Approved for A. Paid to B.

The payment changed. The approval stayed.

ILLUSTRATIVE PAYOUT SCENARIO

APPROVED	DISPATCHED
USD 10,000	USD 10,000
Account A	Account B

A control pattern for fintech founders, CTOs and payment operations leaders.

BRAJKISHOR PANDEY

A queued payment. A changed destination.

ILLUSTRATIVE SEQUENCE

10:00

Approval recorded

USD 10,000 to Account A.

10:01

Beneficiary profile edited

The saved destination becomes Account B.

10:02

Worker reads the latest profile

The job uses B with the old approval.

10:03

Provider receives Account B

The instruction no longer matches approval.

The business risk: misdirected funds, delayed recipients and an approval trail that cannot explain the payout.

Approval belongs to a specific version.

Keep a protected snapshot of the significant payment details shown to the approver.

PAYMENT 1042 / VERSION 1

Amount + currency
USD 10,000

Beneficiary + destination
Supplier A / account ending 1142

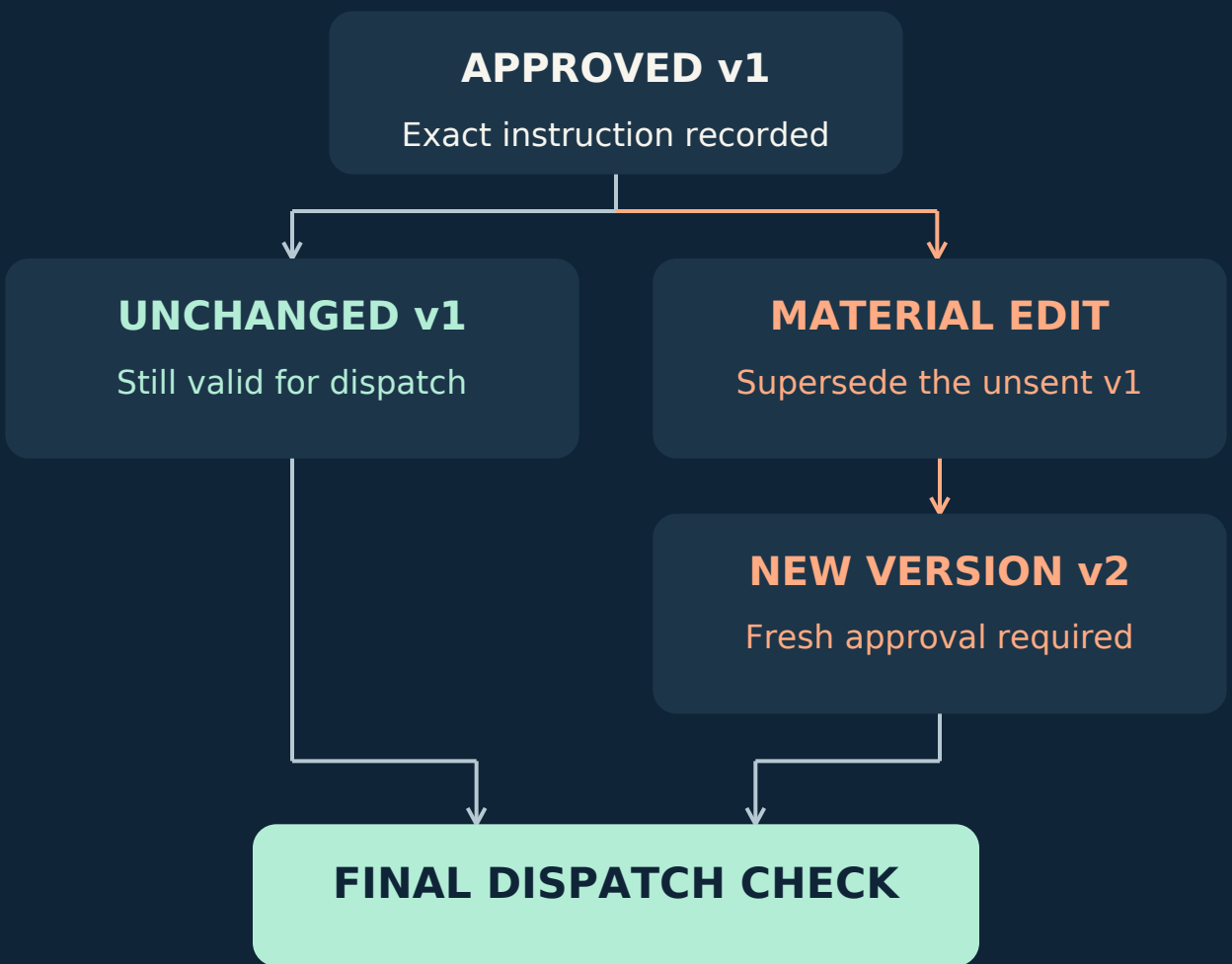
Funding source
Approved USD funding account

Commercial terms
FX quote, fees and expiry, if applicable

Payout method
Bank transfer / approved corridor

Store the approver, timestamp and policy version.
A profile edit must never silently rewrite this snapshot.

A material change starts a new approval.



Already dispatched? Resolve the in-flight instruction through the provider process; never edit it in place.

Close the gap between the check and the send.

APPROVED INSTRUCTION v1

↓

ONE DATABASE TRANSACTION

01 **Validate**

Version, authority, expiry and policy still valid.

02 **Claim**

Claim this version atomically; reject later edits.

03 **Record**

Commit the state and fixed request to an outbox.

↓

WORKER SENDS THE FIXED REQUEST

A failed gate blocks dispatch. It must not become an override.

Keep the evidence with the payment.

APPROVAL

Who approved which version, when, under which policy.

↓

DISPATCH

The exact request sent, its attempt ID and the provider reference.

↓

OUTCOME

Provider evidence and the corresponding ledger and reconciliation references.

Outbox delivery can repeat. Use provider-supported idempotency and a stable request reference. Reconcile unknown outcomes before reissuing. Protect audit records against alteration.

Six tests for your next release review.

TEST CASE	EXPECTED RESULT
Destination or amount changes	Old approval cannot authorise the edit.
Maker approves own high-risk payout	Reject under the separation-of-duties policy.
Approval expires before dispatch	Hold for renewed approval.
Edit and worker run together	Only the valid version can be claimed.
Queue delivers the same job twice	Same request reference; provider deduplicates.
Provider result is unknown	Hold and reconcile before reissuing.

Run these through the API and worker paths as well as the UI.

Can you prove the payment sent is the one approved?

Before scaling transaction volume, bring engineering, treasury and operations into the same review.

Map who can change the instruction, when approval becomes invalid and what the execution worker actually sends.

Brajkishor Pandey

Fintech engineering | Cross-border payments

linkedin.com/in/brajkishorpandey

TECHNICAL REFERENCES

OWASP | Transaction authorization
OWASP | Business logic security
AWS | Transactional outbox pattern