)
[Post Statuses](#)

REST API Handbook

The WordPress REST API provides API endpoints for WordPress data types that allow developers to interact with sites remotely by sending and receiving [JSON](#) (JavaScript Object Notation) objects. JSON is an open standard data format that is lightweight and human-readable, and looks like Objects do in JavaScript; hence the name. When you send content to or make a request to the API, the response will be returned in JSON. This enables developers to create, read and update WordPress content from client-side JavaScript or from external applications, even those written in languages beyond PHP.

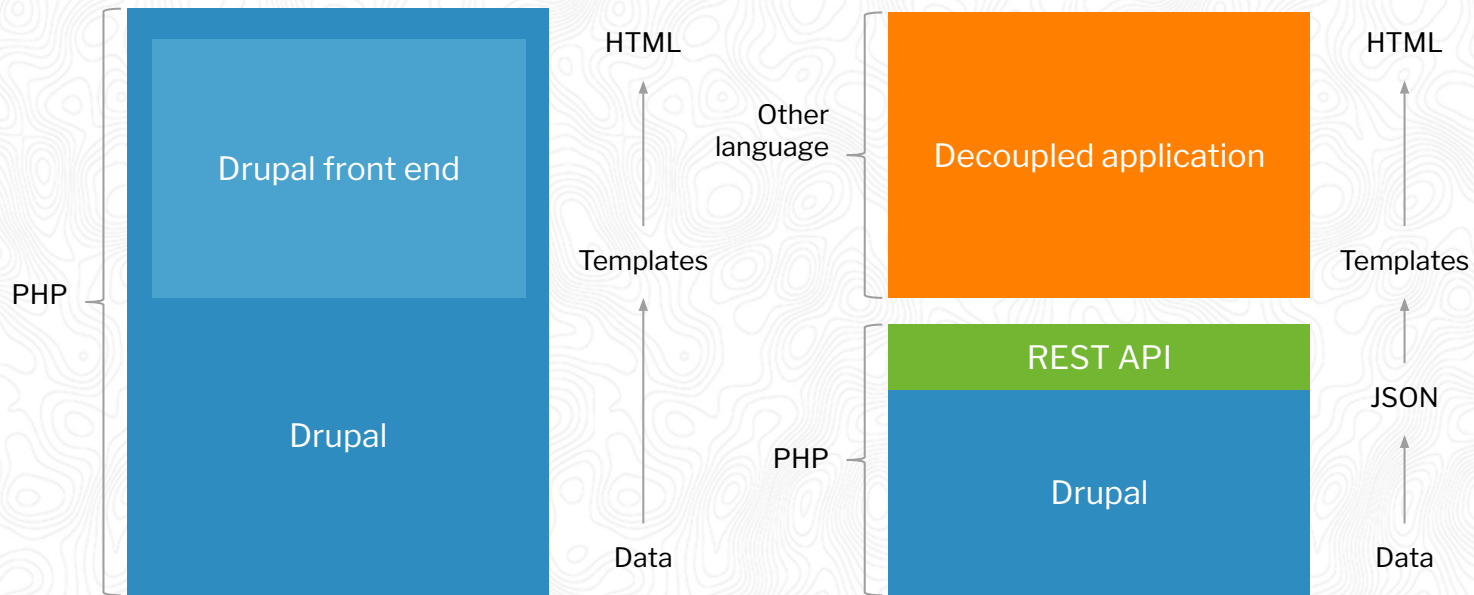
TOPICS

[Why use the WordPress REST API](#)

[Key Concepts](#)

- [Routes & Endpoints](#)
- [Requests](#)
- [Responses](#)
- [Schema](#)
- [Controller Classes](#)

Headless WordPress



Decoupled Drupal

**Decoupled Drupal is the use of
Drupal as a content service for
consumer applications**

**Evolution catalyzed by dual forces of
innovation in front-end development
and in digital experiences**



Interchangeable parts

Traditional development

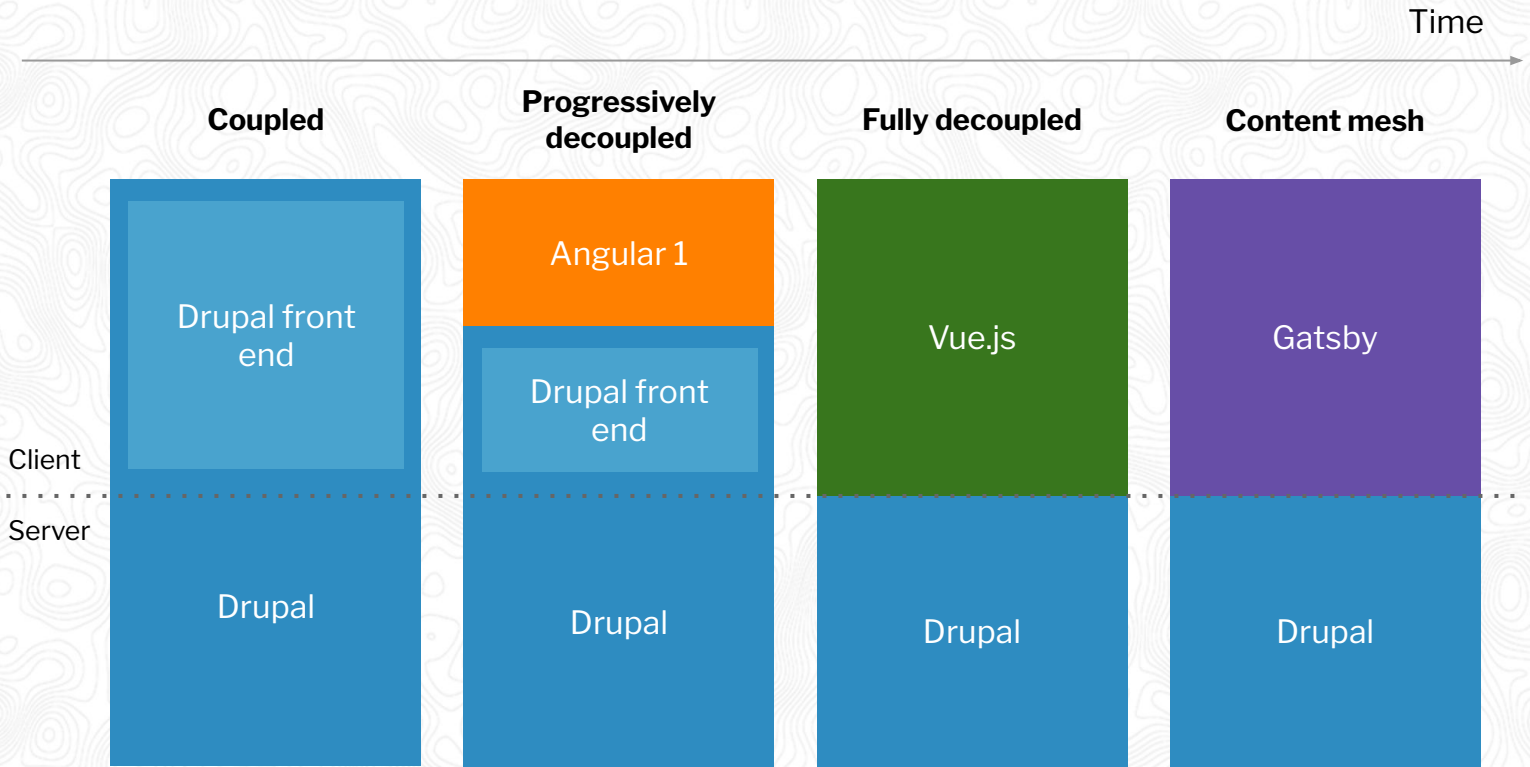


Pipelined development

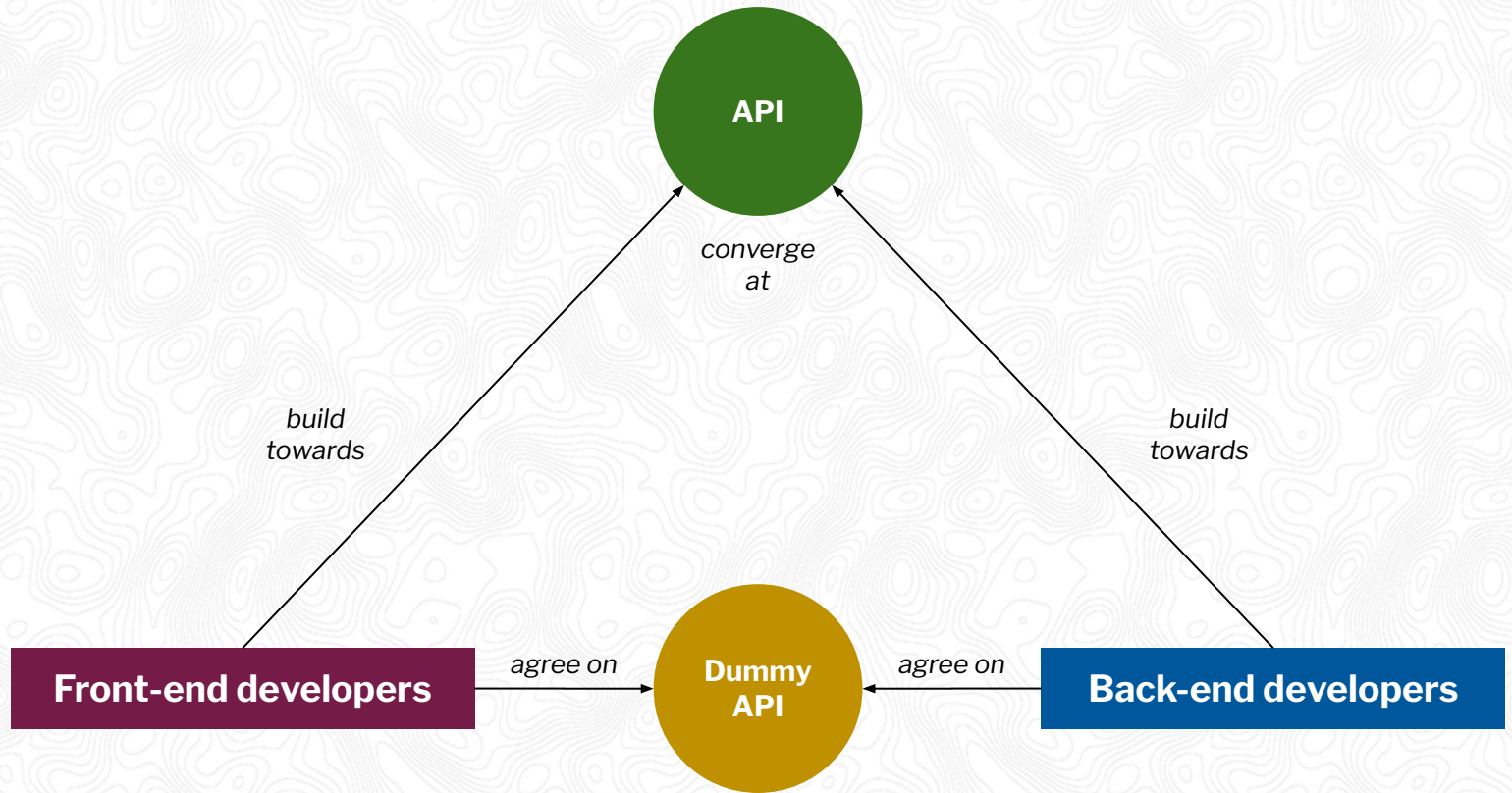


Project duration →

Pipelined development



Interchangeable presentation layers



API convergence

Coupled updates	Monolithic update	Monolithic update
Drupal 6 front end	Drupal 7 front end	Drupal 8 front end
Drupal 6 back end	Drupal 7 back end	Drupal 8 back end

Decoupled updates	Gatsby update	Gatsby update
Gatsby v0	Gatsby v1	Gatsby v2
Drupal 7	Drupal 8	Drupal 9
	Drupal update	Drupal update

Updates and maintenance

3

The distributed CMS **From system to stack**

The fourth wave

2018–

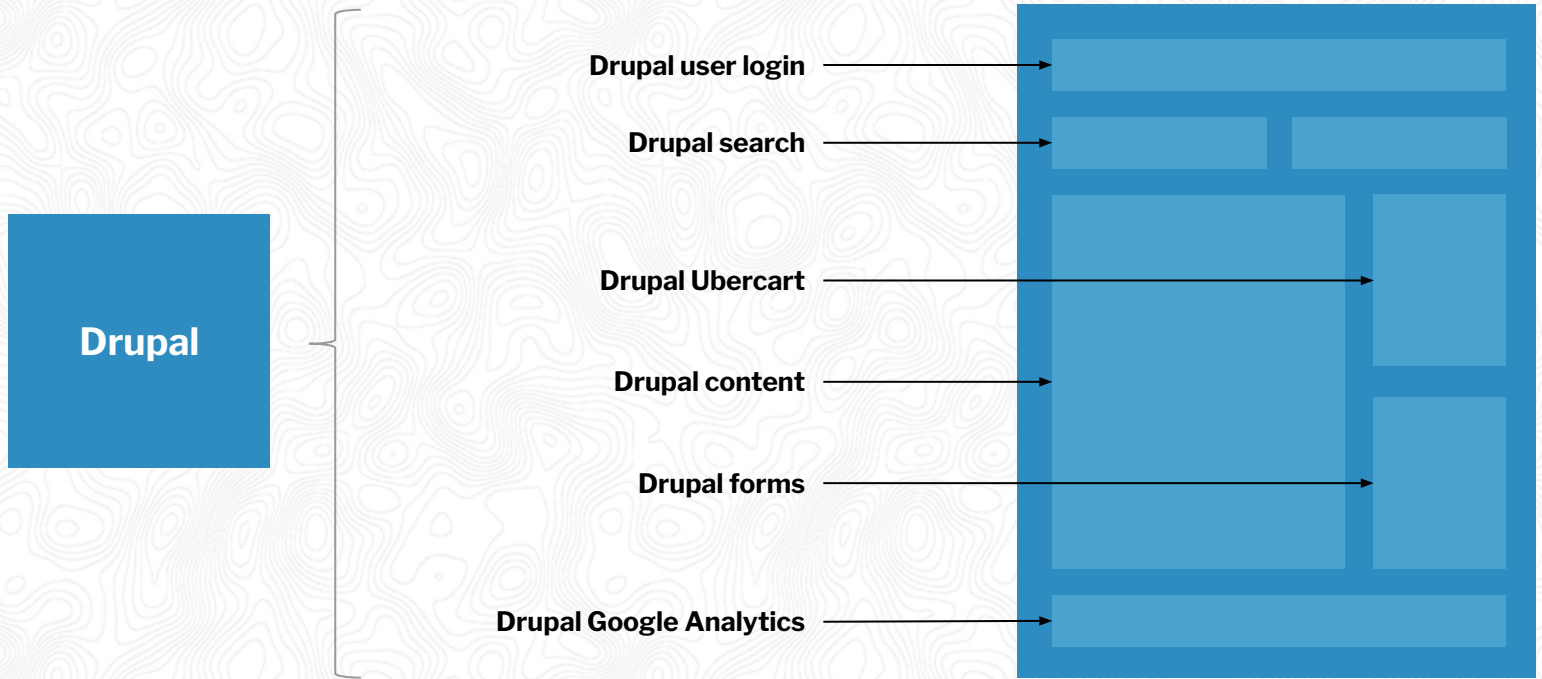
the **distributed** CMS

- Richly interactive and performant universal JavaScript applications with initial server-side rendering
- Full **presentational** decoupling with interchangeable services and data source agnosticism
- Complex interactive client-side behavior as a **first-class concern**

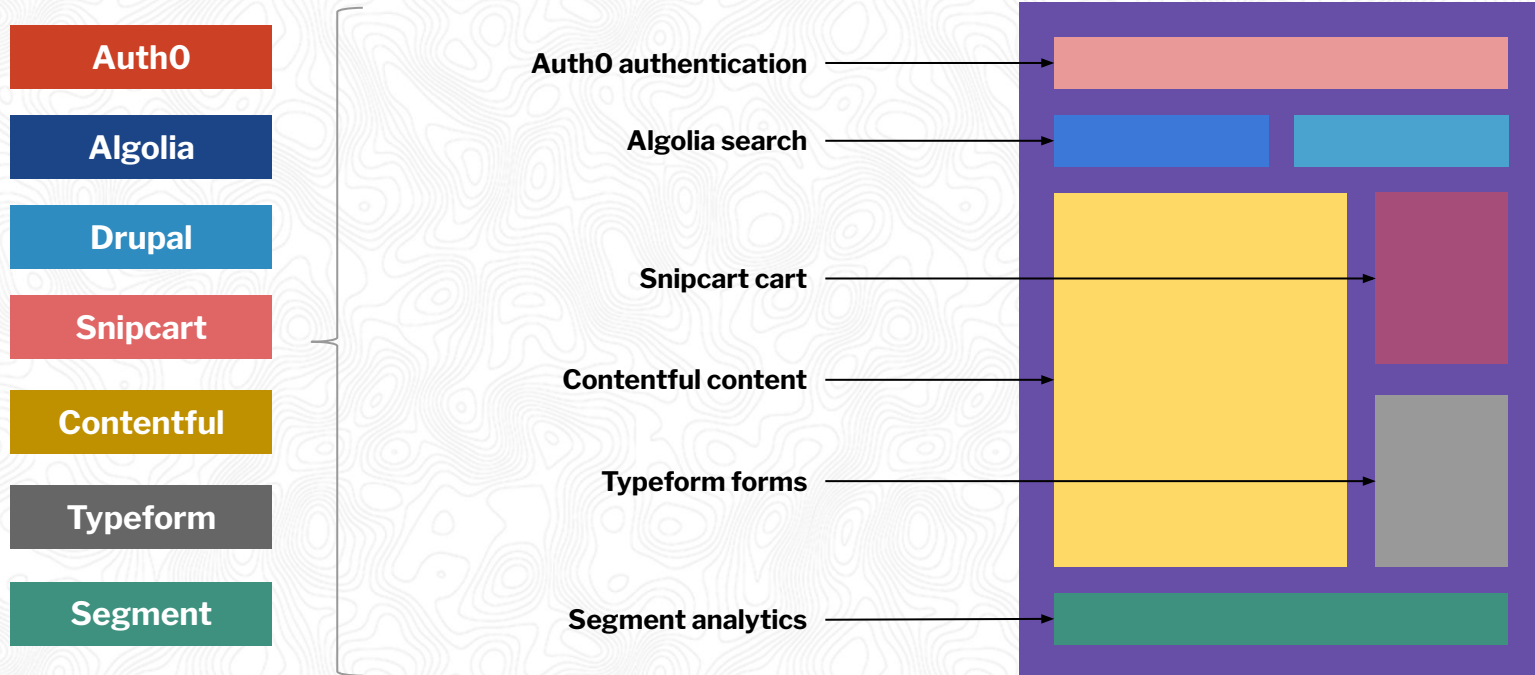
**Microservices and orchestration
introduced the idea of a service mesh**

**A content mesh pairs services with
an agnostic front-end layer**

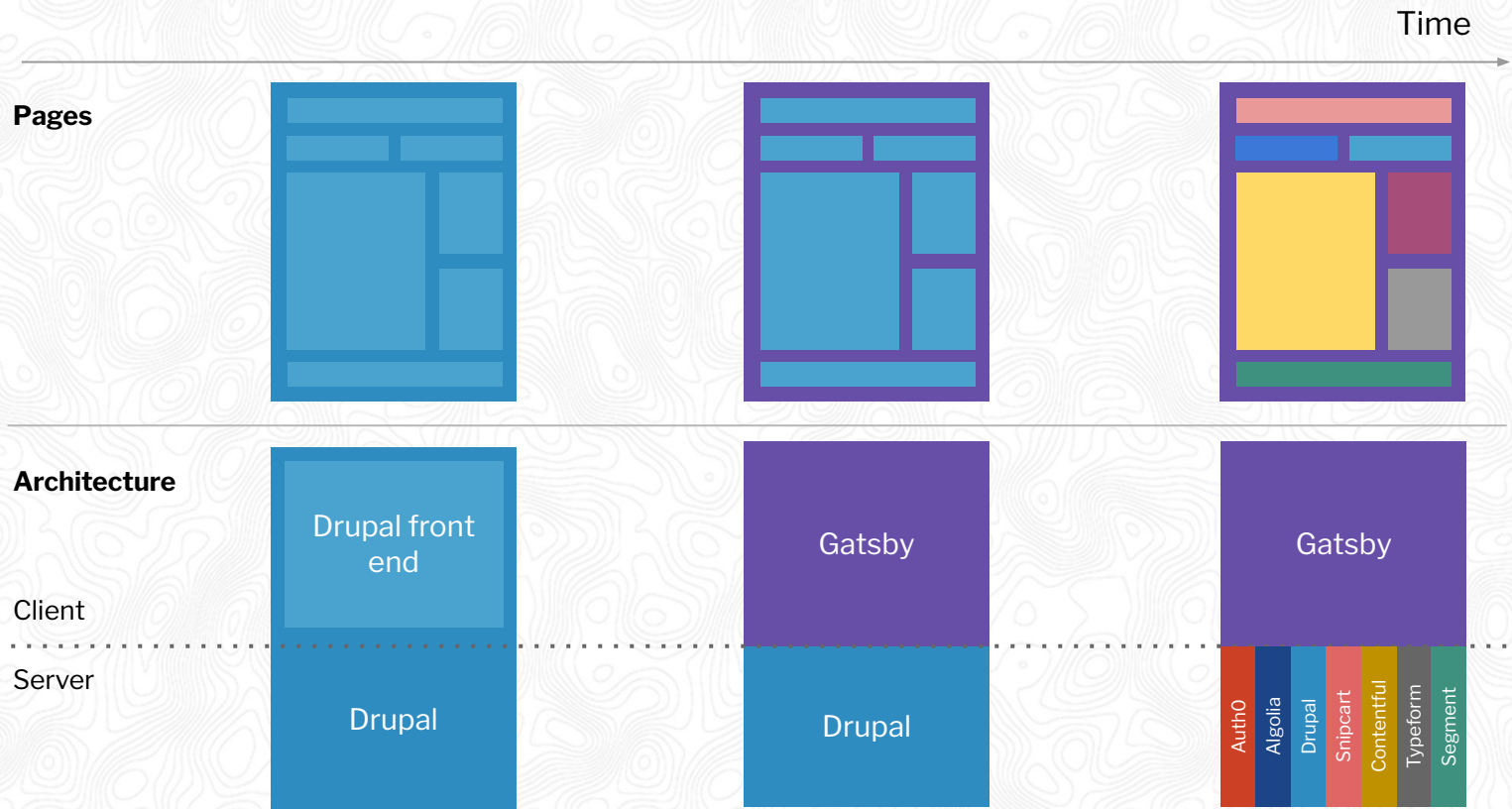
**As a result of using the content mesh
the *page* becomes decoupled too**



Traditional CMSs handled all concerns at the expense of simplicity



But the content mesh allows you to choose your services arbitrarily



From monolithic CMS to decoupled CMS to distributed CMS



Infrastructural affordability

Infrastructural affordability

- **TCO will decrease.** The distributed CMS will reduce your total cost of ownership by bringing your hosting costs to near zero and allowing vendor flexibility without lock-in.
- **Serverless solutions are cheap.** Netlify and Zeit offer powerful free tiers, and AWS is entering the space as well.



Resourcing affordability

Resourcing affordability

- **JavaScript developers are easier to find.** In today's web community, JavaScript is the language of choice and React, Vue.js, and GraphQL the technologies of choice.
- **Resourcing is less onerous.** JavaScript developers are more affordable, more available, and are often more accustomed to approaches like pipelined development.



Flexibility

Flexibility and interchangeability

- **Distributed CMS is flexible by design.**
Interchangeability of services means that decoupled *pages* are possible, with flexible services.
- **Limit scope to plug and play.** With a more focused and services-oriented approach, many feature-based services are seeing success and velocity, such as Algolia, Snipcart, and Typeform.



Performance

Performance and speed

- **Progressive enhancement and interactivity.** Gatsby and other distributed CMS technologies have full support for accessibility and progressive enhancement in addition to interactivity.
- **Performance optimizations.** Thanks to serverless solutions and CDNs, it is now possible to achieve unprecedented levels of performance.

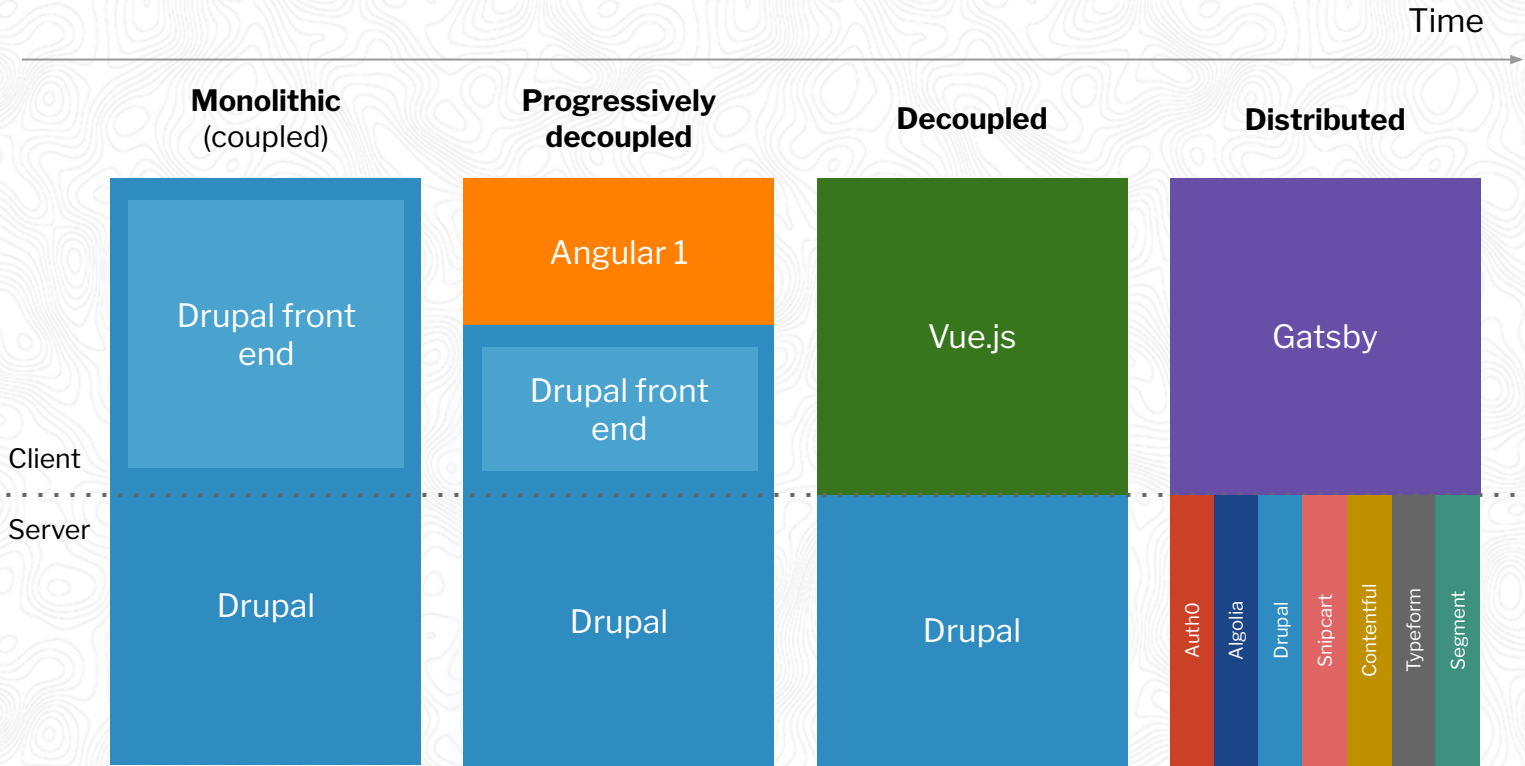


Daniel Case
CC BY-SA 3.0

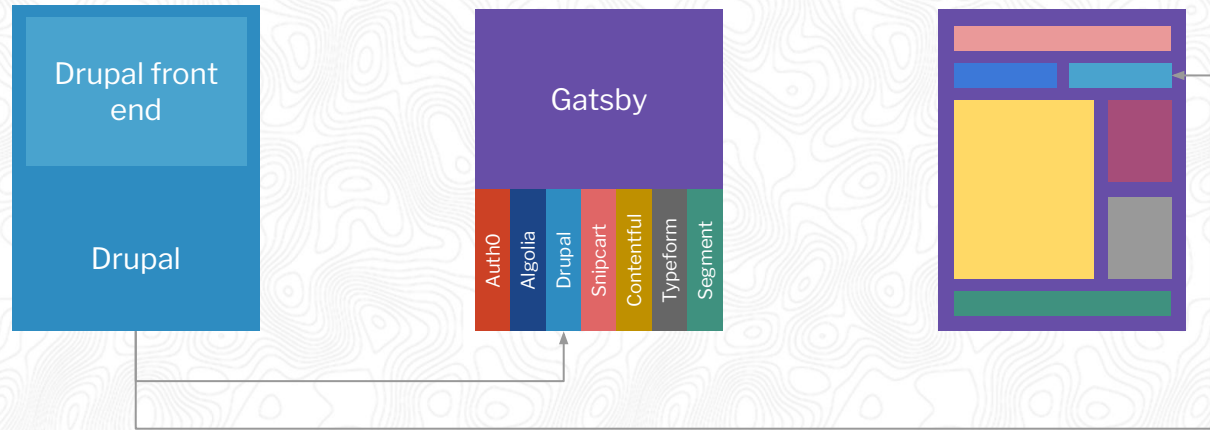
Security

Security

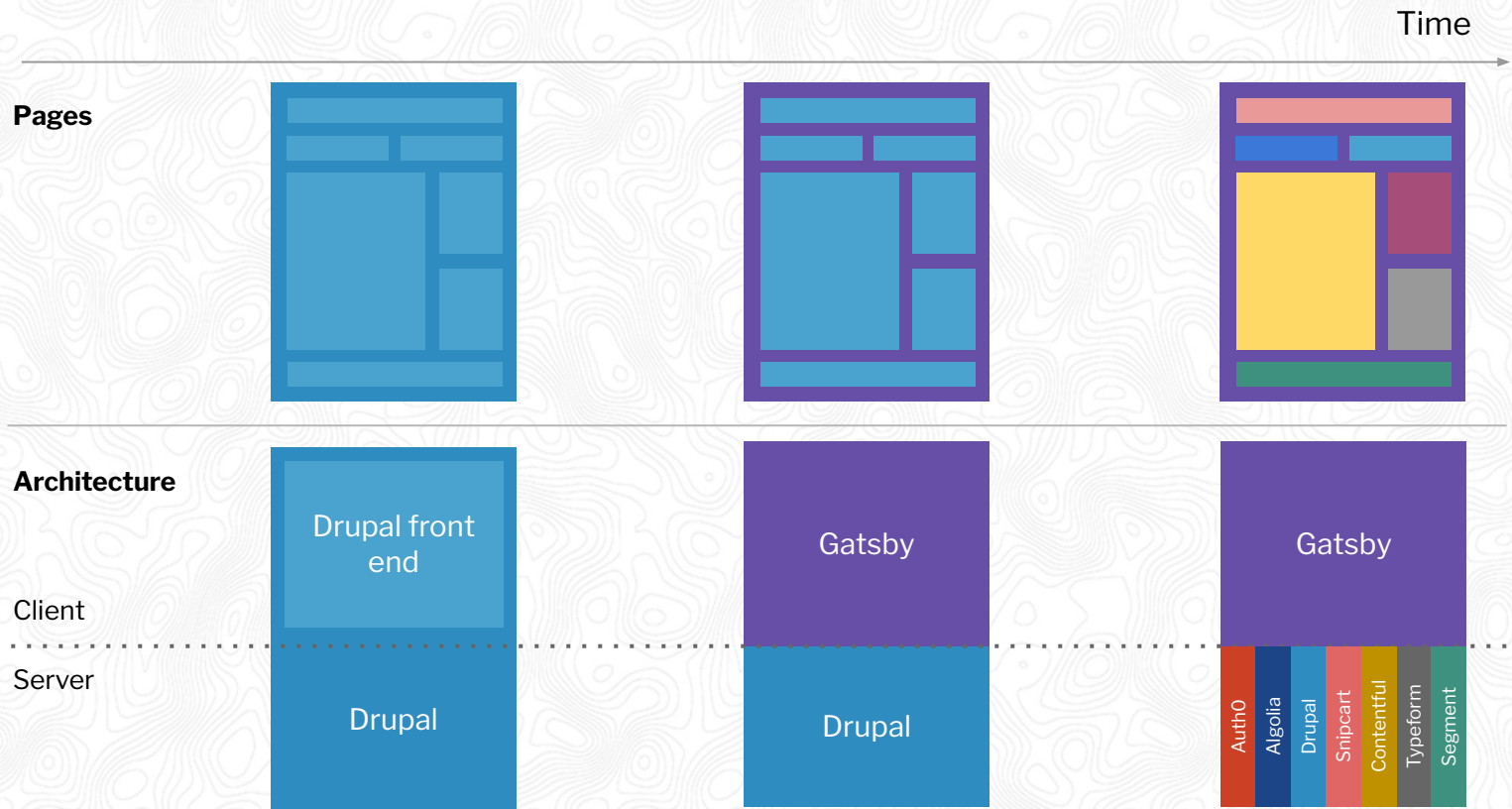
- **Spread vectors across services.** Multiple interchangeable services makes it more difficult for attackers to access critical parts of your CMS architecture (e.g. delegate forms to Typeform and cart functionality to Snipcart).
- **Lock down your CMS.** With web compilers like Gatsby, it's possible to build a fully functional website consuming Drupal without your Drupal site ever needing to be online or accessible.



Evolution of content management systems/stacks over time

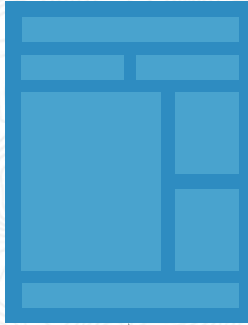


From decoupled CMS to distributed CMS: content management stack

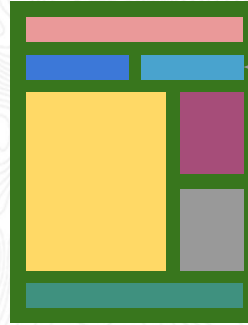


From monolithic CMS to decoupled CMS to distributed CMS

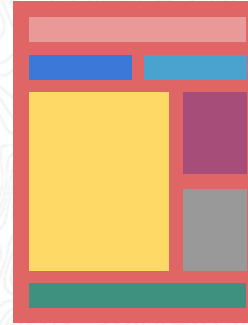
**Drupal front
end**



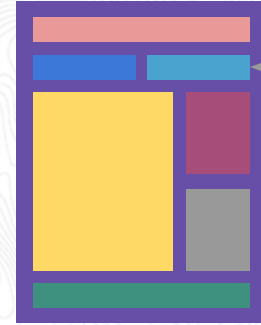
Vue.js



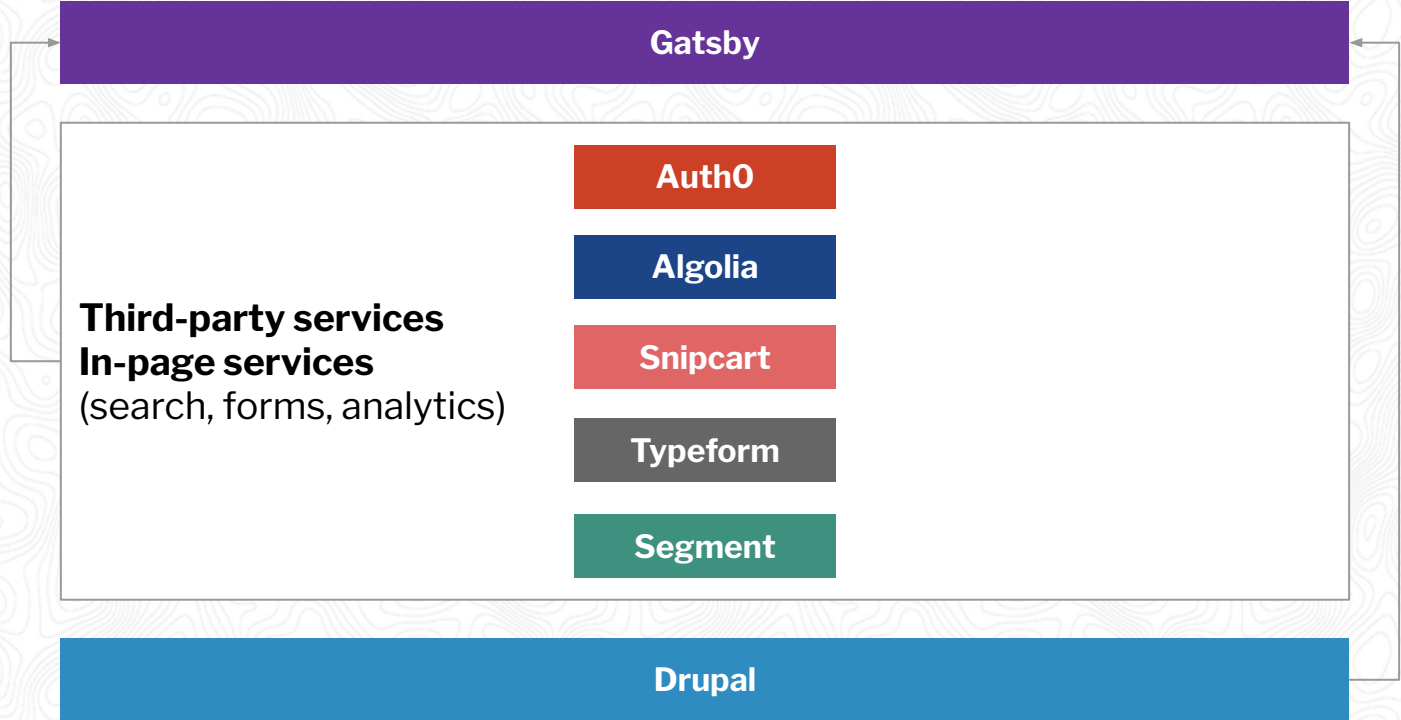
Ember.js



Gatsby



From content management systems to content management stacks



Distributed CMS: Content management stacks

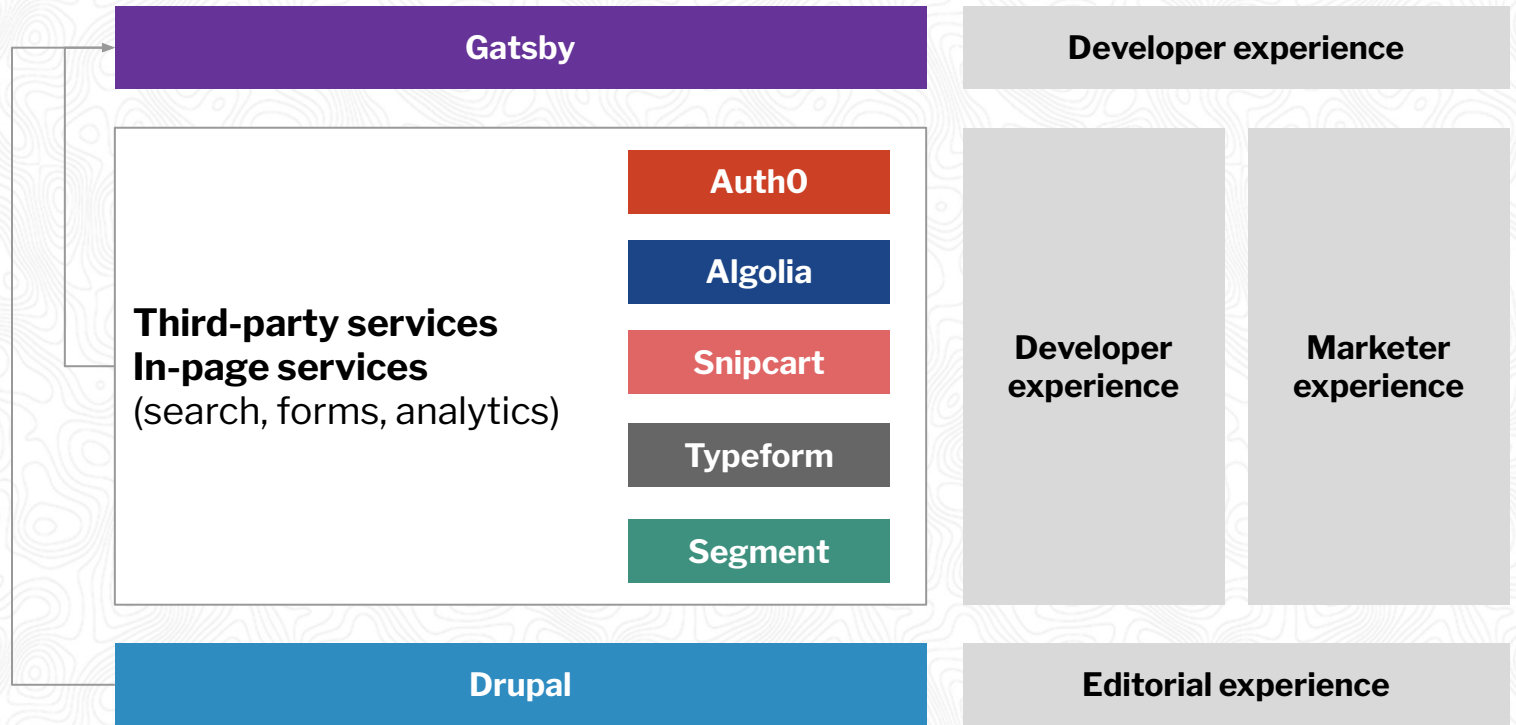
**The new content management system
is not a single, contiguous system**



**The content management *stack*
is a distributed, multifaceted milieu**

4

The new CMS paradox Marketing and tech at odds



Distributed CMS: Content management stacks

**In distributed CMS, editors can keep
their existing editorial interfaces**

[+ Add content](#)

Content

Comments ▾

Files

[Home](#) > [Content](#) > [Ada Lovelace](#)

Edit

Ada Lovelace

Title (required)

Ada Lovelace

Cos

B**I**

Format ▾

Source

Ada Augusta Byron King, comtessa de Lovelace, (Londres, 10 de desembre de 1815 - 27 de novembre de 1852) és la primera programadora en la història dels ordinadors. Matemàtica i física, va col·laborar amb Charles Babbage en el disseny d'una màquina analítica capaç de resoldre equacions diferencials.

A les seves notes de treball sobre aquesta, va incloure el que ara es considera el primer algorisme que es va intentar que dugués a terme una màquina. És per això que se la denomina la primera programadora.

La vida d'Ada Lovelace va transcórrer a la primera meitat del segle XIX, sota la influència de les idees clàssiques de la societat victoriana, molt arrelades a l'alta classe social a la qual pertanyia però a la vegada amarades dels ideals romàntics. El coneixement científic ja no era una referència de prestigi social, sinó la manera de no quedar-se al marge del "progrés", autèntica font de riquesa.

Save

Last save

08/20/2017 - 16:37

Author

admin

Published



Preview

Delete

MENU SETTINGS ▾

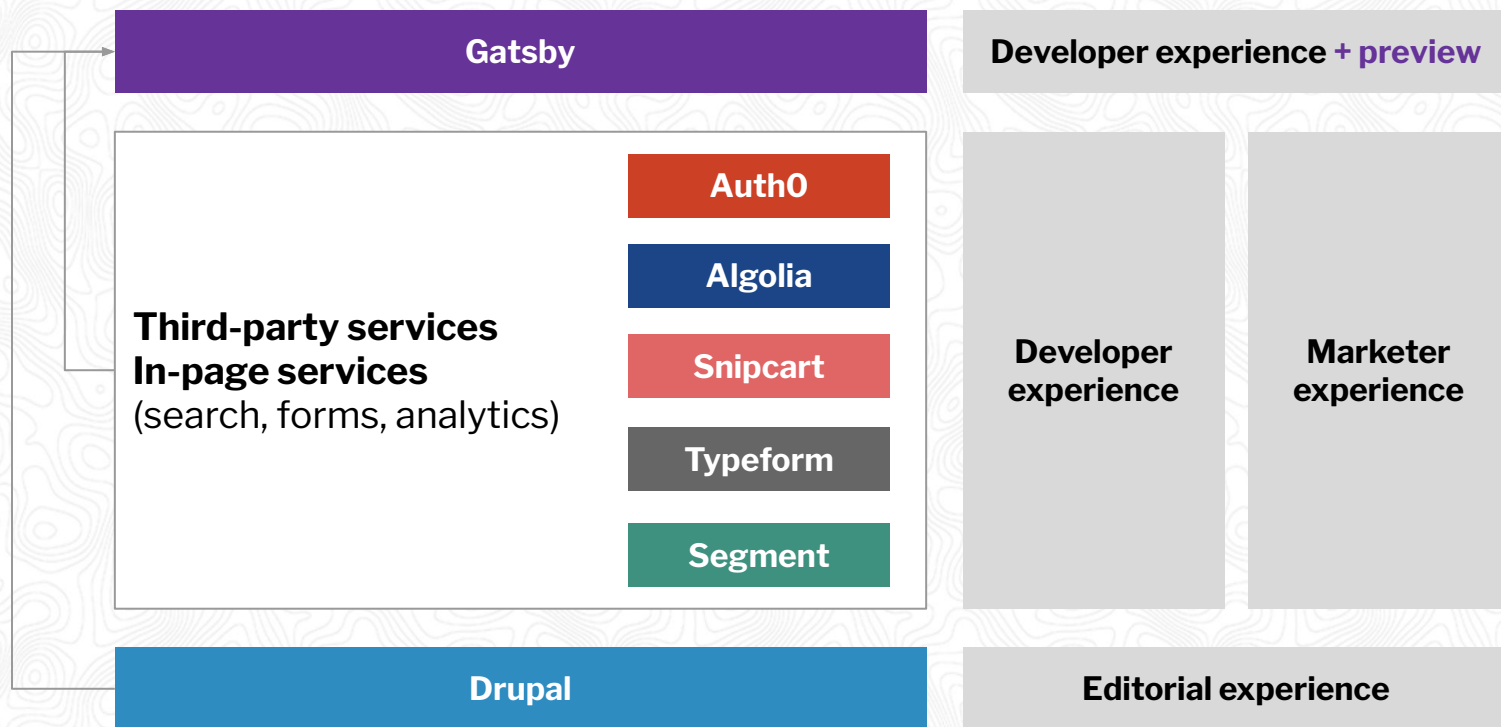
Provide menu link



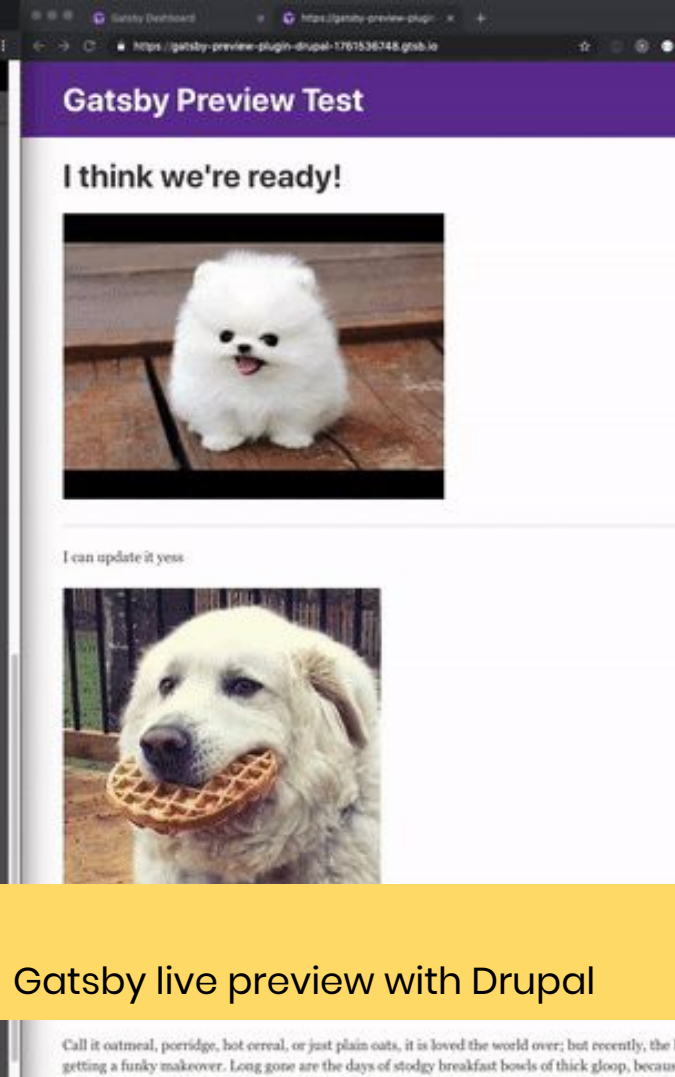
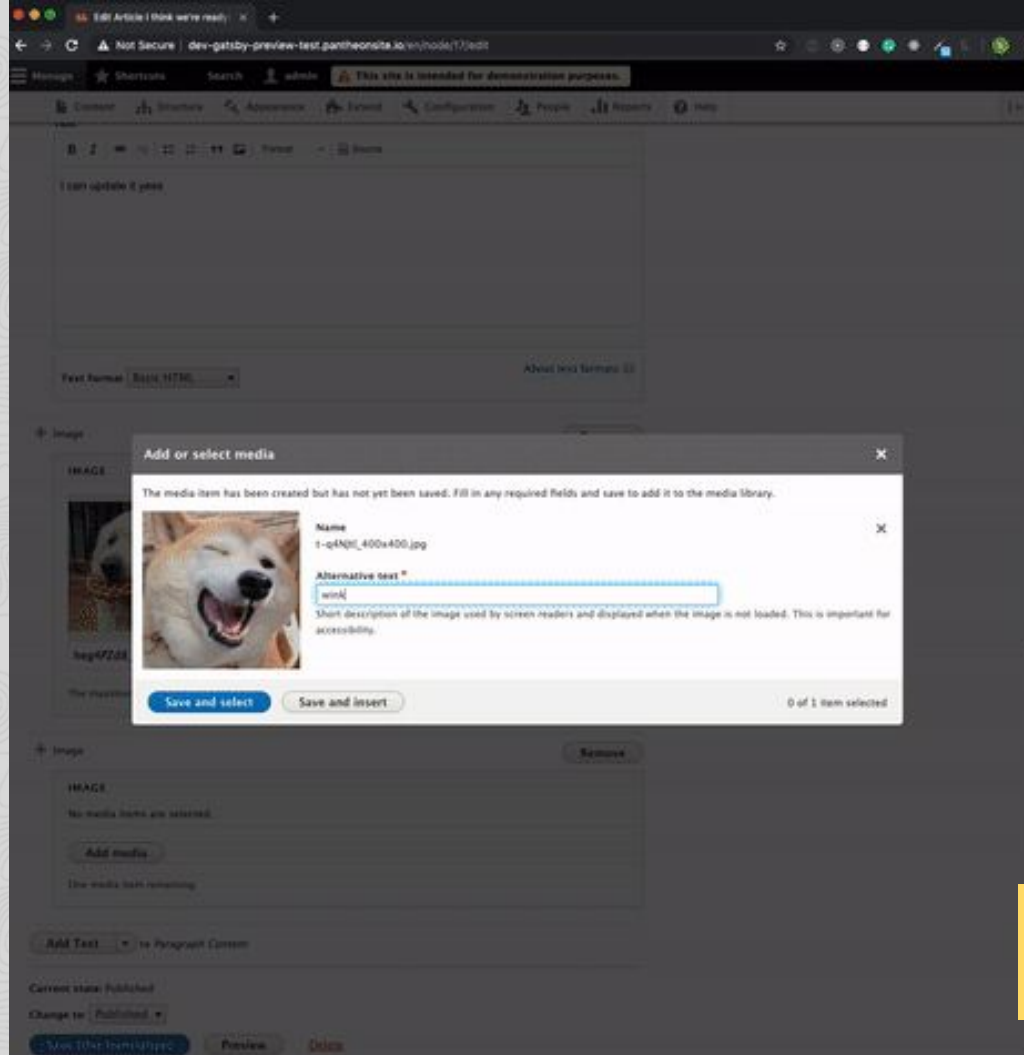
COMMENT SETTINGS >

Admin UI Initiative (ckrina)

**For marketers, diversifying into
new front ends can be challenging**

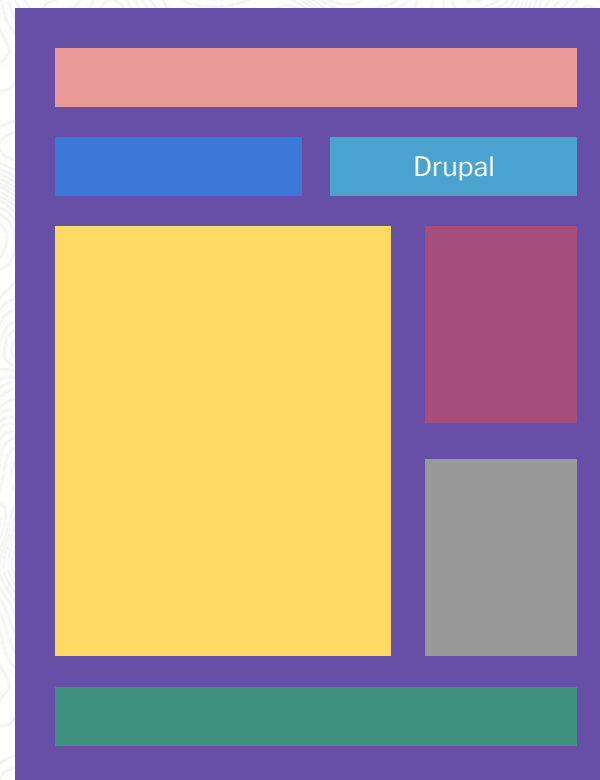
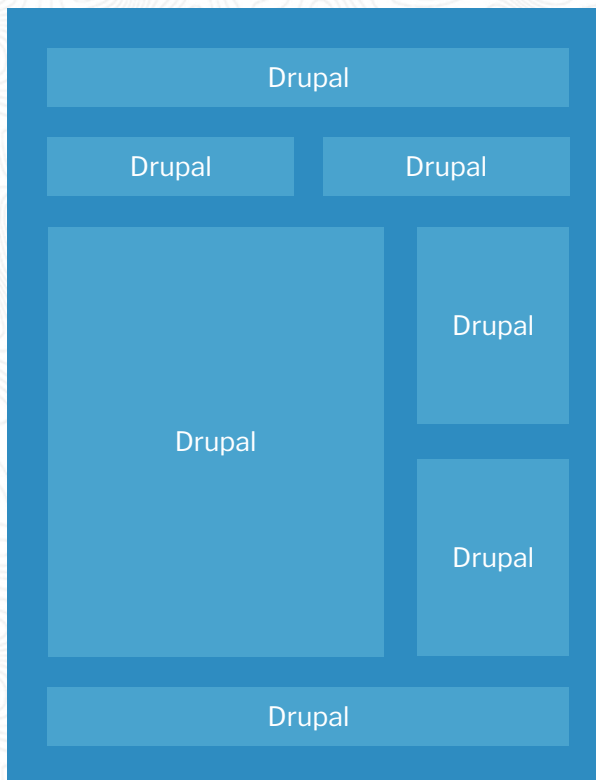


Distributed CMS: Content management stacks



**In distributed CMS, marketers can
access other powerful services easily**

**For marketers, diversifying services
with flexible options can be proactive**



**But they can also lead to the loss of
nonnegotiable marketing features**

Core CMS marketing features

- **Preview.** Preview is becoming more of a solved problem, but user interfaces that are easy to use for marketing teams are still distant.
- **Layout.** With a clean separation between data and presentation, how can marketers create, select, and manipulate layouts?

Core CMS marketing features

- **Contextual links.** Drupal has long been famous for its in-context links that point to administrative pages in the CMS. How will we account for these?
- **In-context editing.** Though Quick Edit in Drupal is used less frequently than before, how can we replicate a similar experience for marketing teams?

**Should CMSs decouple by default and
adopt one front-end framework?**

**Should CMSs refrain from handling
aspects of presentation altogether?**

**What happens when marketers are
subjected to intensifying feature loss?**



**There are no easy answers;
monolithic, decoupled, and
distributed CMS remain relevant**

The distributed CMS: How the decoupled CMS endgame will impact your organization

June 6, 2019

The decoupled content management system (CMS) is both reinventing the way we think about content management and evolving how we implement and deploy the software architectures surrounding it. Today, what I call the third wave of the content management system, namely the proliferation of headless or decoupled CMSs that focus on a separation of concerns between structured content and its presentation, is in full swing. But the next wave, the distributed CMS, is just around the corner.

Organizations everywhere are now contending with two seemingly divergent trends: a *channel explosion* in which users expect content at their beck and call on every device and every form factor, and an *architectural decoupling* in which CMSs are no longer contiguously interconnected software systems but rather networks of interchangeable services with limited purposes. In short, they are seeking *distributed* content management, not simply decoupled content management.

Looking for even more about distributed CMS?
Watch for new content on my blog preston.so and @prestonso

5

Epilogue

Getting ready for the new CMS



**A distributed CMS is a boon
for your developers**



**But consider the impacts
on your clients and marketers**



**Structured content and APIs
will help bridge the gap**



**But we may need solutions for
features marketers rely on daily**



**Coming conflicts will involve
CMSs and emerging front ends**



**Consider your architecture's
ramifications for all users**

Thank you!

Preston So is an innovation lead, researcher, speaker, and author of *Decoupled Drupal in Practice* (Apress, 2018).

A globally recognized voice on decoupled Drupal and subject matter expert in the decentralized web and conversational design, Preston is Principal Product Manager at Gatsby.

Having spoken three times at SXSW Interactive, Preston is a sought-after presenter with keynotes on five continents and in three languages.

[linkedin.com/in/prestonso](https://www.linkedin.com/in/prestonso) • [@prestonso](https://twitter.com/prestonso) • [#NewCMSParadox](https://twitter.com/prestonso)

ありがとうございました！

プレストン・ソーはイノベーションリーダー、研究者、スピーカー、そして「*Decoupled Drupal in Practice* (Apress 2018年)」の著者です。

Decoupled Drupalの世界的なリーダーであり、Decentralized Webや会話形デザインの専門家でもあります。プレストンは、Gatsby Inc.(ギャツビー社)でプリンシパル・プロダクト・マネージャーです。

過去3回にわたり、SXSWインタラクティブで講演し、大陸にて3つの言語で基調講演をおこなう人気スピーカーでもあります。

linkedin.com/in/prestonso • [@prestonso](https://twitter.com/prestonso) • [#NewCMSParadox](https://twitter.com/#NewCMSParadox)



DECOUPLED DAYS 2020

New York City
July 22–23, 2020

Registration opening soon

Visit decoupleddeays.com for more updates about tickets and our call for papers.

Sponsorships

For sponsorship information, reach out to our team at decoupleddeays@gmail.com.

@decoupleddeays

Follow us on Twitter to stay updated on important announcements.

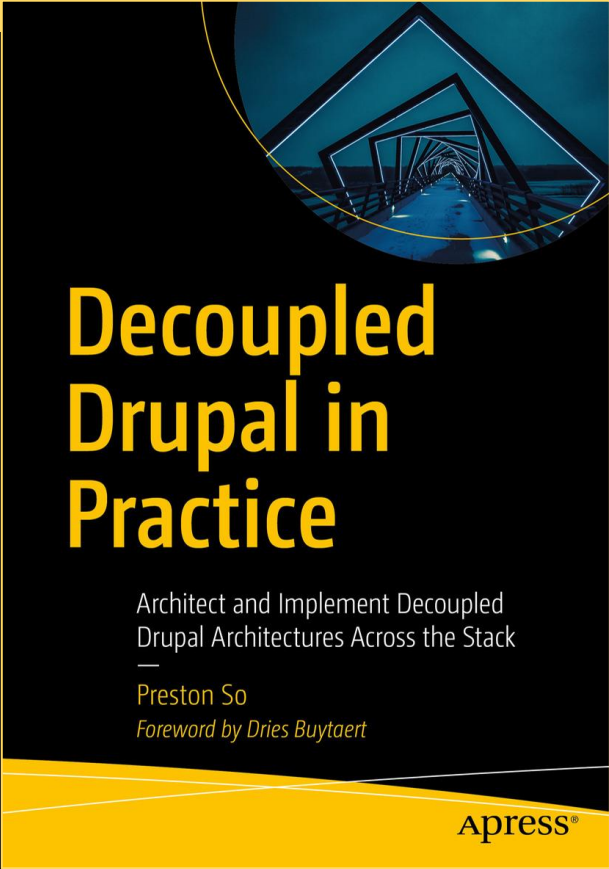
[#decoupleddeays](https://twitter.com/decoupleddeays) • decoupleddeays.com • [@decoupleddeays](https://twitter.com/decoupleddeays)

Decoupled Drupal in Practice

Available now!

tiny.cc/decoupled-in-practice

In the first and only comprehensive guide to decoupled Drupal across the stack, with a foreword by Drupal project lead Dries Buytaert, learn everything you need to know about decoupled Drupal—from building the back end and designing APIs to integrating with front-end technologies.



Decoupled Drupal in Practice

Architect and Implement Decoupled
Drupal Architectures Across the Stack

—
Preston So

Foreword by Dries Buytaert

apress®