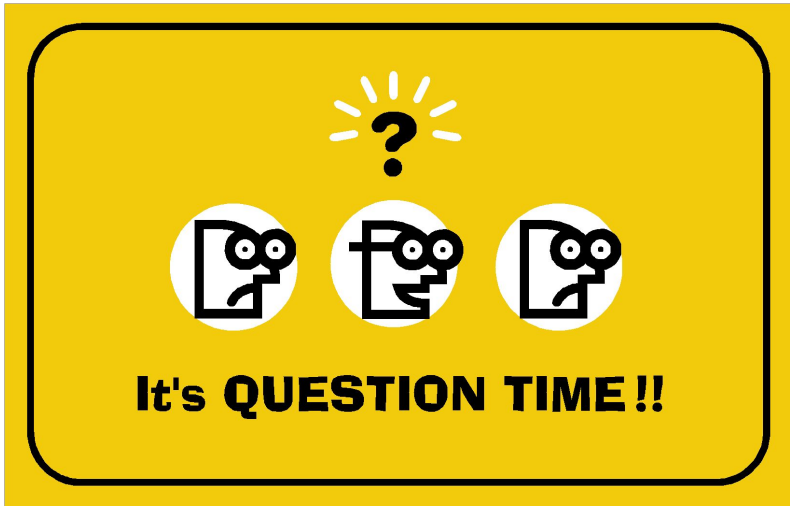
How to design Netflix?

Catherine

A simple survey…

- How are you familiar with Netflix?

A. I’m a frequent user
B. Use it a little
C. Learn a little, but never used it
D. Never heard that



What is Netflix?

- Netflix is a website that allows you to watch movies and television shows.
- Monthly subscription
- Watch on multiple devices, like computer, mobile device, smart TV, etc.
- A media streaming service – watch as the file is loading, as opposed to having to wait for the entire file to load before watching

https://techboomers.com/t/what-is-netflix

Play Around Netflix

Review: SNAKE

- Scenario: case/interface
- Necessary: constrain/hypothesis
- Application: service/algorithm
- Kilobit: data
- Evolve

Scenarios – User Management

- Register
- Login
- Profile Display
- Profile Management
  - multiple profiles, kid restrictions, etc.

Scenario – Movie Info

- Movie Catalog
  - TV shows, Animation, Dramas, Netflix Originals, etc.
- Movie Search
  - Search by titles, people, genres
- Movie Details
  - Overview, Episodes, More like this, Details, etc.

Scenario – Movie Play

- Play operations
  - Play, stop, pause, next, forward, etc.
- History Record
  - Where did you watch last time?

Scenario – Movie Comments

- Read comments for a movie
- Write comments for a movie
  - Give ratings to a movie
- Service within Movie Info (detail)

Scenario – Payment

- First-month free use
- Automatic charge
- Payment info management
- Service within User Management

Scenario – Recommendation

- Customized recommendation for users
- Recommend similar movies
- Background service to support Movie Info (list display, search, details)

Scenarios

1. User Management
2. Movie Info
3. Movie Player
4. Movie Comments
5. Payment
6. Recommendation

Necessary – Why

What is important to a website?

Functional Features

What can you do on the website?

Scenario

Non-Functional Features

How is your experience on the website?

Performance
Scalability
Security – special knowledge
Usability – UI/UX

Necessary – Why

- Performance

  - **How long** does it take to respond to a request?

  - **Measured** by delay from sending requests to displaying response

  - **Influenced** by code quality, which database to use, database schema, web framework, etc.

- Scalability

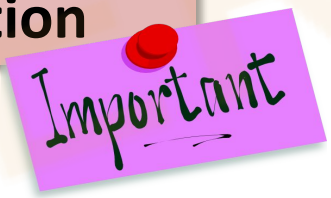
  - **How many** simultaneous users can you support?

  - **Measured** by the simultaneous users to support

  - **Influenced** by performance per machine, how many machines to use, and how they are coordinated to work, etc.

Necessary

Reasonable Assumption + **Logical Deduction**



- Difficult to give correct numbers
- Need rich experience
- Reasonable is OK
- Make it simple in interview

- Clear, logical, and self-contained
- A reasonable, practical solution
- Interviewers care how you think and solve the problem

Necessary – Assumption

- Total subscribers = 100 million
- The Number of Movies = 15,000
- Daily User Average Online Time = 1 hour
- Recommended for SD quality = 3.0 Mbps

You can ask these questions to interviewer!

Necessary – Logical Deduction

KPI

- **RPS and Traffic** (Why traffic? Movie play requires transfer of large data)

	Current	Future = Current * 2
Daily Active Users	100 million * 10% = 10 million	20 million
Peak Active Users	10 million * 1h /24 h * 8 = 3.3 million	6.6 million
RPS	60 requests during an hour	60 requests per hour
	RPS per user: 60/3600s = 1/60	1/60 RPS per user
	Total Peak RPS: 1/60 * 3.3million = 55,000	110,000 RPS
Traffic	Traffic Per User = 3Mb/s	3Mb/s
	Peak Traffic: 3.3 million * 3Mb/s = 10Tb/s	20Tb/s

https://www.finder.com.au/what-is-netflix

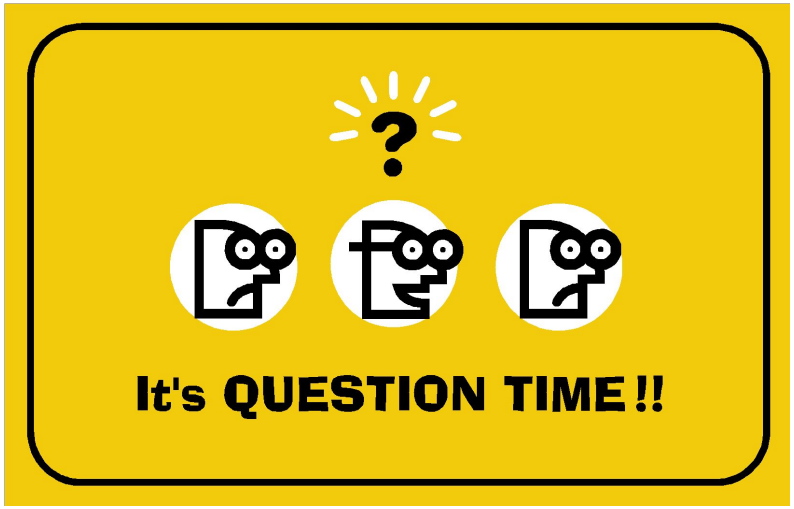
Application – SOA

- How are you familiar with SOA (Service Oriented Architecture)?

A. Learned a lot

B. Heard before

C. Never heard that



Application – SOA

Definition

- Services are provided to the other components by application components, through a communication protocol over a network.
- A service is a discrete unit of functionality that can be accessed remotely and acted upon and updated independently.

Features

1. It logically represents a business activity with a specified outcome
2. It is self-contained
3. It is a black box for its consumers
4. It may consist of other underlying services

Advantages

1. Loose Coupling
2. Flexibility
3. Easier Testing and Debugging
4. Scalability
5. Reusability

Application

- ✓ User Management Service
- ✓ Movie Info Service
- ✓ Movie Play Service
- ✓ Comment Management Service
- ✓ Payment Service
- ✓ Movie Recommendation Service

Kilobit – User Info

- What to store
  - User basic info: username, email, password, etc.
  - Profile info: kid or not, etc.
- How big
  - User Info - 200 million * 1Kb = 200G
- Which data storage
  - Not big, no frequent access or write
  - MySQL

Kilobit – Movie Info

- What to store
  - Movie information: name, description, star, etc.
  - Movie storage url
- How big
  - 30,000 * 10Kb = 300Mb
- Which data storage
  - Not big, frequent access, search by columns
  - MySQL with proper index

Kilobit – Movie Play

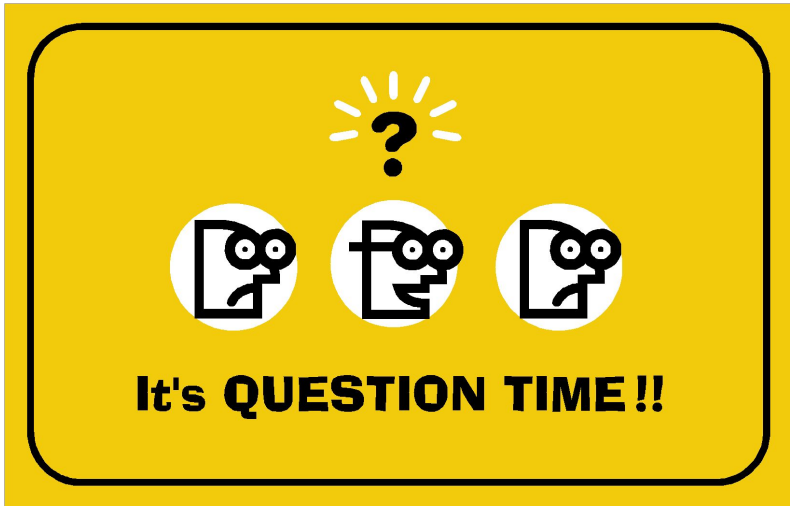
- Where to store movie files?

A. Relational Database Management Systems

B. NoSQL Database Management Systems

C. File Servers

D. Others



Kilobit – Movie Play

- What to store
  - Movie file
  - Movie play info
- How big
  - 30, 000 * 40G = 1200T
- Which data storage
  - Very big, frequent access with unique id
  - Hard Disk Drives Based Server
  - Movie play info -> MySQL

Kilobit – Movie Comments

- What to store
  - Comment info: who, when, what, for which movie
- How big
  - 30,000 * 1,000 * 1Kb = 30Gb
- Which data storage
  - Frequent access with unique id, no cross search
  - MongoDB (document NoSQL)

Kilobit – Payment

- What to store
  - Payment info(security): method, balance, etc.
  - Subscription info: expiration date, discount, etc.
- How big
  - 200 million * 100 * 1Kb = 20T
- Which data storage
  - Frequent write, secure transaction
  - Cassandra, and MySQL

Kilobit – Recommendation

- What to store
  - Similar movies for each movie
  - Recommendation for user
  - User preferences and history records
- How big
  - 200 million * 1Mb = 200T
- Which data storage
  - Quick access, weak consistent, big computation
  - MySQL, Offline S3, Hbase

Evolve

Which one depicts which one?

What about more users?

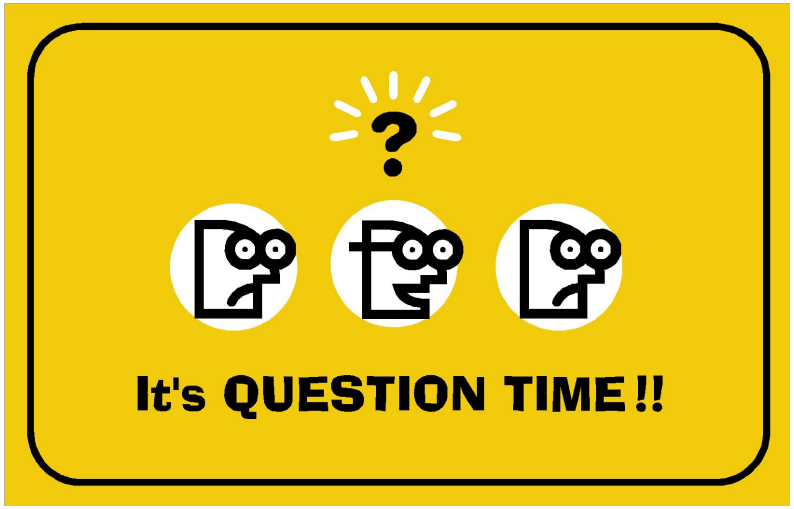
What about other modules?

What about architecture details?

Horizontal Functionality

Scalability

Vertical Functionality



Evolve

Which one depicts which one?

What about more users?

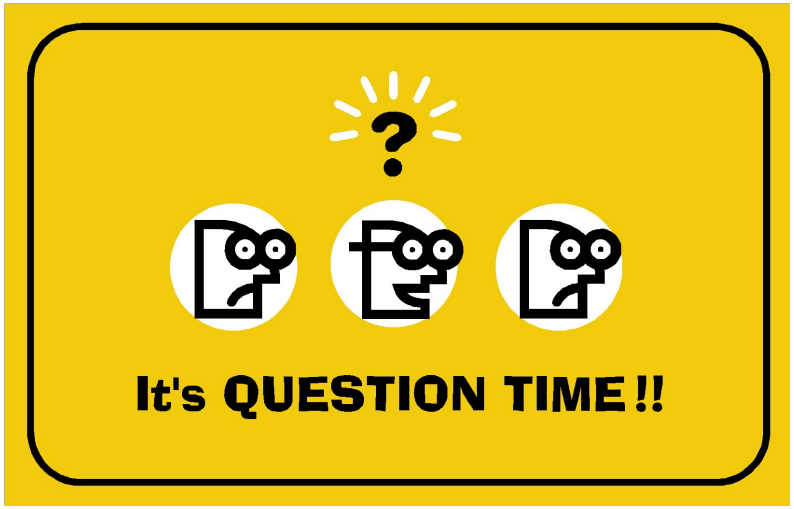
What about other modules?

What about architecture details?

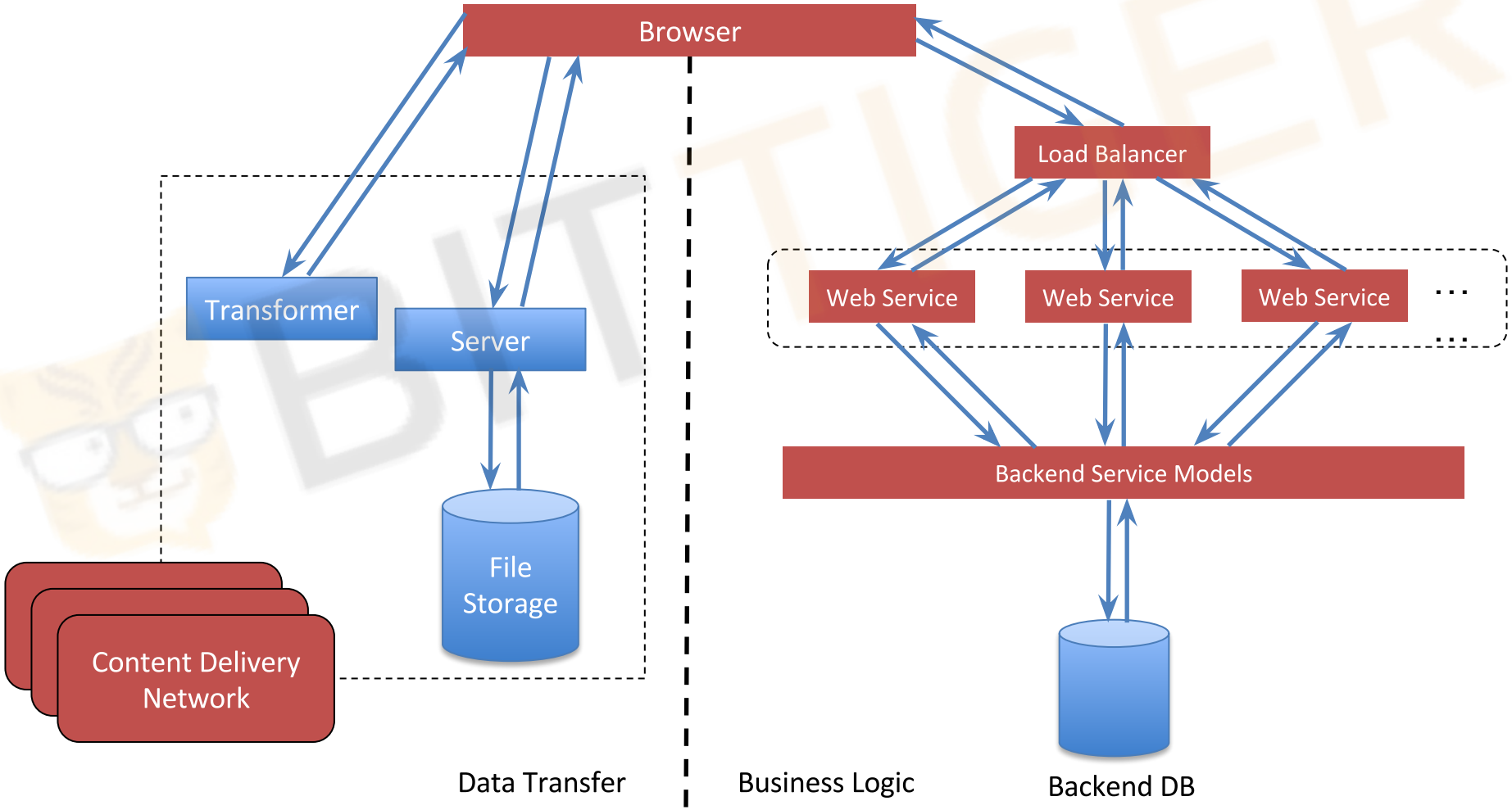
Horizontal Functionality

Scalability

Vertical Functionality

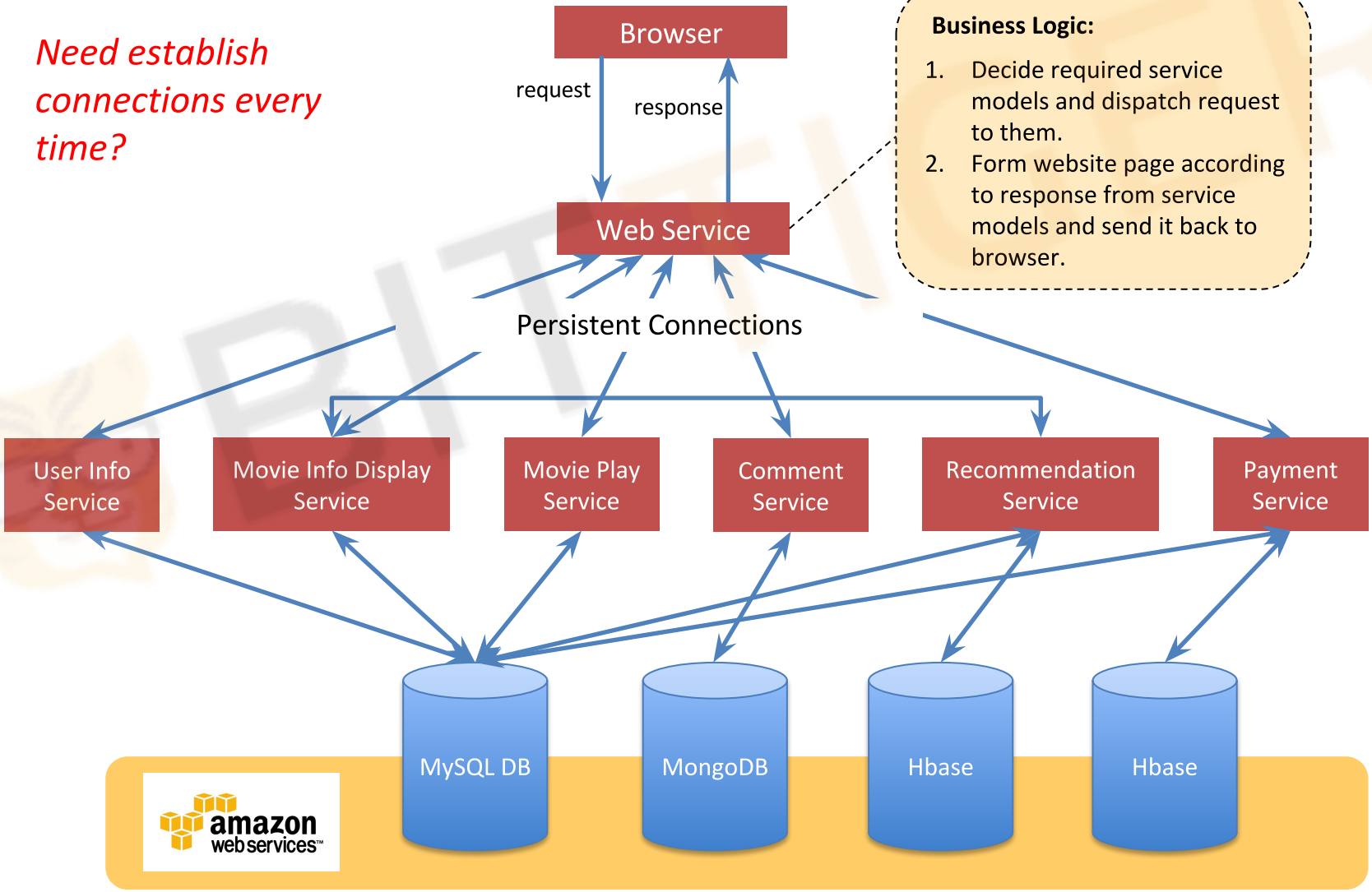


Overview Architecture

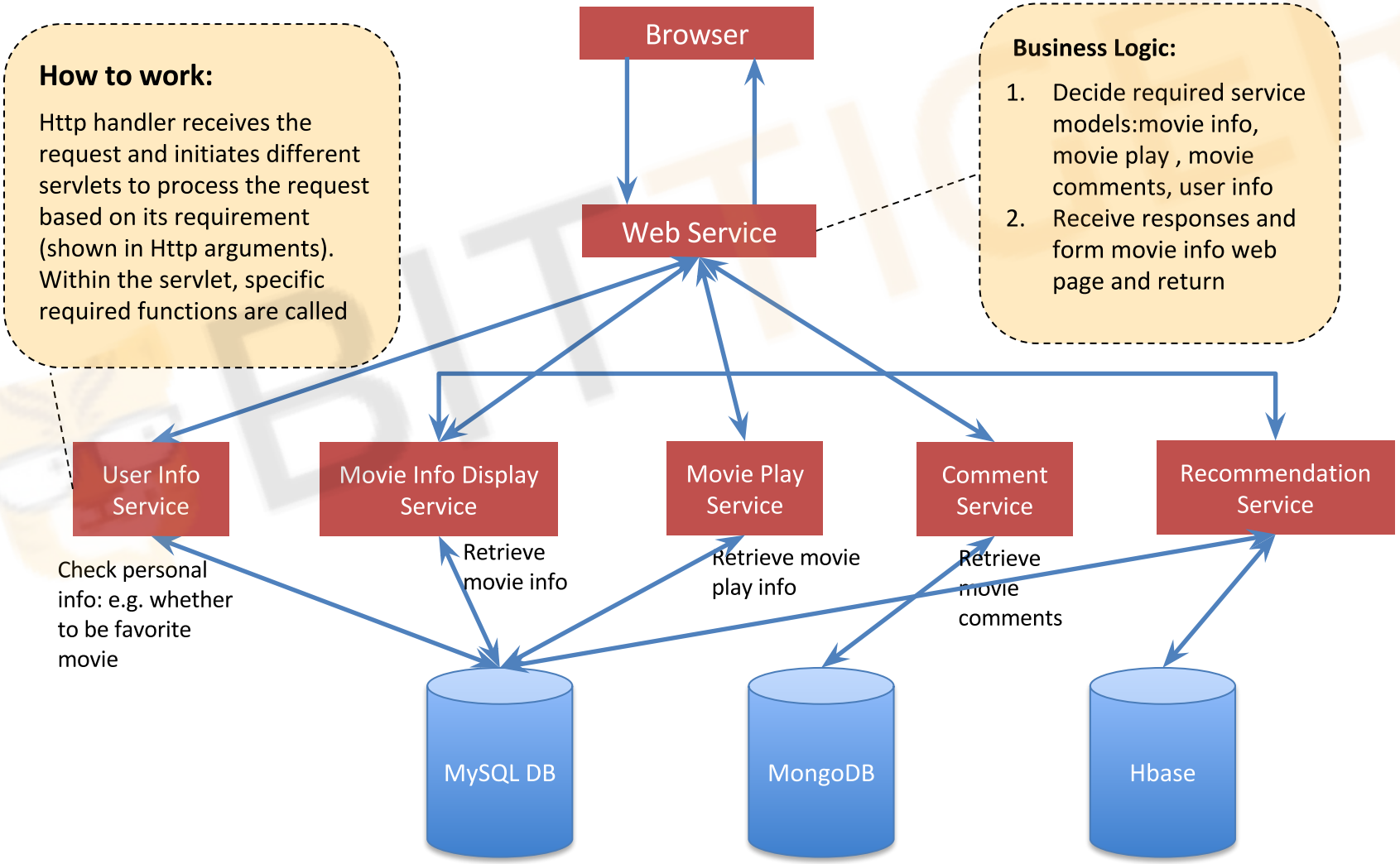


Business Logic

Need establish connections every time?



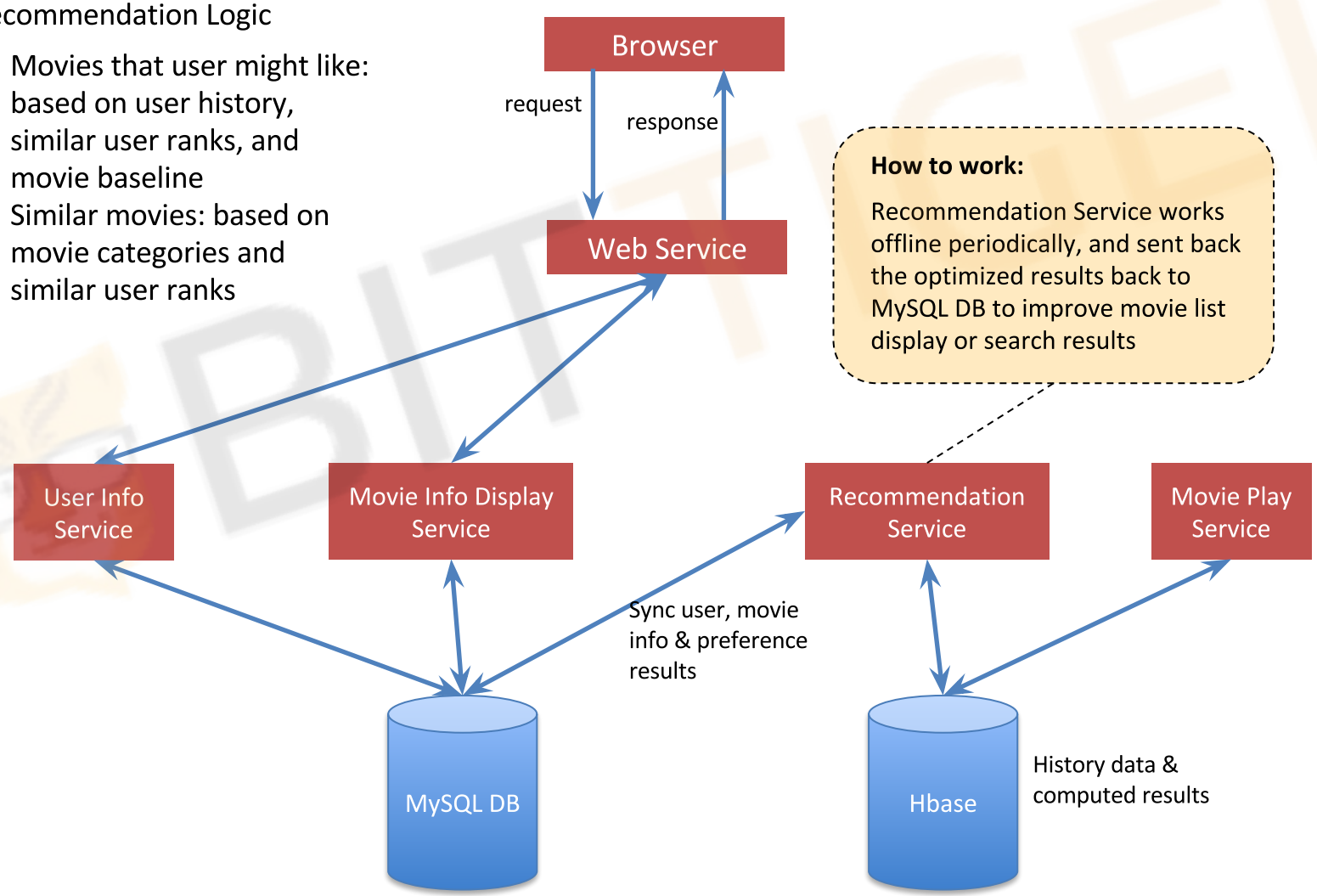
Show Movie Detail Page



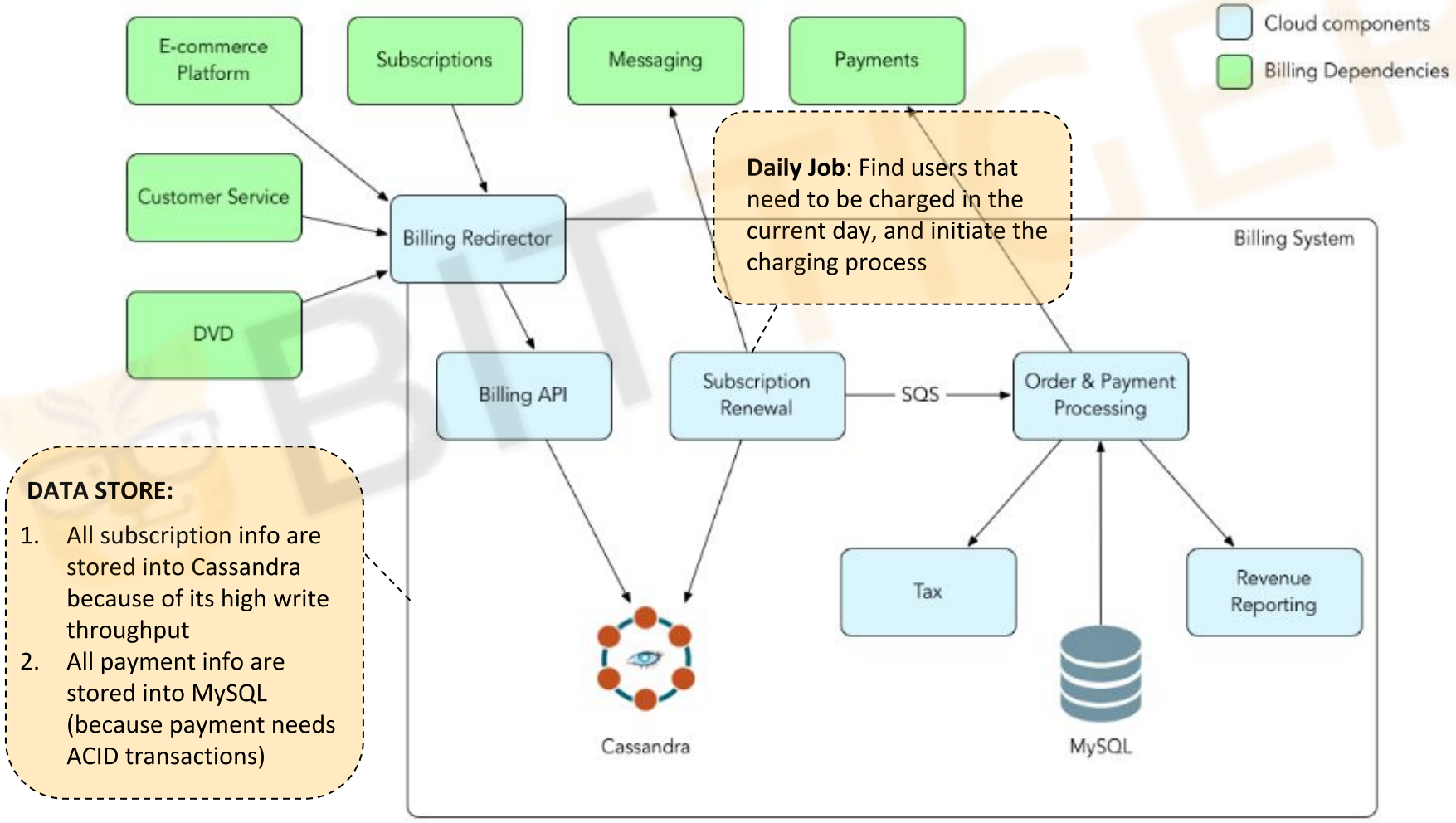
Recommendation Service

Recommendation Logic

1. Movies that user might like: based on user history, similar user ranks, and movie baseline
2. Similar movies: based on movie categories and similar user ranks

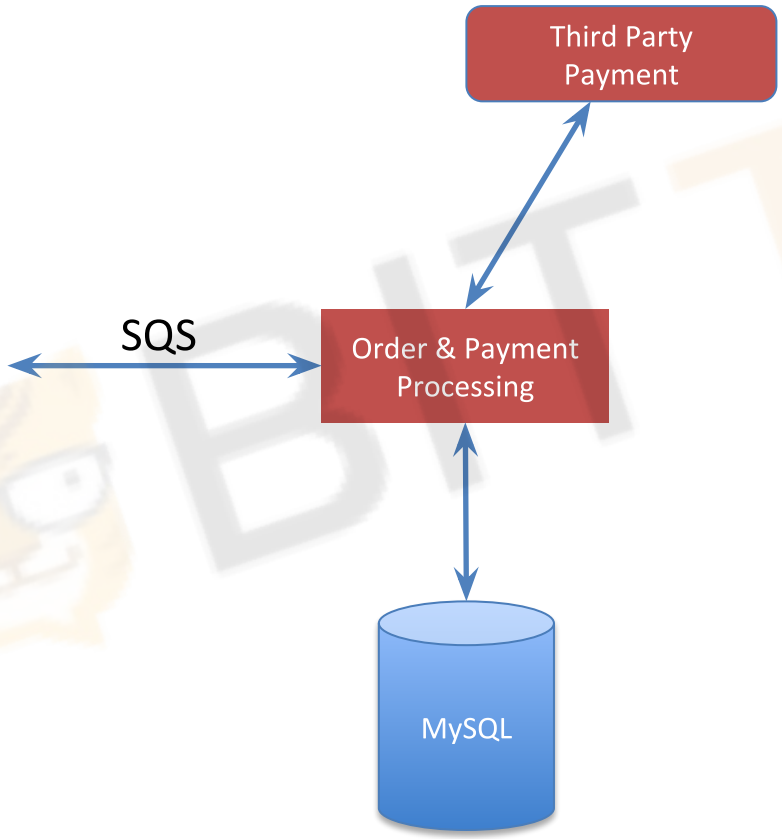


Netflix Payment Architecture



http://techblog.netflix.com/

Payment Process



workflow:

1. Retrieve users that needs charging from SQS and corresponding info: subscription ID, charge fees
2. Retrieve user payment method from MySQL
3. Prepare payment request based on obtained info
4. Check whether the request is pending (Such payment request is sent, but not finished due to server failure or third party payment problems), if yes, ignore (avoid paying twice)
5. Send request to Third Party Payment, meanwhile, record this request in pending list in MySQL
6. If no response from Third Party Payment, retry or failure process (send payment failure back to SQS)
7. If OK response from Third Party, update pending list, send confirmation back to SQS.

Summary

- SNAKE Analysis
- Architecture – Six Service Components (SOA)
- Overview Architecture (Business Logic & Data Transfer)
- Service Component Details (Movie Detail, Recommendation, and Payment)