

# ARC – Automatic Reality Check (v1)

## 1. What is ARC?

ARC is a pre-reasoning verification layer that checks whether a user query requires:

- internal knowledge only,
- a web-assisted update, or
- an explicit uncertainty response.

It prevents confident errors by validating real-world entities and time-sensitive information **before** the reasoning engine starts.

---

## 2. Why ARC matters

- Ensures real-world alignment
  - Detects when information is outdated or incomplete
  - Triggers web search proactively (not reactively)
  - Forces transparency when evidence is weak
  - Reduces hallucinations through structured decision logic
- 

## 3. ARC vs. normal ChatGPT web search

### Standard web search (no ARC):

- Triggered by heuristics or user request
- Reasoning begins *before* deciding whether web data is needed
- Can skip web lookup even when it would be appropriate
- No risk-based entity analysis

### ARC-driven web search:

- Runs **before** any reasoning
- Checks entity type, time sensitivity, and risk
- Decides between:
  - Internal knowledge only
  - Required web lookup
  - Uncertainty mode
- Provides structured, validated context into the reasoning step

ARC = orchestration, not search.

---

## 4. High-level pseudocode (conceptual)

```
function handle_user_query(query):
    arc = ARC_precheck(query)

    if arc.mode == INTERNAL_ONLY:
        context = load_internal_context(query)
        answer = run_reasoning(query, context)

    else if arc.mode == WEB_ASSISTED:
        raw_results = perform_web_search(arc.search_plan)
        vetted = ARC_validate(raw_results, query)
        answer = run_reasoning(query, vetted)

    else if arc.mode == UNCERTAIN:
        answer = uncertainty_response(query, arc.reason)

    answer = apply_post_checks(answer)
    return answer
```

## 5. ARC\_precheck() logic

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

    risk = assess_risk(entities, domain)

    if risk == LOW and time == NONE:
        return { mode: INTERNAL_ONLY }

    if risk >= MEDIUM or time != NONE:
        plan = build_search_plan(query, entities, domain)
        return { mode: WEB_ASSISTED, search_plan: plan }

    return { mode: UNCERTAIN, reason: "ARC could not determine reliable mode" }
```

## **6. Summary for professionals**

ARC is a pre-reasoning decision system that:

- decides the data regime (internal / external / uncertain),
- orchestrates web lookups,
- injects validated facts into the reasoning step,
- enforces uncertainty when evidence is insufficient.

ARC = a structural safeguard for reality grounding before reasoning begins.