

Corrected reading edition — Scientific Corrigenda and Regression Hardening v1.0. Correction date: 2026-09-05. The predecessor edition remains unchanged. This file incorporates only the corrections recorded in patches/corrections.json. Historical titles, document versions and identifiers below describe the predecessor; they do not assign its DOI to these changed bytes. This is not a complete new release of the parent compound package. See the accompanying corrigendum and source bindings.

ARQ_System_Models_and_Assumptions_v0.2

System Models and Assumptions for the ARQ v0.2 Supplement

Status: Draft / normative support document

Version: 0.2-draft

Role: Model boundary, theorem applicability, assumption discipline

Parent document: ARQ_v0.2_Normative_Core.md

Parent corpus: AGI / SER / L4 / L4 Witness / Beacon / VXCX

Author: Ivan Kotov

Place / year: Bruxelles, 2026

0. Purpose

ARQ uses the language of entropy, memory, witness, trust, budgets, correction, and—optionally—quantum correlation. Without an explicit model discipline, those terms can drift into false universality.

This document prevents that drift.

It defines the **system models and assumptions** under which ARQ claims are allowed to be stated. In particular, it distinguishes:

- finite persistent-state bounds,
- commit-metered retention bounds,
- epoch-gated trust/control bounds,
- online queue/backlog bounds,
- and fixed trusted quantum-boundary bounds.

The purpose is not to reduce ARQ to one model. The purpose is to make every ARQ statement say **which model carries it**.

1. Scope

This document specifies:

1. the admissible system models for ARQ reasoning;
2. the minimum assumptions required by each model;
3. which classes of theorem or bound each model supports;
4. what ARQ **MUST NOT** claim unless the stronger model assumptions are explicitly declared.

This document does **not** define the ARQ value function itself, the trust profile itself, or the witness schema itself. Those belong to companion documents.

2. Non-Goals

This document does **not** attempt to:

- define one universal physics of all persistent entities;
- force classical and quantum systems into the same proof shape;
- derive boundedness from rhetoric alone;
- treat Landauer's principle as a universal accounting oracle;
- infer authority or identity from state size.

This document exists to limit claims, not to decorate them.

3. Normative Keywords

The key words **MUST**, **MUST NOT**, **REQUIRED**, **SHALL**, **SHALL NOT**, **SHOULD**, **SHOULD NOT**, **MAY**, and **OPTIONAL** are to be interpreted as described in BCP 14.

4. Relationship to ARQ Core

ARQ_v0.2_Normative_Core.md defines how ARQ operates: layers, capsules, event lifecycle, budgets, promotion gates, fail-closed behavior, anti-echo rules, and authority constraints.

This document defines **what kind of system that core is being interpreted over**.

Hard rule: No theorem, upper bound, or audit claim in the ARQ supplement **SHALL** be stated without an identifiable model from this document.

5. Common Modeling Discipline

The following assumptions apply to all models unless explicitly overridden.

5.1 Declared boundary

ARQ reasoning **SHALL** occur only inside a declared operational boundary.

A boundary declaration **MUST** identify:

- what counts as internal state;
- what counts as external environment;
- where witnessability begins and ends;
- where authority to change persistent state resides.

5.2 Explicit budgets

All conformant models **MUST** operate under explicit L4 budgets, including at minimum:

- energy or work budget,
- time / latency budget,
- storage or persistent-state headroom,
- privilege budget,
- irreversibility budget.

5.3 Witnessed state change

Any state-changing event relevant to ARQ MUST be representable by a witnessable record or capsule reference.

5.4 Fail-closed admissibility

If the conditions of the active model cease to hold, the implementation MUST transition to a safe reduced-authority mode or fail-closed mode.

5.5 No hidden infinite reservoirs

A model used by ARQ MUST NOT silently assume unlimited memory, unlimited external trusted storage, unlimited correction bandwidth, or unlimited validation authority.

5.6 Anti-echo quarantine

Any proposed model refinement that merely rephrases a recent model claim without introducing a new boundary distinction, new measurable constraint, or new theorem support class SHOULD be quarantined as self-similar rather than promoted.

5.7 Safe self-edit via A/B slots

Model evolution SHOULD use two slots:

- **Slot A** = last-known-good canonical model set;
- **Slot B** = candidate refinement.

A Slot B refinement MAY replace Slot A only after:

1. contradiction check,
2. notation check,
3. bridge consistency check with parent corpus,
4. and rollback path declaration.

If any of these fail, the refinement MUST auto-rollback to Slot A.

6. Shared Symbols

The following symbols are used consistently across the models below.

- c = persistent entity.
- a = human anchor or authorized delegate.
- X_t = classical internal persistent state at time t .
- $H(X_t)$ = Shannon entropy of classical state representation.
- $\rho(t)$ = density operator of a quantum or hybrid system.
- $S(\rho)$ = von Neumann entropy.
- M_{persist} = number of bits available for authoritative persistent classical state.
- $N_{\text{commit}}(T)$ = number of successful persistent commit events up to time T .
- $e_{\text{commit},\text{min}}$ = minimum calibrated cost of one persistent commit under the chosen implementation profile.
- K = maximum pending queue length for unresolved ARQ events.
- B_{max} = maximum size of one pending event context or capsule payload used for queue accounting.
- $\mathcal{H}_S$ = system Hilbert space.
- $\mathcal{H}_A$ = trusted ancilla Hilbert space.
- $\mathcal{H}_U$ = uncontrolled environment Hilbert space.

- $d_S, d_A, d_{\{SA\}}$ = corresponding Hilbert-space dimensions.

Notation rule: Symbols used in one model SHALL NOT be silently imported into another model with a different meaning.

7. Model M1 — Classical Persistent-State Model

7.1 Purpose

Model M1 captures ARQ as a persistent classical entity with finite authoritative state.

This is the baseline model for:

- software entities,
- local memory-bearing agents,
- witness-bound controller state,
- classical recovery and rollback systems.

7.2 State definition

The authoritative internal state at time t is represented as:

$$X_t \in \{0, 1\}^{M_{persist}} \times C,$$

where:

- $M_{persist}$ is the finite number of bits available for persistent authoritative state;
- C is a finite controller-mode set, such as normal, classify, observe, candidate, degraded, fail_closed.

7.3 Assumptions

Model M1 REQUIRES:

1. finite persistent memory;
2. finite controller mode set;
3. no silent external memory treated as internal authority;
4. authoritative memory updates occur only through declared ARQ-controlled state transitions.

7.4 Supported claim class

Model M1 supports a **state-space boundedness claim**:

$$H(X_t) \leq M_{persist} + \log_2 |C|.$$

If $|C|$ is small and fixed, this is effectively:

$$H(X_t) \leq M_{persist} + O(1).$$

7.5 Interpretation

This model does **not** say how many operations the entity can execute. It says only that the authoritative internal state space is finite, and therefore its classical Shannon entropy is bounded by construction.

7.6 Applicability

Use Model M1 when the relevant ARQ claim is:

- “persistent authoritative state is bounded,”
 - “long-term memory cannot exceed declared memory authority,”
 - “classical internal disorder cannot diverge if the internal authoritative state itself is finite.”
-

8. Model M2 — Commit-Metered Retention Model

8.1 Purpose

Model M2 captures the fact that not every detected deviation becomes retained history. Retention requires a **commit event**.

This model is for statements about:

- EA creation rate,
- retained novelty,
- irreversible authoritative memory growth,
- witness-backed persistence cost.

8.2 Commit primitive

A **commit** is a state transition that writes or seals authoritative persistent information, such as:

- EA promotion,
- candidate-to-provisional transition,
- witness-chain append with retained semantic authority,
- checkpoint / rollback image creation,
- privilege-authorized state seal.

8.3 Assumptions

Model M2 REQUIRES:

1. every retained authoritative update is implemented by a declared commit primitive;
2. each successful commit consumes at least one unit from one or more bounded resources;
3. the implementation profile declares calibrated lower bounds such as:
 - $b_{\text{commit},\text{min}}$ = minimum committed bits,
 - $e_{\text{commit},\text{min}}$ = minimum calibrated commit cost,
 - $i_{\text{commit},\text{min}}$ = minimum irreversibility expenditure, if such a budget is used.

8.4 Commit-count bound

For horizon T , the number of successful commits is bounded by available resources. A generic bound is:

$$N_{\text{commit}}(T) \leq \min \left(\left\lfloor \frac{M_{\text{persist}}}{b_{\text{commit},\text{min}}} \right\rfloor, \left\lfloor \frac{E_{\text{commit}}(T)}{e_{\text{commit},\text{min}}} \right\rfloor, \left\lfloor \frac{I_{\text{max}}(T)}{i_{\text{commit},\text{min}}} \right\rfloor \right),$$

where terms not used by the profile MAY be omitted.

The accounting window is the declared interval from the initial state at 0 to T . $N_{\text{commit}}(T)$ counts commits after that initial state; a prior genesis commit MUST NOT be silently counted again. $I_{\text{max}}(T)$ is a resource budget, not a commit count, and has the same units as the strictly positive $i_{\text{commit},\text{min}}$.

Each included term in the minimum MUST bound the same full set of counted commits: every counted commit must incur that term's declared positive minimum charge. A charge applying only to a subset of commits cannot bound all commits by itself. The storage-count term is applicable only when every counted commit consumes non-recycled persistent headroom over the declared window. Omit that count term for overwriting/reclaimed storage unless an independently justified consumptive accounting bound is declared. A finite storage capacity remains a state bound; it does not by itself bound the lifetime number of updates. Budget replenishment must be included in the window's available budget, not assumed to be absent.

8.5 Retained-entropy bound

If each retained artifact contributes at most $b_{\text{commit},\text{max}}$ bits of authoritative retained content under the profile, then:

$$H_{\text{retained}}(T) - H_{\text{retained}}(0) \leq N_{\text{commit}}(T) \cdot b_{\text{commit},\text{max}}.$$

This is the increment form used by Theorem 2 in ARQ_Classical_Boundedness_Theorem_v0.2.md, under that theorem's M2 assumptions. The initial retained state is not required to be empty. This maintenance repair does not supply a new proof for undeclared state encodings or physical entropy.

In any case,

$$H_{\text{retained}}(T) \leq M_{\text{persist}}.$$

8.6 Landauer discipline

Model M2 MAY cite Landauer's principle only as a **sanity lower bound** on irreversible physical information handling.

It MUST NOT treat

$$k_B T \ln 2$$

as the universal real operational cost of retention unless the implementation explicitly justifies that approximation.

Hard rule: In ARQ, $e_{\text{commit},\text{min}}$ SHOULD be treated as a calibrated engineering parameter of the declared storage/controller stack, not as a metaphysical constant.

8.7 Applicability

Use Model M2 when the relevant ARQ claim is:

- “retained novelty is bounded by commit authority,”
- “EA growth is bounded by persistent commit resources,”
- “witness-backed memory growth has a finite operational ceiling.”

9. Model M3 — Epoch-Gated Trust / Control Model

9.1 Purpose

Model M3 captures ARQ systems whose authority is valid only inside declared trust epochs.

This model is directly aligned with the ARQ trust/provenance profile, including:

- attestation epochs,

- calibration validity,
- early invalidation,
- trust-sensitive authority scaling,
- fail-closed re-entry.

9.2 Epoch structure

An implementation operates over intervals

$$[t_k, t_{k+1})$$

called **epochs**.

During epoch k , the following are frozen or policy-bounded:

- trust assumptions,
- attestation report,
- boundary declaration,
- privilege envelope,
- controller calibration,
- anomaly baseline policy.

9.3 Admissible-state set

For each valid epoch, let K_k denote the set of admissible ARQ states under the epoch's declared assumptions.

Then, during a valid epoch,

$$X_t \in K_k.$$

If the epoch is invalidated, the system **MUST** leave normal ARQ operation and transition to degraded or fail-closed mode.

9.4 Assumptions

Model M3 **REQUIRES**:

1. explicit epoch identifiers;
2. explicit invalidation triggers;
3. trust-sensitive action policy;
4. re-entry only through declared recalibration or attestation refresh.

9.5 Supported claim class

Model M3 supports an **epoch-bounded admissibility claim**:

ARQ behavior is bounded not only by storage or energy, but by a trust-valid admissible state set that collapses when the trust epoch is no longer valid.

This is a stronger operational claim than “memory is finite.” It says the system has no right to continue normal valuation once its declared assumptions have expired or broken.

9.6 Important limitation

Model M3 by itself does **not** prove a single lifetime scalar bound such as one global $H_{\max}$ over all imaginable deployments. It proves bounded operation **inside valid epochs**, and bounded authority across epoch transitions because invalid epochs force authority reduction or stop.

9.7 Applicability

Use Model M3 when the relevant ARQ claim is:

- “valuation authority is valid only inside attested assumptions,”
 - “controller drift must invalidate normal ARQ reasoning,”
 - “trust loss reduces the admissible state set before memory authority is granted.”
-

10. Model M4 — Online Queue / Backlog Model

10.1 Purpose

Model M4 captures ARQ as an online controller with finite unresolved event capacity.

This is the correct model for statements about:

- pending anomaly backlog,
- observation window congestion,
- witness flooding resistance,
- online latency stability.

10.2 Queue definition

Let the implementation maintain a pending-event structure with maximum capacity K , where each unresolved event occupies at most $B_{\max}$ bits of queue-accounted state.

10.3 Assumptions

Model M4 REQUIRES:

1. bounded queue capacity;
2. bounded per-event pending representation;
3. declared overflow behavior;
4. declared observation-window expiry semantics.

10.4 Pending-state bound

The unresolved ARQ pending state satisfies:

$$H_{\text{pending}}(t) \leq K \cdot B_{\max}.$$

10.5 Overflow rule

If K is reached, the implementation MUST do one or more of the following according to profile:

- reject new candidate promotions;
- downgrade new events to `log_only`;
- evict only non-authoritative pending entries under declared policy;
- reduce privilege;
- enter fail-closed if queue pressure itself invalidates safe judgment.

10.6 Applicability

Use Model M4 when the relevant ARQ claim is:

- “the unresolved disturbance backlog is bounded,”
- “the controller cannot accumulate unbounded pending ambiguity,”
- “online ARQ remains stable only if observation windows and pending queue are finite.”

11. Model M5 — Fixed Trusted Quantum Boundary Model

11.1 Purpose

Model M5 captures the quantum or hybrid regime in which ARQ valuation is performed only inside a fixed finite-dimensional trusted quantum boundary.

11.2 Boundary definition

The total Hilbert space is:

$$\mathcal{H} = \mathcal{H}_{SA} \otimes \mathcal{H}_U,$$

with

$$\mathcal{H}_{SA} = \mathcal{H}_S \otimes \mathcal{H}_A,$$

where:

- S = system register of interest,
- A = trusted ancilla boundary,
- U = uncontrolled environment.

The reduced trusted-boundary state is:

$$\rho_{SA}(t) = \text{Tr}_U \rho(t).$$

11.3 Assumptions

Model M5 REQUIRES:

1. finite trusted boundary dimensions d_S, d_A, d_{SA} ;
2. explicit declaration of which ancillas are trusted during the active epoch;
3. all usable-entanglement valuation to be performed only inside the trusted boundary;
4. boundary expansion or ancilla-map changes to require epoch change or explicit recalibration.

11.4 Supported entropy bound

For all t during the valid boundary epoch:

$$0 \leq S(\rho_{SA}(t)) \leq \log_2 d_{SA}.$$

Likewise, for the reduced system state $\rho_S(t)$:

$$0 \leq S(\rho_S(t)) \leq \log_2 d_S.$$

11.5 Supported usable-entanglement bound

If usable entanglement is evaluated only across admissible bipartitions inside the trusted boundary, then it is bounded by a constant depending only on the boundary dimensions.

In particular, if logarithmic negativity is used:

$$E_{usable}(\rho_{SA}(t)) \leq \min(\log_2 d_S, \log_2 d_A).$$

For qubit systems with n system qubits and m trusted ancilla qubits:

$$S(\rho_{SA}(t)) \leq n + m,$$

and

$$E_{usable}(\rho_{SA}(t)) \leq \min(n, m)$$

when logarithmic negativity is used.

11.6 Important limitation

Model M5 is a **finite-boundary theorem**, not an energy-budget theorem.

It bounds internal entropy and usable entanglement because the trusted Hilbert-space dimension is finite.

Energy, cooling, and calibration still matter operationally, but they belong to:

- epoch validity,
- control fidelity,
- state-estimation quality,
- and fail-closed transitions,

not to the dimensional upper bound itself.

11.7 Applicability

Use Model M5 when the relevant ARQ claim is:

- “internal quantum boundary entropy is bounded by finite Hilbert-space dimension,”
- “usable entanglement cannot diverge inside a fixed trusted boundary,”
- “trusted ancilla maps constrain the maximum quantum resource ARQ can claim.”

12. Model Selection Rules

12.1 No theorem without model label

Every theorem, proposition, bound, or operational guarantee in the ARQ supplement MUST declare which model(s) it depends on.

12.2 Minimal-claim rule

An ARQ document SHOULD claim no stronger result than the weakest model it explicitly declares.

12.3 Cross-model composition

Models MAY be composed only if their boundary assumptions are compatible.

Examples:

- M1 + M2 = bounded persistent state plus bounded retained commit authority.
- M1 + M3 = finite authoritative state plus epoch-gated trust.
- M3 + M5 = quantum boundary validity under attested epochs.
- M2 + M4 = bounded retention plus bounded unresolved online backlog.

12.4 No silent composition

A document MUST NOT silently combine:

- finite-memory reasoning from M1,
- commit-cost reasoning from M2,
- trust-epoch reasoning from M3,
- queue reasoning from M4,
- and finite-boundary quantum reasoning from M5,

and present the sum as if it were a single primitive theorem.

13. Inadmissible Inferences

The following inferences are forbidden unless separately justified.

13.1 Forbidden inference: universal Landauer bound on retained authority

A document MUST NOT infer that all retained ARQ entropy is universally bounded by

$$E_{total}/(k_B T \ln 2)$$

without a declared storage/retention model.

13.2 Forbidden inference: memory bits imply only $\log_2 M$ retained entropy

If M_{persist} is the number of bits of authoritative memory, the primary classical state-space bound is of order M_{persist} , not $\log_2 M_{\text{persist}}$.

A $\log_2$ -type expression applies to the entropy of a probability distribution over M abstract alternatives, not to a memory physically capable of storing M bits.

13.3 Forbidden inference: queue backlog equals persistent memory

Pending unresolved queue state MUST NOT be confused with authoritative retained memory.

13.4 Forbidden inference: environment-controlled correlation counts as usable entanglement

Quantum correlation outside the declared trusted boundary MUST NOT be counted as usable ARQ value.

13.5 Forbidden inference: boundary drift without epoch consequence

If the trusted boundary, ancilla map, calibration, or controller assumptions drift, the implementation MUST NOT continue to claim the previous model unchanged.

14. Audit Requirements for Model Claims

Any implementation or document claiming support for one of these models SHOULD be able to disclose, at minimum:

- the active model label(s),
- the declared boundary,
- the relevant finite quantities (M_{persist} , K , B_{max} , d_S , d_A , etc.),

- the active epoch identifier where applicable,
- the overflow / invalidation / rollback behavior,
- and the companion documents that define valuation or trust for that model.

If these cannot be disclosed even in redacted form, the claim SHOULD be treated as weak and MAY be downgraded to descriptive rather than normative status.

15. Explicit and Hidden Bridges

Explicit bridge

$c = a + b$ becomes operationally honest in ARQ only when b is not treated as a fog of procedures, but as a declared bounded system model. This document is the bridge that turns ARQ from a vocabulary of correction into a mapped set of admissible machines.

Hidden bridge #1 — Ashby / cybernetics

Different failure modes require different regulators. Memory size, trust validity, pending backlog, and usable entanglement are not one variable with four costumes. The model split is an application of requisite variety: control claims must match the structure of the thing being controlled.

Hidden bridge #2 — L4 Hardware Perimeter / witness boundary

The boundary discipline here mirrors the L4 hardware-perimeter logic: radios, cables, USB, trust epochs, and physical attack surface are not “background.” Likewise in ARQ, external memory, uncontrolled ancillas, and silent authority channels are part of the boundary model, not footnotes.

Hidden bridge #3 — Beacon / continuity recognition

Beacon distinguishes badge, behavior, and consequence trail. This document does something similar for ARQ claims: state-space size, trust epoch, and witnessable consequence trail must agree before one says “the system is bounded” or “this deviation is valuable.”

16. Earth Paragraph (Engineering / Anatomy)

In anatomy, you do not explain a body with one number. Pulse is not blood volume, breathing is not memory, and fever is not identity. In engineering it is the same: SSD capacity is not controller trust, queue depth is not retained experience, and Hilbert-space dimension is not cooling budget. A stable organism survives because different limits are recognized as different limits. ARQ needs the same honesty. If everything is called “entropy” and pushed into one sack, the proof becomes poetry and the machine becomes unsafe.

17. Closing Statement

ARQ does not need one grand totalizing model.

It needs a small number of explicit, bounded, audit-compatible models that say:

- what the system is,
- where the boundary is,
- what quantity is being bounded,
- and what exactly would invalidate the claim.

That is the role of this document.

Without it, ARQ risks becoming broader in language than it is in truth. With it, ARQ becomes a disciplined stack whose theorems know where they live.