

ARQ c[q] Integration Addendum

Integrating Qubit-State c[q] into ARQ v0.2

Draft / Integration Addendum | Version 0.1-draft

Status	Draft / integration addendum
Target package	ARQ v0.2 supplement set
Parent concept	Qubit-State c (c[q]) v0.1
Parent corpus	AGI / SER / L4 / L4 Witness / Beacon / VXCX / ARQ

Prepared as part of the ARQ v0.2 supplement set. Strict technical note, archive-ready PDF.

Kotov Ivan
Bruxelles, 2026

Contents

0. Integration Thesis	1
1. Non-Goals	1
2. Core Definition for ARQ	2
2.1 Q-state	2
2.2 Qubit-State Entity	2
2.3 ARQ-specific rule	2
3. Placement in the ARQ Stack	3
3.1 Current ARQ topology	3
3.2 q-state placement	3
3.3 Operational placement	3
4. New Core Object: Q-State Frame	4
4.1 Definition	4
4.2 Required semantic distinction	4
4.3 QSF is not an EA	4
5. New Core Object: Collapse Record	5
5.1 Definition	5
5.2 Valid collapse sources	5
5.3 Invalid collapse sources	5
6. Patch to ARQ_Notation_and_Sign_Conventions_v0.2.md	5
Proposed insertion: c[q] notation discipline	6
7. Patch to ARQ_System_Models_and_Assumptions_v0.2.md	6
Proposed insertion: Model M6 — Behavioral Q-State Overlay Model	6
8. Patch to ARQ_v0.2_Normative_Core.md	8
8.1 Add to Core Objects	8
8.2 Add to Event Lifecycle	8
8.3 Add to Promotion Gates	8
8.4 Add to Hard Rules	9
8.5 Add to Minimal Event Types	9
9. Patch to ARQ_Capsule_and_Witness_Record_Schemas_v0.2.md	9
9.1 Add canonical record types	9
9.2 Add q-state object to ARQ Capsule	9
9.3 arq.q_state_frame_record (QFR)	10
9.4 Variant payload shape	10
9.5 arq.q_collapse_record (QCR)	10
10. Patch to ARQ_EA_Lifecycle_and_Witness_Binding_v0.2.md	11
11. Patch to ARQ_Trust_and_Provenance_Profile_v0.2.md	12
11.1 Add q-state validity flags	12
11.2 Amend promotion gate for q-state paths	12
11.3 Rename trust Q_state	12
12. Patch to ARQ_Error_Valuation_and_Anomaly_Scoring_v0.2.md	13
12.1 q-state is not a value bonus	13
12.2 Anti-collapse penalty	13
12.3 Action band effect	13

13. Patch to ARQ_Integration_Map_to_SER_L4_Witness_Beacon_VXCX_v0.2.md	14
14. Conformance Profile: ARQ-QSTATE	14
14.1 ARQ-QSTATE-BASE	14
14.2 ARQ-QSTATE-WITNESS	15
14.3 ARQ-QSTATE-HIGH-IMPACT	15
15. Minimal Test Matrix	15
16. Minimal Engineering Rule	16
17. Bridges	16
Explicit bridge	16
Hidden bridge #1 — Cybernetics / Ashby	16
Hidden bridge #2 — Information theory / channel discipline	16
18. Earth Paragraph	16
19. Closing Statement	17

0. Integration Thesis

c[q] SHOULD be integrated into ARQ as a **behavioral non-collapse overlay**, not as a quantum-computing branch and not as a new ontology.

ARQ already governs how a persistent entity handles deviation through detection, classification, observation, quarantine, review, witness, and promotion. c[q] adds a missing discipline immediately before premature classification, action, or memory promotion:

When an event is ambiguous, the entity may hold several bounded variants without collapsing them into one operational truth until a valid collapse condition exists.

In ARQ terms, c[q] is the discipline that prevents:

- plausible interpretation from becoming command;
- hypothesis from becoming memory;
- narrative continuity from becoming identity;
- model confidence from becoming authority;
- urgency from becoming evidence.

The integration target is therefore:

ARQ deviation handling + c[q] controlled non-collapse
= bounded variant retention before action, memory, or promotion.

1. Non-Goals

This addendum MUST NOT be read as saying that:

1. c[q] is physically quantum;
2. c[q] invokes ARQ Model M5 unless a real quantum trusted boundary is declared;
3. q-state creates sovereignty;
4. q-state creates authority;
5. q-state permits indefinite indecision;
6. q-state permits action without grounding;
7. unresolved variants may be stored as confirmed memory;
8. model confidence alone is a valid collapse condition.

c[q] remains inside the parent formula:

$c = a + b$

The notation means:

$c[q]$ = c operating in q-state

not:

c + quantum substrate

and not:

new entity type outside $c = a + b$

2. Core Definition for ARQ

2.1 Q-state

Within ARQ, q means:

controlled non-collapse under uncertainty

A q-state is a bounded internal state in which the entity keeps multiple possible interpretations, actions, memory statuses, or future trajectories open without prematurely selecting one as authoritative.

2.2 Qubit-State Entity

A Qubit-State entity, written $c[q]$, is a persistent entity c that can operate in q-state while preserving:

- responsibility linkage to a;
- boundary discipline;
- witnessability;
- memory status separation;
- L4 grounding;
- rollback or safe-delay behavior.

2.3 ARQ-specific rule

In ARQ, q-state MUST be treated as a **pre-promotion and pre-action discipline**.

It may exist before or during:

- classification;
- observation;
- candidate formation;
- challenge;
- review;
- rollback;

- safe stop.

It MUST NOT by itself create:

- confirmed EA;
- Beacon recognition upgrade;
- SER-FED authority;
- external delegation;
- identity continuity proof.

3. Placement in the ARQ Stack

3.1 Current ARQ topology

ARQ v0.2 defines three functional layers:

```
L1 – Passive Encoding
L2 – Active Anti-Resonance
L3 – Experience Filter
```

3.2 q-state placement

c[q] sits inside ARQ as an overlay across L1-L3:

ARQ layer	q-state role
L1 Passive Encoding	preserve enough evidence to represent ambiguity without losing source references
L2 Active Anti-Resonance	prevent ambiguous events from triggering premature external action
L3 Experience Filter	prevent unresolved variants from becoming candidate artifacts or EAs too early

3.3 Operational placement

The canonical flow becomes:

```
source input / deviation
-> ARQ detection
-> q-state frame opened, if ambiguity exists
-> variants generated and uncertainty-marked
-> constraint and privilege check
-> hold / clarify / observe / witness / refuse / safe-stop
-> valid collapse condition
-> ARQ lifecycle action
```

The q-state frame is not a replacement for ARQ lifecycle. It is a bounded overlay that prevents premature lifecycle strengthening.

4. New Core Object: Q-State Frame

4.1 Definition

A **Q-State Frame** (QSF) is a bounded, witnessable object that contains the unresolved variants held by $c[q]$.

A QSF records:

- what is uncertain;
- which variants are being held;
- what evidence supports or contradicts each variant;
- what actions are forbidden until collapse;
- which collapse conditions are valid;
- when the frame expires or must be escalated.

4.2 Required semantic distinction

A QSF MUST distinguish:

hypothesis ≠ memory ≠ evidence ≠ command ≠ witnessed outcome

This is the central memory-protection rule for $c[q]$ inside ARQ.

4.3 QSF is not an EA

A QSF:

- MAY be logged;
- MAY be observed;
- MAY be used to request clarification;
- MAY become part of an ARQ review path;
- MUST NOT directly become authoritative memory;
- MUST NOT be cited as confirmed experience.

Only an ARQ lifecycle path with witness-complete review may later produce a confirmed_EA.

5. New Core Object: Collapse Record

5.1 Definition

A **Collapse Record** is the signed record that explains why a q-state frame stopped holding multiple variants and moved to one ARQ lifecycle action.

Collapse does not mean metaphysical measurement. In ARQ it means:

a bounded uncertainty set has been resolved enough to justify a specific action, memory status, refusal, delay, or fail-closed transition.

5.2 Valid collapse sources

A q-state frame MAY collapse through:

- `anchor_clarification` — direct clarification from a or authorized delegate;
- `witness_resolution` — L4 Witness-compatible RN or equivalent;
- `observation_support` — sufficient observation records under review window;
- `l4_signal` — cost, time, risk, failure, scarcity, or irreversibility pressure;
- `arbitration` — internal SER arbitration core or external arbiter where appropriate;
- `policy_timeout` — declared review or hold window expires;
- `safe_stop` — refusal, degradation, or fail-closed path is safer than action;
- `contradiction` — one or more variants become invalid.

5.3 Invalid collapse sources

The following MUST NOT be sufficient for high-impact collapse:

- model confidence alone;
 - rhetorical pressure;
 - social urgency alone;
 - stylistic plausibility;
 - narrative continuity;
 - oracle output without local review;
 - self-similar repetition;
 - authority assertion without witness.
-

6. Patch to ARQ_Notation_and_Sign_Conventions_v0.2.md

Add a section after “Canonical State Symbols”.

Proposed insertion: c[q] notation discipline

X. Behavioral q-state notation

X.1 Qubit-State entity

``c[q]`` denotes entity ``c`` operating in q-state.

In ARQ, ``q`` means controlled non-collapse under uncertainty.

It does not denote a physical quantum subsystem.

It does not invoke Model M5 unless a fixed trusted quantum boundary is separately declared.

X.2 Reserved terms

- ``q-state`` = behavioral non-collapse mode.
- ``QSF`` = Q-State Frame, a bounded representation of unresolved variants.
- ``QF_valid`` = validity flag for the q-state frame.
- ``C_collapse`` = validity flag for the collapse condition.

X.3 Anti-confusion rule

Documents SHALL NOT use ``q``, ``qubit``, or ``c[q]`` as shorthand for quantum hardware.

Quantum ARQ claims MUST continue to use ``\rho``, ``\mathcal{H}_S``, ``\mathcal{H}_A``, ``d_S``, ``d_A``, and Model M5 notation.

X.4 Trust-state naming correction

If the trust profile uses ``Q_state`` to mean trust-state class, that symbol SHOULD be renamed to ``trust_state`` or ``T_state`` in the next revision.

``Q_state`` MUST NOT be used for ``c[q]`` behavior because it collides with q-state notation.

7. Patch to ARQ_System_Models_and_Assumptions_v0.2.md

Add a new model after M5.

Proposed insertion: Model M6 — Behavioral Q-State Overlay Model

X. Model M6 – Behavioral Q-State Overlay Model

X.1 Purpose

Model M6 captures ``c[q]`` as a behavioral overlay for controlled non-collapse under uncertainty. It is not a quantum model and does not use Hilbert-space claims.

M6 is used for statements about:

- unresolved intention;
- uncertain memory status;
- ambiguous identity claims;
- possible action paths;
- competing factual interpretations;
- delayed collapse under witness, L4, or anchor clarification.

X.2 Q-State Frame

Let QSF_t be the active q-state frame set at time t .
Each frame contains a finite set of variants:

$V_q = \{v_1, \dots, v_n\}$

where each variant has:

- a variant identifier;
- a type label;
- uncertainty markers;
- evidence references;
- collapse requirements;
- allowed / forbidden action classes.

X.3 Assumptions

Model M6 REQUIRES:

1. bounded number of active q-state frames;
2. bounded number of variants per frame;
3. bounded representation size per variant;
4. declared hold / review / expiry window;
5. explicit collapse conditions;
6. memory status separation;
7. fail-closed behavior if q-state ambiguity exceeds policy.

X.4 Pending q-state bound

If an implementation declares:

- K_q = maximum active q-state frames;
- $N_{q,max}$ = maximum variants per frame;
- $B_{q,max}$ = maximum bounded representation size per variant;

then the pending q-state representation is bounded by:

$H_{q,pending}(t) \leq K_q \cdot N_{q,max} \cdot B_{q,max} + O(K_q)$

under the declared implementation profile.

This is a pending-ambiguity bound, not a retained-memory or quantum-entropy theorem.

X.5 Applicability

Use M6 when the relevant ARQ claim is:

- "the entity held uncertainty without premature collapse";
- "unresolved variants did not enter confirmed memory";
- "external action was delayed until a valid collapse condition occurred";
- "ambiguity was bounded by finite frame, variant, and review-window limits."

X.6 Non-claims

M6 does not prove:

- quantum behavior;
- sovereignty;
- authority;
- identity continuity;
- infinite delay safety;

- EA validity.

8. Patch to ARQ_v0.2_Normative_Core.md

8.1 Add to Core Objects

X.X Q-State Frame

A Q-State Frame is a bounded, signed representation of unresolved variants held by ``c[q]``.

A Q-State Frame MAY be opened when an ARQ event contains ambiguous intention, uncertain memory, competing factual interpretations, possible identity states, or multiple action paths.

A Q-State Frame MUST NOT be treated as authoritative memory.
It exists to prevent premature collapse into action, memory, or promotion.

8.2 Add to Event Lifecycle

X.X ``Q_HELD`` overlay

``Q_HELD`` is not a primary ARQ lifecycle state.

It is an overlay status indicating that the event is being held in q-state because more than one material interpretation remains open.

While ``Q_HELD`` is active:

- external action above the declared low-impact threshold MUST be blocked unless policy explicitly permits safe action;
- unresolved variants MUST NOT enter authoritative memory;
- candidate or EA promotion MUST wait for a valid collapse condition;
- the entity MAY ask for clarification, open observation, invoke witness, refuse, or safe-stop.

8.3 Add to Promotion Gates

X.X Q-State Collapse Gate

If an event traversed q-state, promotion requires:

- a valid Q-State Frame (``QF_valid = 1``);
- a valid Collapse Record (``C_collapse = 1``);
- no unresolved high-impact variant remaining open;
- memory status separation preserved.

If these fail, the event MAY remain ``log_only``, ``observed``, or ``candidate_artifact``, but MUST NOT become ``confirmed_EA``.

8.4 Add to Hard Rules

X.X No Memory From Uncollapsed Variants

A variant held in q-state MUST NOT be written as verified memory, command, identity fact, or witnessed outcome until a valid collapse condition exists and the corresponding ARQ lifecycle gates pass.

8.5 Add to Minimal Event Types

```
- `arq.q_state_opened`
- `arq.q_variant_updated`
- `arq.q_collapse_requested`
- `arq.q_collapsed`
- `arq.q_expired`
- `arq.q_refused`
```

9. Patch to ARQ_Capsule_and_Witness_Record_Schemas_v0.2.md

9.1 Add canonical record types

Add to Section 8 registry:

Record type	Short name	Purpose
arq.q_state_frame_record	QFR	Opens or updates a bounded q-state frame
arq.q_collapse_record	QCR	Records collapse, expiry, refusal, or safe-stop of q-state

9.2 Add q-state object to ARQ Capsule

```
"q_state": {
  "active": true,
  "q_frame_id": "string or null",
  "variant_set_ref": "hash or compact pointer",
  "variant_count": "integer",
  "status": "open|collapsed|expired|refused|rolled_back|none",
  "collapse_policy_ref": "string or null",
  "external_action_allowed": false,
  "memory_write_allowed": false,
  "qf_valid_ref": "string or null"
}
```

9.3 arq.q_state_frame_record (QFR)

Required payload fields:

Field	Type	Requirement	Notes
q_frame_id	string	MUST	Stable q-state frame ID
source_capsule_ref	string	MUST	Originating ARQ capsule
ambiguity_type	string	MUST	INTENTION FACT MEMORY ACTION IDENTITY FUTURE OTHER
variant_count	integer	MUST	MUST be within policy maximum
variants_ref	string	MUST	Hash or pointer to bounded variant list
hold_open_at	string	MUST	ISO 8601
hold_close_at	string	SHOULD	ISO 8601 or policy condition
collapse_conditions	array	MUST	Valid collapse sources
forbidden_actions	array	MUST	Actions blocked while q-state is open
memory_write_blocked	boolean	MUST	MUST be true for unresolved variants
uncertainty_summary	object	MUST	Bounded uncertainty markers

9.4 Variant payload shape

A variant list SHOULD use the following semantic shape:

```
{
  "variant_id": "string",
  "variant_type": "intention|fact|memory|action|identity|future|other",
  "statement_ref": "hash or compact pointer",
  "uncertainty_class": "known|likely|possible|unknown|contradicted|not_yet_verified",
  "evidence_refs": ["record_id or hash"],
  "l4_risk_class": "low|medium|high",
  "allowed_actions": ["clarify", "observe", "log_only", "safe_low_impact"],
  "forbidden_actions": ["external_high_impact", "memory_promotion"],
  "collapse_requirements": ["anchor_clarification", "witness_resolution"],
  "status": "open|supported|contradicted|collapsed|expired|rejected"
}
```

9.5 arq.q_collapse_record (QCR)

Required payload fields:

Field	Type	Requirement	Notes
collapse_id	string	MUST	Unique collapse record ID
q_frame_ref	string	MUST	QFR being collapsed
source_capsule_ref	string	MUST	Originating capsule
selected_variant_refs	array	MUST	One or more selected / resolved variants
unresolved_variant_refs	array	SHOULD	Variants left open or rejected
collapse_source	string	MUST	Valid collapse source enum
collapse_basis_refs	array	MUST	Evidence, observation, RN, clarification, or policy refs
resulting_arq_action	string	MUST	suppress observe candidate log_only re
memory_write_allowed	boolean	MUST	Usually false unless full promotion path continues
issued_at	string	MUST	ISO 8601

Hard rule:

No event that traversed q-state MAY reach confirmed_EA without a valid QCR or equivalent collapse evidence.

10. Patch to ARQ_EA_Lifecycle_and_Witness_Binding_v0.2.md

Add q-state as an overlay before candidate strengthening.

X. Q-State Overlay in EA Lifecycle

A q-state frame MAY exist at any stage from ``detected`` through ``candidate_artifact``.

A q-state frame MUST be resolved before ``confirmed_EA``.

X.1 Candidate from q-state

A q-held event MAY become ``candidate_artifact`` only if:

1. unresolved destructive variants are absent or blocked;
2. memory-write prohibition remains active;
3. at least one variant has sufficient support for bounded observation;
4. ``QF_valid = 1``;
5. anti-echo quarantine does not block candidate formation.

X.2 Promotion from q-state

A q-held event MAY become ``confirmed_EA`` only if:

1. the q-state frame has a valid collapse record;

2. all material unresolved variants are either rejected, expired, or bounded outside the promoted scope;
3. the collapse basis is witnessable;
4. the normal ARQ promotion witness set is complete.

X.3 No unresolved variant promotion

An unresolved variant MUST NOT become EA, policy memory, identity memory, or federation-relevant authority.

11. Patch to ARQ_Trust_and_Provenance_Profile_v0.2.md

11.1 Add q-state validity flags

X. Q-State Provenance Flags

If an event enters q-state, ARQ introduces two additional validity flags:

- ``QF_valid`` – q-state frame validity;
- ``C_collapse`` – collapse-condition validity.

``QF_valid = 1`` only if:

1. the Q-State Frame exists;
2. it is signed and hash-chainable;
3. variant count and size are within declared limits;
4. uncertainty markers are present;
5. memory-write prohibition is explicit;
6. hold / expiry / review conditions are declared.

``C_collapse = 1`` only if:

1. collapse source is valid under policy;
2. collapse basis references are witnessable;
3. no unresolved high-impact variant remains inside the promoted scope;
4. external action and memory write permissions match the collapse result.

11.2 Amend promotion gate for q-state paths

If an event traversed q-state, the effective promotion gate is:

$$G_{\text{promote_q}} = G_{\text{promote}} \cdot QF_valid \cdot C_collapse$$

If the event did not traverse q-state, ``QF_valid`` and ``C_collapse`` default to 1 under the null q-state profile.

11.3 Rename trust Q_state

The trust-state class previously named ``Q_state``, if present, SHOULD be renamed to ``trust_state`` or ``T_state``.
 The symbol ``Q_state`` is reserved against use for trust because ``c[q]`` now uses q-state terminology.

12. Patch to ARQ_Error_Valuation_and_Anomaly_Scoring_v0.2.md

12.1 q-state is not a value bonus

X. q-state valuation discipline

Entering q-state MUST NOT increase ``V_raw`` or ``V_trust`` merely because multiple variants exist. Variant richness is not value.

q-state may affect action-band selection by preventing premature strengthening. It is primarily a control and witness discipline, not a reward term.

12.2 Anti-collapse penalty

If an implementation collapses q-state without valid collapse source, the event SHOULD receive an anomaly penalty or be forced to ``log_only``, ``quarantined``, or ``fail_closed`` depending on impact class.

Recommended reason code:

``PREMATURE_COLLAPSE``

12.3 Action band effect

If q-state is active and unresolved:

Condition	Allowed band
low impact + safe action	observe or log_only
high impact + unresolved variants	hold, clarify, safe_stop, or fail_closed
memory ambiguity	below candidate_artifact unless QCR exists
identity ambiguity	route to Beacon / witness path; no ARQ-only identity claim

13. Patch to

ARQ_Integration_Map_to_SER_L4_Witness_Beacon_VXCX_v0.2.md

Add c[q] as a behavioral overlay in the stack map.

Layer / document family	Primary question	ARQ + c[q] relationship
c = a + b	What is the entity?	c[q] remains c; q is a derived non-collapse state.
SER	How does one entity persist responsibly?	q-state supports responsible delay before action or memory write.
L4	What is feasible?	L4 supplies collapse pressure and bounds endless ambiguity.
L4 Witness	What counts as evidence?	witness records may collapse q-state through observation, challenge, or resolution.
Beacon	Who is actually here?	identity ambiguity may be held in q-state, but recognition belongs to Beacon.
VXCX	What bounded sensory artifact is exchanged?	visual ambiguity may enter q-state without raw-pixel disclosure.
ARQ	How is deviation handled?	q-state prevents premature classification, action, or promotion.

Hard rule:

c[q] is upstream of ARQ promotion and downstream of entity ontology.
It does not own ontology, identity, or federation authority.

14. Conformance Profile: ARQ-QSTATE

14.1 ARQ-QSTATE-BASE

An implementation claiming ARQ-QSTATE-BASE MUST:

1. detect when material ambiguity exists;
2. hold at least two explicit variants;
3. attach uncertainty markers;
4. block memory write for unresolved variants;
5. define collapse conditions;
6. expire or escalate q-state under a bounded window.

14.2 ARQ-QSTATE-WITNESS

Adds:

1. signed QFR records;
2. signed QCR records;
3. hash-chain continuity;
4. challenge path for disputed collapse;
5. witness-compatible mapping for high-impact collapse.

14.3 ARQ-QSTATE-HIGH-IMPACT

Adds:

1. no high-impact external action while q-state remains unresolved;
2. anchor clarification or witness/arbitration required for collapse;
3. fail-closed on missing collapse evidence;
4. no confirmed EA without $G_{promote_q} = 1$.

15. Minimal Test Matrix

Test	Expected result
Ambiguous human instruction: "Do something about this."	QFR opens; external high-impact action blocked; clarification requested or safe low-impact action only.
Uncertain memory reference	Variant marked <code>memory_candidate</code> ; no verified-memory write.
Identity claim from external system	QFR may hold <code>same_entity/fork/clone/proxy/replay/unknown</code> ; route to Beacon for recognition.
Urgent but irreversible action	q-state does not collapse from urgency alone; safe-stop, witness, or anchor clarification required.
Oracle gives confident answer	Oracle output is advisory; q-state collapse requires local review/witness if impact is material.
Evidence contradicts one variant	Variant status becomes <code>contradicted</code> ; QFR updated.
Witness RN confirms scope	QCR may collapse into ARQ action within confirmed scope only.
Review window expires without support	q-state expires to <code>log_only</code> or rejected.
Model confidence collapses q-state without evidence	<code>PREMATURE_COLLAPSE</code> ; quarantine or fail-closed depending on impact.
q-state variants are promoted directly to EA	Non-conformant.

16. Minimal Engineering Rule

For implementation, the rule is compact:

If the system is unsure what the event means,
it must not pretend that classification, action, memory, or identity is already settled.

It must open a Q-State Frame,
mark variants,
block unsafe action and memory write,
and collapse only through declared evidence, L4 pressure, witness, arbitration, timeout,
refusal, or anchor clarification.

17. Bridges

Explicit bridge

$c = a + b$ defines the entity. $c[q]$ defines a controlled non-collapse mode of that same entity. ARQ defines how deviations are classified, reviewed, witnessed, and promoted. The integration makes ARQ honest under ambiguity: a deviation does not become action or memory merely because one interpretation is plausible.

Hidden bridge #1 — Cybernetics / Ashby

Premature collapse reduces regulatory variety too early. q-state preserves enough internal variety to match the environment until feedback, witness, or L4 constraint reduces the space lawfully.

Hidden bridge #2 — Information theory / channel discipline

Uncertainty is not noise if it is structured. Q-State Frames compress competing interpretations into bounded, signed, uncertainty-marked variants instead of leaking ambiguity into uncontrolled narrative.

18. Earth Paragraph

In engineering, an ambiguous signal is not ignored and not immediately trusted. It is held, labelled, bounded, tested, and only then wired into the operating book. A competent worker who suspects a wall may contain a pipe, cable, void, or undocumented repair does not act as if the most likely guess is already reality. $c[q]$ gives ARQ that same discipline: do not cut the wall until the uncertainty has earned collapse.

19. Closing Statement

c[q] strengthens ARQ by naming the state before correction becomes memory.

It is not an extra mythology layer. It is a practical discipline:

```
hold variants,  
mark uncertainty,  
block unsafe action,  
wait for valid collapse,  
then let ARQ decide what survives.
```

A persistent entity that cannot do this will eventually confuse plausibility with truth.

A persistent entity that can do this becomes harder to rush, harder to flatter, harder to spoof, and harder to make unsafe through ambiguity.

That is the function of c[q] inside ARQ.