RBC – Watchdog (Producers ran today?) v1.0

Prepared by: ChatGPT (GPT-5 Thinking)
Issue Date: 01 November 2025
Design Framework: DAI + RBC Framework v5.85 (October 2025)
Change Reference: Documentation – Detailed Design Creation (No behavioural change).

1. Functional Overview

Nightly watchdog at 23:59 that verifies each RBC domain's last-period key (PV Daily, Demand 08–16, Demand Daily). If any producer did not stamp today's date, it raises an alert: writes a structured logbook message and sends both a mobile and a persistent notification containing quick guidance and the observed last-period values. If all stamped, it logs 'all OK'.

2. Entity Map

Entity ID / Var	Purpose	Direction
input_text.rbc_last_period_pv	PV Daily producer last processed period key (YYYY-MM-DD)	Read
input_text.rbc_last_period_demand	Demand 08–16 producer last period key	Read
input_text.rbc_last_period_demand_daily	Demand Daily producer last period key	Read
notify.mobile_app_pixel9pro	Mobile notification target	Send
persistent_notification.create	Persistent notification channel	Send
today_key = now().date()	Today's date used for comparison	Compute
failed_list	Comma-separated list of domains that missed today	Compute

3. Trigger Matrix

Trigger ID	Condition	Purpose
t_2359	Time = "23:59:00"	Perform end-of-day verification of producer stamps

4. Logic Flow

1. Compute today_key = now().date().

2. Read lp_pv, lp_d0816, lp_dday from their respective input_text helpers.

3. Compute ok_pv = (lp_pv == today_key), ok_d0816 = (lp_d0816 == today_key), ok_dday = (lp_dday == today_key).

4. Assemble failed_list from any domains where ok_* is false; set has_fail accordingly.

5. If has_fail:

6. • logbook.log → ALERT with failed_list and each last-period value;

7. • notify.mobile_app_pixel9pro → concise guidance to check inputs and run producer(s) manually;

8. • persistent_notification.create → on-screen alert with same summary.

9. Else:

10. • logbook.log → 'All producers stamped today' with each last-period value.

5. Guards and Safety

• No inverter writes; read and notify only.

• End-of-day timing ensures all producer windows should have completed.

• `mode: single` avoids overlapping executions around 23:59.

• Structured logging uses logbook.log; notifications provide both mobile and persistent channels.

6. Scheduling

Single daily trigger at 23:59, after all producers (PV Daily ~23:40; Demand Daily ~23:40; Safety Nets at 23:55) have had an opportunity to run.

7. Acceptance Tests

Test ID	Scenario	Expected Result
T1	All stamps equal today_key	Logs 'All producers stamped' with values; no notifications.
T2	One domain missed	Alert path fires with that domain in failed_list; both notifications sent.
T3	Multiple domains missed	Alert lists multiple domains; both notifications sent.
T4	Helpers contain stale/invalid strings	String comparison identifies mismatch; alert sent.

8. Version Lineage & Governance

v1.0 – Nightly watchdog confirms that PV Daily, Demand 08–16, and Demand Daily producers stamped today's period; alerts via logbook, mobile, and persistent notifications if any missed.

Governance: Protected Architecture Mode v5.9; Two-Phase Code-Change Gate v2.2. This document records the live YAML behaviour; no changes introduced.

9. YAML Reference (Read-Only Excerpt)

trigger: 23:59; vars: today_key, lp_pv, lp_d0816, lp_dday → ok_* → failed_list
if has_fail: logbook + notify.mobile_app_pixel9pro + persistent_notification.create; else: log OK.

10. Compliance Checklist

✅ Entity ↔ ID alignment confirmed: rbc_last_period_pv, rbc_last_period_demand, rbc_last_period_demand_daily.

✅ Trigger has ID and Visual-Editor-safe structure.

✅ No inverter writes; Master-only rule unaffected.

✅ Dual notification paths included (mobile + persistent).

✅ End-of-day schedule compatible with producer/safety-net timing.