AMERICAN EXPRESS
CodeStreet'20

Realtime Payments

Credit-Card Fraud Detection System

A One stop **Real Time Payments Solution** For all businesses

Problem Statement (Idea) :
Developing a Robust and at power Credit-card Fraud Detection System by combining industry leading ML Algorithms with advanced blockchain based verification methods. Along with a multilayered fraud detection system which helps the business to meet consumer demands for speed and convenience.

Team Name : Power Puff Boys

Team Members :
Mohan Arora
Sidhant Goyal
Harshil Sharma
Prateek Bansal

➡ Problem

AMERICAN EXPRESS
CodeStreet'20

- **Maintaining Speed, Accuracy and Convenience:**
  Detecting fraudulent transactions should take place at real time and not in future after the payment has been made, but it should not interfere with customers ease of use

- **Loss of Business:**
  Financial institutions loose a lot of funds every year in order to payback and track fraudulent transactions

- **Highly Skewed Data:**
  The major problem in real world data is that the data is highly skewed in nature and thus difficult to work with.

- **Eliminating Redundant verifications:**
  Some steps of credit card fraud detection scheme are redundant and can be simplified with minor tweaks in the databases

AMERICAN EXPRESS
CodeStreet'20

➞ Types of Frauds Handled

- Realtime payments make it very attractive for fraudsters to attack accounts. With increased customer convenience, there seems to be a cost of security, particular types of bank accounts that offer customers the ability to send money instantly, especially if they allow for relatively large value payments, are likely to be a target for fraudsters.

1. Card-not-present (CNP) fraud
2. Skimming fraud
3. Lost Card Fraud
4. Card never arrived Fraud
5. Identity Fraud
6. Account Takeover fraud
7. Social Engineering Techniques
8. Phishing fraud

AMERICAN EXPRESS
CodeStreet'20

➡ Flowchart

Step1: Pre-processing

Includes a combination of two novel technique called SMOTE Over-sampling + Under-sampling, along with labelling data

Step2: ML Analysis

Using a combination of different industry leading ML models combined into an Ensemble Model

Step3: Prediction

Creating a robust and self-learning ML system which learns from past predictions and makes changes through a layered structure

Step4: Depiction

Showcasing the data to the end user in a constructive way to bring on a comprehensive result

AMERICAN EXPRESS
CodeStreet'20

➡ Our Approach

- **Pre-processing**
  - SMOTE Over-sampling + Under-sampling
  - PCA

- **ML Models Used:**
  - K-means
  - AutoEncoder
  - Xg-Boost
  - RandomForest Classifier
  - ANN

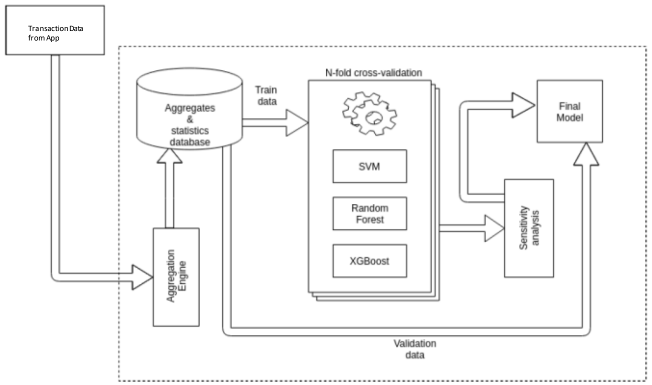


Fig1: Proposed Architecture

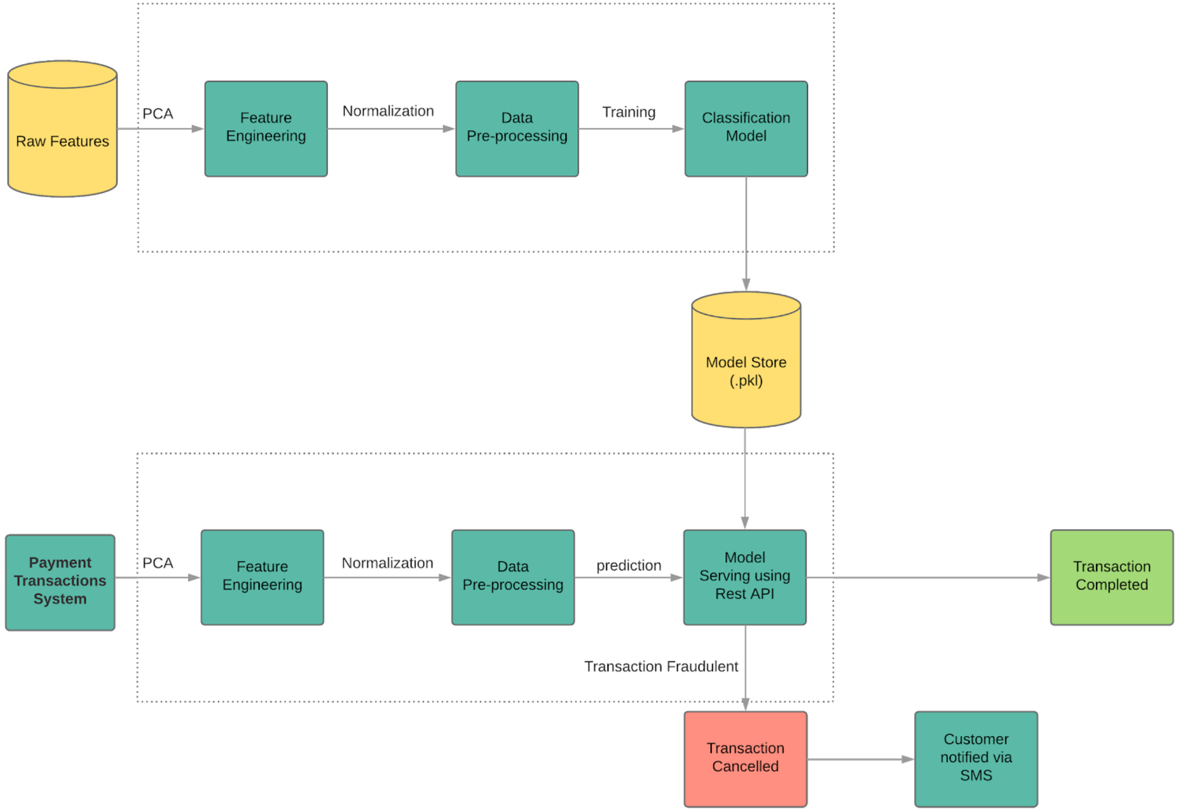
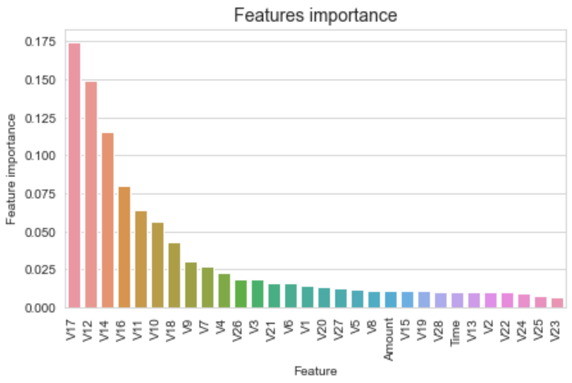


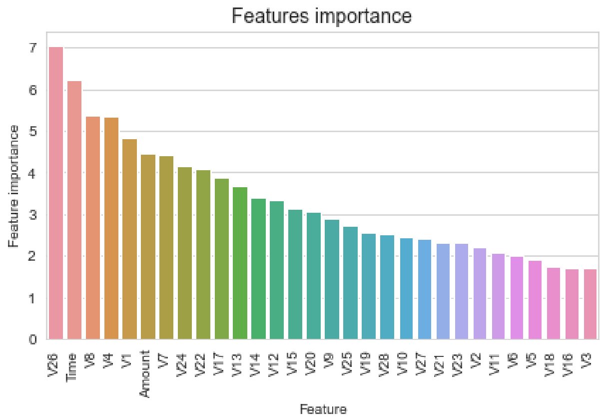
Fig2: Process Visualisation

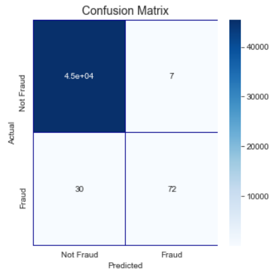
→ Results

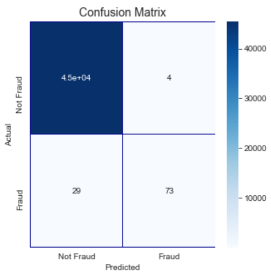
Model	Accuracy
K-means	72.70%
XG-Boost	98.83%
CatBoostClassifier	85%
AdaBoostClassifier	88.30%
ANN	94%
Autoencoders	96.40%
Random Forest Classifier	94.70%
SVM	74.00%
Baysian Networks	93.30%
Ensemble Model	98.94%



The most important features are **V17**, **V12**, **V14**, **V10**, **V11**, **V16**.







Area under curve

```
In [29]: roc_auc_score(valid_df[target].values, preds)
Out[29]: 0.8528641975628091
```

The **ROC-AUC** score obtained with **RandomForrestClassifier** is **0.85**.

Area under curve

```
In [41]: roc_auc_score(valid_df[target].values, preds)
Out[41]: 0.8577991493075996
```

The ROC-AUC score obtained with CatBoostClassifier is **0.86**.

➡ Project Architecture:

- Using 4 different models to increase User experience by minimizing the delay
- Unique thresholding technique in which if weighted probabilities from the 3 preliminary models is less than .72 then only the 4 model is used

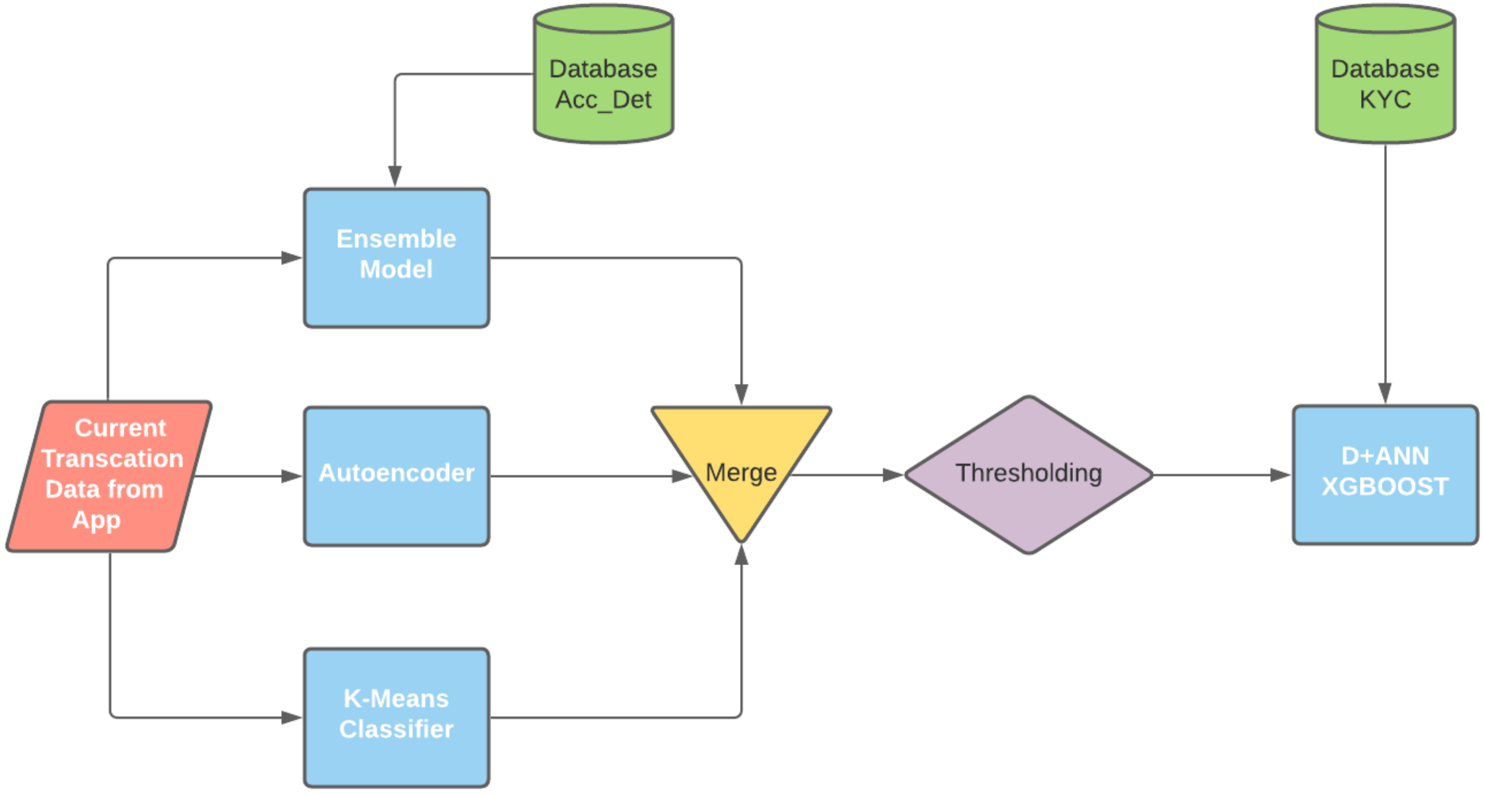


Fig1: Project Architecture

➡ Database Schema(SQL DB)

- SQL base Database to **increase system efficiency**
- Divided into 3 tables i.e.
  - Customer
  - Account
  - Transaction
- Separately linked **KYC details** table

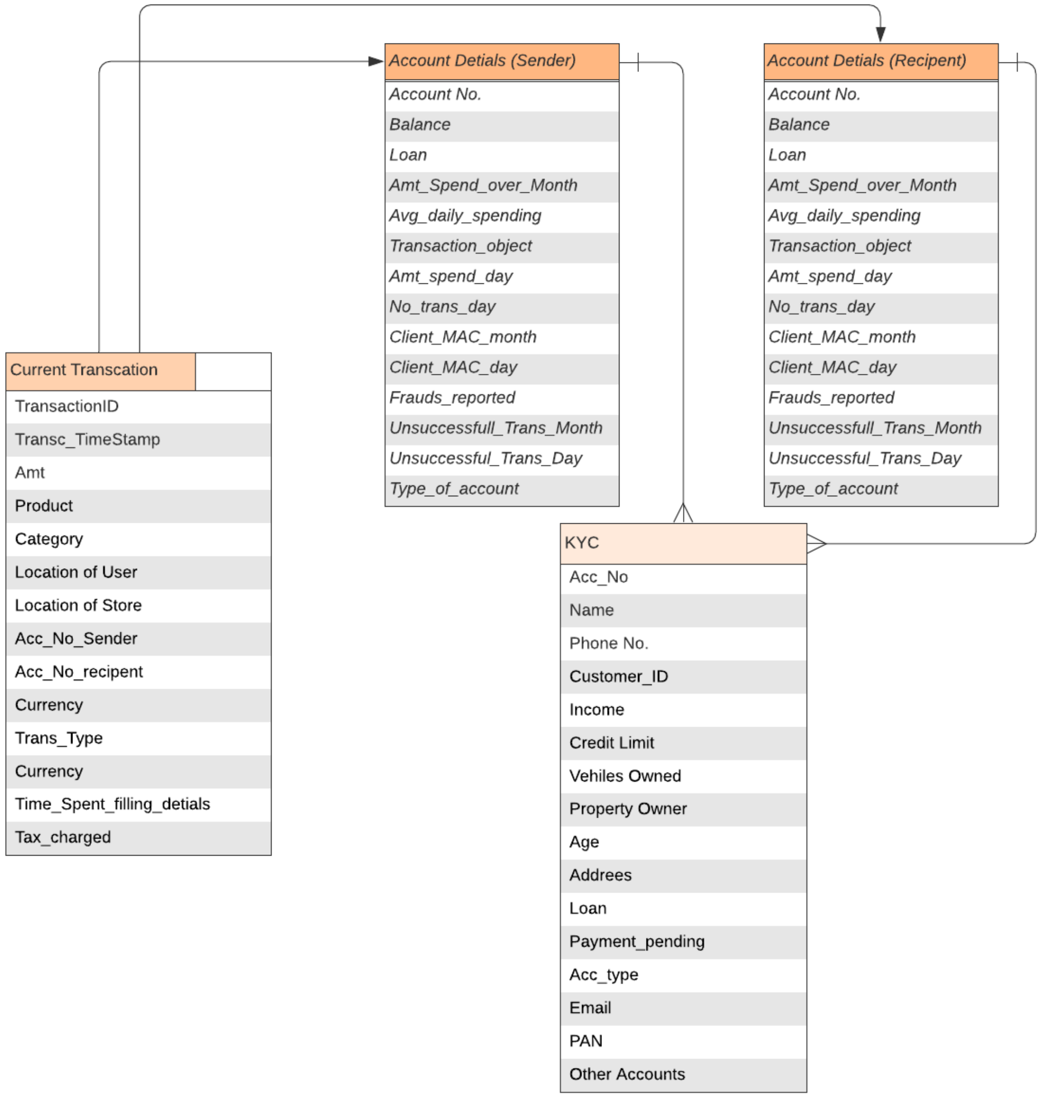
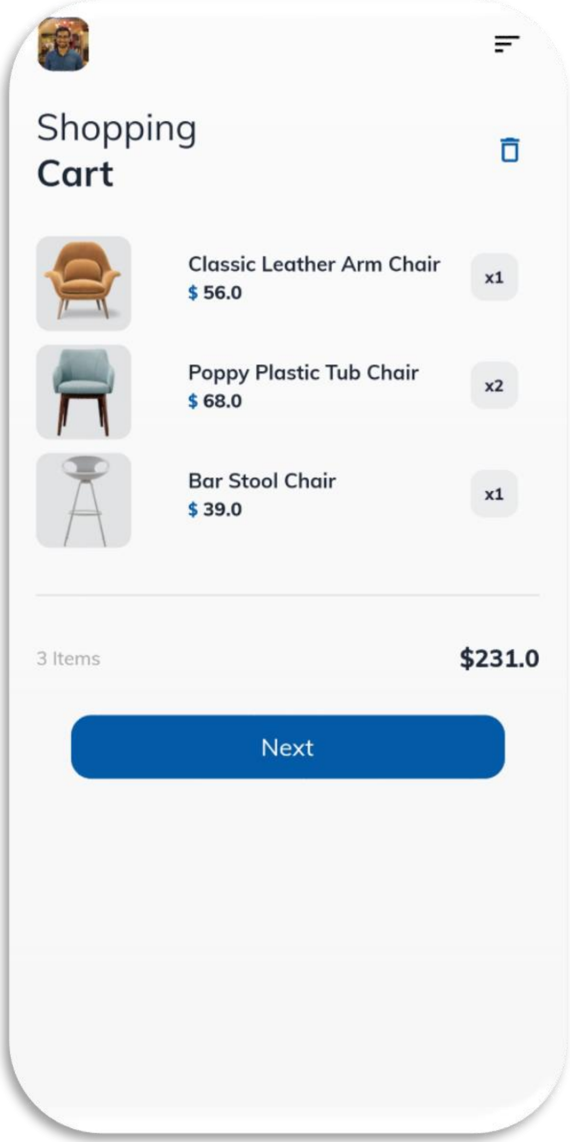
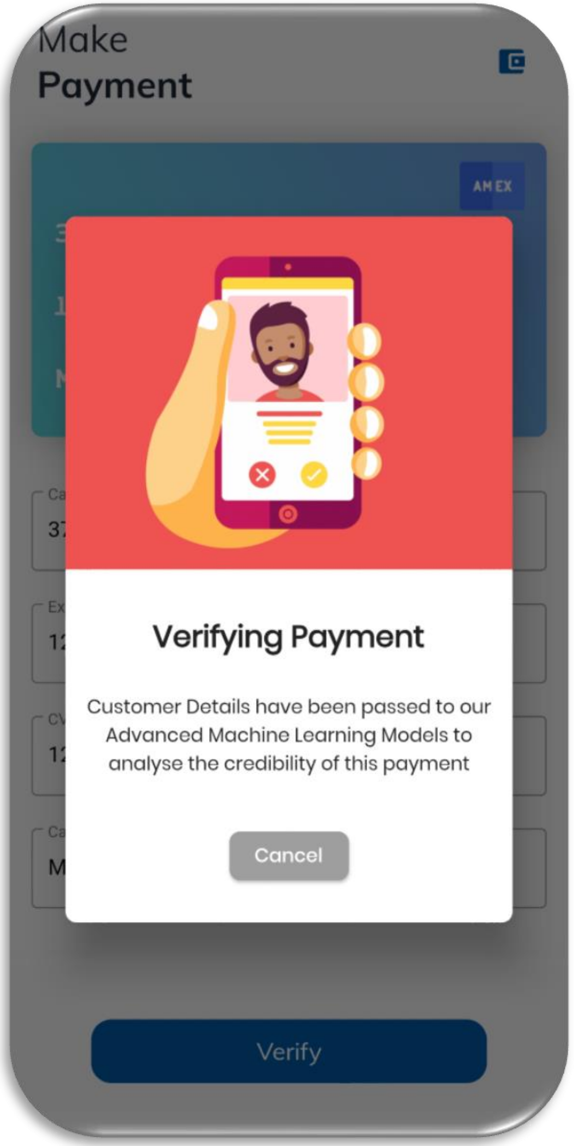


Fig1: SQL DB Table Structure

→ Frontend App

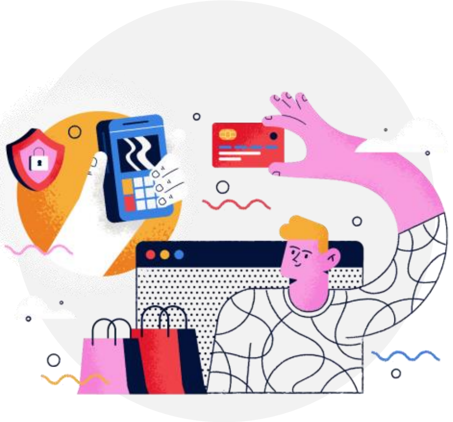
AMERICAN EXPRESS
CodeStreet’20





AMERICAN EXPRESS
CodeStreet'20

➡ Business Impact



Improved Accuracy

- Customers will face less down time and waits on their credit card spending because of the faster resolution of fraud cases

Sensitivity & Specificity

- Resolution of fraud cases from the genuine mistakes with greater accuracy would be a novel feature in our proposed system

Ease of use

- With our ready made plug and play APIs, this fraud detection system can be integrated with real time payment methods instantly

→ Novelty

Behavior and Location Analysis (BLA)

- Based on the geographical location where the credit card user lives he is put in one of the buckets to better understand the spending patterns of a locality and fill in the gaps

Tokenization

- Replacing the 16 digit card number with a **'Token'.** Credit card tokenization helps e-commerce websites improve security, as it eliminates the need for storing credit card data, and reduces security breaches.

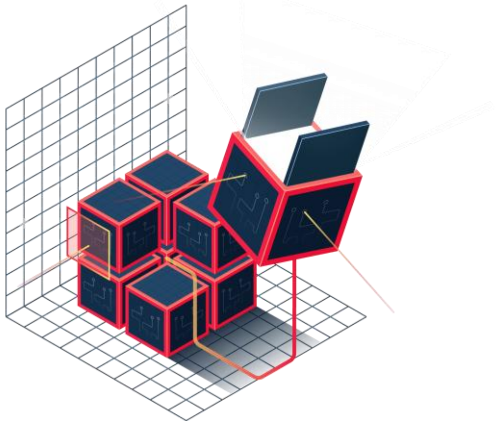
Smart Contracts

- Using a **Keyless Signature Infrastructure** provides more reliability and speed in storing KYC verification data in distributed networks in a blockchain, along with the ability to execute smart contracts

AMERICAN EXPRESS
CodeStreet'20

➡ **Going Forward**, we plan to

- Single, unified KYC verification Database for cutting down the repetitive individual checking by different institutions. Database will be maintained in a safe **BlockChain Architecture** which guarantees no hampering whatsoever. Also the number of fraudulent transactions will be linked to a person's KYC block and after a set threshold is crossed, the person will be temporarily blocked from conducting transactions online
- Deployment of **SmartContracts** which ensure that after a certain action is performed, a series of actions follow

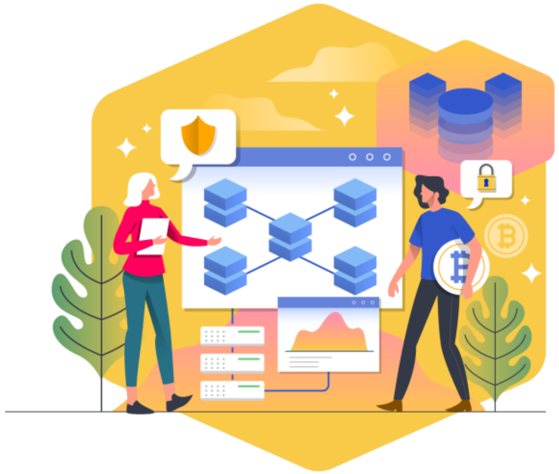


Safety Of Blockchain

+

Reliability of Pen and Paper

+



KYC on a distributed Network

Features of KYC Blockchain Architecture

➡ Financial Model

AMERICAN EXPRESS
CodeStreet'20

Cost1: API

0.2% of transactions

Cost2: Deployment

One-time cost of Rs 3.2 lack

Cost3: SDK

0.1% of transactions

Cost4: Maintenance

Rs 20,000-Rs 30,000/month

Cost5: Hosting

User Dependant(10,000)

→Tech Stack

Flutter

HEROKU

JS

Firebase

twilio

Keras

➞ Our Team

Sidhant Goyal
Data Scientist

Mohan Arora
Flutter Developer

Harshil Sharma
ML Research

Prateek Bansal
ML Research