




# Rendering strategies for the best site performance with Next.js

Including a new edge-first option for your Next.js toolkit.

TRUSTED BY COMPANIES INCLUDING

**VOXMEDIA**

 HACKERNOON

**DATAS**TAX



**stedi**

# //Contents

## 01 Personalization vs. speed

Every second matters

## 02 Next.js rendering strategies

Static-side generation

Server-side generation

Client-side rendering

## 03 Edge-first option

Edge Middleware with Vercel

## 04 Understanding the edge

The edge/edge computing

Serverless Functions

## 05 How does Middleware work?

Short-distance sprinter of the Edge Network

## 06 The world of the edge with Vercel

Edge Functions

Edge Runtime

Vercel Edge Network

Edge Middleware

## 07 Frontend flexibility

Contact Sales

# Personalization vs. speed: every second matters

As online businesses know, page load delays can equal thousands of dollars in lost revenue. However, personalization matters too—if not more. This is one of the key challenges facing developers today. What’s the best way to render a page, when you’re juggling the trade-off between speed and personalization?

Developers and your business shouldn’t have to choose. With frameworks like Next.js—which gives you the flexibility to implement different rendering methods on a per-page basis—and new solutions like Vercel Edge Middleware, you can reduce those trade-offs.

**100 milliseconds**

of latency can cost a

**1% decrease**

in conversions

In this brief, we’re going to examine different rendering strategies, including the newest option to your Next.js toolkit: Edge Middleware.

# Rendering strategies with Next.js

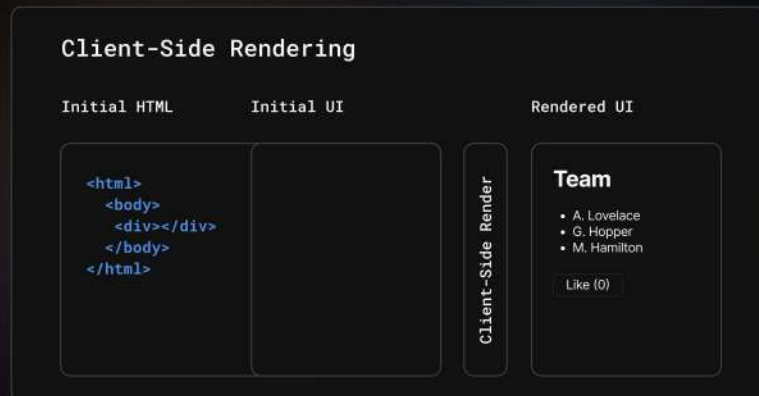
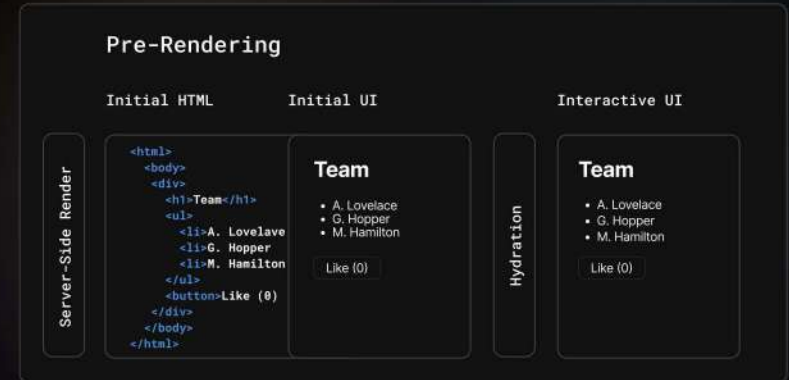
Teams approach solving this problem from different angles and frameworks. A key benefit of Next.js is the ability to choose different rendering strategies on different pages. Right now, Next.js provides three main strategies for page delivery:

## 1. Static-site generation

Serve the same page to everyone by pre-building a version and hosting at locations around the globe. This makes initial response times incredibly fast, because every visitor is shown the same prebuilt site. The drawback is a lack of personalization. It's like a magazine that has been placed in everyone's mailbox—a customer may quickly grab their magazine, but the content is the same for a student in Texas and a CEO in Japan.

## 2. Server-side rendering

Build the site at the moment the user requests it, so information specific to that user is derived. This allows for personalized content based on data like location, authentication state, language, and more. The downside? Up to 10X slower load times, especially if the server is physically distant from the user. Using the same analogy as above, the magazine has a personalized cover, but it takes longer to print and distribute the versions.



## 3. Client-side rendering

Defer the rendering work to users' devices and run JavaScript on the browser after the page loads. The browser does the work, with a lot of client-side code. This strategy is better suited for highly interactive pages but can sometimes result in users seeing a content flicker during the first page load.

# Introducing an edge-first option for your toolkit

It's the best of both worlds: Edge Middleware is code that helps developers escape lengthy configuration files to instead define routing rules like rewrites, redirects, and headers with code. When deployed to Vercel, Edge Middleware is automatically configured as a special set of Edge Functions.



## Meet Edge Middleware

With Edge Middleware, developers can ship multiple versions of the same pre-built static site and fetch the personalized version dynamically without impacting site performance.

Before diving in to ways you can implement this with Vercel, let's learn the basics of edge computing.

Want an Edge Middleware demo?

[Let's Talk](#)

# Understanding the edge

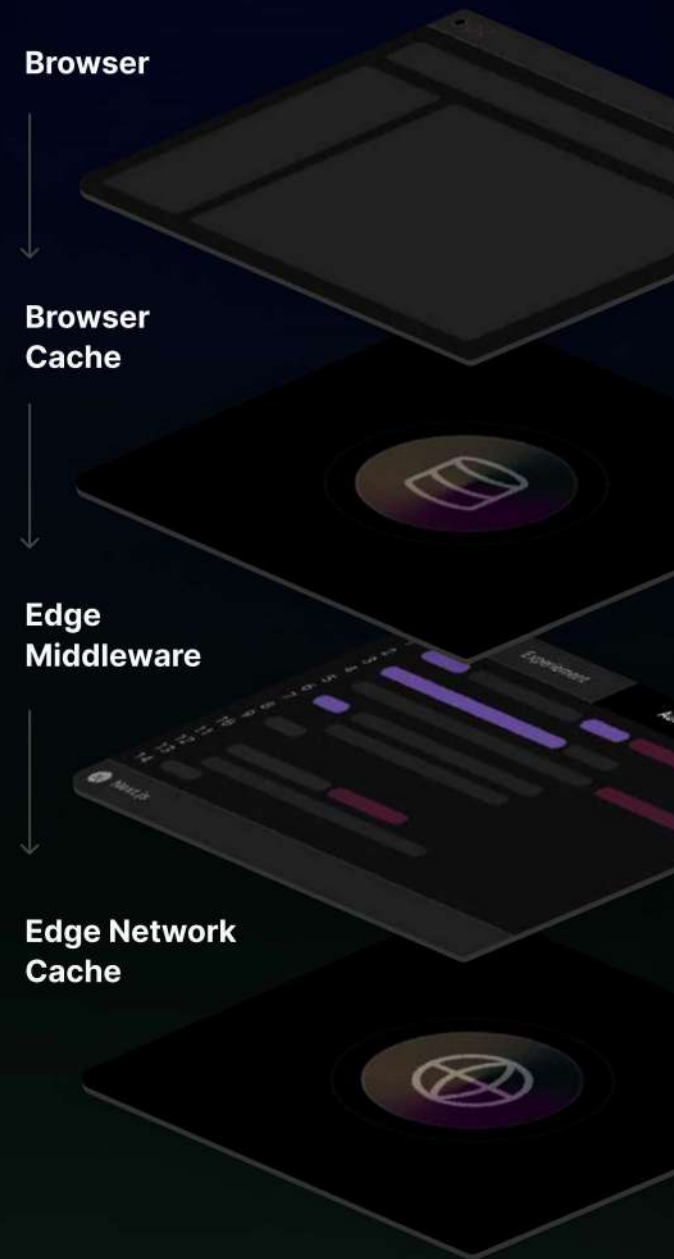
Let's define some basics.

## The edge/edge computing

The edge refers to the edge of the network, closest to the user. Content delivery networks (CDNs) are a part of the edge, managing the caching and distribution of assets around the world, and the edge also has the ability to run and execute code as close as possible to the user.

## Serverless Functions

Serverless Functions enable code to run on-demand without needing to manage your own infrastructure, provision servers, or upgrade hardware. They are co-located with your code and part of your Git workflow, allowing developers to use JavaScript and other languages to handle user authentication, form submissions, database queries, custom Slack commands, and more. As traffic increases, they automatically scale up and down to meet your needs.



# How does Middleware work?

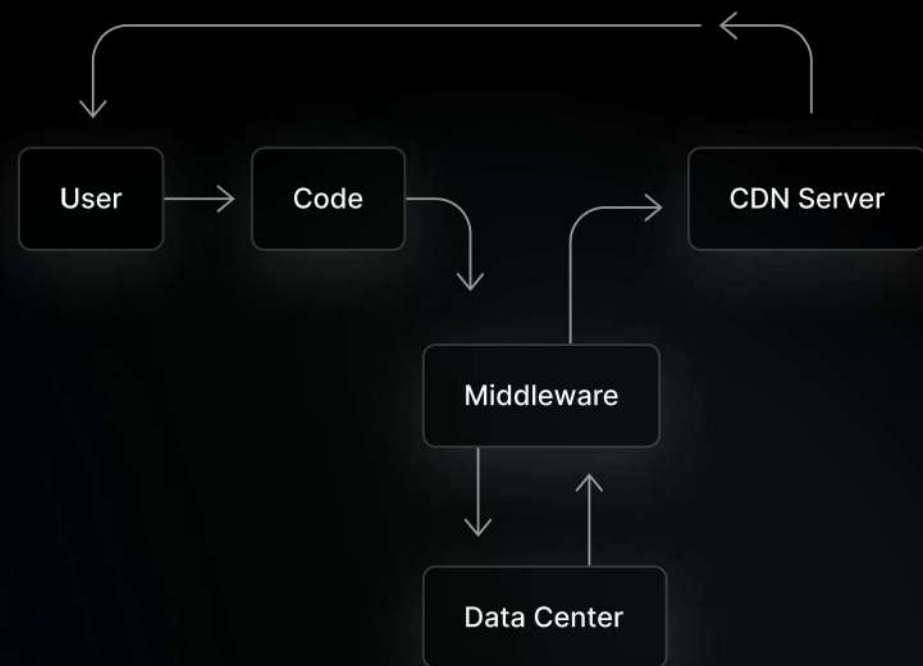
Middleware is another part of your application—it is code that executes before a request is processed on a site. That code will typically process or modify the response by rewriting and redirecting.

It triggers data to run before a page loads, and returns a specific route based on the answer.

As the request is made to the server, Middleware will intercept, and send back the correct page elements (for example: where someone is located or whether they're logged in).

With Middleware, decisions get made on the server side, not in the browser.

① Quickly running content between the user and their neighboring CDN server, Middleware is like the short-distance sprinter of our Edge Network.





In short, Middleware:

- Tells the server which page to return, based on the user requesting it
- Runs before the page is loaded, and returns page based on the answer
- Lives between the code and the data center

# The world of the edge with Vercel

Built by the creators of Next.js, Vercel is an infrastructure platform that handles the hard things: deploying instantly, scaling automatically, and serving personalized content around the globe while also configuring the backend code made available to you in Next.js. Get to know some edge-related terms specific to our products, and learn how they can give you the ultimate frontend flexibility.



## Edge Functions

Similar to their Serverless Function counterpart, Edge Functions on Vercel give developers the ability to deploy backend code inside their frontend application. These are deployed globally by default, and powered by the Edge Runtime. By taking advantage of this small runtime, Edge Functions typically have faster cold boots and higher scalability than Serverless Functions.



## Edge Runtime

Edge Runtime is the execution environment for Edge Middleware and Edge Functions. It runs on top of the JavaScript V8 engine that's used by the Chrome browser. It's leaner and more nimble than the Node.js runtime, allowing for faster cold boots and higher scalability. It's why Edge Middleware and Edge Functions are so fast.



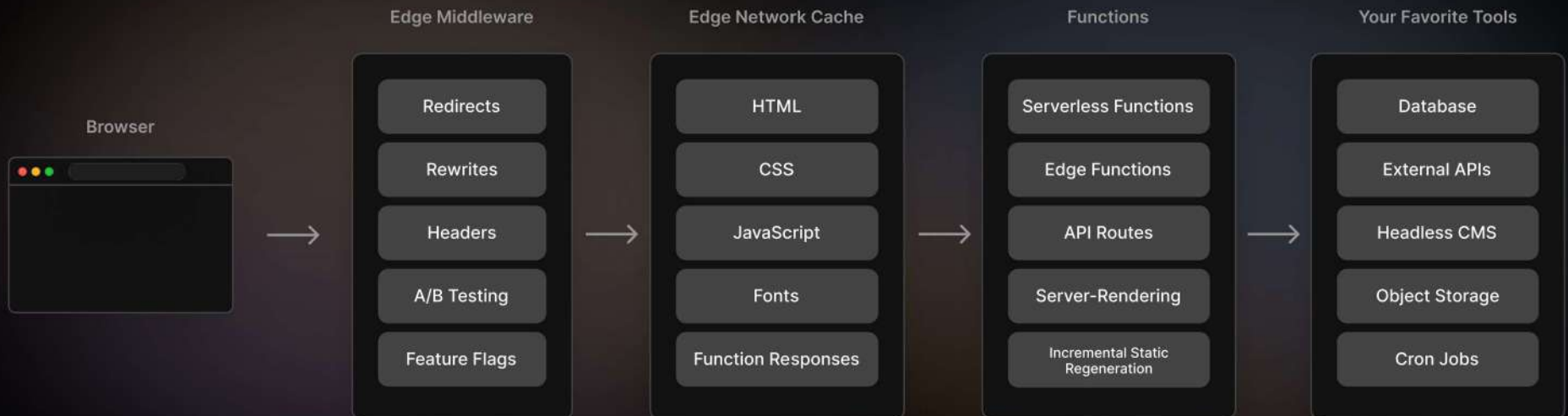
## Vercel Edge Network

The Vercel Edge Network is the globally distributed network that handles your site's requests. Each region contains a cache of your site's data, as well as a computing infrastructure to run Edge Middleware and Functions with the lowest possible latency.



## Edge Middleware

Edge Middleware is a piece of your edge computing architecture that allows your request to be routed dynamically. Written as part of your frontend code, Middleware gets executed before a request is processed on a site, giving it the ability to quickly route and rewrite requests to return pre-rendered pages.



# Frontend flexibility

Site speed or personalization? When using Edge Middleware and Next.js, you can have both. By matching the right rendering strategy to the right page, and incorporating Edge Middleware into your workflow, your site will be more efficient (less application bloat), faster (starts up and returns quickly), and performs better (more easily indexed for better SEO).

This empowers you to build sites that achieve better user and developer experiences, giving you the ultimate competitive advantage.

Developers should have the flexibility to build sites the way they want.

- ◆ Create personalized content at the speed of static sites.
- ◆ Self-serve advanced routing requirements.
- ◆ Couple dynamism and caching.
- ◆ Give your customers unmatched UX.
- ◆ All within your framework.



# Get started with Next.js and Edge Middleware

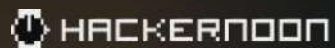
Learn more about rendering strategies with Next.js and how your organization can deliver faster and more personalized content.

Let's Talk

Get Started

TRUSTED BY COMPANIES INCLUDING

VOXMEDIA



DATASTAX



stedi