**Claude Breakdown: Chunk 3 — Action Execution, System State, and Outcome Logging**

**✅ Scope**

Define how ClubOS executes system actions (e.g. reset TrackMan, unlock door), how it confirms outcomes via system APIs (NinjaOne, Ubiquiti, etc.), and how all results are logged and scored for audit and learning.

**🔁 Cross-Referenced Corrections**

| **Early Plan** | **Final Decision** |
| --- | --- |
| Retry/reset logic was linear | ✅ Retry logic is governed by config per category |
| Rollback support discussed | ✅ No AI rollback — flag only, manual operator correction |
| Logs were unstructured | ✅ Outcome logs are fully structured + versioned |

**🔧 Dependencies**

* System API access:
  + NinjaOne: PC status, uptime, remote reset
  + Ubiquiti: door logs, connection attempts
  + TrackMan: local ping or shell tool (custom handler)
* OpenPhone: message confirmations
* Audit logging DB

**📋 Tasks**

**[3.1] Action Execution Wrapper**

All AI-triggered actions run through a central function:

performAction({

action: "reset\_trackman",

target: "bay\_3",

triggered\_by: "OperatorGPT",

reason: "trackman\_freeze\_sop\_12",

booking\_id: "bk-390128"

})

* Calls associated tool/API
* Waits for system confirmation (e.g. PC online, logs cleared)
* Timeout if no state change in X seconds
* Sets action\_status: success | failed | inconclusive

**[3.2] Structured Outcome Logging**

Every action creates an entry:

{

"action": "reset\_trackman",

"bay": "3",

"location": "Bedford",

"result": "success",

"followed\_sop": "trackman\_freeze\_sop\_12",

"trigger\_source": "OperatorGPT",

"human\_override": false,

"linked\_thread": "msg\_812",

"timestamp": "2025-07-31T18:35:20Z"

}

**[3.3] Failure → Escalation Trigger**

If:

* Action fails
* OR no customer confirmation
* OR system state unchanged

Then:

* Thread escalates to operator
* Operator sees full action log + error state
* Claude flags any potential SOP mismatches for later review

**[3.4] No Rollbacks — Only Manual Correction**

If AI makes an incorrect destructive action (e.g. unlocks wrong door):

* Flag added: "rollback\_needed"
* Operator notified in Slack + dashboard
* Claude proposes SOP revision path only — **does not undo**

**[3.5] Action Chain Templates (Optional Extension)**

Later:

* Claude can store chains like:

{

"scenario": "no\_booking\_found",

"actions": ["check\_phone\_variants", "check\_bay\_overlap", "escalate"]

}

But for V2, action logic is one-step at a time + human fallback.

**✅ Expected Behavior**

* All actions pass through a single router
* System state is always checked post-action
* Results are logged in structured format for audit + learning
* Operator has full visibility into what was attempted and why
* Claude can later suggest SOP improvements based on outcome history

proceed to **Chunk 4: SOP Structure, Storage, and Update Workflow**