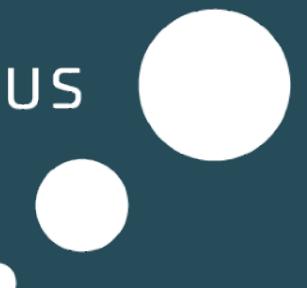


Headless CMS

Yet another Data Source



Headless CMS

Your own data & content available through an
API





What's a Headless CMS?



A CMS without a head 🤔

I'm not kidding

It's a CMS without any frontend



Headless CMS

...is a system to control the content that is separated from the display layer or the front-end user experience.

Headless CMS

... a backend-only content management system that delivers content via APIs, allowing developers to build custom front-ends for web, mobile, or other platforms. Examples: Strapi, Contentful, Sanity, WordPress

What
Is Headless CMS?



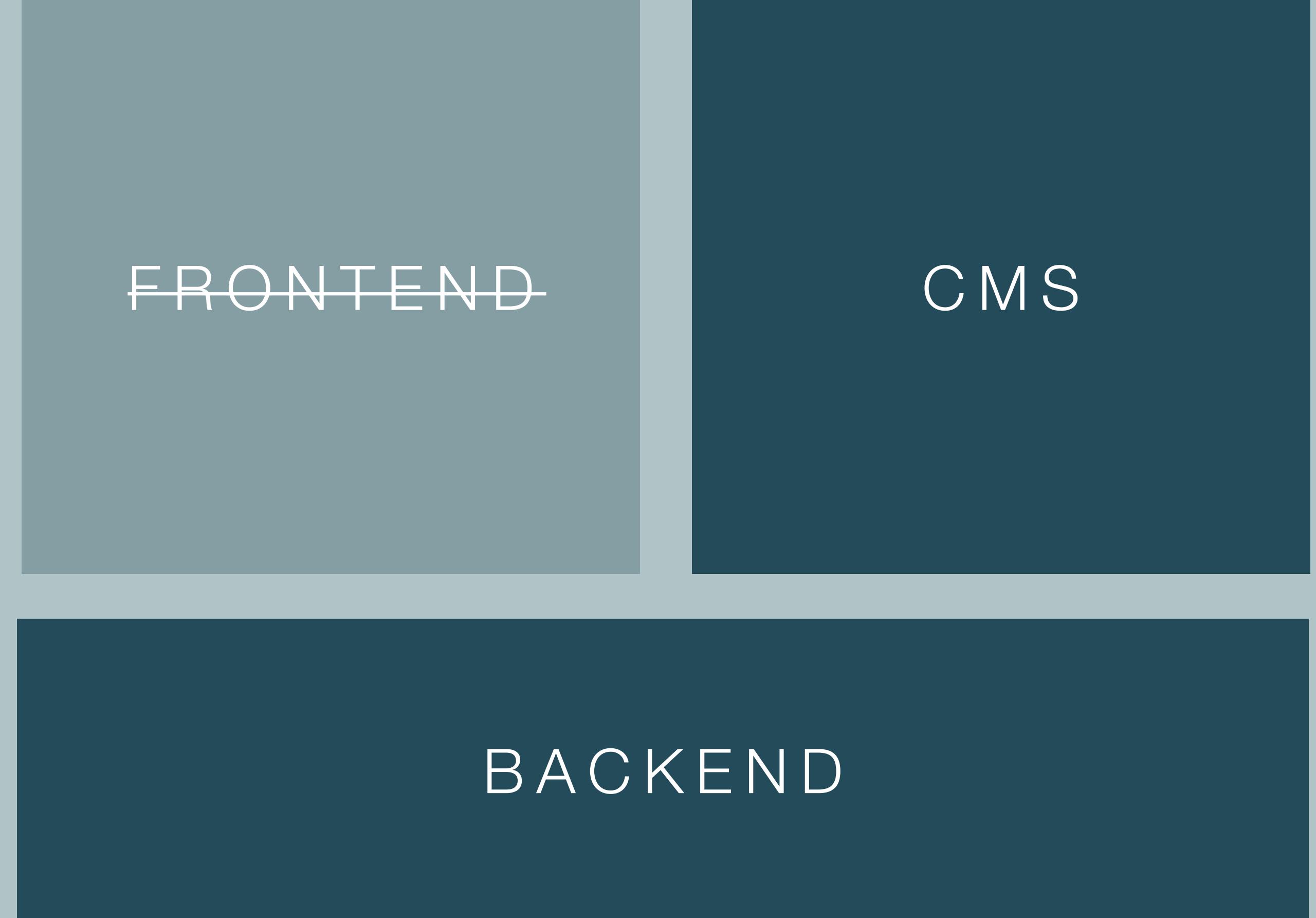
Traditional CMS

FRONTEND

CMS

BACKEND

Headless CMS

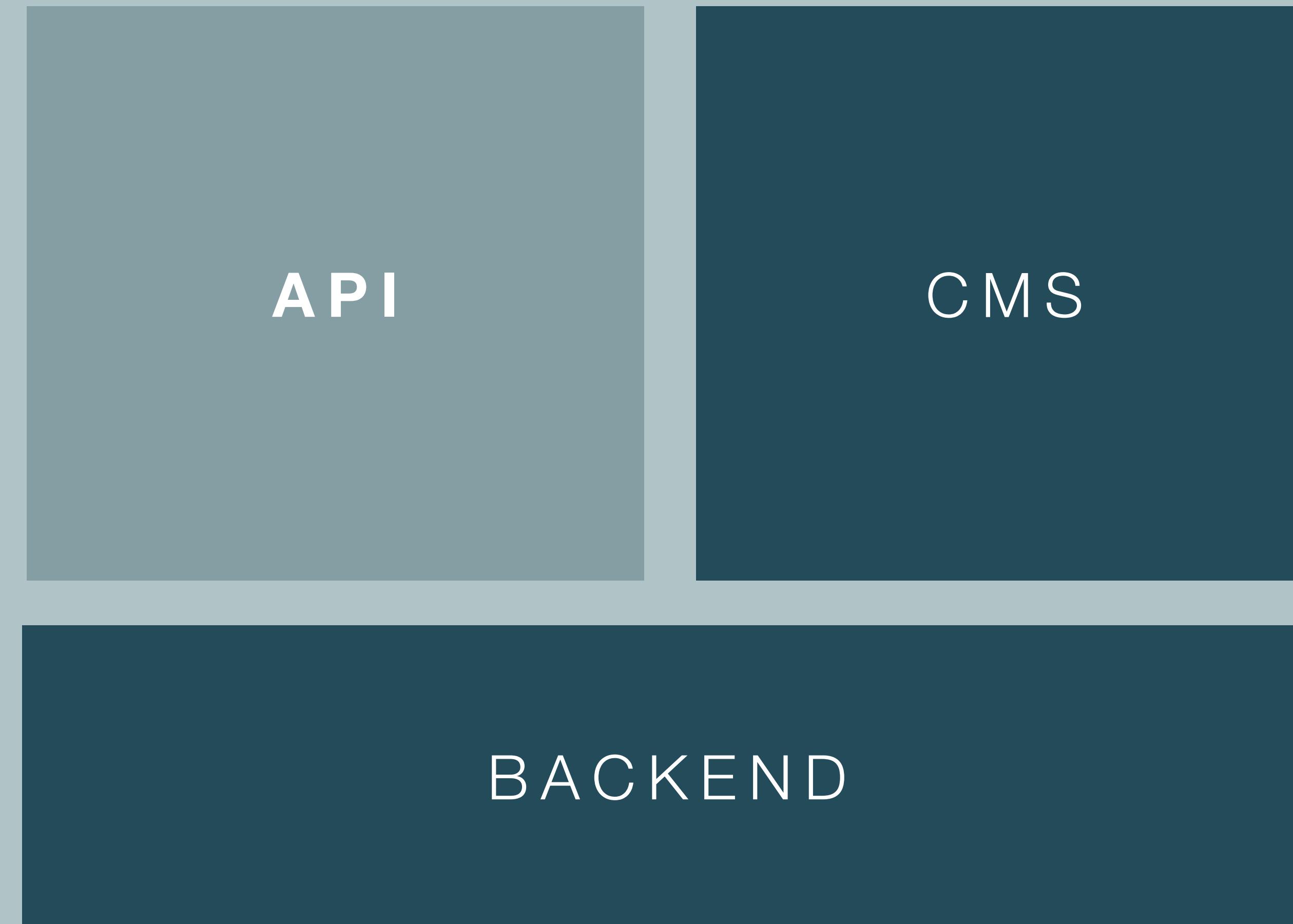


FRONTEND

CMS

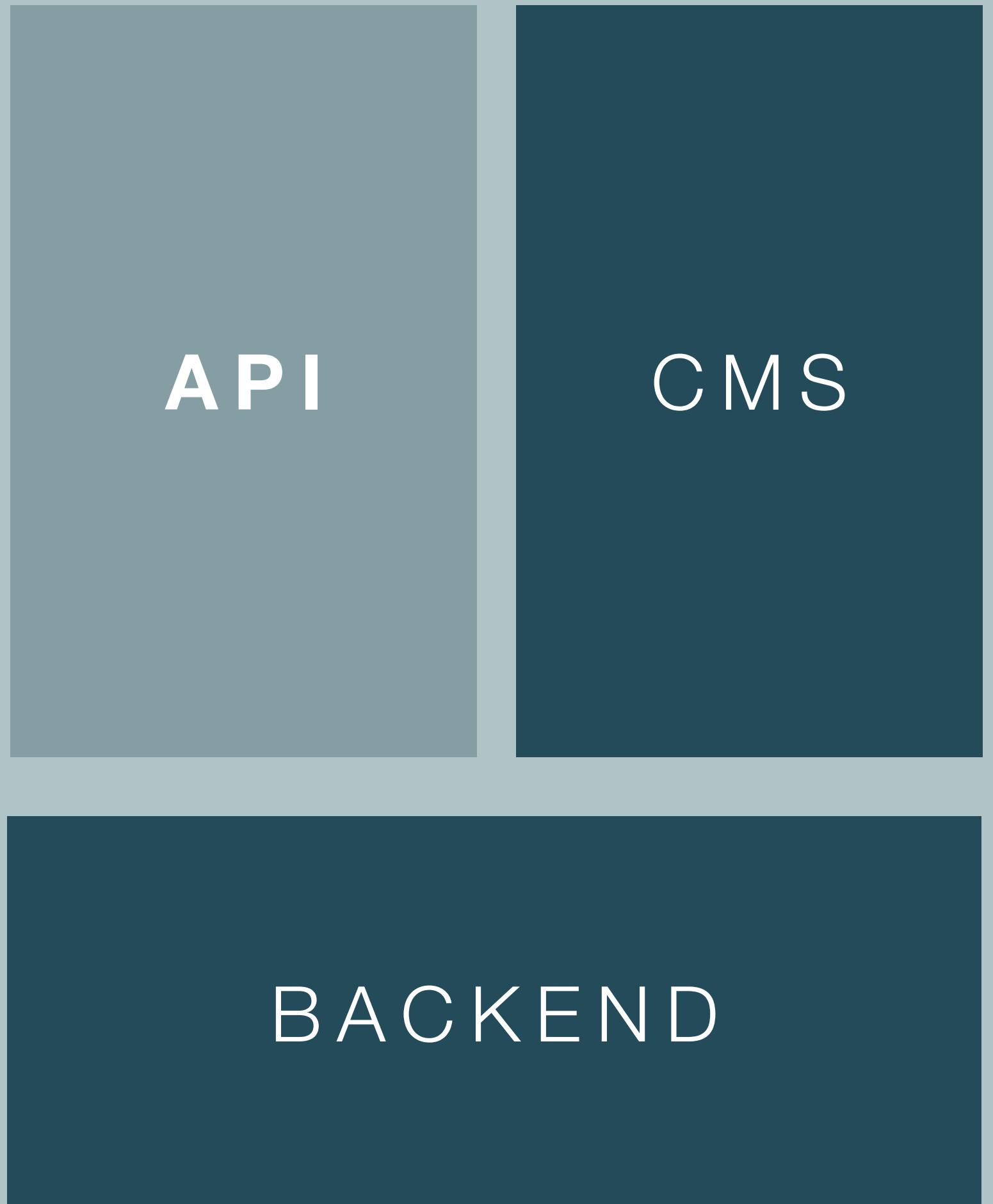
BACKEND

Headless CMS



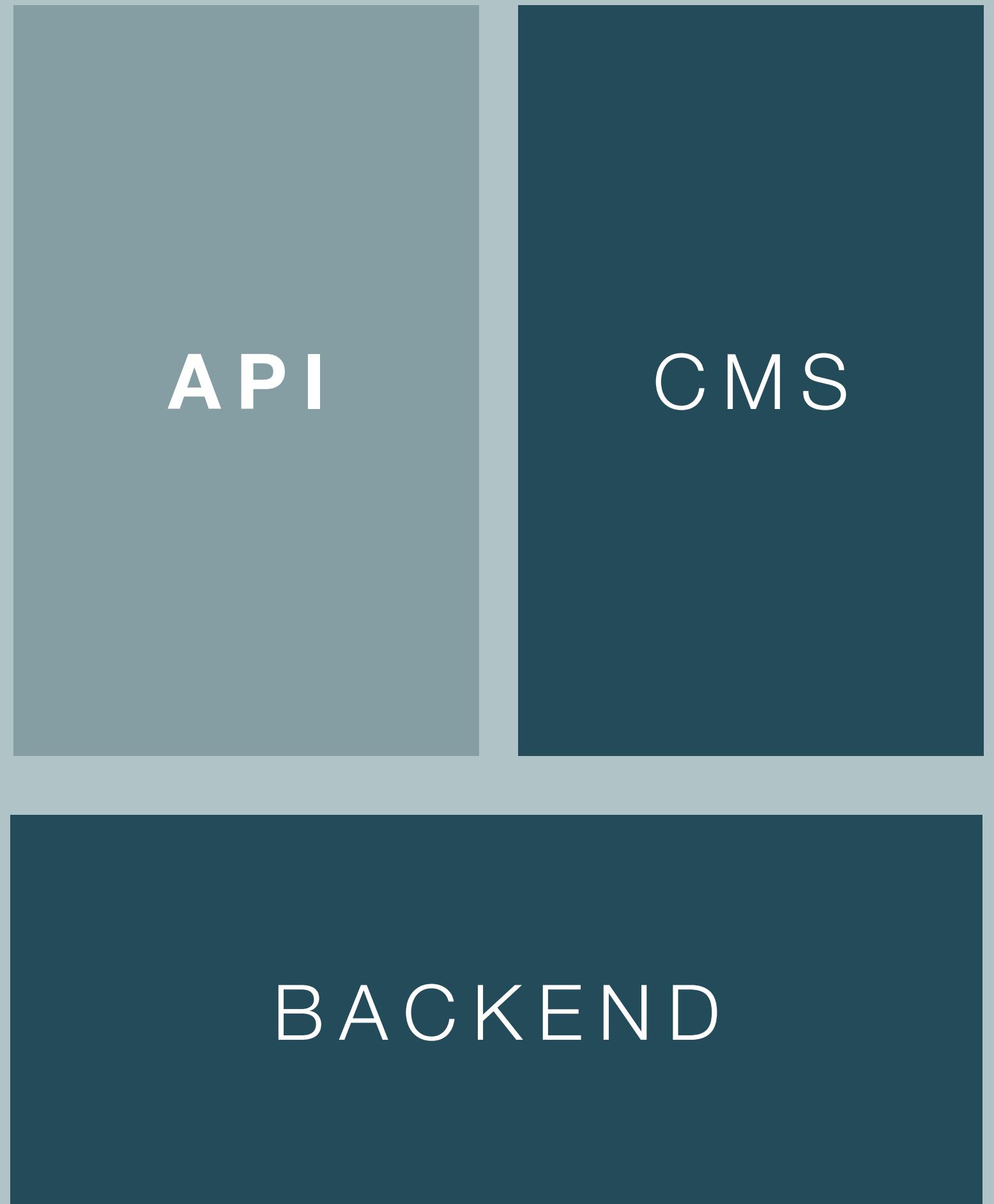
Headless CMS

- A Headless CMS is a backend-only content management system.
- It manages content without a predefined front-end (the “head”).
- Content is delivered via APIs (e.g., REST or GraphQL).
- Developers can build custom front-ends for websites, apps, or digital experiences.
- It supports any framework or tool of the developer’s choice.

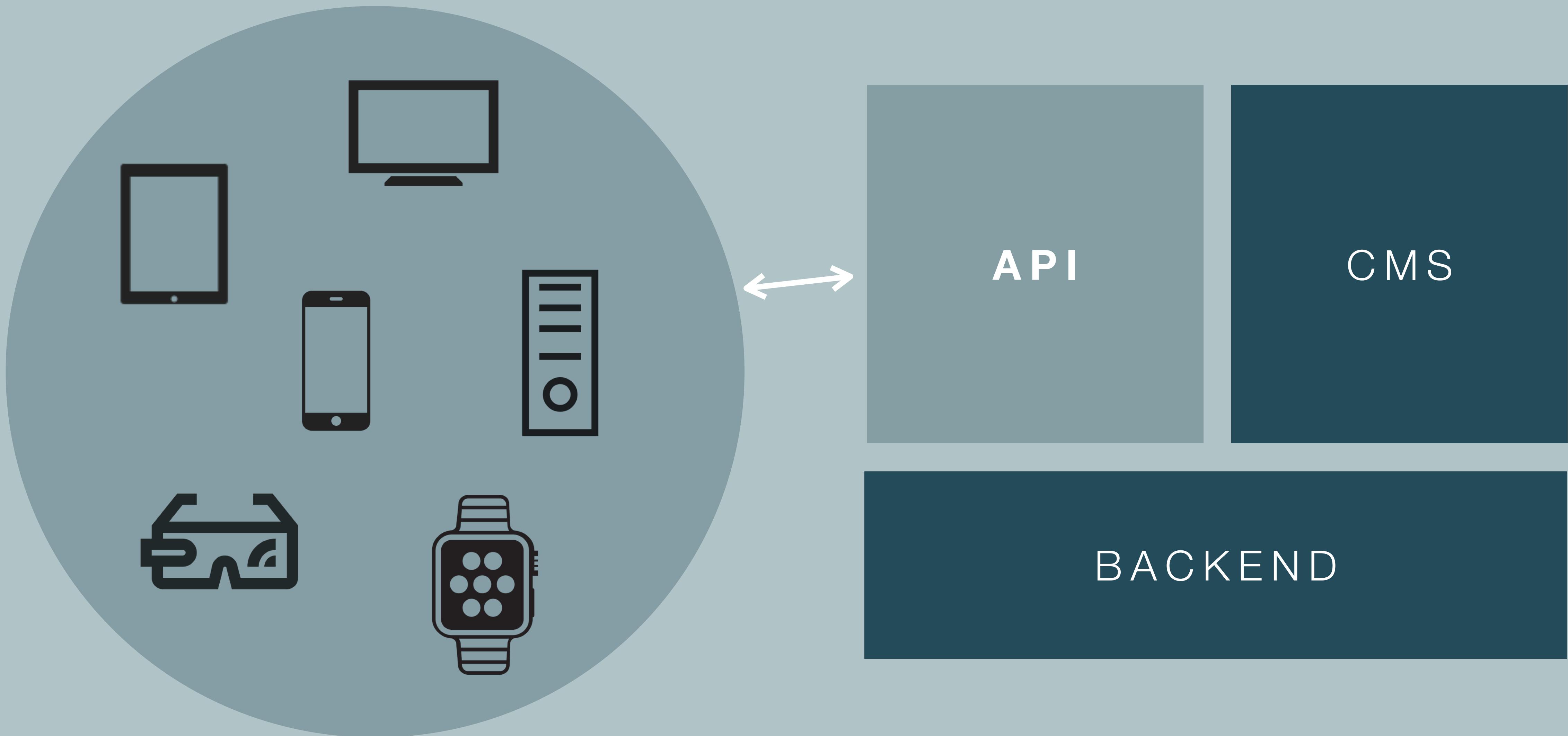


Key Features

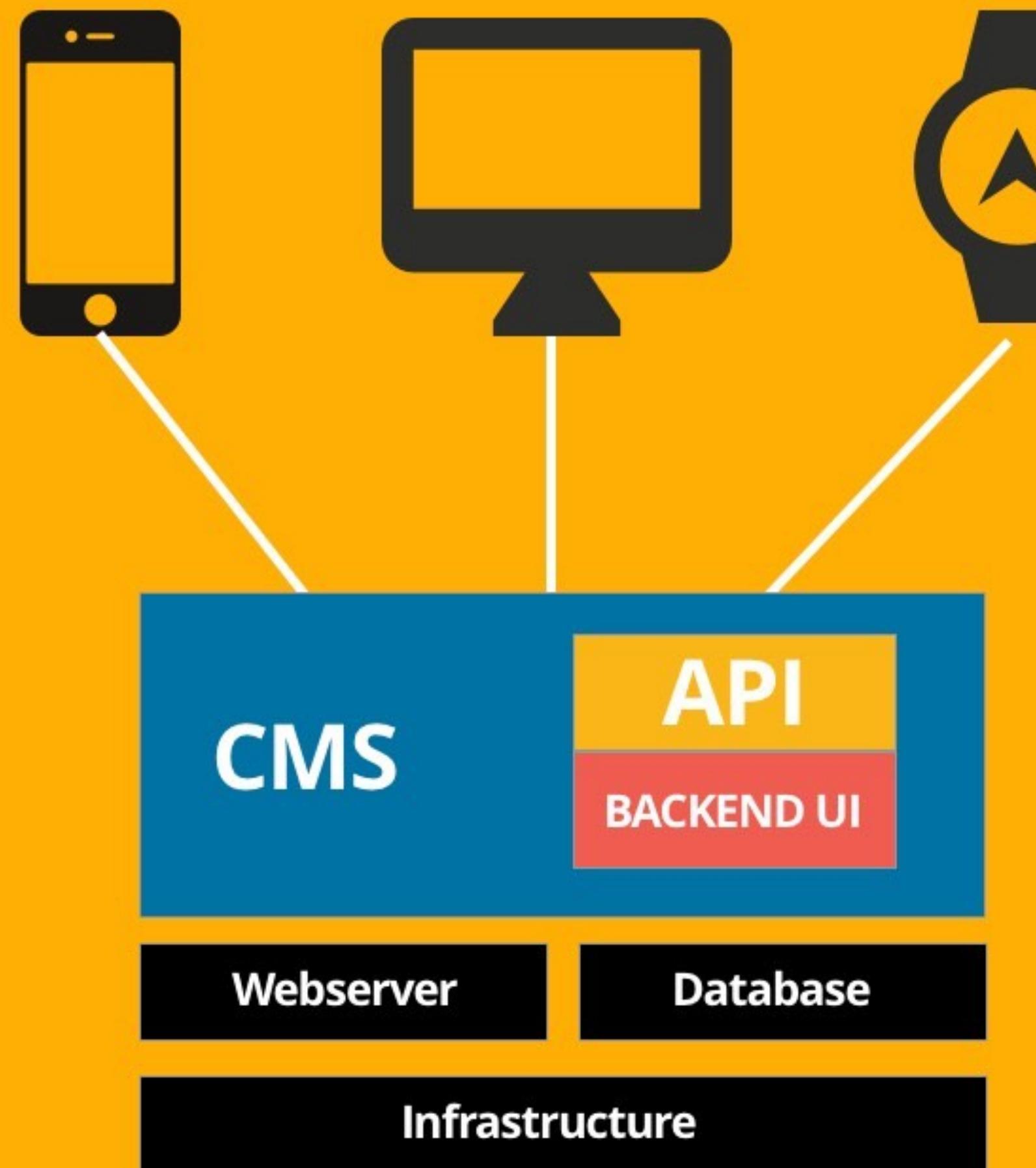
- **Backend-only architecture:** No fixed presentation layer.
- **API-first:** Content is accessed programmatically via APIs.
- **Flexibility:** Works with any front-end technology.
- **Scalability:** Ideal for omnichannel experiences (web, mobile, IoT, etc.).



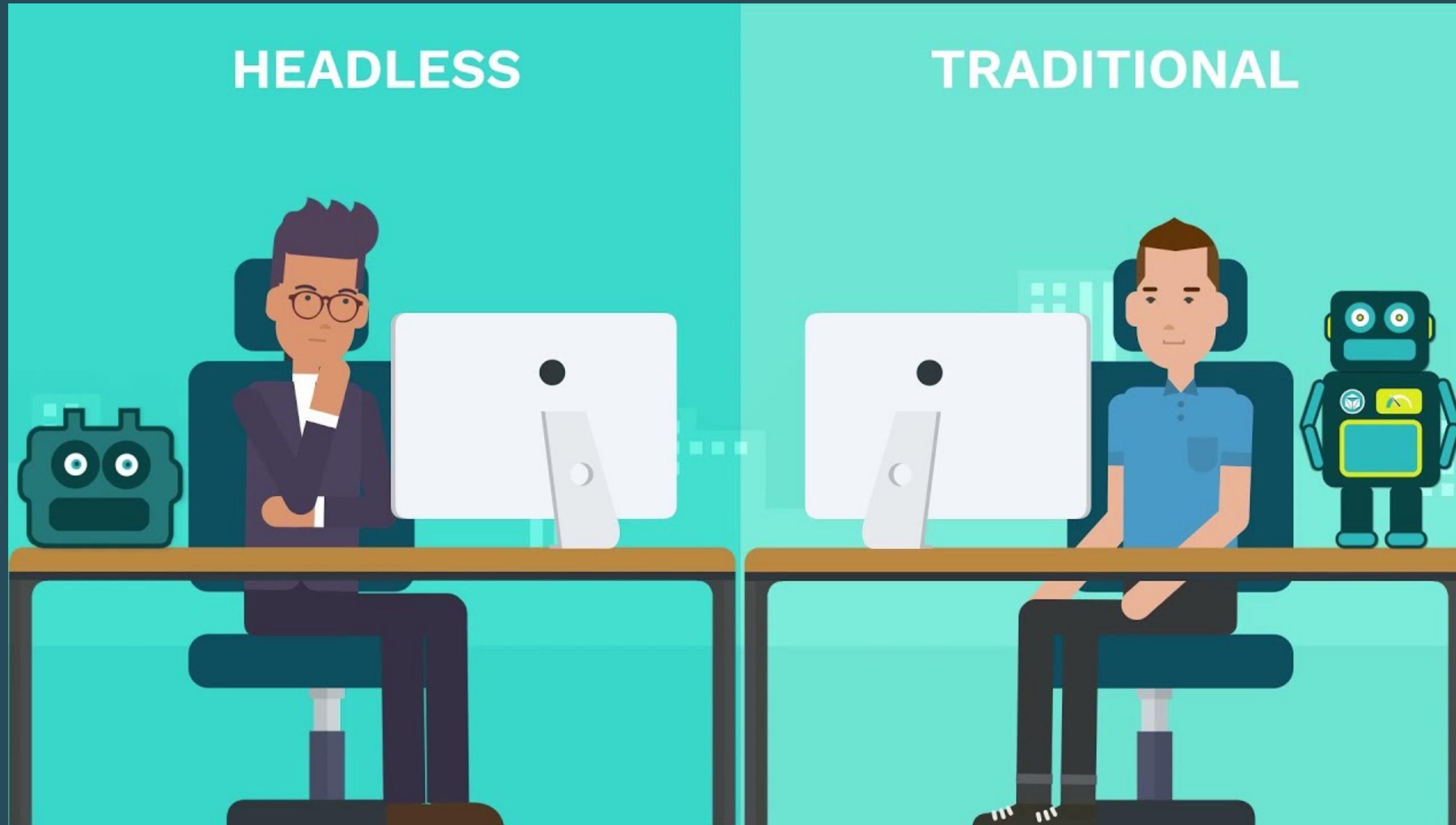
Headless CMS



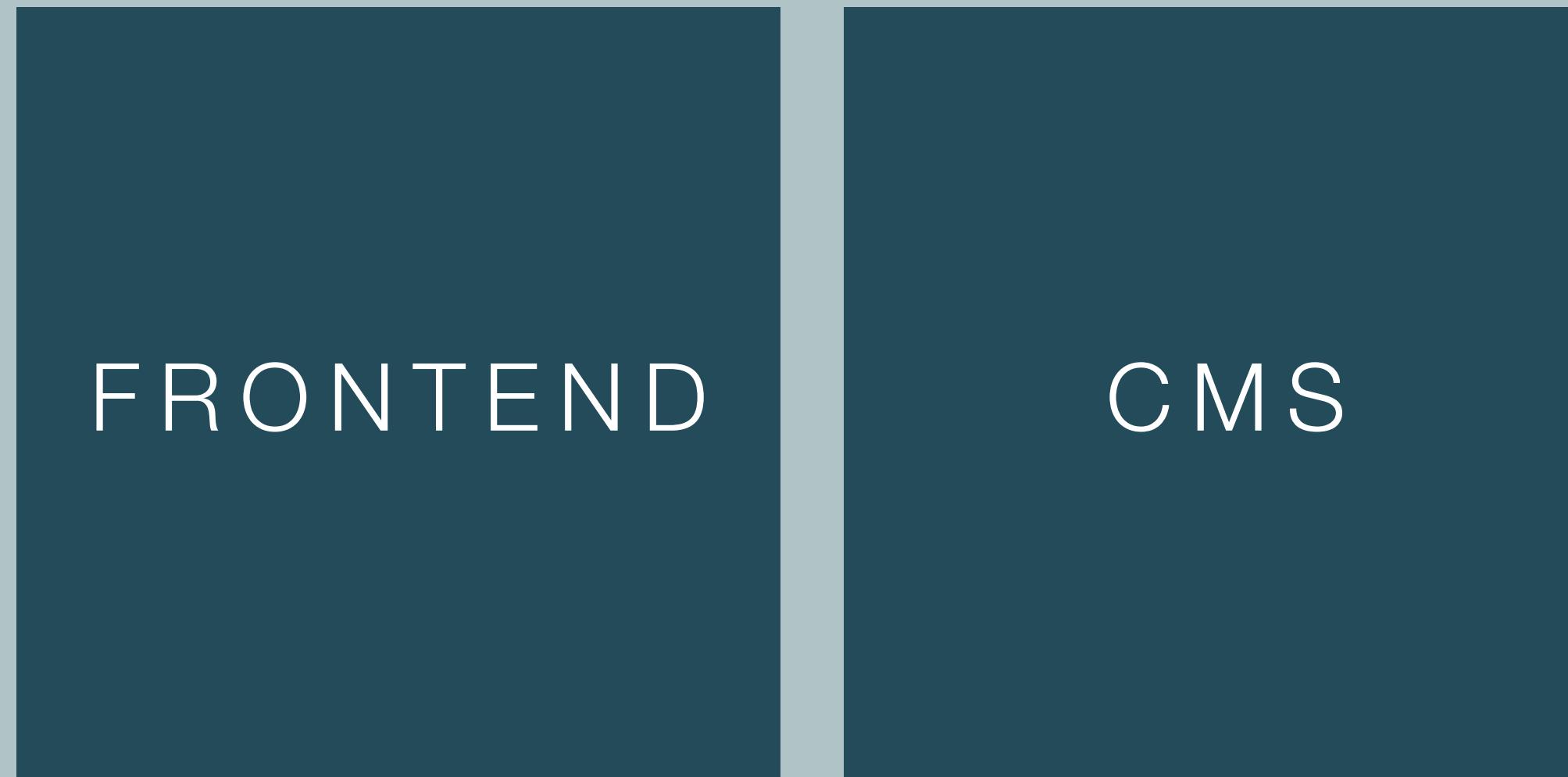
Headless CMS



Headless CMS vs Traditional CMS



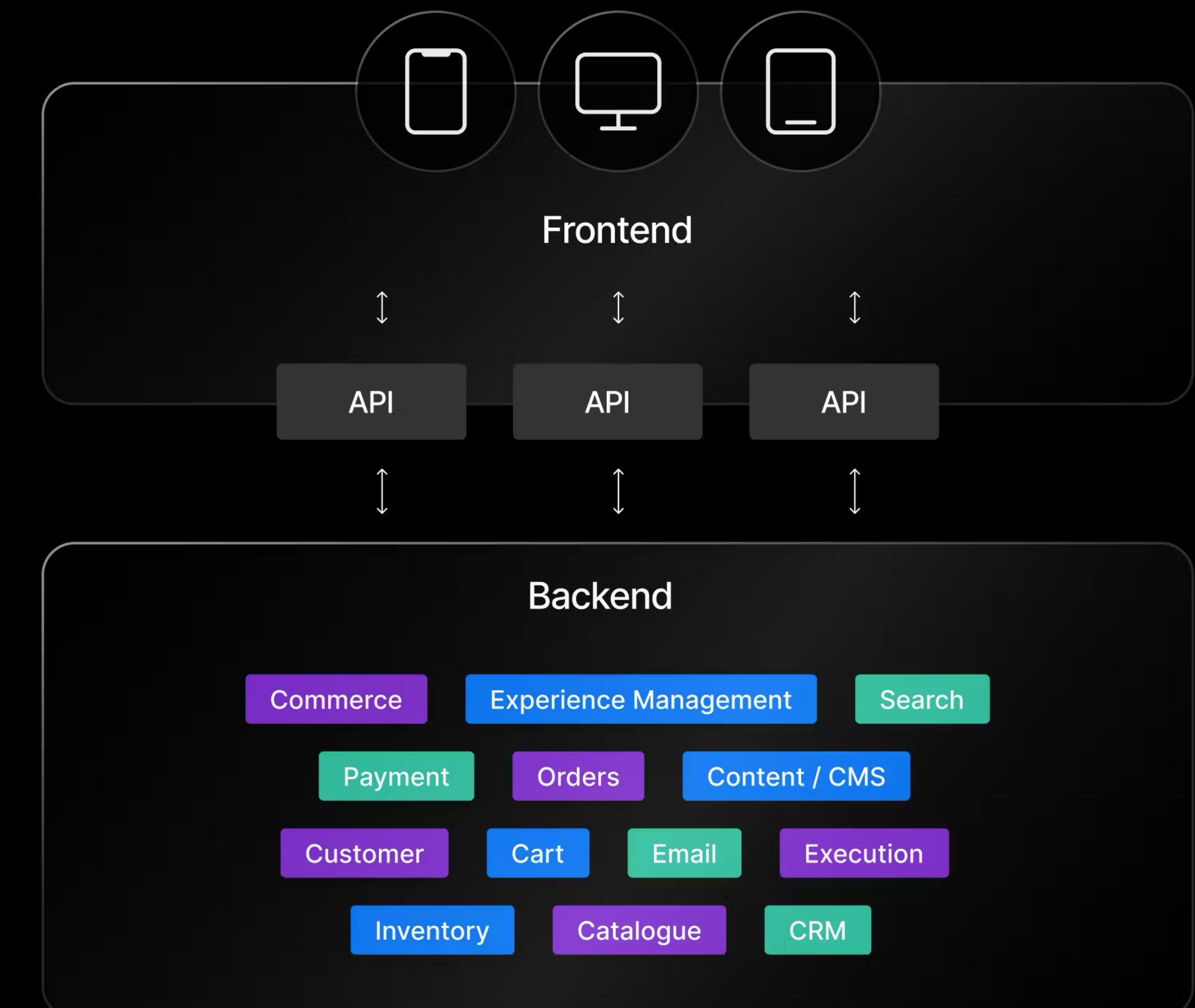
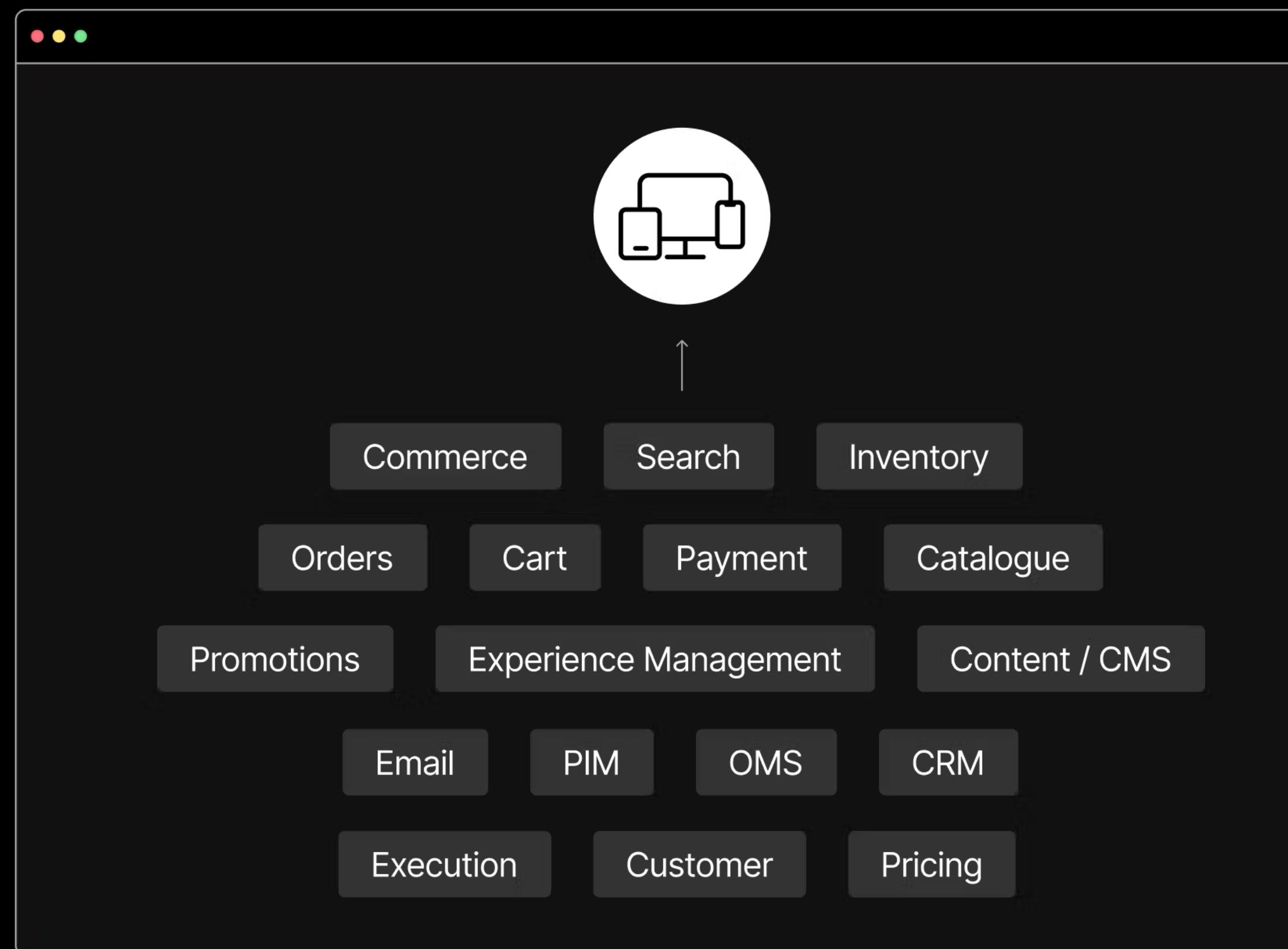
Traditional CMS



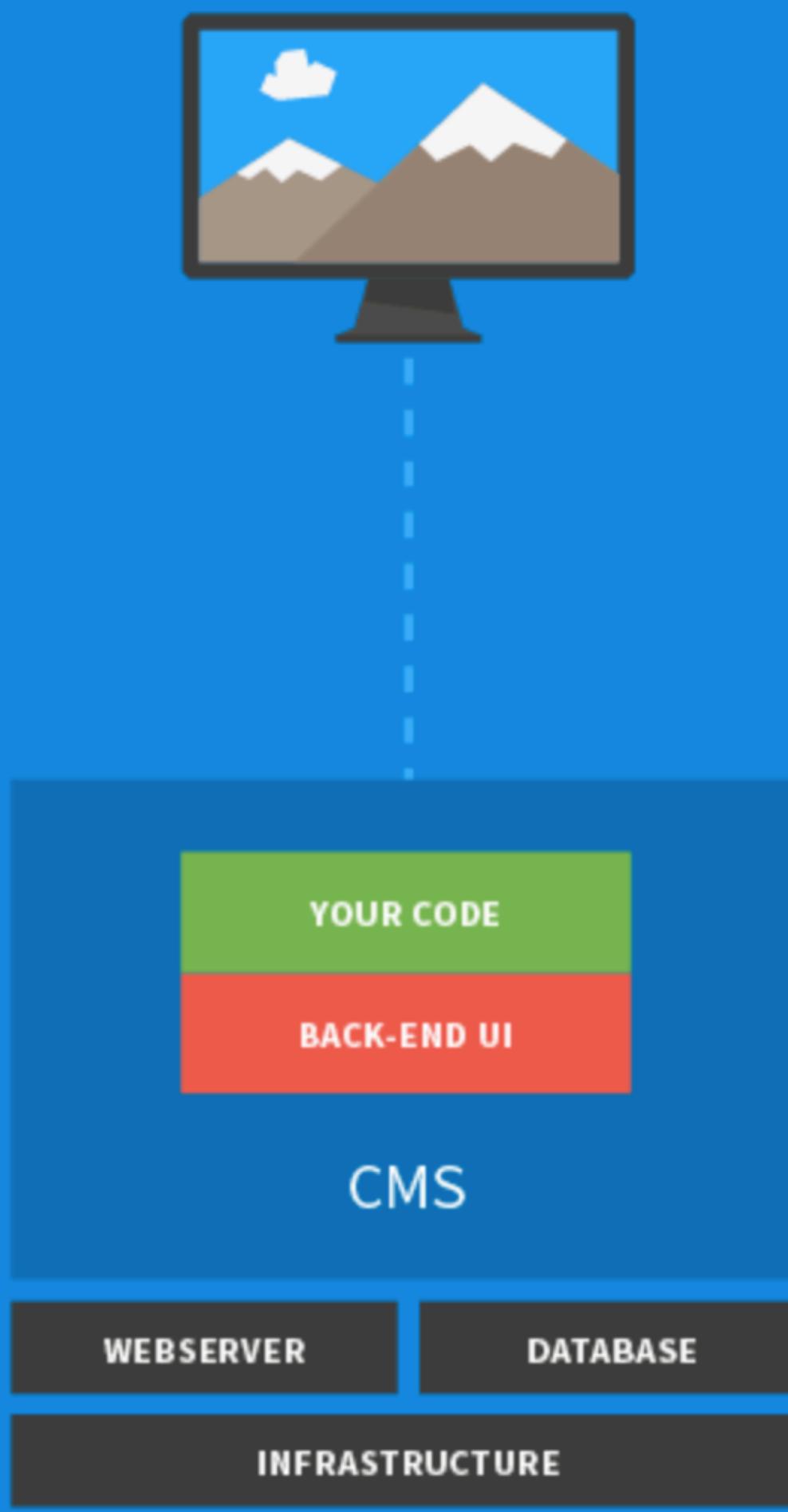
Headless CMS



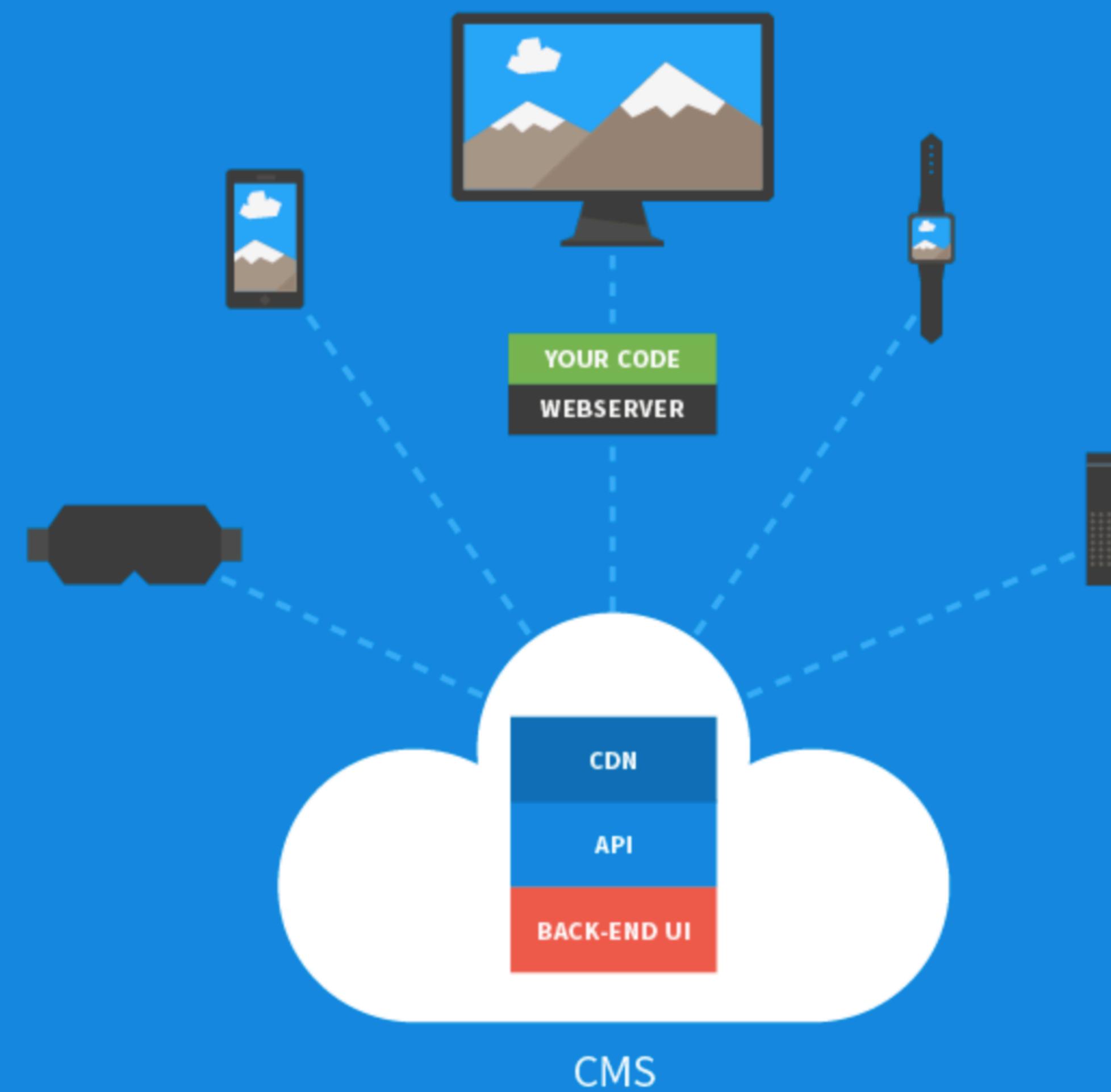
Monolith vs. Headless



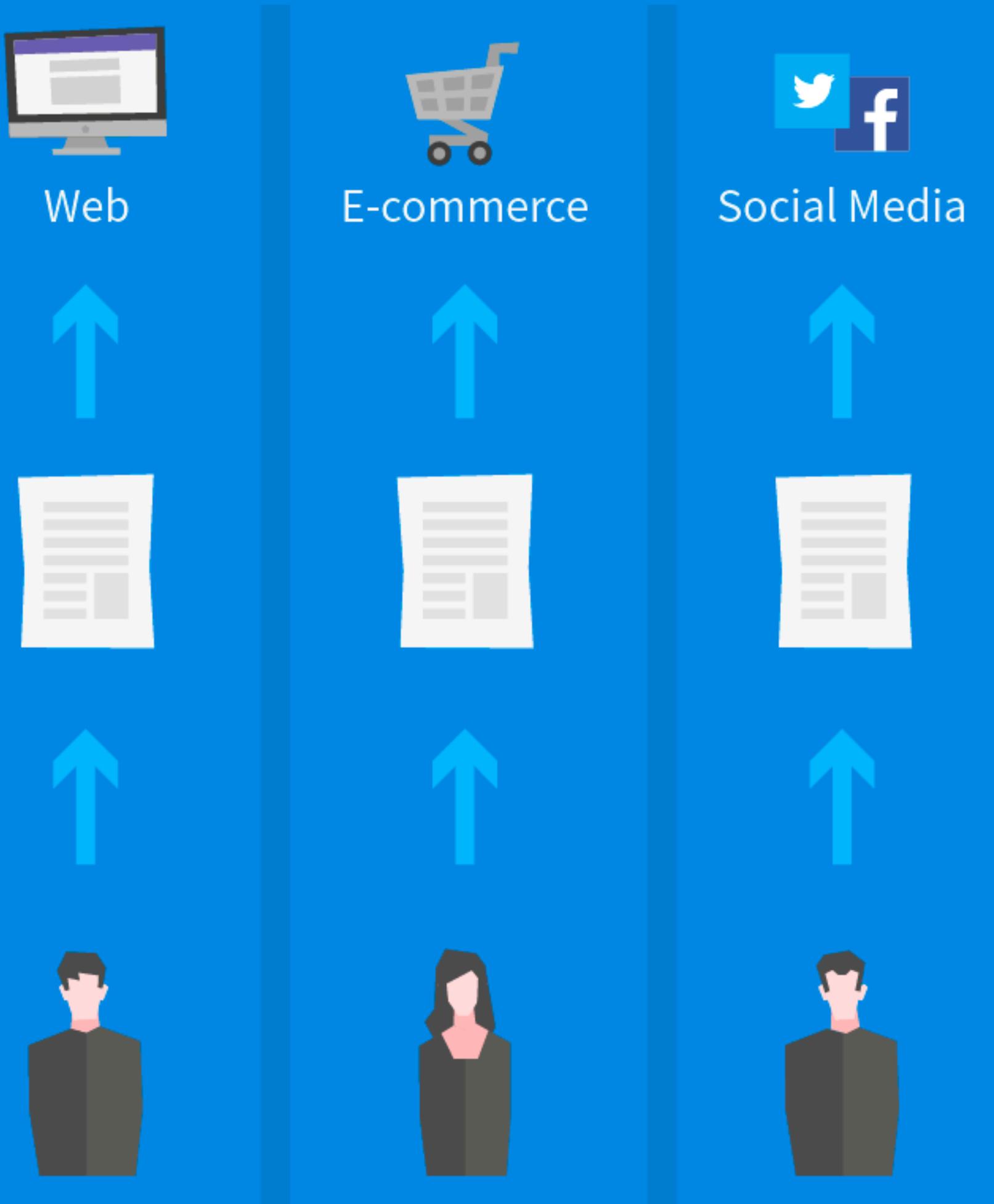
TRADITIONAL CMS



HEADLESS CMS



TRADITIONAL CMS



HEADLESS CMS



TRADITIONAL CMS

A WAY TO STORE DATA

A CRUD UI

A WAY TO DISPLAY THE DATA

CRUD: CREATE, READ, UPDATE & DELETE

HEADLESS CMS

A WAY TO STORE DATA

A CRUD UI

AN API TO ACCESS THE DATA

WordPress

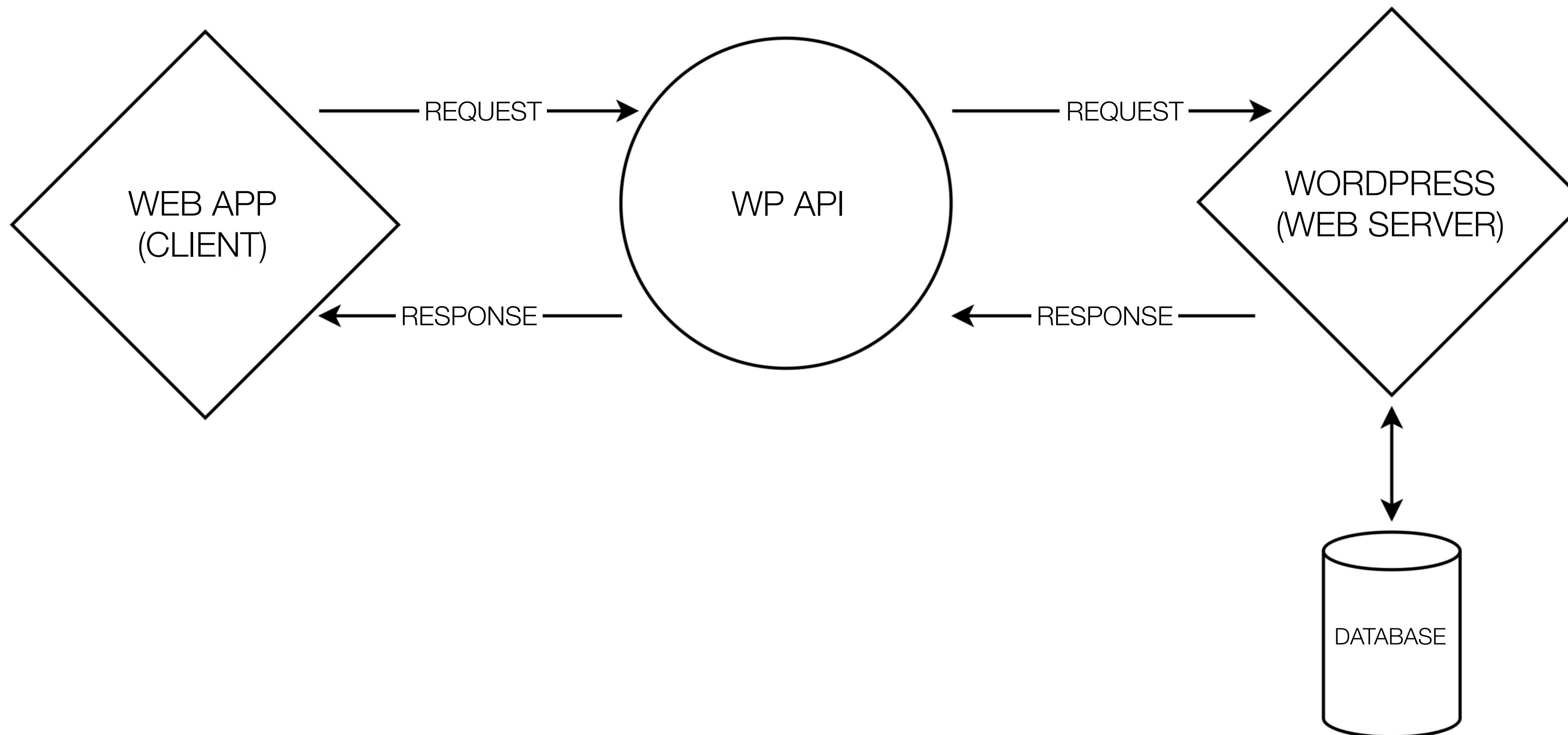
FRONTEND

API

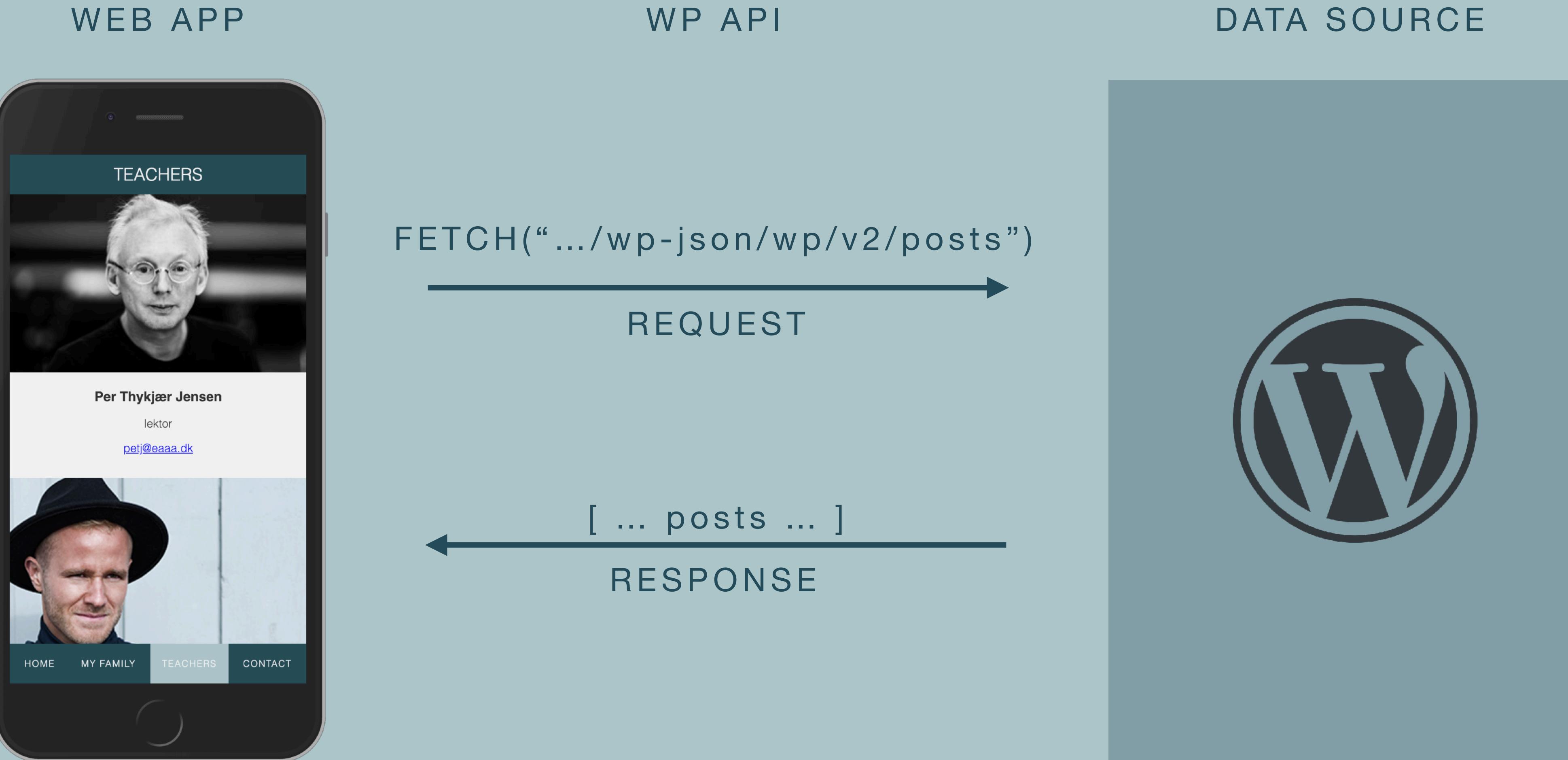
CMS

BACKEND

APPLICATION PROGRAMMING INTERFACE



Fetch & Headless CMS



Posts < MY HEADLESS CMS — +

headlesscms.cederdorff.com/wp-admin/edit.php

MY HEADLESS CMS 27 + New

Howdy, mdu-front

Dashboard

Posts

All Posts Add New

All (8) | Published (8) | Trash (2)

Bulk actions ▾ Apply All dates ▾ All Categories ▾ Filter

8 items

<input type="checkbox"/> Title	Author	Categories	Tags	Comment	Date
Martin	admin	Persons	—	0 1	Published 2020/08/31 at 6:37 am
Peter Cederdorff	admin	Family Members	—	—	Published 2019/02/07 at 4:14 pm
Rasmus Cederdorff	admin	Family Members	—	0 3	Published 2019/02/07 at 4:12 pm
Alicia	admin	Family Members	—	0 3	Published 2019/02/07 at 11:34 am
Per Thykjær	admin	Teachers	—	0 6	Published 2019/02/06 at 7:40 am
Rasmus Cederdorff	admin	Teachers	—	0 4	Published 2019/02/05 at 6:35 pm
Michael Hvidtfeldt	admin	Teachers	—	0 2	Published 2019/02/05 at 6:34 pm
Birgitte Kirk Iversen	admin	Teachers	—	0 8	Published 2019/02/05 at 6:32 pm
<input type="checkbox"/> Title	Author	Categories	Tags	Comment	Date

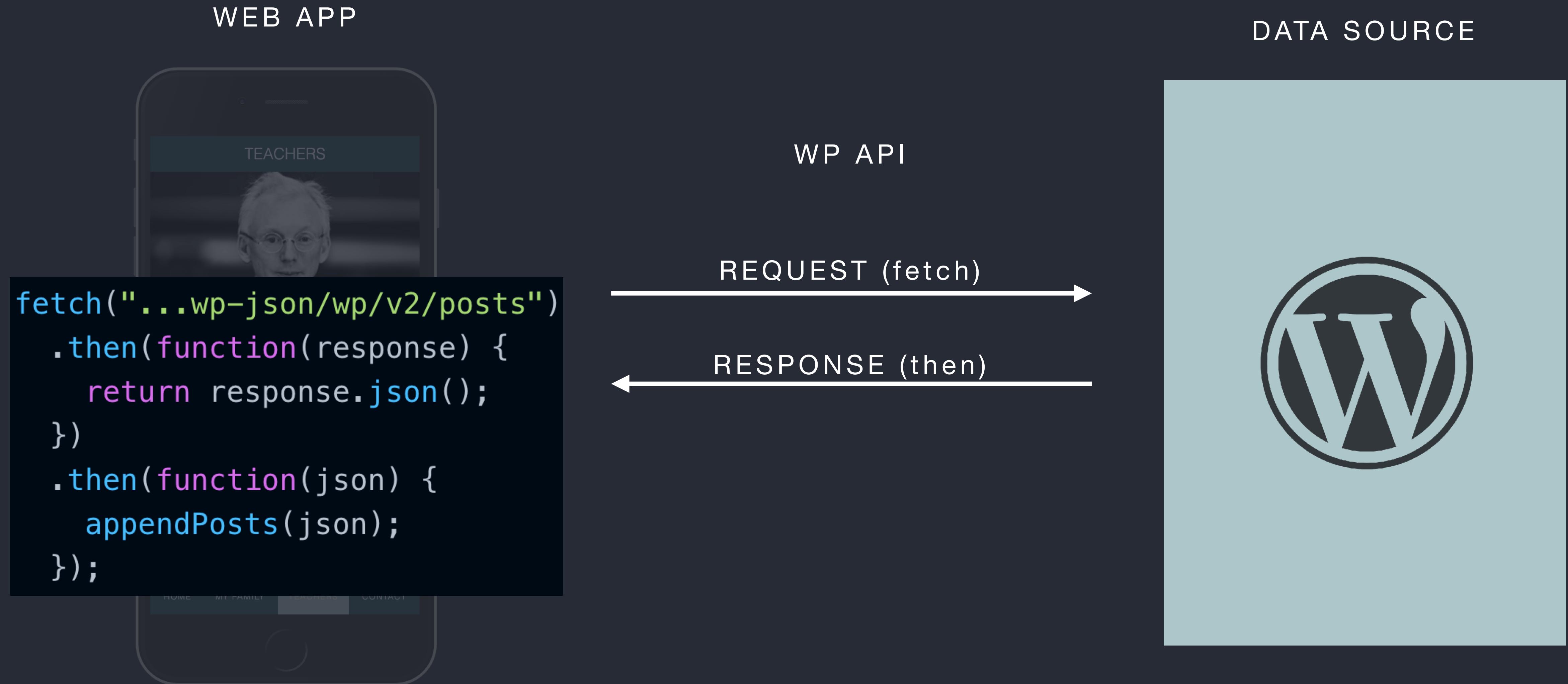
Bulk actions ▾ Apply

8 items

Thank you for creating with [WordPress](#).

Version 5.5.5

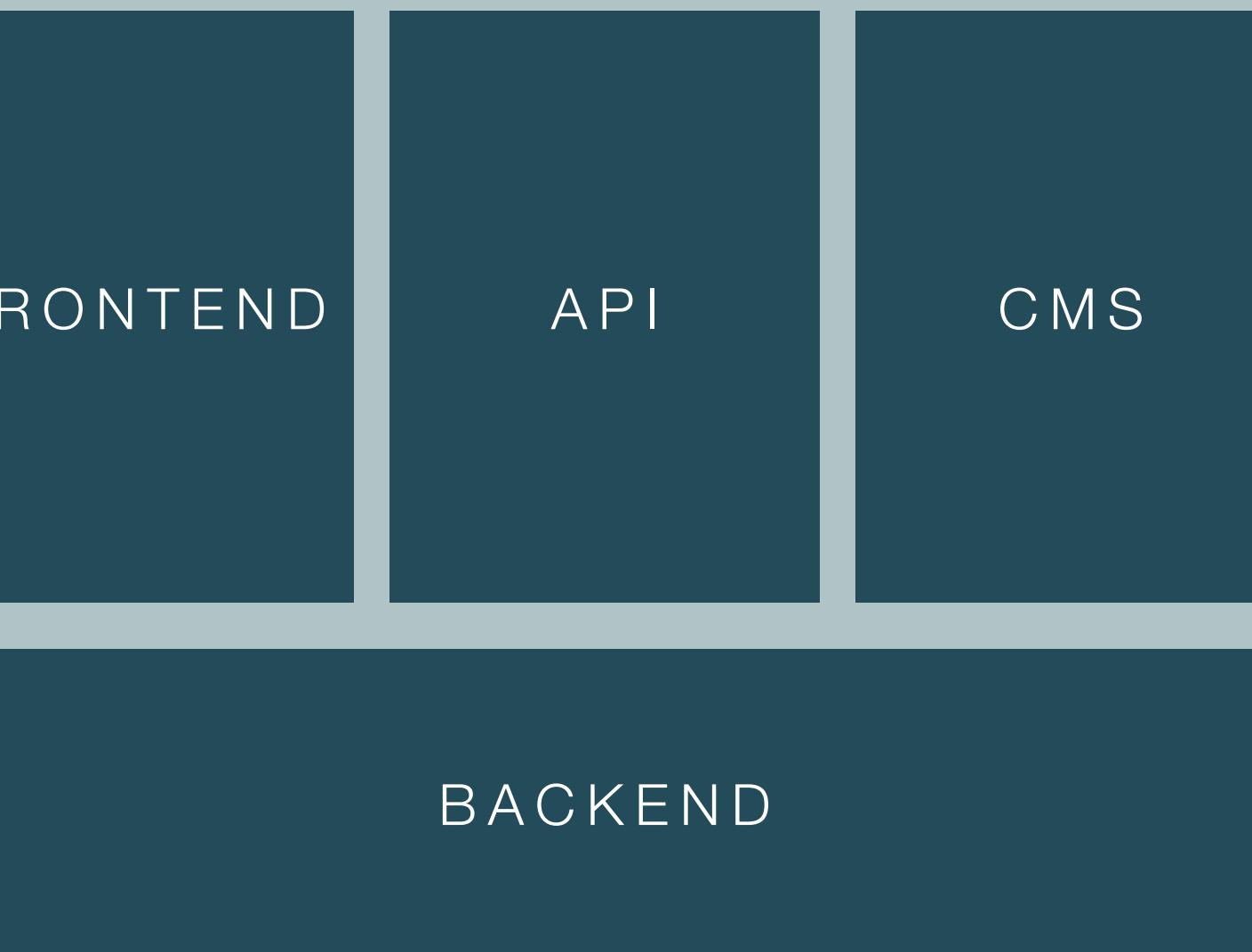
Fetch & Headless CMS



WEB APP



HEADLESS WP



movie-api.cederdorff.com

Fetch posts

Screenshot of the WordPress admin dashboard showing the Posts screen.

The dashboard header includes the WordPress logo, site title "Movie API", a refresh icon with "7", a comment icon with "0", and a "+ New" button.

A notification bar at the top right says "WordPress 5.3.2 is available! [Please update now.](#)"

The left sidebar menu is visible, with "Posts" selected and highlighted in blue. Other menu items include Dashboard, All Posts, Add New, Categories, Tags, Media, Pages, Comments, Appearance, Plugins (with a red "3" badge), Users, Tools, Settings, Custom Fields, and Collapse menu.

The main content area is titled "Posts" with a "Add New" button. It shows a list of published posts:

	Title	Author	Categories	Tags
<input type="checkbox"/>	Frozen	admin	Animation, Comedy	—
<input type="checkbox"/>	Don't Let Go	admin	Drama, Horror	—
<input type="checkbox"/>	Lion King	admin	Animation	—
<input type="checkbox"/>	Title ▲			

Below the table, there are "Bulk Actions" dropdown and "Apply" buttons.

Fetch posts

```
let _movies = [];

// fetch all movies from WP
async function getMovies() {
  let response = await fetch("https://movie-api.cederdorff.com/wp-json/wp/v2/posts?_embed");
  let data = await response.json();
  _movies = data;
  appendMovies(data);
  showLoader(false);
}

getMovies();
```

“Headless” Person Exercise

1. Copy and use the template: headless-cms-template
2. Fetch data from WP API: <https://headlesscms.cederdorff.com/wp-json/wp/v2/posts>
3. Append the fetched posts to the DOM. Display image, title and content.
4. Make sure you fetch the featured image by using `_embed` as argument in the URL
5. Add new posts in WordPress Admin and refresh your web app.

Admin: <https://headlesscms.cederdorff.com/wp-login>

Username: mdu-front

Password: headlessfront01

WordPress REST API

- From version 4.7 WP API is included in WordPress
- Install a fresh copy of WP
- Change Permalink Settings: POST NAME
- Test in the browser: <http://your-domain.com/wp-json/>
- Fetch JSON data from the API



API Routes

Meta data: <http://your-domain.com/wp-json/>

Posts: <http://your-domain.com/wp-json/wp/v2/posts>

Pages: <http://your-domain.com/wp-json/wp/v2/pages>

Media: <http://your-domain.com/wp-json/wp/v2/media>

Types: <http://your-domain.com/wp-json/wp/v2/types>

Taxonomies: <http://your-domain.com/wp-json/wp/v2/taxonomies>

Categories: <http://your-domain.com/wp-json/wp/v2/categories>

Tags: <http://your-domain.com/wp-json/wp/v2/tags>

Users: <http://your-domain.com/wp-json/wp/v2/users>

Comments: <http://your-domain.com/wp-json/wp/v2/comments>

DOCS: <https://developer.wordpress.org/rest-api/>

Embed Featured Image

Posts with featured image:

[http://api.cederdorff.com/wp-json/wp/v2/posts? embed](http://api.cederdorff.com/wp-json/wp/v2/posts?embed)

Pages with featured image:

[http://api.cederdorff.com/wp-json/wp/v2/pages? embed](http://api.cederdorff.com/wp-json/wp/v2/pages?embed)

By Category

Posts by a specific category:

<https://api.cederdorff.com/wp-json/wp/v2/posts?categories=2>

?categories=2

Dashboard

Posts

All Posts

Add New

Categories

Tags

Media

Pages

Comments 4

Appearance

Plugins 2

Users

Tools

Settings

Collapse menu

Edit Category

Name

Client

The name is how it appears on your site.

Slug

client

The “slug” is the URL-friendly version of the name. It is usually all lowercase and contains only letters, numbers, and hyphens.

Parent Category

None

Categories, unlike tags, can have a hierarchy. You might have a Jazz category, and under that have children categories for Bebop and Big Band. Totally optional.

Description

The description is not prominent by default; however, some themes may show it.

Category & Featured Image

Posts with featured image and by specific category:

[https://api.cederdorff.com/wp-json/wp/v2/posts?
_embed&categories=2](https://api.cederdorff.com/wp-json/wp/v2/posts?_embed&categories=2)

? _embed&categories=2

Filter, sort & order

[https://api.cederdorff.com/wp-json/wp/v2/posts? embed](https://api.cederdorff.com/wp-json/wp/v2/posts?embed)

[https://api.cederdorff.com/wp-json/wp/v2/posts?
embed&categories=2](https://api.cederdorff.com/wp-json/wp/v2/posts?embed&categories=2)

[https://api.cederdorff.com/wp-json/wp/v2/posts?
orderby=date&order=desc&after=2019-00-10T17:00:00](https://api.cederdorff.com/wp-json/wp/v2/posts?orderby=date&order=desc&after=2019-00-10T17:00:00)

[https://api.cederdorff.com/wp-json/wp/v2/posts?
orderby=date&order=asc](https://api.cederdorff.com/wp-json/wp/v2/posts?orderby=date&order=asc)

Examples of Headless WPs

<http://nanadorph.dk>

<http://www.sommerhusborkhavn.dk>

<https://headlesscms.cederdorff.com/>

<https://product-api.cederdorff.com/>

<https://api.cederdorff.com/>

<https://movie-api.cederdorff.com/>

<https://react-api.cederdorff.com/>

<https://react-pf-api.cederdorff.com/>



REST API Handbook



CHAPTERS

[REST API Handbook](#)[Key Concepts](#)[Frequently Asked Questions](#)[Using the REST API](#)[Extending the REST API](#)[Reference](#)[Changelog](#)Browse: [Home](#) / REST API Handbook

REST API Handbook



The WordPress REST API provides an interface for applications to interact with your WordPress site by sending and receiving data as [JSON](#) (JavaScript Object Notation) objects. It is the foundation of the [WordPress Block Editor](#), and can likewise enable your theme, plugin or custom application to present new, powerful interfaces for managing and publishing your site content.

Using the WordPress REST API you can create a plugin to provide an entirely new admin experiences for WordPress, build a brand new interactive front-end experience, or bring your WordPress content into completely separate applications.

The REST API is a developer-oriented feature of WordPress. It provides data access to the content of your site, and implements the same authentication restrictions — content that is public on your site is generally publicly accessible via the REST API, while private content, password-protected content, internal users, custom post types, and metadata is only available with authentication or if you specifically set it to be so. If you are not a developer, the most important thing to understand about the API is that it enables the block editor and modern plugin interfaces without compromising the security or privacy of your site.

What Is A REST API?

An API is an Application Programming Interface. REST, standing for "REpresentational State

TOPICS

[What Is A REST API?](#)[Using the WordPress REST API](#)[Next Steps](#)

Projects

headless-cms

headless-cms-acf-products

headless-cms-acf-teachers

headless-cms-template

spa-headless-cms-movies

spa-headless-cms-movies-template

spa-headless-cms-teachers-family-members

spa-headless-cms-template

<https://github.com/cederdorff/react-headless-template>

<https://github.com/cederdorff/react-headless-spa>

<https://github.com/cederdorff/react-cederdorff-headless>

Headless & SPA

Template: spa-template

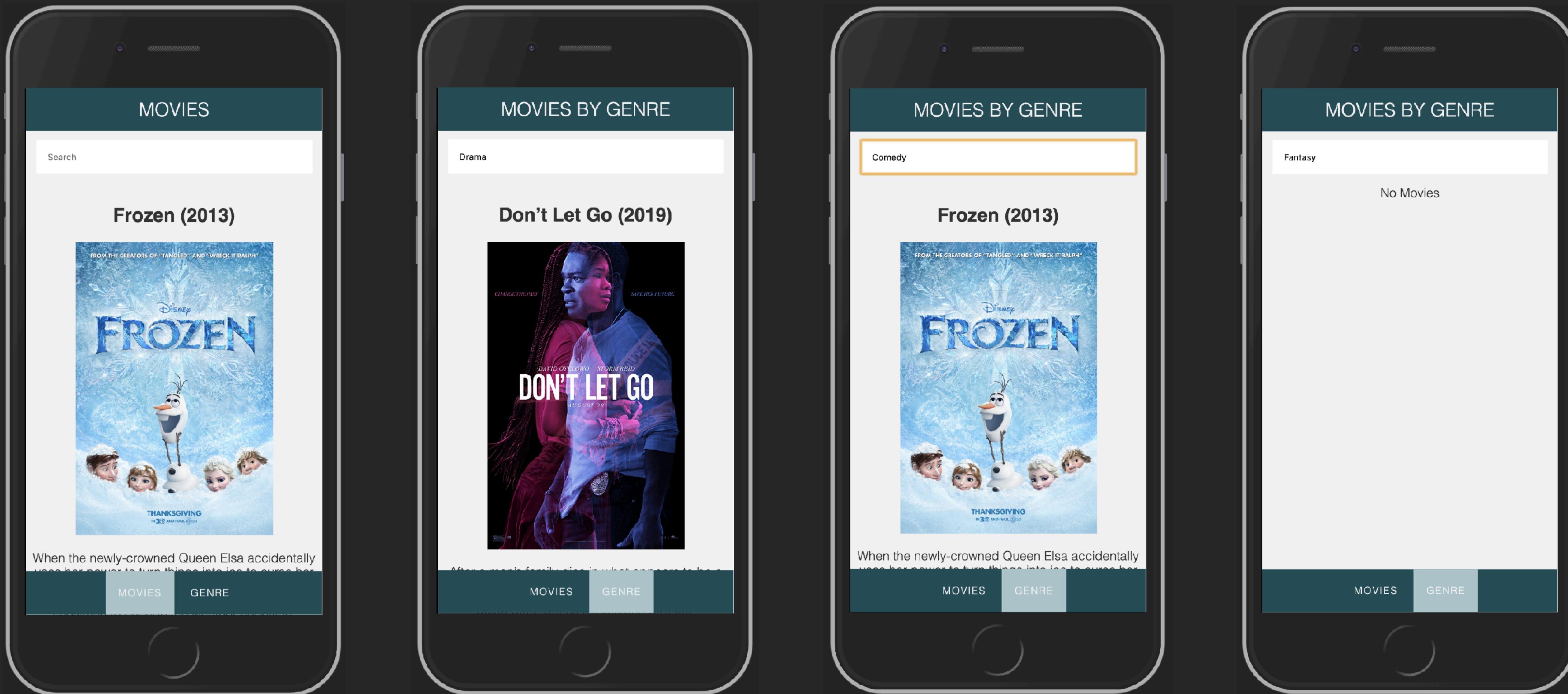
1. Create a SPA with the following views: All Persons, Family & Teachers.
2. Fetch data from the Headless CMS headlesscms.cederdorff.com and use the category id to fetch different post categories:
 1. All Persons: [https://headlesscms.cederdorff.com/wp-json/wp/v2/posts? embed](https://headlesscms.cederdorff.com/wp-json/wp/v2/posts?embed)
 2. Teachers: [https://headlesscms.cederdorff.com/wp-json/wp/v2/posts? embed&categories=2](https://headlesscms.cederdorff.com/wp-json/wp/v2/posts?embed&categories=2)
 3. [https://headlesscms.cederdorff.com/wp-json/wp/v2/posts? embed&categories=3](https://headlesscms.cederdorff.com/wp-json/wp/v2/posts?embed&categories=3)
3. Append the posts to the matching views in your SPA.
4. Implement a functional search field.
5. Order posts by name (title property).

Admin: <https://headlesscms.cederdorff.com/wp-login>

Username: mdu-front

Password: headlessfront01

SPA + HEADLESS CMS



<https://github.com/cederdorff/react-headless-template>

<https://github.com/cederdorff/react-headless-spa>

Setup Headless WordPress

<https://race.notion.site/Setup-Headless-WordPress-af155138ff46436797f51e91aaa66c94>

DATA STRUCTURES

MY OWN SIMPLE STRUCTURE

The image shows a code editor interface with two tabs: 'movies.json' and 'main.js'. The 'movies.json' file contains a JSON array of movie objects. The 'main.js' file contains JavaScript code for fetching movies from a JSON file and appending them to a DOM container.

```
{} movies.json × JS main.js × ...
```

```
1  [
2  {
3      "description": "After a man's family dies in what appears to
4      "img": "https://m.media-amazon.com/images/M/MV5BZjI5N2FkMDQt"
5      "title": "Don't Let Go",
6      "year": "2019",
7      "id": "6oaDYnZKFYMC0wrwoz0j"
8  },
9  {
10     "description": "After the murder of his father, a young lion
11     "img": "https://m.media-amazon.com/images/M/MV5BMjIwMjE1Nzc4I
12     "title": "Lion King",
13     "year": "2019",
14     "id": "870sfKDrtHLTNGSPFE7R"
15 },
16 {
17     "description": "Lorem ipsum, også kendt som mumletekst, er en
18     "img": "https://cdn.cinematerial.com/p/297x/q7nffqcp/the-bee
19     "title": "Test movie",
20     "year": "2020",
21     "id": "XRUmxcDmmxmK1gUdyBwV"
22 },
23 {
24     "description": "A faded television actor and his stunt double
25     "img": "https://www.kirkeogfilm.dk/wordpress/wp-content/uplo
26     "title": "Once Upon a Time... in Hollywood",
27     "year": "2019",
28     "id": "XWJLJLQ1DPEZTMH5W"
```

```
17  /**
18   * Fetch movie from data source: JSON file
19   */
20  async function loadMovies() {
21      let response = await fetch("./json/movies.json");
22      let data = await response.json();
23      return data;
24  }
25
26 /**
27  * Appending movies to the DOM by the giving argument
28 */
29 function appendMovies(movies) {
30     let html = "";
31     for (const movie of movies) {
32         console.log(movie);
33         html += /*html*/
34             <article>
35                 <h2>${movie.title} (${movie.year})</h2>
36                 
37                 <p>${movie.description}</p>
38                 ${generateFavMovieButton(movie.id)}
39             </article>
40     ;
41 }
42 document.querySelector("#movie-container").innerHTML = html;
43 }
```

```
[  
  {  
    "id": 68,  
    "date": "2021-09-03T19:43:15",  
    "date_gmt": "2021-09-03T19:43:15",  
    "guid": {  
      "rendered": "https://movie-api.cederdorff.com/?p=68"  
    },  
    "modified": "2021-09-03T19:45:57",  
    "modified_gmt": "2021-09-03T19:45:57",  
    "slug": "suicide-squad",  
    "status": "publish",  
    "type": "post",  
    "link": "https://movie-api.cederdorff.com/suicide-squad/",  
    "title": {  
      "rendered": "Suicide Squad"  
    },  
    "content": {  
      "rendered": "\n<p>A secret government agency recruits some  
mission: save the world from the apocalypse.</p>\n",  
      "protected": false  
    },  
    "excerpt": {  
      "rendered": "<p>A secret government agency recruits some  
mission: save the world from the apocalypse.</p>\n",  
      "protected": false  
    },  
    "author": 1,  
    "featured_media": 71,  
    "comment_status": "open",  
    "ping_status": "open",  
    "sticky": false,  
    "template": "",  
    "format": "standard",  
    "meta": [],  
    "categories": [  
      7,  
      5,  
      1  
    ],  
    "tags": [],  
    "acf": {  
      "description": "<p>A secret government agency recruits some  
mission: save the world from the apocalypse.</p>\n",  
      "year": 2021  
    }  
  }]
```

```
// append movies to the DOM  
function appendMovies(movies) {  
  let htmlTemplate = "";  
  for (let movie of movies) {  
    htmlTemplate += /*html*/ `  
      <article>  
        <h2>${movie.title.rendered} (${movie.acf.year})</h2>  
          
        <p>${movie.excerpt.rendered}</p>  
      </article>  
  `;  
}  
document.querySelector('#movies-container').innerHTML = htmlTemplate;
```

INSPECT STRUCTURE, ARRAYS, OBJECTS & PROPERTIES

SINGLE PAGE WEB APP TEMPLATE
127.0.0.1:5501

MOVIES

Search

Suicide Squad (2016)



A secret government agency recruits some of the most dangerous incarcerated super-villains to form a defensive task force. Their first mission: save the world from the apocalypse.

Spider-Man: No Way Home (2021)



For the first time in the cinematic history of Spider-Man, our friendly neighborhood hero is unmasked and no longer able to separate his normal life from the high-stakes of being a Super Hero. When he asks for help from Doctor Strange, the stakes become even more dangerous, forcing him to discover what it truly means [...]

MOVIES CATEGORIES

Elements Console Sources Network 4 Default levels

5 Issues: 4 1

Array(8) i

0:

- acf: {description: '<p>A secret government agency recruits some of the most dangerous incarcerated super-villains to form a defensive task force. Their first mission: save the world from the apocalypse.', author: 1}
- categories: (3) [7, 5, 1]
- comment_status: "open"
- content:
 - protected: false
 - rendered: "\n<p>A secret government agency recruits some of the most dangerous incarcerated super-villains to form a defensive task force. Their first mission: save the world from the apocalypse.
- [[Prototype]]: Object
- date: "2021-09-03T19:43:15"
- date_gmt: "2021-09-03T19:43:15"
- excerpt: {rendered: '<p>A secret government agency recruits some of the most dangerous incarcerated super-villains to form a defensive task force. Their first mission: save the world from the apocalypse.'}
- featured_media: 71
- format: "standard"
- guid: {rendered: 'https://movie-api.cederdorff.com/?p=68'}
- id: 68
- link: "https://movie-api.cederdorff.com/suicide-squad/"
- meta: []
- modified: "2021-09-03T19:45:57"
- modified_gmt: "2021-09-03T19:45:57"
- ping_status: "open"
- slug: "suicide-squad"
- status: "publish"
- sticky: false

- tags: []
- template: ""
- title: {rendered: 'Suicide Squad'}
- type: "post"

WORDPRESS API ROUTES

Meta data and routes: <https://movie-api.cederdorff.com/wp-json/>

Posts: [https://movie-api.cederdorff.com/wp-json/wp/v2/posts? embed](https://movie-api.cederdorff.com/wp-json/wp/v2/posts?embed)

Categories: <http://movie-api.cederdorff.com/wp-json/wp/v2/categories>

[http://movie-api.cederdorff.com/wp-json/wp/v2/posts?
_embed&categories=4](http://movie-api.cederdorff.com/wp-json/wp/v2/posts?embed&categories=4)

Admin: <https://movie-api.cederdorff.com/wp-admin>

Username: mdu-front

Password: headlessfront01

ADD CATEGORIES

The screenshot shows the WordPress Admin Categories page. On the left, a sidebar menu is visible with options like Posts, Categories (which is currently selected), and others like Media, Pages, and Comments. The main area has a title 'Categories' and a sub-section 'Add New Category'. It includes fields for 'Name' (with a placeholder 'The name is how it appears on your site.'), 'Slug' (with a placeholder 'The "slug" is the URL-friendly version of the name. It is usually all lowercase and contains only letters, numbers, and hyphens.'), 'Parent Category' (set to 'None'), and 'Description' (with a note that it's not prominent by default). To the right, a table lists existing categories:

Name	Description	Slug	Count
Action	—	action	3
Adventure	—	adventure	3
Animation	—	animation	3
Comedy	—	comedy	3
Drama	—	drama	1
Fantasy	—	fantasy	1
Horror	—	horror	1

ADD CATEGORIES TO POSTS

The screenshot shows the WordPress admin dashboard with the "Posts" screen selected. The left sidebar includes links for Dashboard, Posts, All Posts, Add New, Categories, Tags, Media, Pages, Comments (6), Appearance, Plugins, Users, Tools, Settings, Custom Fields, and Collapse menu. The main area displays a list of posts with columns for Title, Author, Categories, Tags, and Date. The posts listed are:

Title	Author	Categories	Tags	Date
Suicide Squad	admin	Action, Adventure, Fantasy	—	Published 2021/09/03 at 7:43 pm
Spider-Man: No Way Home	admin	Action, Adventure	—	Published 2021/09/03 at 7:40 pm
Cruella	admin	Comedy	—	Published 2021/09/03 at 6:47 pm
Frozen 2	admin	Animation, Comedy	—	Published 2020/09/04 at 6:00 am
Birds of Prey	admin	Action, Adventure	—	Published 2020/09/02 at 8:47 am
Frozen	admin	Animation, Comedy	—	Published 2019/09/02 at

DEFINE HTML

```
<!-- categories page -->
<section id="categories" class="page">
  <header class="topbar">
    <h2>Categories</h2>
  </header>
  <select id="select-category" onchange="categorySelected(this.value)">
    <option selected disabled>Select Category</option>
  </select>
  <section id="movies-by-category-container" class="grid-container">
    <article class="empty">
      <p>Please, select category</p>
    </article>
  </section>
</section>
```

FETCH & APPEND CATEGORIES

```
// fetch all genres / categories from WP
async function getCategories() {
    let response = await fetch("https://movie-api.cederdorff.com/wp-json/wp/v2/categories");
    let data = await response.json();
    console.log(data);
    appendCategories(data);
}

getCategories();

// append all genres as select options (dropdown)
function appendCategories(categories) {
    let htmlTemplate = "";
    for (let category of categories) {
        htmlTemplate += /*html*/
        <option value="${category.id}">${category.name}</option>
    };
}
document.querySelector('#select-category').innerHTML += htmlTemplate;
```

DEFINE SELECT ONCHANGE EVENT

```
// category selected event - fetch movies by selected category
async function categorySelected(categoryId) {
  if (categoryId) {
    showLoader(true);
    let response = await fetch(`https://movie-api.cederdorff.com/wp-json/wp/v2/posts?_embed&categories=${categoryId}`);
    let data = await response.json();
    appendMoviesByCategory(data);
    showLoader(false);
  }
}
```

APPEND MOVIES BY CATEGORY

```
// append movies by genre
function appendMoviesByCategory(moviesByCategory) {
    let htmlTemplate = "";
    for (let movie of moviesByCategory) {
        htmlTemplate += /*html*/ `
            <article>
                <h2>${movie.title.rendered} (${movie.acf.year})</h2>
                
                <p>${movie.excerpt.rendered}</p>
            </article>
        `;
    }
    // if no movies, display feedback to the user
    if (moviesByCategory.length === 0) {
        htmlTemplate = /*html*/ `
            <p>No Movies</p>
        `;
    }
    document.querySelector('#movies-by-category-container').innerHTML = htmlTemplate;
}
```



Code
Every
Day