

Choosing a headless application (CMS) has clear benefits, too.

- Flexibility
- Multi-channel/platform delivery
- Improved performance
- Ease of integration
- Separation of concerns at a system-level

Headless CMS

Content
Content types
Digital assets
Sites
Projects
Versions

GraphQL

REST API

SDKs



We're bringing composable architecture to our market-leading CMS

Orchestrate

Transform the way your team plans, creates and publishes content, and accelerate time-to-market.



→ Giving you **complete flexibility** with the option to choose either SaaS or PaaS

→ With the ability to bring **all your data into 1 place** for streamlined delivery, search & integrations

→ And user-friendly experience creation that's **purpose-built for marketers**

*Currently in beta. [Join the waitlist](#)

**Coming 2024

Choose your architecture, choose your delivery method



Ultimate flexibility & freedom of choice
to compose the solution that **fits your
needs perfectly**

*Currently in beta. [Join the waitlist](#)

What's the difference?

Headless Channels & Front-end Platforms



Headless Channels & Front-end Platforms

Content Delivery
(via Optimizely Graph)

Content Delivery
(via Optimizely Graph)

Content Management
(via Optimizely CMS)

PaaS Core

Content Items

Highly Customizable (optional)

MVC Head (optional)

SaaS Core
Coming H2 2024

Content Items

Versionless

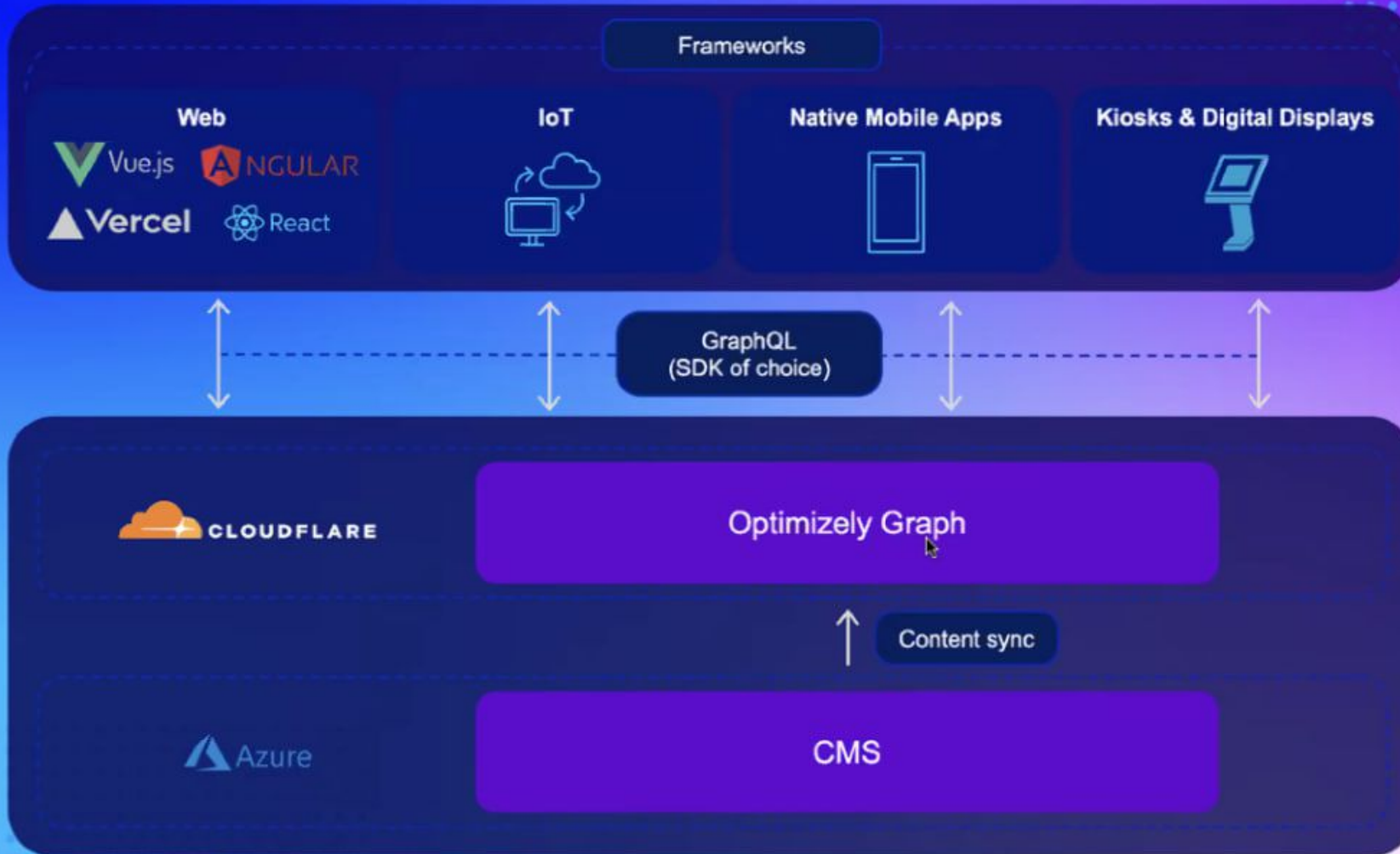
Asset Storage
(via Optimizely DAM)

Asset Storage
(via Optimizely DAM)

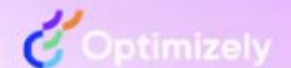
Planning, Collaboration & Workflow
(via Optimizely CMP)

Planning, Collaboration & Workflow
(via Optimizely CMP)

Headless reference architecture



Partner or Customer



Content Modeler

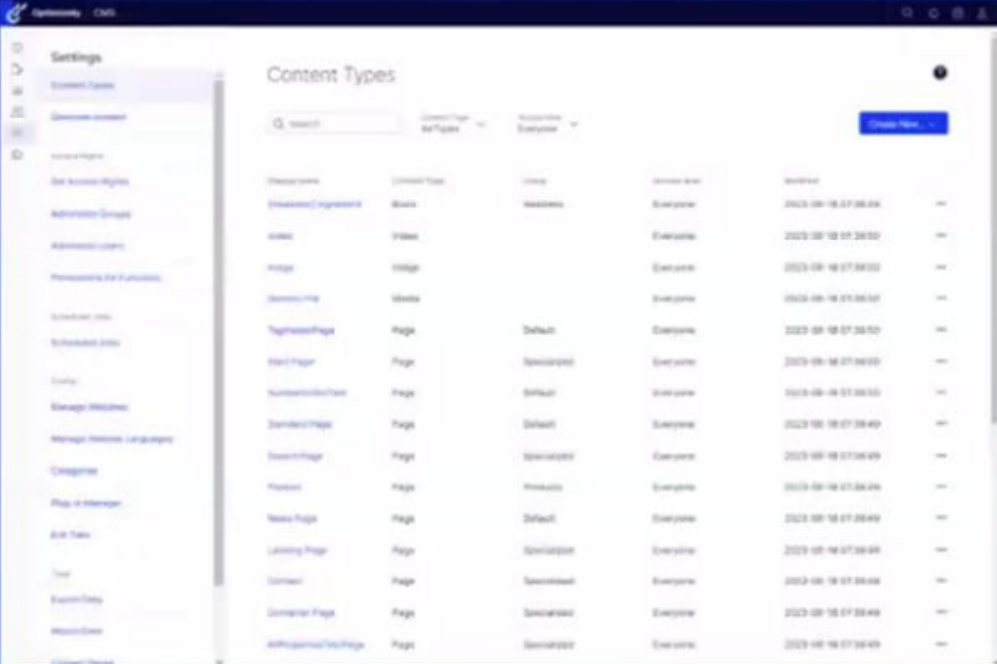
Live

Enhanced Content Modelling

We're bringing content modelling directly into the CMS user interface, providing greater flexibility and choice when it comes to defining your content models.

The enhanced content modelling for Optimizely CMS will provide technology-agnostic APIs as well as a fully featured user interface, enabling both developers and marketers to define the structure of your content.

This feature is already included in SaaS Core and will be available in the PaaS version of our CMS soon.



The screenshot shows the 'Content Types' management page in the Optimizely CMS interface. On the left is a 'Settings' sidebar with various configuration options. The main area displays a table of content types with columns for Name, Content Type, Icon, Status, Version, and Created. A 'Create New...' button is visible in the top right corner of the table area.

Name	Content Type	Icon	Status	Version	Created
Blog Article Page	Page		Active	1.0.0	2023-09-18 07:38:04
Blog Article Page	Page		Active	1.0.0	2023-09-18 07:38:04
Blog Article Page	Page		Active	1.0.0	2023-09-18 07:38:04
Blog Article Page	Page		Active	1.0.0	2023-09-18 07:38:04
Blog Article Page	Page		Active	1.0.0	2023-09-18 07:38:04
Blog Article Page	Page		Active	1.0.0	2023-09-18 07:38:04
Blog Article Page	Page		Active	1.0.0	2023-09-18 07:38:04
Blog Article Page	Page		Active	1.0.0	2023-09-18 07:38:04
Blog Article Page	Page		Active	1.0.0	2023-09-18 07:38:04
Blog Article Page	Page		Active	1.0.0	2023-09-18 07:38:04

Optimizely Graph .NET SDK



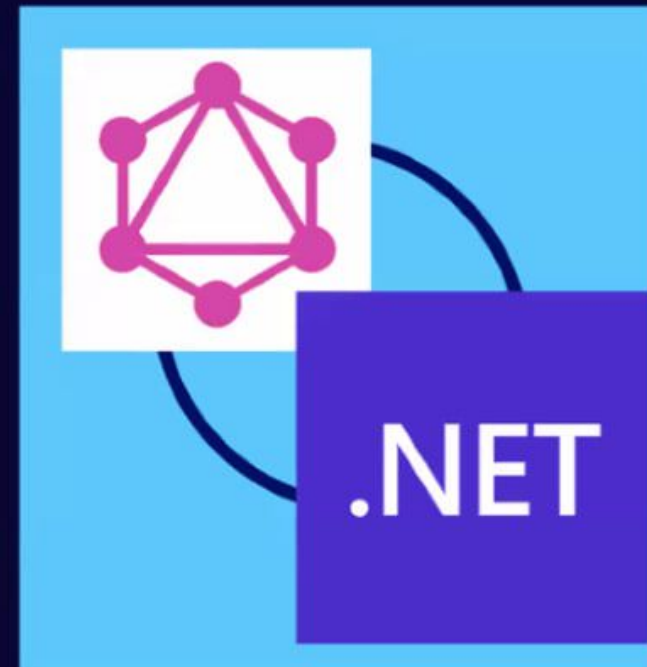
In Beta

The Graph .NET SDK is largely based on the popular Search & Navigation SDK used by many developers today. If you prefer ASP.NET templating over a JS stack we have a .NET SDK available in Q1.

- Leverage Graph for existing .NET templates
- Build headless frontends based on .NET Core
- Build UWP apps using Optimizely CMS content

Before you contribute to this project, ensure you have the following installed:

- .NET SDK 6+
- SQL Server 2016 Express LocalDB ([download here](#))



<https://github.com/episerver/graph-net-sdk>

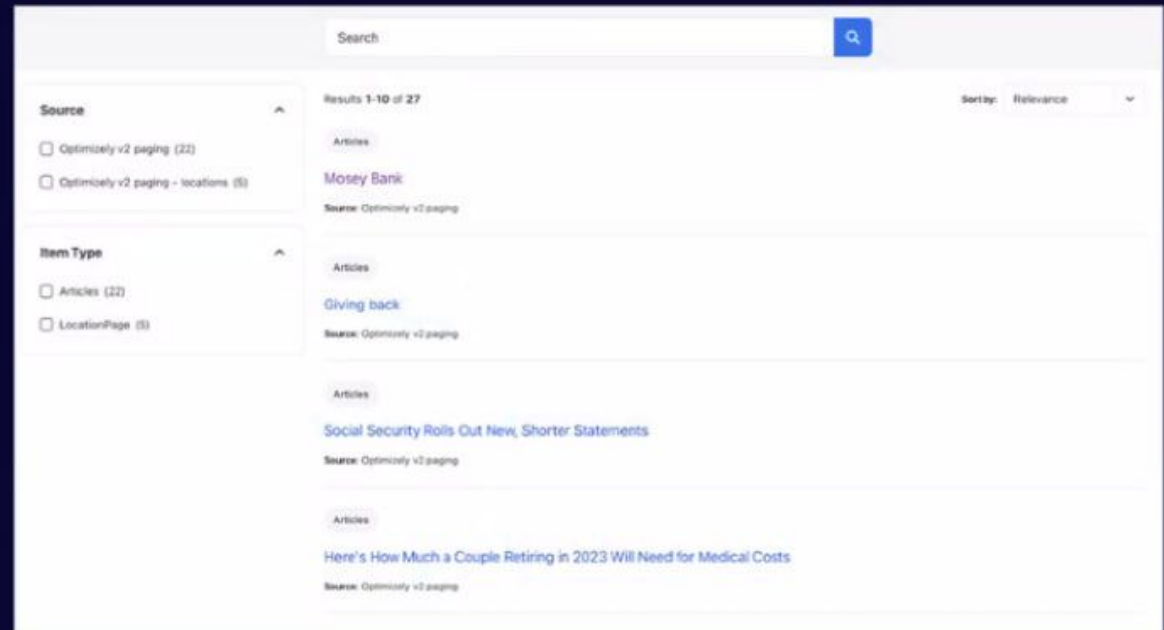
Integrating Graph with Coveo



In Beta

Coveo is an industry-leading AI-powered enterprise search engine which extends Optimizely Graph to fulfil a wide number of use cases.

- Personalized search
- Consolidate disparate sources into a single search source of truth
- Predictive search that automatically detects intent so that user queries yield relevant results



SaaS Core Updates



SaaS Core is a SaaS-based version of our market-leading CMS, currently in beta.

What's Next

Content Creator Experience

- Side-by-side preview
- Pre-built elements
- Preview/edit resize
- Mobile/Tablet/Desktop preview
- Outline built up of sections
- Section-level highlight and interaction in preview
- Drag-and-drop rearranging of sections and elements
- No-code save and reuse experience and section templates

- Template gallery dialogue for sections and experiences
- Publish from Visual Builder view
- Thumbnail generation for section and experience templates
- Custom elements
- Element-level interaction in preview
- Render settings on elements, sections, and experiences
- Merged layout and content tabs*

Developer Experience

- Headless site configuration
- Refined Optimizely Graph integration

Other Updates

- Production-grade Application Lifecycle Management (ALM) process
- Restriction management*
- Pre-built DAM integration

[Be part of our Early Adopter Program!](#)

*Prioritized based on customer feedback



OPTION 1

Pure headless

SaaS Core

**Optimizely
Graph**

Frontend

(flexibility to pick a front-
end of your choosing!)

OPTION 2

Decoupled

SaaS Core

**Optimizely
Graph**

Frontend
Website that
renders HTML

OPTION 3

Decoupled with presentation control

SaaS Core
Or
PaaS Core

**Visual
Builder**

**Optimizely
Graph**

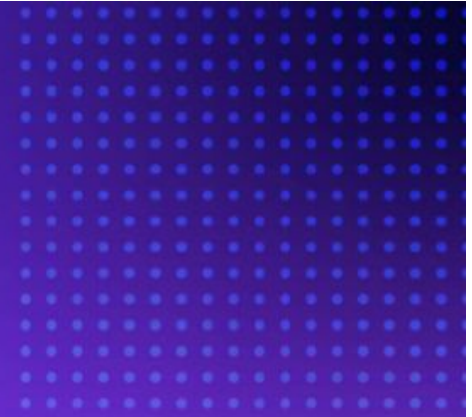
Frontend

OPTION 4

Coupled

PaaS Core

Front end



OPTION 5

Hybrid



OPTION 6

Aggregated

**SaaS Core
or
PaaS Core**

**External
Source(s)**

**Optimizely
Graph**

**Visual
Builder**

Front end

Optimizely Headless Forms



In Beta

Using headless Optimizely Forms with [Optimizely Graph](#) or [Content Delivery API](#) serves customers who already work with [Optimizely Forms](#) in a new way. Working with Forms can be delivered through API so customers can build a flexible and robust decoupled or headless system.

Install the headless Optimizely Forms package to leverage functionality and features.

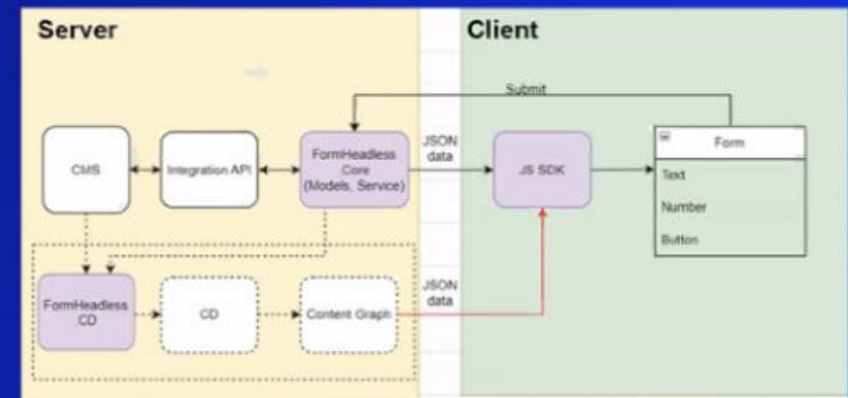
Headless Forms and CMS is one possible head a customer can use as a management site that serves and processes data.

Optimizely also provides the headless JavaScript SDK or React as a starter kit to build a front-end site to deliver data and UI that works in decoupled scenarios.

JS SDK has 2 main components:

- **Form SDK** – Definitions of models, form container, element block, and validator models.
- **Form React SDK** – React components are elements (TextblockElementBlock.tsx, TextareaElementBlock.tsx, and so on) and forms (Form.tsx, FormContainerBlock.tsx) that receive form metadata to render forms, input, select, textarea, and so on, as HTML

Optimizely Graph query and React component options work with returned Optimizely Graph data.



<https://docs.developers.optimizely.com/content-management-system/v1.2.0-forms/docs/get-started-with-headless-optimizely-forms>

Optimizely Headless Forms

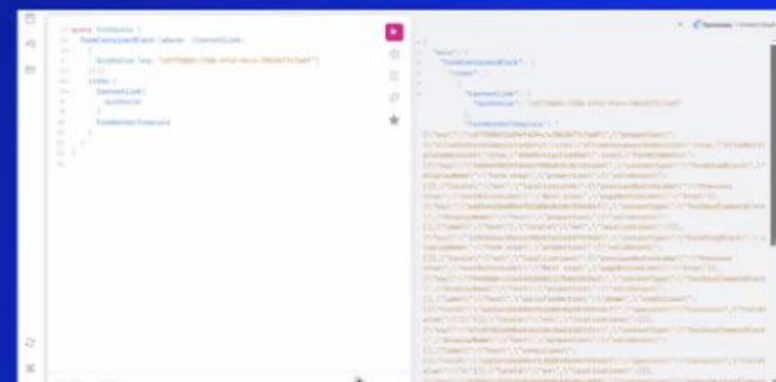


In Beta

Headless.Forms.Service provides a built-in set of APIs to fulfill a complete flow of forms, including Get Forms structure or data, submit, and get- or list-submitted data.

Headless.Forms.Contentgraph and Optimizely Graph let the search system index the form data into a specialized property in a schema.

Site customers have a single experience working with an Optimizely Graph query to get data, including Forms in the query result.



Getting Started With Headless Forms

Install Graph with Headless Forms

- Get Access to Graph
- Set up the Management Site (add in Graph settings etc.)
- Set up React App

Install Forms SDK with client app

- Create a React app template
- Install SDK
- Modify Source Code
- Run Client Site (**npm start**)
- Update ports

SDK for Headless Optimizely Forms

- Install **@epi-server/form-sdk**
- Install **@epi-server/form-react** (optional)
- Add the Form component imported from **@epi-server/form-react** along with the form key and base URL to render the form

Your (*the customer*) investment

Optimizely headless ecosystem

Front end-agnostic



Model content



Creator experience



Search



Experimentation + personalisation



Future capabilities



Support ecosystem



What it really takes to build a digital property

Front end-agnostic



Model content



Creator experience



Search



Experimentation + personalisation



Future capabilities



Support ecosystem



Most headless vendors

Front end-agnostic



Model content



Graph vs Search & Nav Trade-Offs:

Optimizely Graph has some limitations today:

1. **No Best Bets** (*Planning, In Development soon*) – Boosting Yes
2. **No Personalized Search** (*Planning, Later this year*)
3. **No Tracking** (*Should be done via ODP tracking*)
4. **No More Like Search Recommendations** (*In Progress*)
5. **No binary file support**

Semantic Search now supports 30+ languages

Content Sources

Live

Custom Data Sources

Optimizely Graph provides endpoints to let you manage custom data in your graph service. The custom data can be queried in the same way as content indexed automatically.

To index data, you must send data according to the NdJSON format. NdJSON is regular JSON separated by new lines. It is useful for streaming large amounts of data without having to parse everything as one large JSON string.

Each data item you want to index must be prepended with a separate action line:

```
{"index": {"_id": 1, "language_routing": "en"}}
```

The endpoint lets you send your content data to the Graph service: <https://cg.optimizely.com/api/content/v2/data?id=<content source id>>

The id query parameter is the ID of your content source. It should be the same as the one used when defining the content types for the

content source

```
{  
  "label": "Wordpress Posts",  
  "languages": ["en"],  
  "contentTypes": {  
    "WordpressPosts": {  
      "contentType": [],  
      "properties": {  
        "_ts": {  
          "type": "Date"  
        },  
        "source": {  
          "type": "String"  
        },  
        "id": {  
          "type": "String"  
        },  
        "Thumbnail": {  
          "type": "String"  
        },  
        "Title": {  
          "type": "String",  
          "searchable": true  
        },  
        "Content": {  
          "type": "String",  
          "searchable": true  
        }  
      }  
    }  
  }  
}
```

<https://docs.developers.optimizely.com/platform-optimizely/v1.4.0-optimizely-graph/docs/synchronize-content-data>

episerver/cms-saas-vercel-demo: Complete demo site for Optimizely ... github.com

Finish update

Memory usage: 205 MB

optly-cms-demo

- Repository
- Usage
- Domains
- Visit

Production Deployment

- Build Logs
- Runtime Logs
- Instant Rollback

The deployment that is available to your visitors.

Helping you save more for the things that

Deployment

optly-cms-demo-mavec4z8q-simon-chapmans-projects.vercel.app

Domains

optly-cms-demo-opal.vercel.app +2

Status Created

Ready 35d ago by schap-opti

Source

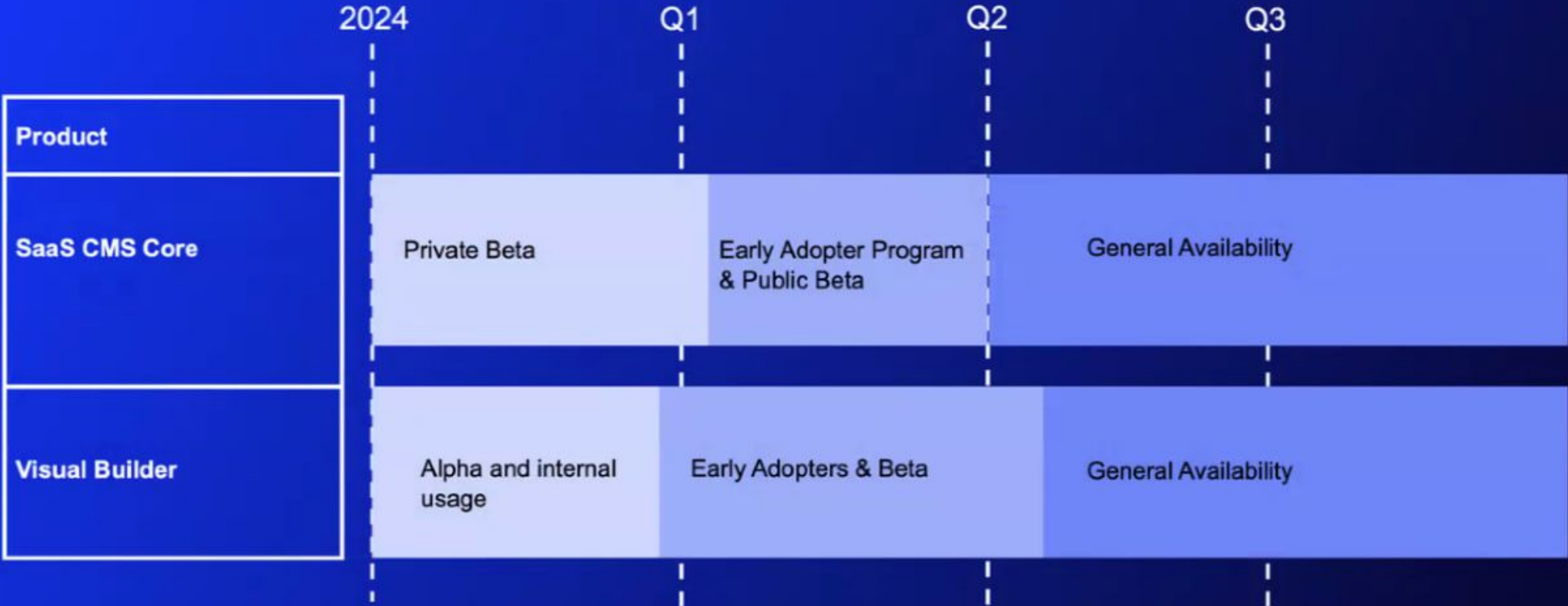
main

4f39a30 Initial commit

To update your Production Deployment, push to the "main" branch.

Learn More

SaaS CMS (Core and Visual Builder) Release timeline



Next steps

How we can win together

Join the SaaS Core Waitlist

Many partners and customers already signed up to the beta.

Join the waitlist here:

[Optimizely.com/beta](https://optimizely.com/beta)

Start your marketing now!

Get ahead of the pack with thought leadership pieces, technical guides and solution overviews.

Optimizely will amplify for broadest possible reach.

Nominate early adopter customers

We are now assessing customers suitable to join the SaaS Early Adopter Program (EAP).

Get in touch for consideration.

Technical Resources

Further reading:

<https://docs.developers.optimizely.com/digital-experience-platform/v1.4.0-content-graph/docs/project-graphql>



Sample Site:

<https://cg-blog.vercel.app/>

Github Repo:

<https://github.com/optimizely/vercel-examples>

<https://github.com/episerver/cms-saas-vercel-demo>

Content Sources:

<https://docs.developers.optimizely.com/platform-optimizely/v1.4.0-optimizely-graph/docs/synchronize-content-data>

.Net Graph SDK

<https://github.com/episerver/graph-net-sdk>

