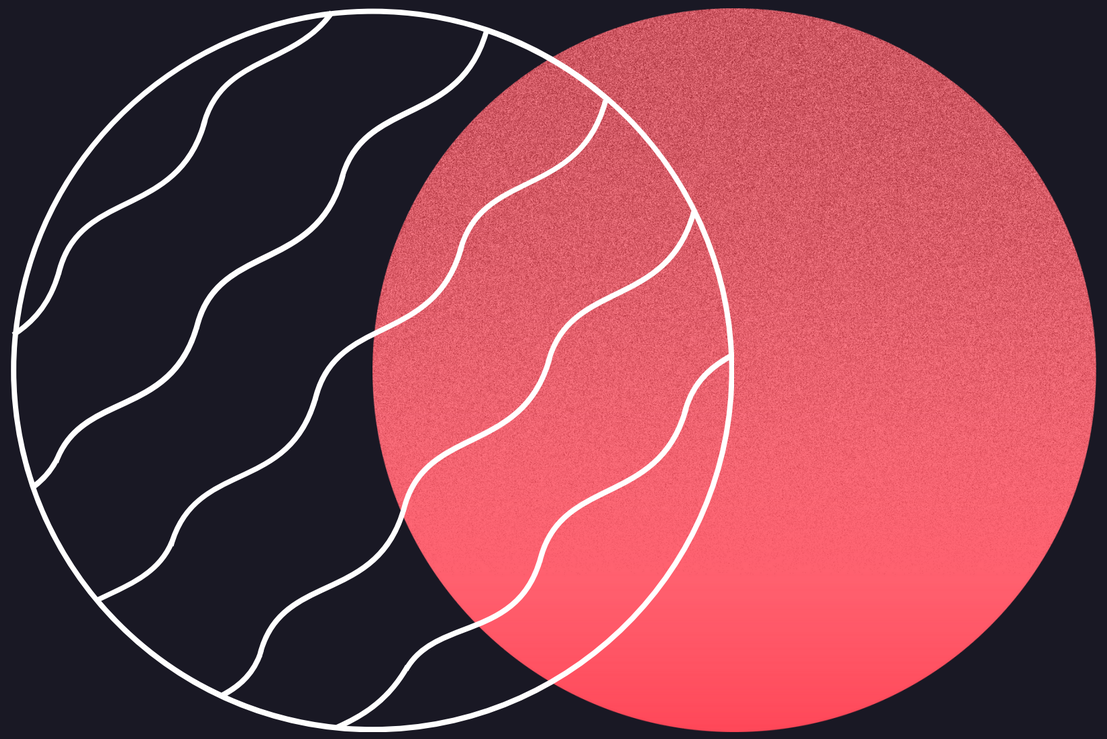
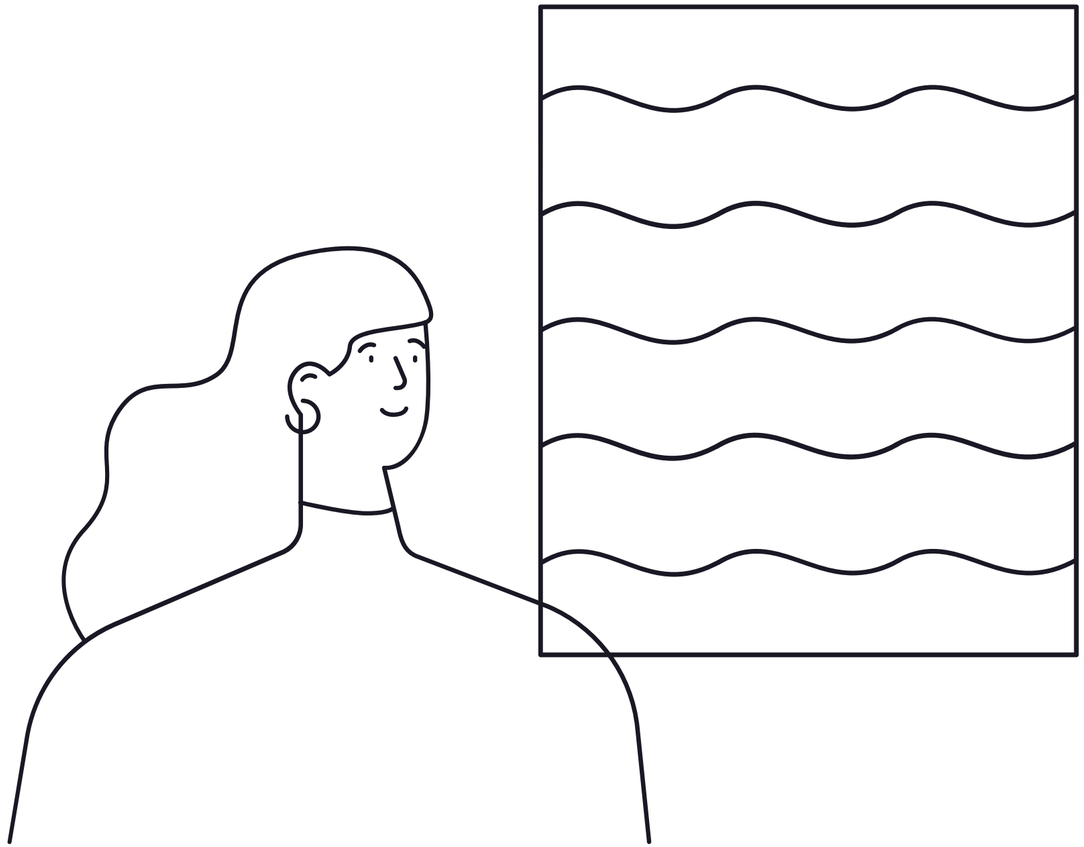


Data Drive

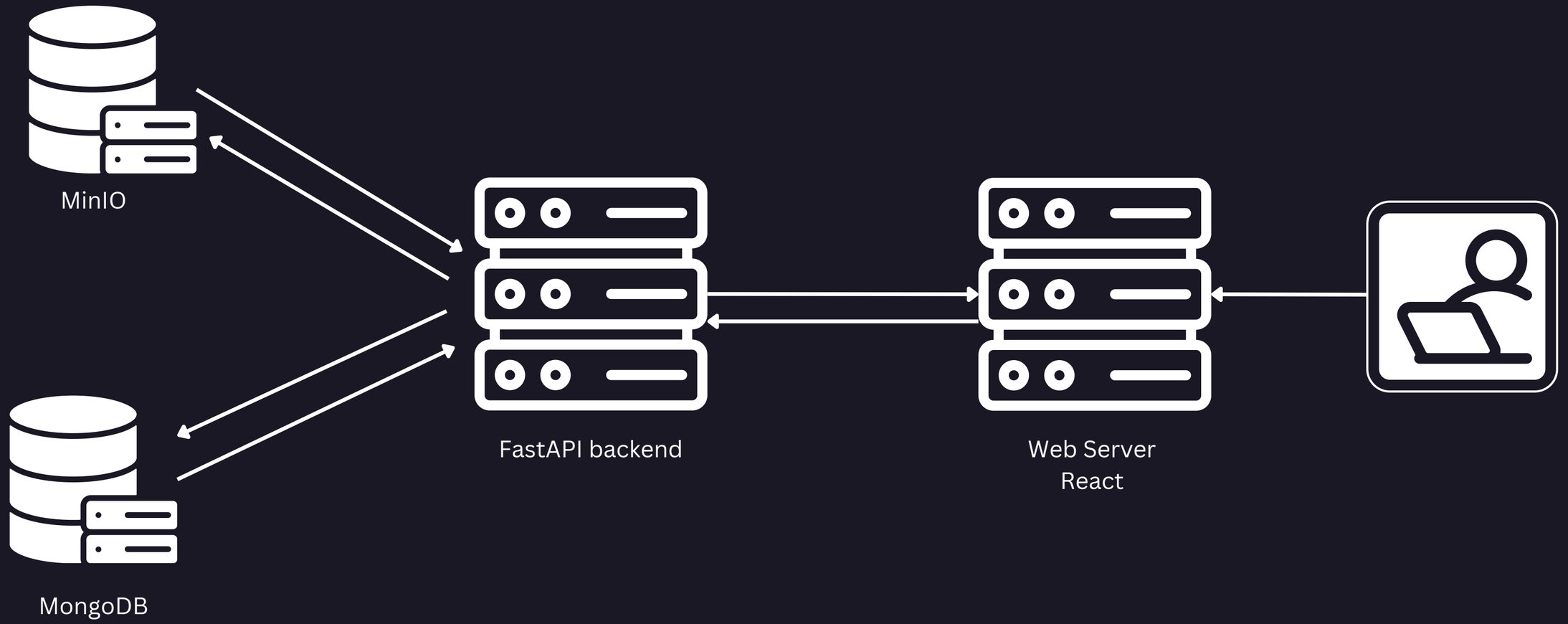
Rudransh Pratap Singh - 2020111007
Abhijnan Vegi - 2020101004
Shavak Kansal - 2020101023

What is Data Drive ?

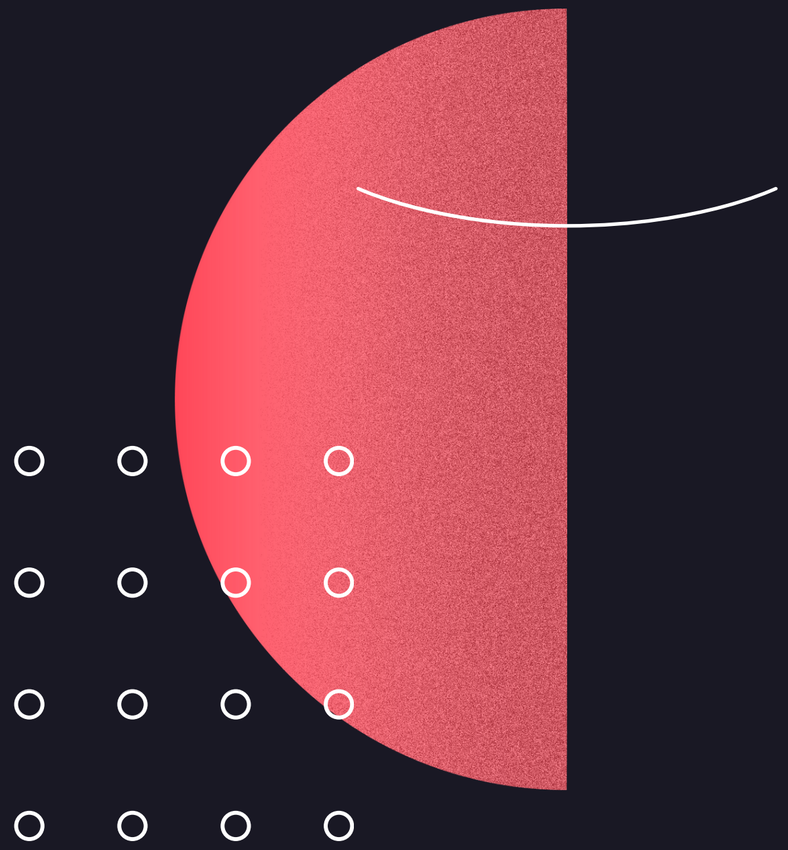
Data Drive is a versatile data management platform that seamlessly integrates with MinIO. It offers an abstraction layer on top of MinIO, enhancing it with additional features while providing an intuitive and user-friendly interface. This combination empowers us to efficiently manage, access, and collaborate on data.



Architecture Diagram



Expected Tasks



FRONTEND

User

- Login
- Home View
  - Folder view
  - List View
  - Sort
  - Properties
  - Move/Rename
  - Copy
  - Share
  - Upload/Download
- Shared Folders
- Disk Quota Remaining

Admin

- Dashboard - Usage Statistics
- Global Configuration
- User quota and access

BACKEND

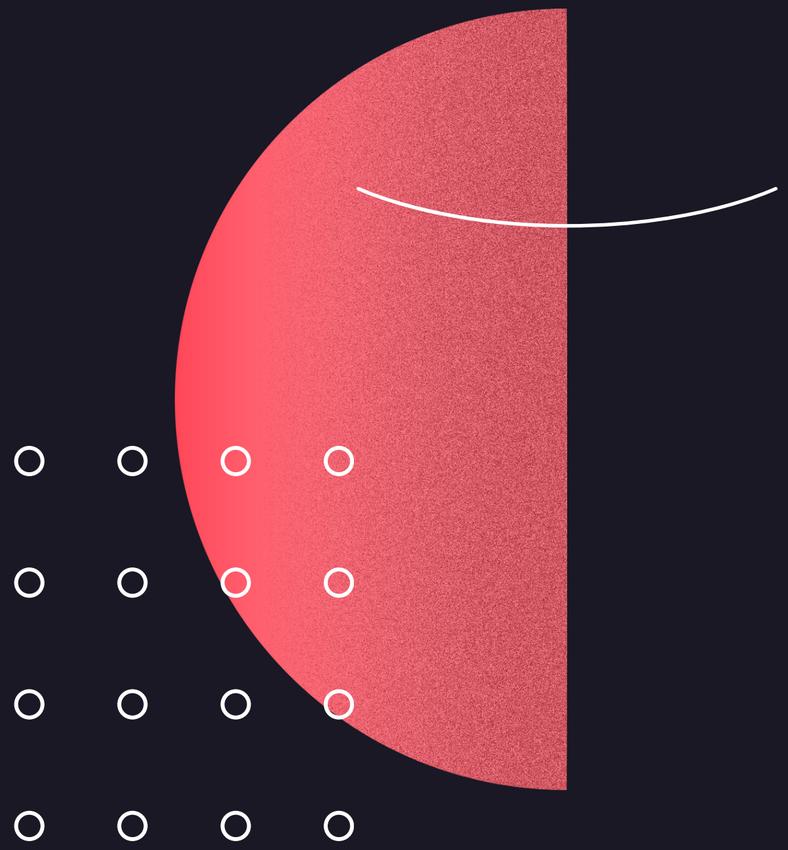
User

- Login
- List Directories
- Create files
- Download Files
- Share Folders
- Move Folders
- Copy Folders
- File Properties

Admin

- Get usage Statistics
- User Wise config changes

What we have done



FRONTEND

User

- Login
- Home View
  - Folder view
  - List View
  - Sort
  - Properties
  - Move/Rename
  - Copy
  - Share
  - Upload/Download
- Shared Folders
- Disk Quota Remaining

Admin

- Dashboard - Usage Statistics
- Global Configuration
- User quota and access

BACKEND

User

- Login
- List Directories
- Create files
- Download Files
- Share Folders
- Move Folders
- Copy Folders
- File Properties

Admin

- Get usage Statistics
- User Wise config changes

Challenges

1. Conserving server space
2. Handling shares

Conserving server space

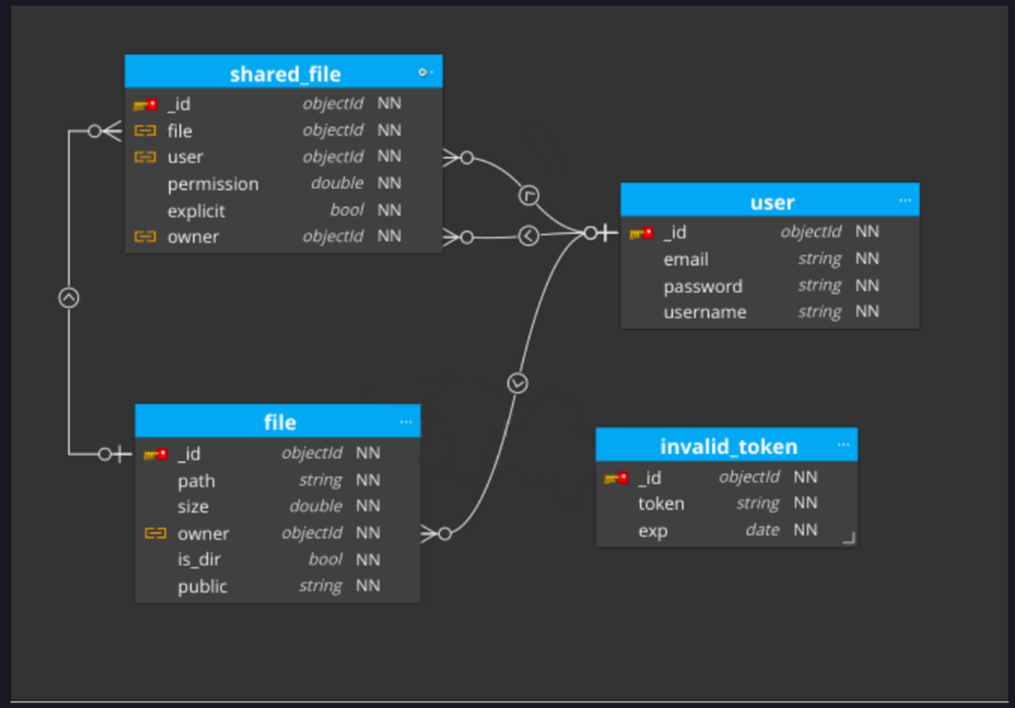
- To serve data to the user, the data must first be downloaded from MinIO to the server. Thus not cleaning this space up regularly can lead to server failures due to insufficient storage.
- To remedy this, we firstly create background download jobs for each download task.
- We then estimate the amount of the time the file will be required on the server heuristically and clean up the file once that amount of time has passed.

Handling shares

- Implementing sharing involves a lot of conflicting edge cases and even more when you add copy and move transactions.
- While performing move and copy operations we have to make sure appropriate existing shares are invalidated and new appropriate shares are added properly.
- Handling nested shares was also a challenging problem.

Database Design

File **Shared File** **Invalid Token** **User**



File

Database model for a file object, here file can represent either a file or a folder.

- Path - Path of the file.
- Size - Size of the file.
- Owner - Owner/Creator of the file.
- is_dir - Weather file object represents a folder or file.
- Public - Public permission for the file/folder (read, write, none), Public permission is used when a file is accessed by a user either when the user is not the owner of the file or the file is not explicitly shared with the user.

Shared File

Database model for a shared file object, here file is a generic term for both file and folder.

- File - Reference to the file object that is shared.
- User - Reference to the user object with whom the file is shared.
- Permission - Permission granted to the user for the file.
- Owner - Owner of the file.
- Explicit - Weather the file is explicitly shared with the user or the user has access to the file because of a parent folder that is explicitly shared with the user.

Invalid Token

Database model for a invalid token object.
An invalid token object represents an access token which is invalid.

- token : The invalidated access token.
- exp : The expiry time of the access token.

User

Database model for a user object.

- email - Email of the user.
- password - Password of the user.
- username - Username of the user.