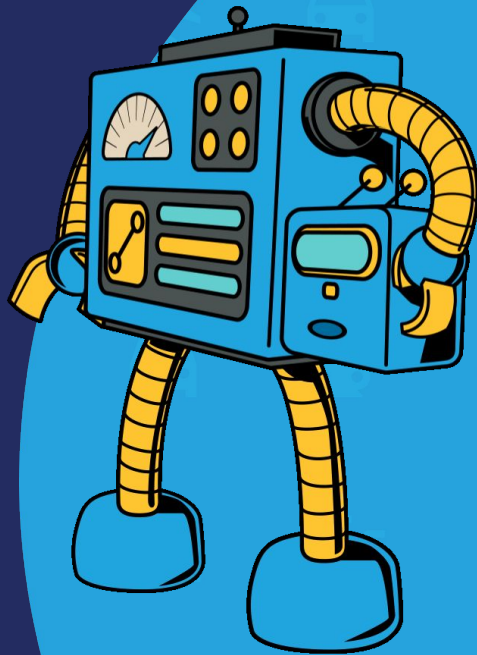
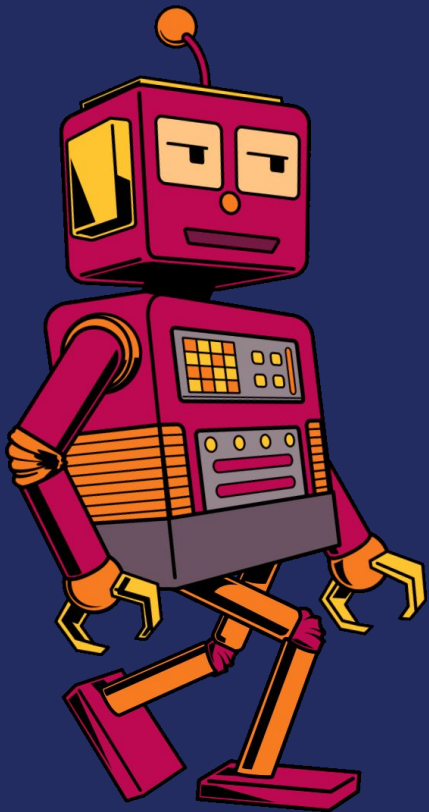




Battle For The Frontend:  
**Low Code vs.  
Headless**





# Speakers



**Allie Jones**

Technical Solutions  
Manager  
Acquia



**Pavlos Daniel**

Senior Technical  
Architect  
Acquia

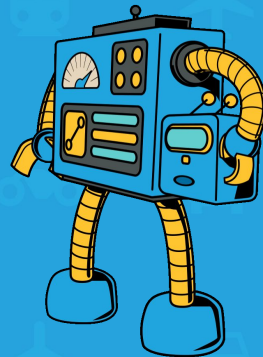
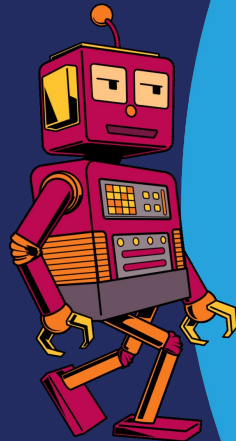


**John Picozzi**

Solutions Architect  
EPAM



# Buzzword Bingo



# Session Overview

## Overview of Concepts

- What is a design system?
- Leveraging design systems

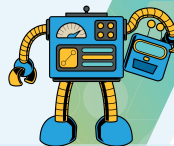
## Low Code

- What is Site Studio?
- Architecture



## Headless

- What is Headless Drupal?
- Software architecture



## Breakout Session

- Focus on the Why
- Get hands-on experience with a starter kit and component design



# Overview of Design Systems and Components



# What is a Design System?

“A Design system is a set of interconnected patterns and shared practices coherently organized. Design systems aid in digital product design and development of products such as applications or websites.” Wikipedia

## Other Names

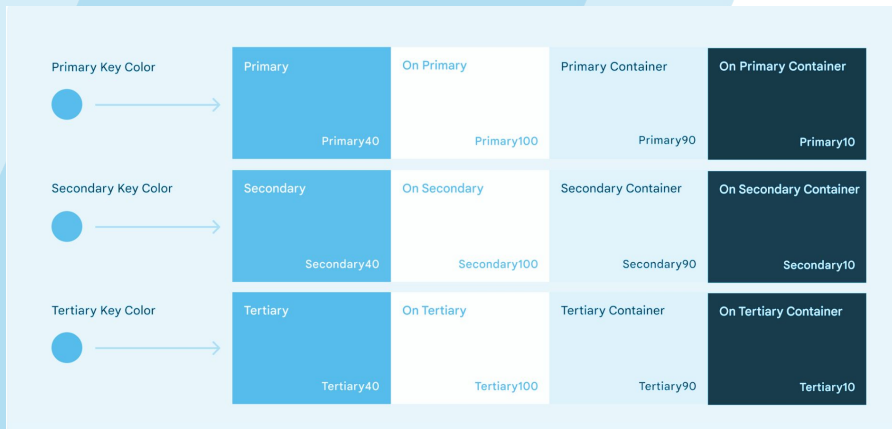
- Component Based Design
- Pattern Library
- Design System
- Living Style Guide
- Atomic Design System



# What Makes Up a Design System?

A system to enforce brand guidelines by creating rules for:

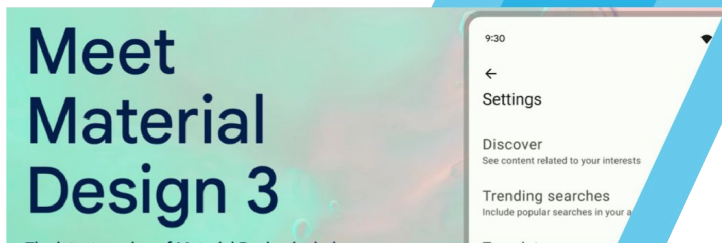
- Font use
- |
- Allowed colors
- |
- Icons libraries
- |
- Standard components
- |
- Overall design principles





# Some Well Known Design Systems

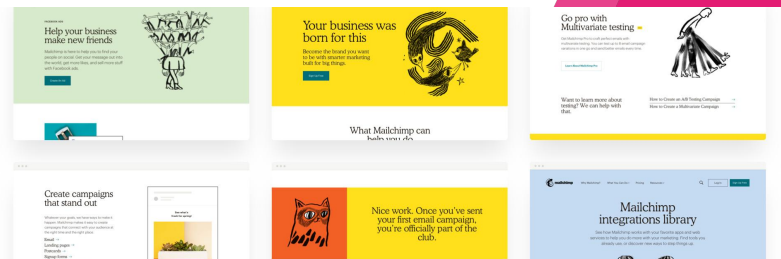
## Google Material



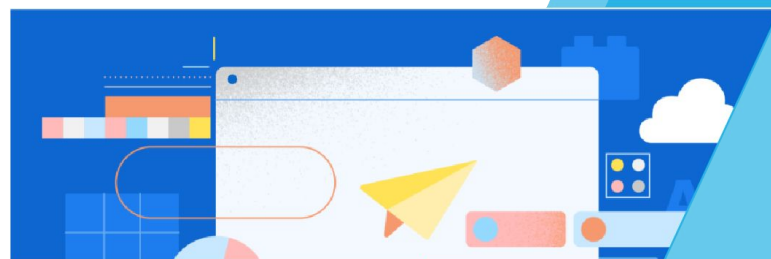
## IBM Carbon



## Mailchimp Identity & Design System



## Spectrum, Adobe's design system





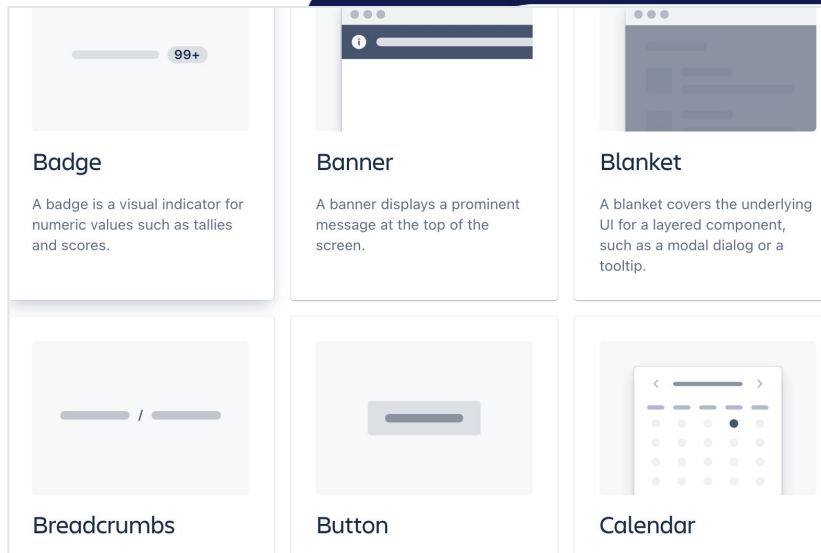
# Components vs. Web Components



# What is a Design Component?

“Components are the reusable building blocks of our design system. Each component meets a specific interaction or UI need, and has been specifically created to work together to create patterns and intuitive user experiences.”

Atlassian (Design System)



# What is a Web Component?



“Web Components are a suite of different technologies allowing you to create reusable custom elements — with their functionality encapsulated away from the rest of your code — and utilize them in your web apps.”

Mozilla

# Why Use A Component Based Approach?

Medium length placeholder heading.

Find out more > Get a demo >

Short placeholder heading

This is placeholder text. If you are reading this, it is here by mistake and we would appreciate it if you could email us with a link to the page you found it on.

Find out more >

Short placeholder heading

This is placeholder text. If you are reading this, it is here by mistake and we would appreciate it if you could email us with a link to the page you found it on.

Find out more >

## 1. Create Once, Use Many

Create a library of branded components to use everywhere

## 2. Consistent Branding Design and UX

Enforce Brand Standards through code

## 3. Reduce Redesign Time

A single component with some basic options for configuration will be more useful than five separate components

## 4. Reduce Testing Overhead & Increase Reliability

Ensure your components meet standards by testing once and using anywhere

## 5. Flexibility of Page Design for Site Builders

Once you have a library of components you can quickly create dynamic pages



# Acquia CMS and Composable Architecture



# What is ACMS?

Acquia Content Management System (Acquia CMS) is an open source, composable CMS for managing digital content and experiences. It is powered by Drupal and optimized for the Acquia platform.

## Key Features

- Vetted Drupal modules
- Acquia Provided Security patches and updates
- Opinionated Code Architecture
- Content Model and Demo Content
- New starter kit wizard



**ACQUIA**  
**CMS**

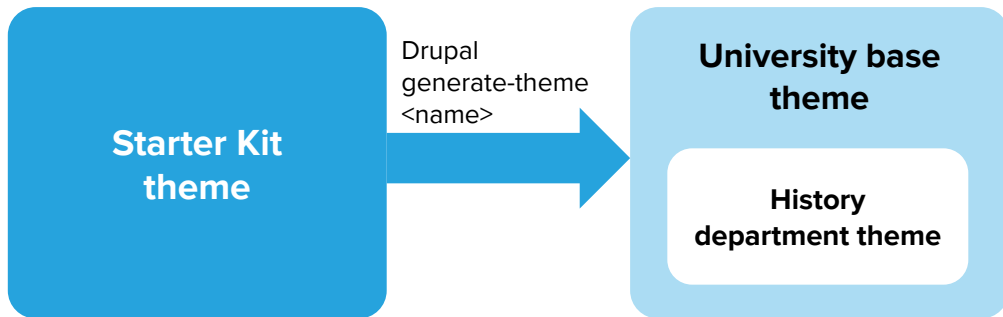
# What is a Starter kit?

Drupal Core Starter kits are a system of templates and styles to be copied at installation instead of loading at runtime.

**Current:**  
**Runtime theme inheritance**



**Starter Kit:**  
**Generate, customize, and inherit**  
**(if needed)**







## Key Points of Starter Kits

### 1. Starting Point

Composable and replaceable templates to start with

### 2. Extendable

Once instantiated you will continue to sub-theme your theme but your starter kit is no longer tied at runtime.

### 3. Composable

Your distribution can have multiple starter kits. Instead of one monolithic install profile your distribution can follow microservice principles.

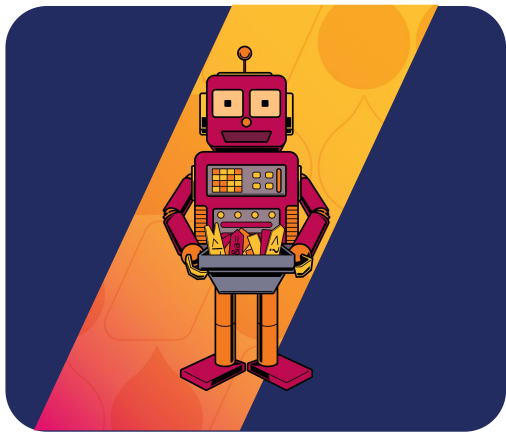
### 4. Decoupled and Upgradeable

Backwards compatibility for themes were not possible before starter kits. Upgrades to Starter kit will not affect your production themes.

### 5. Command line tools for creation

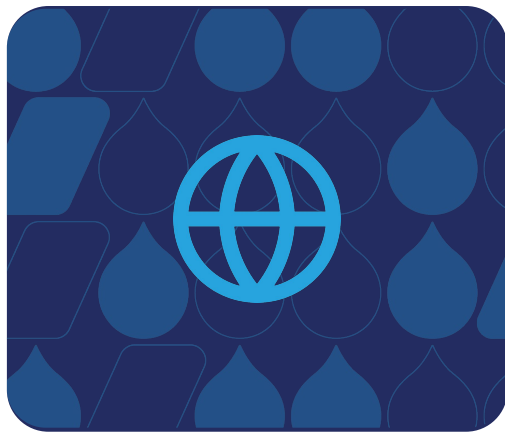
Starter kits provide a CLI command for generating new sites using a starter kit

# Acquia CMS - Starter Kits



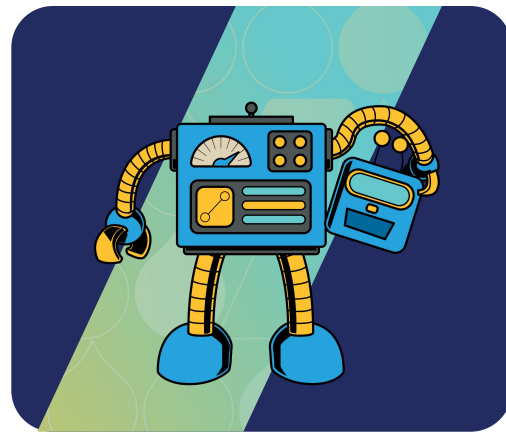
## Enterprise Low Code

For control and creation of layout and display in Drupal powered by Acquia Site Studio



## Community

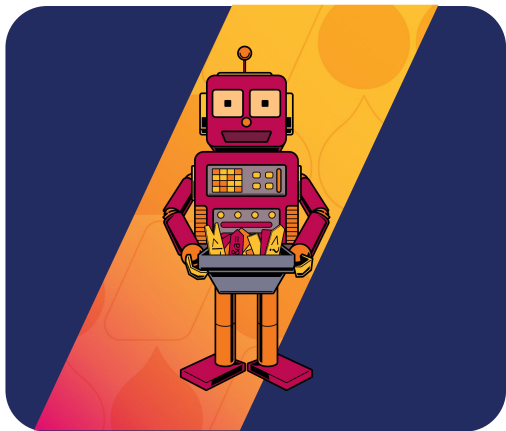
Ideal for greater autonomy over layout and display in Drupal built from community projects



## Headless

Optimized for a headless environment and easy to for frontend developers to use

# Acquia CMS - Starter Kits



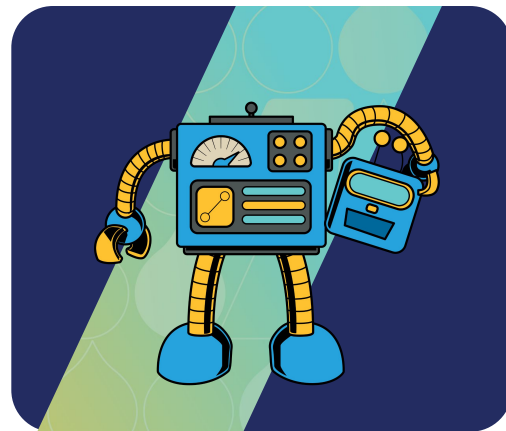
## Enterprise Low Code

For control and creation of layout and display in Drupal powered by Acquia Site Studio



## Community

Ideal for greater autonomy over layout and display in Drupal built from community projects

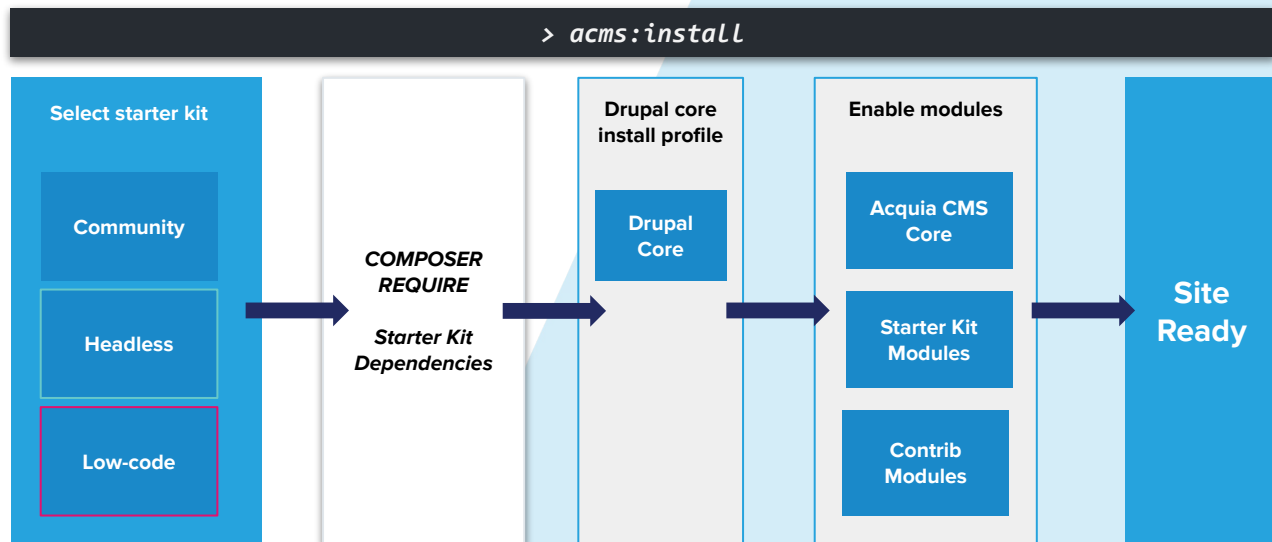


## Headless

Optimized for a headless environment and easy to for frontend developers to use

# Composable architecture powers starter kits

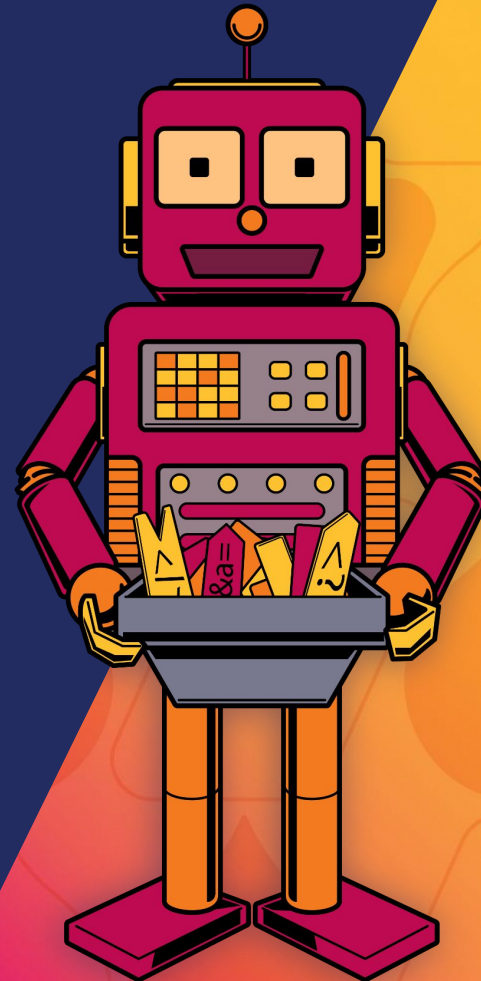
- Starters kits manage an initialization process, not a code distribution.
- Starter kits do not use a custom profile.
- Dependencies are decoupled from distribution.
- Starter kit output is 100% customer owned and customisable.
- Starter kits do not have an upgrade path.





# Enterprise Low Code

## Starter kit

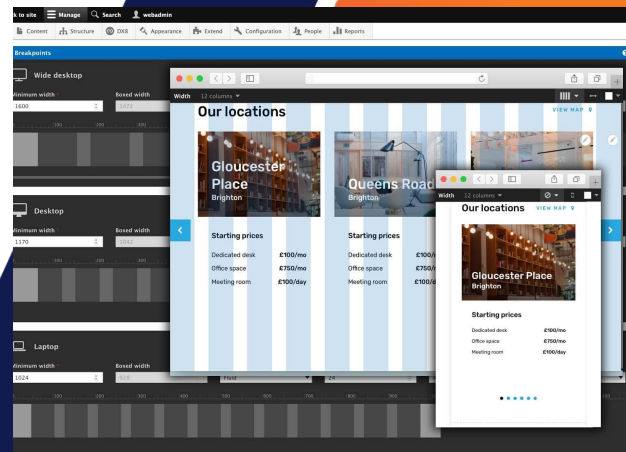


# What is Site Studio?

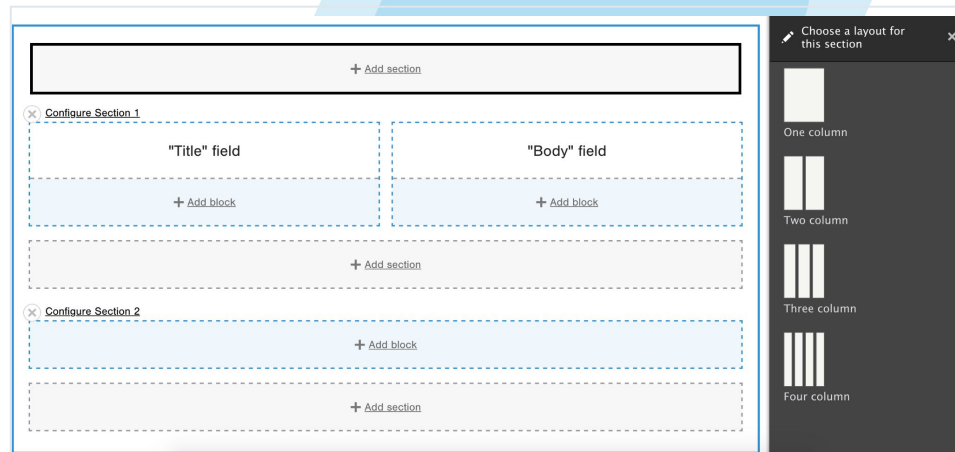
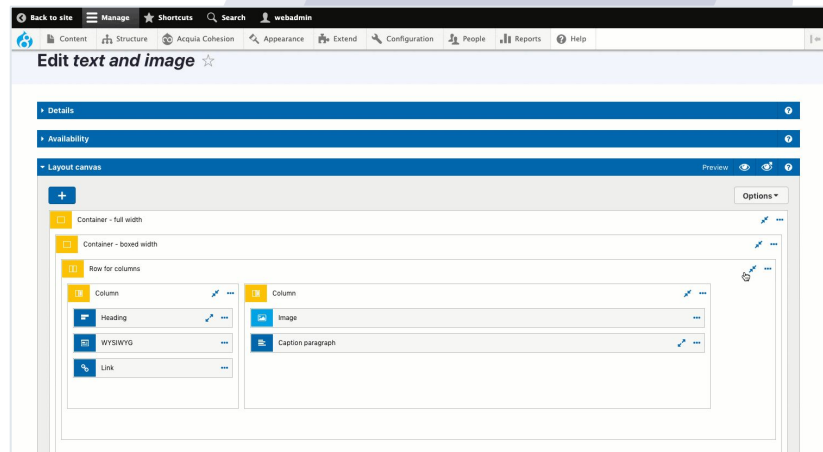
Site Studio is a Drupal add-on service available for Acquia subscribers that empowers non-developers to make layout and design changes inside the CMS.

Designers with little coding knowledge can create within the CMS

- Drag and drop components with Layout Canvas
- Base and custom CSS styles
- Reusable responsive templates
- Component based elements

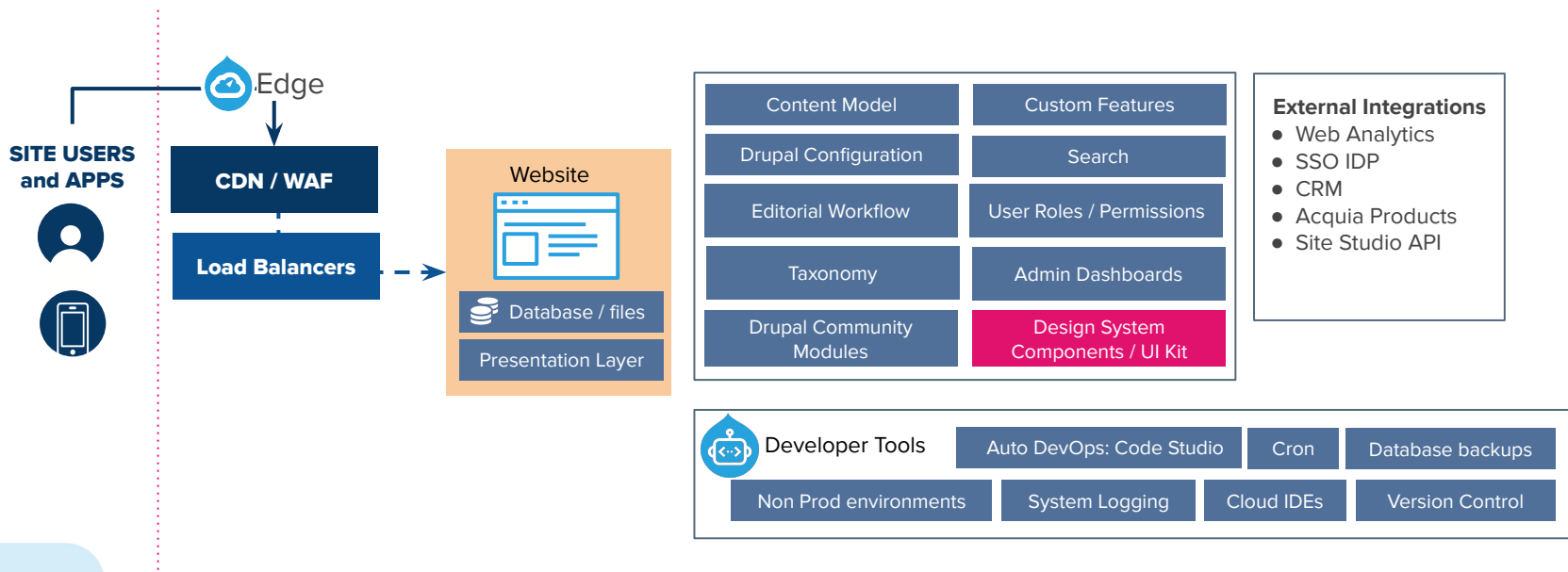


# Layout Canvas vs. Layout Builder



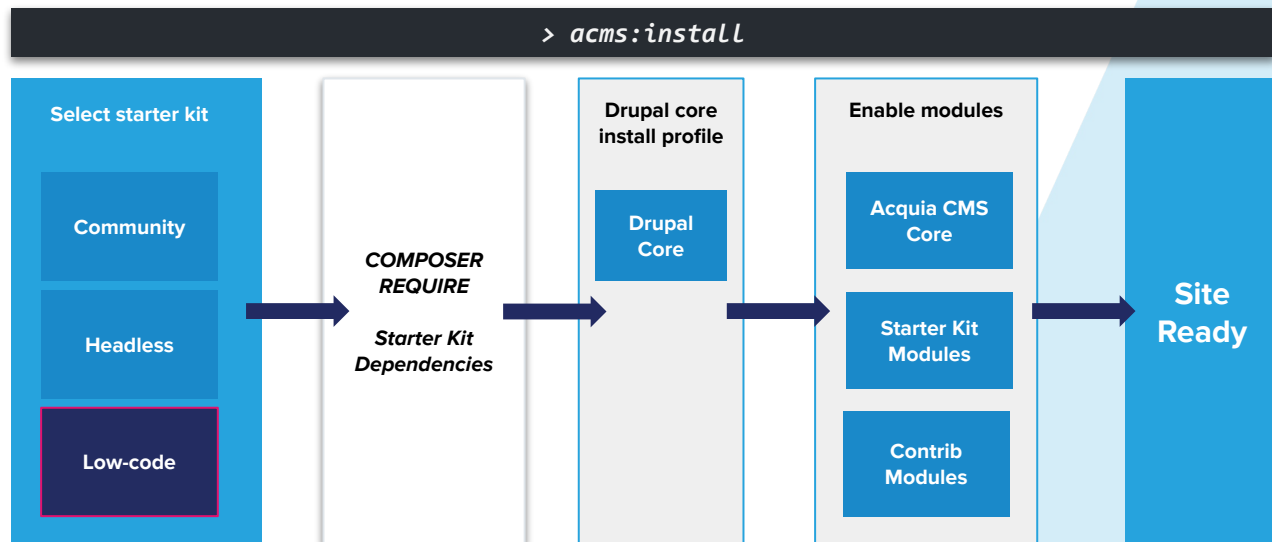
# ACMS Low Code Architecture

## Low Code Application





# Low Code Starter Kit



Acquia Product Integrations

Content Model

Site Building Templates

Site Building Components

Starter Content

Security & Performance  
Standards built in

Best in Class Community  
Modules

# ACMS Site Building Templates



## Low Code Core Components

Integrations

Content Model

Site Building  
Templates

Site Building  
Components

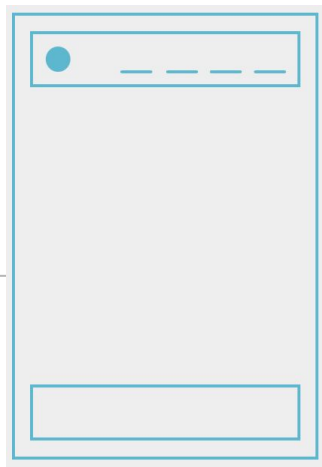
Ui Starter Kit

Security &  
Performance

Best in Class  
Community Modules

## Templates

Control Page layouts across multiple entities



Master Template



Content Template



Component

# ACMS Site Building Components



## Low Code Core Components

Integrations

Content Model

Site Building  
Templates

Site Building  
Components

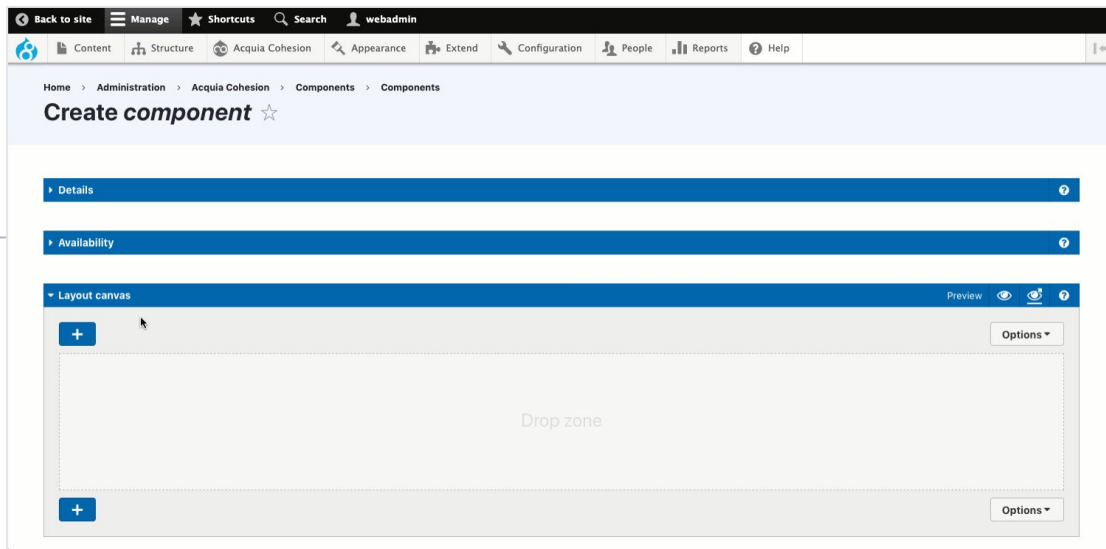
Ui Starter Kit

Security &  
Performance

Best in Class  
Community Modules

## Layout Canvas Page Builder

Drag and drop component library for frontend page building





## Low Code Core Components

## Integrations

## Content Model

## Site Building Templates

## Site Building Components

## UI Starter Kit

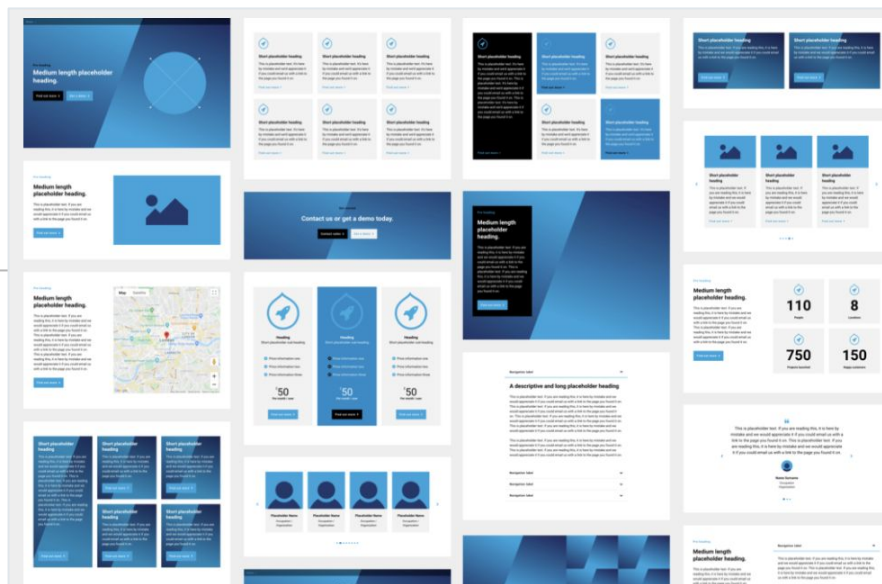
## Security & Performance

## Best in Class Community Modules

## Site Studio Ui Kit

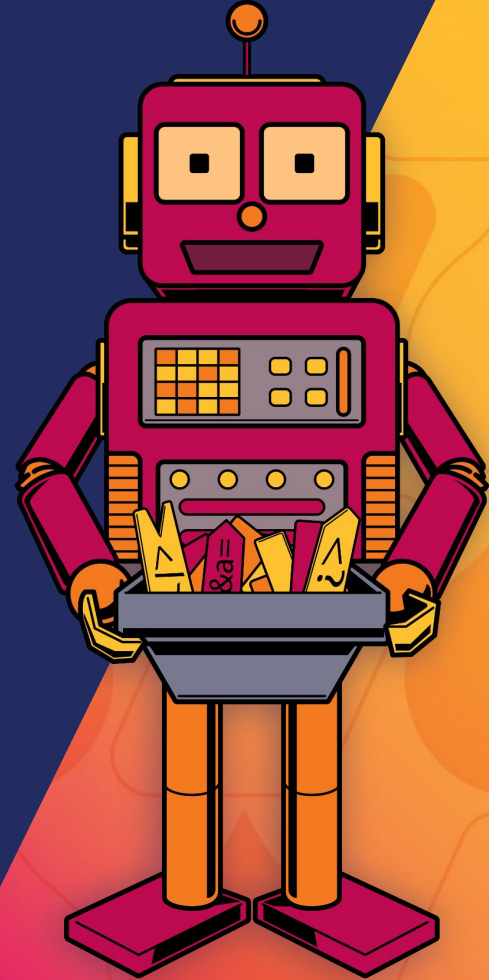
With over 70 pre-built drag and drop components and layouts

With over 70 pre-built drag and drop components and layouts





# Enterprise Low Code Case Studies



# Site Studio Case Study



## Challenge

When the global pandemic hit, 15below had to let thousands of customers know their travel plans were being cancelled. The web site engagement was low and conversion non-existent, and the site was inflexible and unable to pivot to engage its pandemic-aware visitors. The challenge was to build a new digital strategy.

## Solution

- New design system enabled rapid publishing and conversion rate optimization
- Build content and adapt messaging quickly and easily → provide updated site messaging to help support travel industry and its new challenges

## Results

Site was built in 10 days. Within weeks of launch, the small team was able to launch the 3 most successful campaigns in the company's history achieving:

- 2 new customers in 3 months
- Average sales cycle decreased by 83%

# Site Studio Case Study



## Challenge

CGG is a large, matrixed organization with contributors to the website from diverse business lines, geographies and business functions. CGG's website had become dated, was difficult to update, and no longer reflected the environment in which the company operated.

## Solution

- Expedited development timeline: using Site Studio's framework, front-end team developed site components in under 3 months without writing significant amounts of code
- Superior editing interface allowed content authors immense flexibility to deliver dynamic front-end experiences
- Manage content at scale: reusable components eliminated need for duplication of content

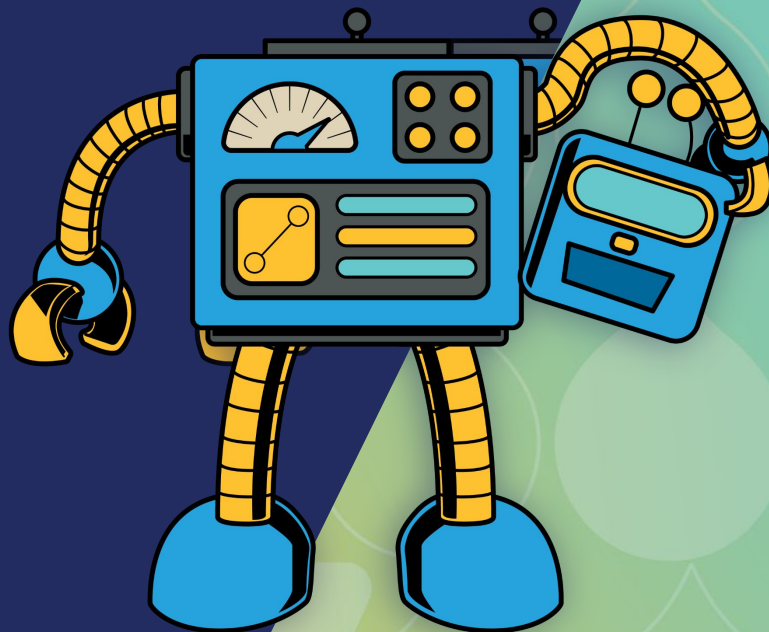
## Results

- 55.75% increase in unique pageviews
- 85.4% increase in average time on page
- Authors can now easily edit content in one place and it's updated throughout the site instantly



# Headless

## Starter kit



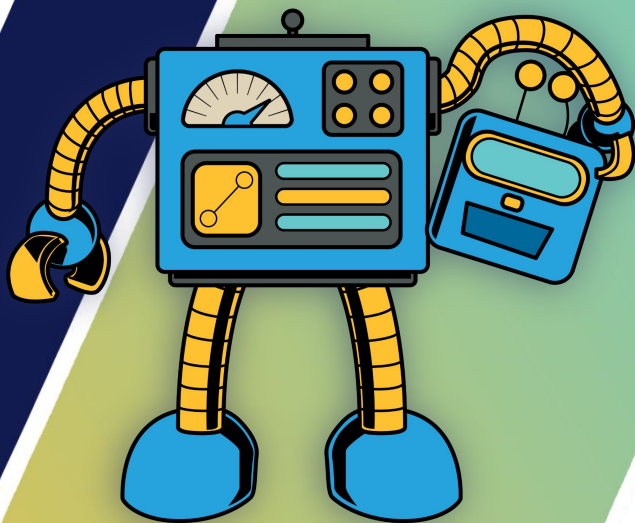


# What is Headless?

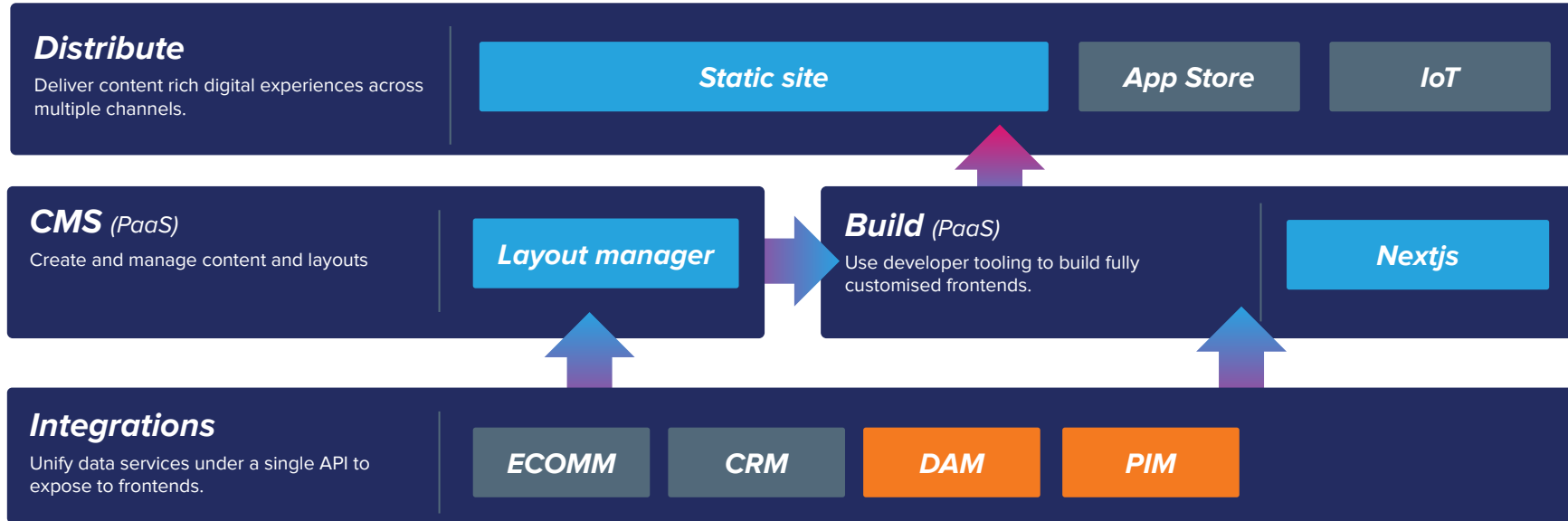
Headless is a loose term that refers to decoupling frontend UX from the backend systems that provide, manage and control data.

## Key Features

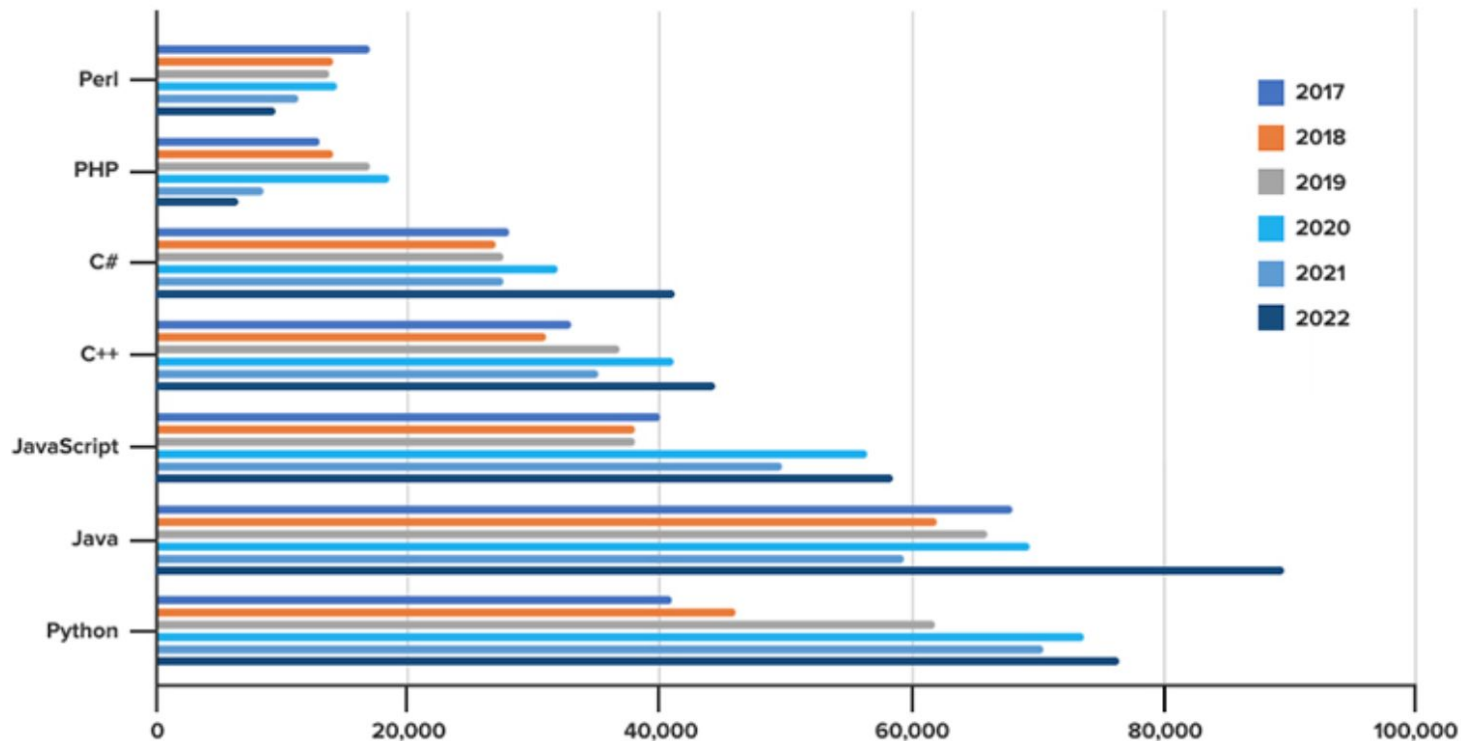
- Single Unified Editorial Experience
- Omni-channel delivery of content
- Composable architecture
- Content Service
- API First



# Composable Architecture



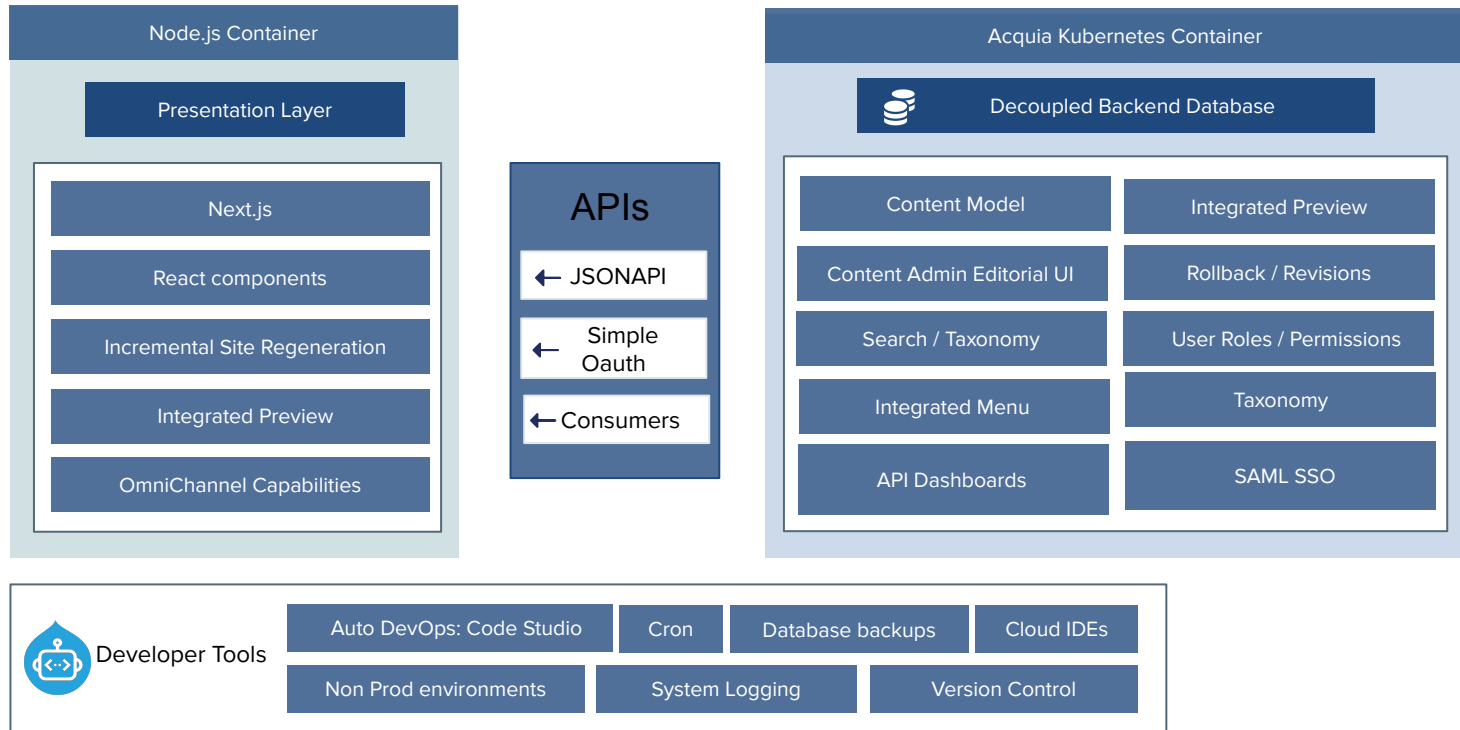
# Top Programming Languages by Year



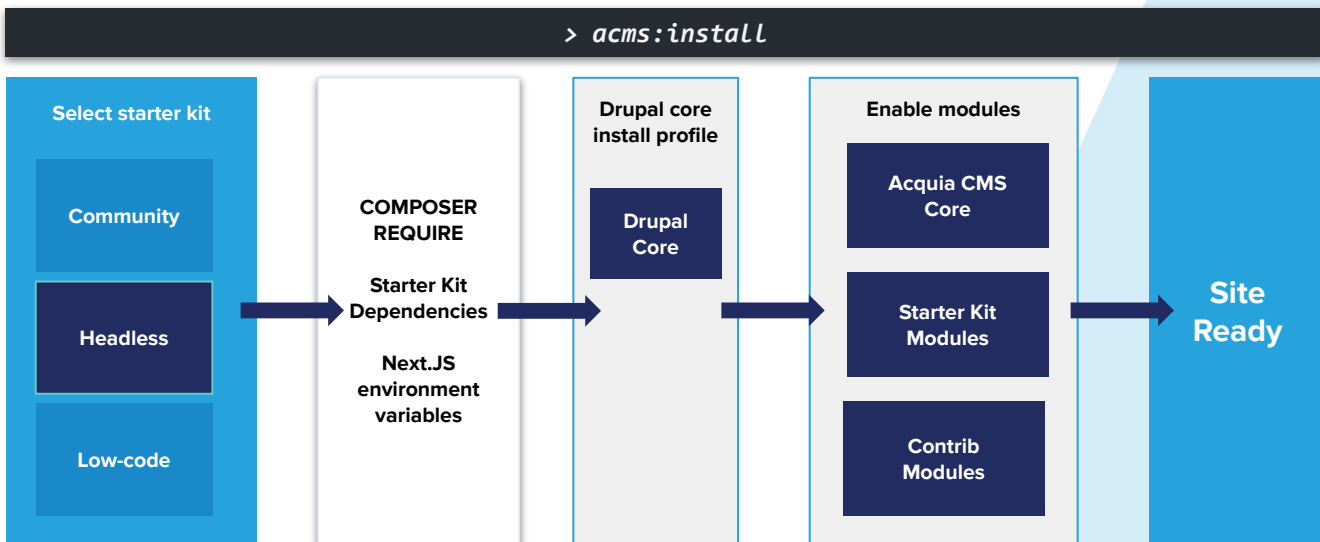
# ACMS Headless Code Architecture



Hybrid Headless Application



# Headless Starter Kit



Acquia Product Integrations

Content Model

Menu Integration

OmniChannel  
Capabilities

Integrated Preview

Security & Performance  
Standards built in

Best in Class Community  
Modules

# ACMS Headless



## Headless Core Components

API Integrations

Content Model

Menu Integration

OmniChannel  
Capabilities

Integrated Preview

Security &  
Performance

Best in Class  
Community Modules

## OmniChannel Capabilities

### Next.js Sites

+ Add Next.js site

ID	Name	Site URL	Operations
<a href="#">headless_site</a>	Blog	<a href="http://learningservicesprod.prod.acquia-sites.com">http://learningservicesprod.prod.acquia-sites.com</a>	Environment variables
<a href="#">large_displays</a>	Large Displays	<a href="http://learningservicesprod.prod.acquia-sites.com/">http://learningservicesprod.prod.acquia-sites.com/</a>	Environment variables
<a href="#">smart_watch_devices</a>	Smart Watch Devices	<a href="http://learningservicesprod.prod.acquia-sites.com">http://learningservicesprod.prod.acquia-sites.com</a>	Environment variables



# ACMS Headless



## Headless Core Components

API Integrations

Content Model

Menu Integration

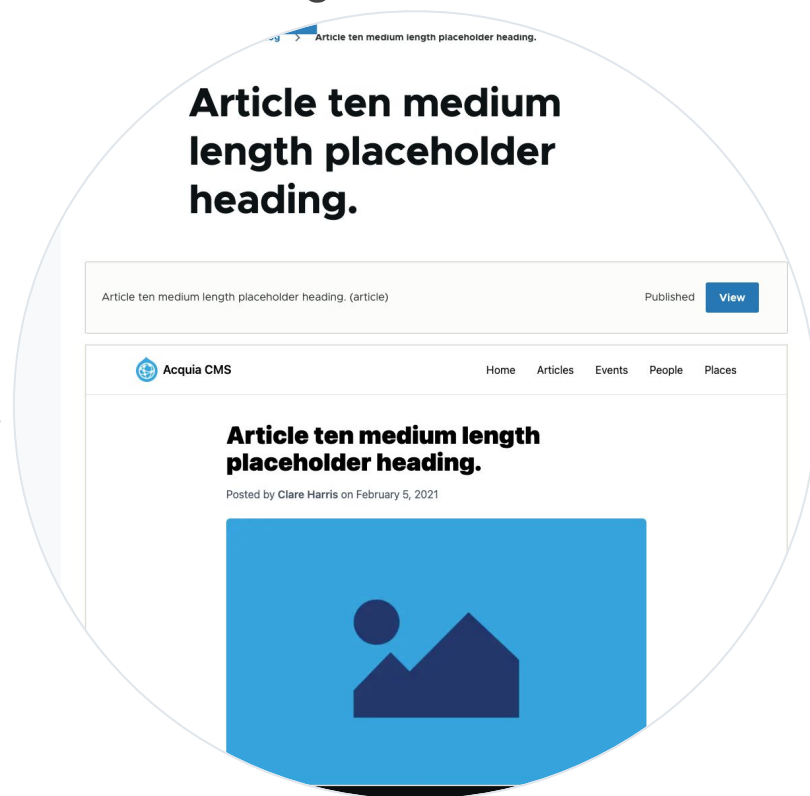
OmniChannel  
Capabilities

Integrated Preview

Security &  
Performance

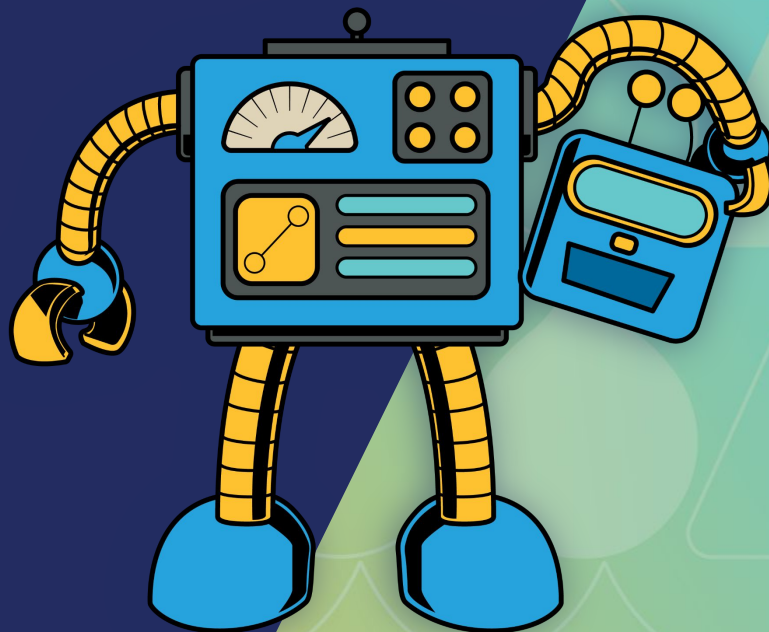
Best in Class  
Community Modules

## Integrated Preview





# Headless Case Studies





# Headless: Use Case

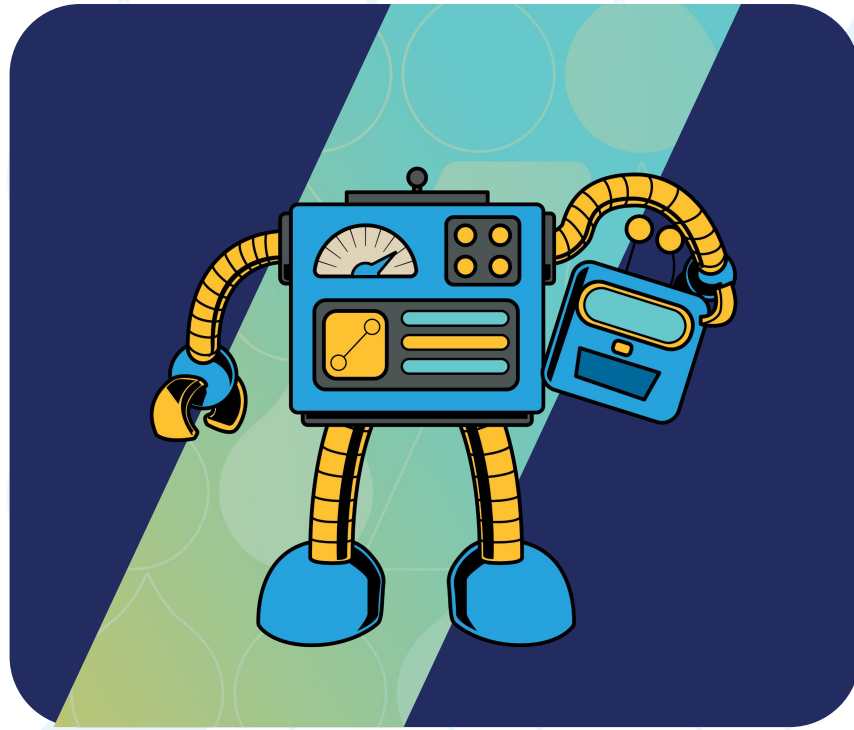
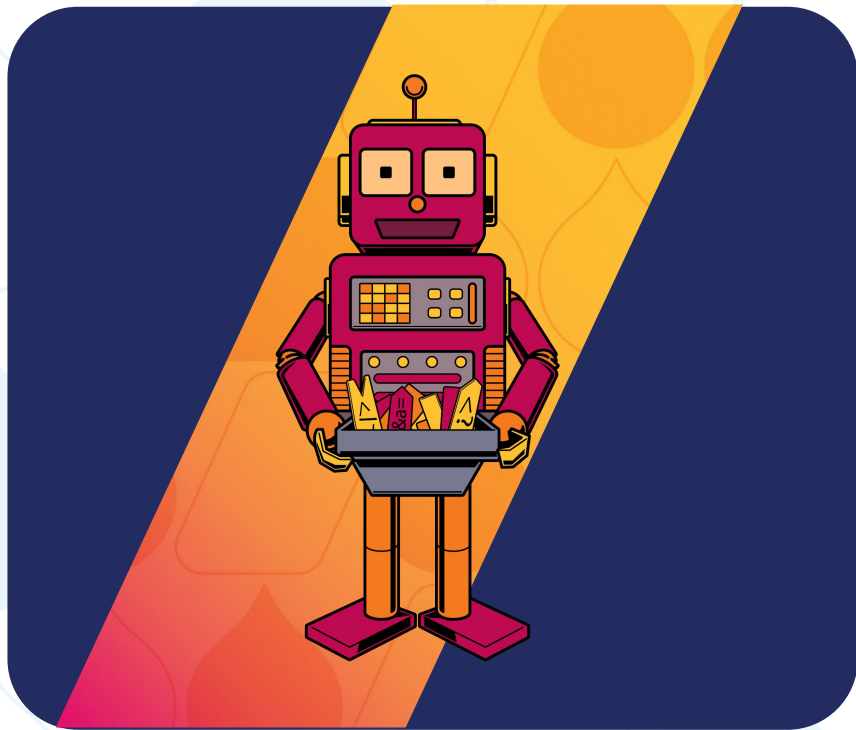
## Challenge

A Large Pharmaceutical company needed a web platform to support its 100+ unique U.S. brand websites. By using Drupal as content management & layout tool, coupled with a new component based design system, they were able to quickly bring sites to market while both reducing the amount of maintenance and giving each site control of its look and feel.

## Key Takeaways

- An API first approach allows for content and layout in the CMS to be used by multiple sites or in a number of front end applications.
- A unified component based design system allows for many improvements and benefits to be applied to all sites seamlessly.
- Server side rendering improves site speed and SEO performance on brand websites
- A Drupal & API based architecture allows for easier integration with 3rd party services

# How to Decide? Focus on Why



# Low Code: Use Cases

## Target Customers

Marketing Team or small development team looking to make dynamic page changes to a production site fast

## Pain Points

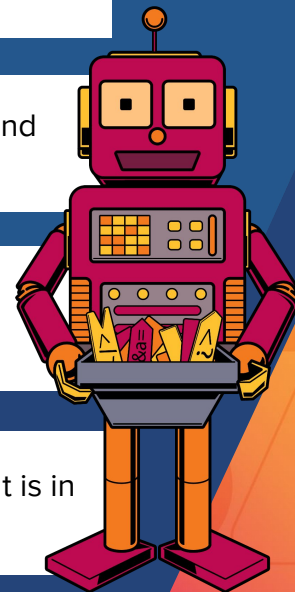
Requires a site redesign by taking over the frontend architecture and styling is now apart of the Drupal Config system

## Complementing Products

Drupal Cloud, Acquia Site Factory

## Exceptions

All in one service for frontend and backend. All component content is in JSON.



# Headless: Use Cases

## Target Customers

Digital experiences built on a microservices architecture.

## Pain Points

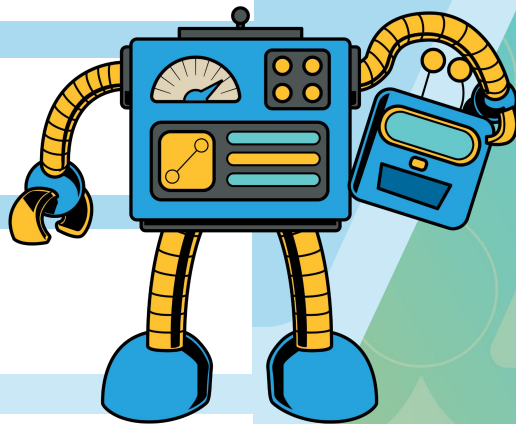
Makes working with Drupal in headless projects easier for content managers and frontend developers.

## Complementing Products

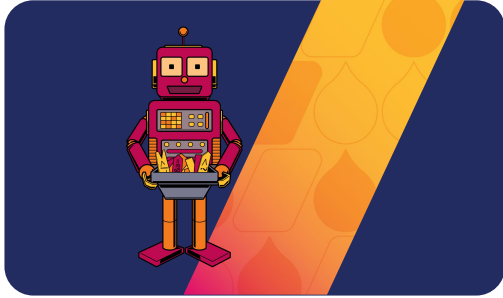
Drupal Cloud, Node.js

## Exceptions

Headless CMS has fewer requirements for integrations making it incompatible with many of our products: Search, Personalization, CDP, etc.

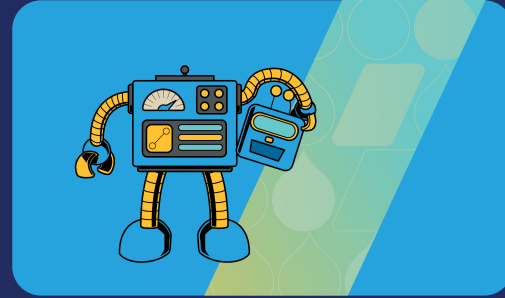


# What are my needs?



## Low Code

- Low code
- Configurable
- Simpler architecture
- Tried and true
- Web pages



## Headless

- High code
- Developer freedom
- Flexible architecture
- Cutting edge
- Apps and IoT

# Important Questions

**Who is going to maintain the application long term?**  
(Think about your team capabilities.)

**How many different channels do I need to display my web content to?**

**What is my main frontend display objective?**

**How much flexibility do I want in my developer and deployment workflow?**

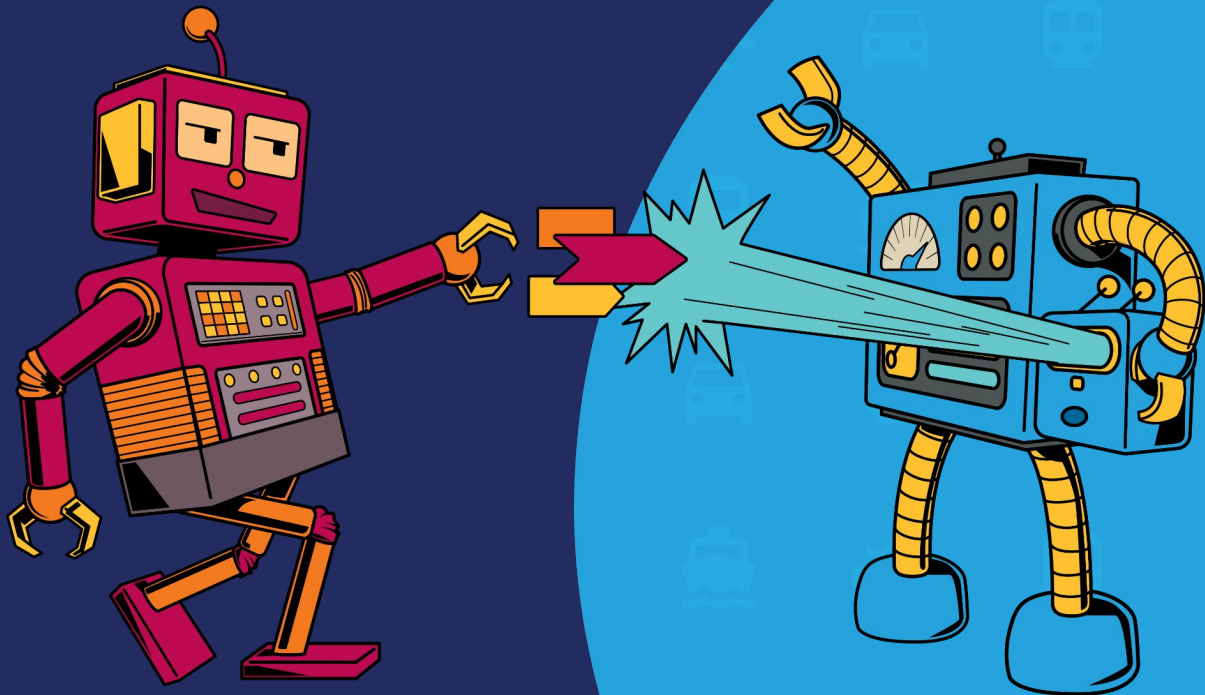
**How do I want to get content in/out of the CMS?**

**How complex is my content?**

**How flexible is my component design system?**



# Quiz



# Quiz Results

**Who is going to maintain the application long term? (Think about your team capabilities)**

**What is my main frontend display objective?**

**How many different channels do I need to display my web content to?**

**How do you want to get content in/out of the CMS?**



Marketing team  
limited developer capabilities

I want to make dynamic frontend  
layouts quickly within the CMS

Single channel that will be  
maintained by your marketing  
team

Content and Authoring Role  
Workflow directly in the  
database



Marketing and Developer team  
Developers with Javascript  
Experience

I want to display my content on  
many different channels

Omnichannel Capabilities

Multiple data sources via APIs



# Quiz Results

**How flexible is my component design system?**

**How complex is my content?**

**How much flexibility do I want in my developer and deployment workflow?**



I want pre built drag and drop components to speed up time to market

Mainly web content based. I want all of my content and frontend to be in all in one system

I want to make my deployment as easy as possible



I want custom components that leverage a javascript frontend framework like React

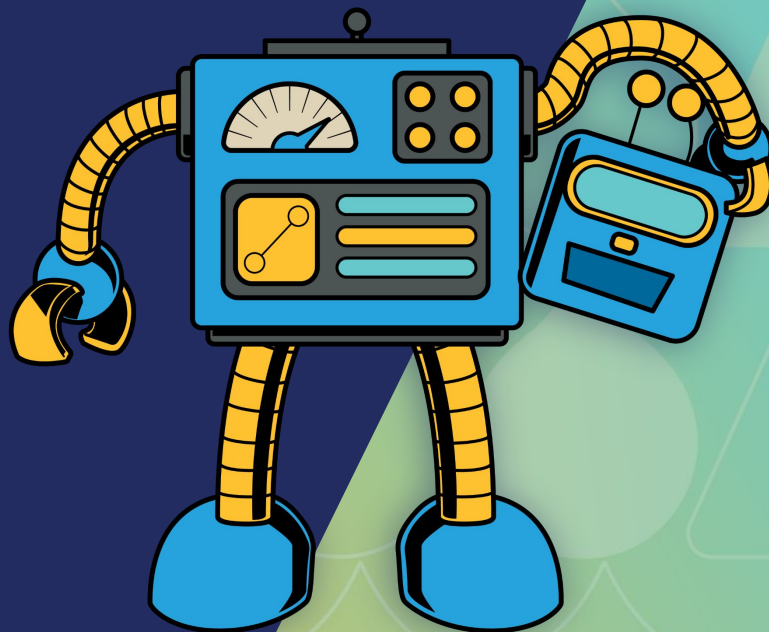
Complex with many microservices

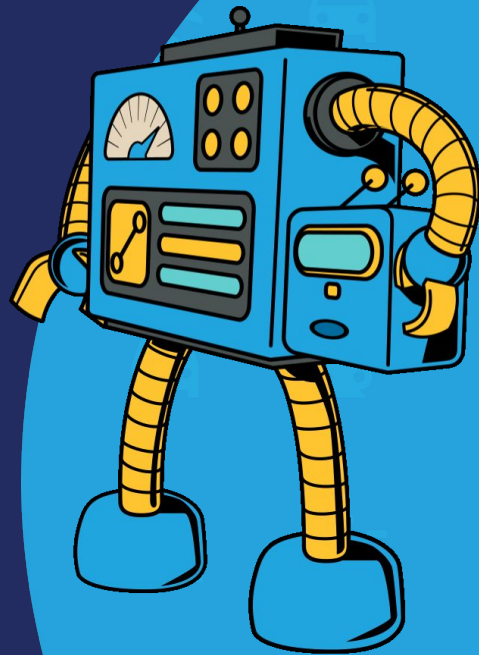
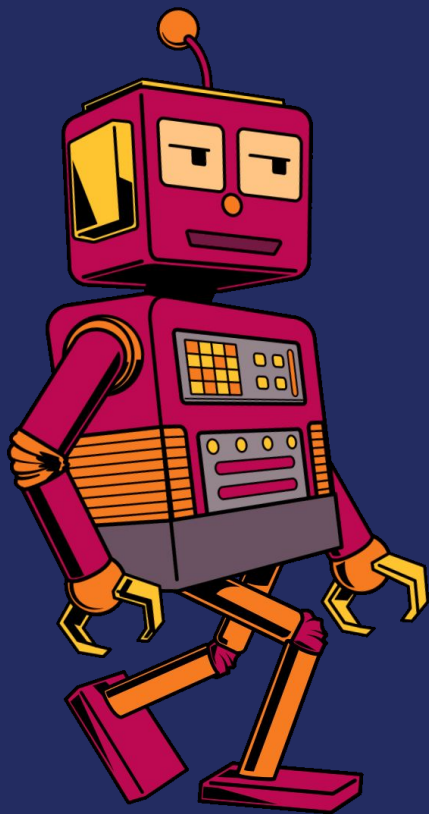
I want complete flexibility and capabilities to make a custom frontend deployment



# Headless Roundtable

Tomorrow @ 3:15





Thank you