The twelve pass-or-fail criteria a payment flow I build is accepted against

Version 1 · October 2026 · Faruk Kurt, independent payments architect, Amsterdam · farukkurt.nl/acceptance

I publish these criteria so that anybody buying work like this can hand them to any supplier, and know at the end whether they got a working system or a demonstration.

- **State** What is true before the test starts.
- **Input** What goes into the system.
- **Action** What a named person on the buyer's side does.
- **Result** What that person sees for themselves.
- **Evidence** What the buyer keeps afterwards.

1. Reproducible build

The release builds from the buyer's own code, with nothing from the supplier in it.

- **State** A clean machine with only the documented tools installed.
- **Input** The buyer's repository at the release commit.
- **Action** The person the proposal names on the buyer's side clones the repository and runs the documented build command.
- **Result** The build completes and produces the named release.
- **Evidence** The build log, the release commit hash and the name of the person who ran it.

2. Independent run

The release starts in staging and reports itself ready, without the supplier.

- **State** The buyer's staging, set up from the documented configuration.
- **Input** The release built in the first criterion.
- **Action** The person the proposal names on the buyer's side starts the system and runs its documented health check.
- **Result** The system starts and the health check reports ready.
- **Evidence** The deployment log and the timestamped health-check output.

3. State and transition coverage

Every transition in the signed payment flow is tested, or recorded as unreachable with the condition that makes it so.

- **State** The payment flow's signed state model and an empty transition register.
- **Input** The list of every state and transition.
- **Action** The person the proposal names on the buyer's side tests every transition and records each unreachable one with the condition that makes it so.
- **Result** Every transition is marked tested or unreachable, and none is left unmarked.
- **Evidence** The completed transition register, with a test reference or the reason it is unreachable on every row.

4. Decision record on completion

A completed transaction records what was decided, on what evidence, under which rule and when.

- **State** A transaction that meets the signed conditions for completion.
- **Input** That transaction.
- **Action** The person the proposal names on the buyer's side runs the transaction to completion and opens its decision record.
- **Result** The record shows the decision, the evidence used, the rule identifier and the decision time.
- **Evidence** The decision row, with transaction reference, rule identifier and timestamp.

5. Decision record on refusal

A refused transaction records the refusal, its reason and the next action a person can take.

- **State** A transaction that meets one signed refusal condition.
- **Input** That transaction.
- **Action** The person the proposal names on the buyer's side runs the transaction and opens its decision record.
- **Result** The system refuses the transaction, and the record shows the reason, the rule used and the named next action for a person.
- **Evidence** The refusal row, with transaction reference, rule identifier, timestamp and next action.

6. Boundary failure behaviour

Each named failure at an external boundary produces the one behaviour signed for it.

- **State** One named external boundary, its counterparty stubbed and its four failure behaviours written into the signed sheet.
- **Input** A timeout, a malformed response, a duplicate response and an unavailable response, presented one at a time.
- **Action** The person the proposal names on the buyer's side drives each input through the boundary.
- **Result** Each input produces its signed state transition, retry, refusal or operator route, with no unrecorded effect.
- **Evidence** Four logged outcomes, each linked to its input, resulting state and rule identifier.

7. Idempotence

The same inbound event delivered more than once moves the money once.

- **State** A payment flow that has not yet received the named event.
- **Input** Two deliveries of the same inbound event, with the same event identifier.
- **Action** The person the proposal names on the buyer's side delivers both.
- **Result** The system records both deliveries and produces one money effect.
- **Evidence** Both delivery records, the deduplication decision and the resulting effect count.

8. Reproducible amounts

Any completed amount can be recomputed to the cent from the retained record alone.

- **State** A completed transaction with its computed amount and retained decision record.
- **Input** The retained record, without re-running the system.
- **Action** The person the proposal names on the buyer's side recomputes the amount from the recorded inputs and rules.
- **Result** The recomputed amount equals the original to the cent.
- **Evidence** The original amount, the recorded inputs, the rule version and the recomputation.

9. Data boundary

Only approved data crosses each named boundary, and every identifier in it can be traced to where it came from.

- **State** One named boundary, with its recipient and approved data classes recorded in the signed sheet.
- **Input** A transaction that crosses that boundary.

- **Action** The person the proposal names on the buyer's side runs the transaction and captures the payload that actually crosses.
- **Result** The payload contains only approved data, and the record shows where every identifier came from and what it was derived from.
- **Evidence** The captured payload, the approved data-class list and the identifier-provenance record.

10. Operability by a stranger

An operator who didn't build the system completes every named operating task from the manual alone.

- **State** The staging system ready, and a named operator who didn't build it, holding only the manual.
- **Input** The signed list of operating tasks.
- **Action** The operator completes each task from the manual, with no instruction from the supplier.
- **Result** Every task reaches its documented end state, and the operator can find the evidence it produces.
- **Evidence** The completed task list, the referenced system records and the operator's signature.

11. Failure rehearsal

An operator takes a named failure from detection to documented recovery with the runbook and nothing else.

- **State** The staging system running, and the runbook entry for one named failure.
- **Input** That failure, injected into the payment flow.
- **Action** An operator who didn't build the system follows the runbook entry from detection to recovery.
- **Result** The system reaches the documented recovery state, and the operator can say what became of every affected transaction.
- **Evidence** The rehearsal log, the runbook entry used, the resulting state and the operator's signature.

12. Reversibility

The prior software version can be restored without duplicating any effect, and every transaction already executed or still in flight ends reconciled or forward-corrected.

- **State** The new version running in staging, with transactions both executed and still in flight.

- **Input** The signed rollback runbook.
- **Action** Whoever will run the system follows the runbook and restores the prior version.
- **Result** The prior version runs, no effect is duplicated, and every executed or in-flight transaction is reconciled or forward-corrected under the named procedure.
- **Evidence** The rollback log, the version identifiers and the reconciliation or forward-correction record for every affected transaction.

	For the supplier	**For the buyer**
Name		
Role		
Date		
Signature		