Institute of Information Technology
University of Dhaka

Documentation of System Architecture For Google Drive

Submitted To
Mridha Md. Nafis Fuad
Lecturer,
Institute of Information Technology
University of Dhaka

Submitted By
Nusrat Jahan Lia (BSSE-1306)
Arnab Das Joy (BSSE-1308)
Tanjuma Tabassum Jerin (BSSE-1312)
Mahir Faysal (BSSE-1316)
Reza Abdullah (BSSE-1335)

Table Of Contents

User Story

User Management Module

1. As a user, I want to register for an account so that I can securely log in and use the system.
    - *Handled by: User Registration and Profile Service*
2. As a user, I want to log in using my credentials so that I can access my files and data securely.
    - *Handled by: Authentication and Authorization Service*
3. As a user, I want to update my profile information (e.g., name, profile picture, email) so that my account reflects accurate details.
    - *Handled by: User Registration and Profile Service*
4. As a user, I want to reset my password if I forget it so that I can regain access to my account.
    - *Handled by: Account Recovery Service*
5. As a user, I want to manage my subscription plan and payment details so that I can upgrade my storage or renew my plan.
    - *Handled by: Billing and Payment Service*

File Storage Management Module

1. As a user, I want to upload files to my drive so that I can store them securely in the cloud.
    - *Handled by: File Upload Service*
2. As a user, I want to download my files anytime so that I can access them offline.
    - *Handled by: File Retrieval and Download Service*
3. As a user, I want my files to be stored securely and backed up so that I never lose my important data.
    - *Handled by: File Storage Service*

Search and Filter Management Module

1. As a user, I want to search for files and folders by their names or content so that I can quickly find what I need.
    - *Handled by: Search Query Processing Service*
2. As a user, I want to filter files based on metadata (e.g., type, owner, size, date) so that I can easily organize and locate them.
    - *Handled by: Indexing Service*

3. As a user, I want to view a list of recent files and favorite files so that I can quickly access the files I frequently use.
    - *Handled by: Recent and Favorite Files Service*

Collaboration Management Module

1. As a user, I want to share my files with other users via links or direct sharing so that I can collaborate with them.
    - *Handled by: Sharing Service*
2. As a user, I want to set permissions (view, edit, comment) when sharing files so that I can control how others interact with my files.
    - *Handled by: Permissions Management Service*
3. As a user, I want to collaborate on documents in real time with my team so that we can work together efficiently.
    - *Handled by: Real-time Collaboration Service*
4. As a user, I want to manage and revoke access to shared links so that I can maintain the privacy of my files.
    - *Handled by: Link Management Service*

Notification Management Module

1. As a user, I want to receive notifications for important actions (e.g., file shared with me, changes to a file I own) so that I can stay informed.
    - *Handled by: Notification Service*
2. As a user, I want to receive notifications in real time on my device so that I am immediately aware of any updates or events.
    - *Handled by: Notification Delivery Service*

Activity Log Management Module

1. As a user, I want to view a log of all actions (e.g., edits, uploads, deletions) performed on my files so that I can track changes.
    - *Handled by: Event Logging Service*
2. As a user, I want to see previous versions of a file so that I can restore an older version if needed.
    - *Handled by: File Version Control Service*

Analytics Management Module

1. As a user, I want to view insights about my storage usage (e.g., total space used, file type distribution) so that I can better manage my files.
    - *Handled by: System Usage Analytics Service*
2. As an admin, I want to generate reports about system usage (e.g., most active users, storage trends) so that I can make data-driven decisions.
    - *Handled by: Report Generation Service*

Modules in this System

1. User Management [handles all operations related to user accounts, authentication, and account lifecycle management]
2. File Storage Management [handles file-related services]
3. Search and Filter Management [handles operations related to searching files and filtering based on various metadata]
4. Collaboration Management [handles file sharing and collaborating operations]
5. Notification Management [generates notifications and sends to the subscribers]
6. Activity Management [handles file-specific activities and modifications history]
7. Analytics Management [provides insights about system usage and generates reports]

Microservices Under Each Module

1. User Management:
    a. User Registration and Profile Service: manage user account creation and profile updates
    b. Authentication and Authorization Service: ensures secure user access to the system and manages permissions
    c. Account Recovery Service: helps users regain access to their accounts if credentials are lost or compromised
    d. Billing and Payment Service: manages subscriptions, payments, and storage quota upgrades for users

2. File Management:
    a. File Upload Service
    b. File Storage Service
    c. File Retrieval and Download Service

3. Search and Filter Management:
    a. Indexing Service
    b. Search Query Processing Service
    c. Recent and Favorite Files Service

4. Collaboration Management:
    a. Sharing Service
    b. Permissions Management Service
    c. Real-time Collaboration Service
    d. Link Management Service

5. Notification Management:
    a. Notification Service
    b. Notification Delivery Service

6. Activity Log Management:
    a. Event Logging Service
    b. File Version Control Service

7. Analytics Management:
    a. System Usage Analytics Service
    b. Report Generation Service

Necessary Relational Databases

1. User Management Module

Database: PostgreSQL or MySQL
Purpose:

- Store user data such as email, username, password hash, and profile details.
- Maintain user subscription plans, billing information, and transaction history.
- Track user authentication tokens and session data.

2. Notification Management Module

Database: PostgreSQL or MariaDB

Purpose:

- Store notifications metadata (e.g., notification type, status, timestamps).
- Track notification delivery status (sent, failed, etc.).

Schema Design:

- Notifications Table: (notification_id, user_id, type, content, status, created_at)
- Notification Delivery Table: (delivery_id, notification_id, delivery_time, delivery_status)

3. Analytics Management Module

Database: PostgreSQL or Amazon Aurora (for transactional and summary data)

Purpose:

- Store system usage summaries, periodic reports, and user-specific analytics.

Schema Design:

- Usage Analytics Table: (user_id, date, total_storage_used, active_files_count)
- Reports Table: (report_id, user_id, type, generated_at, report_file_path)

Distributed Databases

1. File Management Module

- Database: Amazon S3 or Google Cloud Storage (GCS) for file storage; MongoDB for metadata
- Purpose:
    - Store the files and associated metadata like file size, type, upload time, and owner.
- Schema Design (MongoDB):
    - Files Metadata Collection: { file_id, user_id, file_name, file_path, size, file_type, created_at }

2. Search and Filter Management Module

- Database: Elasticsearch
- Purpose:

    - Index file and folder metadata for efficient full-text search and filtering.
- Schema Design:
    - File Index: (file_id, file_name, file_type, owner_id, tags, permissions)
    - Folder Index: (folder_id, folder_name, owner_id, tags, permissions)

3. Collaboration Management Module

- Database: Cassandra or MongoDB
- Purpose:
    - Store sharing links, permissions, and real-time collaboration data.
- Schema Design (MongoDB):
    - Sharing Links Collection: { link_id, file_id, shared_with_user_id, permission_level, expiration_date }
    - Real-time Collaboration Collection: { file_id, active_users: [user_id], last_edit_timestamp }

4. Activity Log Management Module

- Database: Apache Kafka (for real-time logging) and Amazon DynamoDB (for long-term storage)
- Purpose:
    - Track user and system events (file edits, downloads, and version changes)
- Schema Design (DynamoDB):
    - Event Logs Table: { event_id, user_id, file_id, event_type, timestamp }
    - File Version Table: { version_id, file_id, version_number, edited_by, timestamp }

5. Notification Management Module

- Database: Redis or RabbitMQ (for real-time notifications) and Cassandra (for durable storage)
- Purpose:
  - Real-time notification delivery and persistent storage for historical notifications.
- Schema Design (Cassandra):
  - Notifications Table: { notification_id, user_id, type, content, status, timestamp }

6. Analytics Management Module

- Database: Apache HBase or Google BigQuery
- Purpose:
  - Store and analyze large-scale analytics data (user activity, file interactions, storage usage).
- Schema Design:
  - Analytics Table: { metric_id, user_id, metric_name, value, timestamp }

Summary

Module	Relational Database	Distributed Database
User Management	PostgreSQL, MySQL	-
File Management	-	Amazon S3, MongoDB
Search and Filter	-	Elasticsearch
Collaboration Management	-	Cassandra, MongoDB
Activity Log Management	-	Kafka, DynamoDB
Notification Management	PostgreSQL	Redis, Cassandra
Analytics Management	PostgreSQL, Amazon Aurora	HBase, BigQuery

Diagram of Dependency among Services and Storages

List of Databases Needed for Google Drive Services

1. User Management

User Database:
Usage: Stores user account information, including profile details, authentication credentials, billing, and subscription details.

Billing Database:
Usage: Tracks payment transactions, subscription plans, and storage quota upgrades.

2. File Management

File Metadata Database:
Usage: Stores metadata for files, such as names, sizes, types, timestamps, version history, and ownership details.

File Storage Database:
Usage: Stores actual file data in a distributed storage system.

3. Search and Filter Management

Search Index Database:
Usage: Indexes file metadata for efficient and quick search queries.

Favorites Database:
Usage: Tracks user-specific recent and favorite files for easier retrieval.

4. Collaboration Management

Collaboration Database:
Usage: Tracks shared files, permissions, real-time collaboration sessions, and link-sharing configurations.

5. Notification Management

Notification Database:
Usage: Stores notifications and their delivery statuses for users.

6. Activity Log Management

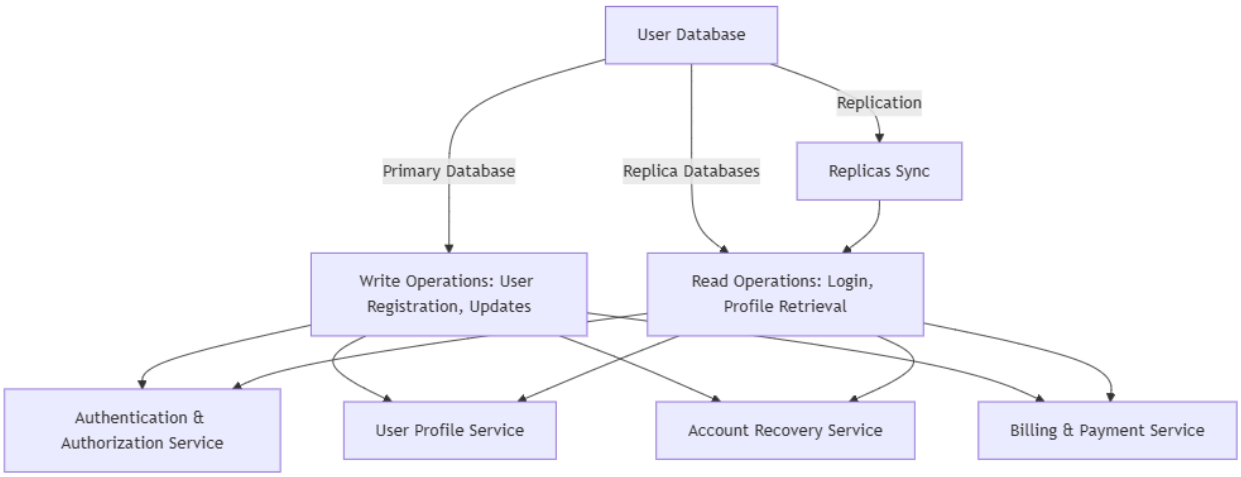
Activity Log Database:
Usage: Logs file activities (e.g., edits, downloads, shares) and manages version history.

Replication Strategies for Databases

1. User Database

- Related Services:
    - Authentication and Authorization Service
    - User Profile Service
    - Account Recovery Service
    - Billing and Payment Service
- Replication Strategy: Primary-Replica Replication
- Why Needed:
    - Read-heavy operations, especially authentication, benefit from replicas to offload the primary database.
    - Ensures data consistency for writes (e.g., user account updates).
- How to Replicate:

- The primary database will handle writes (e.g., user registrations and updates).
- Replicas will handle read operations (e.g., login requests, user profile retrieval).



2. File Metadata Database

Related Services:

- File Upload Service
- File Storage Service
- File Retrieval and Download Service
- Search Query Processing Service
- Recent and Favorite Files Service

Replication Strategy: Change Data Capture (CDC)

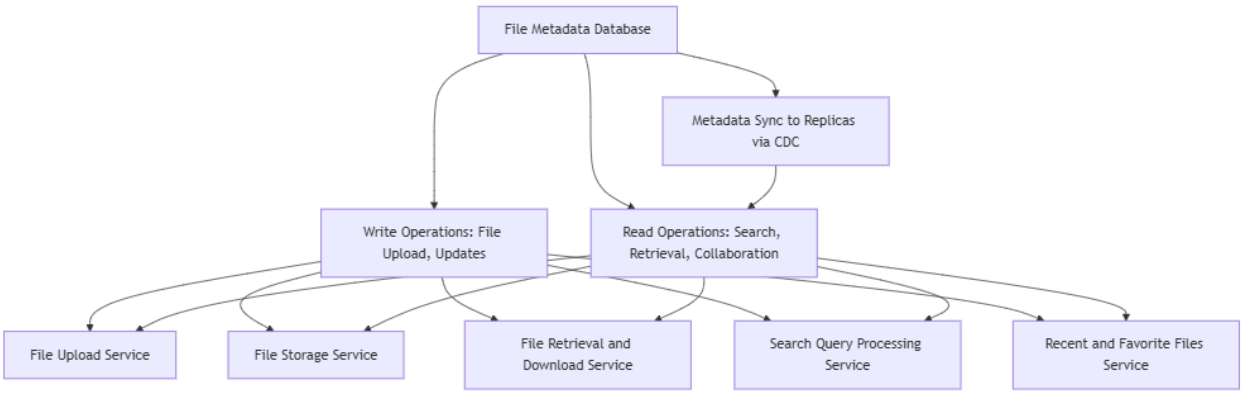
Why Needed:

- Metadata changes (like file uploads, and updates) need to be synchronized with services like search, real-time collaboration, and versioning.

- Low latency is critical for services dependent on metadata (e.g., search index).

How to Replicate:

- Use CDC to stream metadata changes from the primary to replicas.
- The replicas will quickly reflect changes like file uploads and metadata modifications in the search or collaboration services.



3. File Storage Database

Related Services:

- File Upload Service
- File Storage Service
- File Retrieval and Download Service

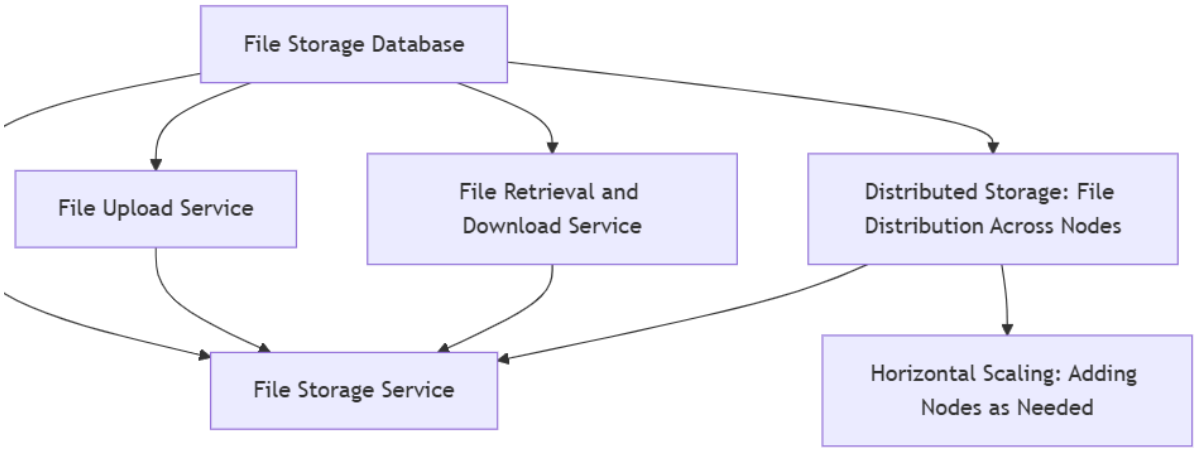
Replication Strategy: Distributed Storage

Why Needed:

- This database manages large file storage, and distributed storage ensures fault tolerance and scalability.
- Multiple storage nodes provide availability and performance for large files.

How to Replicate:

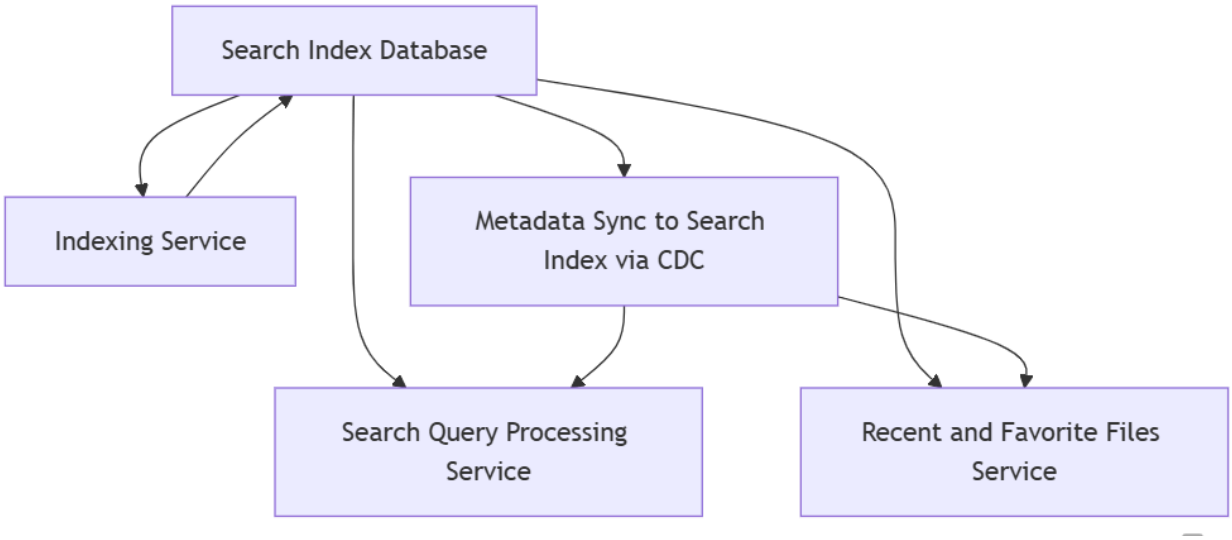
- Files are distributed across multiple nodes, ensuring redundancy.
- The system can scale horizontally by adding new nodes to the distributed system as the file storage demand increases.



4. Search Index Database

- Related Services:
  - Indexing Service
  - Search Query Processing Service
  - Recent and Favorite Files Service

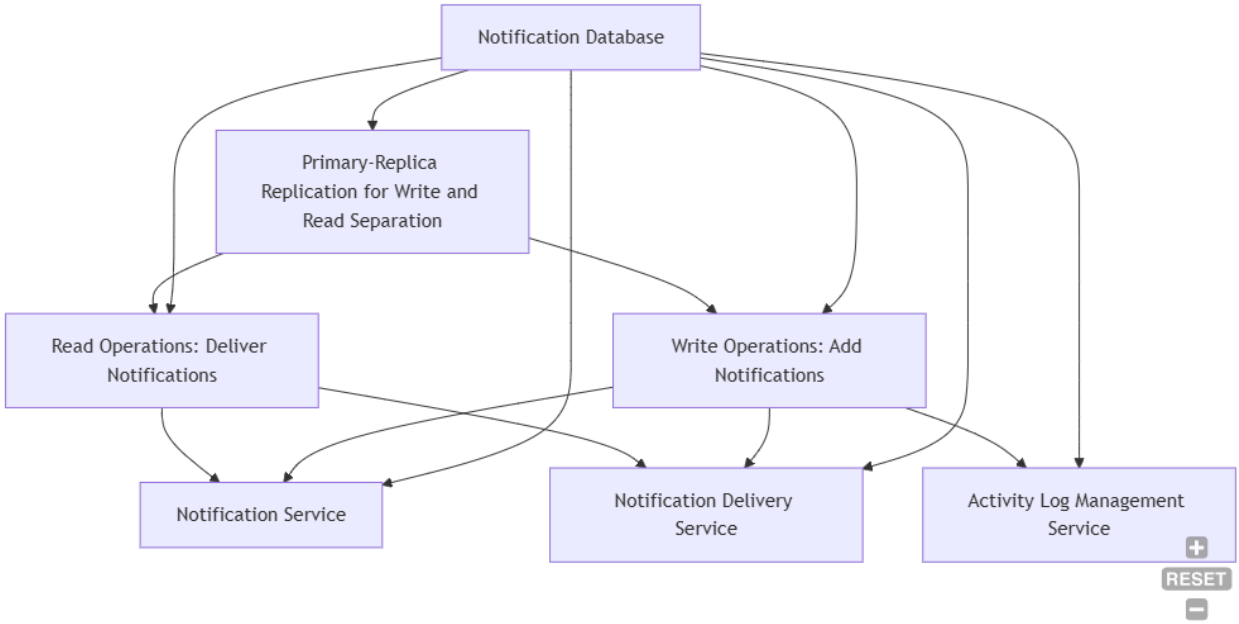
- Replication Strategy: Change Data Capture (CDC)
- Why Needed:
    - Near real-time synchronization between file metadata and search indexes is necessary to maintain accurate search results.
    - Ensures that newly uploaded or updated files are reflected immediately in search results.
- How to Replicate:
    - CDC captures changes in file metadata from the primary and applies those changes to search index replicas.
    - Replicas handle search query processing while keeping search indexes up to date.



5. Notification Database

- Related Services:
    - Notification Service
    - Notification Delivery Service
    - Activity Log Management Service
- Replication Strategy: Primary-Replica Replication
- Why Needed:

    - Real-time notifications are required for users, and the database needs to handle high write throughput.
    - Replicas allow the system to deliver notifications without overloading the primary database.
- How to Replicate:
    - The primary database will handle writes (e.g., adding new notifications for users).
    - Replicas will handle read operations, allowing notifications to be delivered quickly to users.



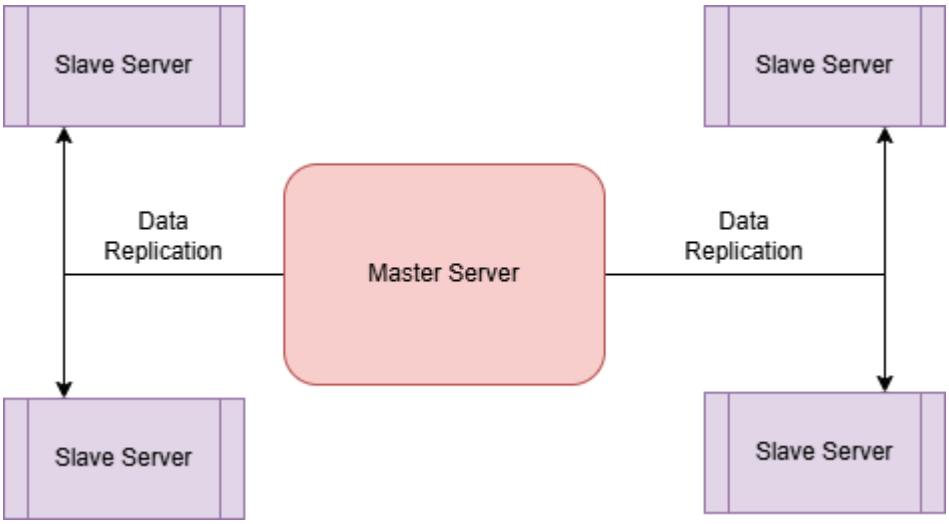
Summary Table

Database	Related Services	Replication Strategy	Why Needed	How to replicate
User Database	Authentication, User Profile, Account Recovery, Billing	Primary-Replica Replication	Offload read-heavy requests (e.g., logins) while ensuring data consistency.	Primary handles write; replicas handle reads.
File Metadata Database	File Upload, Storage, Retrieval, Search	Change Data Capture (CDC)	Stream real-time updates to dependent services like search and collaboration.	CDC captures metadata changes and applies them to replicas.
File Storage Database	File Upload, Storage, Retrieval	Distributed Storage	Manage large file storage across distributed nodes for fault tolerance.	Files split across distributed storage nodes for redundancy.
Search Index Database	Indexing, Search Query, Recent Files	Change Data Capture (CDC)	Keep search indexes in sync with file metadata for accurate search results.	CDC captures file metadata changes and updates replicas.
Notification Database	Notification Service, Delivery, Activity Logs	Primary-Replica Replication	Handle real-time notifications while offloading read-heavy requests.	Primary handles write, and replicas handle reads for delivery.

Replication Strategies from a Client-Server Perspective

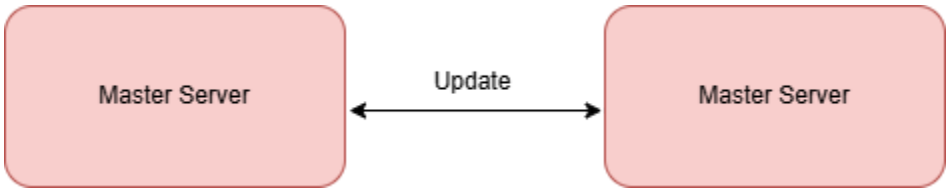
1. Master-Slave Replication

Data changes are written to a central Master Server and propagated to multiple Slave Servers, ensuring data availability and fault tolerance.



2. Multi-Master Replication

Changes can be made on multiple data centers, and updates are synchronized across all replicas with conflict resolution.



3. Eventual Consistency

Multiple Replicas:

- Data is stored across several servers for fault tolerance and scalability.

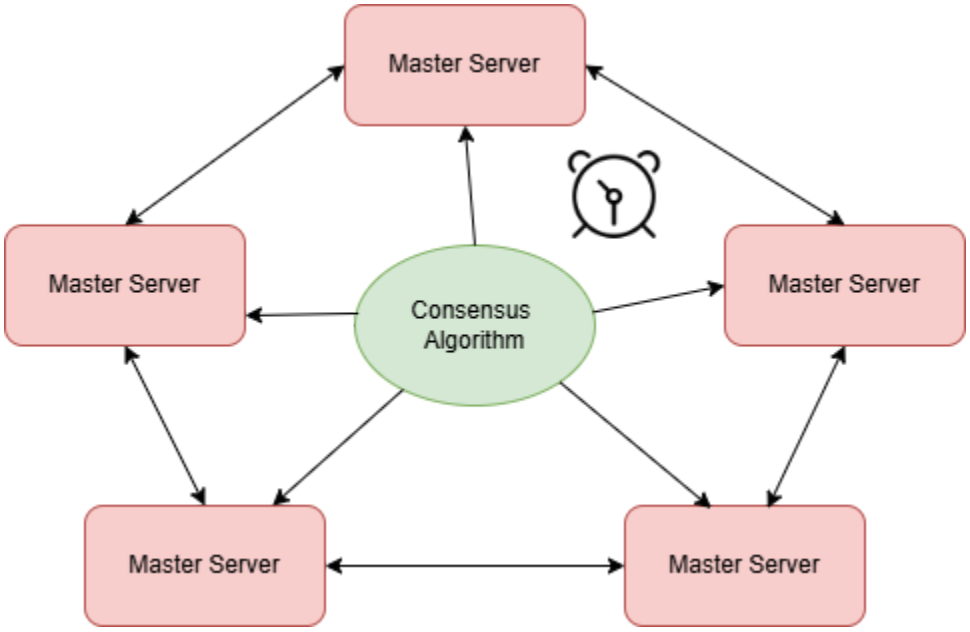
- These servers may temporarily hold slightly different states due to network delays or node failures.

Consensus Protocol:

- Algorithms like Paxos or Raft are used to ensure consistency.
- They achieve agreement among servers on the data state, even during failures.

Convergence Over Time:

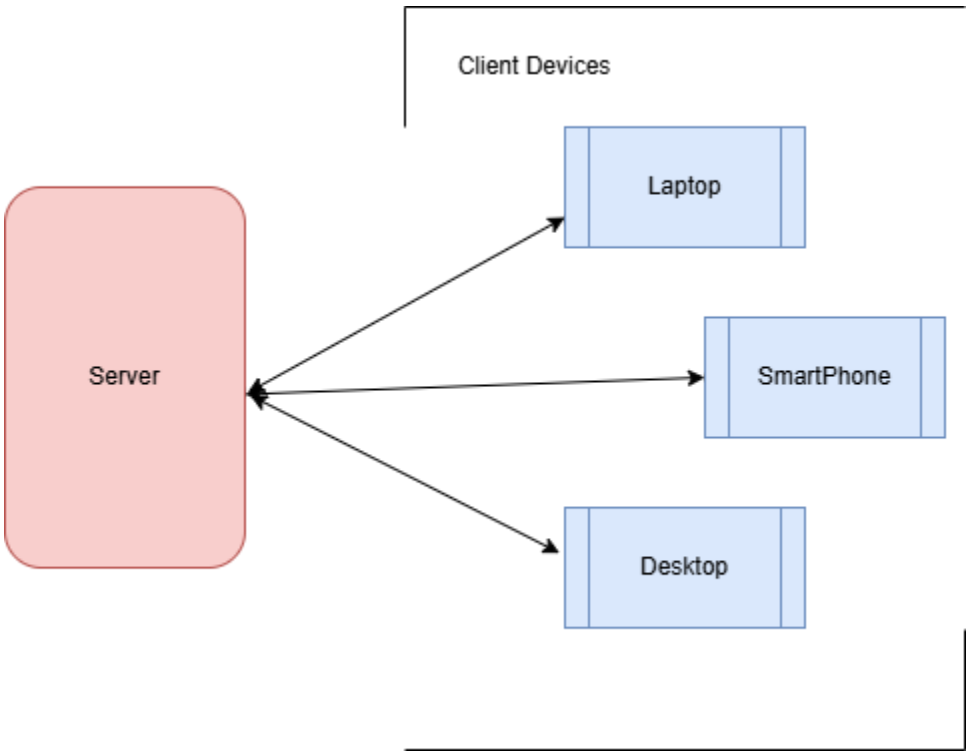
- Replicas may not be synchronized immediately.
- Emphasize that the system achieves consistency eventually, not instantly.



Synchronization Strategies from Client-Server Perspective

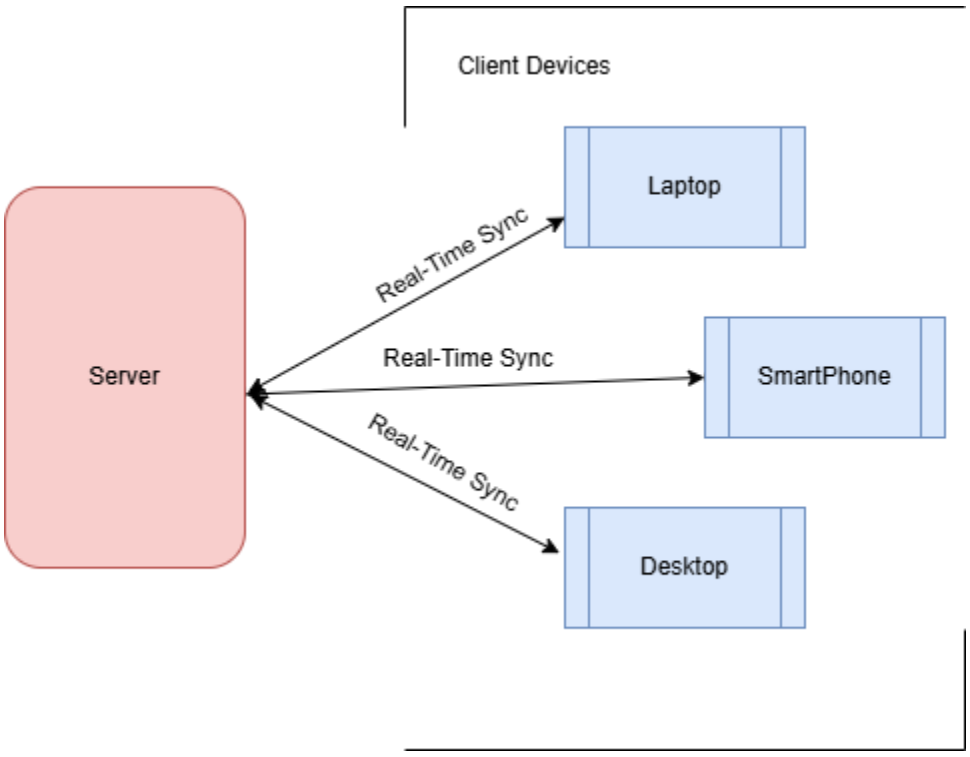
1. Client-Server Synchronization

Clients synchronize their local data with a central server. A change log keeps track of updates.



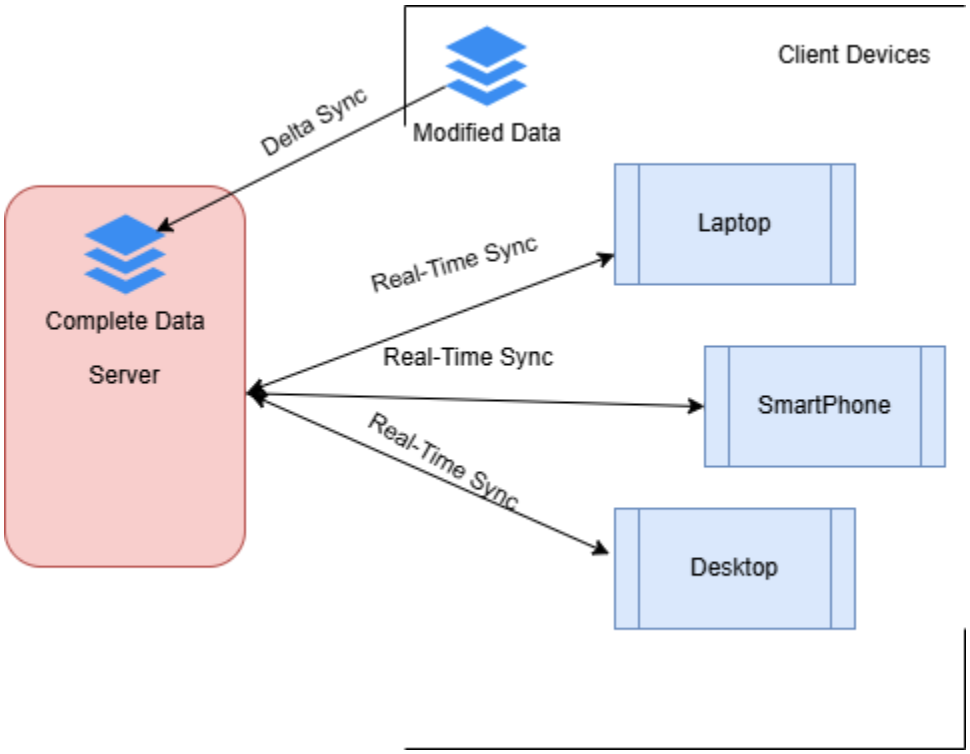
2. Real-Time Synchronization

Uses WebSockets or long-polling to provide real-time updates for collaborative editing.



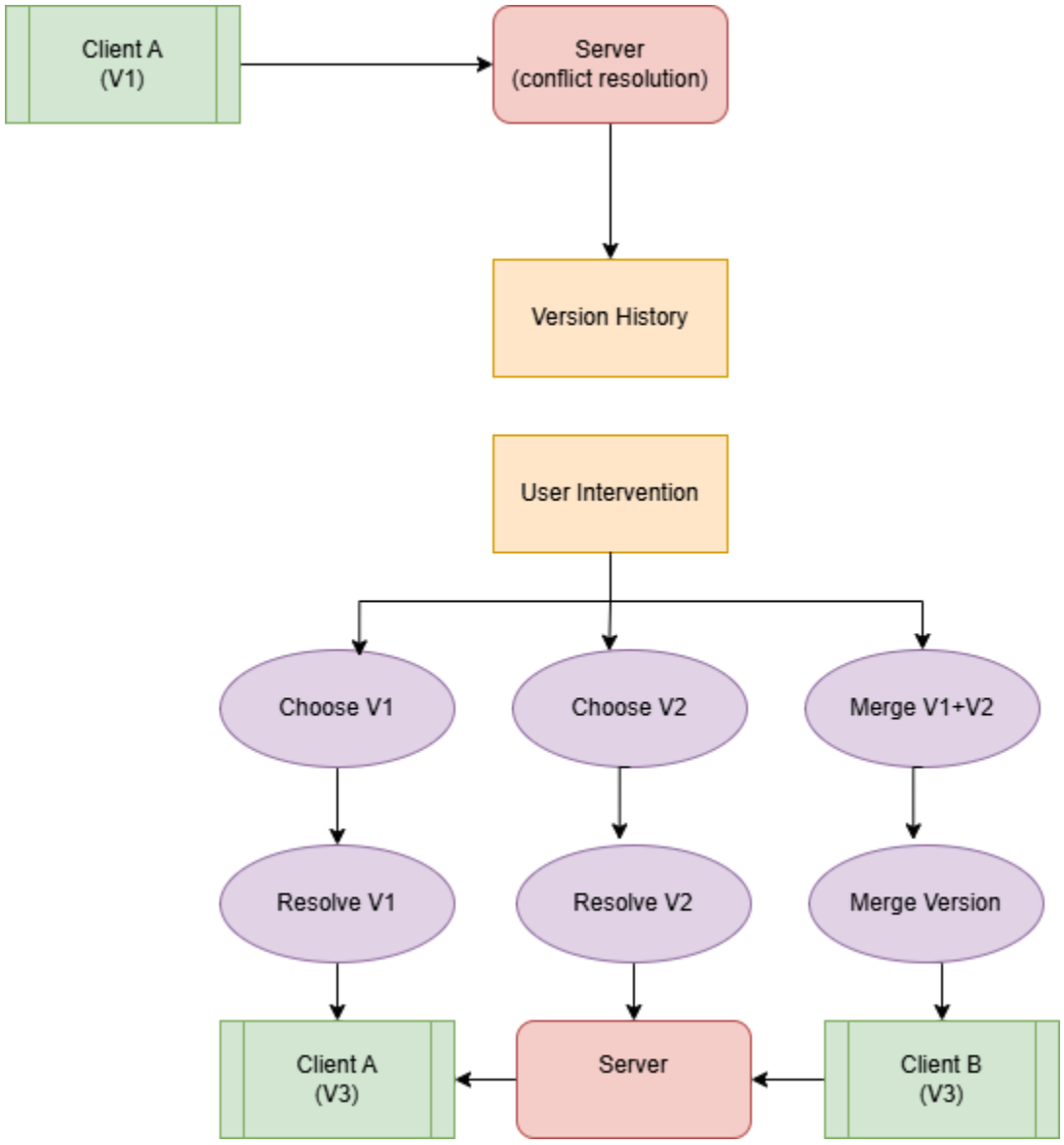
3. Delta Synchronization

Only the modified portions of files (diffs) are transferred during synchronization.



4. Conflict Resolution

Ensures that conflicting updates are resolved using version history or user intervention.



Load Balancer Strategy

1. Centralized Client Interaction

- The Client sends requests to the system, which are routed to dedicated Load Balancers for each module.
- These load balancers act as the entry point to each module, ensuring that requests are directed to the appropriate microservices.

2. Module-Specific Load Balancers

Each module has a dedicated load balancer that ensures traffic distribution, scalability, and fault tolerance.

a. User Management Module

- Load Balancer (LB1):
  - Distributes requests for services like user registration, authentication, account recovery, and billing.
  - For example:
    - Requests for /register are routed to the User Registration and Profile Service.
    - Requests for /auth are sent to the Authentication Service.
- Database Interaction:
  - The services interact with a relational database (PostgreSQL/MySQL) to store user profiles, credentials, sessions, and billing data.

b. File Management Module

- Load Balancer (LB2):
  - Handles traffic for file uploads, storage, and retrieval.
  - Distributes upload requests across multiple instances of the File Upload Service.
  - Ensures storage and retrieval services interact efficiently with the file storage backend.
- Database Interaction:
  - Amazon S3/Google Cloud Storage stores the files themselves.
  - MongoDB stores metadata such as file names, sizes, types, and ownership.

c. Search and Filter Management Module

- Load Balancer (LB3):
    - Routes search and indexing requests to the respective services.
    - Distributes search queries (e.g., file search or metadata search) to multiple instances of the Search Query Processing Service for faster results.
- Database Interaction:
    - Elasticsearch provides efficient indexing and full-text search capabilities.

d. Collaboration Management Module

- Load Balancer (LB4):
    - Distributes requests for services related to file sharing, permissions, real-time collaboration, and link generation.
    - Manages persistent WebSocket connections for real-time updates in the Real-Time Collaboration Service.
- Database Interaction:
    - Cassandra/MongoDB stores sharing links, permissions, and active collaboration sessions.

e. Notification Management Module

- Load Balancer (LB5):
    - Handles requests for creating and delivering notifications.
    - Manages real-time notification delivery efficiently through a message queue system.
- Database Interaction:
    - Redis/Cassandra is used for storing notification metadata and ensuring real-time delivery.

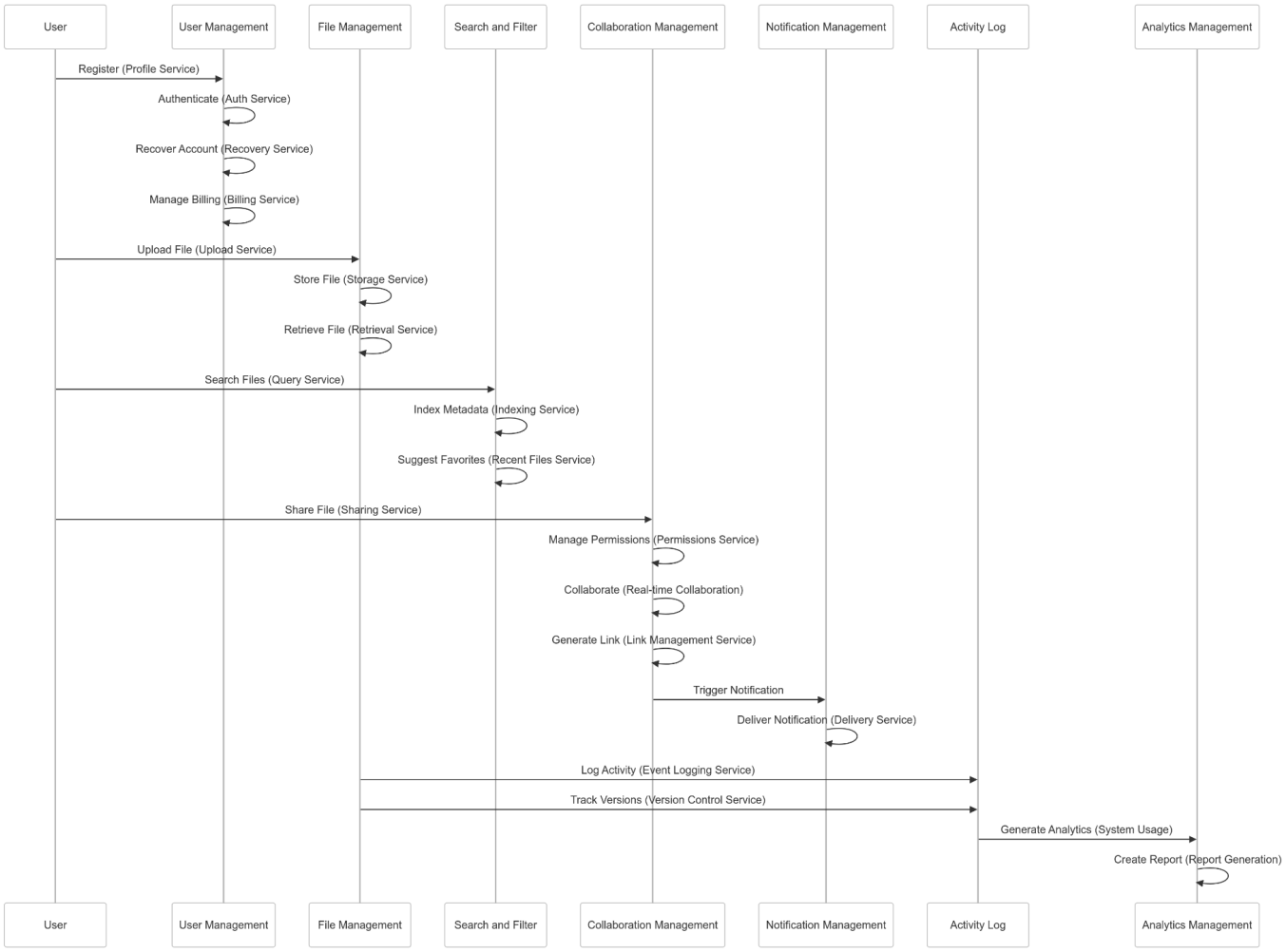
f. Activity Log Management Module

- Load Balancer (LB6):
    - Distributes requests for logging activities and managing file version control.
    - Ensures event logging services handle high-frequency writes efficiently.
- Database Interaction:
    - Kafka captures real-time activity logs.
    - DynamoDB is used for storing logs and file version histories for long-term access.

g. Analytics Management Module

- Load Balancer (LB7):
    - Manages requests for analytics-related tasks, such as generating usage reports and processing system metrics.
    - Distributes traffic between services like System Usage Analytics and Report Generation.
- Database Interaction:
    - HBase/BigQuery is used for storing and processing large-scale analytics data.

Sequence Diagram



[Link to this diagram]