OpenSource Headless CMS

FOR CONTENT MANAGEMENT

By ***Sangram Rath*** *for* 20th Edition OPEN SOURCE INDIA

Agenda

- Introduction
- Headless CMS Architecture
- Advantages
- OpenSource Options
- Headless or NOT
- Hands-on w/Squidex
- Deploying Squidex (Backend)
- Manging Content
- Deploying App (Frontend)
- End

Pre-requisites for the Workshop

Laptop & Browser

A laptop with enough resources to run Docker and a modern browser

GitHub Account

A GitHub account to fork the code

Docker & Docker Compose

Latest version of Docker Desktop or Docker CLI with Docker Compose

Sangram Rath

Cloud Architect &

Technology Advisor, Founder - OD10

Over 18 years of experience managing projects and teams across domains, evolving from traditional datacenters and virtualization to cloud computing, DevOps, and security. Special interest in CMS, Web Apps on Cloud.

10+ years on cloud technologies such as AWS, Azure, and Google Cloud Platform.

Certifications - Multiple certifications on AWS, Azure, and Google Cloud Platform and other technologies, such as Kubernetes, Oracle Cloud, VMware, CCSK, and more.

Had the privilege to train thousands of professionals on these technologies since 2013, including professionals from Fortune 500 companies.

Author: Hybrid Cloud Management using RedHat CloudForm

Talks: Open Source Summit (NA and Europe), Digital Ocean Tide, and AzConf.

01

Headless CMS

Introduction

What's a Headless CMS?

A headless CMS is an API driven, backend-only CMS.

Headless CMS Features

Adaptable

Flexibility in development helps deliver a better DX unique to the customers

Future Proof

Change UX, FEs, create new functionalities without worrying about backend

Agnostic

Send/Receive content on any device or endpoint

Open Source

Highly customizable, developer friendly, community support, easy troubleshooting, adaptable, control over security

Read Only Content

FE's typically read content only and through API, hence eliminating security threats from FE.

Faster Time to Market

Optimized content managment, reusability resulting in consistentcy

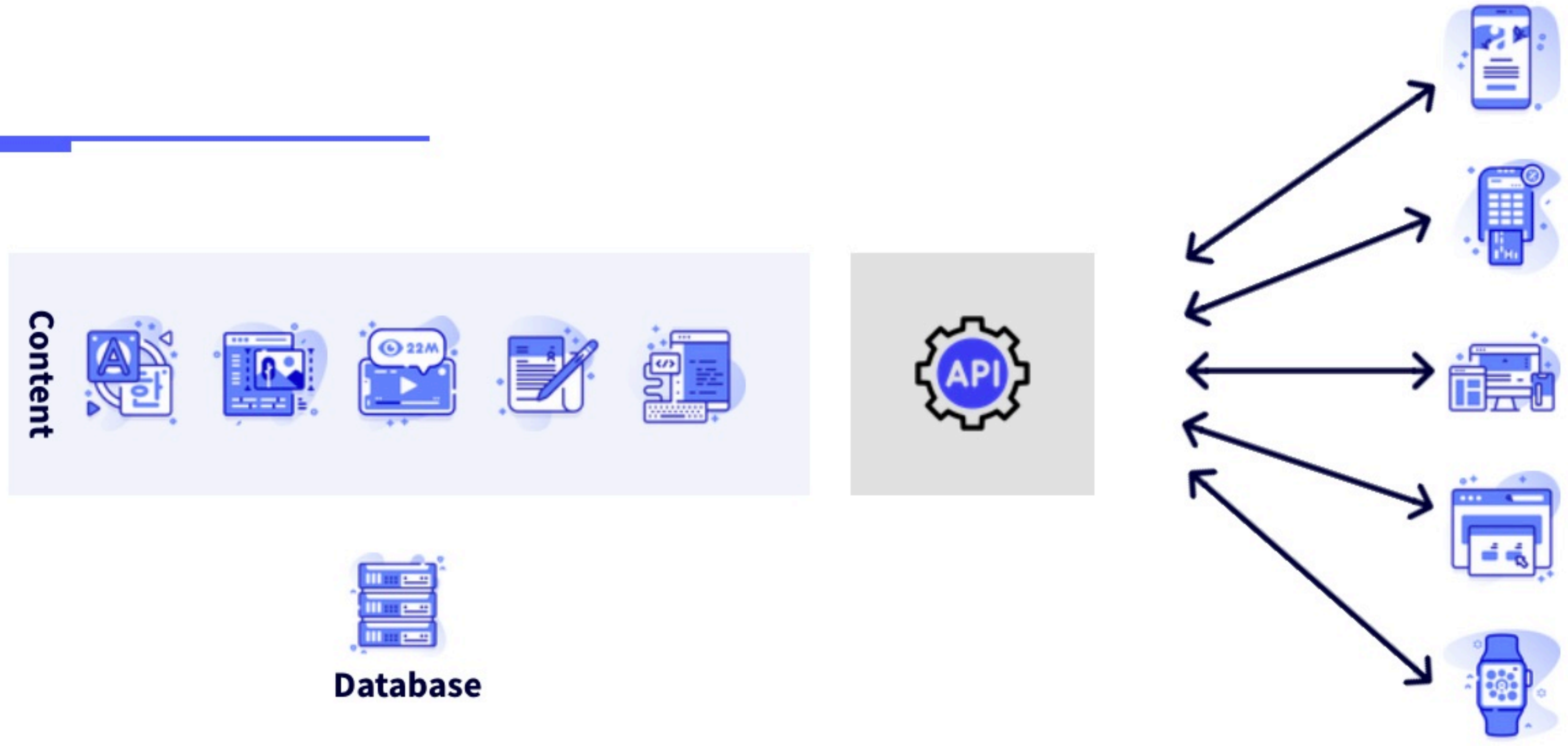
Performance

Client side loading, JAMstack, SSG can improve app performance dramatically

Choice

Choice of Headless CMSes, choice of programming for front ends

Headless CMS Architecture



Advantages

- Omnichannel Content Delivery
- Collaboration
- Consolidation
  - Content
  - Access
- More secure
- Reusability (Modify once)
- Content creator friendly

Advanced Use Cases

CMS for a Voice Assistant

Portuguese
Oxford French Mini Dictionary
JAPANESE FOR
POCKET FRENCH VERB TABLES
Oxford Spanish Mini Dictionary
TEACH YOURSELF
Beginner's german
BBC ACTIVE
Portuguese
talk

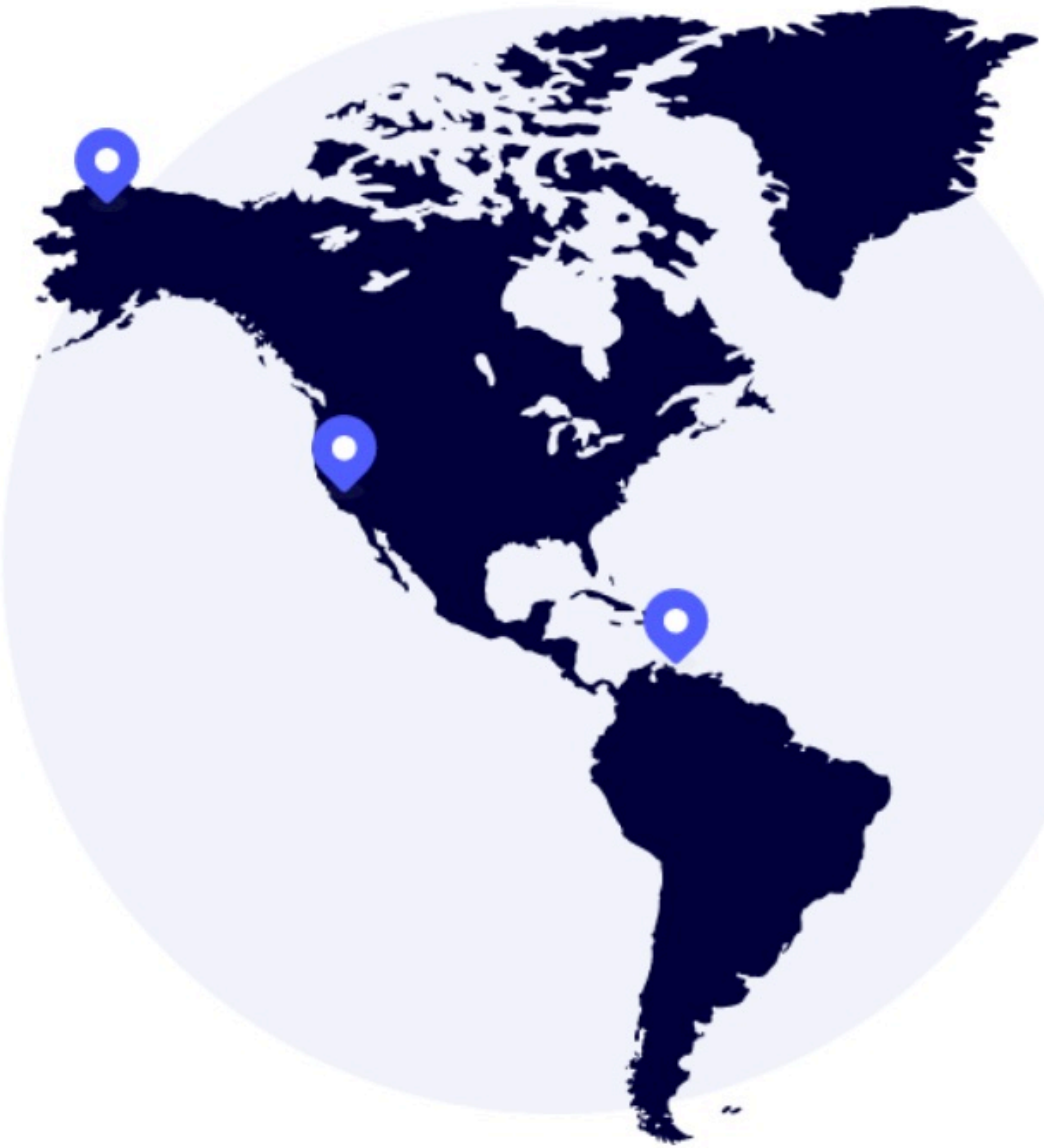
Translation

Personalization

Personalization - Language, Location etc

Headless CMS is great for customer personalization (more important than ever).

Decoupling, reusing, integration capabilities help businesses deliver this.



Security

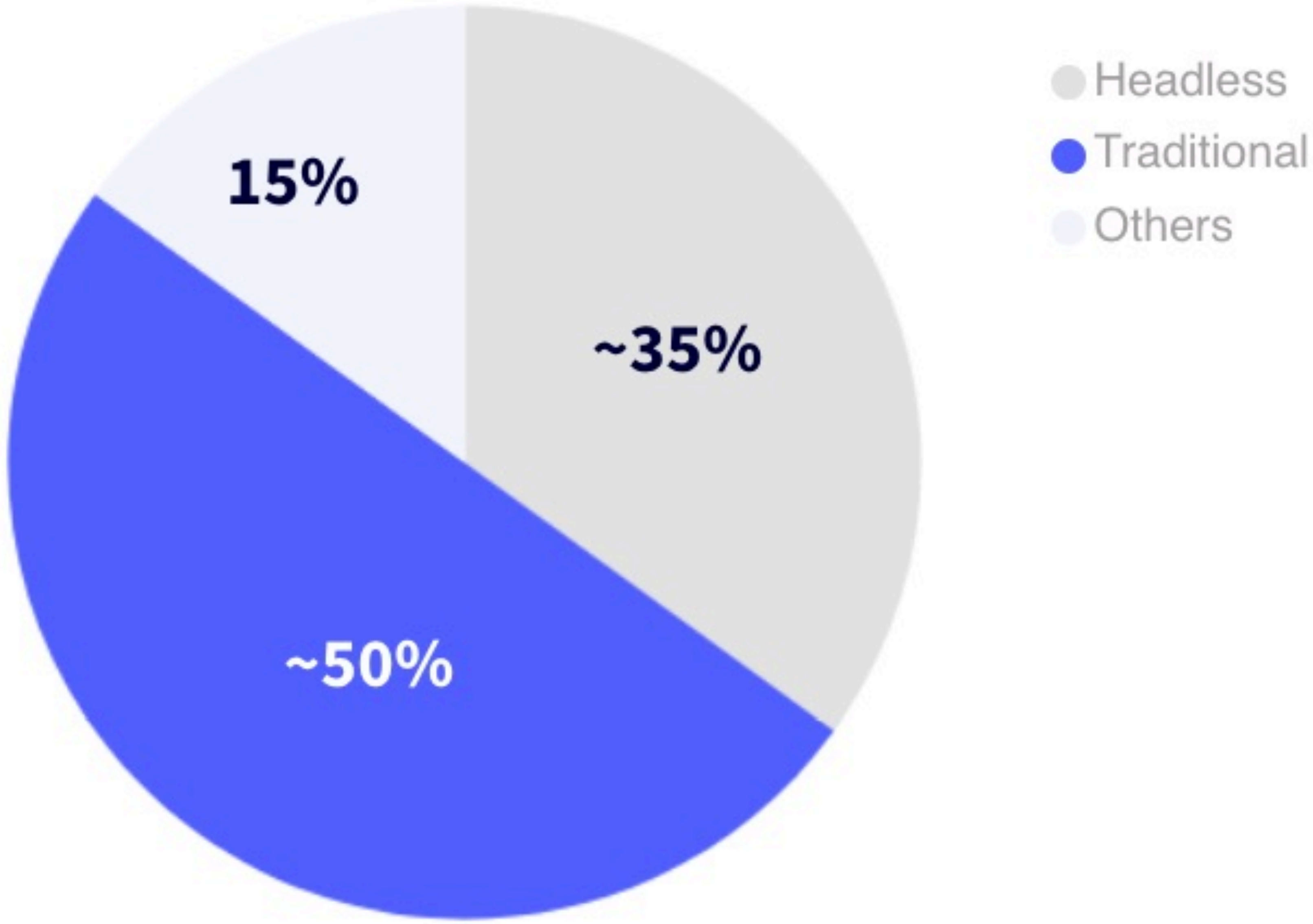
Separate attack surface due to the decoupled architecture.

API access means CMS is not internet-exposed. Hence reduced attack surface.

Less suspectible to DDoS

Implement API best practices, put it behind multiple code layers and a firewall for an even secure setup

Headless CMS Stats



Report by Storyblok

>400%
Projected Growth

Businesses want to provide the bext DXP across a variety of devices and channels

$3.8 billion by 2032

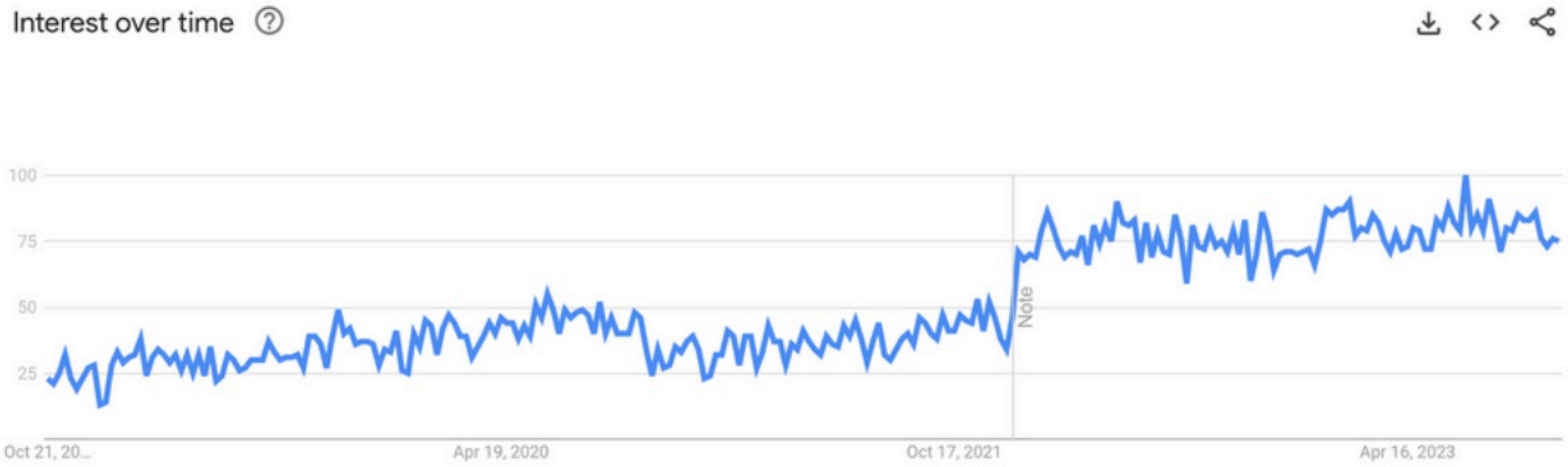
22.1%
CAGR (2022-2032)

Anticipated growth in the Headless CMS market

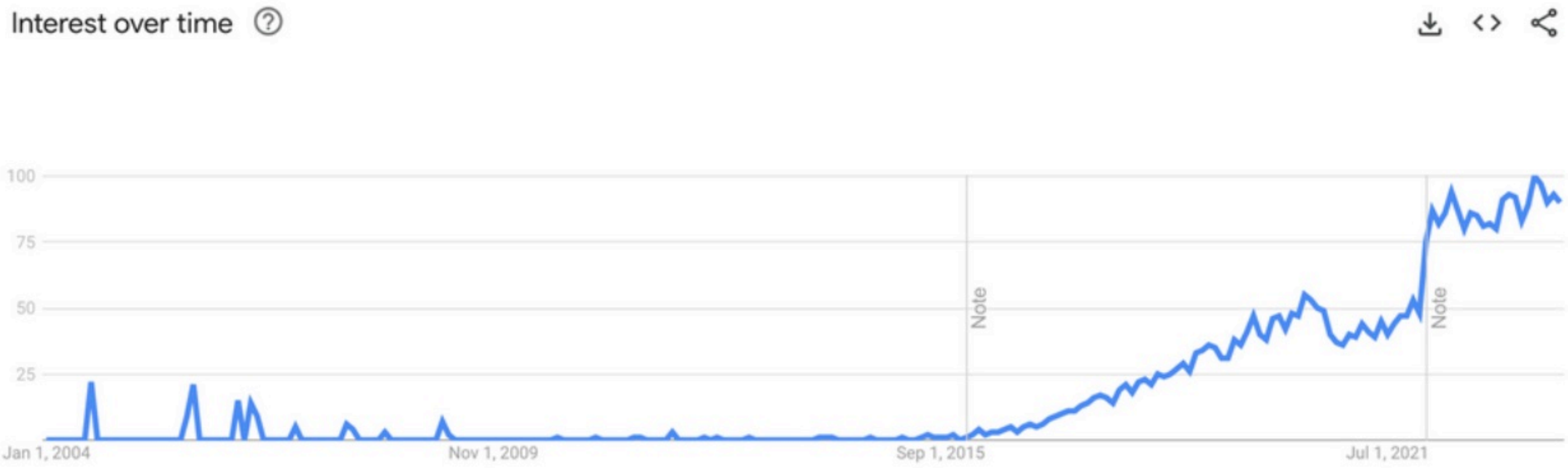
83%
Noticed Improvements

Participants confirmed improved time, budget, productivity, meeting their KPIs, and revenue/growth

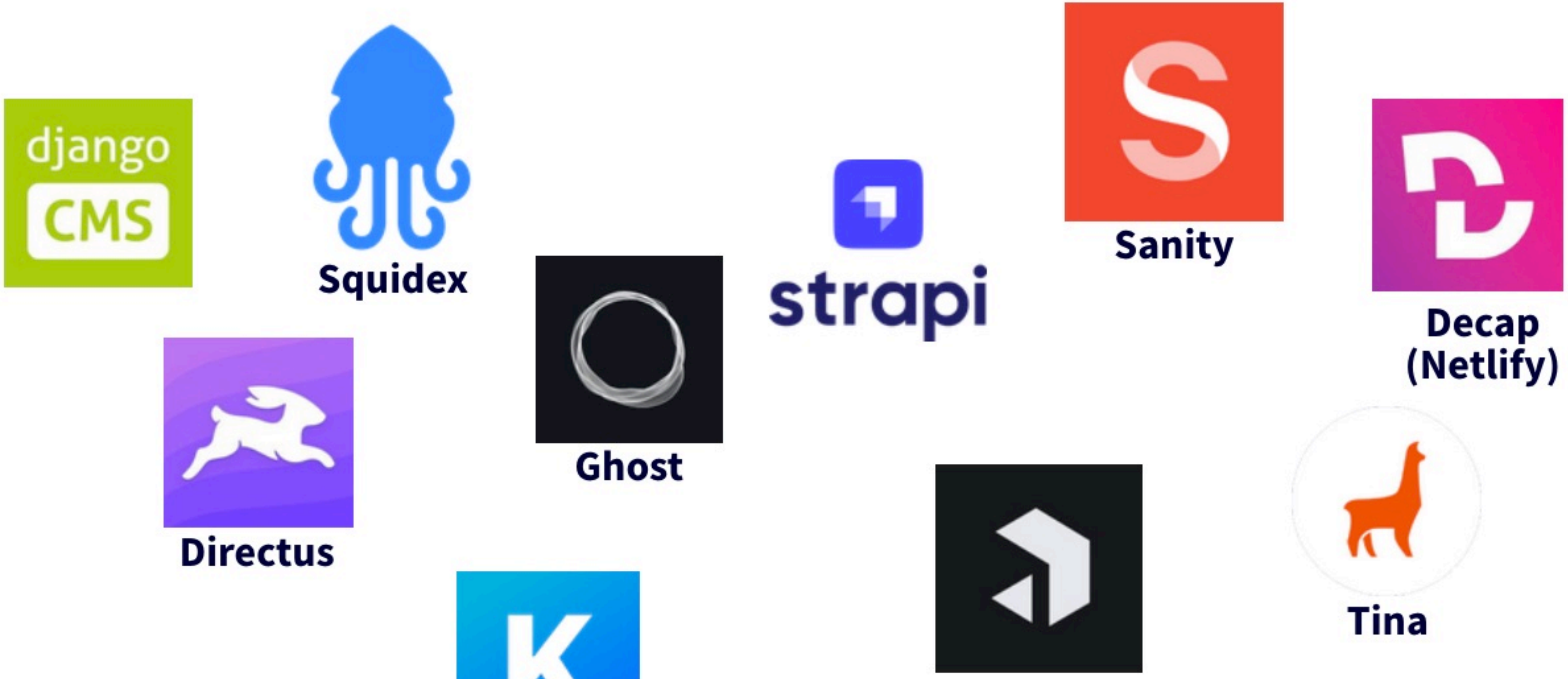
Google Trends - 5 Years



Google Trends - Since 2004



Open Source Headless CMS Software



Squidex

Sanity

Decap (Netlify)

Directus

Ghost

Tina

Keystonejs

Headless or Not

Headless

Presentation of the Web Page is decoupled from Content Management

Only content creation, management and delivery.

Mostly API based

+

VS

Traditional

Presentation and content management are tightly coupled

CMS does everything

+

Headless or Not

Headless

Presentation of the Web Page is decoupled from Content Management

Only content creation, management and delivery.

Mostly API based

+

VS

Traditional

Presentation and content management are tightly coupled

CMS does everything

+

Headless or Not

Headless

- Frequent content change
- Skills and resources are not a challenge
- Multi platform apps
- Agility, performance and scalability are requirements

+

VS

Traditional

- Content is relatively static
- Limited development capability and small teams
- App targets a specific platform
- Simple app, predictable use

+

02

WorkShop

Hands-on Headless CMS with Squidex

WorkShop Overview

The headless CMS used for the workshop is **Squidex** (https://squidex.io/). It is a .NET Core based open source headless CMS that uses MongoDB.

The workshop covers two approaches to using Squidex. One that shows how to self-host and another that shows the SaaS option.

WorkShop 01
Self-Hosting Squidex (and FrontEnd) using Containers

The instructions along with the code is hosted in GitHub.

Refer to the README file for instructions.

https://github.com/sangramrath/OSIWorkShop2023

WorkShop 02
SaaS based Squidex (and FrontEnd on Netlify)

The instructions along with the code is hosted in GitHub.

Refer to the README file for instructions.

https://github.com/sangramrath/OSIWorkShop2023-SaaS

Thanks!