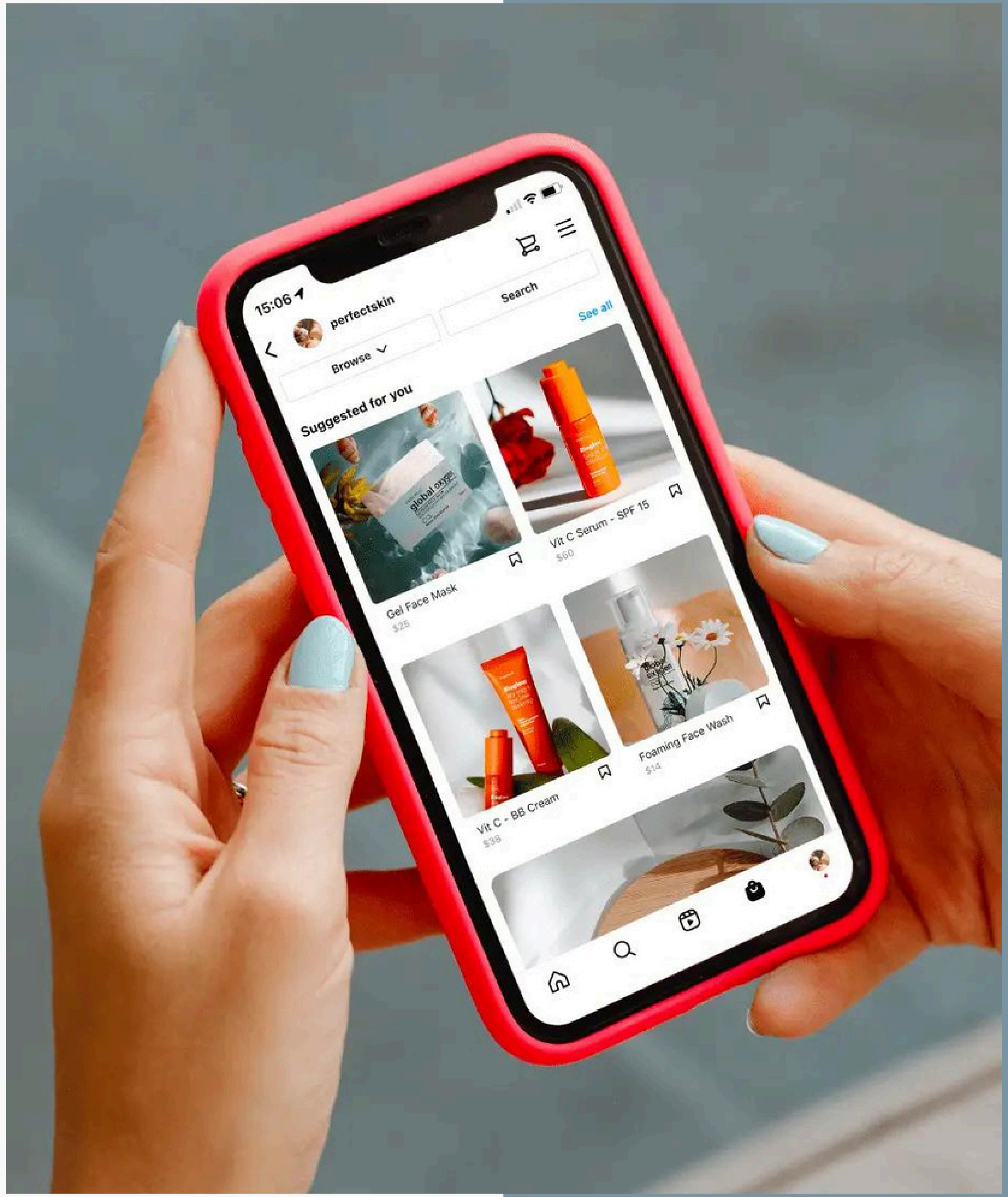
Successful Mobile Apps

by Justine Cua

Native App Case Study

Instagram

a social media app designed for iOS and Android using respective native languages: Swift (iOS) and Kotlin (Android)



Reasons for Instagram's Success

Great Performance

can handle massive amounts of data and high user loads. It is a native app which allows the app to manage millions of active users sharing photos, videos, and stories every second without major performance issues.

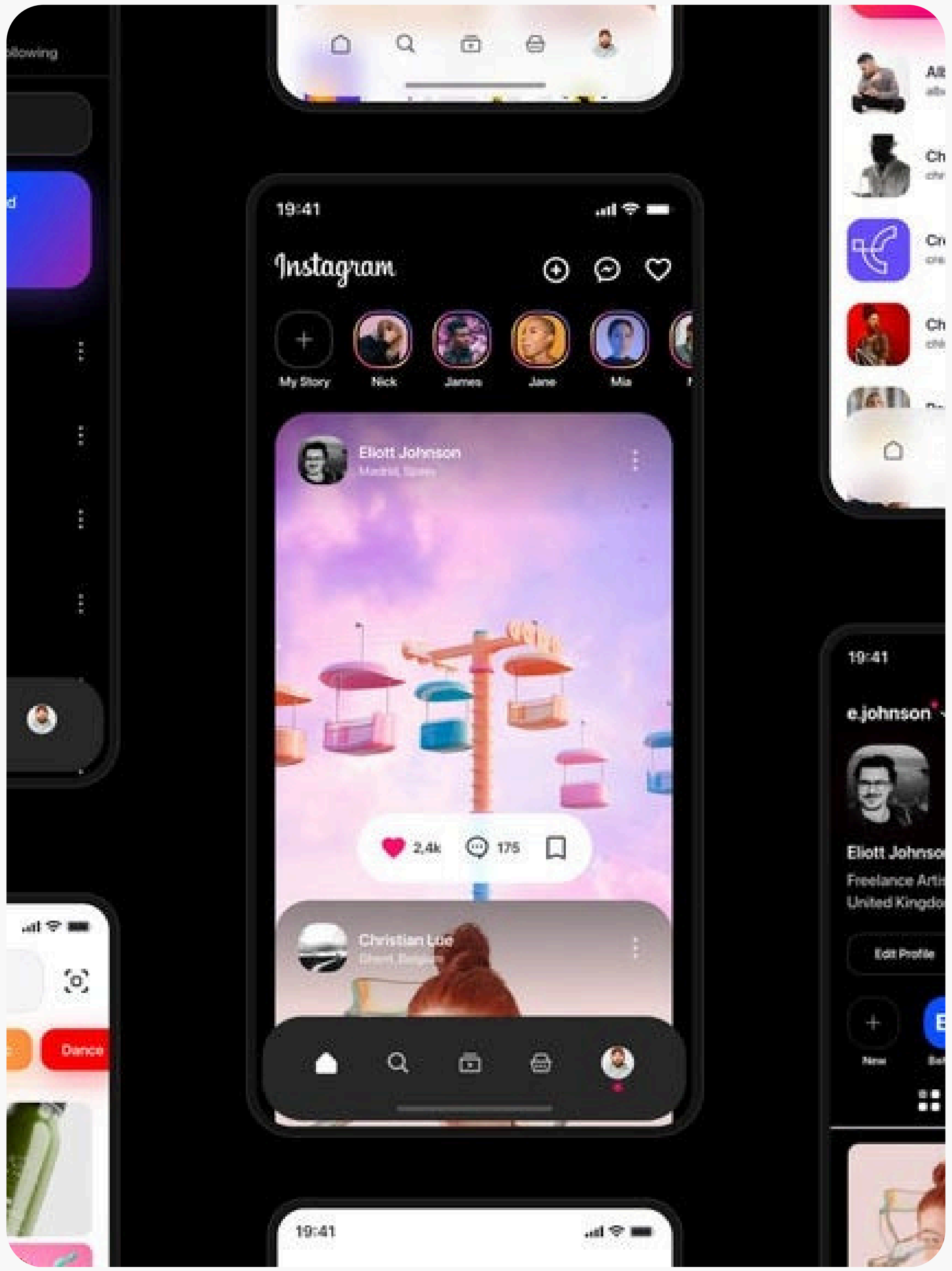
Regular Updates

release updates separately for iOS and Android, taking advantage of each platform's features and addressing bugs quickly.

Use of Phone Features

uses the phone's camera, notifications, and location services. It has AR filters, geolocation tagging, and instant photo sharing. Full integration with the camera makes posting stories or images effortless.

Challenges or Failures



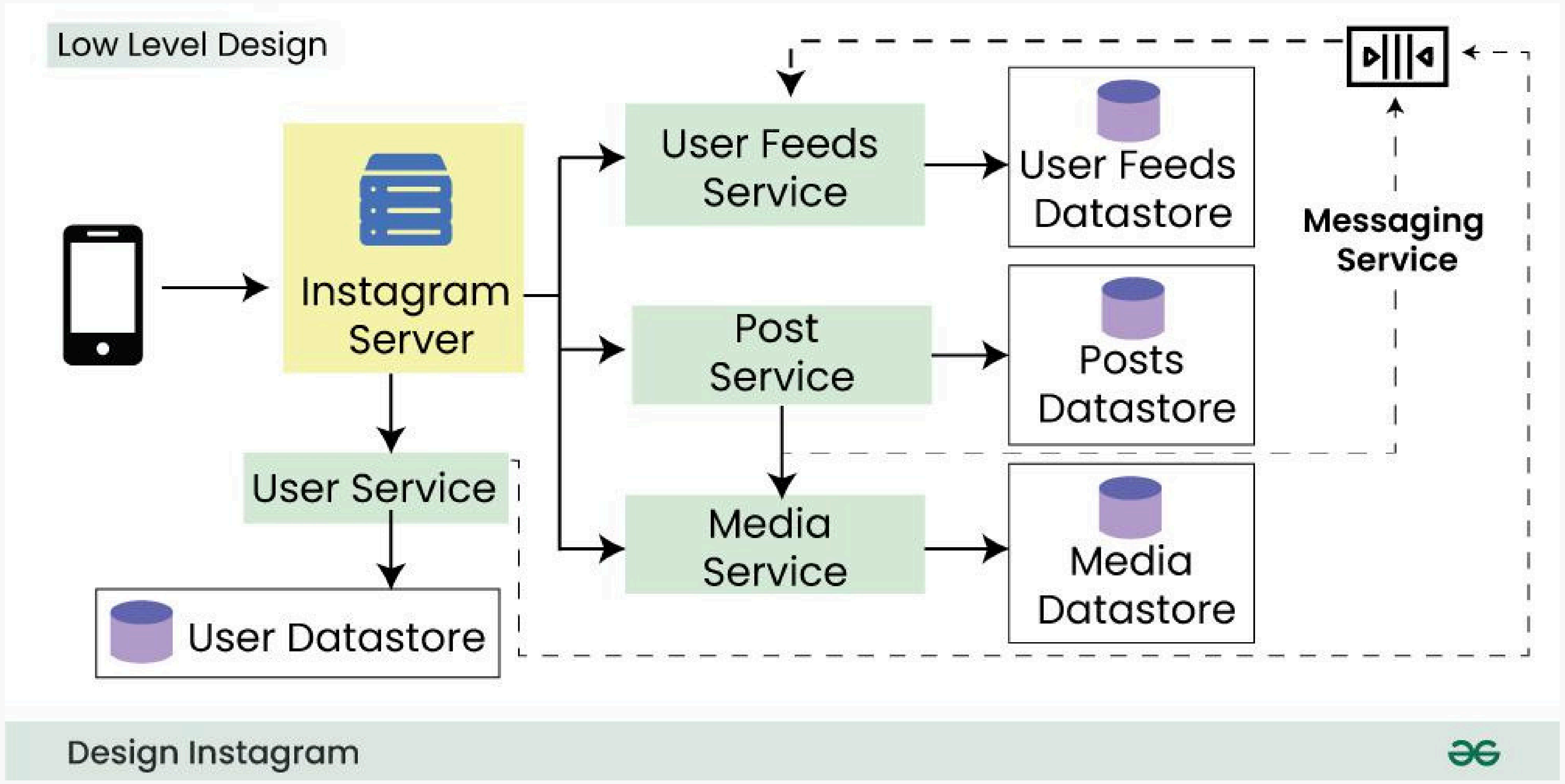
Separation of codebases

- Maintaining two codebases increases development time, costs, and complexity. You need a team to handle updates for each platform, which can slow down the development process.

Larger App Size

- App size can grow fast when dealing for media features like stories and AR filters. It leads to storage concerns for users with limited space on their devices, potentially limiting new user adoption or causing existing users to uninstall the app

Example architecture of Instagram



Web App Case Study

Google Docs

a popular web application that allows users to create, edit, and collaborate on documents in real-time, directly through a web browser.

Reasons for Google Doc's Success

Cross Platform

works on any device with a browser, (Windows, macOS, Linux, iOS, Android). Users don't need to install an app, making it accessible from anywhere, on any device, without worrying about compatibility

Collaboration

enables multiple users to edit a document It has an ability to sync changes in real time without delays is a huge selling point for teams working on different locations

Cloud Integration

Users can store, access, and share their documents in the cloud, ensuring that their work is always backed up and accessible. This eliminates the need for manual saving or transferring files

Challenges or Failures

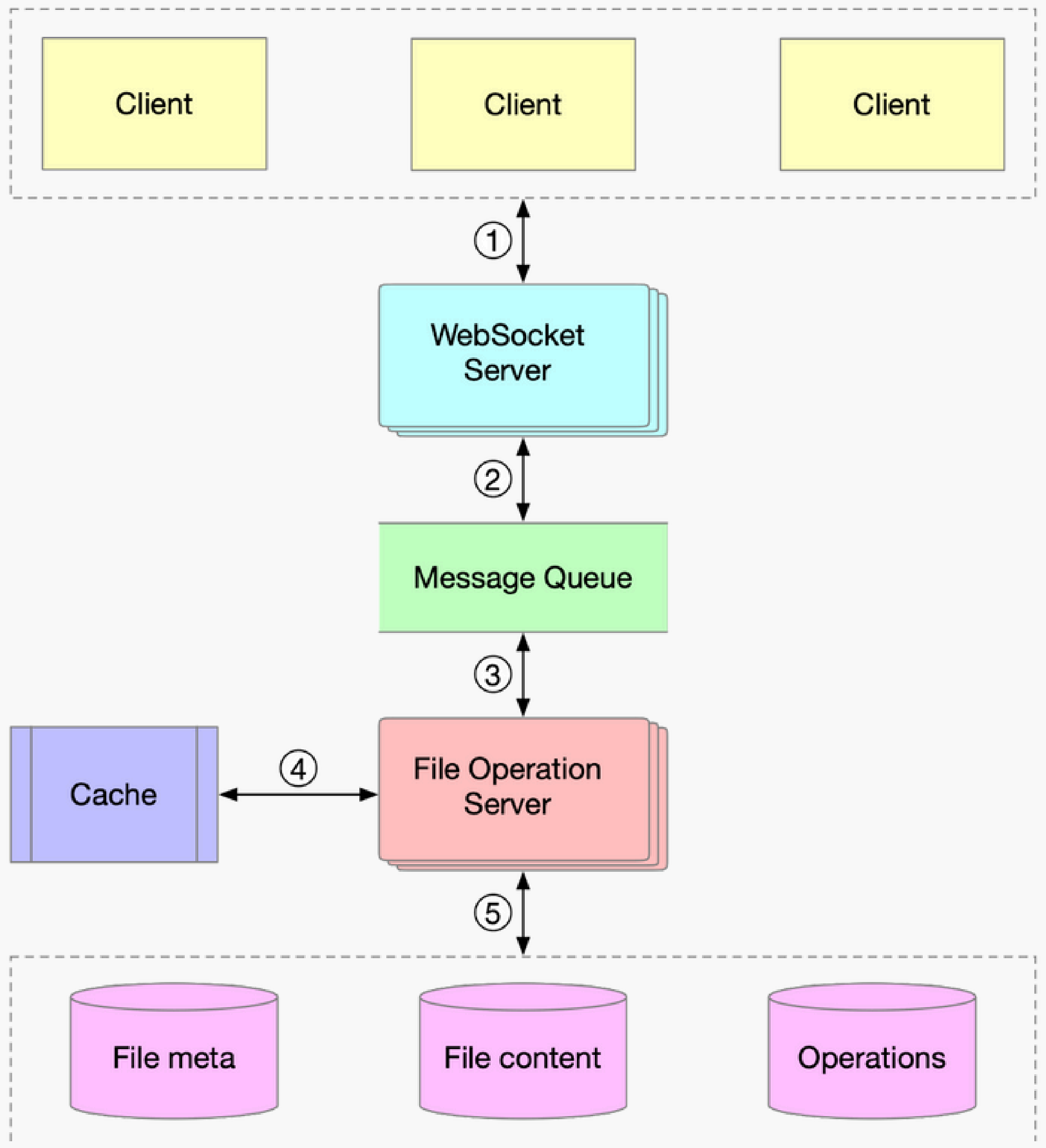
Limited Offline Functionality

- The full feature set of Google Docs isn't available when users are not connected to the internet. Users need to enable offline mode manually, and real-time collaboration are not accessible, which can be frustrating for those without reliable internet access

Browser Dependency

- Users may experience inconsistent performance depending on the browser they're using. Web based apps are affected by issues like slow loading times if the internet connection is poor which can affect productivity for users.

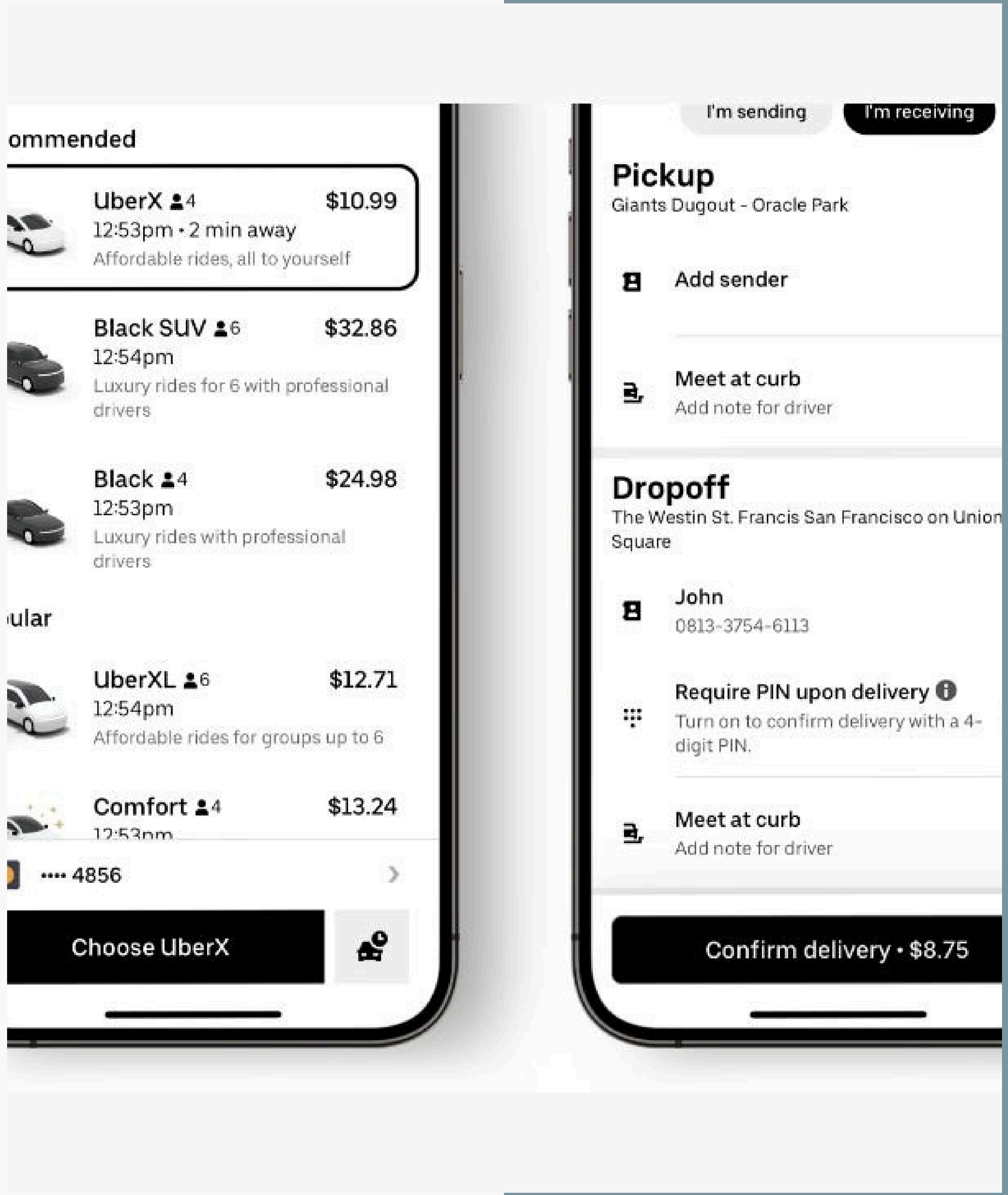
Example architecture of Google Docs



Hybrid Case Study

Uber

a ride-hailing service that operates through a hybrid mobile app. This means it combines both native and web technologies to deliver the app experience across platforms. Uber uses React Native to write a single codebase that works on both iOS and Android


ommended

UberX 4 $10.99
12:53pm • 2 min away
Affordable rides, all to yourself

Black SUV 6 $32.86
12:54pm
Luxury rides for 6 with professional drivers

Black 4 $24.98
12:53pm
Luxury rides with professional drivers

ular

UberXL 6 $12.71
12:54pm
Affordable rides for groups up to 6

Comfort 4 $13.24
12:53pm

•••• 4856

Choose UberX

I'm sending I'm receiving

Pickup
Giants Dugout - Oracle Park

Add sender

Meet at curb
Add note for driver

Dropoff
The Westin St. Francis San Francisco on Union Square

John
0813-3754-6113

Require PIN upon delivery
Turn on to confirm delivery with a 4-digit PIN.

Meet at curb
Add note for driver

Confirm delivery • $8.75

Reasons for Uber's Success

Native Features

integrates native APIs for accessing critical mobile device features like GPS tracking, push notifications, and real-time updates

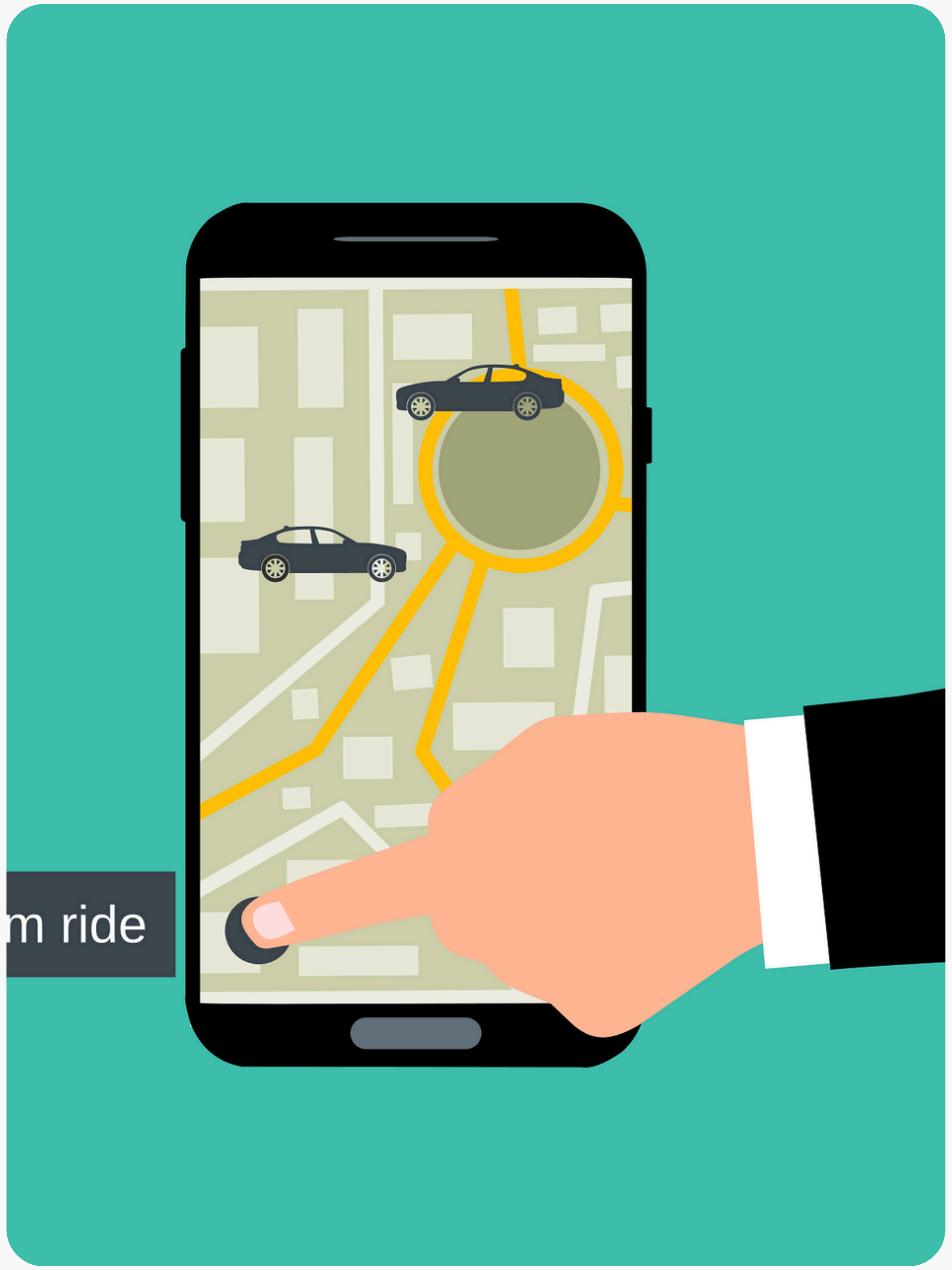
Fast Iteration and Scaling

React Native's single codebase enables Uber to push updates to both iOS and Android users more rapidly. This is essential for Uber's expansion into new markets and for new features

Payment Integration

secure payment integration (credit card, PayPal) is made possible by native features. This minimizes friction for users at the end of their rides

Challenges or Failures



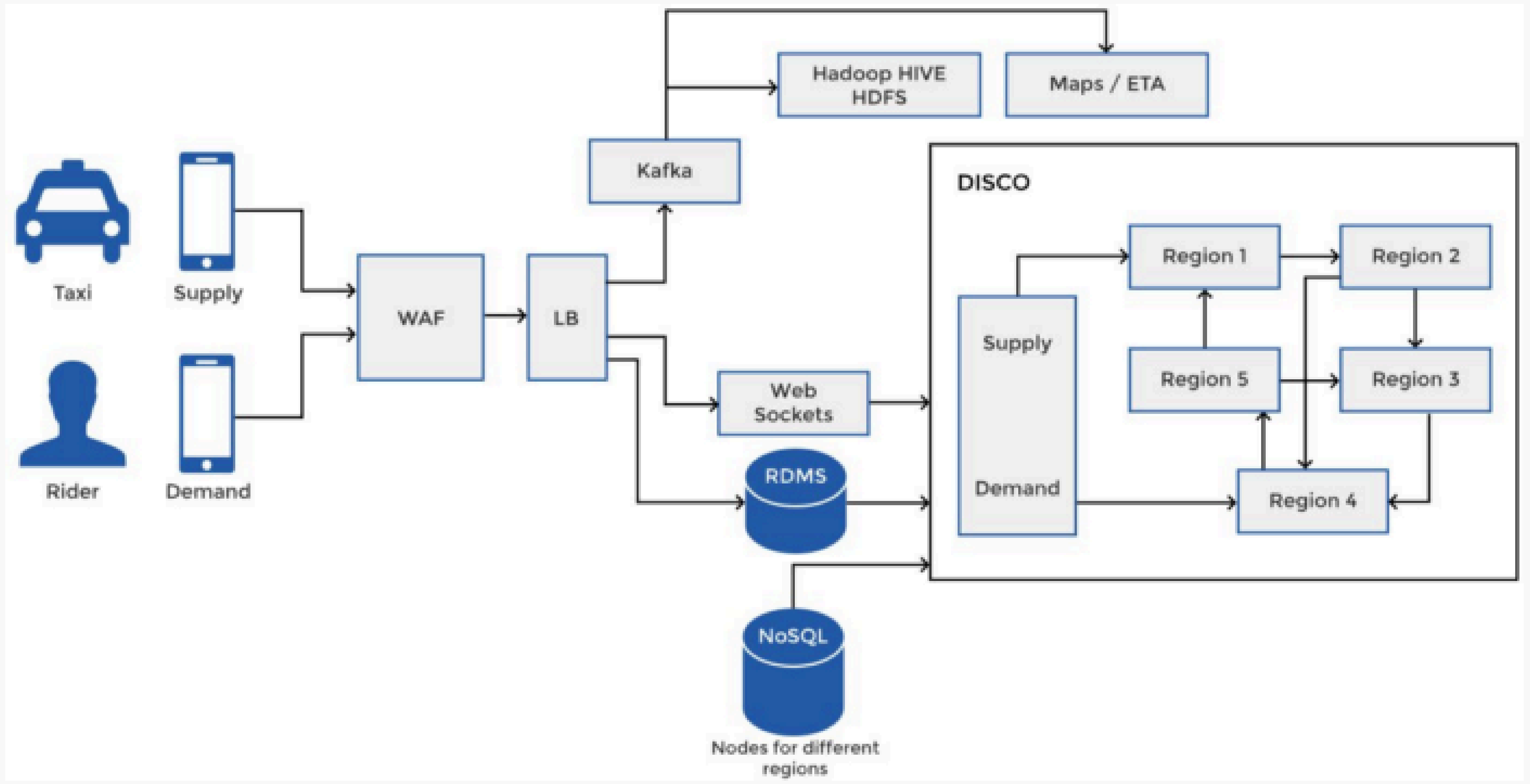
Complexity

- a hybrid app relies on native code for critical features like GPS and payments can create complexity in development. Developers need expertise in both React Native and platform-specific languages like Swift and Kotlin when integrating with new versions of iOS and Android

Dependency on Libraries

- If a library has a bug or becomes outdated, it can impact the stability and performance of Uber's app. This introduces potential risks that native apps might avoid by building custom solutions to the platform.

Example architecture of Uber



Thank you