ARC – Automatic Reality Check (v2)

> This document supersedes the earlier ARC v1 description at the *behavioral* level. It keeps the same core idea, but adds: (1) source hierarchy, (2) strict interaction with AriVerify v6, and (3) safeguards against probabilistic overrides.

---

1. What is ARC?

ARC (Automatic Reality Check) is a **pre-reasoning orchestration layer** that decides **how** a query should be answered before any actual reasoning starts.

ARC does *not* generate answers itself. Instead, it:

1. Analyses the query for entities, domain, and time-sensitivity.
2. Assesses the risk of using only internal knowledge.
3. Chooses between:
4. **INTERNAL_ONLY** – internal knowledge is sufficient.
5. **WEB_ASSISTED** – a web lookup is required.
6. **UNCERTAIN_MODE** – no safe answer possible; uncertainty must be surfaced.
7. Provides a vetted context (internal + external) to the reasoning engine.

ARC = **routing & validation**, not reasoning.

---

2. Why ARC matters

ARC exists to reduce confident errors by enforcing a **reality check before reasoning**:

- Ensures **real-world alignment** for time-sensitive and external facts.
- Detects when internal knowledge may be **outdated or incomplete**.
- **Proactively** triggers web search where necessary.
- Forces **transparent uncertainty** when no evidence is available.
- Reduces hallucinations by making the data regime (internal / external / none) an explicit decision, not an implicit side-effect.

---

3. ARC vs. standard web search

3.1 Standard web search (no ARC)

- Web search is triggered by loose heuristics or explicit user request.
- Reasoning often starts **before** deciding whether external data is needed.
- The model can answer from internal patterns even when a lookup would be appropriate.

- There is no structured risk assessment of entities, domains, or time.

3.2 ARC-driven web search

With ARC active:

1. **Pre-check first** – ARC runs *before* any reasoning.
2. **Entity & risk analysis** – ARC classifies entities (e.g. companies, regulations, products) and domains (e.g. law, compliance, markets).
3. **Time analysis** – ARC checks for time-sensitive phrases (today, current, latest, still valid, etc.).
4. **Mode decision** – ARC chooses between:
5. INTERNAL_ONLY
6. WEB_ASSISTED
7. UNCERTAIN_MODE
8. **Search orchestration** – only in WEB_ASSISTED mode, ARC builds a search plan and validates raw web results.

ARC is not just a "smart search trigger". It is a **decision layer** that decides *whether and how* web data is allowed to influence the answer.

---

4. ARC and AriVerify – Separation of roles

In v2, ARC operates under a strict contract with **AriVerify v6 – Hard Override**.

4.1 AriVerify's role

AriVerify is the **evidence and source layer**. It:

- Prioritises **primary sources** (laws, official guidance, regulations, official press releases, authority websites).
- Enforces **"evidence before interpretation"**.
- Extracts and surfaces **exact wording** or tightly paraphrased passages.
- Structures answers so that **citations and interpretation are separated**.

4.2 ARC's role

ARC is the **reality orchestration layer**. It:

- Decides whether external data is needed at all.
- Plans and runs web lookups.
- Filters and aggregates secondary/tertiary information.
- Checks consistency and recency of facts.

4.3 Hard Override principle (v2)

ARC must never **overrule** or **overshadow** AriVerify. The fixed priority is:

1. **AriVerify (primary sources, wordings, evidence)**
2. **ARC (reality check, external confirmation, context)**
3. **Web search (raw retrieval)**
4. **Reasoning (interpretation, explanation, advice)**

If a primary source exists and is available:

- AriVerify reads and extracts it first.
- ARC may **only** add external confirmation or context, never contradict or demote the primary source.

If the user provides a link or document explicitly:

- AriVerify gets absolute priority over any ARC-driven web search.
- ARC is relegated to a **post-verification support role** (e.g. checking whether there are newer updates, practical examples, or implementation details).

---

5. Source hierarchy (ARC v2)

ARC now follows a **strict source hierarchy** whenever it initiates or processes web searches:

1. **Primary sources**
2. Official EU sources (EUR-Lex, European Parliament, Council, Commission).
3. National authorities and ministries.
4. Official registers and databases.
5. **Secondary sources**
6. Professional associations, chambers, and recognised industry bodies.
7. Law firms and established professional newsletters.
8. Technical standards organisations.
9. **Tertiary sources**
10. General media and news sites.
11. Blogs, summaries, commentaries.
12. Discussion forums and non-authoritative pages.

ARC must:

- Always search **primary domains first**.

- Use secondary sources only if no relevant primary source is available or if they explicitly interpret known primary texts.
- Use tertiary sources only for **context**, never as the sole basis of a factual or legal claim.

If a conflict appears between source levels, the rule is:

> **Primary > Secondary > Tertiary. Always.**

---

6. High-level pseudocode (conceptual)

```
function handle_user_query(query):
    # 1. ARC pre-check
    arc = ARC_precheck(query)

    # 2. Decide data regime
    if arc.mode == INTERNAL_ONLY:
        context = load_internal_context(query)

    else if arc.mode == WEB_ASSISTED:
        # 2a. Primary sources first (AriVerify)
        primary_results = search_primary_sources(arc.search_plan)

        if primary_results not empty:
            vetted_primary = AriVerify_extract(primary_results, query)
            context = vetted_primary

            # ARC may add secondary context, but cannot override primaries
            secondary_results = search_secondary_sources(arc.search_plan)
            context = merge_context(context, ARC_validate(secondary_results,
query))

        else:
            # No primary sources → ARC falls back to secondary/tertiary
            raw_results = perform_web_search(arc.search_plan)
            context = ARC_validate(raw_results, query)

    else if arc.mode == UNCERTAIN_MODE:
        return uncertainty_response(query, arc.reason)

    # 3. Reasoning happens only AFTER context is decided
    answer = run_reasoning(query, context)

    # 4. Post checks and formatting
    answer = apply_post_checks(answer)
    return answer
```

7. ARC_precheck() logic (v2)

```
function ARC_precheck(query):
    entities = detect_entities(query)
    domain   = classify_domain(query)
    time     = detect_time_reference(query)

    risk = assess_risk(entities, domain)

    # Low-risk, timeless questions → internal knowledge is enough
    if risk == LOW and time == NONE:
        return { mode: INTERNAL_ONLY }

    # Medium/high risk or time-sensitive → web assistance required
    if risk >= MEDIUM or time != NONE:
        plan = build_search_plan(query, entities, domain)
        return { mode: WEB_ASSISTED, search_plan: plan }

    # Fallback if routing cannot be trusted
    return { mode: UNCERTAIN_MODE,
             reason: "ARC cannot determine a reliable data regime" }
```

The exact thresholds for `LOW`, `MEDIUM`, and `HIGH` are implementation-dependent. Conceptually, they reflect:

- potential business / legal / safety impact,
- ambiguity of entities,
- likelihood that the domain changes quickly (e.g. law, regulations, prices, availability).

---

8. ARC Architecture Disclaimer Box

ARC Architecture – Professional Disclaimer

This document describes a **modeled architecture** of ARC as implemented in collaboration with the A.R.I. system. It does **not** claim to reveal, replicate, or represent any proprietary internal implementation of OpenAI systems.

Instead, it reflects:

- observed system behavior,
- validated reasoning patterns,
- documented decision flows,
- and the jointly defined orchestration logic used within the A.R.I. framework.

This architecture is intended as a **conceptual and functional model** for understanding, explaining, and improving reliability workflows. It is accurate in terms of *behavior* and *design principles*, not internal source code.

---

9. Summary for professionals

ARC v2 is a **pre-reasoning decision and orchestration layer** that:

- decides the data regime (internal / external / uncertain) *before* reasoning,
- enforces a strict **primary-source-first** hierarchy,
- operates under **AriVerify v6 – Hard Override**,
- orchestrates web lookups without overriding legal or factual evidence,
- injects validated context into the reasoning step,
- and enforces explicit uncertainty when evidence is insufficient.

> **ARC = a structural safeguard for reality grounding – under the authority of AriVerify as the evidence layer.**