Distributed juke box

read-heavy system

like dispatcher service

CD Player Service

PlayList in-memory db

Jukebox Service

user

cluster
- CD database
- User database
- Song database
- Artist database
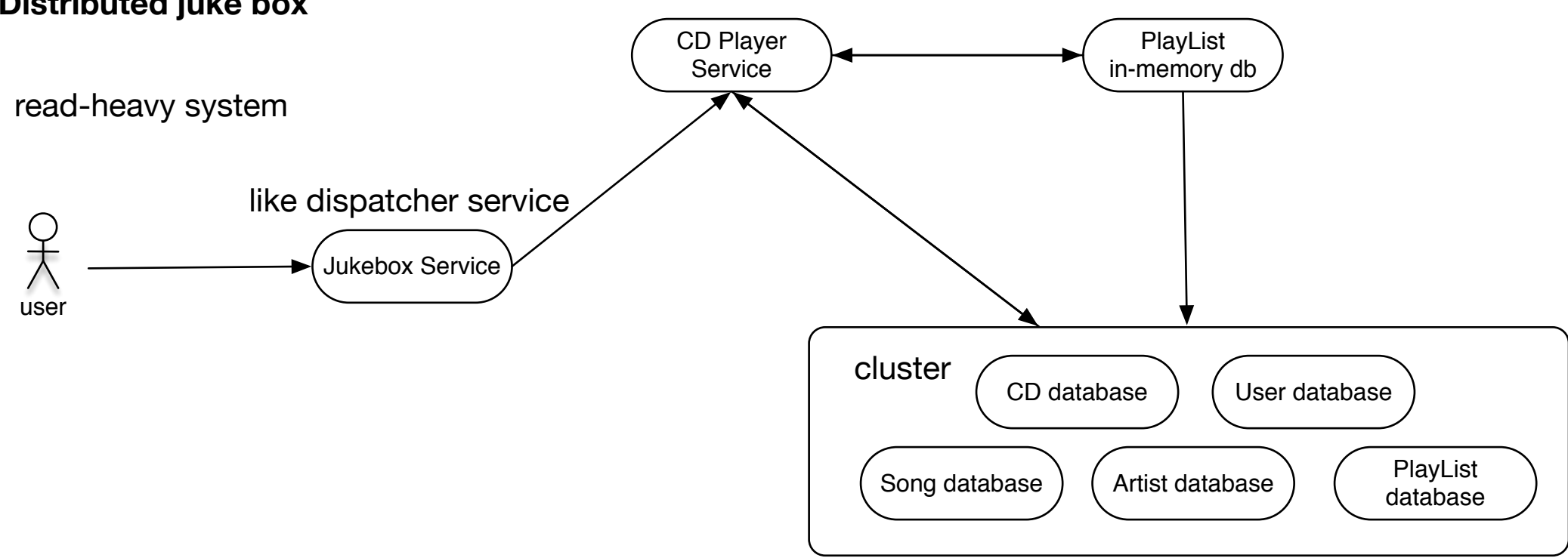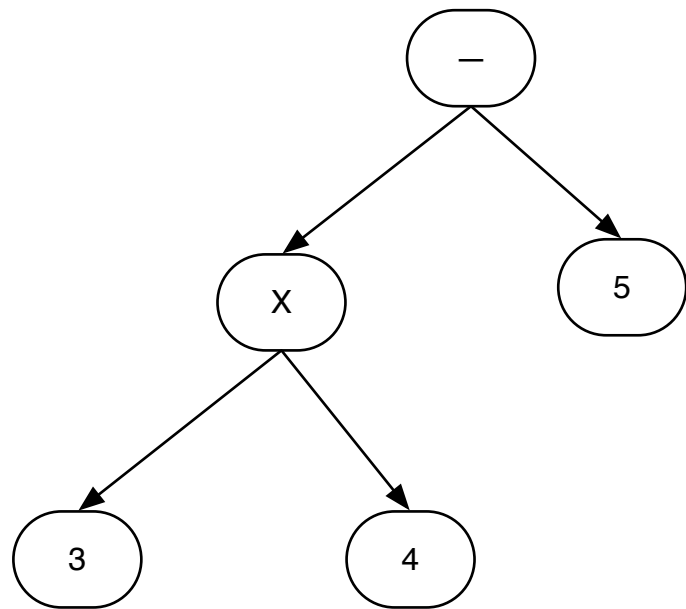- PlayList database

DB schema:
CD: {id, title, artist_id, a list of song id}
Song: {id, title, a list of artist_id}
Artist: {id, name, gender}

Image Processing System

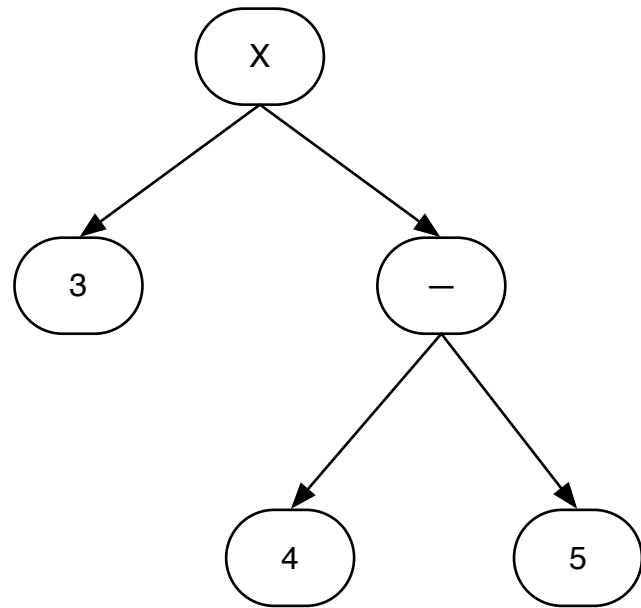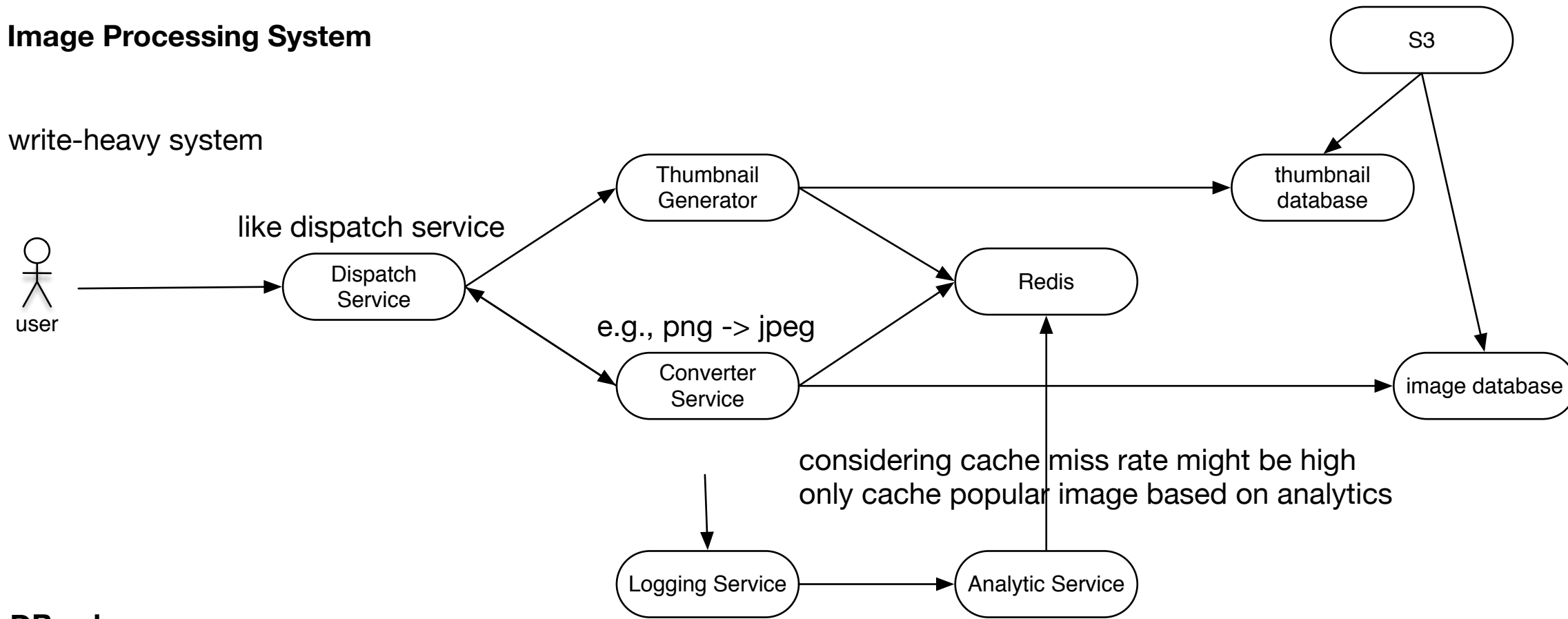write-heavy system

like dispatch service

S3

user

Dispatch Service

Thumbnail Generator

e.g., png -> jpeg

Converter Service

Redis

thumbnail database

image database

Logging Service

Analytic Service

considering cache miss rate might be high
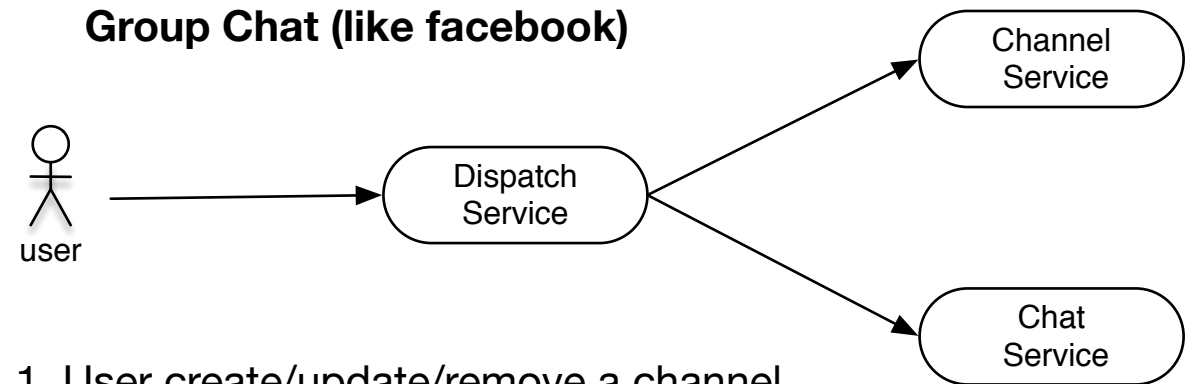only cache popular image based on analytics

DB schema:
Thumbnail: {id, s3_link}
Image: {id, format, s3_link}
Cache: key - id, value - thumbnail_s3_link, image_format_s3_link

assuming that we have an algorithm that can generate a unique identifier for each image

Expression Tree

3 x 4 - 5

3 x (4 - 5)

−
X
5
3
4

X
3
−
4
5

**Online Game Chat Log System -
Group Chat (like facebook)**

user

Dispatch Service

Channel Service

Chat Service

1. User create/update/remove a channel
2. User join a channel
3. Users in a channel chat with each other

DB schema:
Channel: {id, owner_id, a list of user ids}
Chat: {id, }
User: {id, …}