Payment System

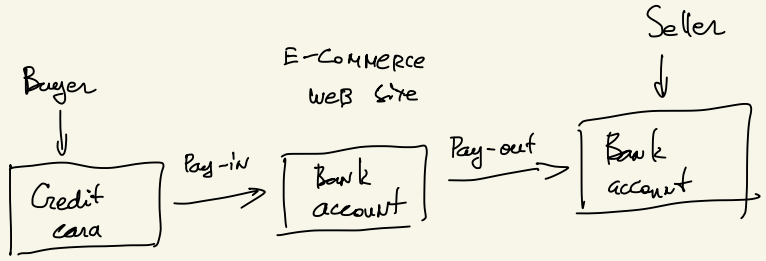
Use third-party payment processor Stripe, Brain tree, Square. Do not store card numbers directly.

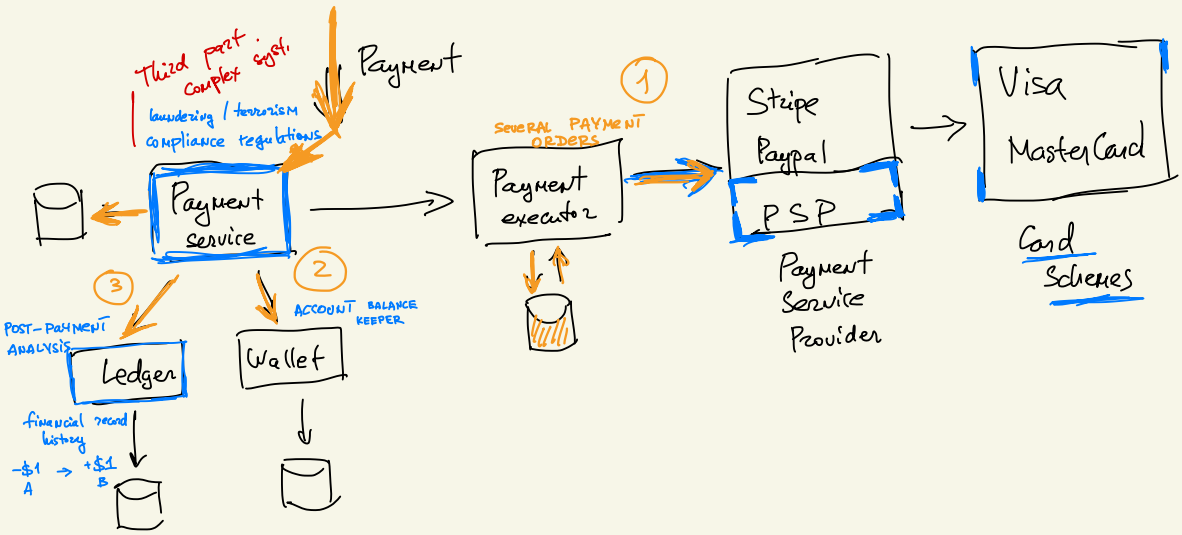
1 mill daylay transactions.

Reconciliation and fix inconsistencies.

Pay-in / pay-out flow.

$$\frac{1.000.000 \text{ trans. day}}{10^5 \text{ secs}} = 10 \text{ TPS}$$





Post /v1/payments

req:

buyer_id		json
checkout id		string
credit_card_info	ENC!	json
payment_orders		list

payment_orders:

seller_account	str
amount	str
currency	str
Paym_order_id	str

Send money to seller

Paym_order_id: globally unique

GET /v1/payments/{:id}

returns execution status of a single payment order

Data model:

Relational ACID, maturity RDBMS, monitoring tools.

payment_event

checkout_id	Str
buyer_info	Str
seller_info	Str
credit_card_info	PSP format
is_payment_done	bool

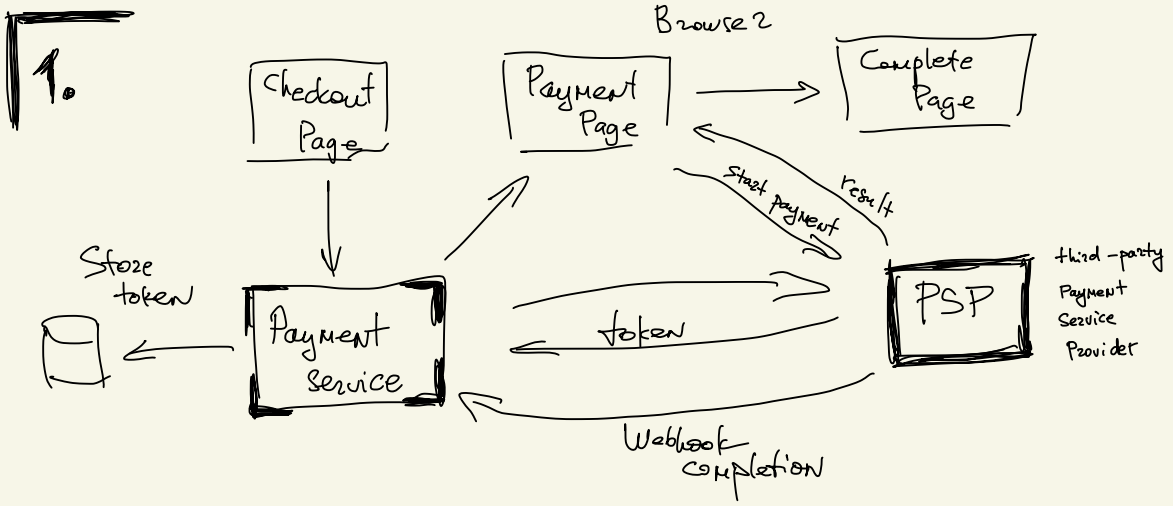
payment_order

payment_order_id	Str
buyer_account	Str
amount	Str
currency	Str
checkout_id	Str
payment_order_status	ENUM: NOT_STARTED, EXECUTING, SUCCESS, FAILED
ledger_updated	bool
wallet_updated	bool

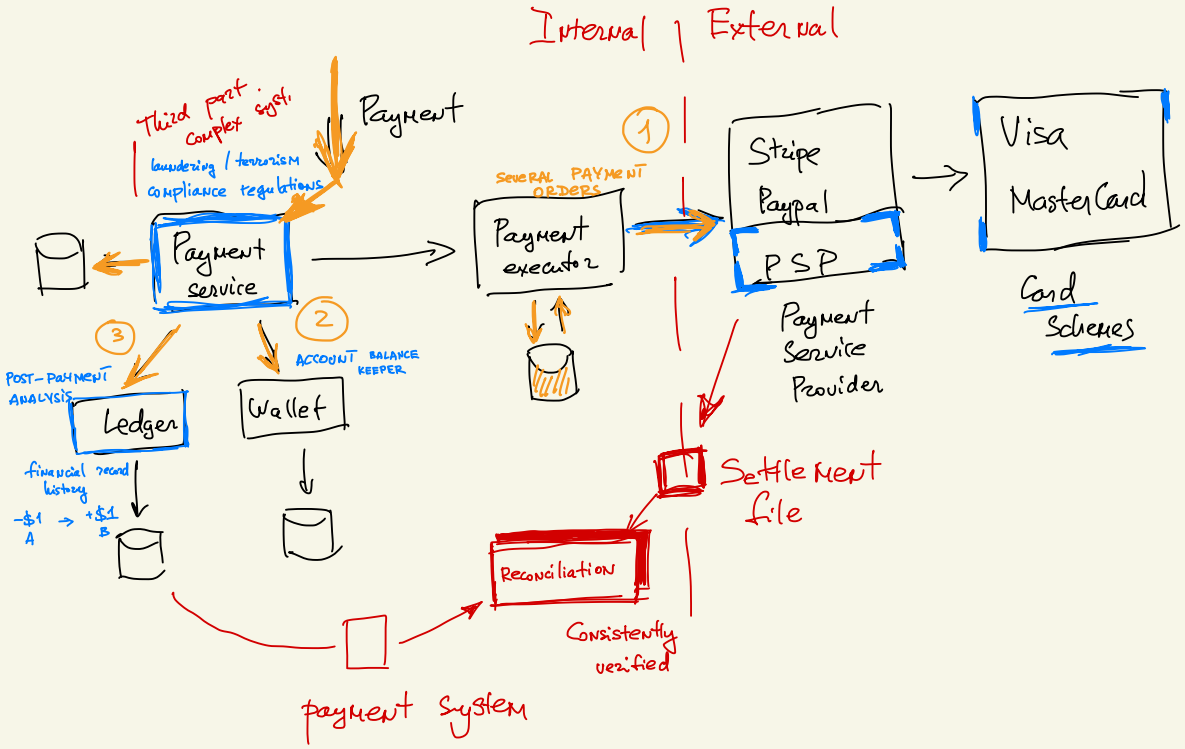
Design Deep Dive

1. PSP integration
2. Reconciliation
3. Handling payment processing delays
4. Communication among internal services
5. Handling payment payments
6. Exact once delivery
7. Consistency
8. Security

1.



2. Ensuring correctness delivery in Async.
Reconciliation.



3.

Delay because : high risk, requires human to review
credit card extra protection

+ PSP return pending status
+ PSP tracks pending payments on our behalf and notifies
Payment Service.

4.

Sync. HTTP for small-scale system
Low perform / tight coupling / poor failure isolation
Hard to scale

Async. MQ – single receiver
Kafka – multiple receiver

5. Reliability, fault tolerance.

- tracking payment state
- retry queue

6. Double charge customer problem.

at-least-once Retry (network error)

at-most-once Idempotency Check
(Server will reject second request with the same UUID)
429 Too Many Requests Status Code

7. Stateful services:

- Payment Service (Payment-related data)
- Ledger (Account data)
- Wallet (Balance)
- PSP (Payment execution status)
- Replicated Data among Replicas

! Exactly one processing between internal services is important.

PSP - external service. ← Rely to Reconciliation } Night job
& Idempotency.

1. Reads & Writes from the Primary DB only!
2. All replicas must be always in-sync!

8. Security prevention :

Req/resp tampering	—	HTTPS
	—	encryption / monitoring
Man-in-the Middle	—	SSL + certs
Data loss	—	DB replicas across multi AZ, snapshots
Distributed Denial-of-service	—	FW, Rate Limiting
Card theft	—	Tokenization
PCI compliance	—	PCI DSS
Fraud	—	Address / card verification (CVV), user behavior analysis