INVESTIGATION

Two libraries, one checked order-release workflow

WRITTEN AGAINST	events.audit · 12 record(s) · 13:50:06.389->00:00:01.000
AUTHORED VIEW	all event types
LOG	events.audit - loaded snapshot; log unchanged-metadata; graph unchanged-met...
CREATED	2026-11-01T14:18:25.689892Z
RENDERED	2026-11-01 14:18:25.700

NARRATIVE - THE AUTHOR'S ACCOUNT, NOT LOG EVIDENCE

Current synthetic workflow, not a customer integration or before/after bug fix. Codex inspected actual captured records through an isolated Analyser REST socket. Native run.py/WorkflowCheck executed the application; Analyser does not replay it.

NARRATIVE - THE AUTHOR'S ACCOUNT, NOT LOG EVIDENCE

Seven processing nodes coordinate value, stock, limit and risk checks from two separately packaged synthetic libraries. The graph additionally shows the OrderRequest event vertex. Seven native scenarios pass: normal and equality accepted; over-limit, stock shortage and combined refusal rejected with defined limit-first priority; one deliberate risk exception emits no output; the next valid request succeeds. Six outputs total. No prior defect or corrected revision is claimed.

Record #4

```
event: over_limit quantity=10 price=1200 limit=10000 stock=10 riskUnavailable=false
1. input  thread=main  method=onOrder  annotation=OnEventHandler  id=over_limit  quantity=10
2. risk  thread=main  method=check  annotation=OnTrigger  riskStatus=CLEARED
3. stock  thread=main  method=check  annotation=OnTrigger  stockAvailable=true  stock=10
4. value  thread=main  method=calculate  annotation=OnTrigger  valuePence=12000
5. limit  thread=main  method=check  annotation=OnTrigger  withinLimit=false  limitPence=10000
6. decision  thread=main  method=decide  annotation=OnTrigger  outcome=REJECTED_LIMIT
7. recorder  thread=main  method=record  annotation=OnTrigger  recordedOutcome=REJECTED_LIMIT
outputCount=2
```

NARRATIVE - THE AUTHOR'S ACCOUNT, NOT LOG EVIDENCE

Record4 shows12000p value against10000p limit, withinLimit=false, REJECTED_LIMIT, outputCount2. Record5 independently shows stockAvailable=false and REJECTED_STOCK. Record6 tests defined limit-first priority when both checks fail.

Record #8

```
event: risk_unavailable quantity=5 price=1200 limit=10000 stock=5 riskUnavailable=true
1. input  thread=main  method=onOrder  annotation=OnEventHandler  id=risk_unavailable  quantity=5
2. risk  thread=main  method=check  annotation=OnTrigger  riskStatus=UNAVAILABLE
```

NARRATIVE - THE AUTHOR'S ACCOUNT, NOT LOG EVIDENCE

Record8 logs riskStatus=UNAVAILABLE and no decision/recorder entry. The native harness separately catches the intentional exception and asserts outputCount remains5; silence alone does not prove non-execution. Record9 shows riskStatus=CLEARED, ACCEPTED and outputCount6. No transaction rollback or arbitrary network recovery is established.

Record #9

```
event: recovery quantity=5 price=1200 limit=10000 stock=5 riskUnavailable=false
1. input  thread=main  method=onOrder  annotation=OnEventHandler  id=recovery  quantity=5
2. risk  thread=main  method=check  annotation=OnTrigger  riskStatus=CLEARED
3. stock  thread=main  method=check  annotation=OnTrigger  stockAvailable=true  stock=5
4. value  thread=main  method=calculate  annotation=OnTrigger  valuePence=6000
5. limit  thread=main  method=check  annotation=OnTrigger  withinLimit=true  limitPence=10000
6. decision  thread=main  method=decide  annotation=OnTrigger  outcome=ACCEPTED
7. recorder  thread=main  method=record  annotation=OnTrigger  recordedOutcome=ACCEPTED
outputCount=6
```

NARRATIVE - THE AUTHOR'S ACCOUNT, NOT LOG EVIDENCE

Ordinary unsigned audit, stream completeness unknown. Clock is changed to1000ms after init: time-order warnings are expected; inspect by record index. Source/run relationship is unverified. Bundle carries no source/replay; download source separately and rerun native harness. No vendor compatibility, measured authoring saving, human approval or general correctness claim.