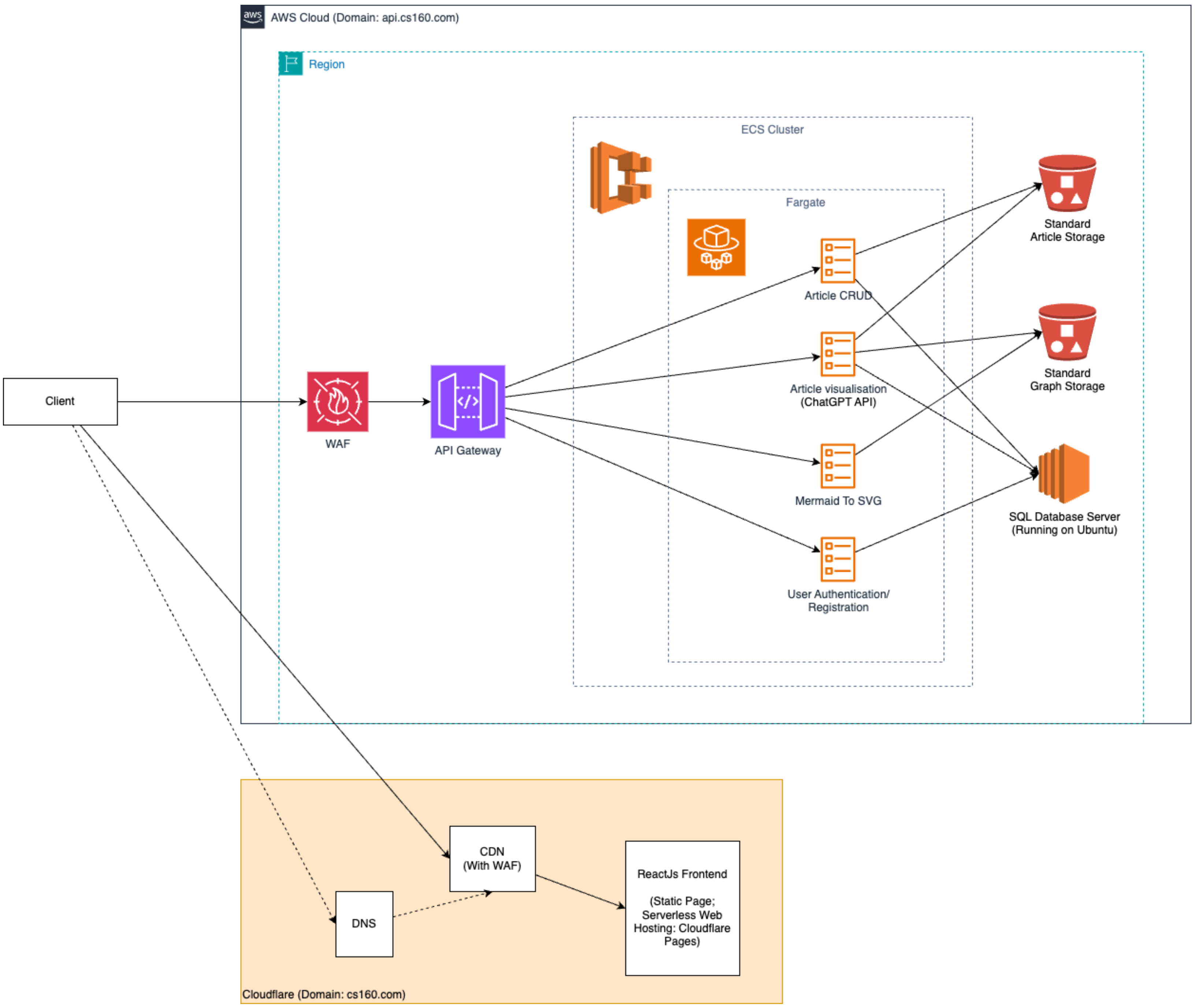
# System Design Specification


AWS Cloud (Domain: api.cs160.com)
Region
ECS Cluster
Fargate
Client
WAF
API Gateway
Article CRUD
Article visualisation (ChatGPT API)
Mermaid To SVG
User Authentication/ Registration
Standard Article Storage
Standard Graph Storage
SQL Database Server (Running on Ubuntu)
CDN (With WAF)
DNS
ReactJs Frontend
(Static Page; Serverless Web Hosting: Cloudflare Pages)
Cloudflare (Domain: cs160.com)

We have two separate network environments, which are publicly accessible to our users.

# Cloudflare Environment

## Reasons

We use this environment to serve our front-end web-based application.
Since we are probably not going to do server-side rendering (SSR), we don't need the website server to get any dynamic data (e.g. user data, diagrams, articles).

The serverless web hosting service will just serve a static application with no dynamic data contained. We do rendering (combining the dynamic data with the static web app) on the client side, i.e. Client Side Rendering (CSR).

## Services

### CDN (Content Delivery Network)

We have the frontend app deployed all around the world so that users in every country can get the app within probably a second.

### WAF (Web Application Firewall)

It's a built-in feature for preventing attacks.

### Cloudflare Pages

*Temporary associated domain: cs160.com*

Serverless website hosting service for hosting our front-end application.

Reason for using this:
We don’t need to worry about managing a web server for serving website fetching requests.

### DNS

It's a built-in feature for storing DNS records for our domain.

# AWS (Amazon Web Service) Environment

We use this environment to host our backend API services, file storage, and database.

## AWS Services

### API Gateway

*Temporary associated domain: api.cs160.com*

An AWS-managed service for

- Managing API endpoints
  - OpenAPI Spec is needed
- Forwarding clients' requests to different API endpoints (different dockerised containers)
- Authorization
  - Verify the JWT that a user has

### WAF (Web Application Firewall)

An AWS-managed service for

- Filtering out malicious requests
- Filtering out requests that are not acceptable in our services
  - OpenAPI Spec includes what requests are acceptable

### ECS Cluster

We use ECS Cluster so that each microservice can scale up and down, meaning adding compute power and removing compute power, based on real-time traffic.

### Fargate

Fargate is for deploying containerized (Dockerised) applications and having them running in the "serverless" way.

We use Docker to containerize our applications so they can be deployed on any machine that has Docker installed.

## Containerized Microservices

1. Article CRUD (Create, Read, Update, Delete)
   *Temporary path: api.cs160.com/article*
   *Allow clients to upload, read, update, and delete articles.*

   Connect to a file storage (bucket), which is used for storing articles

2. Article Visualisation
   *Temporary path: api.cs160.com/article-visualise*
   *Allow clients to get graphs generated using ChatGPT API.*

   It connects to two file storage locations: "Article Storage" and "Graph Storage"
   It fetches an article from Article Storage and sends it to ChatGPT. It then saves it to Graph Storage

3. Mermaid To SVG
   *Temporary path: api.cs160.com/mermaid-svg*
   *Convert mermaid graphs in mermaid-format text from Graph Storage into SVGs.*

4. User Authentication/ Registration
   *Temporary path: api.cs160.com/login*
   It verifies user login credentials (username and password) and gives out JWT once authenticated.

   *Temporary path: api.cs160.com/register*
   It does user registration and creates user accounts.

## Bucket

An AWS-managed SaaS file storage service.

1. Article Bucket
   For storing Articles that are in PDF format
2. Graph Bucket
   For storing Diagrams that are in Mermaid text format, and markdown file format.

## EC2 (Elastic Compute Cloud)

An IaaS (Infrastructure as a Service) service for hosting virtual machines.

1. Ubuntu Server
   Running a MySQL database server.

   Reason for using EC2 over Amazon Aurora:
   Aurora is much more expensive because it’s an Amazon-managed database.