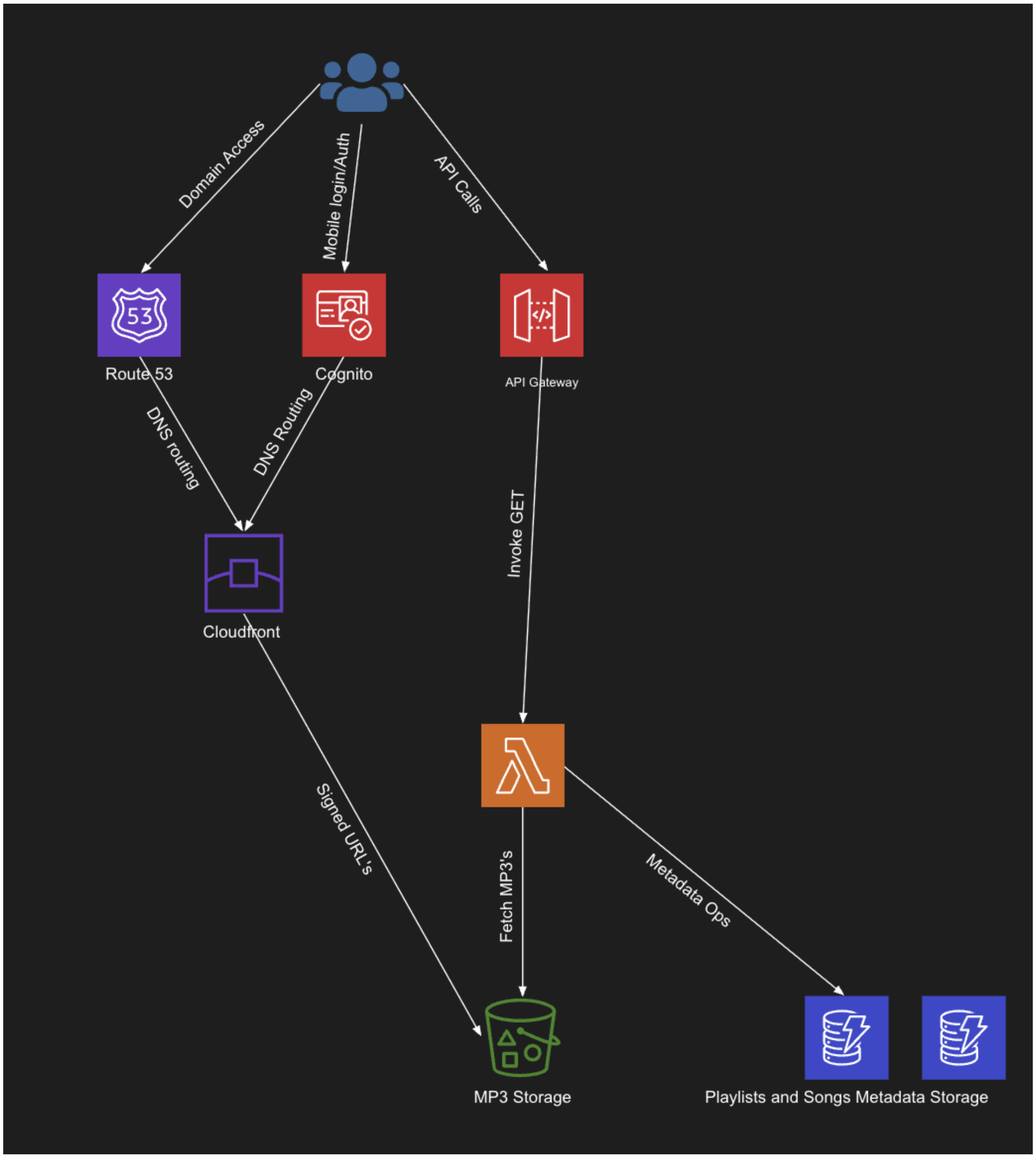
Faithful streams MP3 streaming site

Below I map out how the pieces connect for the Faithful Streams **MP3 streaming site on AWS**.



Storage & Streaming

- **Amazon S3** is used to stores theMP3 files.
- **Amazon CloudFront** distributes them globally for fast streaming.
  - CloudFront has an origin set to the S3 bucket that stores the music.
  - Users will stream MP3s from CloudFront through signed URLS not directly from S3 . This ensures better speed & security.

User Access

- **Amazon Cognito will** handle user sign-up, sign-in, and authentication via mobile and Web based applications
  - Integrates with your front-end app (web/mobile).
  - After login, Cognito issues tokens (JWT) that control access.

Backend Logic

- **API Gateway** provides endpoints access functionality through calls like:
  - GET /songs → list songs from DynamoDB
  - GET /stream/{songID} → request a streaming link
  - POST /playlist → create or update playlists
- **AWS Lambda** will execute backend logic for these endpoints.

Database

- **Amazon DynamoDB** stores song info, user playlists, likes, etc.
  - Example Item:

```
{
 "songID": "123",
 "title": "AWS Pro",
 "artist": "Faith Dakwa",
 "fileKey": "music/AWS Pro.mp3"
}
```

Domain & Access

- **Amazon Route 53** will point to the www.faithfulstreams.com domain to CloudFront for the front-end.
- **Front-end App** (HTML, Javascript, etc.) Fetches API data, requests MP3 files from CloudFront, and streams in the browser or app.

Connectivity Flow (simplified)

1. **User** opens faithfulstreams.com (via Route 53 + CloudFront)
    a. Logs in Social Networks (via Cognito + Cloudfront)
    b. Calls API (API Gateway → Lambda → DynamoDB/S3)
2. Gets back song list + signed CloudFront/S3 URLs
3. Streams MP3 via CloudFront