Integrating Payments: Design Principles For A Cashless Future

Monojit Basu, Founder and Director,
TechYugadi IT Solutions & Consulting

techYugadi

Agenda

- Current and emerging techniques to integrate merchant apps seamlessly with payment service providers
- Point-to-point Security between merchant apps and payment gateways
- Data Security in the context of online payments
- Could BHIM app be a game-changer?

INTEGRATING PAYMENTS SEAMLESSLY

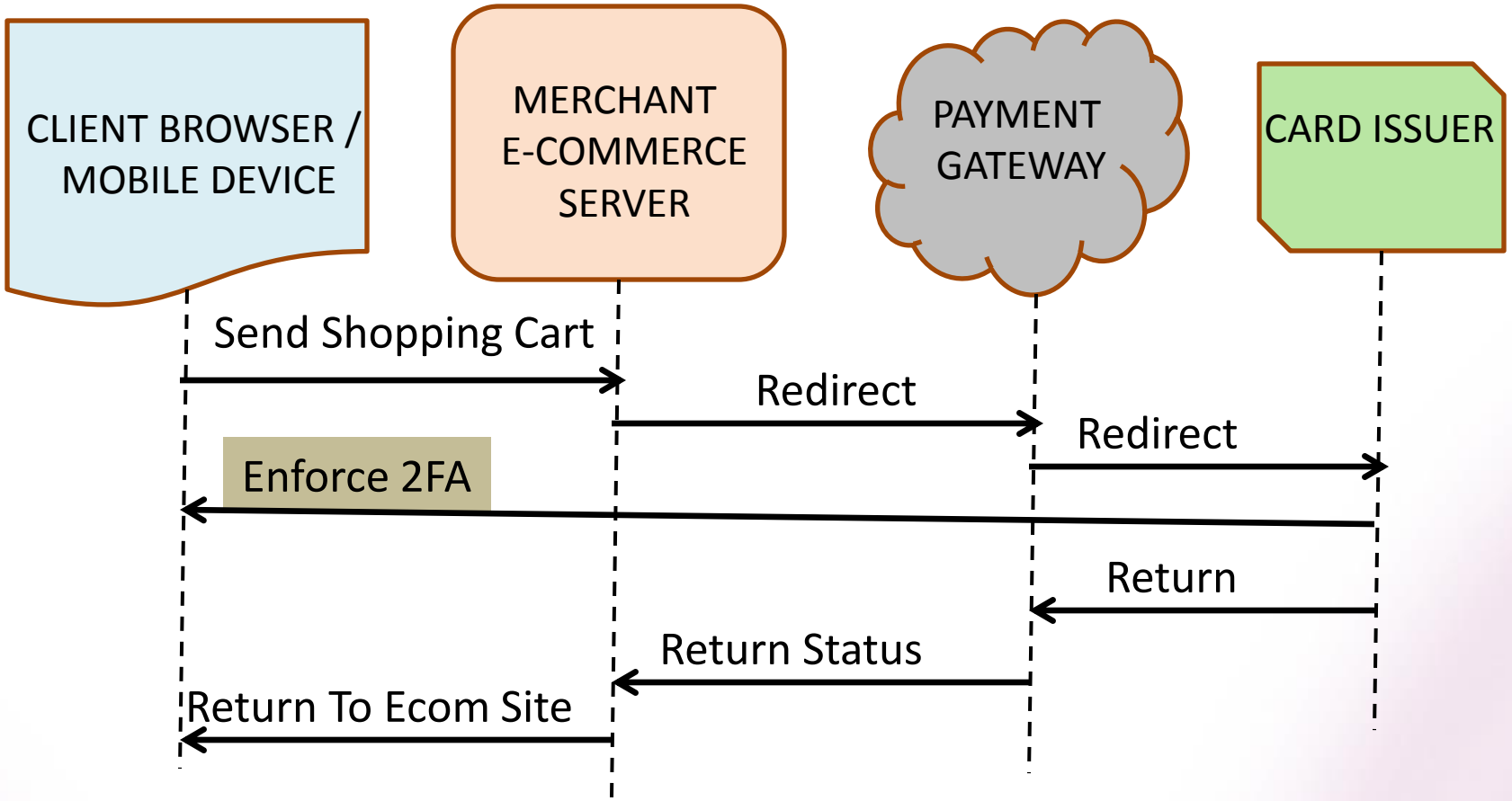
Seamless Integration Is A Moving Target

- Payments Pre-Processing Steps
  - Currency Conversion
  - Shipping Charges
  - Tax Calculation (Think GST if you are a marketplace or B2B exchange !)
- Payment Processing Steps
  - 2 Factor Authentication
  - Transaction Analysis
- Payment Post-Processing Steps
  - Deferred Approval / Payment Notifications
  - Fraud Analysis
  - Refunds

Transactions Within A Threshold: 1-Step Checkout

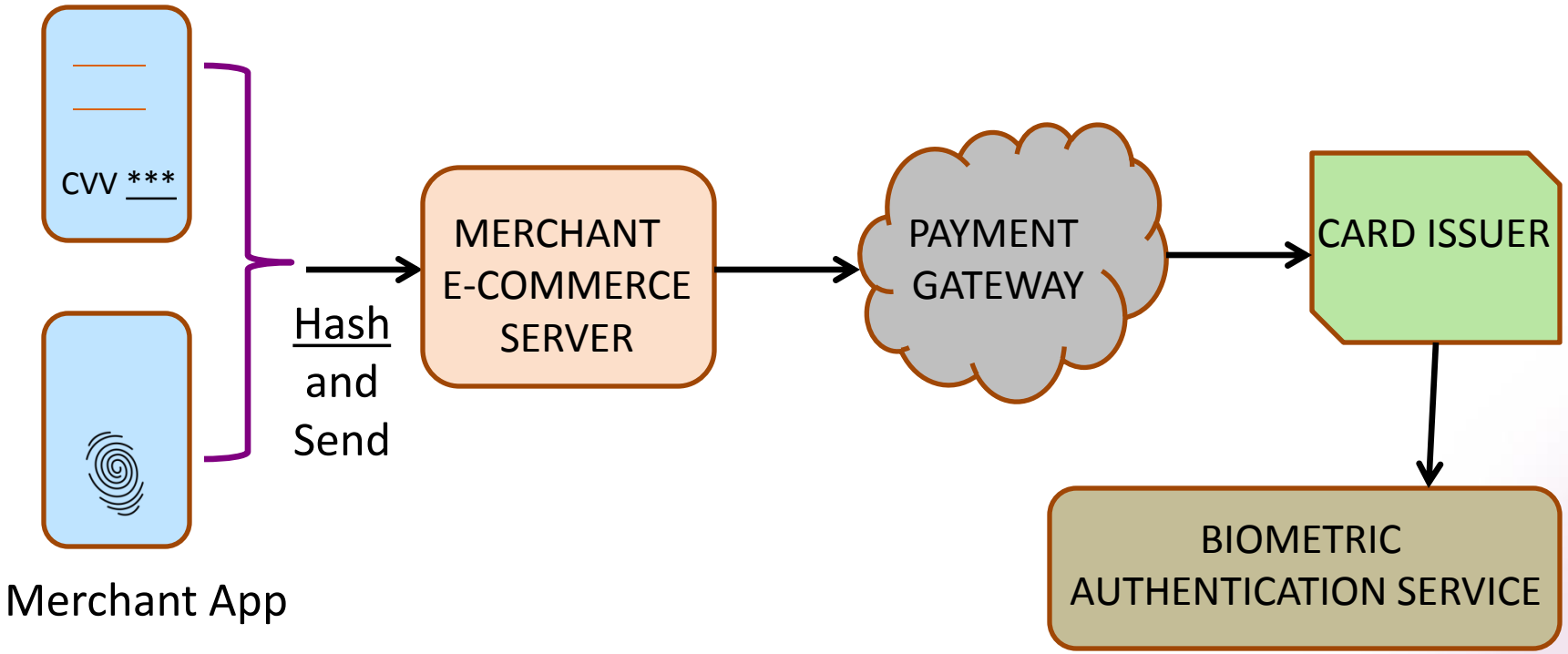
*Merchant Server may not be PCI-compliant and must avoid reading / storing card info as far as practicable

Higher Value Transactions: 2-Factor Authentication



2FA breaks seamless user experience

Mobile Apps: Streamlining 2-Factor Authentication



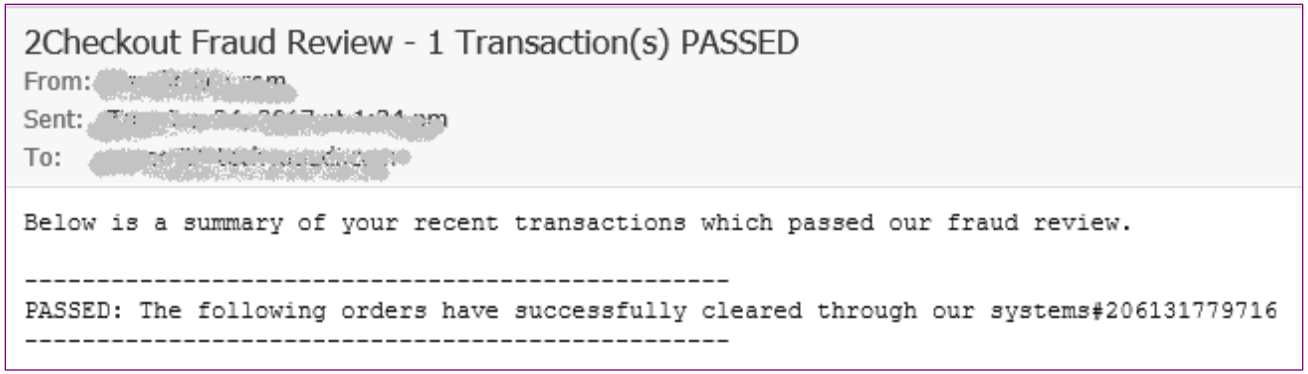
Will merchant app be trusted to capture biometrics?

May depend on the app: booking a hotel room *vs.* booking a restaurant table

Fingerprint image: courtesy Andrew Forrester and thenounproject.com

Fraud Avoidance And Analysis

- Currently fraud analysis on a transaction completes within 3 seconds on most payment gateways
  - Suspicious transactions kept *on hold* with notification on status change


2Checkout Fraud Review - 1 Transaction(s) PASSED
From:
Sent:
To:

Below is a summary of your recent transactions which passed our fraud review.

PASSED: The following orders have successfully cleared through our systems#206131779716

- With more powerful Big Data platforms the latency could come down significantly in the near future
- As a merchant you could be satisfied with pre-set fraud filters
- Or opt for a real-time cloud-based fraud analytics service (possibly provided by your payment gateway itself)

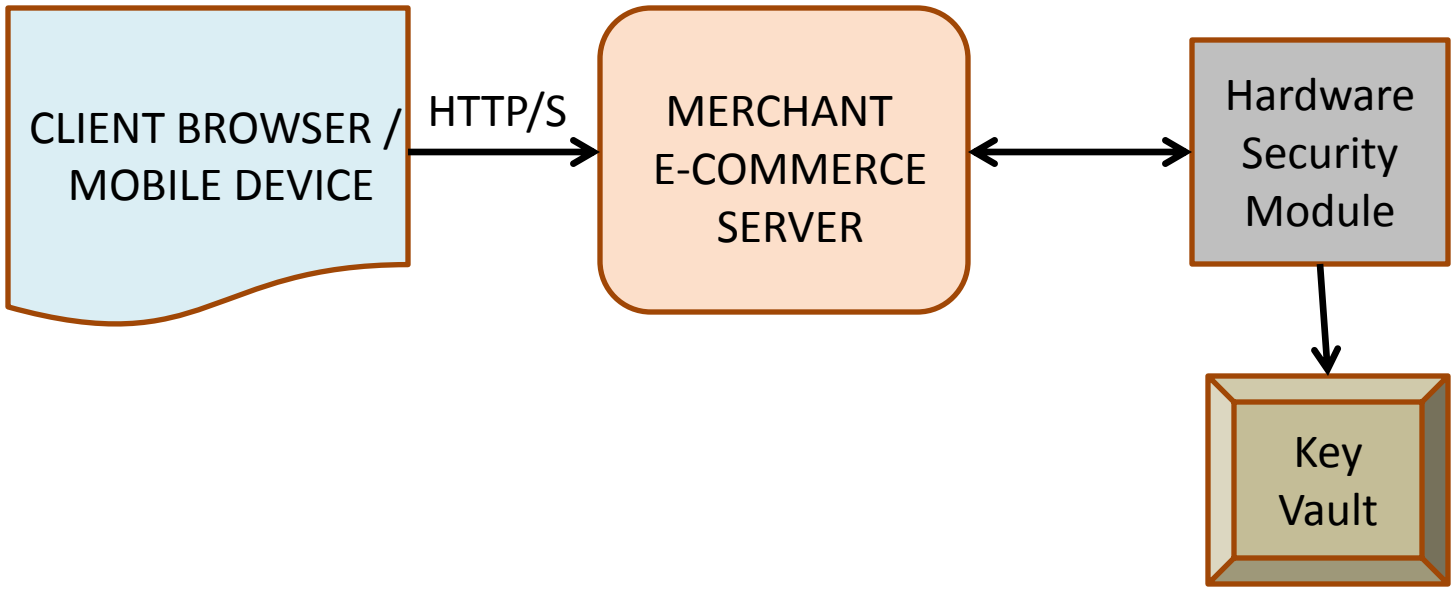
Virtues of Reactive / Asynchronous Programming

- A programming tip
- Payment integration is more than just an API call
- Its turning out to be like a workflow with synchronous and asynchronous elements
- You can model it using Reactive / Asynchronous Programming

```
callPaymentGatewayAPI()
then { result ->
  return
  result.fraudStatus()
} then { result ->
  return
  result.txnStatus()
} fail { error ->
  return 'Txn Failed'
} then { result ->
  display result on
  web-page
}
```

SECURING TRANSACTIONS

Point-to-Point Encryption



- PCI Data Security Standards includes guidelines for P2P encryption
- Decryption must be performed in a PCI-compliant HSM
- HSM can shield against in-memory scanning and other attacks

Encryption Best Practices

CRITERIA	DO	DON'T
SSL Certificate	Use SHA-256 based certificate	Don't use SHA-1
Transport Layer Security (TLS)	Use version >= 1.2	Don't use v1.0 or v1.1
Cipher	Avoid hard-coding	Don't use RC4 or DES
Root Certificate	Use secure 2048-bit Root Certificate	Don't use 1024-bit Root Certificate

Non-Repudiation

- Buyer also has a responsibility !
- Check-out page should require log-in or user registration
  - But this requires merchant to take adequate safeguards against phising
- IP Address from where a transaction was initiated, should be captured

  Customer IP Address: xxx.xxx.xxx.xxx

- Biometric authentication could make it simpler to enforce non-repudiation

Trends In Integrating Identity Management

- Merchant can delegate buyer login to a third-party federated identity management service

- Google Sign-In
  - Buyer can log-in to merchant site with securely with Google login
  - And pay with Android Pay seamlessly from merchant app

- Log In with Paypal
  - Based on OpenID 2.0 standards
  - Supports seamless checkout feature

Using Payment APIs Securely

- Managing API Keys provided by a Payment Gateway is a crucial to avoiding internal security threats
- API Keys – not only sandbox, but also *production,* will have to be accessed by multiple development and testing teams
  - API Key access must be strictly audited
  - API Keys must be subjected to life-cycle policies – in particular, changed periodically
- Some Payment Gateways (like *Authorizenet*) support multiple user accounts for each merchant
  - Create users from merchant organization with different set of privileges for administration of a Merchant Account

KEEPING DATA SAFE AND SECURE

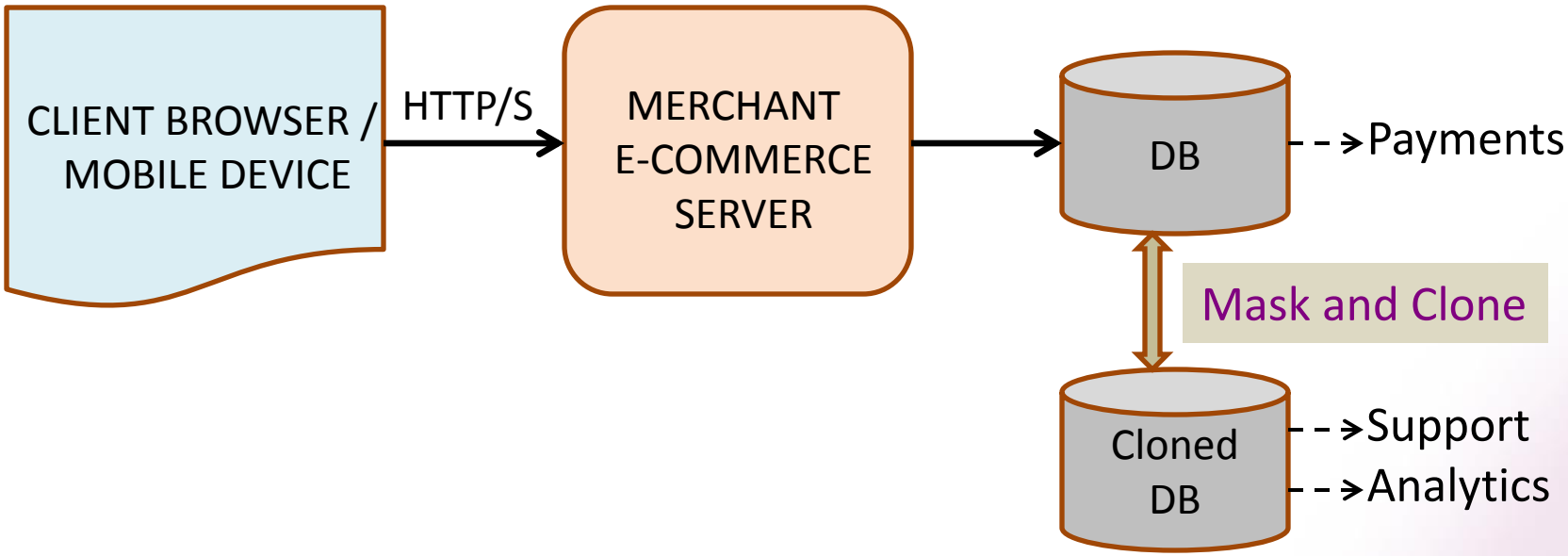
Avoid Storing Sensitive Data

```
public void refund(String txnId) {

    Sale sale = new Sale();
    sale.setId(txnId);

    RefundRequest refund = new RefundRequest();
    Amount amount = new Amount();
    amount.setCurrency("USD");
    amount.setTotal("10.00");
    refund.setAmount(amount);

    try {

        // Paypal API
        APIContext apiContext = new APIContext(clientID, clientAPIKey, mode);
        sale.refund(apiContext, refund);
```

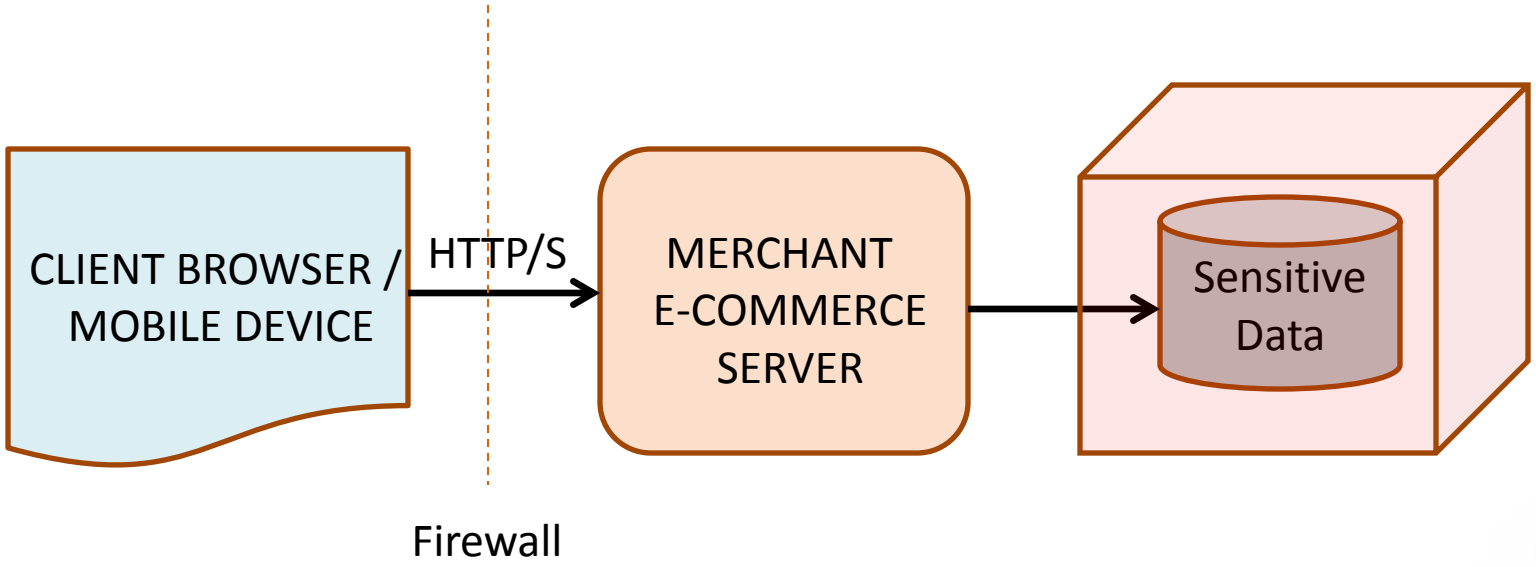
Merchant shouldn't need Credit Card number to initiate refunds

But What If You Have To

Example: Recurring Payments for a Cloud-hosted Service

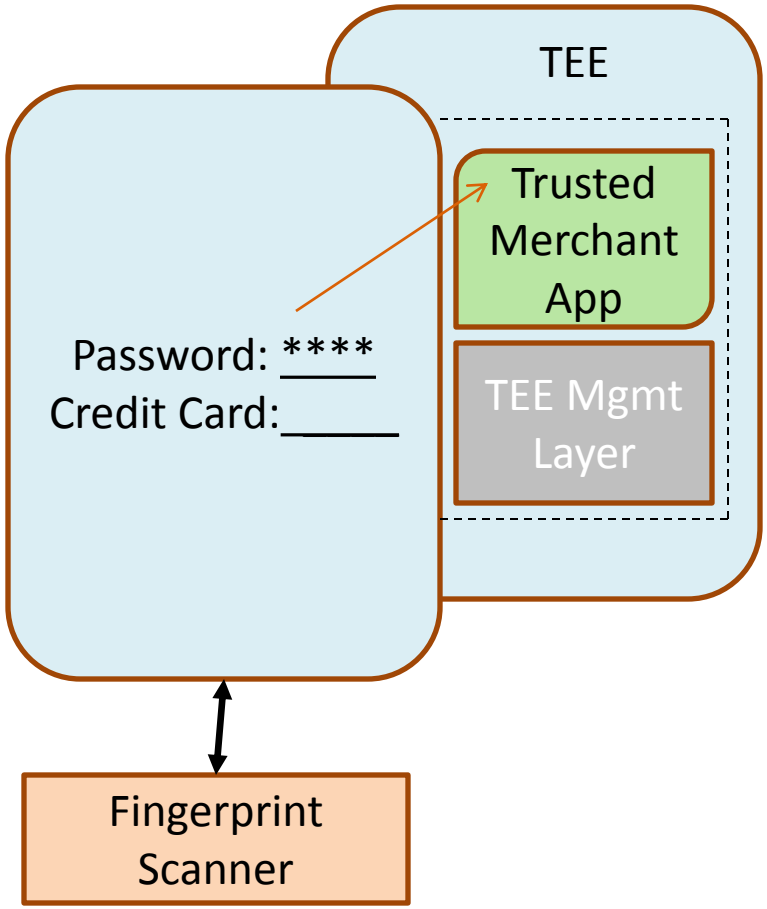


Guidelines For Storing Sensitive Data



- ✓Strong Encryption with hashed indexes
- ✓Disconnected from Internet access
- ✓Physically isolated and locked

Data Capture On Mobile Devices

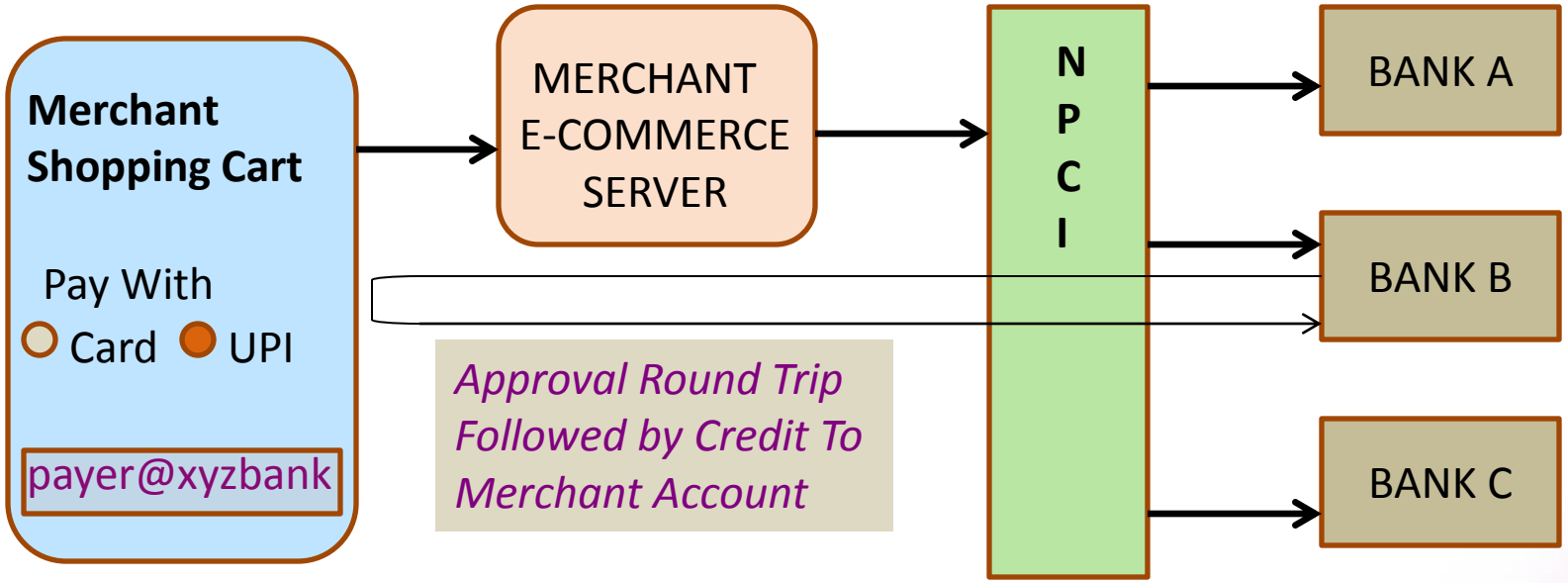


(*Ideally*) data captured through user input should be passed through a trusted path into a **Trusted Execution Environment (TEE)**

Support for TEEs on mobile devices is work in progress

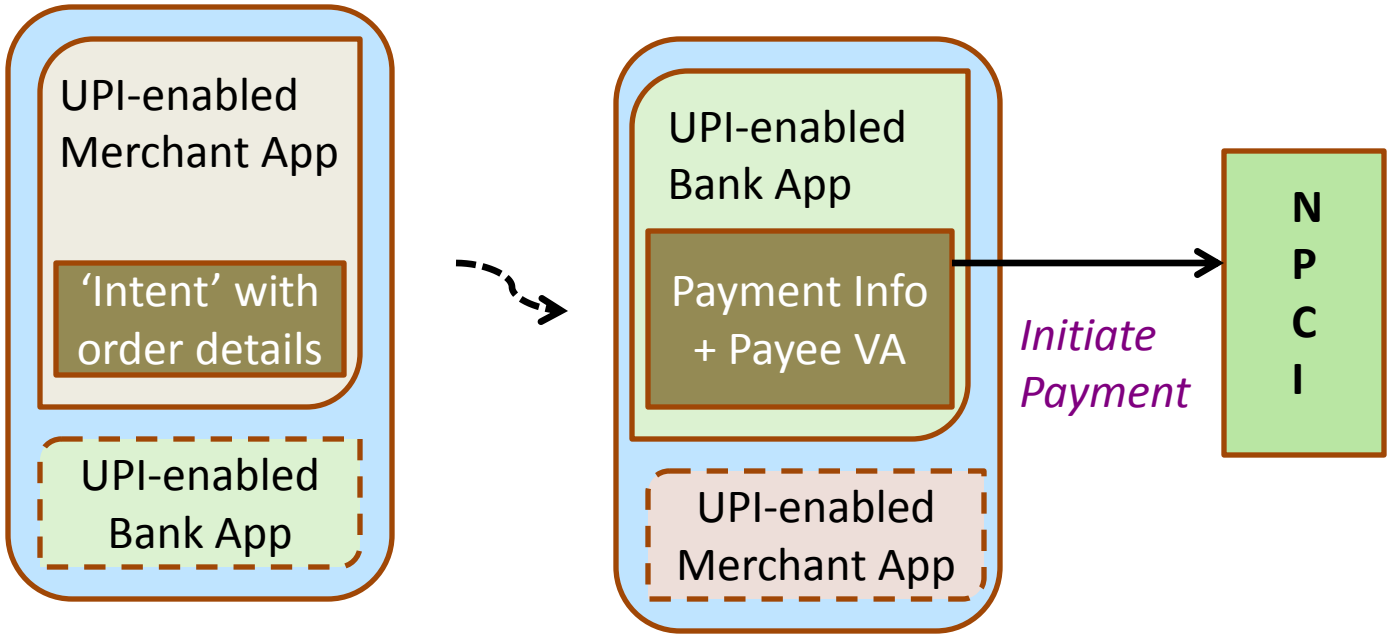
INVOKING BHIM FROM A MOBILE APP

Unified Payments Interface v1.0



NPCI: National Payments Corporation of India

Unified Payments Interface v1.1



Payment Service Provider (Bank) App must listen for UPI payment 'Intents'

UPI Checks Most Of The Boxes

- Disintermediated direct transfer: maximum financial inclusion
  - Less than 25 million active credit cards in India as of March 2016
- Built-in multi-factor authentication
- Smooth transition between shopping cart and payment screens
- In-built data masking through use of Virtual Addresses
- Extensible mechanisms for fraud filtering

```
<RiskScores>
 <Score provider="sp" type="TXNRISK" value=""/>

<Payer >
 <Info>
   ...
   <Rating VerifiedAddress="TRUE|FALSE"/>
```

Thank You

October 5-6, 2017
Indian Institute of Science, Bangalore
www.modsummit.com

Register early and get the best discounts

April 23-28, 2018
Indian Institute of Science, Bangalore
www.developersummit.com