

Systems Architecture SOP

Backend automation build standard · sample document · Ruben John Mendoza · Backend Automation Engineer

SOP-ARCH-001 · v1.0 · status: sample for review

1. Purpose

This SOP defines the sequence every backend automation build follows, from intake through verified delivery, together with the evidence required at each stage. It exists so that a system can be handed over, audited or rebuilt without relying on the memory of the person who built it.

2. Scope

- Lead capture, qualification and follow-up systems.
- Order-to-invoice and document-generation pipelines.
- CRM routing, booking and notification workflows.
- Data reconciliation, exception handling and reporting automation.

3. Build sequence

Stage	Action	Required evidence
01 Intake	Capture the payload at a single entry point (webhook).	Sample payload committed alongside the build.
02 Normalise	Map source fields to the internal schema. No field renamed silently.	Field mapping table with source and target names.
03 Resolve	Resolve or create the account or contact before any downstream write.	Lookup result and the branch taken, logged.
04 Execute	Perform the branch action: create invoice, dispatch message, route lead.	Execution log with timestamps.
05 Verify	Assert on totals, references and resulting downstream state.	Assertion output, including the values compared.
06 Log	Persist a run record with duration and manual step count.	Run record retained for the reporting cycle.

4. Naming and versioning

- Scenario names follow `domain.entity.action`, for example `orders.invoice.create`.
- Versions increment as `vMAJOR.MINOR`. A MINOR change cannot alter output shape.
- Identifiers contain no spaces. Field names are lower snake case.
- Every change adds one line to the changelog stating what moved and why.

5. Error handling and retry policy

Failure class	Retries	Backoff	Escalation
Upstream 5xx	3	2s / 8s / 32s	Alert after final attempt
Malformed payload	0	n/a	Quarantine, daily review
Duplicate event	0	n/a	Idempotency key discarded, counted
Auth failure	0	n/a	Immediate alert, build halted

6. Reconciliation and reporting handoff

Records are matched on the account identifier first, with a normalised name and address fallback so records still join when source systems format the same account differently. Every mismatch is classified rather than skipped, so the review queue is actionable.

missing in one source · conflicting value · duplicate account · orphan record

Reporting reads only from the reconciled dataset, never from raw extracts. Re-running the pipeline against the same input produces the same output, so a rerun is always safe.

7. Delivery checklist

- [] Sample payload attached to the build record.
- [] Field mapping documented at stage 02.
- [] Contact or account resolution branch tested with both outcomes.
- [] At least one failure path exercised, with the retry or quarantine behaviour observed.
- [] Run log captured and retained.
- [] Totals reconciled against the source before reporting runs on the output.
- [] Handover note written for whoever operates the system next.

Appendix A — reference architecture

```
source (order, lead, record extract)
|
v
webhook — single entry point
|
v
normalise --> validate
|
v
resolve contact / account
|
+--> existing contact --> create invoice
|
+--> new contact -----> create contact --> invoice
|
+--> no match -----> exception queue
|
v
verify --> log --> report
```