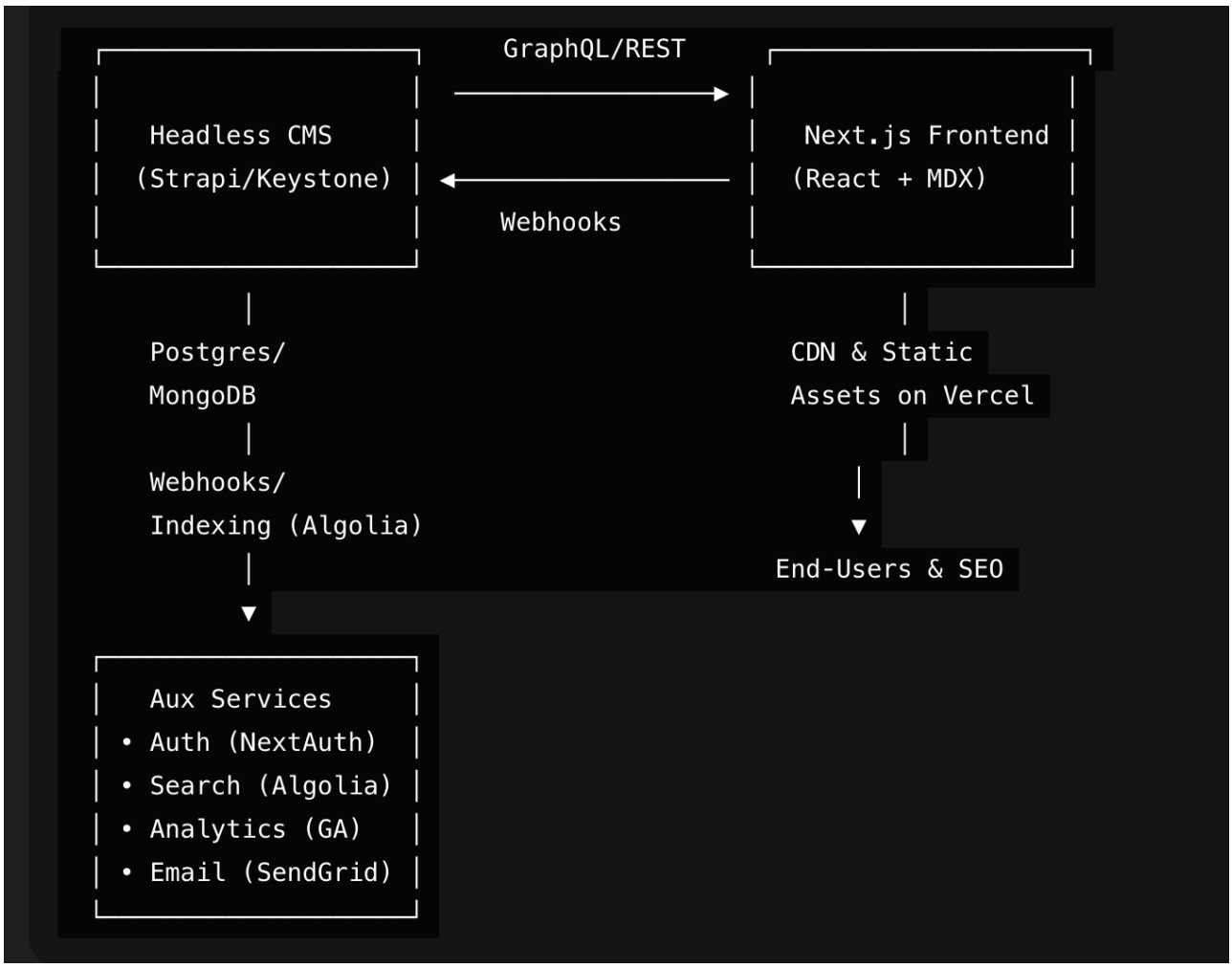
Technical Design for Blog

Monday, July 21, 2025 8:56 PM

High-Level Overview
You'll build a **JAMstack**-style CMS/blog using modern JavaScript frameworks: a **headless CM** friendly rendering and rich interactivity. The system will let you author and organize blog po (About, Services, Contact), capture inbound leads (contact form, newsletter), integrate socia showcase portfolio piece.

1. Architecture Diagram (Logical Components)



2. Technology Stack

Layer	Choice

S backend to manage content, and a **Next.js** frontend for fast, SEO-
sts (with code snippets, images, tags, categories), manage pages
l auth so employers/businesses can connect, and serve as a

Questions:

What is MDX?

MDX is a flavor of
Markdown files. Th
import and use int
your content

Why do we need a REST

While GraphQL oft
simpler for many u

- **Support too**
- **Leverage sta**
- **Maintain co**

For example, Strap
integration

Is NextAuth.js open sou

Yes. NextAuth.js (r
It's maintained by
and supports OAut

What is Algolia?

Algolia is a "search
delivers instant, fu
build or maintain y
Yes — Algolia offe
developing and tes

- **1 000 000 re**
- **10 000 searc**
- Core feature
- No credit ca

Markdown that lets you embed JSX (i.e. React components) directly within your
his means you can write standard Markdown for prose and headings, then
teractive components—like charts, call-outs, or code sandboxes—inline with

T fallback?
fers powerful, client-driven querying, REST remains universally supported and
use cases. A REST fallback lets you:
ols or teams that aren't set up for GraphQL
andard HTTP caching and tooling
mpatibility during GraphQL migrations or plugin gaps
pi v5 exposes both GraphQL and REST APIs so you can pick the best fit for each

rce and free to use?
now also branded as Auth.js) is fully open source (MIT-licensed) and free to use.
the community, designed specifically for Next.js and serverless environments,
th, email/passwordless flows, and more out of the box

n-as-a-service" platform providing a hosted, high-performance search API. It
ull-text and faceted search results—often in under 100 ms—so you don't have to
your own search infrastructure
rs a **free tier** (called the **Build** plan) that you can use indefinitely while you're
sting. Under the Build plan you get:
ecords of searchable data
ch requests per month (applies to both Search and Recommend)
es like analytics, A/B testing, AI synonyms, dynamic re-ranking, and NLP
rd required to sign up

Layer	Choice
Frontend	Next.js (React + TypeScript + MDX)
Styling	Tailwind CSS
CMS Backend	Strapi (Node.js/TypeScript)
Database	PostgreSQL (or MongoDB)
API Protocol	GraphQL (preferred) + REST fallback
Auth	NextAuth.js (GitHub, LinkedIn OAuth)
Search	Algolia (via Strapi webhook)
Comments	Giscus (GitHub Discussions) or custom
Email/Forms	SendGrid (SMTP/API)
Analytics	Google Analytics / Plausible
Deployment	Vercel (frontend) + Heroku/DigitalOcean
CI/CD	GitHub Actions (lint, test, deploy)

3. Detailed Component Design

3.1 Headless CMS (Strapi)

- **Content Types**
  - **Post**: title, slug, date, excerpt, content (MDX), featured image, tags, categories, a
  - **Page**: title, slug, content (MDX)
  - **Author**: name, bio, avatar, social links
  - **Tag/Category**: name, slug
  - **ContactSubmission**: name, email, message, timestamp
- **Roles & Permissions**
  - Public: read posts/pages
  - Admin: create/update/delete posts & pages
- **Webhooks**
  - On publish/update post → trigger Algolia indexing & trigger redeploy of fronten
  - On contact form submission → send email via SendGrid

3.2 Frontend (Next.js)

- **Rendering Strategy**
  - **Static Generation (SSG)** for blog posts & pages at build time

- Limits: 1 app
  3 queries/se

Once you exceed t
then billed **pay-as**
additional 1 000 re

What is Giscus?

Giscus is an open-s
maintaining your c
thread on GitHub,

author

d (if using ISR)

plication, up to 10 indices, 1 GB total index size, 10 KB max record size,
econd, 30-day inactivity rule Algolia
those quotas, the free plan will suspend until you upgrade. Any excess usage is
-you-go (e.g. ~$0.50 per additional 1 000 search requests and ~$0.40 per
ecords)

source commenting system powered by GitHub Discussions. Instead of
own comment database, it maps each page (by URL or title) to a Discussion
letting visitors post comments and reactions via their GitHub accounts

    - **Incremental Static Regeneration (ISR)** to pick up new posts within minutes
- **Routing**
    - /posts/[slug] → dynamic MDX page
    - /[page] → site pages (About, Services)
- **MDX Support**
    - Embed React components (e.g. interactive code sandboxes, charts)
- **Components**
    - <PostList>, <PostCard>, <TableOfContents>, <CodeBlock> (with syntax highlighti
- **Styling**
    - Tailwind utility classes, custom theme for blog
- **Search UI**
    - Algolia InstantSearch component on /search
- **Comments**
    - Integrate Giscus (lightweight, no backend work) or build a small comments API i

3.3 Authentication & Networking

- **NextAuth.js**
    - Providers: GitHub & LinkedIn (so potential employers can "Sign in with…" and yo
    - Stored sessions in secure cookies
- **User Dashboard** (optional)
    - List posts they've liked/commented on, messages, network connections

3.4 Contact & Lead Capture

- **Contact Form**
    - Frontend form → Next.js API route → validate → save to Strapi ContactSubmissi
- **Newsletter Signup**
    - Integrate with Mailchimp/ConvertKit API

4. Data Model (ER Diagram Snippet)

Author—<Post>—<Post_Tag>—Tag
Page
ContactSubmission

- **Post** *m:1* → **Author**
- **Post** *m:n* → **Tag** (through join table)
- **Page** standalone

5. CI/CD & Deployment

ng)

n Strapi

ou can capture who’s engaging)

on + email via SendGrid

5. CI/CD & Deployment

1. **GitHub Actions**
   - On PR: lint (ESLint, Prettier), type-check, unit tests
   - On merge to main: build Strapi Docker image, deploy to Heroku; build Next.js, d
2. **Environment Variables**
   - CMS_URL, DATABASE_URL, SENDGRID_API_KEY, ALGOLIA_KEYS, NEXTAUTH_SE

6. Non-Functional Considerations

Requirement	Solution
Performance	CDN edge caching (Vercel), ISR, image CDN
SEO	Next.js Head meta tags, sitemap.xml, robots.txt
Scalability	Serverless functions + managed DB
Security	HTTPS everywhere, CSP headers, rate-limit forms
Maintainability	TypeScript everywhere, modular components
Analytics	Page-view/event tracking with GA + custom events (newsletter, contact)

7. Future Extensions

- **Portfolio Section**: showcase projects with live demos
- **Client Testimonials**: a Testimonial content type in Strapi
- **Job Board**: integrate remote job listings via API
- **Multi-author blog**: invite guest authors with limited CMS roles

Next Steps

1. Scaffold Strapi project (npx create-strapi-app).
2. Define content types & set up PostgreSQL.
3. Scaffold Next.js site (create-next-app --typescript).
4. Wire up GraphQL client & MDX parsing.
5. Implement core pages (home, posts listing, post page).
6. Add auth, forms, search, deploy to Vercel/Heroku.

eploy to Vercel

CRET, OAuth keys