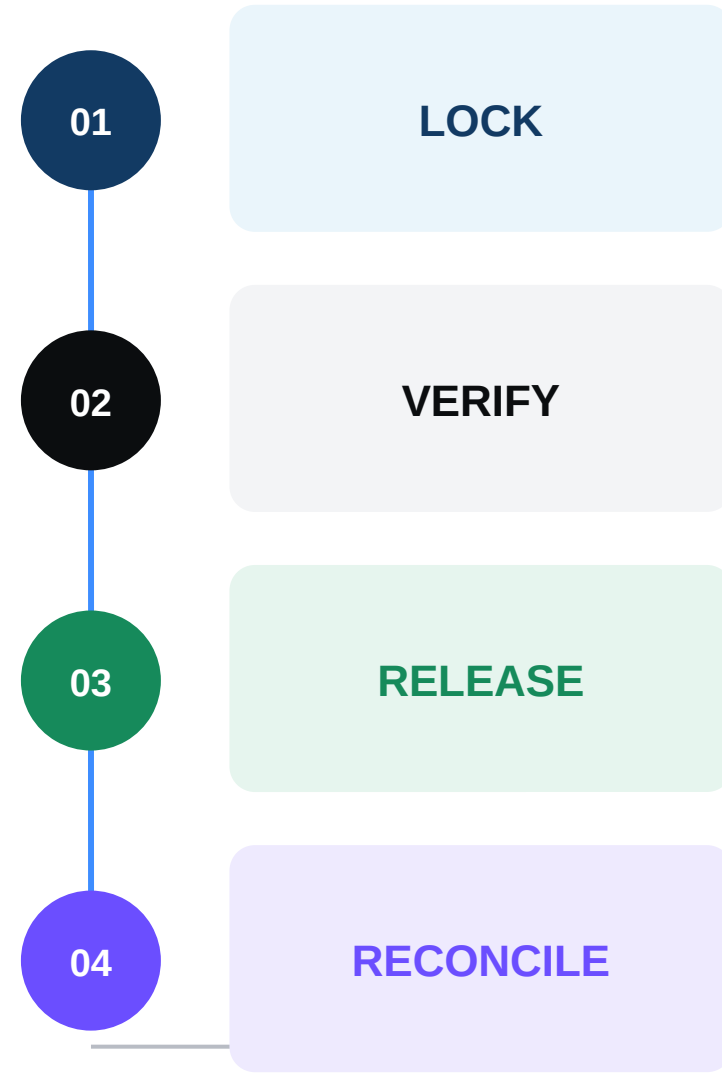
FINTECH × SMART CONTRACTS

Smart contracts in fintech

A production architecture for programmable B2B settlement

ARCHITECTURE • DFD • FLOWCHART • SECURITY • ROLLOUT



Brajkishor Pandey · Fintech Expert

Use the contract as a settlement controller

The winning design is hybrid—not chain-only.

SMART CONTRACT

What it should do

- Lock tokenized settlement value
- Enforce deterministic state transitions
- Release, refund or pause by rule
- Emit verifiable settlement events

FINTECH PLATFORM

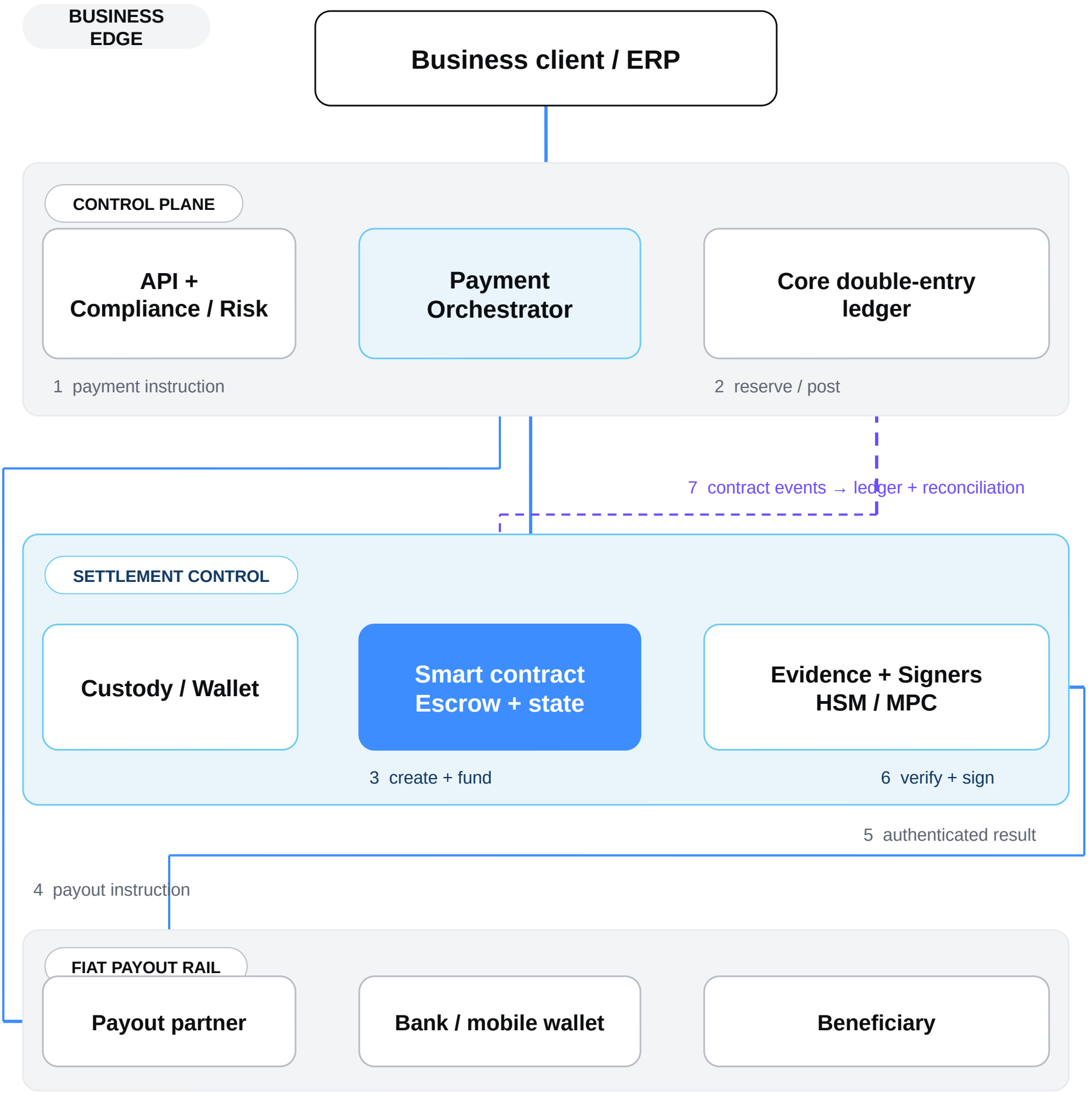
What stays off-chain

- KYC/AML, sanctions and fraud controls
- FX pricing, treasury and liquidity
- Double-entry ledger and accounting
- Bank and mobile-money orchestration

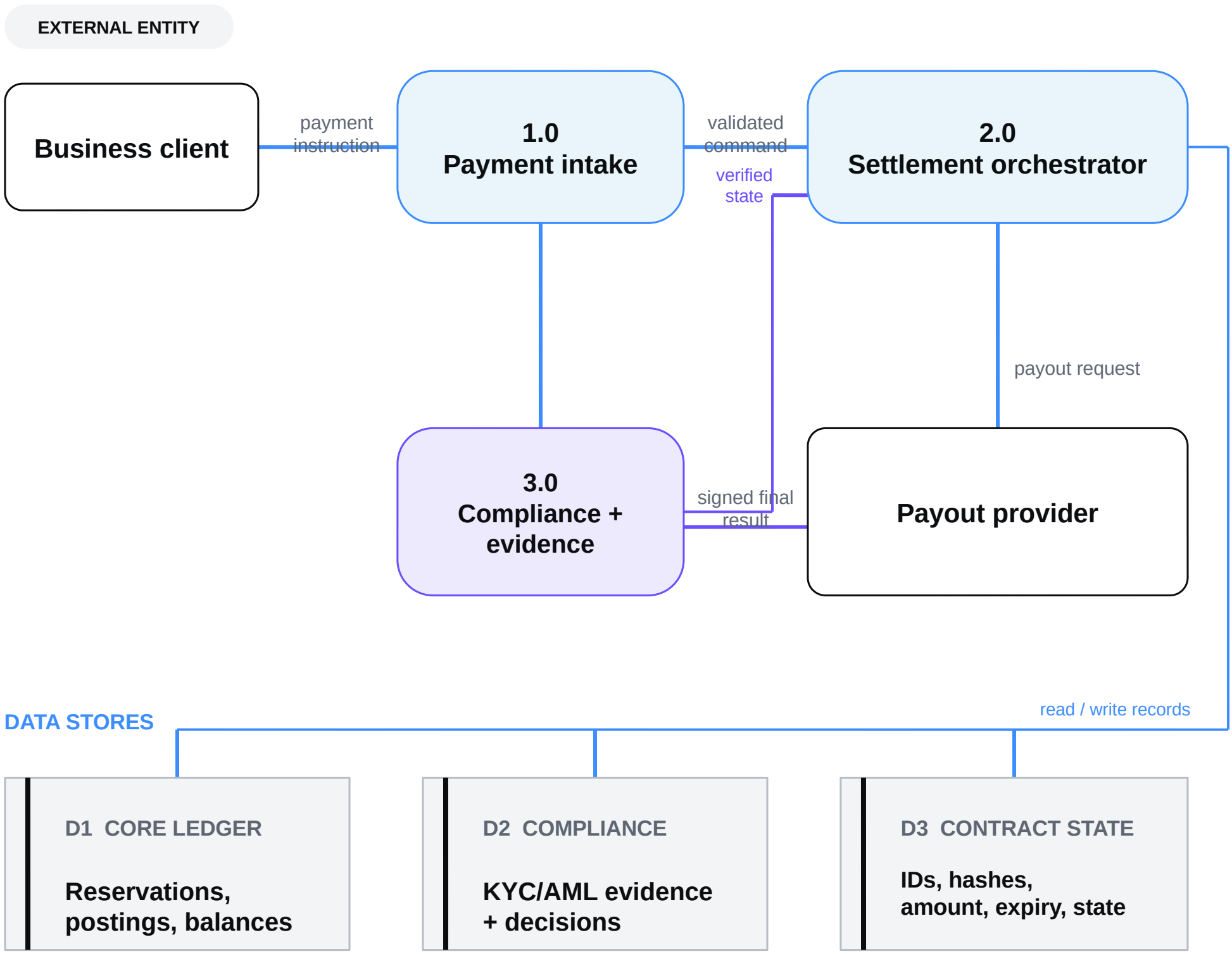
Regulated rails + Internal ledger + Smart contract

= a programmable settlement flow with controlled risk

Hybrid B2B payment stack



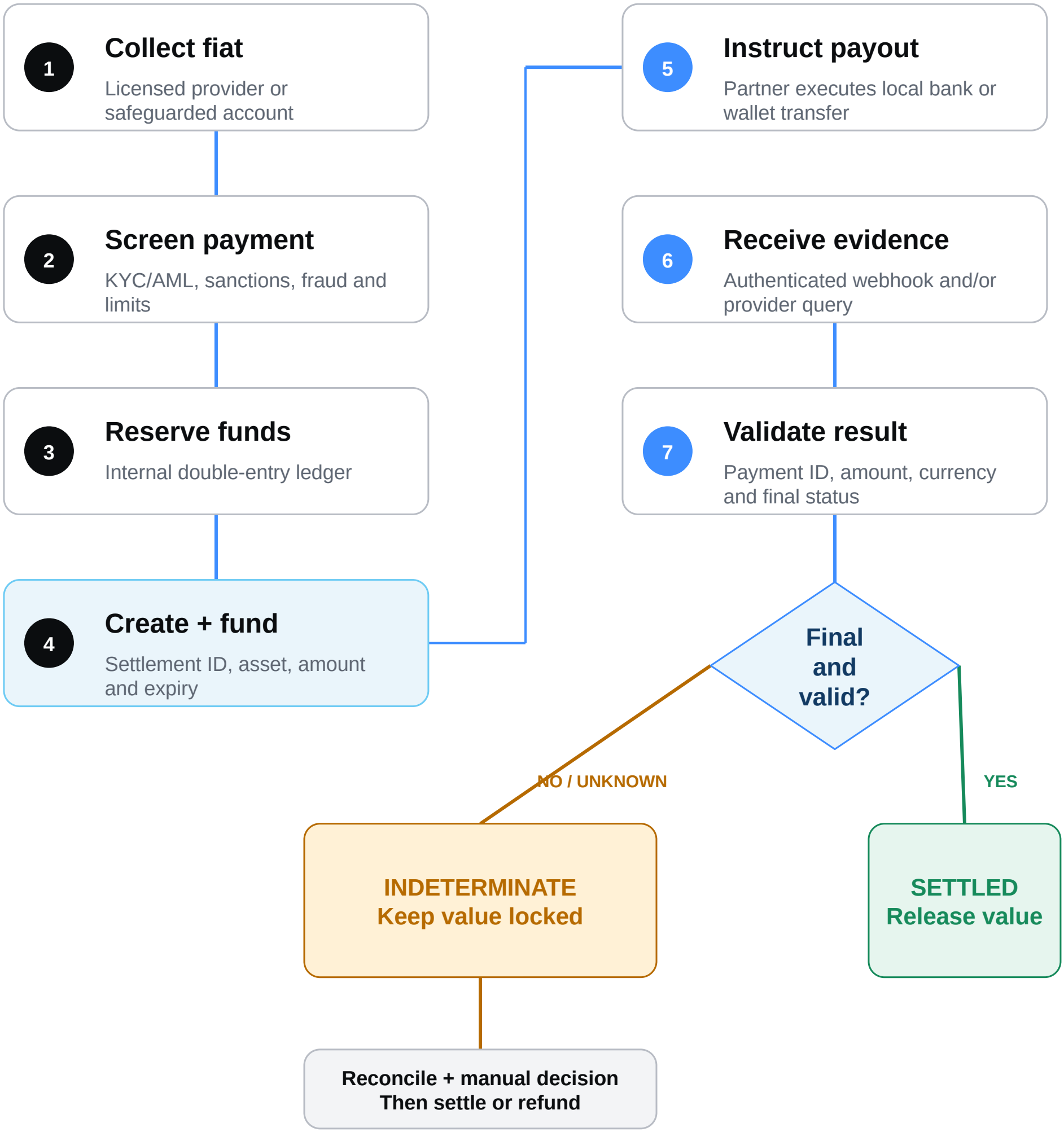
DFD Level 0: data moves; secrets do not



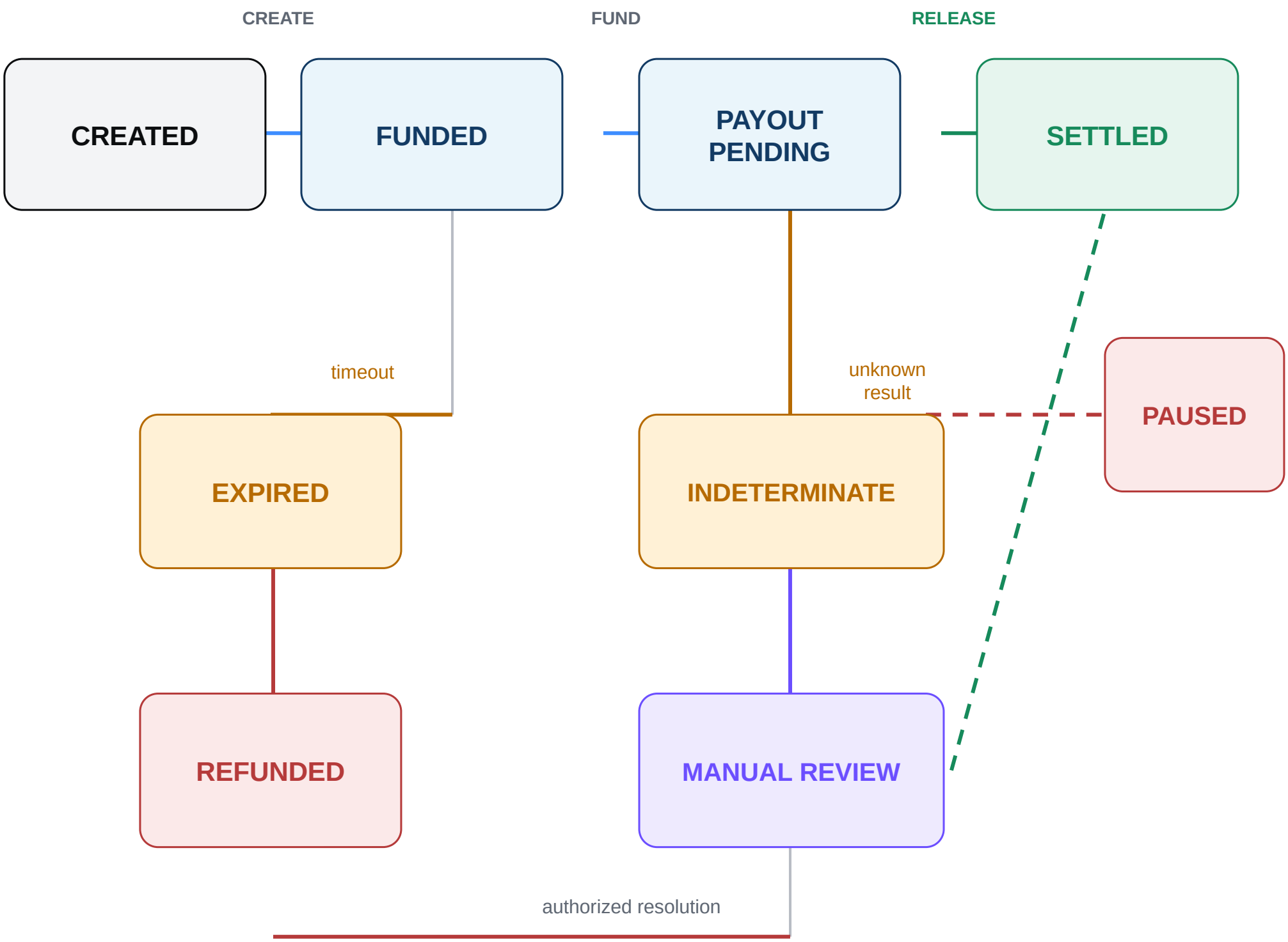
Privacy boundary

Names, KYC files, bank details and provider payloads stay off-chain. The contract receives references, hashes and authorized state transitions.

From payment instruction to final settlement



Every settlement transition is explicit



Critical invariant

A provider timeout is not a failed payout. Move to INDETERMINATE, stop blind retries, and reconcile before the next state transition.

On-chain vs off-chain

Put proof on-chain. Keep personal and commercially sensitive data in controlled systems.

ON-CHAIN	OFF-CHAIN
IDENTITY **Pseudonymous settlement ID**	IDENTITY **Names, KYC files, contact data**
MONEY **Asset, amount, expiry**	MONEY **FX quote logic, spread, liquidity**
STATUS **State + permitted transitions**	STATUS **Risk scores, sanctions decisions**
EVIDENCE **Evidence hash + signer proof**	EVIDENCE **Provider payloads and webhooks**
AUDIT **Contract events + timestamps**	AUDIT **Bank / wallet account details**

Design test: if a blockchain explorer exposed this field, could it identify a person or reveal pricing strategy? If yes, keep it off-chain.

How a bank result reaches the contract

The contract cannot call a bank API. An off-chain service must transform a provider result into verifiable, authorized evidence.



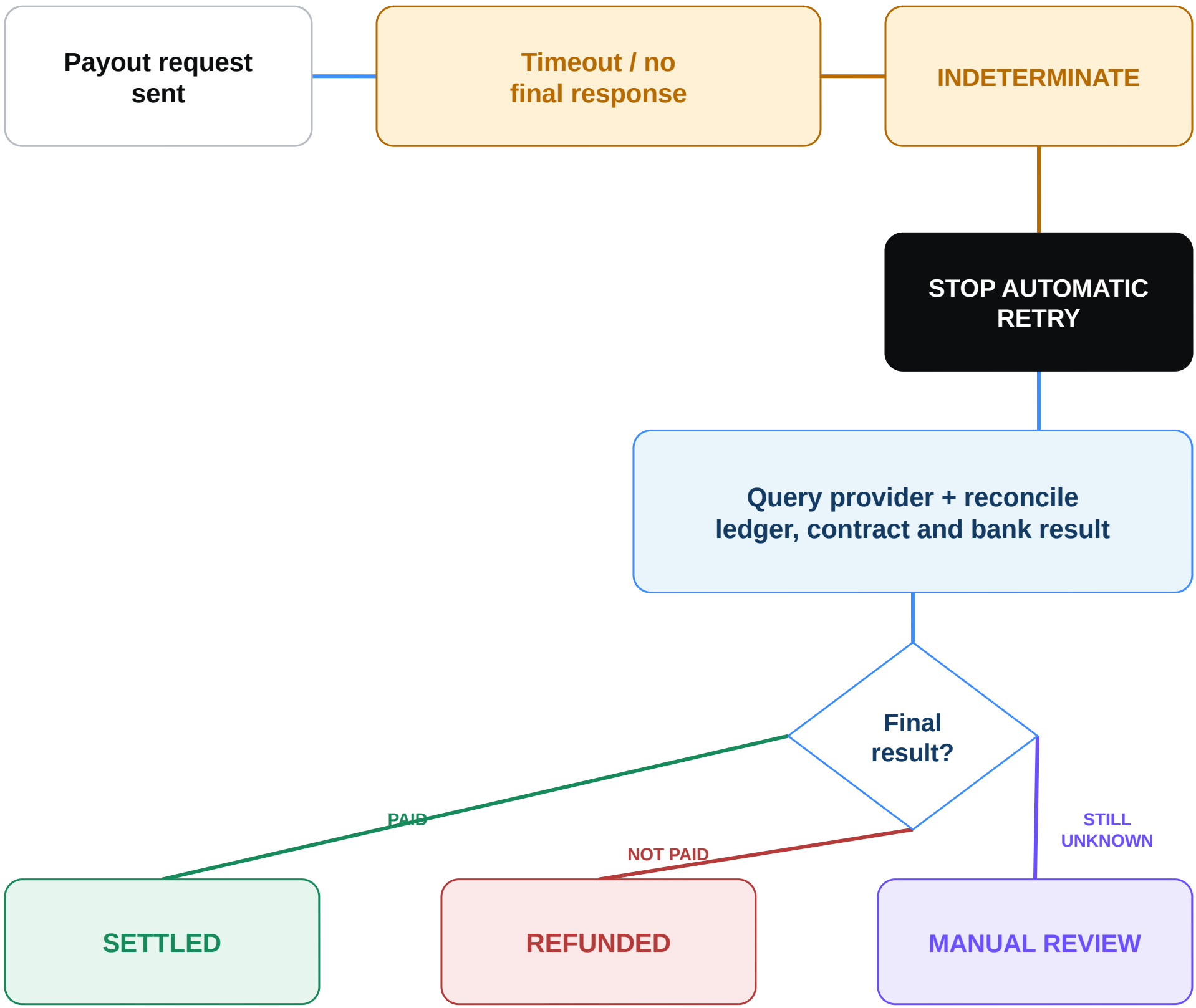
Five validation gates

1	**Authenticate the source**	mTLS/signature, allowlisted endpoint, timestamp window
2	**Bind references**	provider transaction reference ↔ internal paymentId
3	**Match financial facts**	amount, currency, beneficiary reference and corridor
4	**Accept terminal evidence**	only a documented final status can release value
5	**Prevent replay**	nonce, idempotency key, signer quorum and role checks

Only then: release, refund—or remain safely locked.

Unknown is a real payment state

Treat uncertainty as a controlled state—not as permission to retry the money movement.



Production safeguards

- Idempotency keys
- Funds stay reserved
- Safe retry policy
- Continuous reconciliation

Five gates to production

Start narrow, prove invariants, then expand corridors and value limits.

1

SANDBOX

Model states + invariants

Unit, fuzz, timeout and recovery tests

2

INTEGRATE

Connect regulated rails

Wallet/custody, on-ramp and payout adapter

3

PROVE

Build the evidence path

Signed evidence, quorum and event indexing

4

SECURE

Harden every authority

Audit, RBAC, multisig, HSM, pause and limits

5

PILOT

One controlled corridor

One asset + partner, low limits, live reconciliation

BLOCKCHAIN
Settlement control

+

REGULATED RAILS
Fiat movement

+

INTERNAL LEDGER
Financial truth

Build for production—not just for the demo.