EP18: Build a chat application. Also...

ALEX XU
AUG 6, 2022

♡ 94 💬 5 ⟳ Share ···

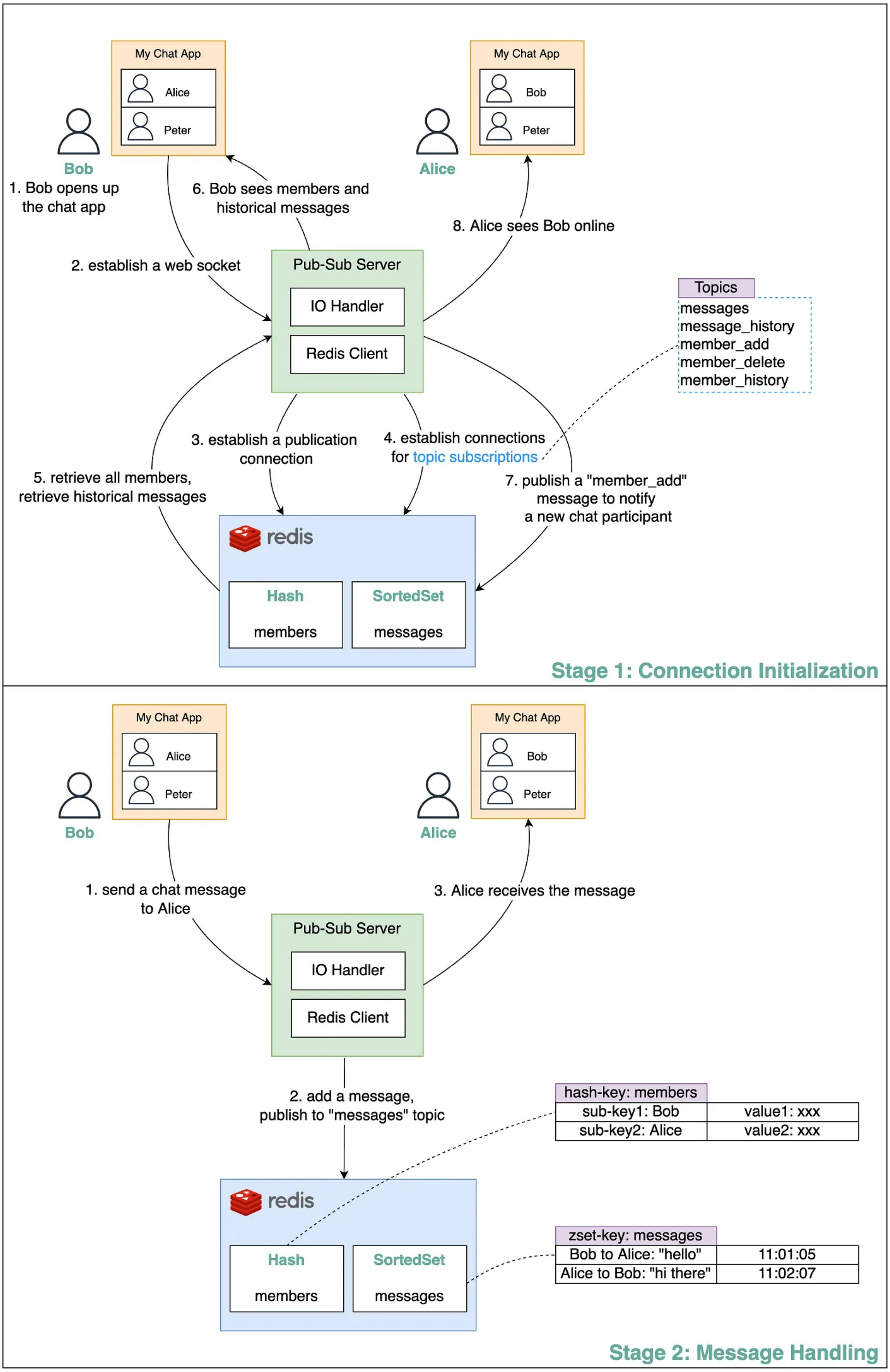
Build a simple chat application

How do we build a simple chat application using Redis?

The diagram below shows how we can leverage the pub-sub functionality of Redis to develop a chat application.

How to Build a Chat Application using Redis?

blog.bytebytego.com


My Chat App
Alice
Peter
Bob
1. Bob opens up the chat app
6. Bob sees members and historical messages
My Chat App
Bob
Peter
Alice
8. Alice sees Bob online
2. establish a web socket
Pub-Sub Server
IO Handler
Redis Client
Topics
messages
message_history
member_add
member_delete
member_history
3. establish a publication connection
4. establish connections for topic subscriptions
5. retrieve all members, retrieve historical messages
7. publish a "member_add" message to notify a new chat participant
redis
Hash
members
SortedSet
messages
Stage 1: Connection Initialization
My Chat App
Alice
Peter
Bob
My Chat App
Bob
Peter
Alice
1. send a chat message to Alice
3. Alice receives the message
Pub-Sub Server
IO Handler
Redis Client
2. add a message, publish to "messages" topic
redis
Hash
members
SortedSet
messages
Stage 2: Message Handling

hash-key: members	
sub-key1: Bob	value1: xxx
sub-key2: Alice	value2: xxx

zset-key: messages	
Bob to Alice: "hello"	11:01:05
Alice to Bob: "hi there"	11:02:07

🔹 Stage 1: Connection Initialization

Steps 1 and 2: Bob opens the chat application. A web socket is established between the client and the server.

Steps 3 and 4: The pub-sub server establishes several connections to Redis. One connection is used to update the Redis data models and publish messages to a topic. Other connections are used to subscribe and listen to updates for topics.

Steps 5 and 6: Bob's client application requires the chat member list and the historical message list. The information is retrieved from Redis and sent to the client application.

Steps 7 and 8: Since Bob is a new member joining the chat application, a message is published to the "member_add" topic, and as a result, other participants of the chat application can see Bob.

🔹 Stage 2: Message Handling

Step 1: Bob sends a message to Alice in the chat application.

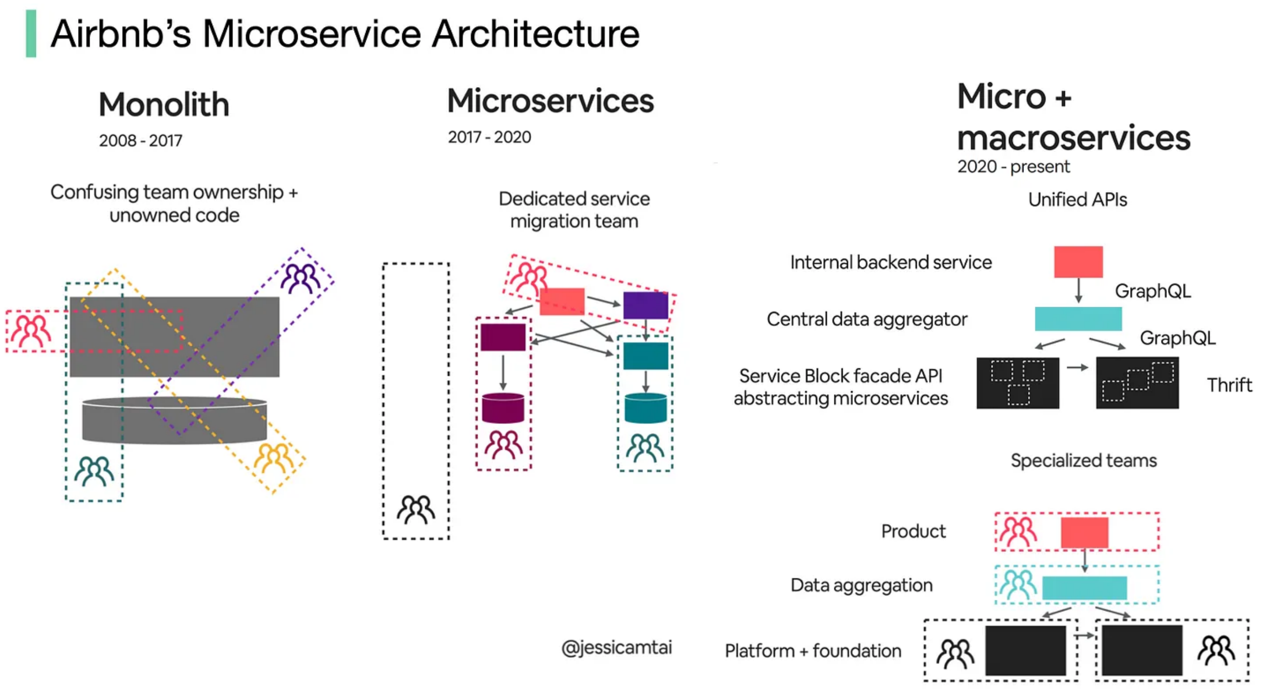
Step 2: The new chat message is added to Redis SortedSet by calling 'zadd.' The chat messages are sorted based on arrival time. The pub-sub server then publishes the chat message to the "messages" topic so subscribers can pick it up.

Step 3: Alice's client application receives the chat message from Bob.

👉 Over to you: What backend stack is commonly used to build a large-scale chat application?

Evolution of Airbnb's microservice architecture

Airbnb's microservice architecture went through 3 main stages. This post is based on the tech talk by Jessica Tai. See the reference link at the end of the thread.



Monolith (2008 - 2017)

Airbnb began as a simple marketplace for hosts and guests. This is built in a Ruby on Rails application - the monolith.

What's the challenge?

- Confusing team ownership + unowned code

- Slow deployment

Microservices (2017 - 2020)

Microservice aims to solve those challenges. In the microservice architecture, key services include:

- Data fetching service

- Business logic data service

- Write workflow service

- UI aggregation service

- Each service had one owning team

What's the challenge?

Hundreds of services and dependencies were difficult for humans to manage.

Micro + macroservices (2020 - present)

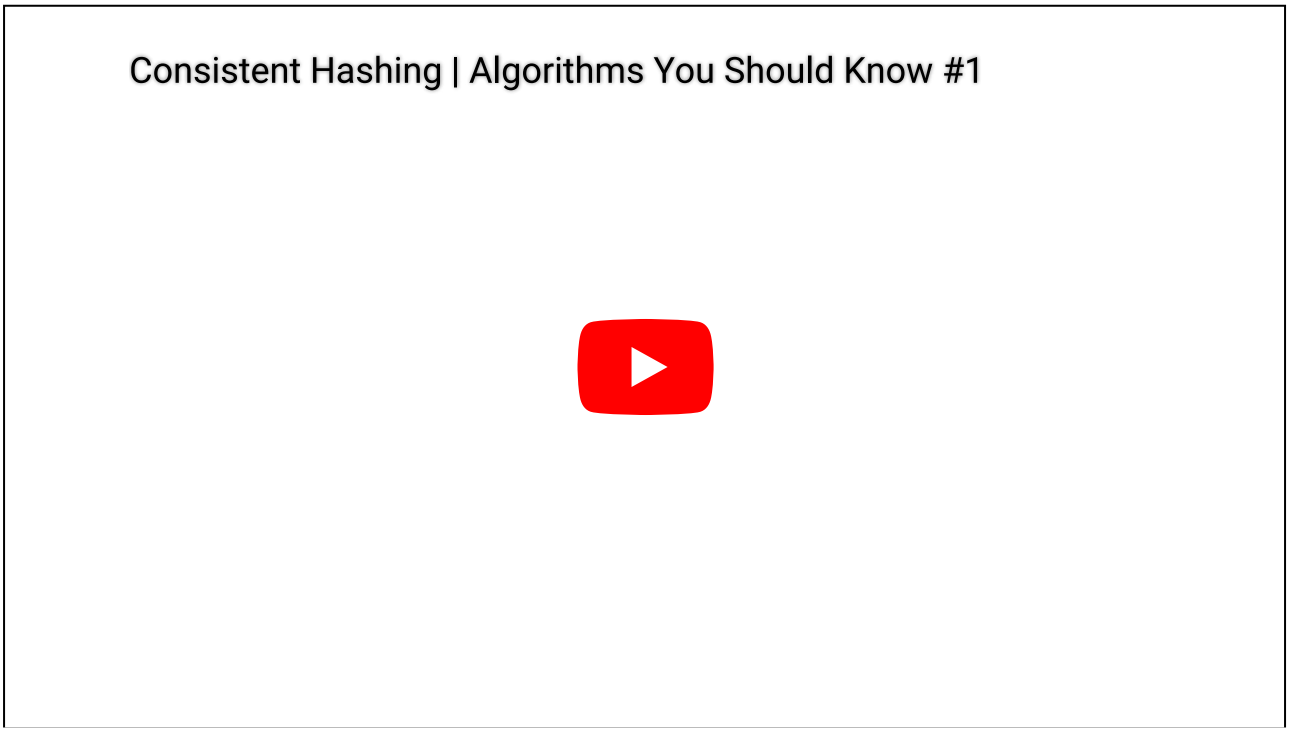
This is what Airbnb is working on now. The micro and macroservice hybrid model focuses on the unification of APIs.

Over to you - why do you think both Airbnb and Netflix use GraphQL?

Reference: [The Human Side of Airbnb's Microservice Architecture]

Consistent Hashing

Algorithms you should know for System Design | Algorithm 1

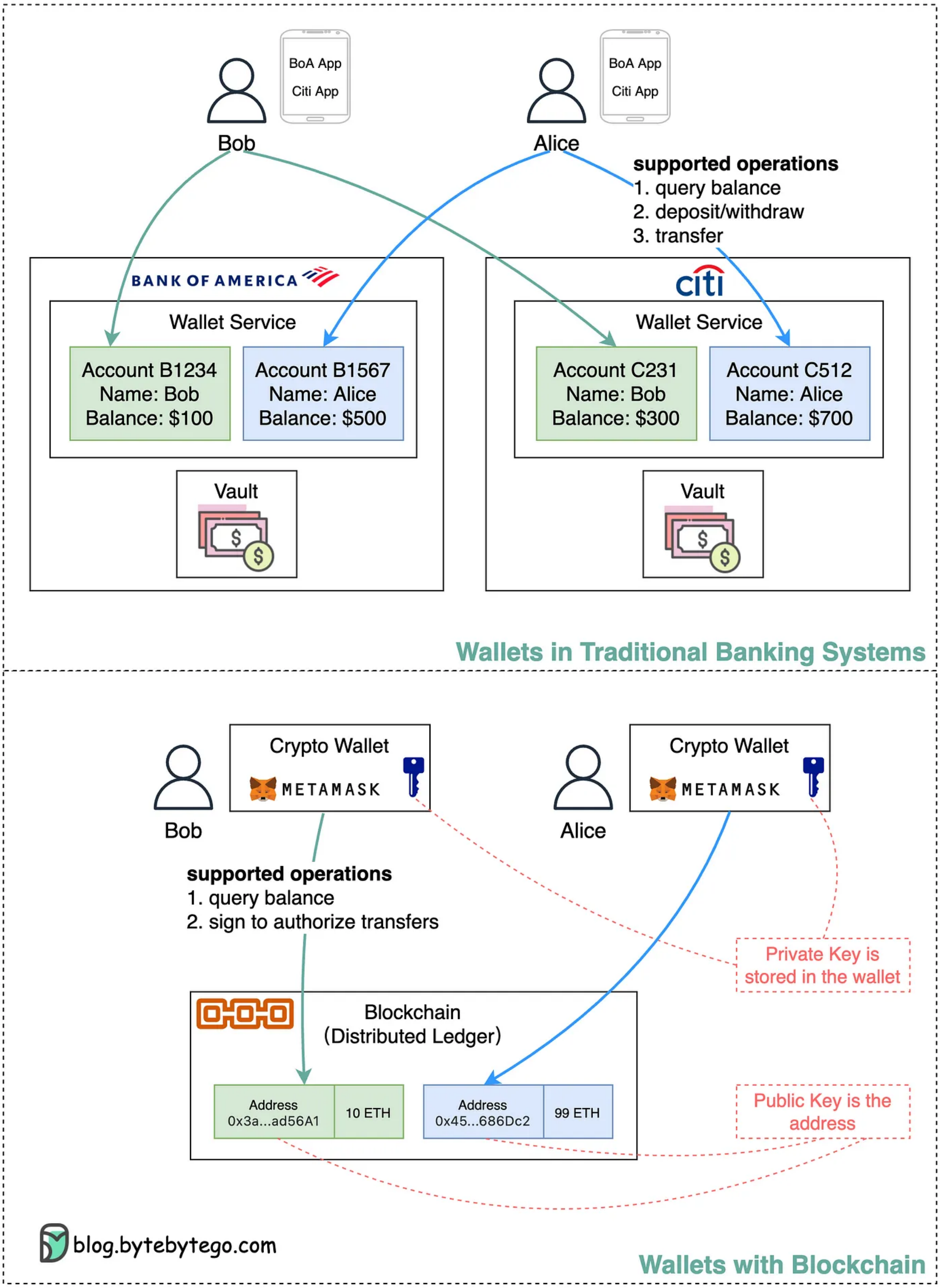


Digital wallet in traditional banks vs wallet in blockchain

How does blockchain change the design of digital wallets? Why do VISA and PayPal invest in blockchains?

The diagram below explains the differences.

How do Blockchains Change Digital Wallets



In banking systems

🔹 Deposit process: Bob goes to Bank of America (BoA) to open an account and deposit $100. A new account B1234 is created in the wallet system for Bob. The cash goes to the bank's vault and Bob's wallet now has $100. If Bob wants to use the banking services of Citibank (Citi,) he needs to go through the same process all over again.

🔹 Transfer process: Bob opens BoA's App and transfers $50 to Alice's account at Citi. The amount is deducted from Bob's account B1234 and credited to Alice's account C512. The actual movement of cash doesn't happen instantly. It happens after BoA and Citi settle all transactions at end-of-day.

🔹 Withdrawal process: Bob withdraws his remaining $50 from account B1234. The amount is deducted from B1234, and Bob gets the cash.

With Blockchains

🔹 Deposit & Withdraw: Blockchains support cryptocurrencies, with no cash involved. Bob needs to generate an address as the transfer recipient and store the private key in a crypto wallet like Metamask. Then Bob can receive cryptocurrencies.

🔹 Transfer: Bob opens Metamask and enters Alice's address, and sends it 2 ETHs. Then Bob signs the transaction to authorize the transfer with the private key. When this transaction is confirmed on blockchains, Bob's address has 8 ETHs and Alice's address has 101 ETHs.

👉 Can you spot the differences?

Blockchain is distributed ledger. It provides a unified interface to handle the common operations we perform on wallets. Instead of opening multiple accounts with different banks, we just need to open a single account on blockchains, which is the address.

All transfers are confirmed on blockchains in pseudo real-time, saving us from waiting until end-of-day reconciliations.

With blockchains, we can merge wallet services from different banks into one global service.

Thanks for making it this far! 🤗

If you want to learn more about System Design, check out our books:

Paperback edition: https://geni.us/XxCd

Digital edition: https://bit.ly/3lg41jK

94 Likes

5 Comments

Write a comment...

Guillaume Writes Arranged ROMs (Rational Opinion... Aug 21, 2022

I can imagine Airbnb and Netflix use GraphQL so they can have a business layer abstracting micro-services. If there are 10 services involved to respond to a business domain query, sending the request to a GraphQL will hide the complexity of gathering all information from the micro-services. This will also help shield services when their dependencies migrate to macro-services.

LIKE REPLY SHARE ...

gaurav Aug 7, 2022

so are we going to have different topic for each subscriber , or is it related to the group (like messages, message_history,)

LIKE REPLY SHARE ...

3 more comments...