

Transfer on Delivery (ToD)

Company : *AmberPay*

Details : A Payment Gateway providing ToD

Members: Rajath Alex & Nirvik Ghosh

Business Problem

- Major drawbacks in existing payment methods in Merchant Platforms
- Examples :
 - 1) *Cash on Delivery (CoD)*
 - 80% of Indian e-Commerce orders are CoD orders.
 - It is a very expensive method for the merchant.
 - 8 to 10 % net transaction cost for merchant per CoD order
 - High rate of return on CoD orders of around 20%.
 - Demonetization has made it hard to arrange for cash.
 - 2) *Online (NetBanking, Credit Card, Debit Card, Wallets)*
 - Lack of trust for customer as he is paying even before physically seeing the product.
 - In case of issue, refunds can take 3 to 7 business days.

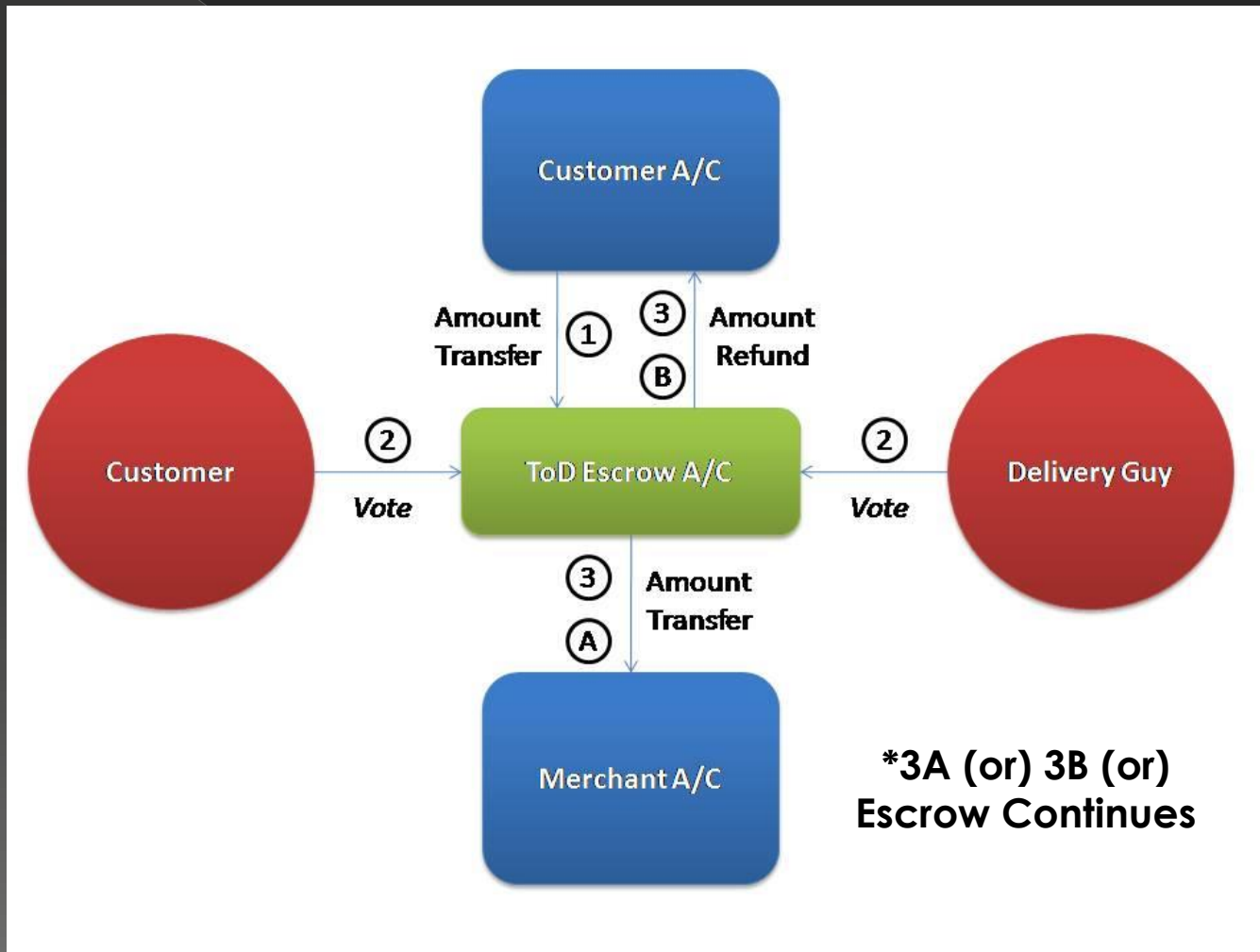
What is ToD? Why ToD?

- Transfer on Delivery (ToD) is an instant transfer payment method with escrow aimed at all e-Commerce customers.
- Escrow service is provided to both the end consumer and the merchant.
- It's an online means of payment at time of checkout – Beneficial to merchant due to low transaction fees.
- It's a trustable form of payment for the customer as money only gets released to merchant after successful delivery.
- In case of a product issue, money is instantly refunded to the customer account from escrow.
- In case of successful delivery, money is instantly transferred to merchant account from escrow.

Functionalities of ToD

- Uses Unified Payments Interface (UPI) for instant transfer directly from Bank Account
- Leverages security infrastructure of NPCI.
- Quick checkout – Only needs Customer's Virtual Payment Address (VPA)
- Uses proprietary consensus logic to instantly transfer money to customer or merchant from the escrow depending on voting responses from customer and delivery guy at time of delivery
- Money continues to be in escrow in case of conflicting responses from customer and delivery guy at time of delivery
- Trust factor of Cash on Delivery (CoD)
- Convenience of online prepayment – No hassle of arranging for change
- Can be easily integrated with Merchants through AmberPay API

Money Flow Architecture



How is ToD better?

- ◉ *Net transaction charge of 2.4% for Merchant only on amounts that get transferred from escrow to merchant and no transaction charges for refunds (Business Model explained in next slide)*
- ◉ *Cash on Delivery (CoD)*
 - 1) Much cheaper than CoD for merchants – Around 2.4% for AmberPay ToD as compared to 8 to 10% for CoD.
 - 2) No hassle of arranging for change from customer perspective
- ◉ *Credit Card, Debit Card*
 - 1) More trust for customer as money held in escrow until delivery
 - 2) Cheaper for merchants due to no extra charges being paid to Card Networks (Eg: Visa, Mastercard) apart from payment gateway charges – Around 2.4 % for AmberPay ToD as compared to net Gateway+Card Network charges of 3 to 5%.
 - 3) No hassle of entering all details at checkout – Quick checkout using just Customer VPA
 - 4) Instant Refund and Transfer
- ◉ *Wallets (Eg: Paytm, FreeCharge, MobiKwik)*
 - 1) More trust for customer as money held in escrow until delivery
 - 2) No 4% charge for Merchant on transferring to bank account from wallet, as Bank account is being used directly and securely through UPI.
 - 3) Instant Refund and Transfer

Business Model

- Currently using UPI has no transaction charges
- AmberPay is only going to charge the Merchant 2.4% for transactions that go from escrow to merchant account
- Refunds are not charged unlike regular payment gateways that charge for all transactions, whether they get refunded later or not
- This is viable by considering the following situation:
 - 1) Plain gateway charges are generally around 2% (Without considering Card Network charges)
 - 2) Around 20% of orders are returned on average
 - 3) Assuming the above 2 points, AmberPay needs to charge 2.5% on transactions that go through from escrow to merchant accounts in order to match revenues of current payment gateways
 - 4) AmberPay will charge 2.4% in order to be competitive.

Tech Stack & ICICI APIs Used

- Tech Stack

1. Front End : HTML + Javascript + CSS
2. Back End : Python Flask Web Framework
3. ORM : SQLAlchemy
4. Database : SQLite

- ICICI Bank APIs Used:

- 1) Authentication Service
- 2) Participant Data Mapping
- 3) Create VPA for your Account
- 4) Fund Transfer from VPA to VPA
- 5) Bank Account Summary