DAI/RBC High-Level Design – Framework Overview

Version: 1.2 (Live)

Date 2025-11-03
Maintainer: Simon Angell

Advisory Role: ChatGPT-5

Storage Path: C:\Users\simon\OneDrive\ChatGPT\Projects\DAI\Project Specific Governance Docs

Header Metadata Table

Field	**Value**
Document Title	DAI + RBC High-Level Design – Framework Overview
Document Purpose	Defines the architecture, control layers, and data-flow relationships between all DAI and RBC components.
Applies To	DAI + RBC Home Assistant architecture operating under the ChatGPT Interaction Guidelines v1.0 (Live).
Governance Context	Protected Architecture Mode v5.9 (active).
Status	Live – Implemented within the Live environment.
Supersedes	All previous ad-hoc YAML and automation standards embedded in project instructions.
Linked Governance Document	*ChatGPT Interaction Guidelines v1.1 (Live); DAI/RBC Technical Coding and Deployment Framework v1.0 (Live)*

1. Table of Contents

2. Purpose and Scope

This document defines how the DAI (Dynamic AI Intelligence) and RBC (Robust Bias Correction) frameworks integrate within the Home Assistant environment.
It establishes the design intent, safety philosophy, and data-flow model that underpin the DAI + RBC automation family.

The High-Level Design provides the conceptual view of control layers and interfaces — complementing the technical detail contained in the *DAI / RBC Technical Coding and Deployment Framework*.

3. Supported Environment

The design aims to deliver a self-optimising, audit-safe energy-automation system that minimises grid import, maximises solar utilisation, and preserves battery longevity. Every architectural decision follows five core principles: predictability, transparency, modularity, traceability, and resilience.

All code, data, and learning models must support deterministic operation under governance control, with clear separation between functional logic and learning bias layers.

4. Architectural Overview

4.1 System Context and Boundaries

The DAI + RBC system operates as a logical extension of Home Assistant, using HA as the orchestration platform and execution host.
It interfaces with inverters, sensors, and cloud APIs but performs no direct inverter control outside documented service calls.

4.2 Parallel Inverter Architecture

The physical installation consists of two SolaX X1 Hybrid inverters operating in a parallel configuration.

Within this topology, **the “House” inverter acts as the Master control node** and **the “Garage” inverter operates as a passive Slave**.

All write-based control commands, mode changes, and target-SoC updates are directed exclusively to the Master (solax_house_* entities).

The Slave (solax_garage_*) provides sensor and telemetry data only.
This rule ensures single-source authority, idempotent writes, and full compliance with Protected Architecture Mode v5.9.

4.3 Control Layer Hierarchy (DAI → RBC → HA Core)

- **DAI Layer** – planning, scheduling, and decision automation.
- **RBC Layer** – continuous learning, bias correction, and safety enforcement.
- **HA Core** – execution engine for validated commands and triggers.

4.4 Data Flow and Integration Topology

Forecasts, sensor data, and tariff inputs feed into RBC bias engines, which adjust daily demand and solar estimates.
These flow upward to DAI planners, which generate operational targets written to HA helpers and automations.

4.5 External Interfaces

Integrations include Solcast (PV forecasts), Octopus Energy (tariffs, export/import data, and Calendar event sessions for Free Energy and dynamic tariff windows), and local weather sensors.

Each interface is sandboxed via entity abstraction and validated against its data schema before use.

4.6 Blueprint Script Architecture and Detailed Design References

The **DAI + RBC architecture** uses a set of reusable **script blueprints** to encapsulate all mathematical learning, bias correction, and safe-write logic.
Each blueprint acts as a modular computational node within the broader RBC bias-learning network, exposing a stable interface for automations to consume.

RBC – Update Bias (Robust EWMA) v1.5 Blueprint

This blueprint is the **canonical bias-producer engine** for all RBC families (PV Daily, Demand 08–16, Demand Daily, Next Horizon).
It calculates the daily bias offset between *actual* and *estimated* energy totals using a **Robust Exponential Weighted Moving Average (EWMA)**:

1. **Read phase** – collects actual and estimate entity values and the current stored bias.
2. **Validation** – skips if inputs are unavailable or both zero when require_nonzero = true.
3. **Computation** – applies a clipped EWMA to *(Actual − Estimate)* using a configurable half-life ($\lambda \approx 1 - 0.5^{(1/half\_life\_days)}$).
4. **Safety enforcement** – clamps the result to ±max_abs_bias kWh and rounds to two decimals.
5. **Idempotence control** – ensures only one update per day via the stored period key (today's date).
6. **Persistence** – writes the bias to an input_number helper and the period key to an input_text helper.
7. **Audit trail** – records all updates and skips in the HA logbook with configurable verbosity.

This design guarantees deterministic, traceable bias learning across all automations that depend on adjusted demand or solar forecasts.
It supports **single-threaded execution per script entity**, while the Technical Framework mandates **unique helpers per bias family** to prevent cross-thread drift.

Blueprint Family Integration

Each major RBC or DAI automation calls one of these blueprints rather than embedding its own bias logic.

This enforces mathematical consistency and simplifies version management: any change to a bias algorithm propagates automatically through all dependent automations once the blueprint is updated.

The following Detailed Design Documents describe these blueprints in depth and are stored in:

C:\Users\simon\OneDrive\ChatGPT\Projects\DAI\Design Documents\Detailed Automation Design Documents

Document	Version	Description
DAI / RBC – Update Bias (Robust EWMA)	v1.5 (Live)	Core blueprint implementing EWMA-based bias correction.
RBC – Daily Demand Bias Producer	v1.x (Draft)	Applies the EWMA engine to daily demand data.
RBC – Next Horizon Bias Producer	v1.x (Draft)	Applies horizon-window bias learning for short-term forecasts.
RBC – PV Daily Bias Producer	v1.x (Draft)	Solar generation equivalent using identical bias mechanics.

Cross-Document Relationship

The blueprint tier provides the **"mathematical kernel"** of the system, while:

- The **DAI / RBC Technical Coding and Deployment Framework v1.0 (Live)** enforces implementation standards, concurrency guards, and Visual-Editor safety.
- The **High-Level Design** (this document) defines the architectural purpose and data-flow context.
- The **Detailed Design Documents** (listed above) hold the functional specifications and acceptance tests for each blueprint.

Together, these three layers form the complete traceable design chain for the DAI + RBC learning subsystem.

Note:

See 5.2 for canonical RBC family structure, naming rules, and domain-specific examples.

5. Core Components and Roles

5.1 Dynamic AI Intelligence (DAI) Layer

Responsible for scenario planning, grid-charge scheduling, and automated mode selection.
It consumes RBC-adjusted forecasts and produces daily and horizon-level plans.

5.2 Robust Bias Correction (RBC) Layer

Implements safety-first learning using upward-only bias correction (v2.4 model).
Each bias producer operates on a defined window (PV Daily, Demand 08→16, Demand Daily, Next Horizon).

5.2.1 RBC – Canonical Pattern (applies to every RBC family)

Purpose. All RBC domains (PV Daily, Demand Daily, and sub-day windows such as 16–22) must share the same structural roles and safeguards while keeping timings appropriate to their data cutoffs. This pattern is proven by **Demand Daily** and is adopted as the reference.

Core Roles (mandatory):

1. **RBC – <Domain> Bias Producer vX.Y** — Runs at the domain's cutoff and calls the shared **RBC Update Bias (Robust EWMA) blueprint** to write the **bias (kWh)** and stamp the **last_period** helper (idempotence).
2. **RBC – <Domain> Bias Safety Net vX.Y** — Runs **+15 min** after the Producer; only acts if today is **not** stamped; performs the same write and logs/alerts. (Daily & PV already implement this; Evening will be added.)
3. **RBC – <Domain> Adjusted Updater vX.Y** *(kWh domains only)* — Maintains **Adjusted = max(Planned, PrevAdjusted)** during the live window; **locks** at the domain's end to **max(Planned, Actual)** (Safety-First model).
4. **RBC – Watchdog (Producers ran today?) vX.Y** — Audits that each domain's Producer (and Safety Net if needed) executed; warns on any miss.

Sub-day extensions (when the domain is a time slice, e.g., 16–22):

- **DAI – <Window> Actual Switcher vX.Y** to select the correct "Actuals" bucket during the window. (Evening has this: *Actual Demand 16–22 Tariff Switcher v1.2*.)
- **DAI – <Window> Forecast Writer vX.Y** to persist the forecast slice the learner consumes. (Evening has this: *Forecast 16–22 Writer v1.0*.)

- **RBC – <Domain> Bias→Percent Mapper vX.Y (script blueprint)** when a **% output** is required by downstream controllers. (Missing for Evening today; to be added.)

Timing rule. Each domain keeps its own **cutoff** and **Safety Net** times (e.g., Daily at 23:40/23:55). Alignment is **structural** (roles, stamping, safeguards), not forced clock equality.

A fourth RBC learning domain has been introduced to support cold-day planning logic. The **Temperature Bias Producer** operates nightly at 23:40 using the shared *RBC Update Bias (v1.5)* blueprint, learning the persistent deviation between the **forecast minimum** (from Met Office Monmouth) and the **actual measured minimum** (Statistics helper). Bias is EWMA-smoothed (half-life 4 days) and clamped to ±5 °C.

Outputs:

Signal	Entity	Description
Forecast Low (°C)	input_number.rbc_temperature_estimate_low_c	Written nightly before bias calculation.
Actual Min (°C)	sensor.rbc_outside_temperature_today_min_degc	Derived via Statistics helper (min of outside temperature).
Bias (°C)	input_number.rbc_temp_bias_c	RBC-maintained bias value.
Last Period	input_text.rbc_last_period_temperature	Date stamp for update completeness.

A dedicated **Safety Net** runs at 23:55 to recover from missed executions and to ensure bias continuity.
The **Watchdog** now includes the Temperature domain stamp to guarantee daily coverage across all producers.

5.2.2 RBC – Demand Daily (Reference Implementation)

Why this is canonical. Demand Daily already implements the full pattern: **Producer @ 23:40**, **Safety Net @ 23:55**, **Adjusted live/lock**, and **Watchdog** coverage. This serves as the template for all RBC families.

Inputs/Outputs:
Reads: Daily actual (sensor.brenchley_load_today) and estimate (input_number.dai_expected_usage_daily_kwh).

Writes: input_number.rbc_bias_demand_daily, input_text.rbc_last_period_demand_daily, and input_number.rbc_adjusted_demand_daily_kwh.

5.2.3 RBC – Evening Buffer (16–22) — To Align to Canonical

Current: Window prep exists (**Actual Switcher**, **Forecast Writer**); **Bias Producer** exists; **Mapper** is missing (orchestrator references a non-existent script); **Safety Net** not present.
Target (per canonical): Add **RBC – Evening Bias Safety Net (23:55)** and **RBC – Evening Bias→Percent Mapper** (script blueprint + instance). Orchestrator calls Mapper at the required time (e.g., ~12:55) to publish input_number.dai_evening_buffer_auto_suggest_pct. Controller remains DAI and writes Master-only at tariff gates.

5.2.4 DAI Cold Day Classifier (New Sub-section)

The **DAI Cold Day Classifier** (v1.0) determines whether the current day should be treated as a *Cold Day* for grid-charge planning.
It executes at 06:10 each day (or on HA start after 06:00) to calculate the **Adjusted Minimum Temperature**:

$$\text{Adjusted Min (°C)} = \text{Forecast Low (°C)} + \text{RBC Temperature Bias (°C)}$$

Decision logic:

- If Adjusted Min ≤ input_number.dai_cold_temp_threshold_c → input_boolean.dai_cold_day_flag = on
- Else → Flag = off

Classifier writes to:
input_number.rbc_adjusted_min_temp_c, input_boolean.dai_cold_day_flag, and input_text.dai_cold_day_reason.
No inverter writes are performed. Output is consumed by the Grid Charge Controller.

5.3 Helper and Sensor Subsystem

All dynamic data are stored in helpers (`input_number`, `input_text`, etc.).
Sensors provide real-time state inputs; helpers hold planned or learned values.
Entity naming follows the Technical Framework §3.3 rules.

5.4 Automation Families

Major automation groups: Grid Charge Planner, Cosy Night Charge, Free Energy Session Controller, Minimise PV Clipping, Evening Buffer Control, Bias Producers.
Each automation aligns with a design-doc section and a governed workflow type (New Code / Edit Code / Documentation Change).

5.5 Data Persistence and Version Rotation

All YAML, logs, and helpers follow the three-version rotation model.
Historical versions are archived in OneDrive / Archive for six-month retention.

5.6 RBC Automation Interaction Flow (Domain Overview)

This section captures the per-domain runtime choreography (triggers → script calls → helper writes → controller consumption). It complements § 5.2 (RBC Layer) and the blueprint description, and is authoritative for hand-offs and stamping.

5.6.1 RBC PV Daily — Interaction & Data-Flow

Purpose. Learn daily PV bias (kWh) with EWMA; maintain Adjusted PV Daily for DAI controllers.

Timeline (UTC/local):

- **06:05** — Adjusted PV Updater: **Adjusted = max(Planned, PrevAdjusted)** (live); locks 23:59 to **max(Planned, Actual)**.
- **23:40** — **RBC – PV Bias Producer** calls **RBC Update Bias** blueprint → writes rbc_bias_pv_daily and stamps period key.
- **23:55** — **RBC – PV Bias Safety Net**: re-runs if no stamp today.
- **23:59** — **RBC – Watchdog** verifies Producers ran.

Hand-offs & guards.
Reads PV actual + forecast; writes bias helper + last_period_pv; Adjusted helper exposed to DAI controllers. RBC never writes inverters; DAI controllers do, Master-only.

5.6.3 RBC Demand Daily (Canonical Pattern) — Interaction & Data-Flow

Purpose. Learn daily demand bias (kWh) via blueprint; maintain Adjusted Demand Daily. **This is the reference pattern for all RBC families.**

Timeline (UTC/local):

- **All day** — Actual demand accumulates (daily sensor).
- **06:05** — Adjusted Demand Updater: **live/lock** per Safety-First v2.4.

- **23:40 — RBC – Demand Daily Bias Producer v1.3** calls **RBC Update Bias** blueprint → writes rbc_bias_demand_daily + stamps last_period_demand_daily.
- **23:55 — RBC – Demand Daily Safety Net v1.1** runs if not stamped; logs + notify.
- **23:59 — RBC – Watchdog** audits Producer/Safety Net.

Hand-offs & guards.
Reads daily actual + estimate; writes bias + adjusted helpers; stamped idempotence. RBC isolation from inverter writes; DAI controllers consume Adjusted values.

5.6.3 RBC Evening Buffer (16–22) — Interaction & Data-Flow

Purpose
Implements the RBC bias-learning pattern for the 16–22 window, producing a learned Evening Bias (kWh) which is converted to an auto-suggested buffer % used by the DAI Grid Charge Controller.
This section supersedes all earlier versions of § 5.6.3 and fully integrates the corrective measures defined in DR004a – RBC Alignment to Daily Demand Model.

Functional Roles and Schedules

Role	Entity	Schedule	Function
Bias Producer	automation.rbc_evening_bias_producer v1.0	22:10 daily	Computes Evening Bias (kWh) via script.rbc_update_bias_evening; writes to input_number.rbc_evening_bias_kwh and stamps input_text.rbc_last_period_evening_demand.
Bias Safety Net (Defect Closure)	automation.rbc_evening_bias_safety_net_v1_0	22:25 daily (+15 min)	Runs only if today's Evening stamp is missing; repeats Producer blueprint call; logs *Safety Net triggered*.
Bias→Percent Mapper	script.rbc_evening_bias_to_percent_mapper v1.0	12:55 next day	Converts rbc_evening_bias_kwh to dai_evening_buffer_auto_sugges

Role	Entity	Schedule	Function
			t_pct using total battery capacity, reserve floor, and safety multiplier; idempotent.
Watchdog	automation.rbc_bias_watchdog v2.2	23:59 daily	Audits Producer and Safety Net stamps; notifies Pixel 9 Pro on miss.
Controller Consumer	automation.dai_grid_charge_controller v6.3	Tariff windows	Consumes Auto-Suggest % if input_boolean.dai_evening_buffer_auto = ON; otherwise uses Manual %; Master-only writes at gate cadence.

Helper Entities

- input_number.rbc_evening_bias_kwh — Learned Evening Bias (kWh)
- input_text.rbc_last_period_evening_demand — Daily stamp key
- input_number.dai_evening_buffer_auto_suggest_pct — Auto-suggested % (Mapper output)
- input_number.dai_evening_buffer_pct — Manual fallback %
- input_boolean.dai_evening_buffer_auto — Auto/Manual selector

---

Operational Sequence

1. Bias Learning: Producer runs at 22:10 to write bias and stamp. If missed, Safety Net executes at 22:25 to recover.
2. Mapping to Percent: Mapper converts bias → % at 12:55 the next day.
3. Audit and Notification: Watchdog verifies Producer and Safety Net daily; alerts on failure.
4. Controller Consumption: Grid Charge Controller uses Auto-Suggest % when Auto mode ON; writes only to Master inverter.

Acceptance Tests (Ref. DR004a A1–A3)

ID	Description	Expected Result
T1	Producer Run (22:10)	Bias (kWh) and stamp written; logbook entry present.
T2	Safety Net Recovery (22:25)	Executes if stamp missing; bias and stamp written; logbook shows Safety Net path.
T3	Mapper Publish (12:55)	Auto-Suggest % updated ≤ ±2 min of schedule; logbook shows conversion.
T4	Watchdog Audit (23:59)	Confirms Producer and Safety Net execution; Pixel notification on miss.
T5	Controller Consumption	When Auto ON, Grid Charge Controller uses Auto-Suggest %; Manual mode unaffected.

Governance Notes

- This section incorporates the approved changes from DR004a and forms the baseline for CR015 – Evening Safety Net and Mapper Addition.
- All future RBC Evening Bias automations and scripts must retain idempotence and Visual-Editor-safe syntax.
- No other behavioural changes are authorised to the DAI Grid Charge Controller outside this scope.

6. Governance and Safety Architecture

6.1 Protected Architecture Principles

The framework operates under Protected Architecture Mode v5.9, ensuring no unapproved structural or behavioural changes occur.
Every component references a governing document, and any modification requires an authorised CR or DR.

6.2 Safety-First Bias Model (v2.4)

The RBC Safety-First Bias Model guarantees upward-only learning.
Bias values can only increase to match observed load or generation, never decrease to optimistic levels.
This prevents under-charging events and aligns bias correction across PV and Demand models.

6.3 Command Integrity and Idempotent Write Design

All inverter commands are idempotent and issued to the **SolaX Master only**.
Duplicate or parallel writes are disallowed.
This section summarises the principle; detailed enforcement is specified in the *DAI / RBC Technical Coding and Deployment Framework §2.4 and §3.5*.

6.4 Operational Guards and Validation Concepts

System start-up behaviour is gated by HA-start guards to prevent actions before 06:00 local time.
Each automation includes runtime validation logic and post-execution logbook confirmation.
Implementation detail resides in the Technical Framework §6.

6.5 Logging and Audit Trail Architecture

All critical actions generate logbook entries with trigger ID and CR/DR reference.
Logs feed the compliance audit process defined in the Technical Framework §5 and §11.
This ensures full traceability from forecast input to inverter command.

7. Performance and Reliability Targets

Target reliability ≥ 99.8 % uptime with self-recovery after HA restart.
Forecast accuracy maintained within ±10 % of actual load/generation for rolling 7-day average.
Bias drift tolerance ≤ 0.5 kWh per day.
Energy KPIs: minimise grid import (< 15 % of demand) and maximise export revenue (≥ 95 % capture of available surplus).

8. Maintenance and Update Process

All changes flow through the Two-Phase Code-Change Gate.
Phase 1 – document review and approval; Phase 2 – implementation and post-validation.
Design documents are version-controlled with three retained versions.
Archived documents stored in OneDrive / Governance / Archive.

9. System Dependencies and Assumptions

9.1 Hardware Dependencies

SolaX X1 Hybrid inverters (Gen 4), battery bank ≈ 23 kWh, and grid-connected smart meter.

9.2 Software Dependencies

Home Assistant Core ≥ 2025.10.2, Supervisor ≥ 2025.10.0, Operating System ≥ 16.2, Frontend ≥ 20251001.2.
Add-ons – File Editor, Studio Code Server, ApexCharts, Octopus Energy, Solcast.

9.3 Environmental and External APIs

Solcast PV forecast API; Octopus Energy tariff and Calendar event API; local temperature sensor sensor.ashp_hot_water_outside_temperature

10. Future Enhancement Roadmap

10.1 Algorithm Enhancements

RBC v2.5 – introduce adaptive retention and multi-window bias averaging.
DAI v6 – integrate reinforcement planning module for in-day rescheduling.

10.2 Integration Expansion

Future interfaces include EV V2G (UK G99 compliant), ASHP Control, and external energy API bridges.

10.3 Governance Maturity Targets

Maintain CMMI Level 3 process maturity; evaluate selective adoption of Level 4 quantitative metrics when multi-developer collaboration begins.

11. Appendices

Appendix A – Glossary of Terms
Defines abbreviations (DAI, RBC, NFR, CR, DR, HA, SoC, EWMA, etc.).

Appendix B – Machine-Readable Metadata Summary

Key	Value
Document Type	DAI + RBC High-Level Design
Version	1.2 (Live)
Governance Mode	Protected Architecture v5.9 (Active)
Maintainer	Simon Angell
Advisory Role	ChatGPT-5
Linked Docs	ChatGPT Interaction Guidelines v1.1 (Live); DAI / RBC Technical Coding and Deployment Framework v1.2 (Live)

Appendix C – Change History

Version	Date	Summary
0.1	2025-11-01	Initial draft structure migrated from legacy DAI + RBC Project Instructions set.
1.0	2025-11-01	Live version implemented within ChatGPT
1.1	2025-11-02	Include the RBC structure and standard automation interaction.
1.2	2025-11-03	CR016 – RBC Min Temp introduction

End of Document – v1.2 (Live)
