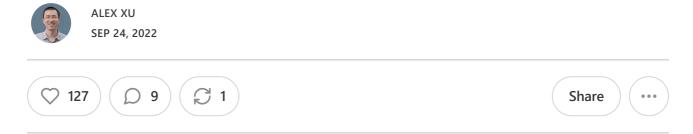
EP25: How Apple/Google Pay handle card info. Also...



In this newsletter, we'll cover the following topics:

- How Apple Pay and Google Pay handle card info
- JSON visualization tool
- Unified Payments Interface (UPI) in India
- How to pick the right database
- How Google/Apple maps blur license plates and human faces on Street View

Postman the API platform for building and using APIs (sponsored)



Postman simplifies each step of the API lifecycle and streamlines collaboration so you can create better APIs—faster. The platform is cloud-native and includes the comprehensive suite of features enterprises are looking for, including SSO, audit, platform security, and much more.

API repository

Postman can store and manage API specifications, documentation, workflow recipes, test cases and results, metrics, and everything else related to APIs.

Workspaces

Postman workspaces are collaborative places where teams gather and solve problems. Every person in a workspace sees the same API tools, collections, and environments, which are updated in real-time.

API lifecycle

The Postman platform includes a comprehensive set of tools that help accelerate the API lifecycle—from design, testing, documentation, and mocking to the sharing and discoverability of your APIs.

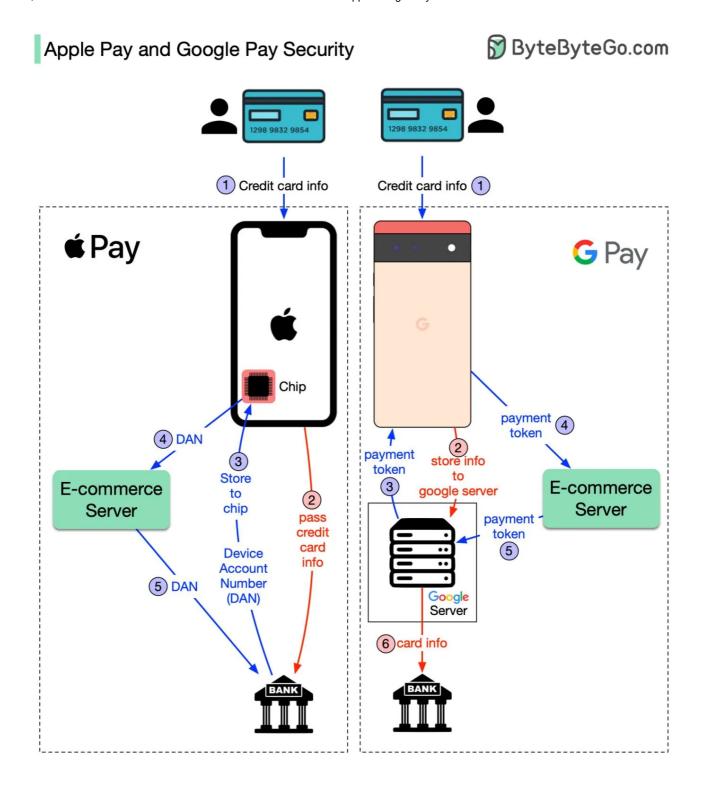
Governance

Postman's full-lifecycle approach to governance lets adopters shift left in their development practices, producing better-quality APIs and fostering collaboration between developer and API design teams.

How do Apple Pay and Google Pay handle sensitive card info?

The diagram below shows the differences. Both approaches are very secure, but the implementations are different. To understand the difference, we break down the process into two flows.

- 1 Registering your credit card flow
- 2 Basic payment flow



1. The registration flow is represented by steps 1~3 for both cases. The difference is: **Apple Pay**: Apple doesn't store any card info. It passes the card info to the bank. Bank returns a token called DAN (device account number) to the iPhone. iPhone then stores DAN into a special hardware chip.

Google Pay: When you register the credit card with Google Pay, the card info is stored in the Google server. Google returns a payment token to the phone.

2. When you click the "Pay" button on your phone, the basic payment flow starts. Here are the differences:

Apple Pay: For iPhone, the e-commerce server passes the DAN to the bank.

Google Pay: In the Google Pay case, the e-commerce server passes the payment token to the Google server. Google server looks up the credit card info and passes it to the bank.

In the diagram, the red arrow means the credit card info is available on the public network, although it is encrypted.

References:

- [1] Apple Pay security and privacy overview
- [2] Google Pay for Payments
- [3] Apple Pay vs. Google Pay: How They Work

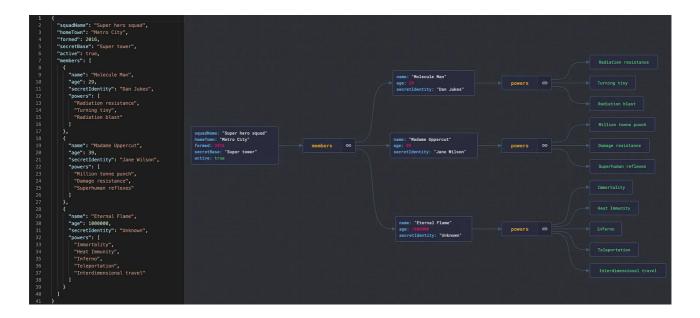
JSON visualization tool

If you use JSON files, you'll probably like this tool.

Nested JSON files are hard to read.

JsonCrack generates graph diagrams from JSON files and makes them easy to read.

Additionally, the generated diagrams can be downloaded as images.



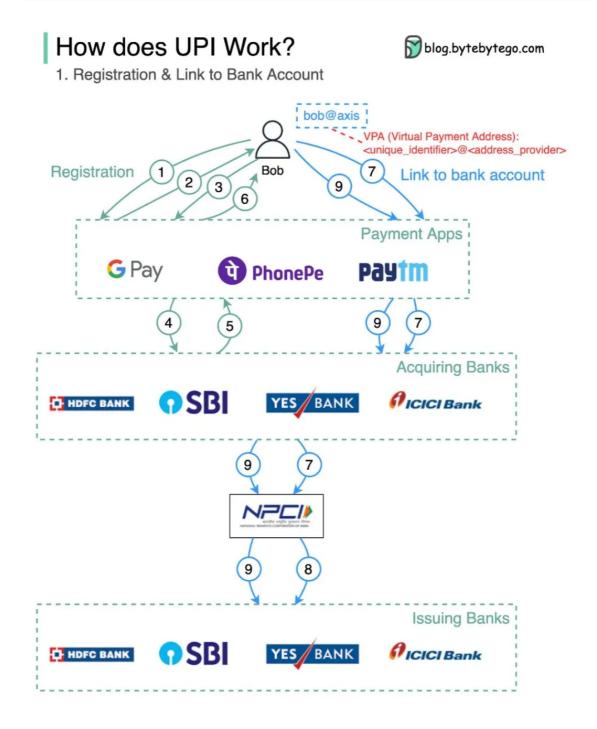
Unified Payments Interface (UPI) in India

What's UPI? UPI is an instant real-time payment system developed by the National Payments Corporation of India.

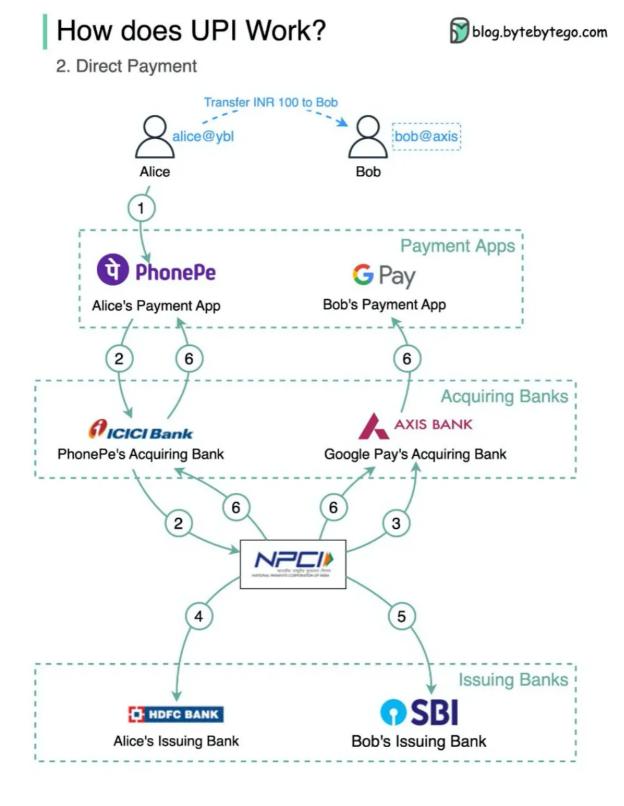
It accounts for 60% of digital retail transactions in India today.

UPI = payment markup language + standard for interoperable payments

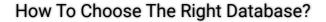
Registration & Link to Bank Account



Direct payment



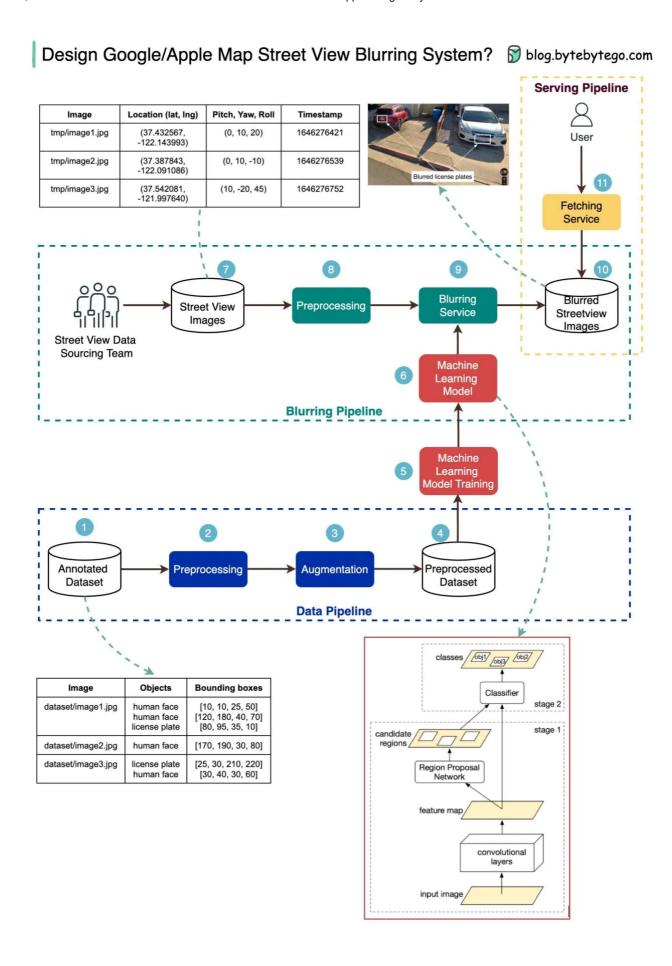
How to pick the right database?





How do Google/Apple maps blur license plates and human faces on Street View

The diagram below presents a possible solution that might work in an interview setting.



The high-level architecture is broken down into three stages:

- 1. Data pipeline prepare the training data set
- 2. Blurring pipeline extract and classify objects and blur relevant objects, for example, license plates and faces.
- 3. Serving pipeline serve blurred street view images to users.

Data pipeline

Step 1: We get the annotated dataset for training. The objects are marked in bounding boxes.

Steps 2-4: The dataset goes through preprocessing and augmentation to be normalized and scaled.

Steps 5-6: The annotated dataset is then used to train the machine learning model, which is a 2-stage network.

Blurring pipeline

Steps 7-10: The street view images go through preprocessing, and object boundaries are detected in the images. Then sensitive objects are blurred, and the images are stored in an object store.

Serving pipeline

Step 11: The blurred images can now be retrieved by users.

Thanks for making it this far!

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





Paperback edition

Digital edition



127 Likes · 1 Restack

9 Comments



Write a comment...



Mani Sep 24, 2022 · edited Sep 24, 2022 ♥ Liked by Alex Xu

Great stuff yet again. I would have loved a more detailed explanation on UPI.

 \bigcirc LIKE (7) \bigcirc REPLY $\mathring{\bot}$ SHARE





Poorna Sep 30, 2022 ♥ Liked by Alex Xu

Great read. Would love to read detailed explanation on UPI.

C LIKE (2) REPLY T SHARE	• • •
7 more comments	

© 2023 ByteByteGo \cdot <u>Privacy</u> \cdot <u>Terms</u> \cdot <u>Collection notice</u> <u>Substack</u> is the home for great writing