

EP 42: Designing a chat application



ALEX XU

JAN 21, 2023



242



5



1

Share



This week's system design refresher:

- CI/CD in 5 minutes (Youtube video)
- Visa dispute and chargeback
- Deployment strategy
- Designing a chat application

Thanks for reading ByteByteGo Newsletter!
Subscribe for free to receive new posts and
support my work.

Building a CSV importer? (Sponsored)

The image shows the OneSchema logo on the left, which consists of a stylized 'S' icon followed by the text 'one schema'. To the right of the logo is the text 'The embeddable CSV Importer for developers'. Below this text is a blue button with the text 'Learn More'. On the right side of the image is a screenshot of the OneSchema web interface. The interface has a dark blue header with the OneSchema logo. Below the header, there are three tabs: 'UNLOADED COLUMNS', 'SAMPLE DATA', and 'TEMPLATE COLUMNS'. The 'SAMPLE DATA' tab is active, showing a table with columns: First Name, Last Name, Email, Phone number, and Account. The table contains several rows of data, some of which are highlighted in red. A modal window titled 'Review & finalize' is open in the center, showing a progress bar and the text 'Cleaning data Automatically fixing 5293 formatting errors'. The modal also shows a table with columns: First Name, Last Name, Email, Phone number, and Account, with a 'Clean' button at the bottom.

OneSchema is the embeddable CSV Importer for developers ([demo](#)).

Product and engineering teams use OneSchema to save months of development time building a CSV importer. With a large library of prebuilt validations and robust SDKs, OneSchema only takes 30 minutes to get running in your app.

OneSchema turns CSV import from a headache into a breeze:

- **Cost-effective:** [70-80% cheaper than building a CSV importer in-house](#)
- **Low effort:** Launch CSV import in hours, not quarters
- **Delightful:** Resolve errors in your import, without leaving your product
- **Converts:** Increase data onboarding completion rates to 90%+

With features like intelligent mapping and data correction, importing clean customer data into your product is easier than ever.

What is CI/CD? How does it help us ship faster? Is it worth the hassle?

CI/CD In 5 Minutes | Is It Worth The Hassle: Crash Course System Design



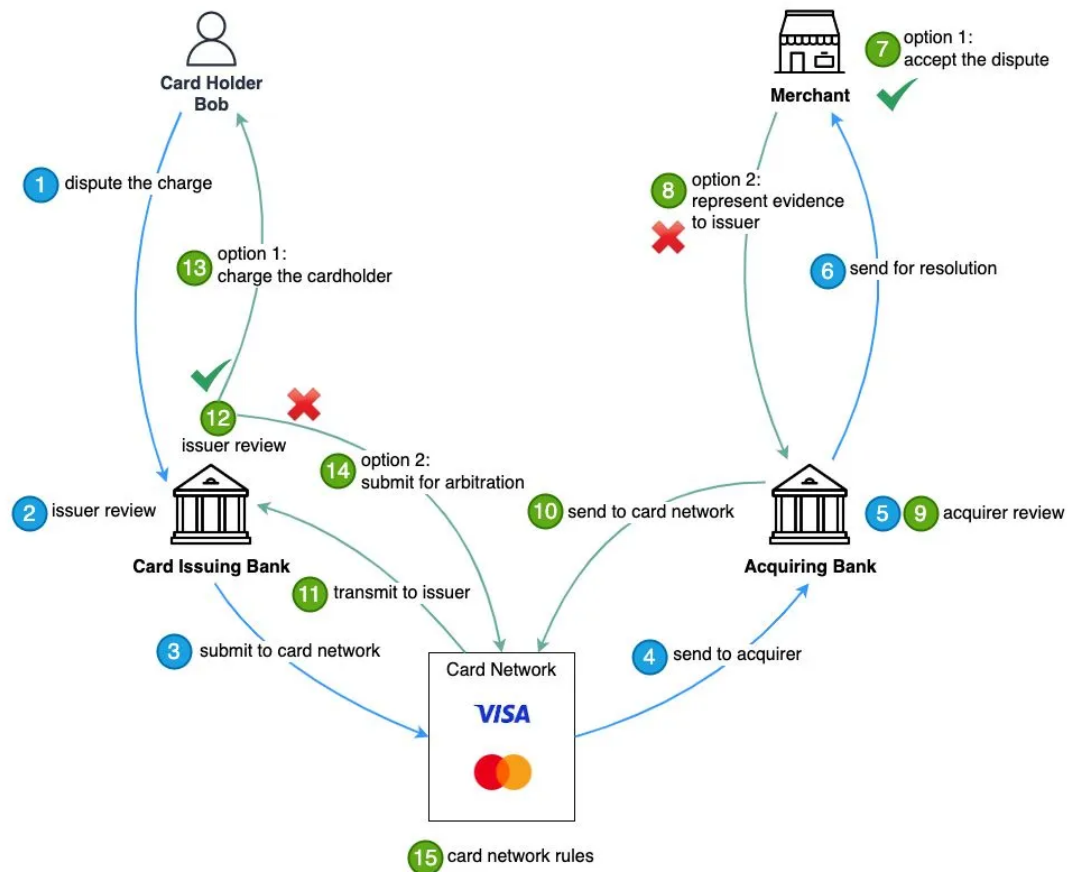
How do Visa disputes and chargeback work?

A dispute happens when a cardholder disagrees with a merchant's charge. A chargeback is a process of reversing the charge. Sometimes, the two terms are used interchangeably.

A dispute is **expensive**: for every dollar in disputed transactions, an additional \$1.50 is spent on fees and expenses.

How do Dispute and Chargeback Work?

blog.bytebytego.com



Source: Javelin Strategy & Research, 2018

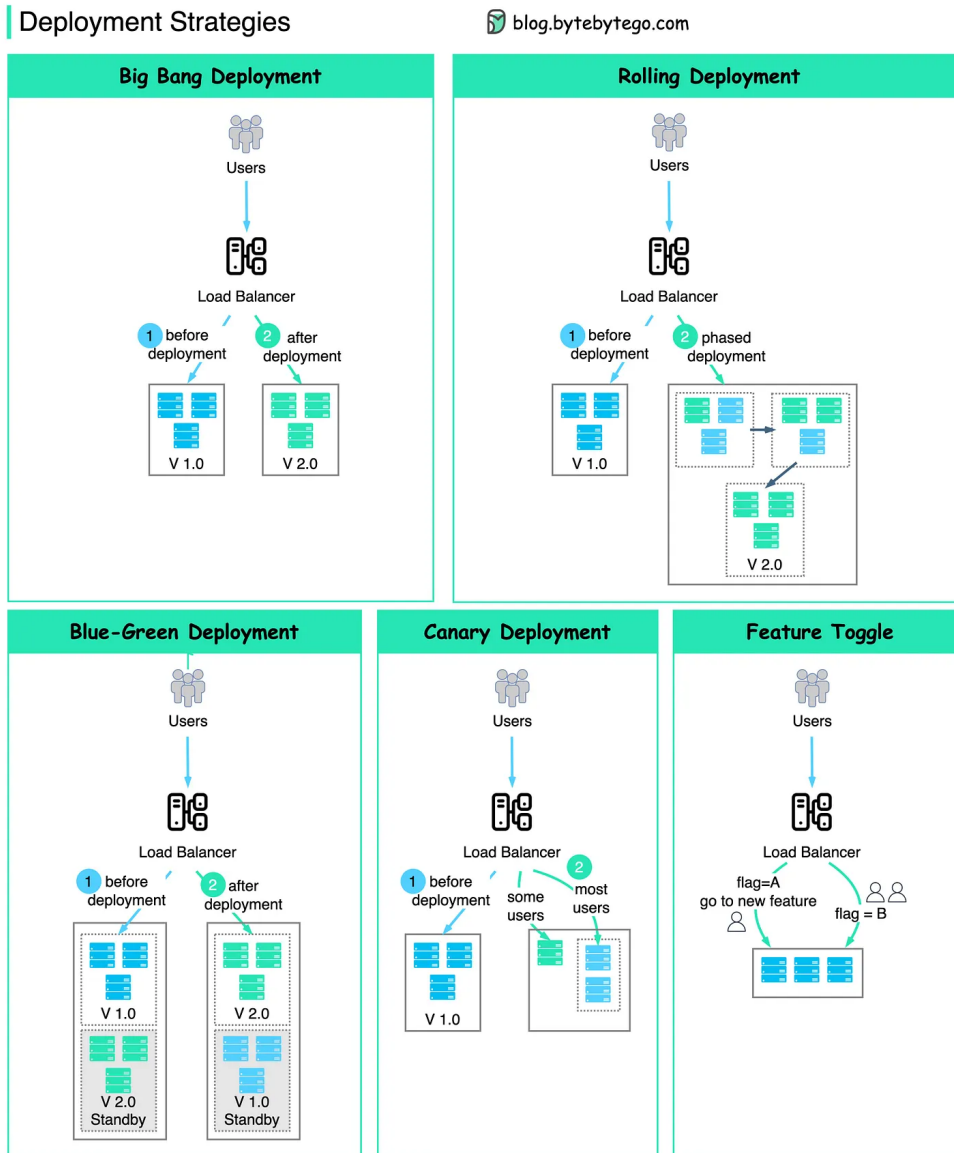
- Steps 1-3: The cardholder, Bob raises a dispute with the card issuer. The issuing bank reviews details. In cases of legitimate disputes, the issuing bank submits a chargeback request to the card network.
- Steps 4-6: The card network sends the dispute to the acquiring bank. After reviewing the details, the acquiring bank might ask the merchant to resolve the issue.
- Steps 7,8: The merchant has two options:
 1. Merchants can accept chargebacks if they appear legitimate.
 2. The merchant can represent to the issuer the document that supports the transaction.

- Steps 9-11: The acquiring bank reviews the evidence and represents the transaction to the card network, which forwards it to the issuer.
- Steps 12-14: The issuer reviews the representment. There are two options:
 1. The issuer charges the transaction back to the cardholder;
 2. The issuer submits the dispute to the card network for arbitration.
- Step 15: The card network rules based on the evidence and assigns the final liability to either the cardholder or the merchant.

👉 Over to you: Dispute is expensive. How can we reduce it and make the process more streamlined?

What is the process for deploying changes to production?

The diagram below shows several common **deployment strategies**.



Big Bang Deployment

Big Bang Deployment is quite straightforward, where we just roll out a new version in one go with service downtime. Preparation is essential for this strategy. We roll back to the previous version if the deployment fails.

- 💡 No downtime ❌
- 💡 Targeted users ❌

Rolling Deployment

Rolling Deployment applies phased deployment compared with big bang deployment. The whole plant is upgraded one by one over a period of time.

- 💡 No downtime ✅
- 💡 Targeted users ❌

Blue-Green Deployment

In blue-green deployment, two environments are deployed in production simultaneously. The QA team performs various tests on the green environment. Once the green environment passes the tests, the load balancer switches users to it.

💡 No downtime ✅

💡 Targeted users ❌

Canary Deployment

With canary deployment, only a small portion of instances are upgraded with the new version, once all the tests pass, a portion of users are routed to canary instances.

💡 No downtime ✅

💡 Targeted users ❌

Feature Toggle

With feature toggle, A small portion of users with a specific flag go through the code of the new feature, while other users go through normal code. This can be used in combination of other strategies: either the new branch of code is upgraded in one go, or only a few instances are upgraded with new code.

💡 No downtime ✅

💡 Targeted users ✅

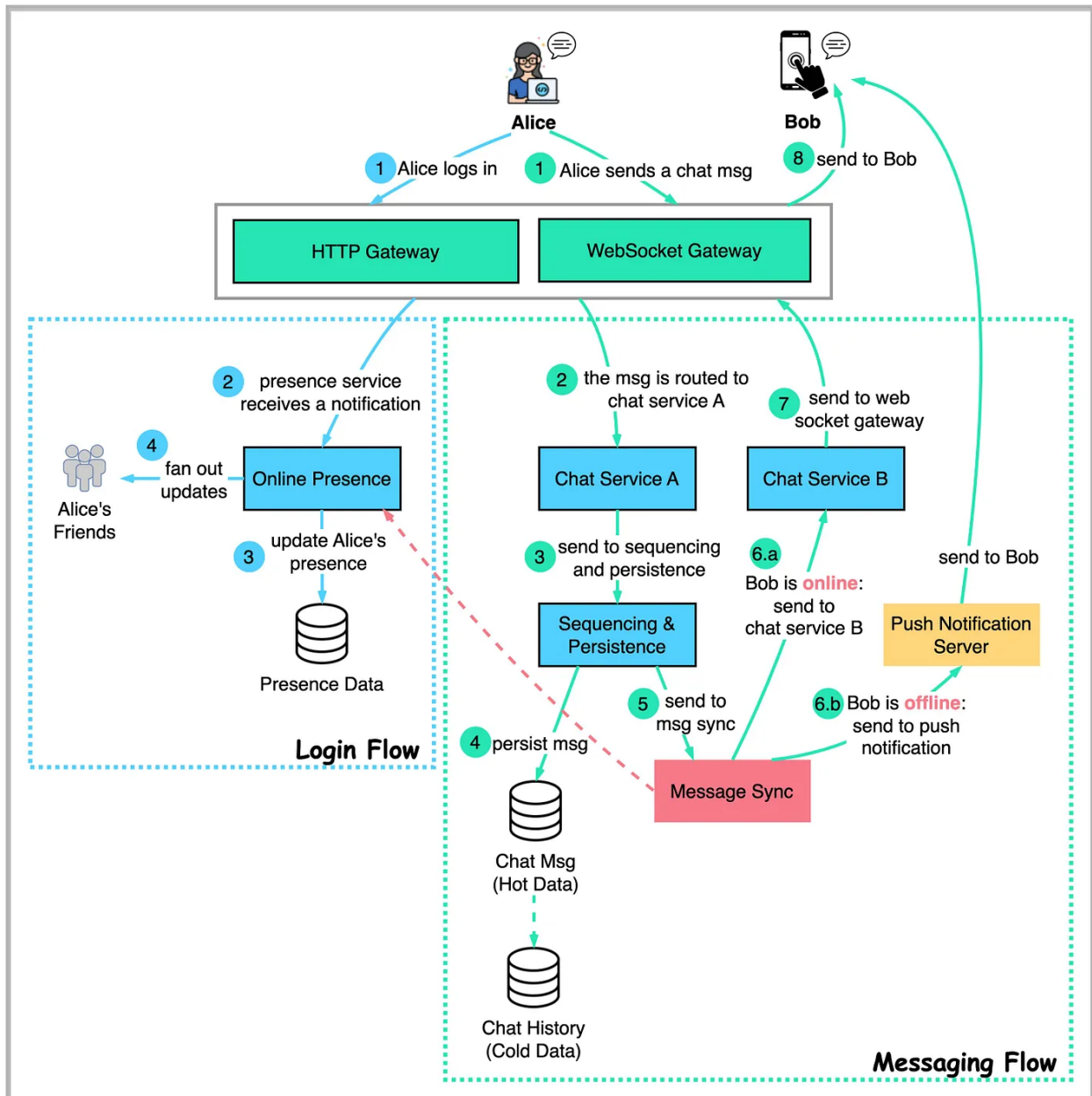
👉 Over to you: Which deployment strategies have you used?

How do we design a chat application like WhatsApp, Facebook Messenger, or Discord?

The diagram below shows a design for a simplified 1-to-1 chat application.

How to Design a Chat Application?

 blog.bytebytego.com



User Login Flow

- Step 1: Alice logs in to the chat application and establishes a web socket connection with the server side.
- Steps 2-4: The presence service receives Alice's notification, updates her presence, and notifies Alice's friends about her presence.

Messaging Flow

- Steps 1-2: Alice sends a chat message to Bob. The chat message is routed to Chat Service A.

- Steps 3-4: The chat message is sent to the sequencing service, which generates a unique ID, and is persisted in the message store.
- Step 5: The chat message is sent to the message sync queue to sync to Bob's chat service.
- Step 6: Before forwarding the messaging, the message sync service checks Bob's presence:
 - a) If Bob is online, the chat message is sent to chat service B.
 - b) If Bob is offline, the message is sent to the push server and pushed to Bob's device.
- Steps 7-8: If Bob is online, the chat message is pushed to Bob via the web socket

👉 Over to you: How to sync the chat messages among Alice's different devices?

Thanks for reading ByteByteGo Newsletter!
Subscribe for free to receive new posts and
support my work.



242 Likes · 1 Restack

5 Comments



Write a comment...



Mangaldeep Jan 21

Hello Alex,

Great information in your newsletter as always. I'm working in the chargeback backend process for large bank and we have following additional steps :-

Step 14 : Issuer bank submits a pre-arbitration to the network which gets passed to the acquirer bank and then the merchant.

Step 15 : Merchant either accepts it or sends a pre-arbitration response back to acquirer bank which is passed to network and then to issuer bank.

Step 16 : Issuer accepts the pre-arbitration response or files an arbitration with the network.

Thank you.

♡ LIKE (6) 💬 REPLY ↗ SHARE ...



Oleksii (Alex) Polivanyy Jan 22

My answer to the chargeback question is: connect the cardholder with the merchant.

✅ If the merchant agrees, the chargeback process starts from the merchant steps, which should cut costs twice;

❌ if the merchant disagrees, then the standard flow.

Of course, it will work if the merchant and the cardholder are inside one platform marketplace.

♡ LIKE (1) 💬 REPLY ↗ SHARE ...

1 reply

3 more comments...