

Functional and Non-Functional Requirements

Functional Requirements:

1. **File Upload:** Allow other microservices to upload files of any extension.
2. **File Download:** Provide a mechanism for retrieving stored files.
3. **File Deletion:** Allow deletion of files.
4. **Metadata Management:** Store and manage metadata for each file.
5. **Security:** Ensure only authorized services can interact with the storage service.
6. **Search Capability:** Ability to search files based on metadata.

Non-Functional Requirements:

1. **Scalability:** Handle large volumes of files and high request rates.
2. **Reliability:** Ensure high availability and durability of stored files.
3. **Performance:** Optimize for fast file uploads and downloads.
4. **Maintainability:** Easy to update and maintain.
5. **Monitoring:** Log all actions and provide metrics.

High-Level Design

1. **API Gateway:** Routes requests to the storage microservice.
2. **Storage Microservice:** Handles file operations (upload, download, delete, search).
3. **Authentication and Authorization Service:** Ensures secure access.
4. **Database:** Stores metadata about the files.
5. **Object Storage Service (e.g., AWS S3, Azure Blob Storage):** Stores the actual files.

Low-Level Design

Components:

1. **Controllers:** Handles HTTP requests and responses.
2. **Services:** Contains the business logic for file operations.
3. **Repositories:** Interacts with the database for metadata operations.
4. **Storage Clients:** Interacts with the object storage for file operations.
5. **DTOs:** Data Transfer Objects for API requests and responses.
6. **Middleware:** Handles authentication, logging, and error handling.

File Storage:

Type of Storage:

- **Object Storage (e.g., AWS S3, Azure Blob Storage):** Suitable for storing large volumes of unstructured data with high availability and durability.

Saving Files:

1. Receive file upload request.
2. Validate and process the request.
3. Store file in object storage.
4. Save metadata in the database.

Retrieving Files:

1. Receive file download request.
2. Validate and process the request.
3. Retrieve metadata from the database.
4. Fetch the file from object storage.
5. Return the file to the requester.

Deleting Files:

1. Receive file delete request.
2. Validate and process the request.
3. Delete metadata from the database.
4. Remove the file from object storage.

Communication with Other Microservices

Other microservices will communicate with the storage microservice via HTTP RESTful API calls. They will send HTTP requests (GET, POST, DELETE) to the storage microservice's endpoints.