
Introduction to decoupled Drupal

Preston So • 26 Sep 2017 • DrupalCon Vienna 2017

Herzlich Willkommen!

Preston So has been a web developer and designer since 2001, a creative professional since 2004, and a Drupal developer since 2007. As Director of Research and Innovation at Acquia, Preston leads new open-source and research initiatives and helms Acquia Labs, the innovation lab featured in international press.

Previously, Preston contributed to the Spark initiative in the Drupal project, co-founded the Southern Colorado Drupal User Group (est. 2008), and operated an award-winning freelance web and print design studio. Preston has presented keynotes at conferences on three continents in multiple languages.

preston.so@acquia.com • [@prestonso](https://twitter.com/prestonso) • [#introdecoupledDrupal](https://twitter.com/introdecoupledDrupal)

What we'll cover

- What is decoupled Drupal?
- Motivating decoupled Drupal
- Decoupling the front end from the back end
- Risks and rewards of decoupling Drupal
- Web services and REST APIs
- Decoupled Drupal and developer experience
- Decoupled Drupal and editorial experience
- Epilogue: Drupal's new incongruity

1 What is decoupled Drupal?



A RILEY SCOTT FILM

BRING
HIM
HOME

MATT DAMON

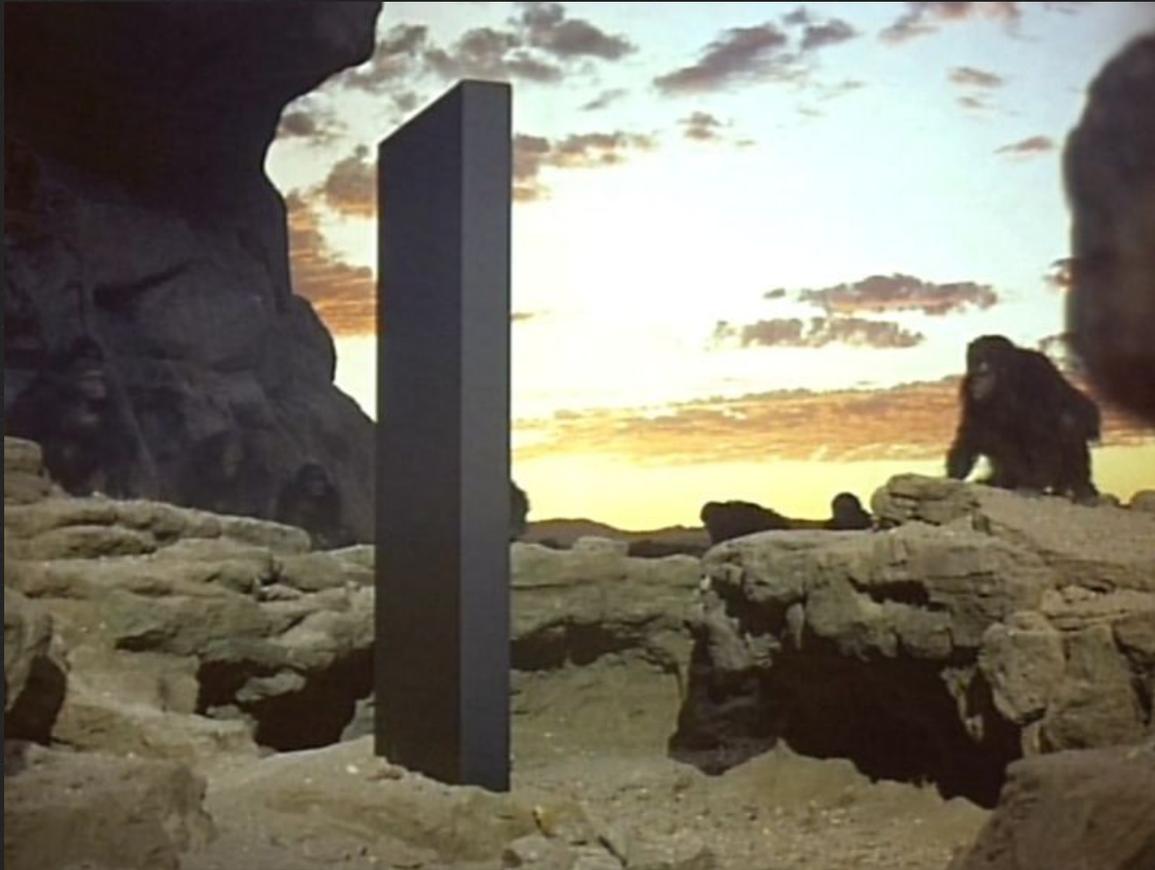
THE
MARTIAN

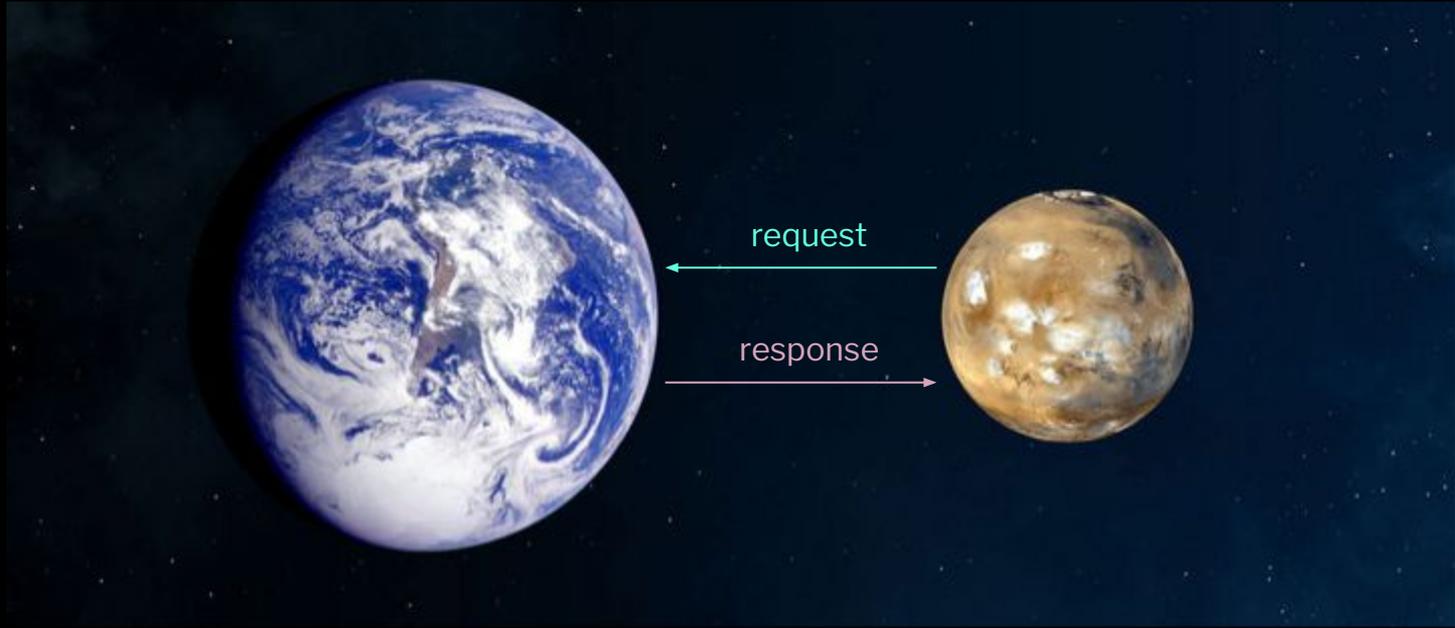
PRODUCED BY HARRY GREESON-WILLIAMS AND SIMON KINBERG. WRITTEN BY RILEY SCOTT. DIRECTED BY RILEY SCOTT. CASTING BY MICHELLE SCHAEFER. COSTUME DESIGNER ADRIAN SCOTT. MARK PUFFAM. EXECUTIVE PRODUCERS RANDY WEIR AND DREW GOODARD. PRODUCED BY RILEY SCOTT

IN CINEMAS SEPTEMBER 30 IN 3D

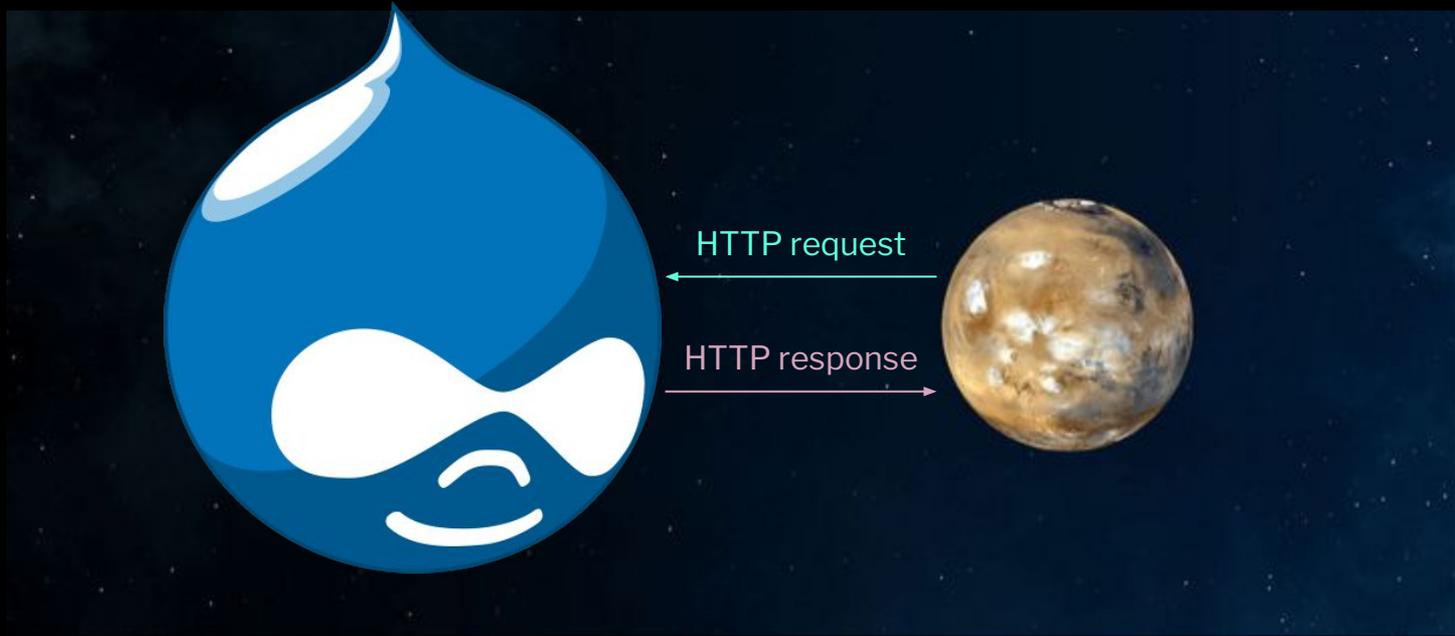


Think of Earth as one contiguous unit: a **monolith**.

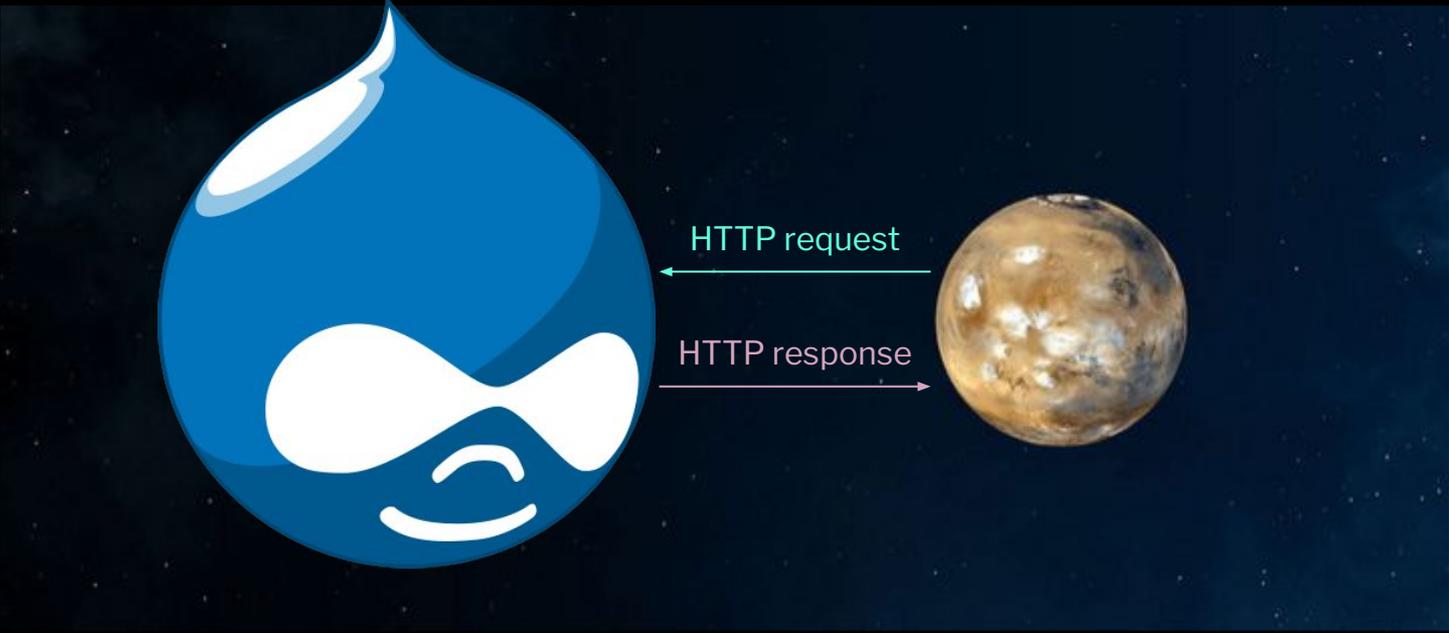




Earth responds to requests from **decoupled** bases.



Drupal is a **monolithic** CMS backing **decoupled** applications.



Site or repository
built in Drupal

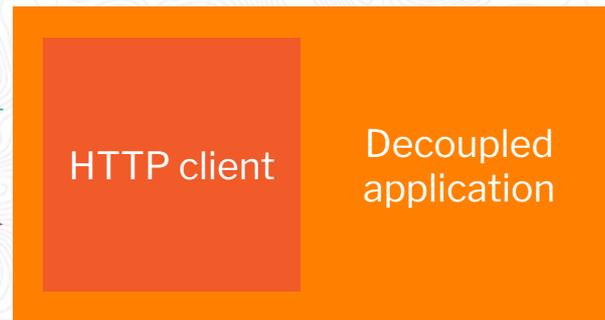
Decoupled
application



Site or repository built in
Drupal

← HTTP request

→ HTTP response

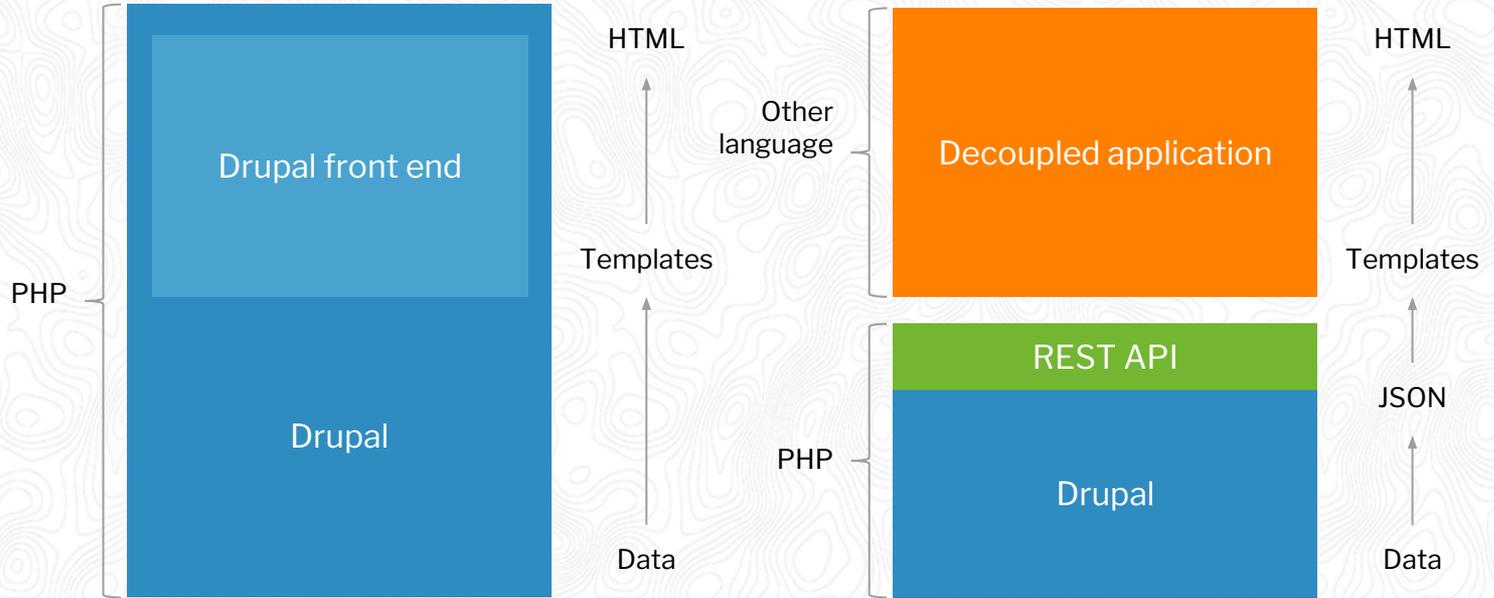


Decoupled
application

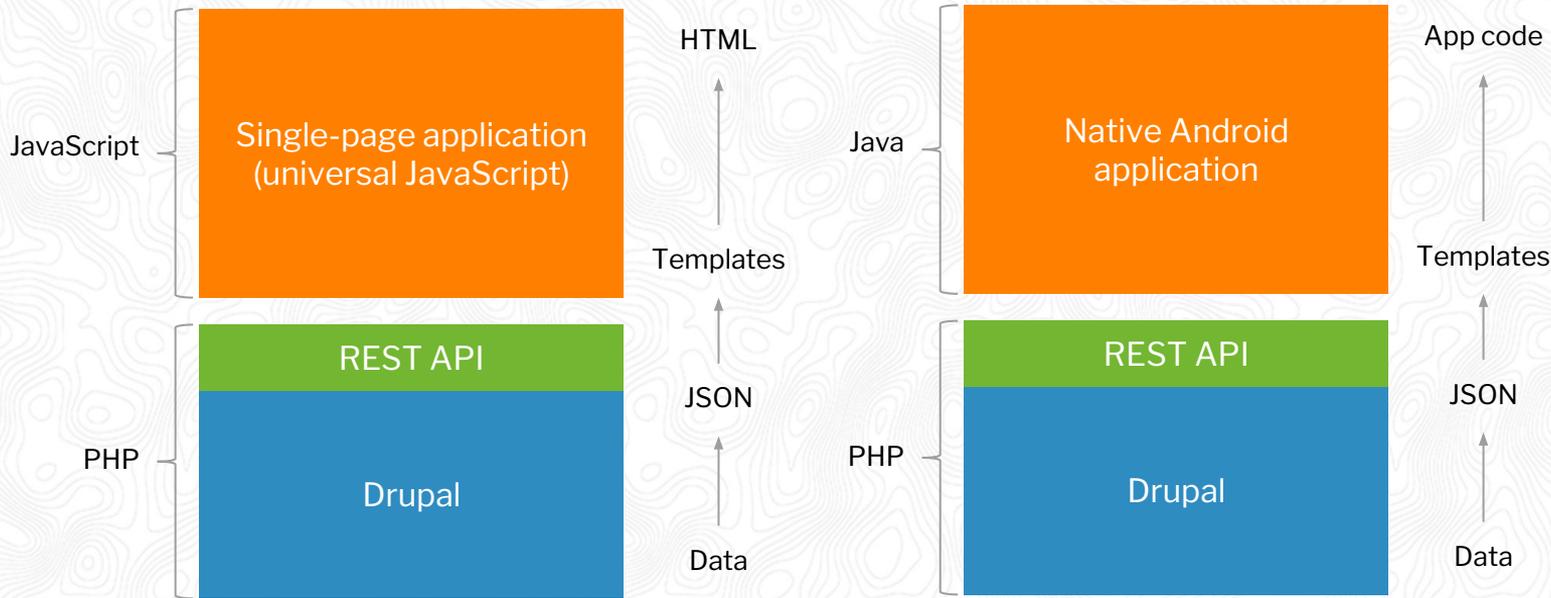
Decoupled Drupal
is the use of Drupal as a content
service for consumption and
manipulation by other applications

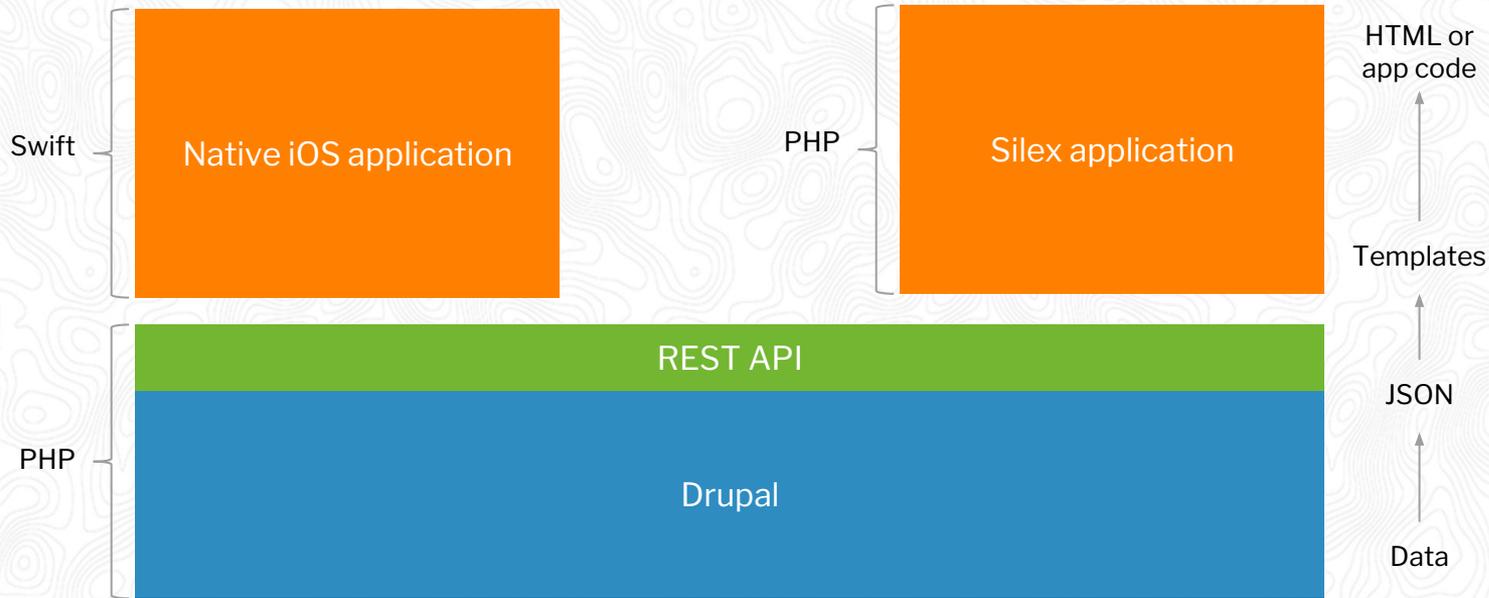
Decoupled Drupal

- Drupal can be used to back other back-end applications as well as native applications, single-page applications, digital signage, AR-driven interfaces, and many others.
- Web services such as the core REST API, JSON API, and GraphQL are typical entry points for Drupal consumers.



Monolithic vs. decoupled Drupal





de•coup•led

head•less

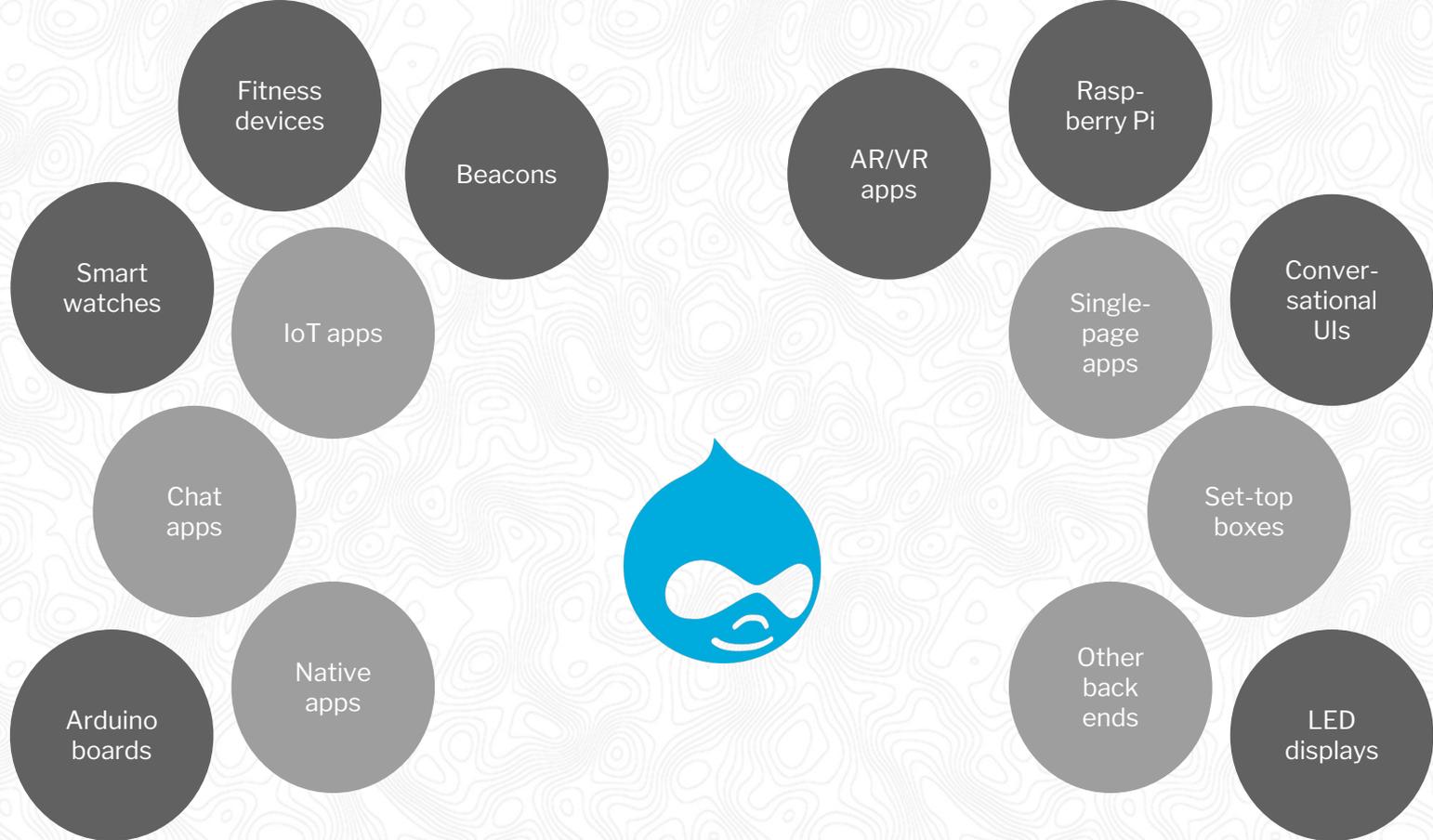
API•first

Aside: What's in a name?

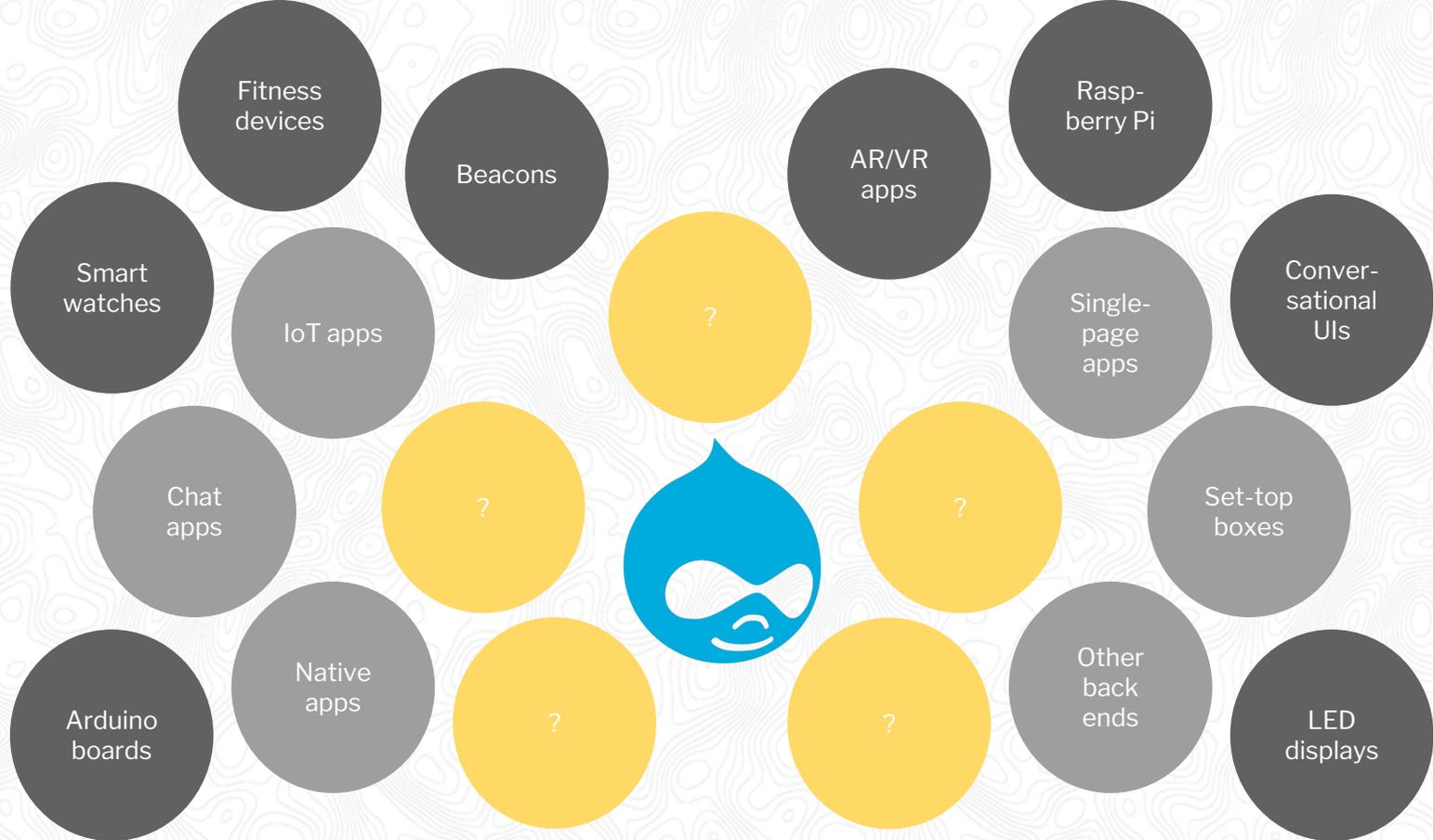
2

Motivating decoupled Drupal

**For marketers and publishers
decoupled Drupal offers the promise
of omnichannel delivery**



We have new channels every day.

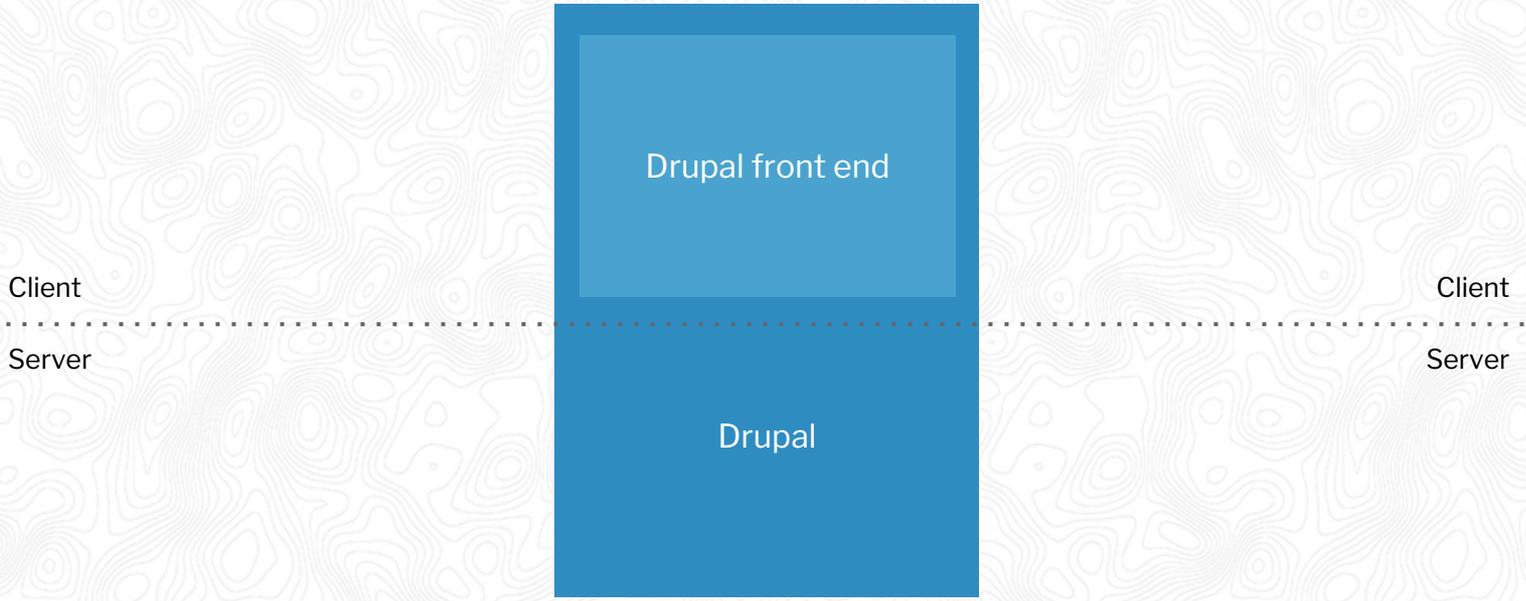


And ones we've never heard of yet.

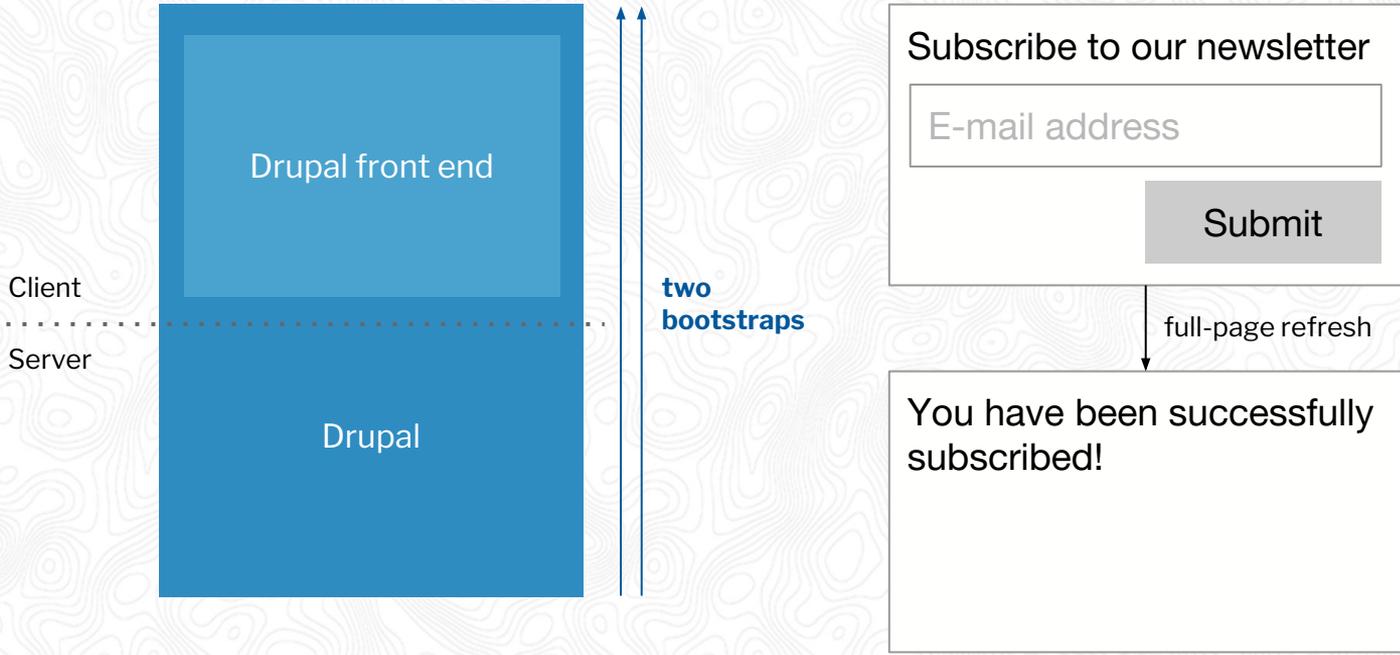
For developers
decoupled Drupal offers the promise
of front-end freedom

3

**Decoupling the front
end from the back end**

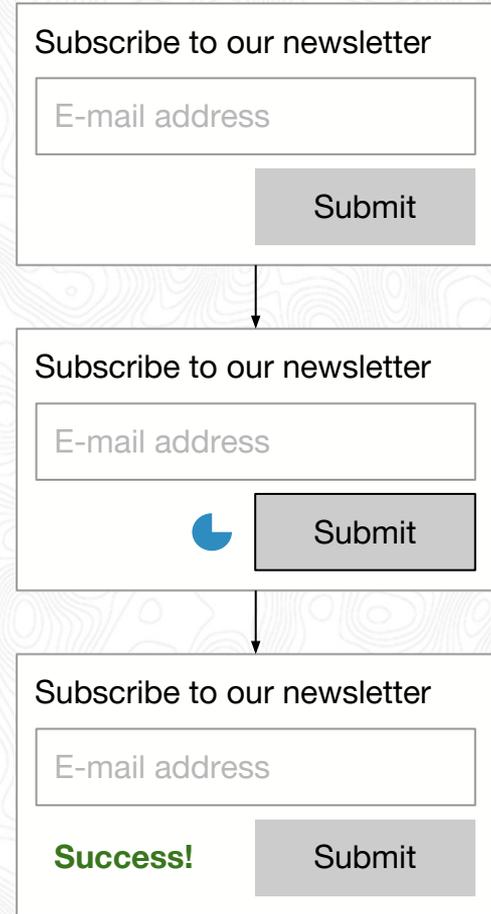
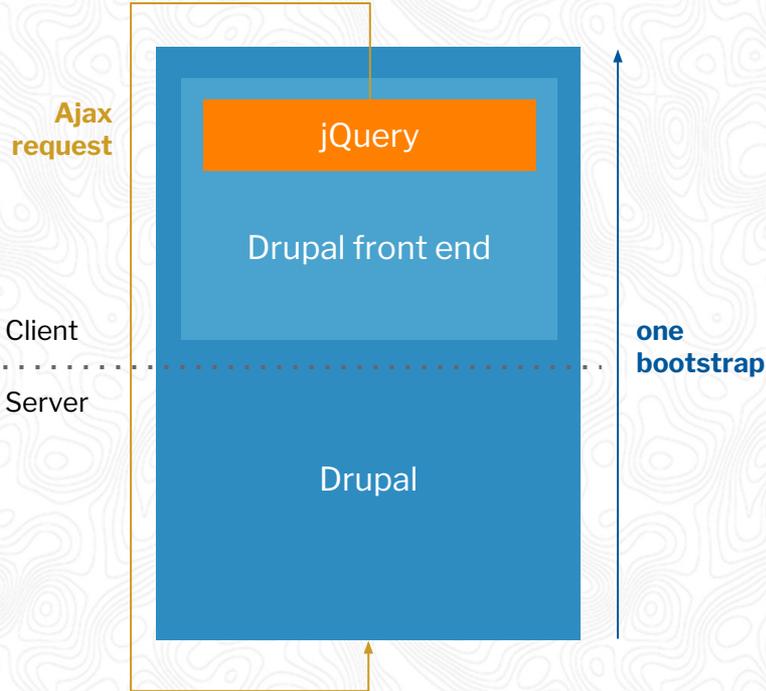


Monolithic Drupal

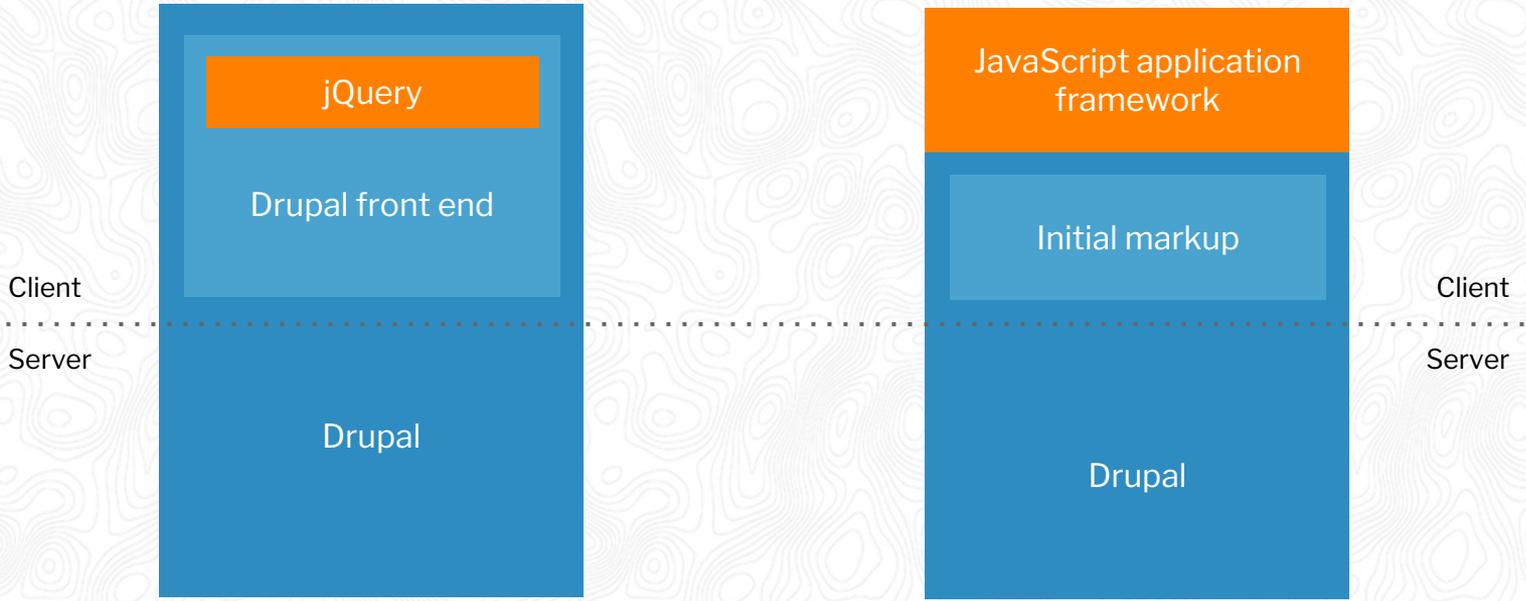


Static pages

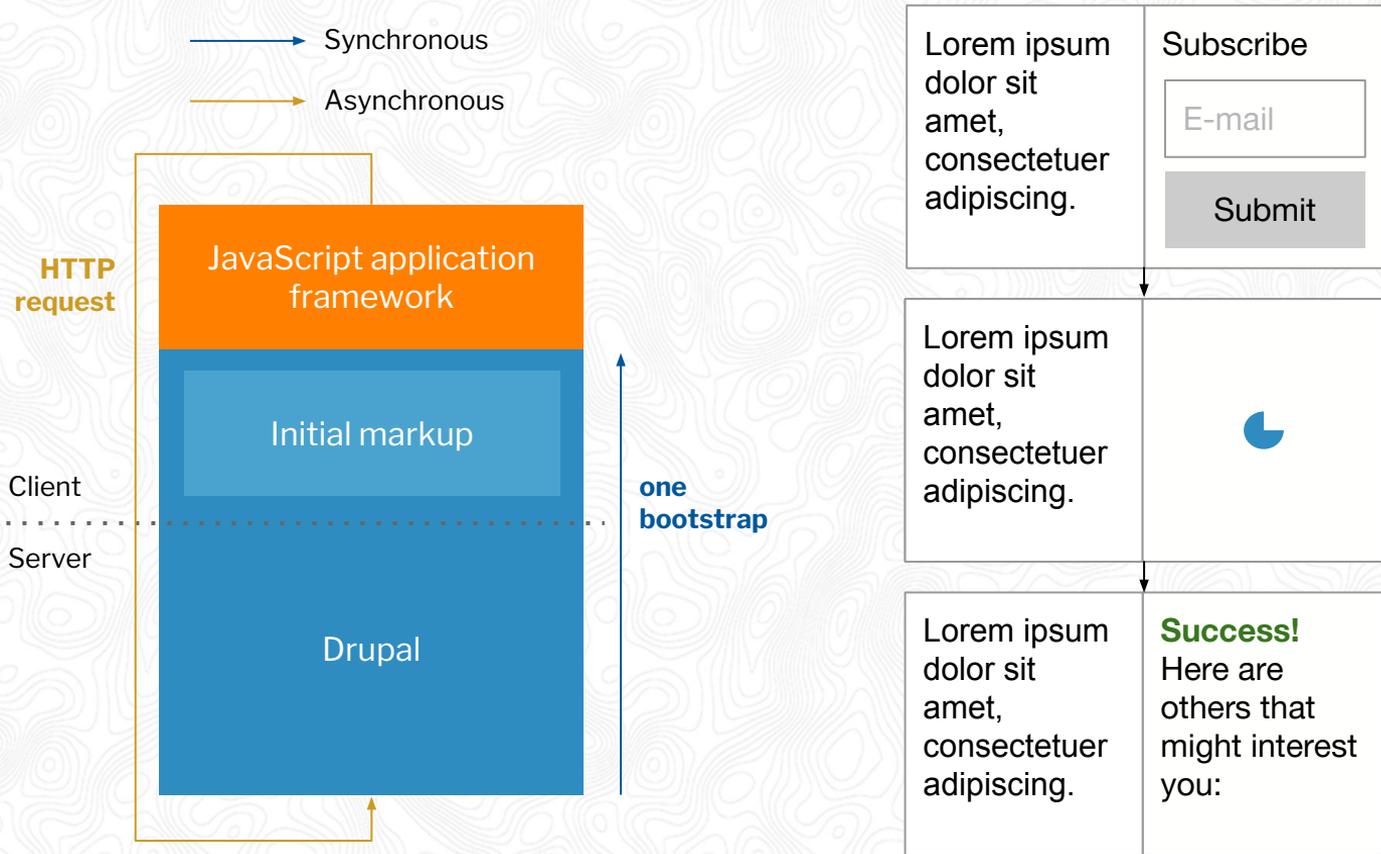
- Synchronous
- Asynchronous



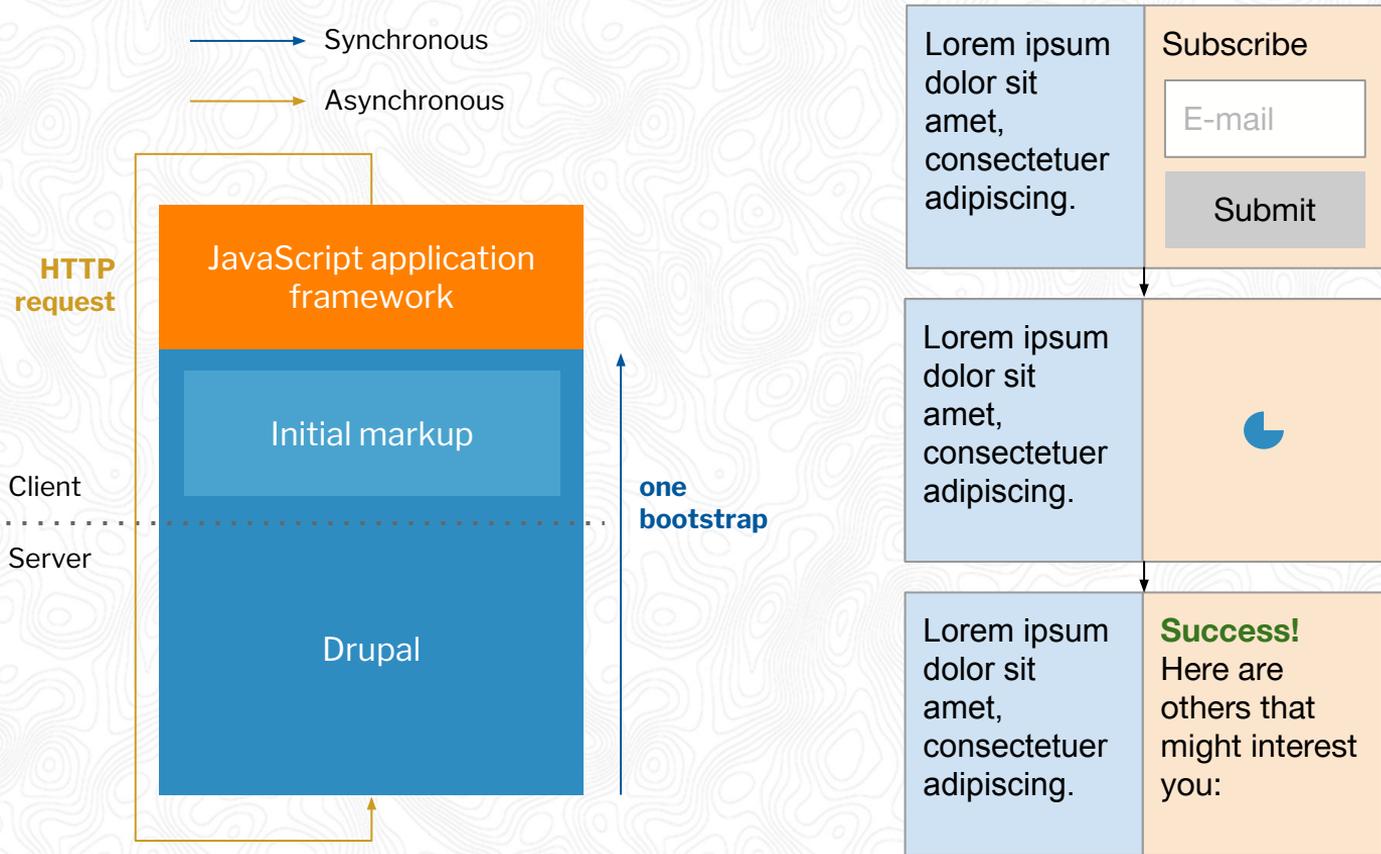
Dynamic pages with Ajax



Client-side JavaScript



Dynamic pages with client-side rendering

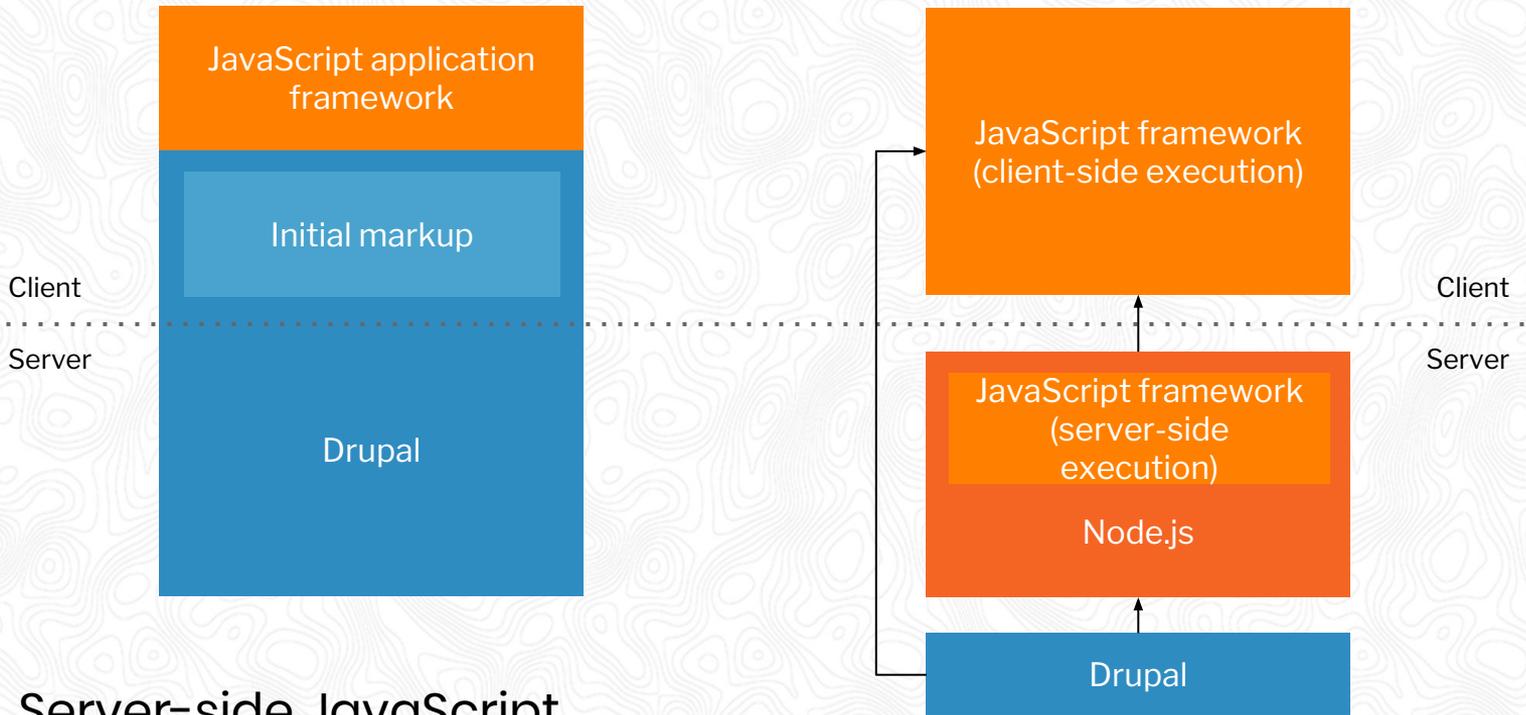


Dynamic pages with client-side rendering

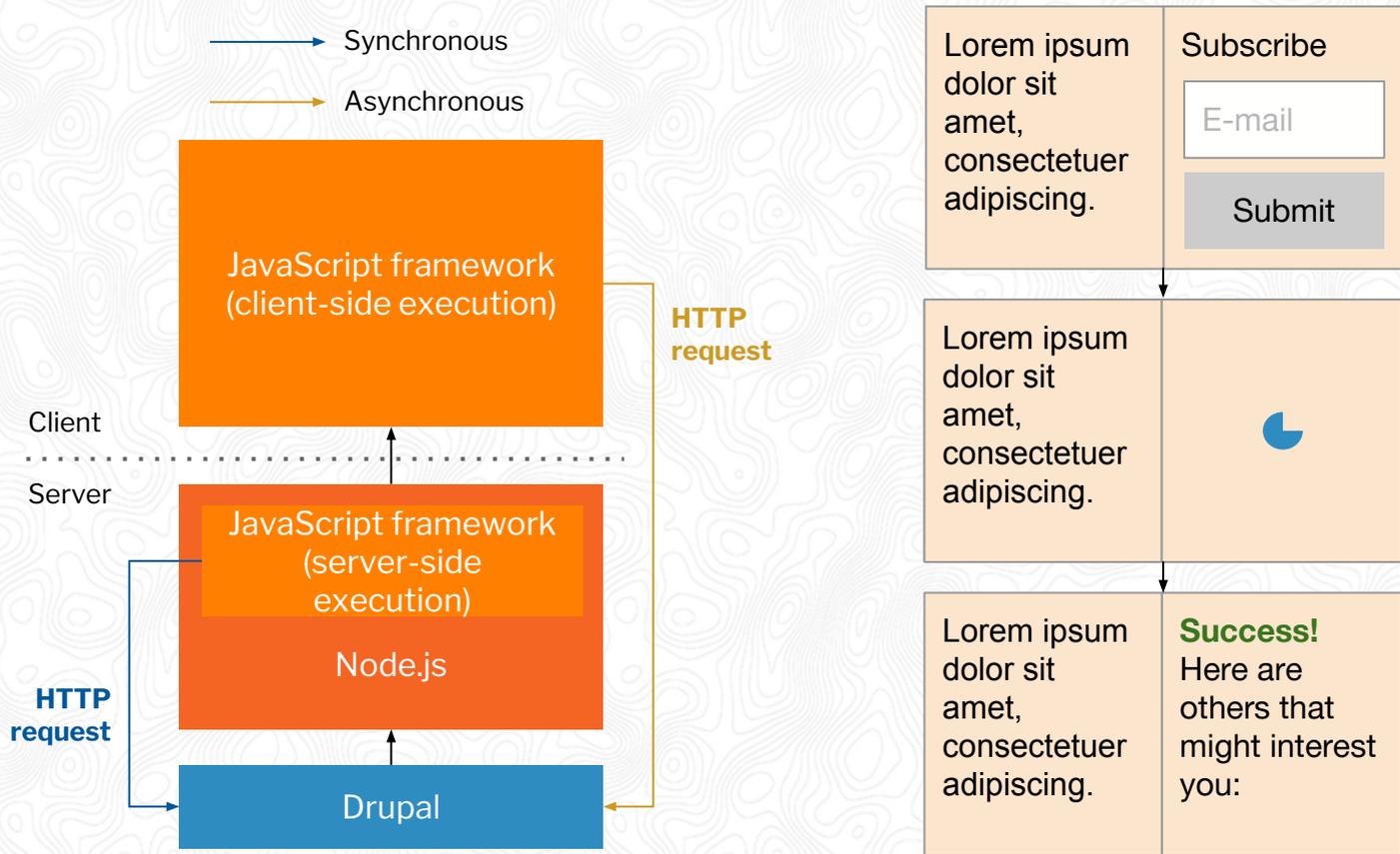
**The JavaScript renaissance
ushered in an era of innovation and
new development practices**

**Server-side JavaScript
meant client-side frameworks could
be executed server-side**

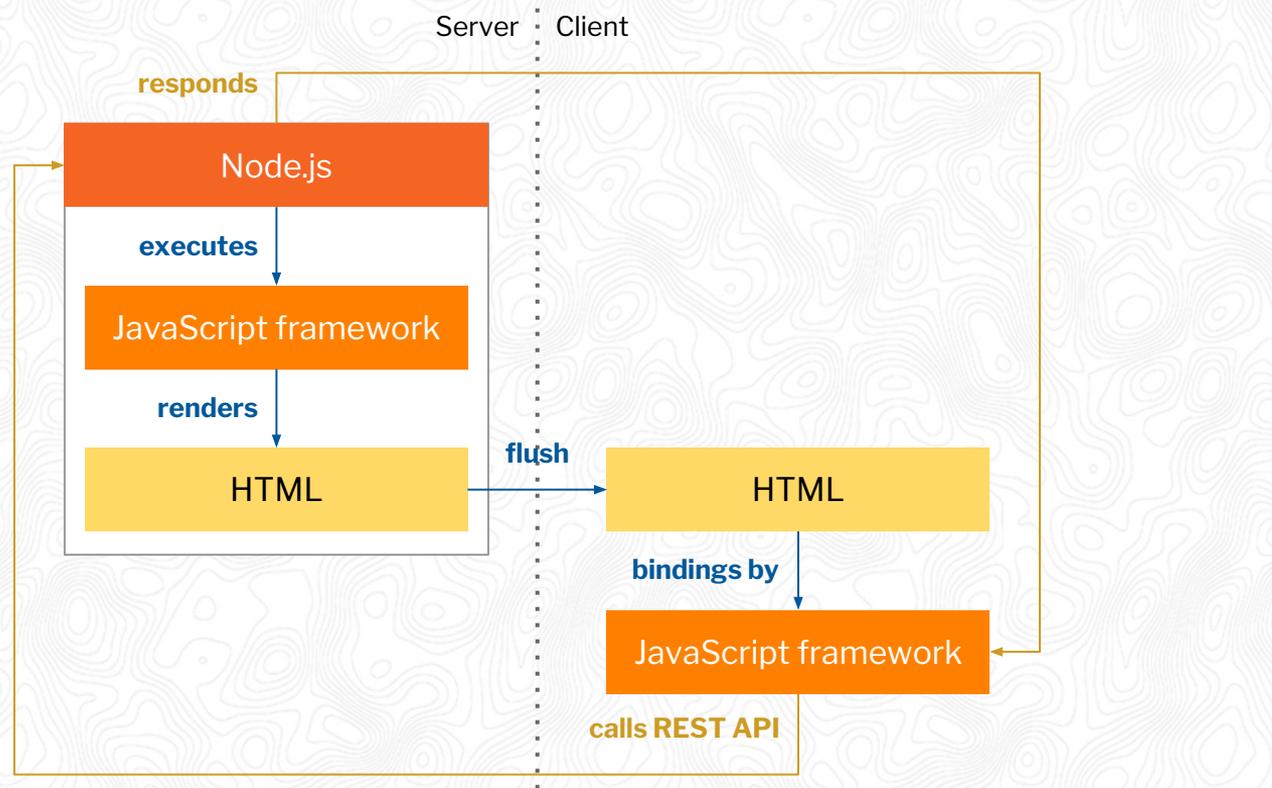
Universal JavaScript
meant shared code in the form of
JavaScript framework executions



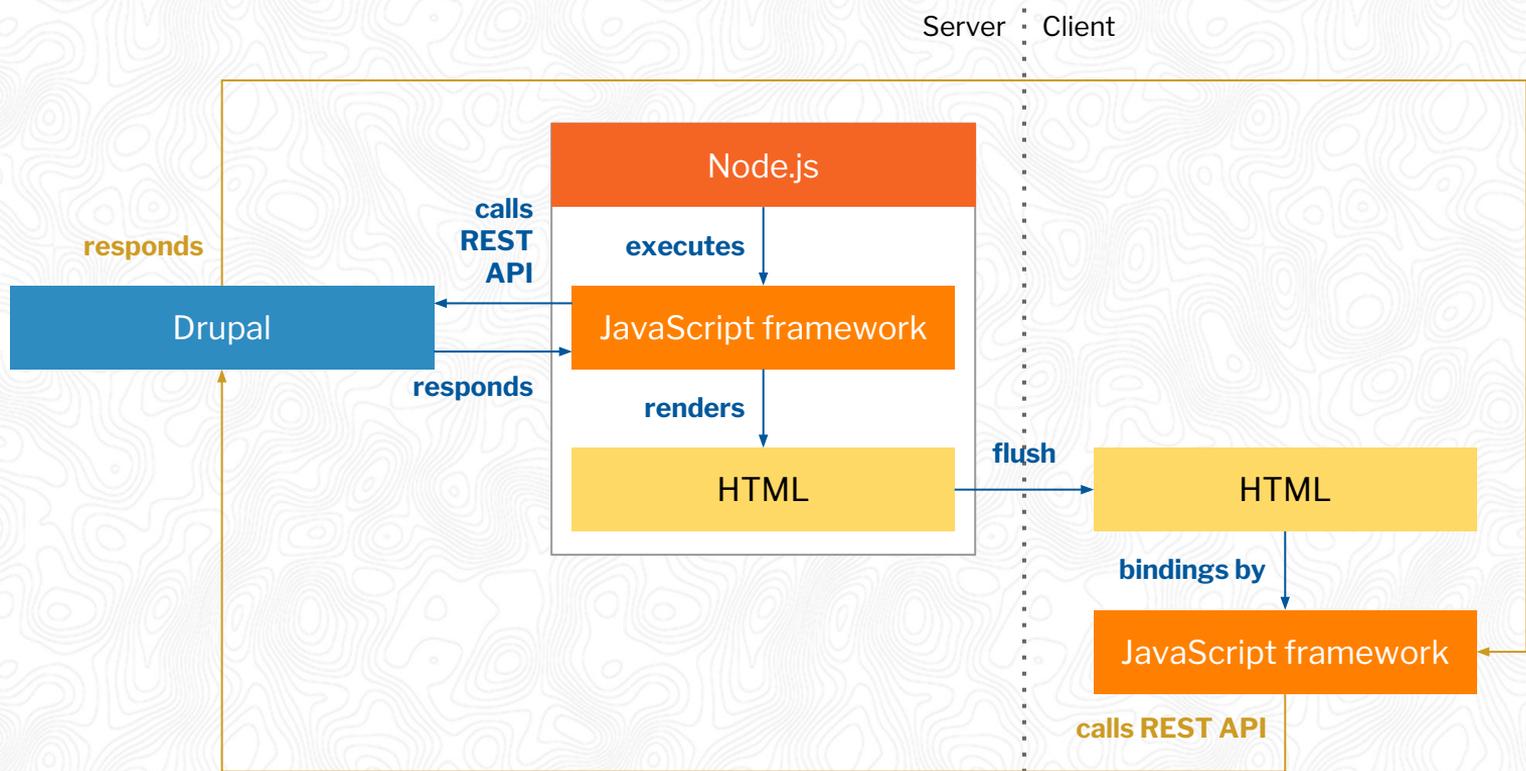
Server-side JavaScript



Dynamic pages with server-side prerendering

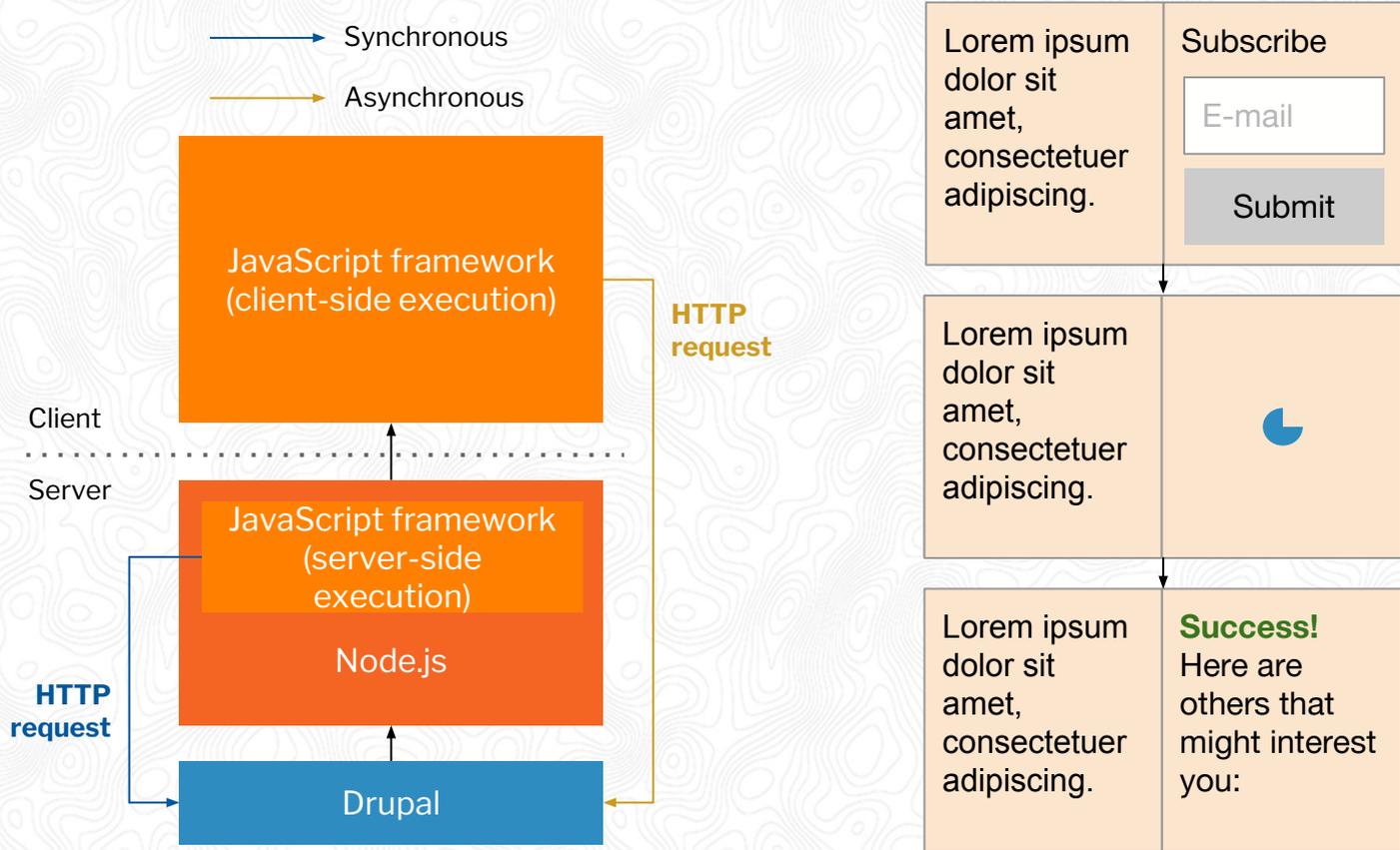


Universal (isomorphic) JavaScript

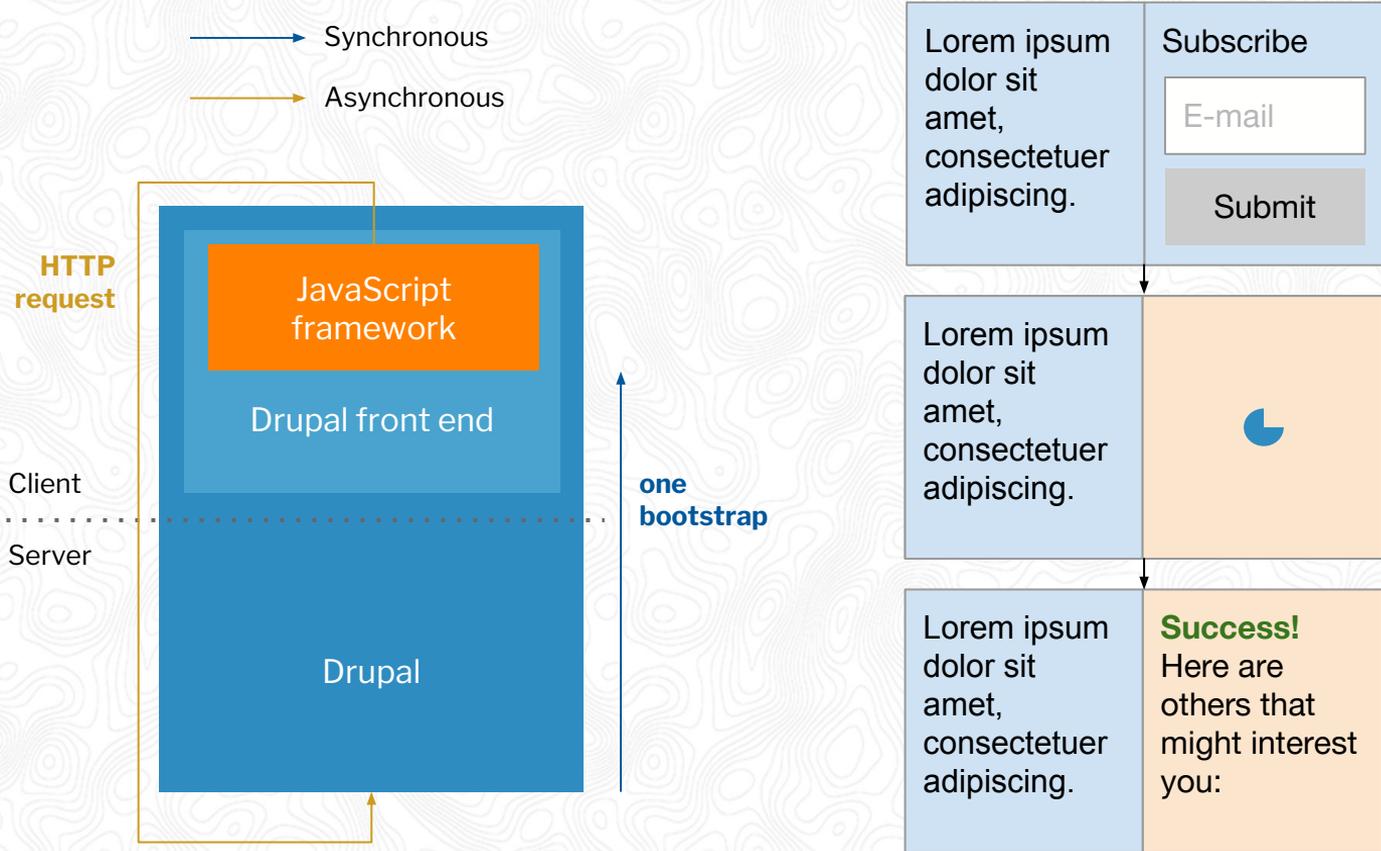


—————> Synchronous
 —————> Asynchronous

Universal JavaScript consuming Drupal



Fully decoupled Drupal



Progressively decoupled Drupal

4

Risks and rewards of decoupled Drupal

Rewards of decoupled Drupal

- **Separation of concerns.** Structure versus presentation is increasingly relevant in decoupled architectures.
- **Pipelined development.** An API-design-first approach gives front-end and back-end teams independence.

Rewards of decoupled Drupal

- **Administrative security.** Keeping the administrative CMS far from public view is increasingly a best practice.
- **Easier resourcing.** Drupal developers are now considered less affordable than JavaScript developers.

Risks of decoupled Drupal

- **Loss of contextual administration.** Crucial features such as in-place editing and toolbar can be key for editors.
- **Maintainability difficulties.** Maintaining both Drupal service and consumers requires extensive planning.

Risks of decoupled Drupal

- **Additional point of failure.** Node.js downtime can lead to inaccessible content in the case of private Drupal.
- **On-boarding overhead.** JavaScript developers need to learn consumption; Drupal developers change focus.

Summary

| Area | Rewards | Risks |
|--------------------------|-------------------------|-----------------------------------|
| Architecture | Separation of concerns | Loss of contextual administration |
| Development experience | Pipelined development | Maintainability difficulties |
| Security and performance | Administrative security | Additional point of failure |
| Project management | Easier resourcing | On-boarding overhead |

5

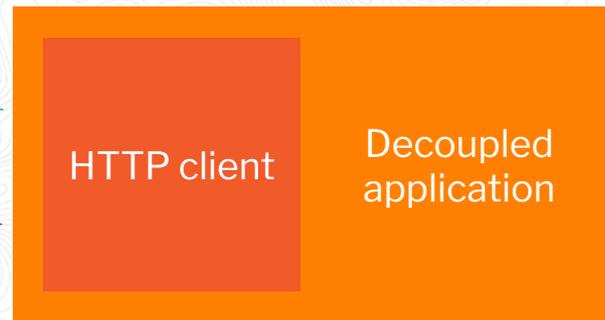
Web services and REST APIs



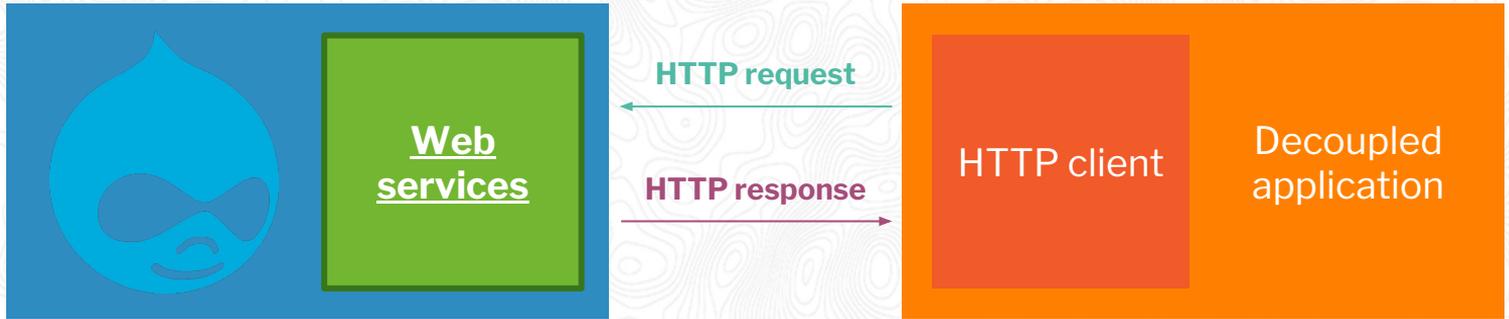
Site or repository built in
Drupal

← HTTP request

→ HTTP response



Decoupled
application

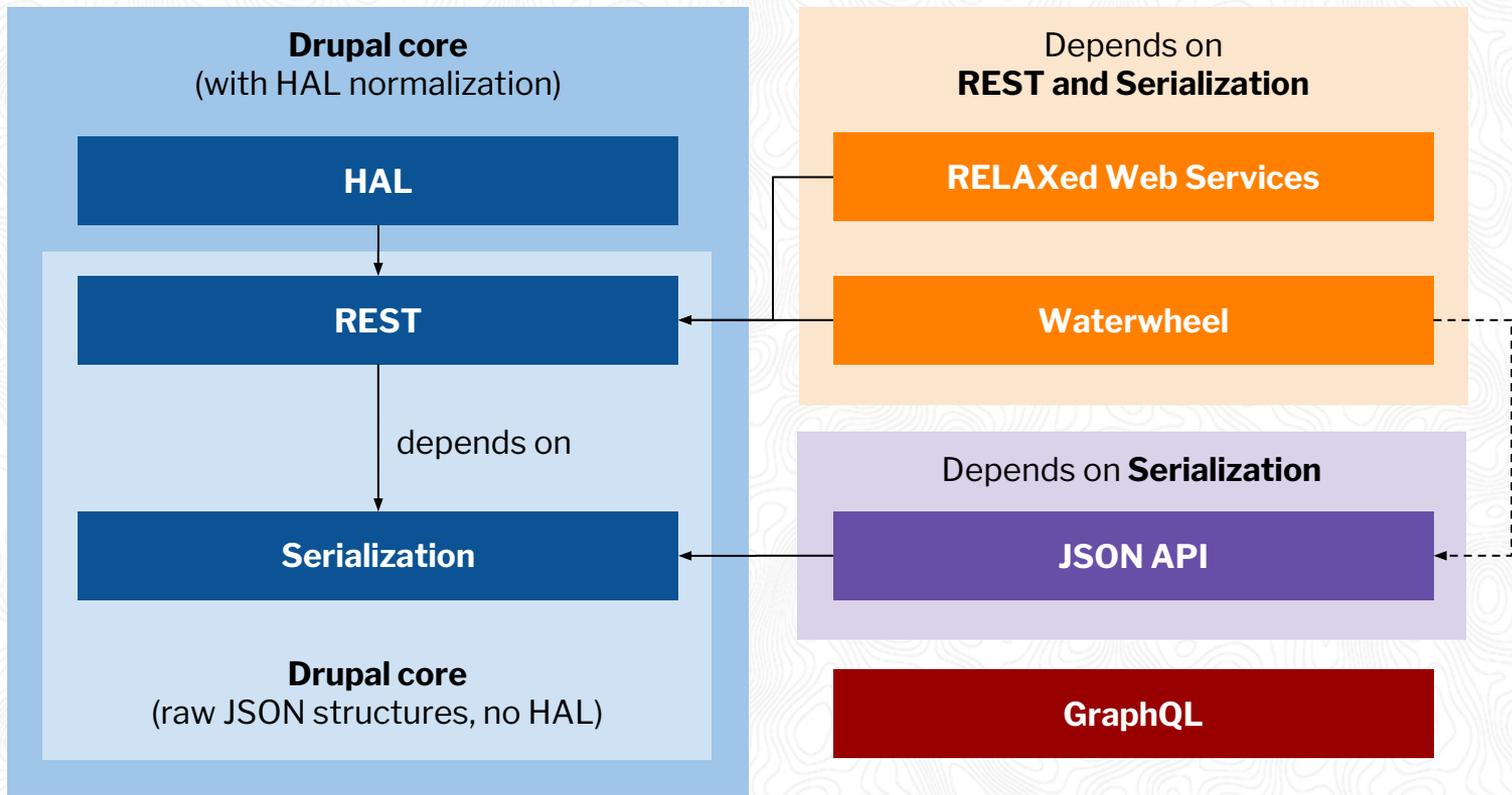


Not all web services are REST APIs

Web services
support interoperable
machine-to-machine interaction over
a network

REST APIs

**follow REST principles and operate
using HTTP methods**



Drupal's web services ecosystem (not exhaustive)



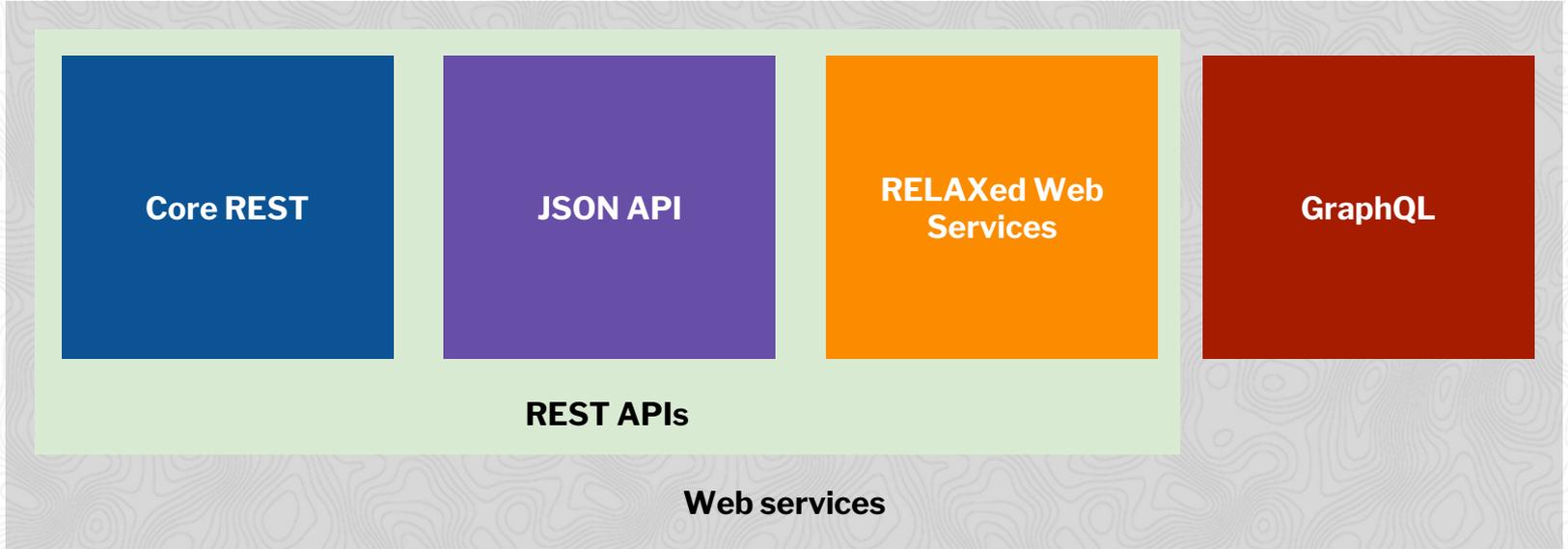
Core REST

JSON API

**RELAXed Web
Services**

GraphQL

Core and contrib web services modules



Euler diagram of Drupal web services

Core REST

The core REST modules allow for all content entities (nodes, users, taxonomy terms, comments) to be exposed as JSON+HAL or as JSON representing Drupal's internal storage, and Views natively supports "REST export" as a new display type.

Current core REST
(abbreviated
payloads)

Client issues request for an entity
resource

```
GET /node/1?_format=json
```

```
GET /user/5?_format=json
```

Server issues response with single
entity resource

```
"field_author": [  
  {  
    "value": "5"  
  }  
]
```

```
"name": [  
  {  
    "value": "Joe"  
  }  
]
```

Core REST

JSON API

- JSON API is a specification for REST APIs in JSON popular among JavaScript developers and adopted by the Ember and Rails communities.
- JSON API provides a standard way to query single entities, but it also provides all relationships contained therein and query operations via query string parameters.

JSON API

- JSON API allows you to fetch lists of content entities (filter, sort, pagination) – which is currently only possible via multiple requests or Views REST exports.
- drupal.org/project/jsonapi

JSON API
(abbreviated payloads,
merely illustrative)

Client issues request for an entity
resource

```
GET /jsonapi/node/1  
?include=field_author
```

Server issues response with single
entity and all relationships

```
{  
  "data": [{  
    "type": "articles",  
    "id": "1",  
    "attributes": {  
      "title": "Foo",  
      "body": "Lorem ipsum",  
    },  
    ...  
  }],  
  "included": [{  
    "type": "people",  
    "id": "5",  
    "attributes": {  
      "name": "Joe"  
    }  
  }]  
}
```

GraphQL

- GraphQL, originally created by Facebook to power its data fetching, is a query language that enables fewer queries and limits response bloat.
- Rather than tightly coupling responses with a predefined schema, GraphQL overturns this common practice by allowing for the client's request to explicitly tailor a response so that the client only receives what it needs.

GraphQL

- GraphQL shifts responsibility from the server to the client: the server publishes its possibilities, and the client publishes its requirements instead of receiving a response dictated solely by the server.
- drupal.org/project/graphql

GraphQL
(simplified payloads,
merely illustrative)

Client issues request declaring
data requirements

```
{  
  title  
  fieldAuthor {  
    title  
  }  
}
```

Server issues response matching
structure of declared requirements

```
{  
  "title": "Foo"  
  "fieldAuthor": {  
    "title": "Joe"  
  }  
}
```

GraphQL

RELAXed Web Services

- RELAXed Web Services includes support for translations, parent revisions (via Multiversion), file attachments, and cross-environment UUID references.
- It uses the CouchDB API specification, which means that CouchDB integration with client-side libraries such as PouchDB and Hoodie makes possible offline-enabled Drupal.
- drupal.org/project/relaxed

6

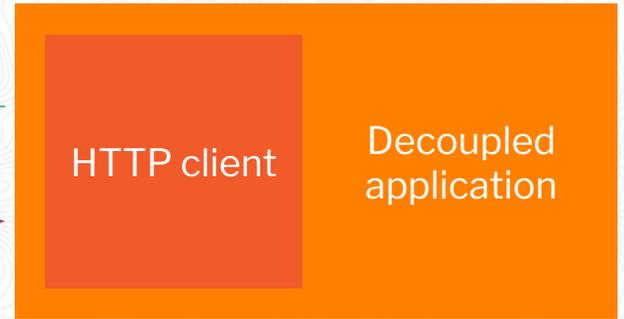
Decoupled Drupal and developer experience



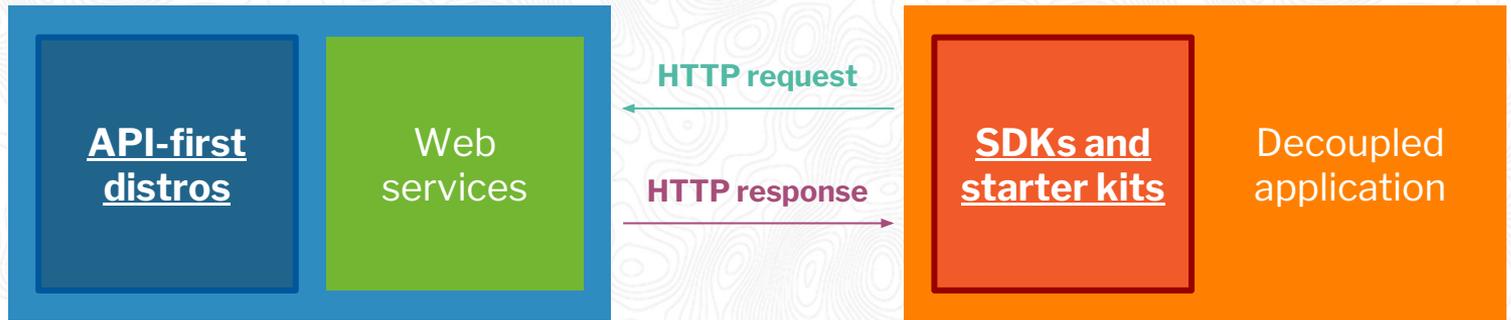
Site or repository built in
Drupal

← HTTP request

→ HTTP response



Decoupled
application



API-first distros, SDKs, and starter kits improve decoupled DX

SDKs and starter kits

- **SDKs** can help to make it easier for developers to consume Drupal as a content service. These tend to be lightweight HTTP client wrappers which make requests on the developer's behalf and process responses.
- **Starter kits** and **references** help jumpstart implementations by providing an out-of-the-box experience for developers to get started right away.



Waterwheel

The Waterwheel ecosystem

The Waterwheel ecosystem

- **Waterwheel** is an emerging ecosystem of SDKs designed to accelerate development of Drupal-backed applications in a variety of languages.
- Currently, a helper SDK is available for **JavaScript** as an initial release and a full SDK is available for **Swift** to power iOS, tvOS, and other Apple applications.

The Waterwheel ecosystem

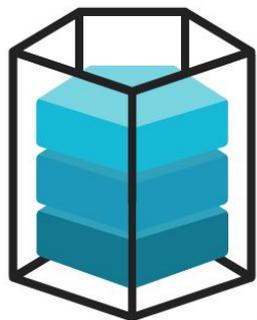
- Waterwheel is intended to be a “translator” for developers in other technologies: in other words, Waterwheel helps non-Drupalists “speak” Drupal.

The Waterwheel ecosystem

- github.com/acquia/waterwheel.js
- github.com/kylebrowning/waterwheel.swift
- drupal.org/project/waterwheel
- github.com/acquia/ember-drupal-waterwheel
- github.com/acquia/ember-waterwheel-app
- github.com/acquia/react-waterwheel-app

API-first distributions

- An ecosystem of distributions oriented for developers consuming decoupled Drupal is emerging. The most prominent distributions are **Reservoir** and **Contenta**.
- github.com/acquia/reservoir
- github.com/contentacms/contenta_jsonapi



reservoir



Reservoir and Contenta

**A good developer experience
means that it's easy to provision
APIs for a group of consumer
applications**

**A good developer experience
means that it's easy for consumers to
work with the provisioned APIs**

7

Decoupled Drupal and editorial experience

In-place editing

One of Drupal 8's core features is in-place editing. In short, you can edit fields in context.

Title

Lorem ipsum dolor sit a|

Body

Consectetur adipiscing elit
lorem ipsum dolor sit amet.

Lorem ipsum
dolor sit a|

Consectetur adipiscing
elit lorem ipsum dolor sit
amet.

[Read more](#)

In-place editing

On websites and responsive sites, in-place editing can still make perfect sense.

Title

Lorem ipsum dolor sit a|

Body

Consectetur adipiscing elit
lorem ipsum dolor sit amet.

Websites
and responsive
sites can both
leverage
in-place editing

Lorem ipsum
dolor sit a|

Consectetur adipiscing
elit lorem ipsum dolor sit
amet.

[Read more](#)

In-place editing

Digital signage could still be possible if and only if Drupal renders it as well rather than exporting data.

Slogan

Mehr Pizza ess|

Website URL

www.drupizza.at

Digital signage
could leverage
in-place editing
if Drupal is
responsible
for rendering



In-place editing

But what about single-page applications or zero interfaces that can't be controlled by Drupal?

Title

Lorem ipsum dolor sit a|

Body

Consectetur adipiscing elit
lorem ipsum dolor sit amet.

Content
edited solely
in Drupal's
administration
layer

Lorem ipsum
dolor sit a|

Consectetur adipiscing
elit lorem ipsum dolor sit
amet

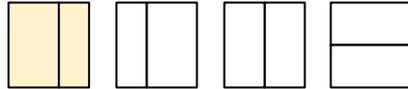
[Read more](#)



Display and layout management

What about choosing layouts and managing displays of content?

Choose a layout



Consectetur
adipiscing elit
lorem ipsum
dolor sit amet.

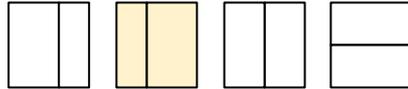
[Read more](#)

Brand
About
Work
Team
Blog

Display and layout management

What about choosing layouts and managing displays of content?

Choose a layout



Brand
About
Work
Team
Blog

Consectetur
adipiscing elit
lorem ipsum
dolor sit amet.

[Read more](#)

Display and layout management

What about toggling between teaser and full content display modes in a decoupled context?

Brand
About
Work
Team
Blog

Consectetur
adipiscing elit
lorem ipsum ...
[Read more](#)

Lorem ipsum
dolor sit amet
consectetur ...
[Read more](#)

Teaser

Brand
About
Work
Team
Blog

Consectetur
adipiscing elit
lorem ipsum
dolor sit amet.
Ipsum amet sit
adipiscing lorem
consectetur elit.
Elit adipiscing
amet lorem.

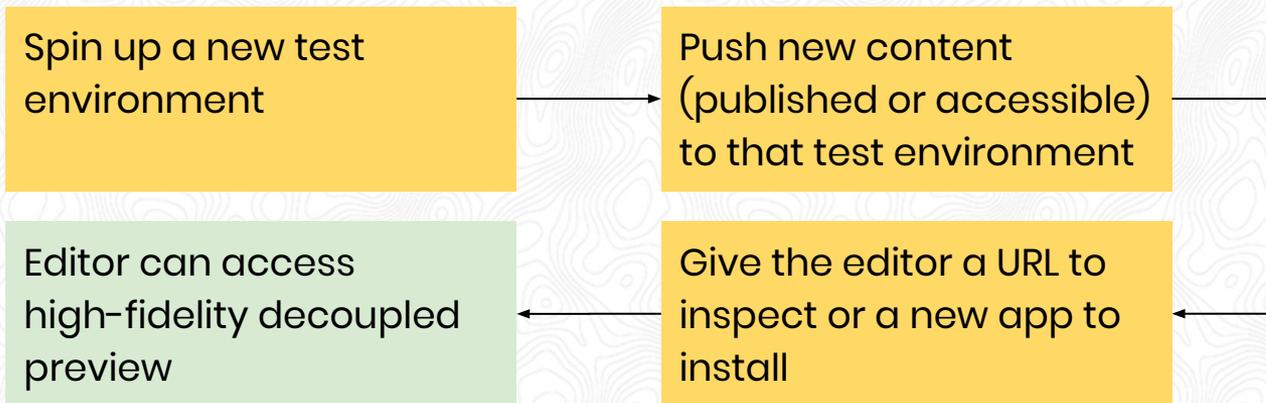
Full content

Editorial preview

- What about accessing or previewing unpublished or draft content?
- Currently, previewing a single piece of content on a single Drupal site (workspaces soon!) is trivial for editors. No developer needed!
- In a decoupled setting, content can only be previewed directly on the device or through an emulator, which may result in lower fidelity.

Editorial preview

If an editor wants a high-fidelity preview of content on their single-page application or native mobile application, developers are required, for now.



Decoupled Drupal
means we need to think about
content in a more structured way

Decoupled Drupal
means we may need to accept that
full visual control is impossible

Decoupled Drupal

**means many capabilities we are used
to need to be exposed to consumers**

8

Epilogue

Drupal's new incongruity

Drupal's new incongruity

Drupal's value proposition is becoming incongruous between the three personas: the user, the marketer, and the developer.

Is this irreconcilable?

Wearables

Conversational

Digital signage

Augmented reality

Mobile

Set-top boxes

Websites

Responsive

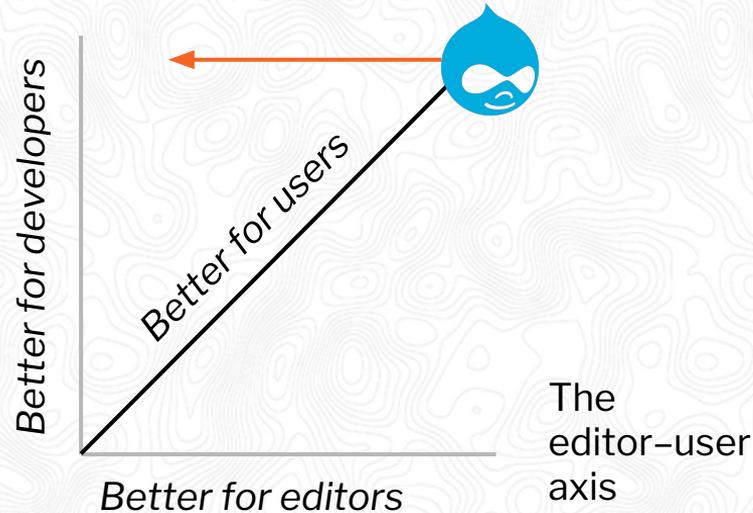
Developer experience

Marketer experience

Drupal's new incongruity

Drupal's new incongruity

The
developer-
user axis

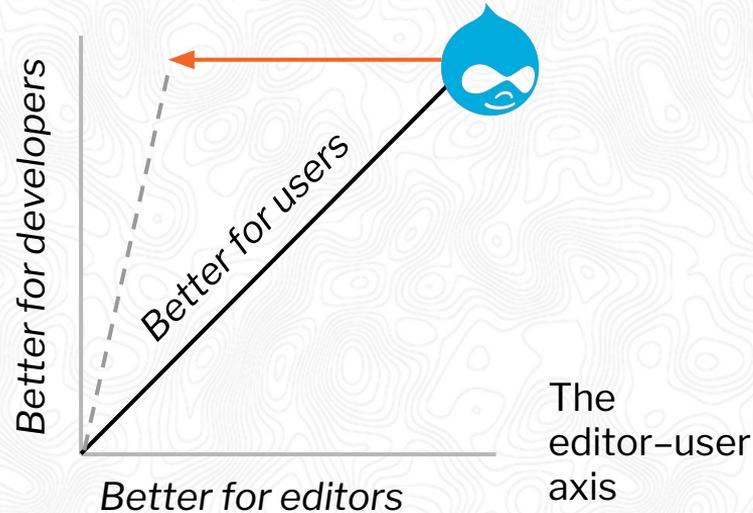


The
editor-user
axis

"Better for users" increasingly means "better on more devices."

Drupal's new incongruity

The
developer-
user axis



A better outcome for users relies more on custom work by developers.

Food for thought

- We have to be prepared to accept that the ideal future state of Drupal may not be anything close to what it is today.
- We have to be prepared to accept that Drupal may be made up of many disparate parts, each essential to a larger whole.

**This is the great test that will dictate
the next decade of Drupal**



**A multifaceted Drupal is
a more future-proof Drupal**



**Decoupled Drupal means
Drupal's story has only just begun**

More decoupled Drupal at Vienna

- **Breaking Drupal out of the box with AR and beacons** (with hampercm)
Wed, 27 Sep — 15:45-16:45 — Strauss
- **Decoupled site building: Drupal's next challenge**
Thu, 28 Sep — 10:45-11:45 — Strauss
- **Decoupled Drupal and editorial bliss** (mrjmd)
Thu, 28 Sep — 14:15-15:15 — Lehar 1

Join us for contribution sprint

- **Mentored Core Sprint**
Fri, 29 Sep — 09:00–18:00 — Stolz 2
- **First-time Sprinter Workshop**
Fri, 29 Sep — 09:00–12:00 — Lehar 1, Lehar 2
- **General Sprint**
Fri, 29 Sep — 09:00–18:00 — Mall

#drupalsprints

What did you think?

- **Evaluate this session**
events.drupal.org/vienna2017/sessions/breaking-drupal
- **Take the survey!**
surveymonkey.com/r/drupalconvienna

Vielen Dank! • Thank you!

Preston So has been a web developer and designer since 2001, a creative professional since 2004, and a Drupal developer since 2007. As Director of Research and Innovation at Acquia, Preston leads new open-source and research initiatives and helms Acquia Labs, the innovation lab featured in international press.

Previously, Preston contributed to the Spark initiative in the Drupal project, co-founded the Southern Colorado Drupal User Group (est. 2008), and operated an award-winning freelance web and print design studio. Preston has presented keynotes at conferences on three continents in multiple languages.

preston.so@acquia.com • [@prestonso](https://twitter.com/prestonso) • [#introdecoupledDrupal](https://twitter.com/introdecoupledDrupal)