

Your sales-order batch, saved as a verified work order.

Your existing flow already retrieves sales orders and inventory. I finish the return trip: a work order saved in SOS, verified after creation, with a clear recovery path when a request fails.

USD 790 fixed Five calendar days · Delivery 17 October

1 · Read & group

One agreed batch rule, existing sales orders, valid item references.

2 · Create & verify

Write the work order, retrieve it, compare the saved lines.

3 · Recover

Persistent batch ledger, failed/uncertain branch and operator guide.

The result you receive

- A Make scenario using your existing SOS connection, one batch rule and one work-order creation route.
- Item quantities aggregated by the agreed rule; references checked before writeback.
- A stored batch key so replaying a run does not silently create another work order.
- A readback check, an error/review branch, the blueprint and operating notes.
- One correction round and 14 days of support for conformity issues in this delivered workflow.

12 October: start and validate the mapping in your sandbox.

14 October: first verified batch for your review.

17 October: delivery, acceptance record and recovery guide.

Delivered to this scope, or a full refund of my fee.

Reply **OK** and I start Monday 12 October at 9:00 Paris time.

This produces a work order; manufacturing/build transactions and inventory consumption are separate. The result is one defined workflow, not the other future automations. Your existing software subscriptions remain yours.

One batch. One saved result. A replay stays a replay.

19

passed checks

16

concurrent replays

1

work order for the original batch

Two illustrative sales orders contain the same item with quantities 2 and 1.5. The saved work order contains 3.5 of that item, plus 3 of the second item. The connector then retrieves the created object and compares the lines.

```
{
  "number": "W0-7e9f2861bf81",
  "customer": {
    "id": 10
  },
  "lines": [
    {
      "item": {
        "id": 101
      },
      "quantity": 3.5
    },
    {
      "item": {
        "id": 102
      },
      "quantity": 3.0
    }
  ]
}
```

Cases that pass

- Exact decimal aggregation.
- Creation without inherited transaction/line IDs.
- Reordered duplicate notifications.
- Stored state reopened before replay.

Cases that stop

- Missing item or source order.
- Invalid quantity or unapproved grouping.
- 401 authorization failure.
- Uncertain response or mismatched saved lines.

Evidence boundary: I executed real local HTTP requests against a personal simulator based on the SOS v2 work-order contract. This is not a live SOS or customer-account execution. The Make connection, actual SOS authorization and sandbox acceptance are the delivery steps. No customer inventory was read or changed. An uncertain write is placed in review and reconciled by exact work-order number; it is never blindly reposted.