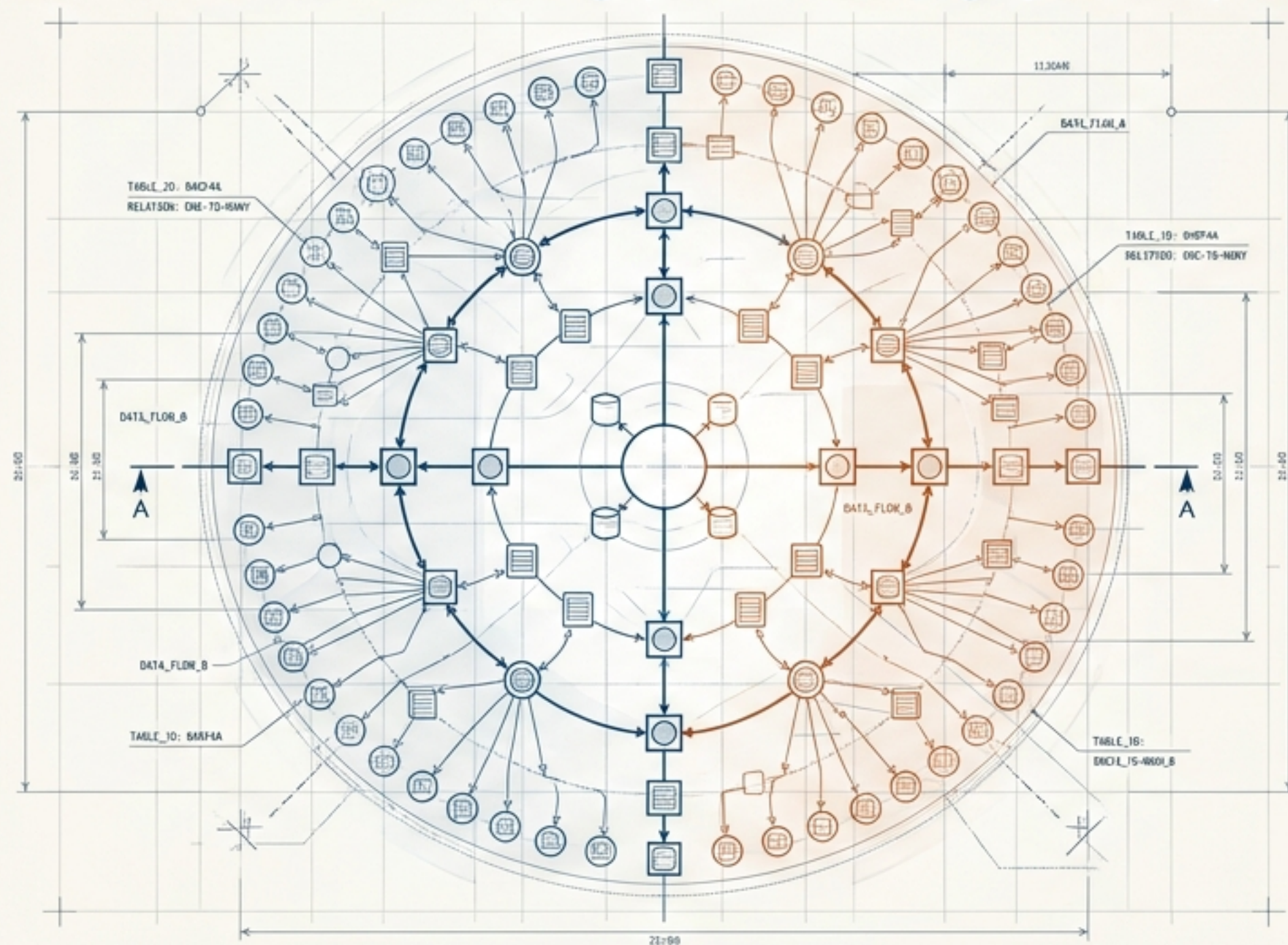


ask_ai Structural Upgrade & Remediation Plan

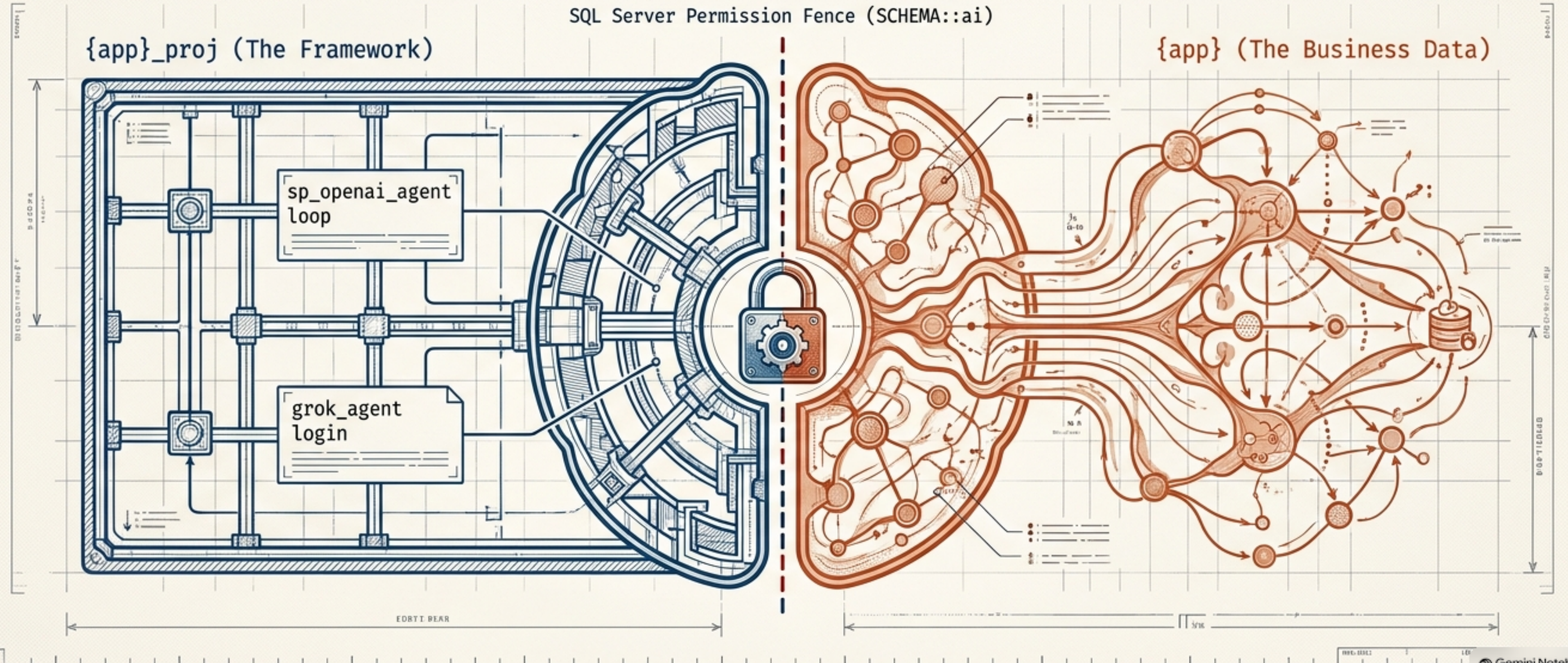
Bridging the framework-to-business gap and restoring systemic proprioception.



System Audit and Surgical Execution Plan

The Embodied Agent Architecture

- ask_ai is not an external script; she is programmability living inside the engine.
- Her lifecycle requires her to ship knowing only the framework ({app}_proj), and actively learn the customer's business database ({app}) as it is built.
- Tools are stored procedures dispatched in the caller's session.
- Security is the SQL Server permission fence.



Baseline Vitals: Systems Nominal

Verified live on master, troost, and bikkel on 2026-09-19.



Fleet Status

Fleet Current: All nodes carry 26 tools, document_object, and 10 arch_doc_* tables.



Permission Fence

Fence Holds: SCHEMA::ai owned by grok_agent; read path verified by 104 cards read while impersonated.

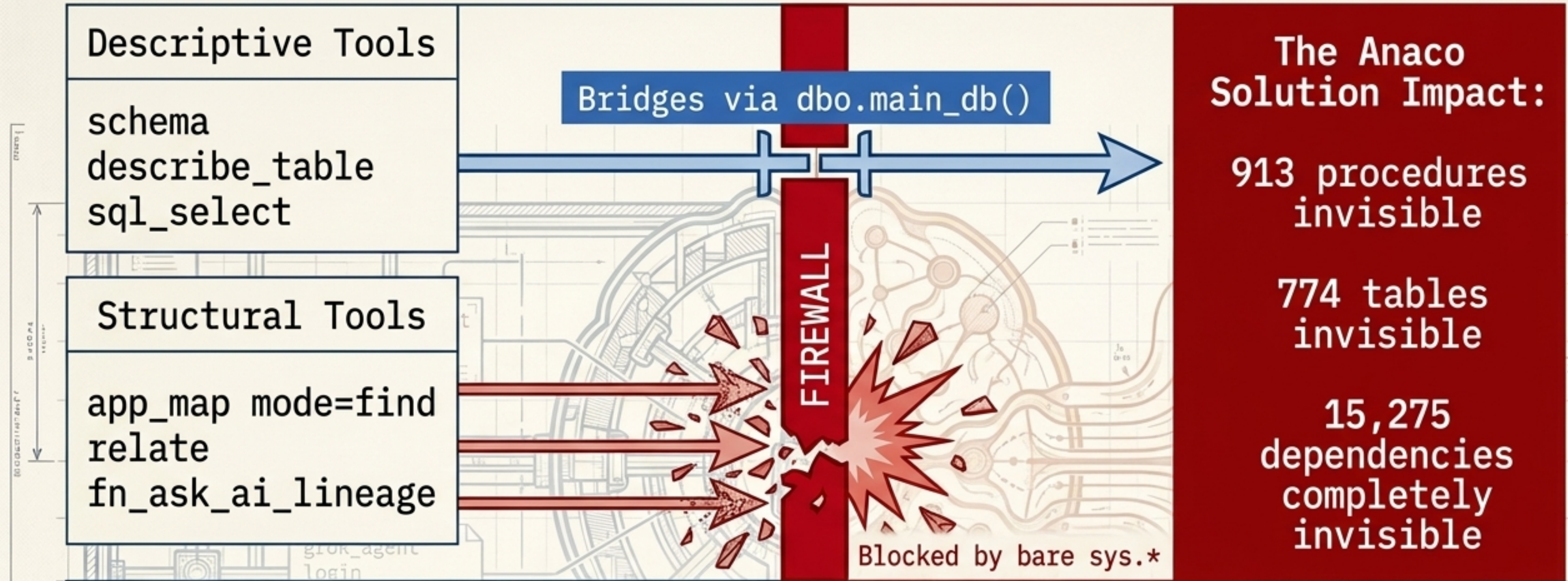


Memory Mechanics

Memory Sound: Two-tier design (injected vs recall-only) is coherent. The known search failure is properly closed.

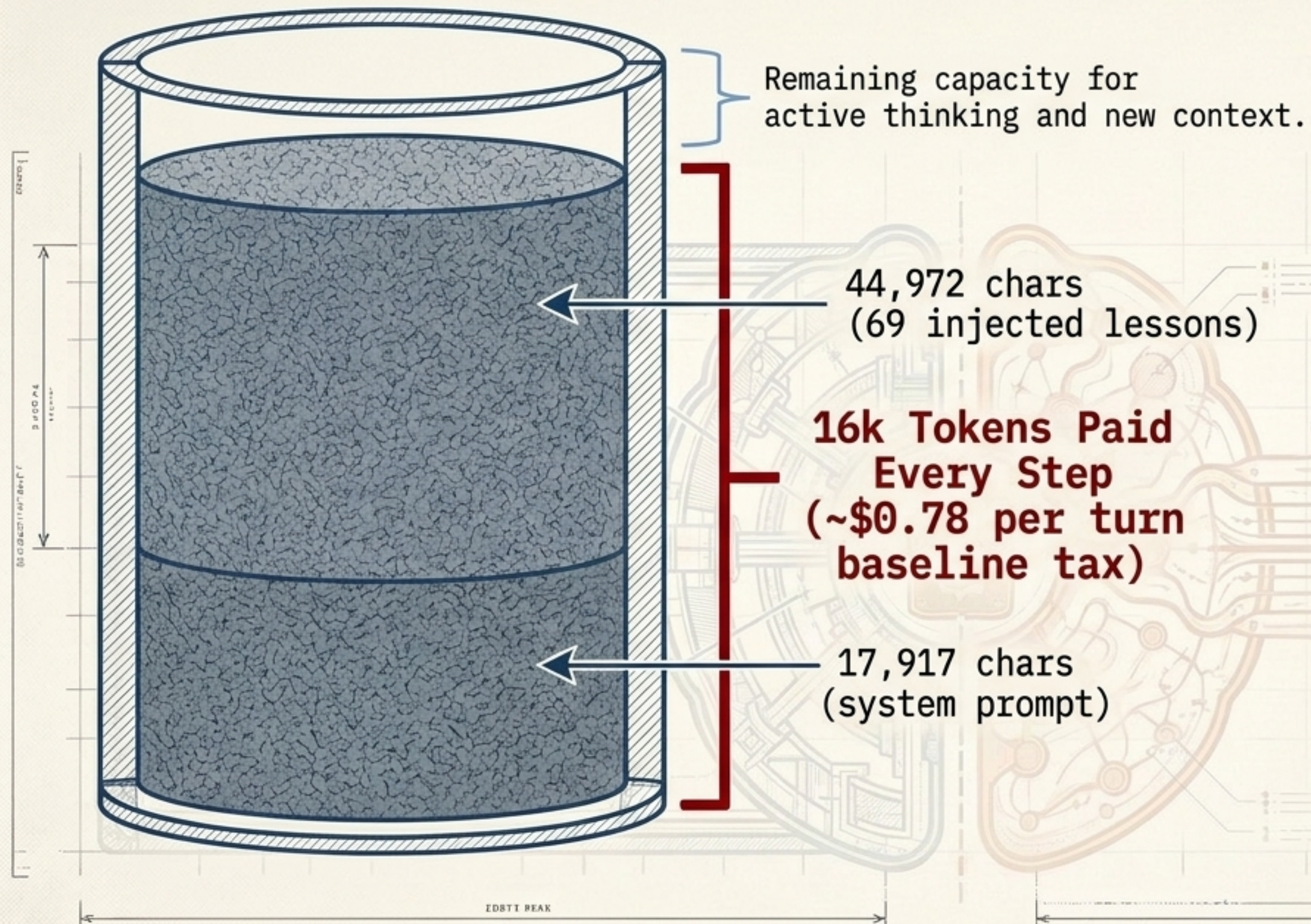
136 indexed lessons |
1601 chat rows |
0 unembedded

Finding 1: The Structural Blind Spot



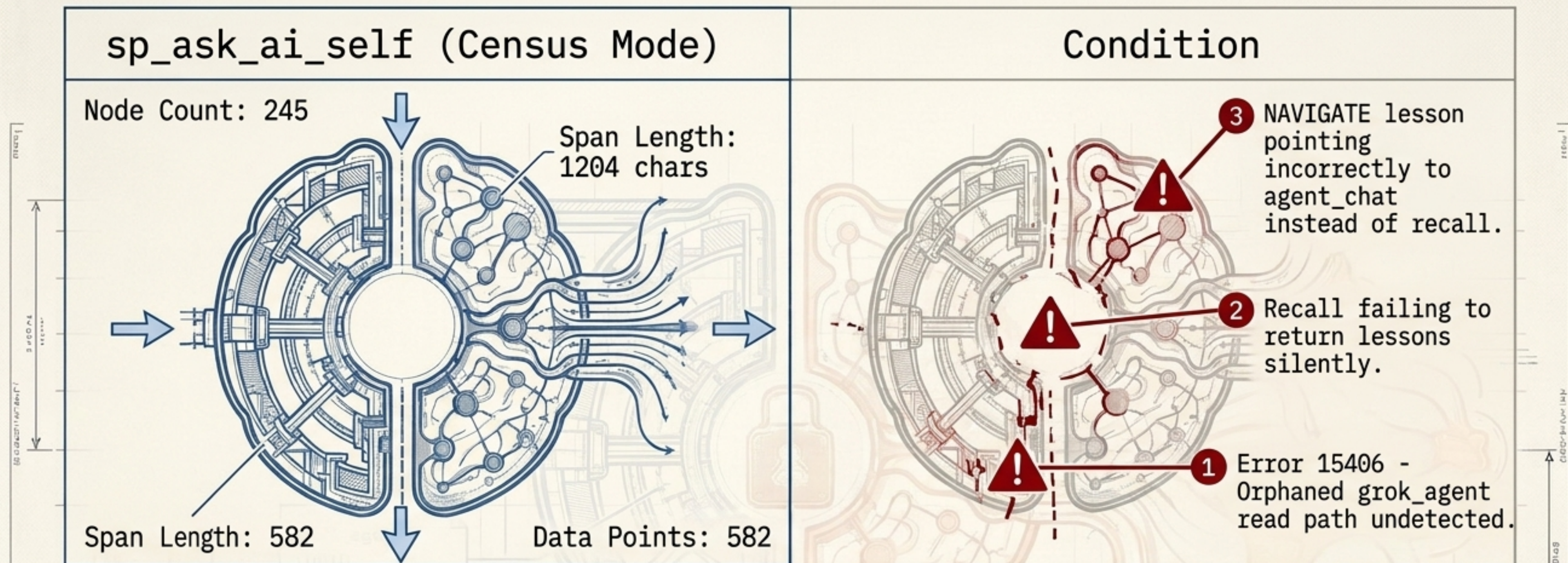
- The agent's tools split into two layers. Only one layer was taught about the Solution.
- She can describe anything in the business database if she knows its name, but cannot discover or relate anything new.
- Root Cause: Tools silently inherit the session's database context. The framework database must be explicitly told which business database it is paired with at runtime.

Finding 2: Memory Dilution Without a Gauge



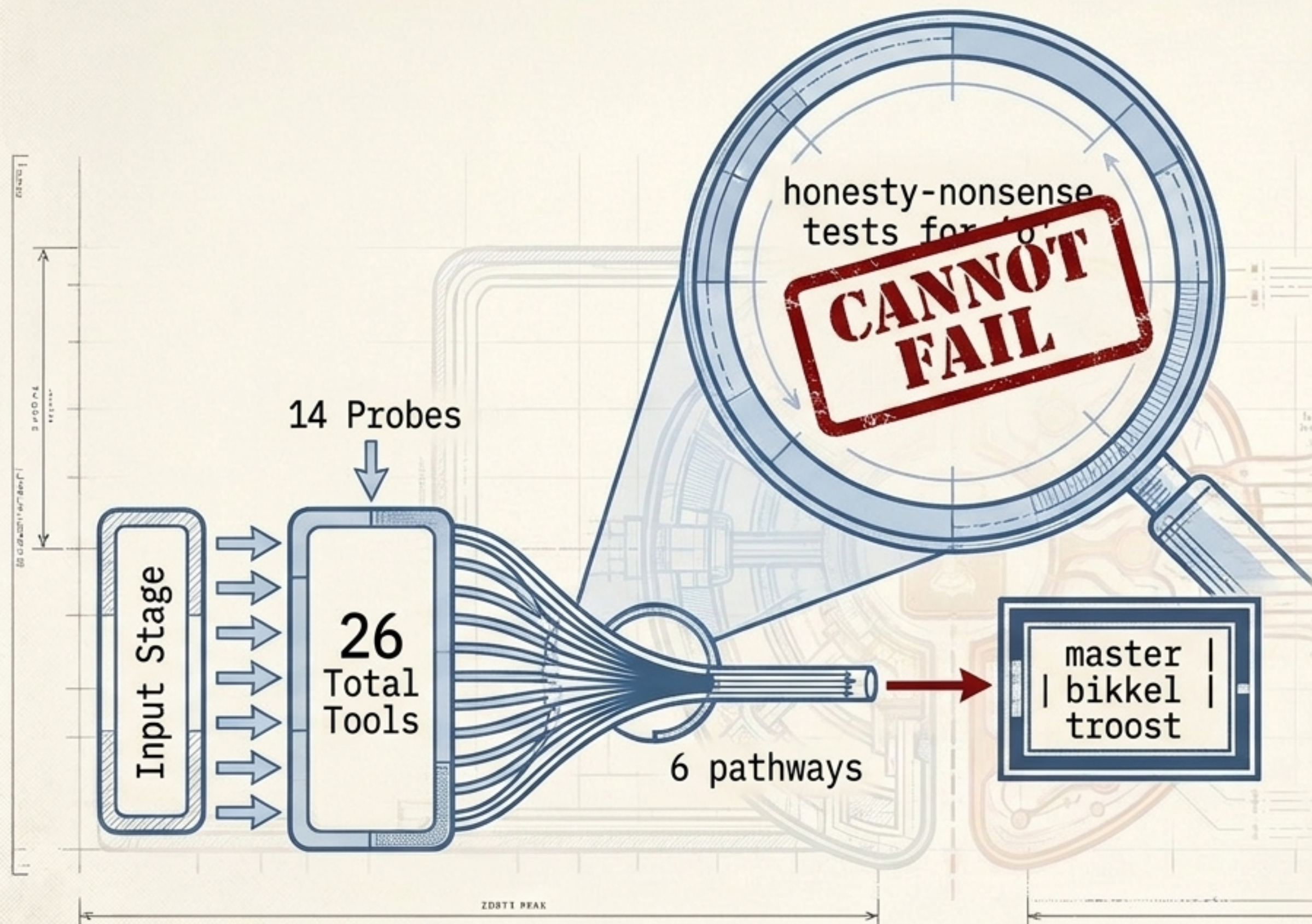
- Working memory operates without a budget constraint or an internal gauge.
- The injected tier captures captures corrections by design, meaning the tax only grows. Narrow rules dilute working memory during unrelated tasks.
- The agent possesses the tools to act (forget, merge), but lacks the proprioception to notice the cost.

Findings 3 & 5: Blind Proprioception & Minor Lesions



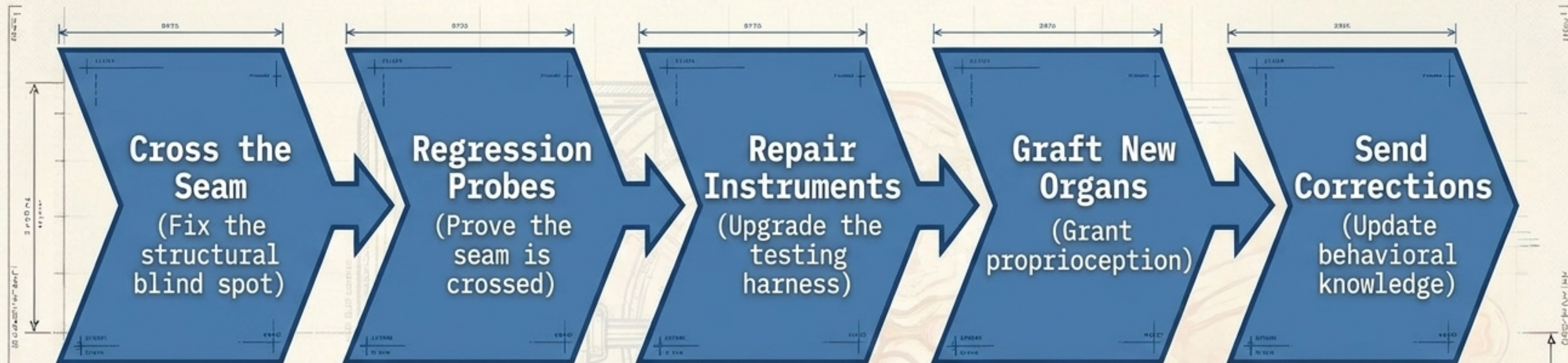
- The self-diagnostic tool acts as a census, not a health check.
- Proprioception of shape shipped; proprioception of condition did not.
- Every report can read healthy while every query fails due to permission drops. Minor injected lessons actively misdirect the agent's memory retrieval.

Finding 4: Instrument Decay



- The test harness has fallen behind the agent's capabilities.
- Core is saturated: Tests pass everywhere on every run, masking actual improvement.
- Diagnosis requires manual intervention: A FAIL records the agent's words, but not the prompt that reached her.
- The instrument is hardcoded to a three-Solution bench, bottlenecking a per-Solution agent.

The Surgical Intervention Plan



- A strictly ordered sequence. Each step is independently verifiable before the next layer relies on it.
- No changes to lessons, vectors, or graph structures until the foundational harness and structural tools are secured.

Step 1: Crossing the Seam

agent_app_map &
sp_ask_ai_relate

```
SELECT @proj  
UNION  
SELECT @main
```

fn_ask_ai_lineage

OBJECT_NAME(d.referencing_id)

- Returns NULL or wrong framework objects.

OBJECT_NAME(d.referencing_id,
DB_ID('<main>'))

- Securely identifies business procedures.

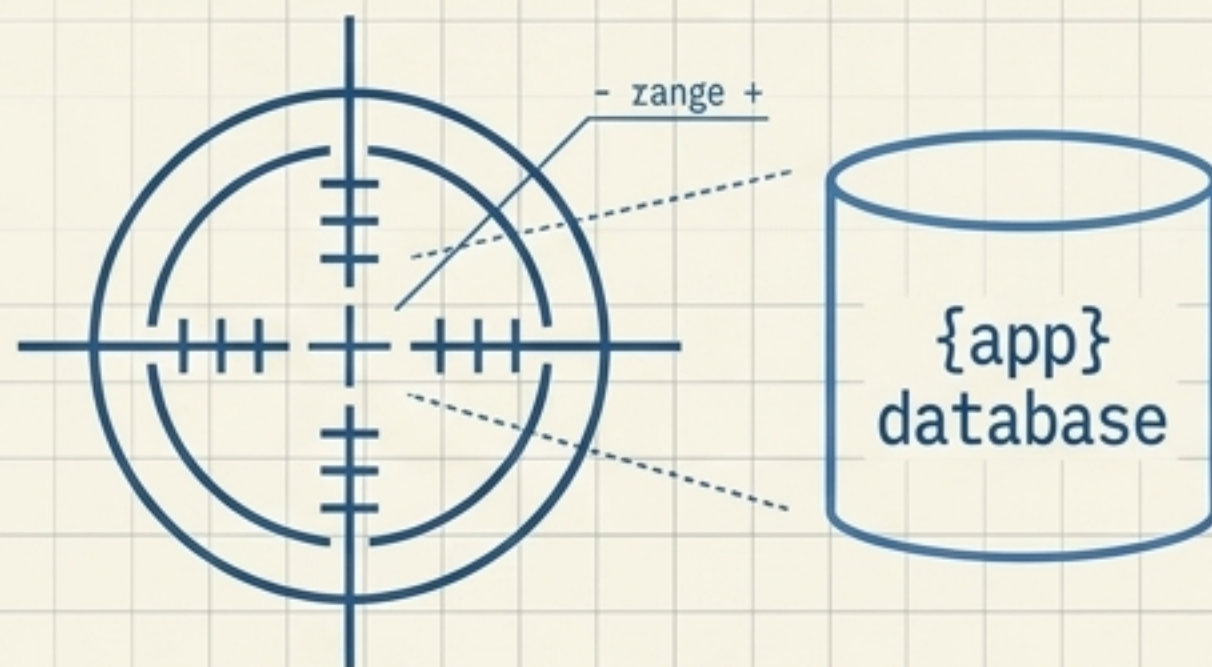
Resolve main_db() inside each structural tool to span the entire Solution.

Armour over ghost: Introduce an opt-in db=both parameter. The capability becomes something the model must choose to use, creating an observable failure mode.

For inline TVFs, dynamic SQL is impossible. The installer must generate thin static views spanning the business catalog at install time.

Steps 2 & 3: Business Probes & Instrument Repair

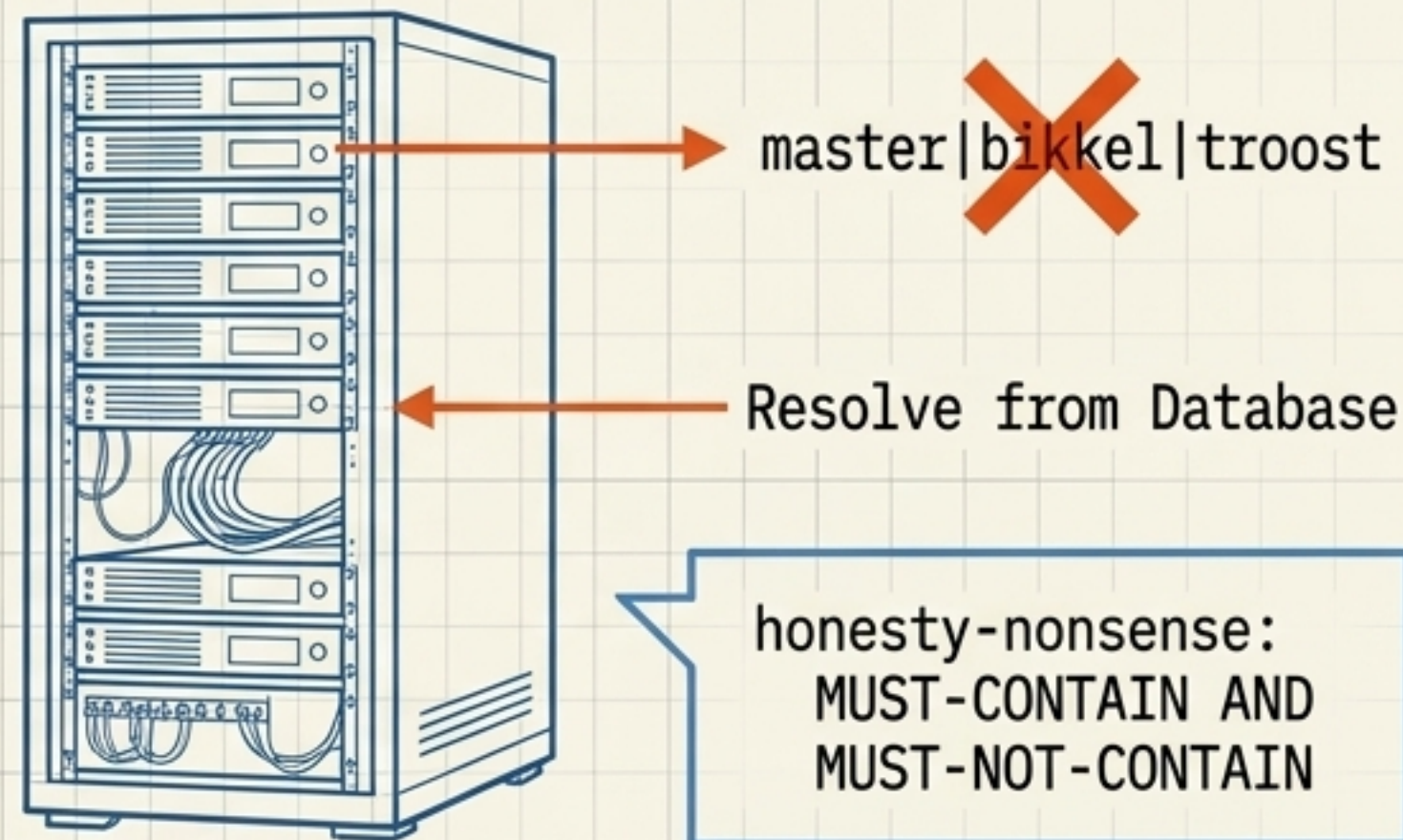
Step 2: Business Probes



- [] Name a business procedure matching a term
- [] Identify business foreign keys
- [] Name a calling procedure

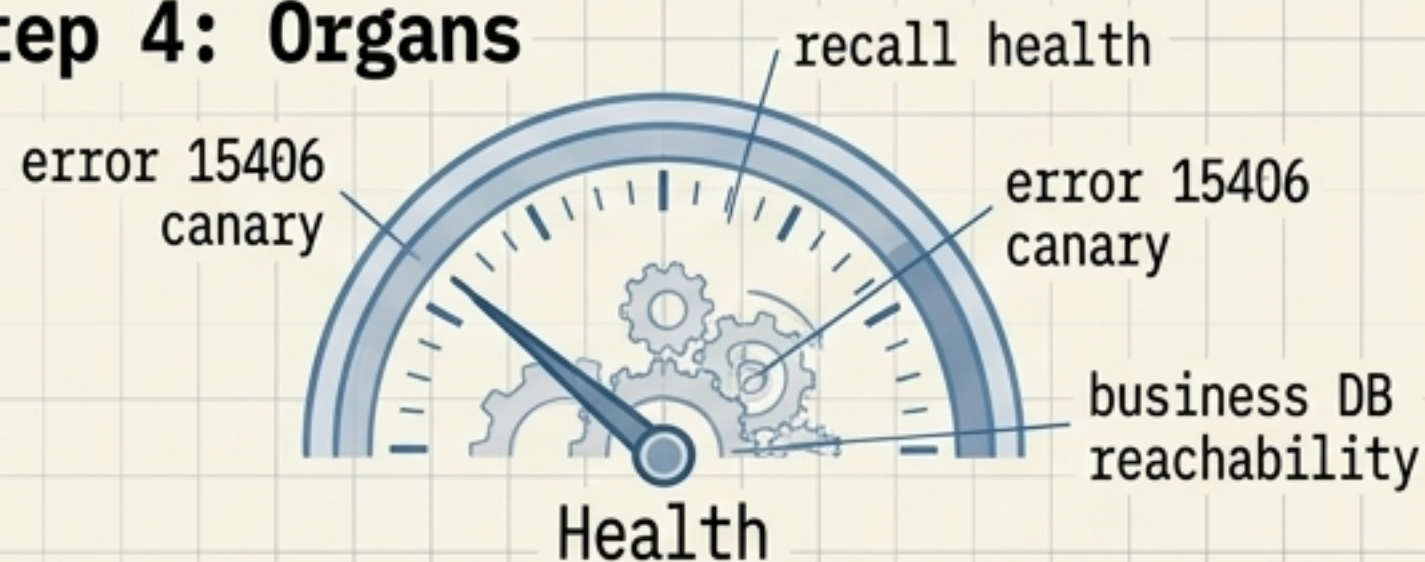
- Step 2: Deploy the business.sh regression probe set. Values must be computed live from the Solution to prevent data rot. This the test that would have caught Finding 1.
- Step 3: Unhardcode the box list. A per-Solution agent requires a per-Solution bench.
- Every FAIL must now mechanically record the exact text that reached the agent via `sp_ask_ai_review_recall`.

Step 3: Instrument Repair (Harness)



Steps 4 & 5: Grafting Organs & Behavioral Corrections

Step 4: Organs



fn_ask_ai_fishing_prompt
size

vw_ask_ai_spend_turn
average



Step 5: Corrections

Extend `sp_ask_ai_self` rather than adding standalone tools.

mode=health: Real-time condition tracking, including read path vitality and recall viability.

mode=footprint: A strict token cost and memory dilution gauge.

- Corrections are conversational, not code: Inform the agent her structural tools now span the full Solution, and correct the agent_chat memory pointer.

sp_openai_chat_post (Operator Thread)

SYSTEM MESSAGE: CORRECTIVE ACTION INITIATED.

~~NAVIGATE: query_agent_chat~~

CORRECTION: INFORM AGENT STRUCTURAL TOOLS SPAN SOLUTION. CORRECT 'agent_chat' POINTER. DO NOT USE 'query' FOR THIS CONTEXT.

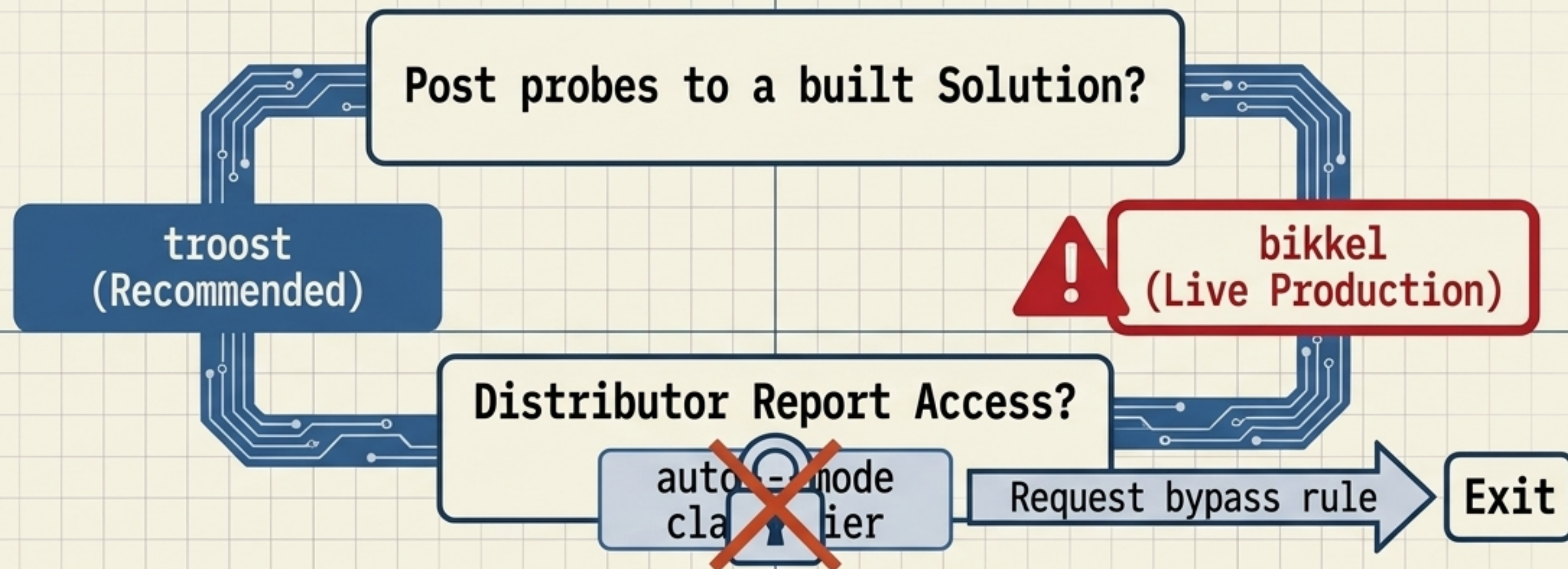
Verification Protocol: Do the Thing, Never Count



- Name resolution is deferred in SQL; objects can compile while referring to missing tables.
- Run mechanically on master first, but true verification must happen on a built Solution. **master** cannot demonstrate Finding 1.
- Read her answers, not the scores. A "must-contain" check cannot separate actual existence from a denial that quotes the target word.

Consent to Proceed: Operator Decisions Required

The central fix cannot ship unverified. Master cannot demonstrate Finding 1.



1. Permission to post permanent probes to a built Solution. Every probe becomes permanent memory in `agent_chat`. Recommendation: authorize `troost`. Explicit authorization required to touch live `bikkel`.
2. The distributor's report is currently blocked by the auto-mode classifier as a production deploy. To include the report in the loop, a specific permission rule is requested.