DECODING THE OPAQUE

REST, The Greybox Mindset, and how to reason about distributed systems without seeing the source code.

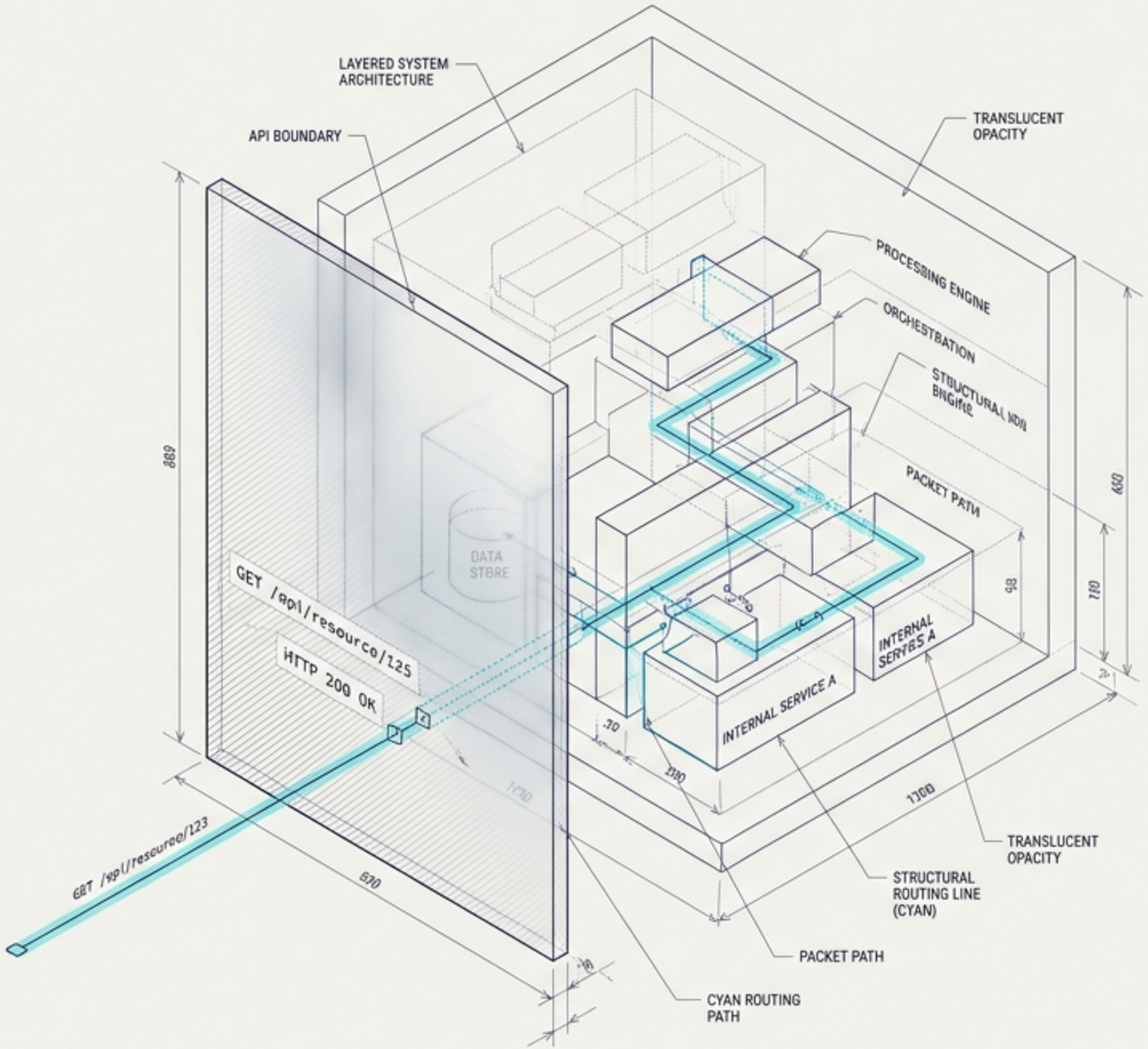
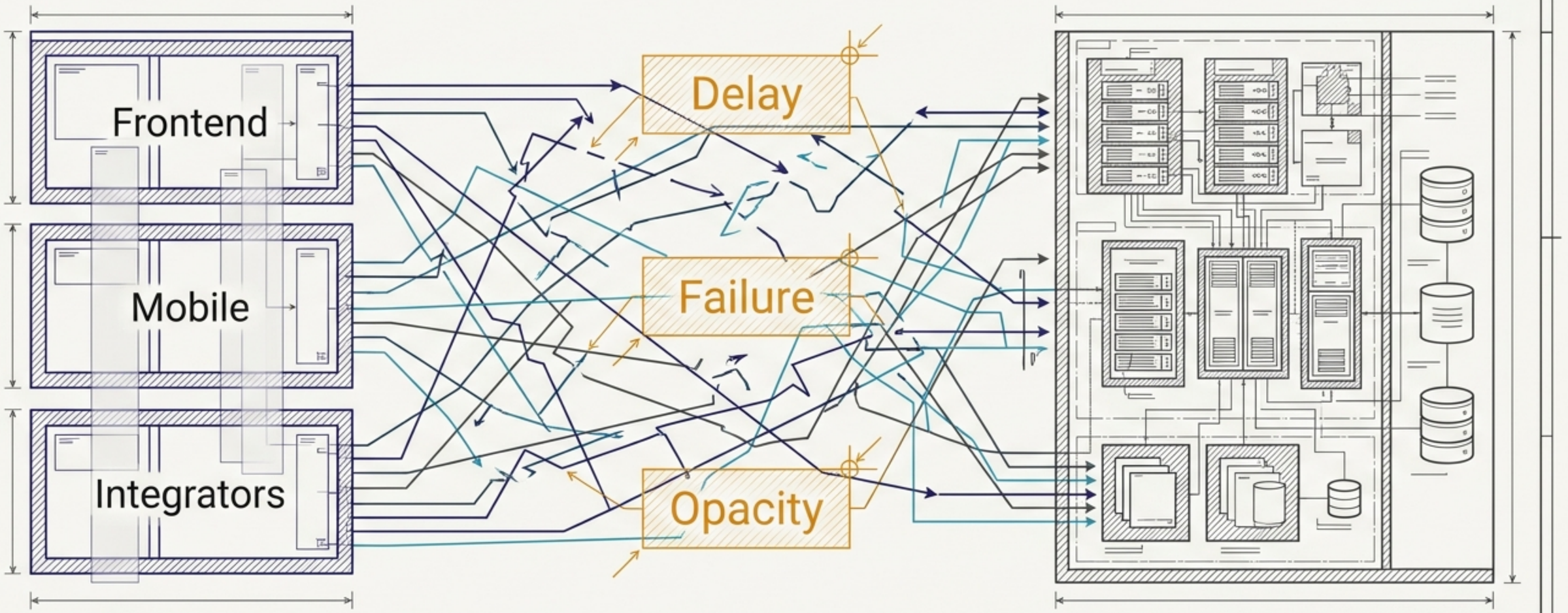


FIGURE 1: GREYBOX VISUALIZATION

Networks Introduce Ambiguity

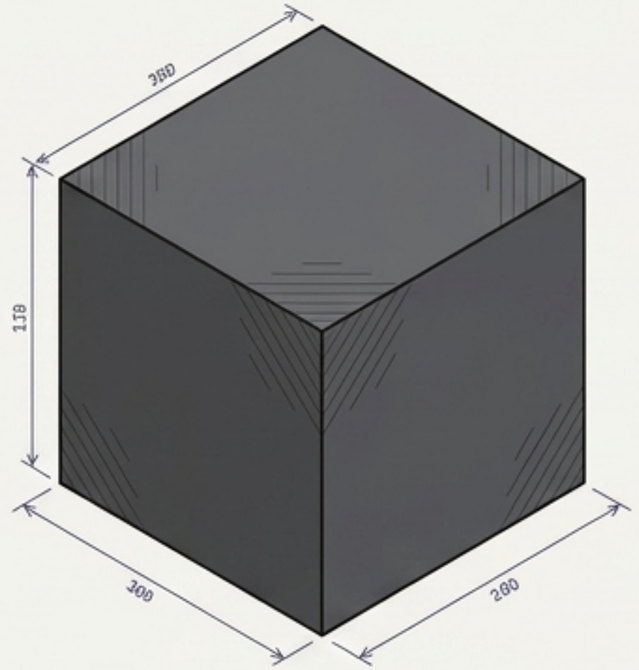
Frontend, mobile, and backend teams build independently across a void. Without a shared framework, every endpoint becomes a custom negotiation, and failures look entirely random.



The Visibility Spectrum

How we reason about a system depends entirely on how deeply we can observe it.

Blackbox (Observation)

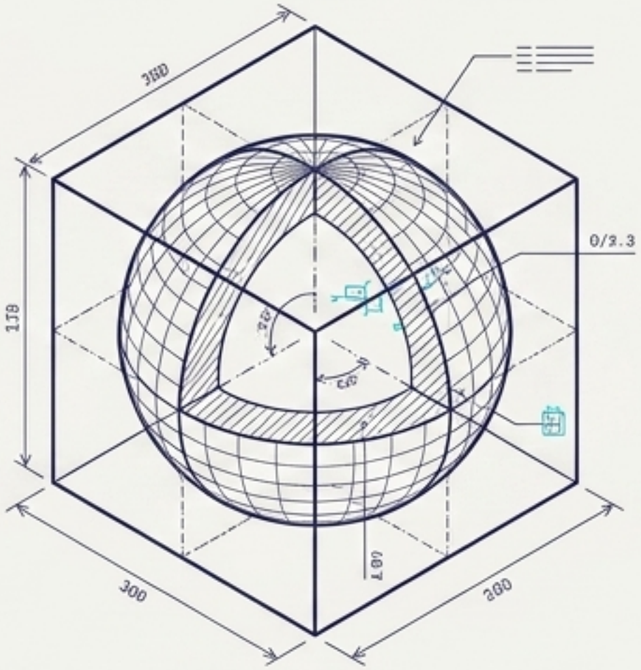


I send a request,
I get a response.

Limitation

Shallow reasoning.

Whitebox (Inspection)

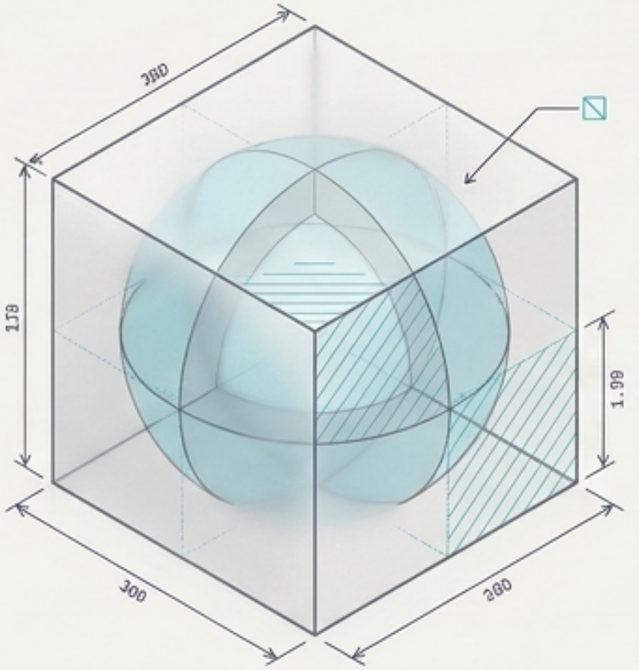


I see the code and the database.

Limitation

Rarely accessible.

Greybox (Inference)



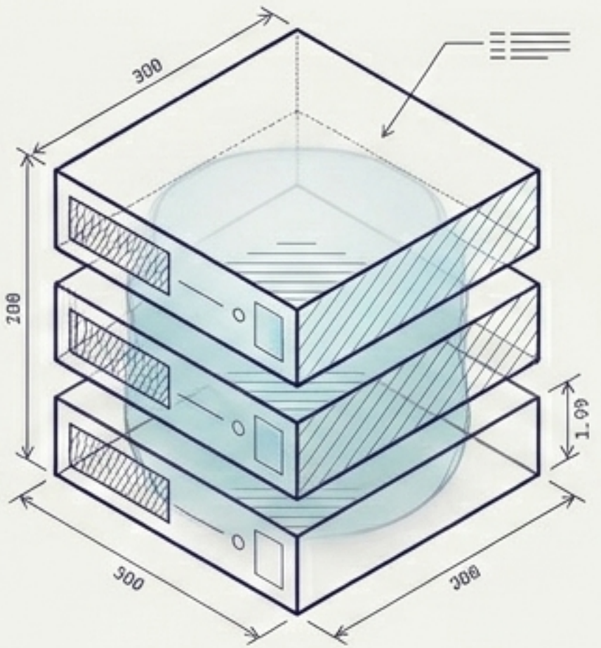
I see inputs/outputs and infer internal behavior.

Strength

Flexible and powerful.

Defining the Greybox Mindset

Most engineers live in the grey zone—consuming APIs they didn't build. Greybox thinking means you don't treat the system as a total mystery, nor do you demand the source code.



[[Visible Signals]] + [[Structural Mental Models]] = [[Internal Inference]]

REST: The Blueprint for Deduction

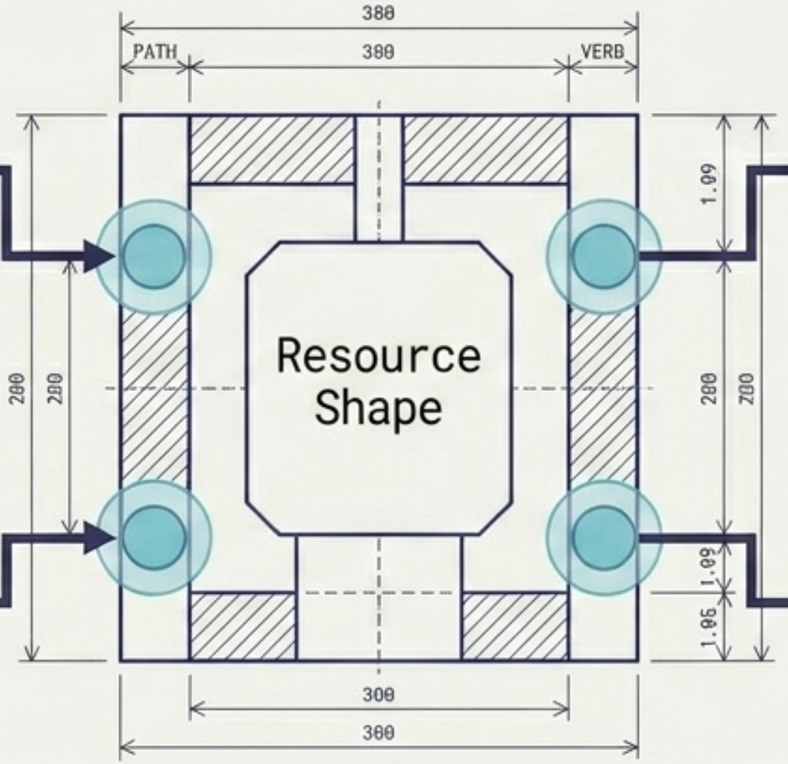
REST exists to give networked systems a shared grammar. Because the contract is standardized, engineers can infer expected behavior before inspecting a single line of backend code.

URL = Resource Identity

/users/123

Headers = Metadata

Content-Type: application/json
Content-Type: --



Method = Intent

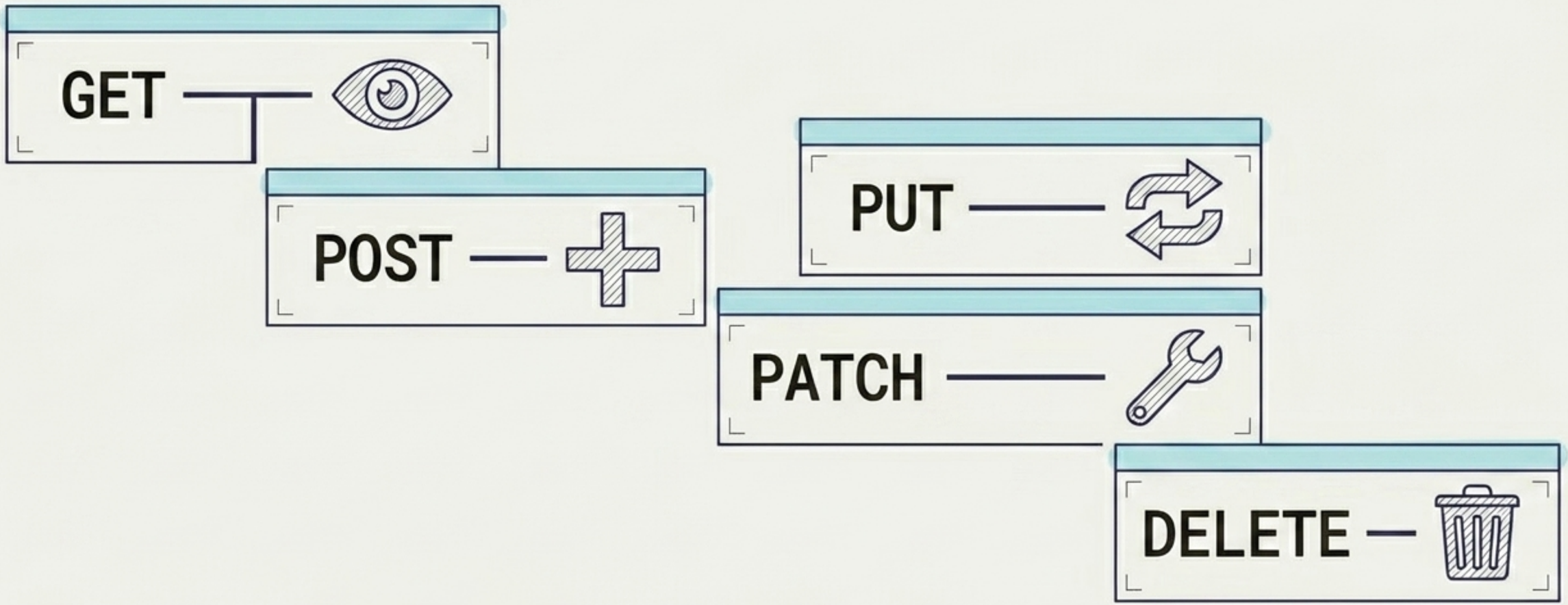
GET
POST
PUT
DELETE

Status Code = Outcome

200 OK
404 Not Found

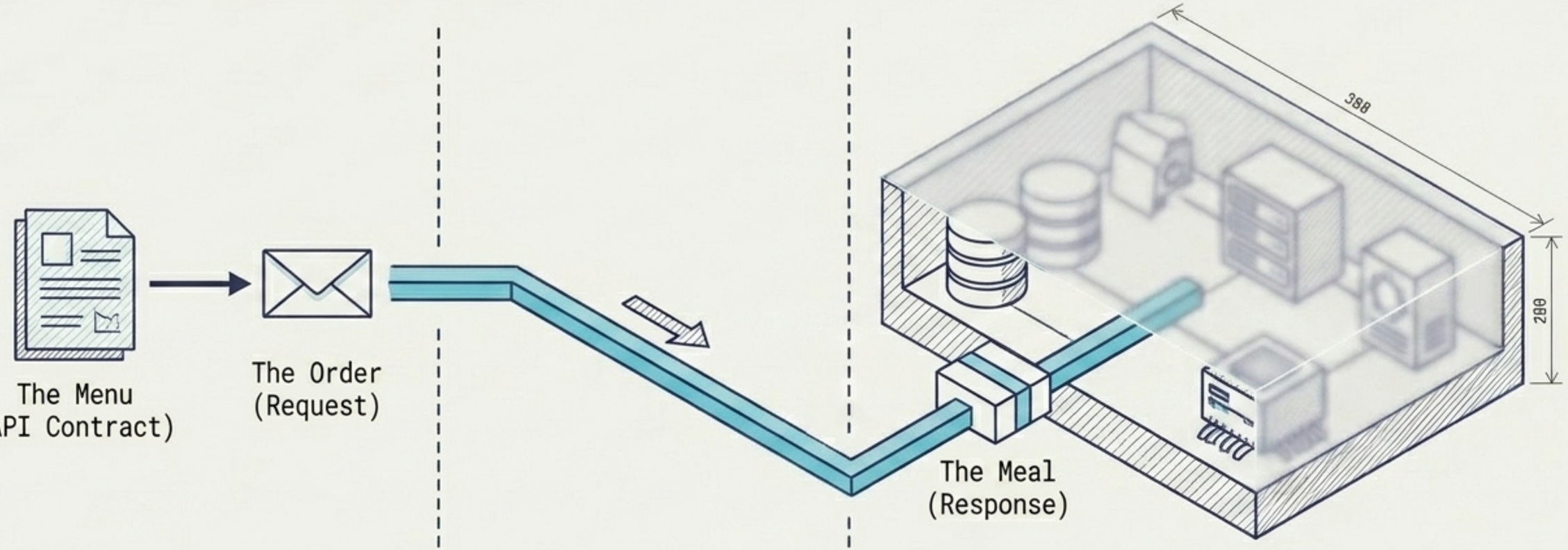
The Grammar of Intent

Standard HTTP methods eliminate guesswork by declaring exactly what the client wants to happen to the resource.



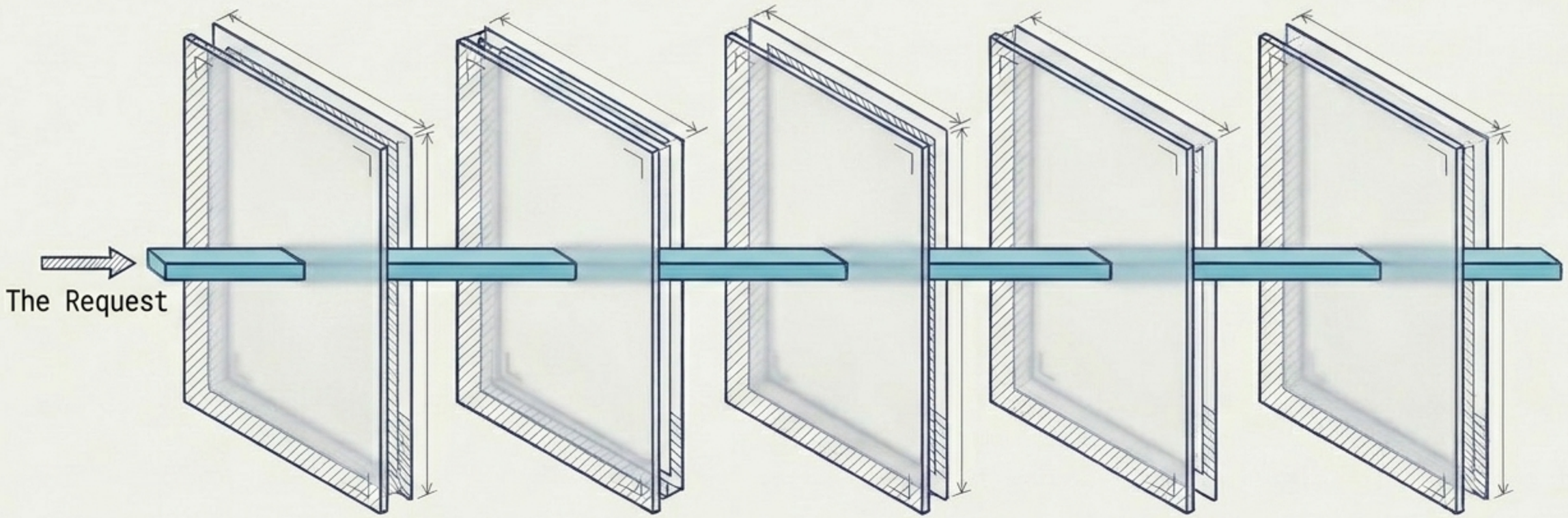
The Lifecycle of Intent

You do not stand in the kitchen, but you can infer what happened by looking at what you ordered, how long it took, and what arrived.



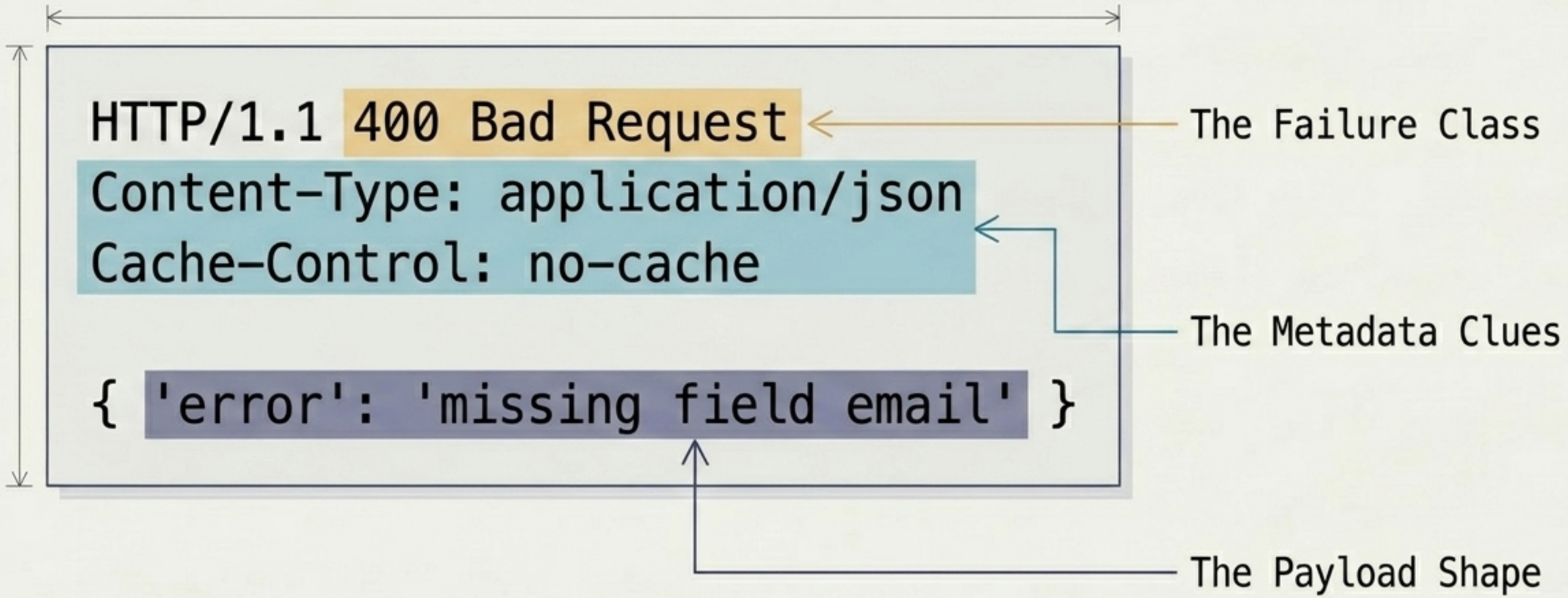
Piercing the Hidden Layers

An API is not a single wall; it is a sequence of defensive gates. Every request attempts to pierce these sequential layers to reach the core.



Responses as Forensic Evidence

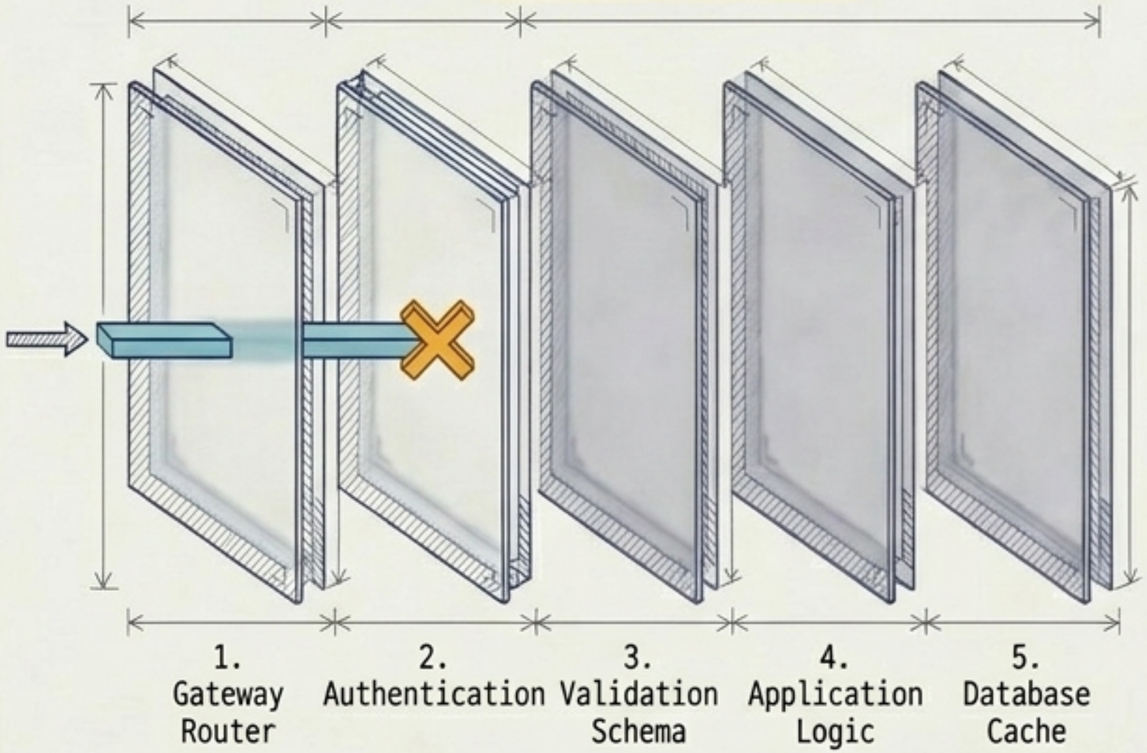
Because a request must pass through sequential rules, the final response is never random. It is hard evidence of exactly where a request succeeded or failed.



Deductive Diagnostics: The Outer Gates

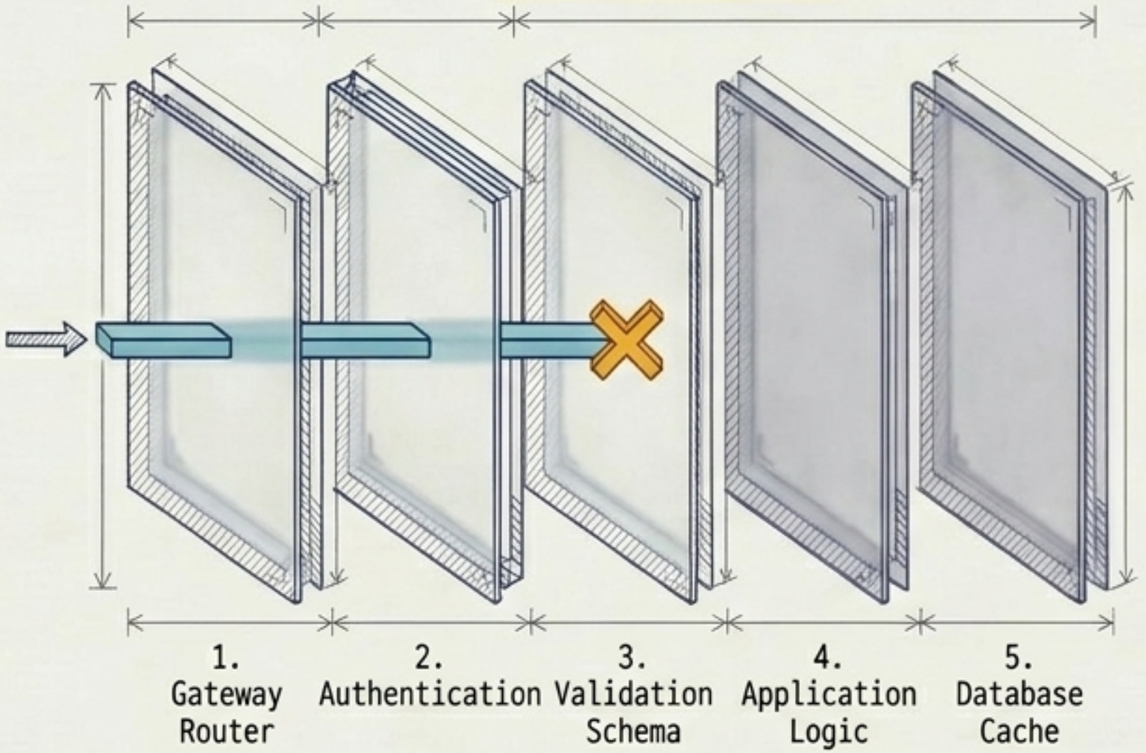
A blackbox thinker sees two failed requests. A greybox thinker sees exactly which defensive layer triggered the rejection.

401 Unauthorized



Business logic never executed.

400 Bad Request

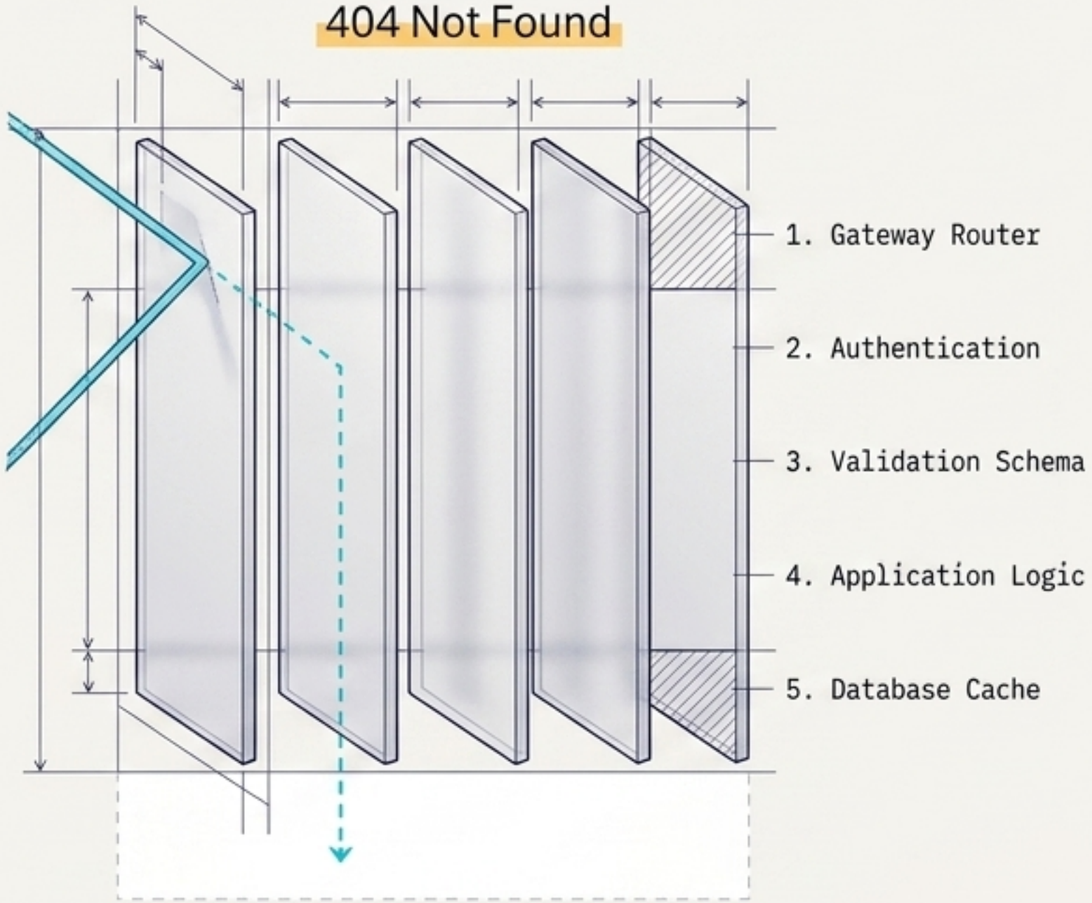


Identity confirmed, but payload shape rejected.

Deductive Diagnostics: The Inner Core

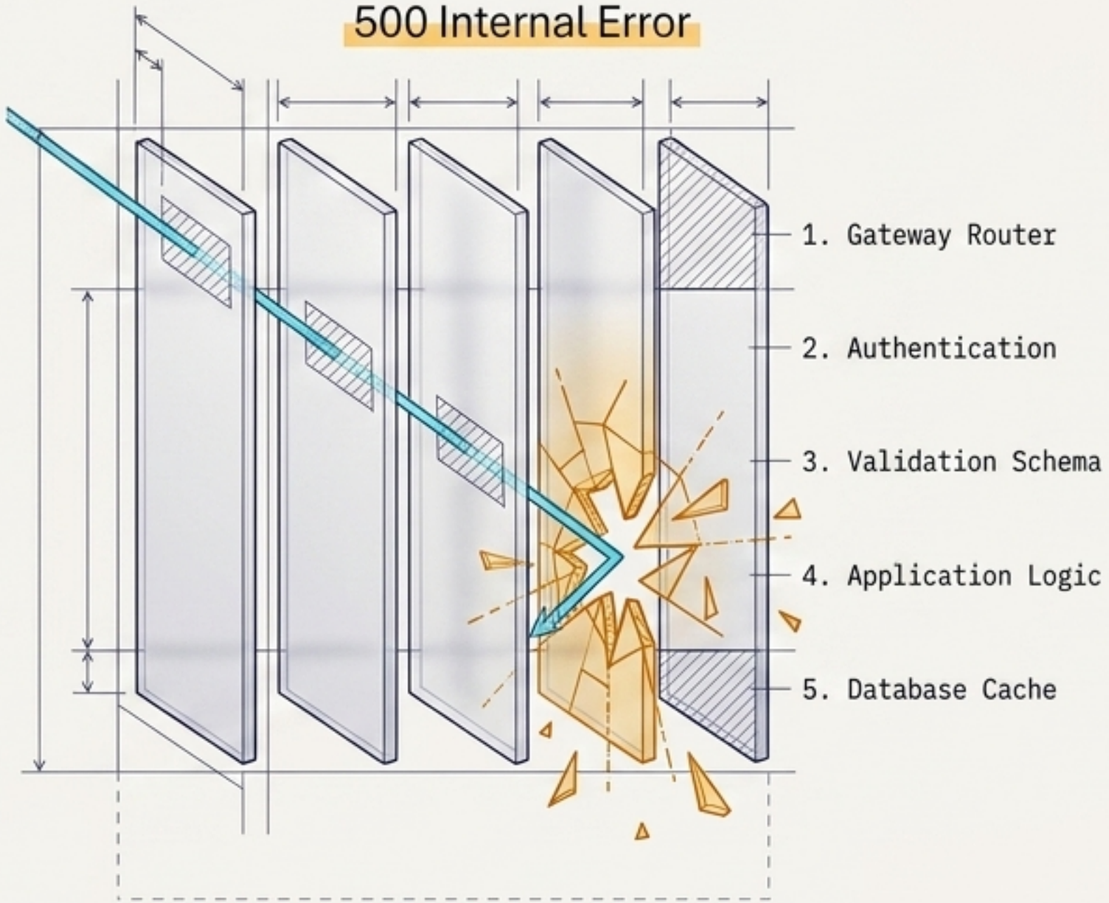
Differentiating between a missing map, a missing target, and a catastrophic system crash.

404 Not Found



Missing route or misdirected path.

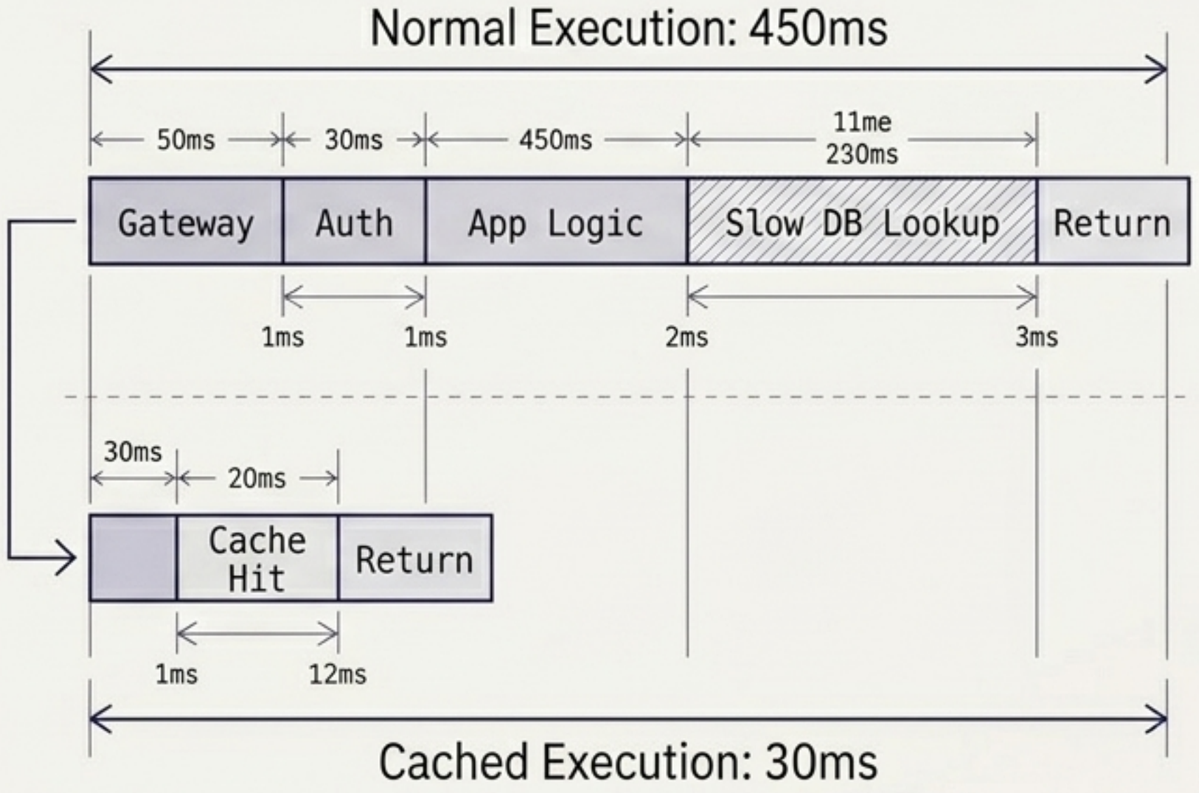
500 Internal Error

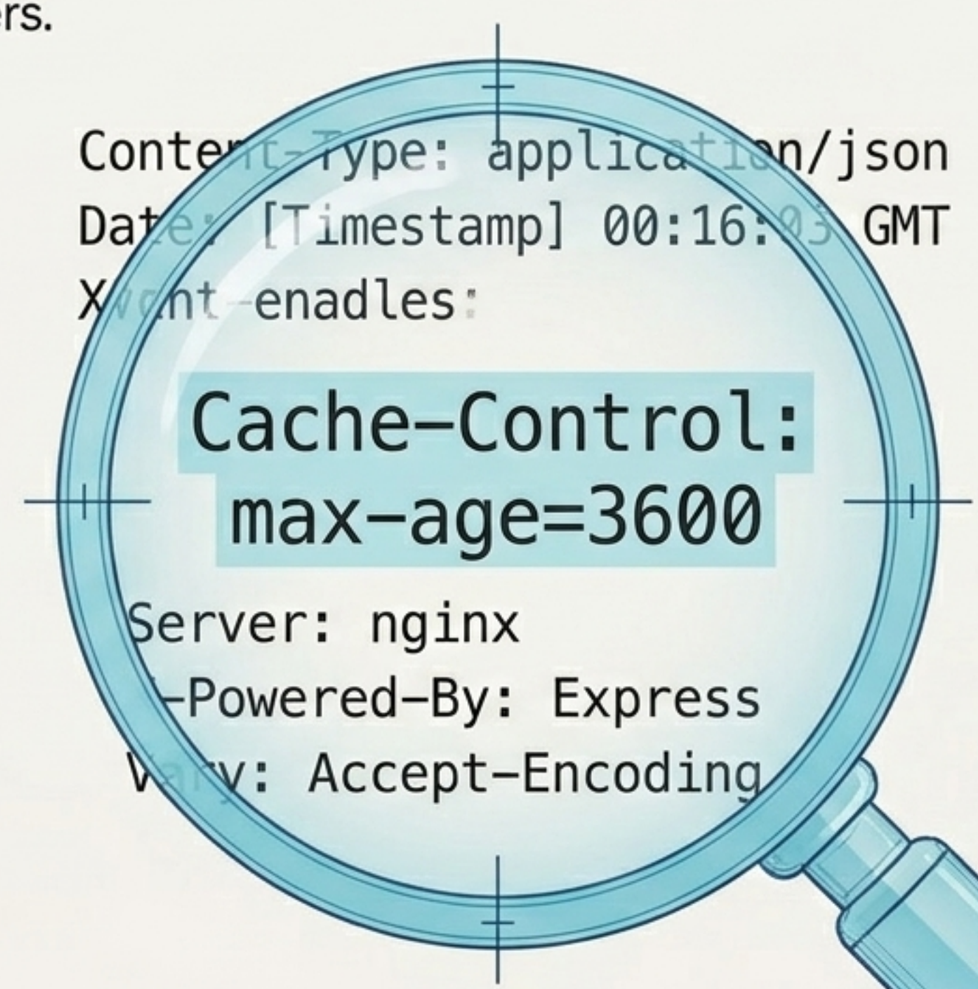


Core logic failure. Request was fully valid.

The Invisible Dimension: Time

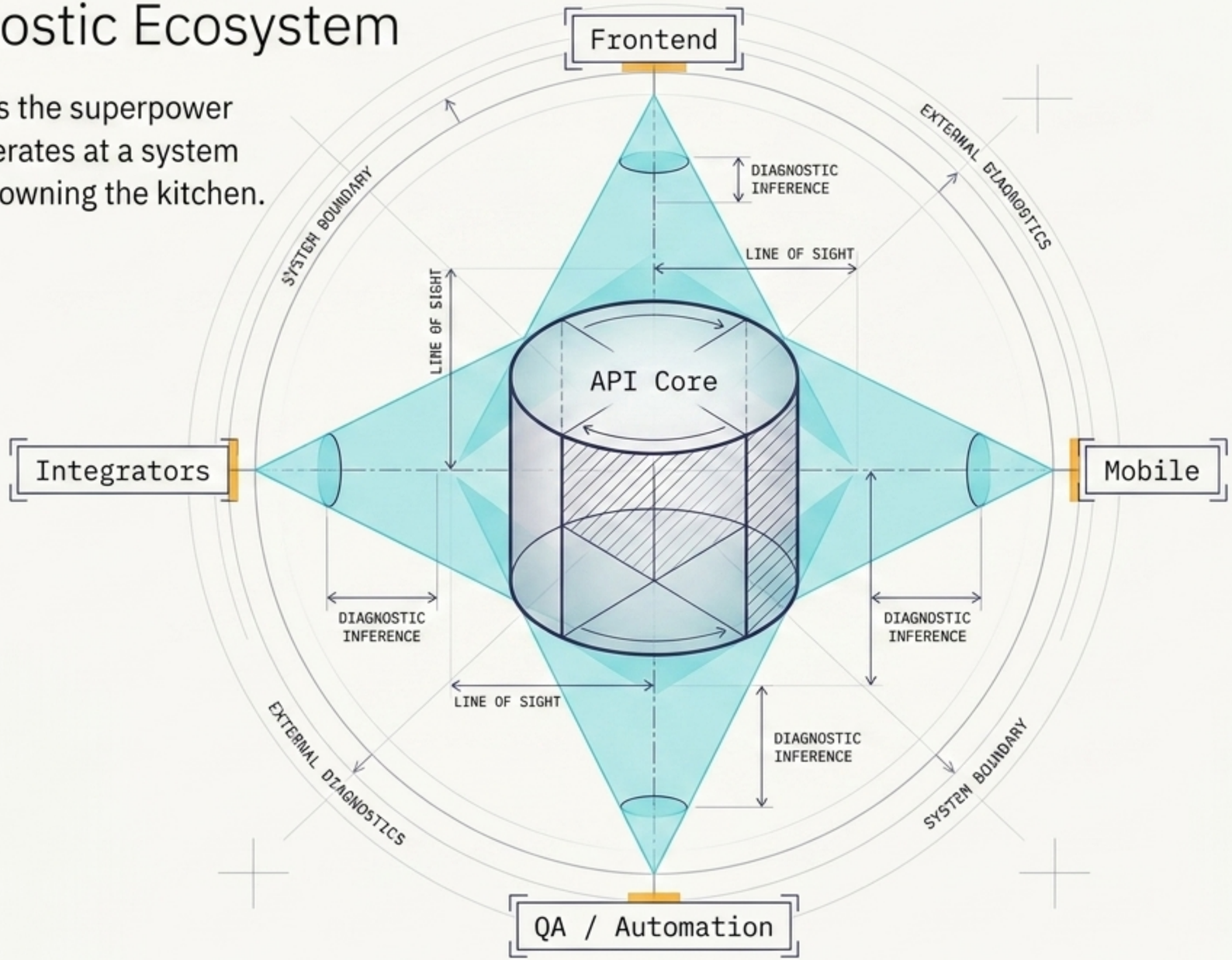
When a request returns 200 OK, latency becomes your primary clue. Slow requests imply heavy upstream dependencies; unexpectedly fast requests imply caching layers.





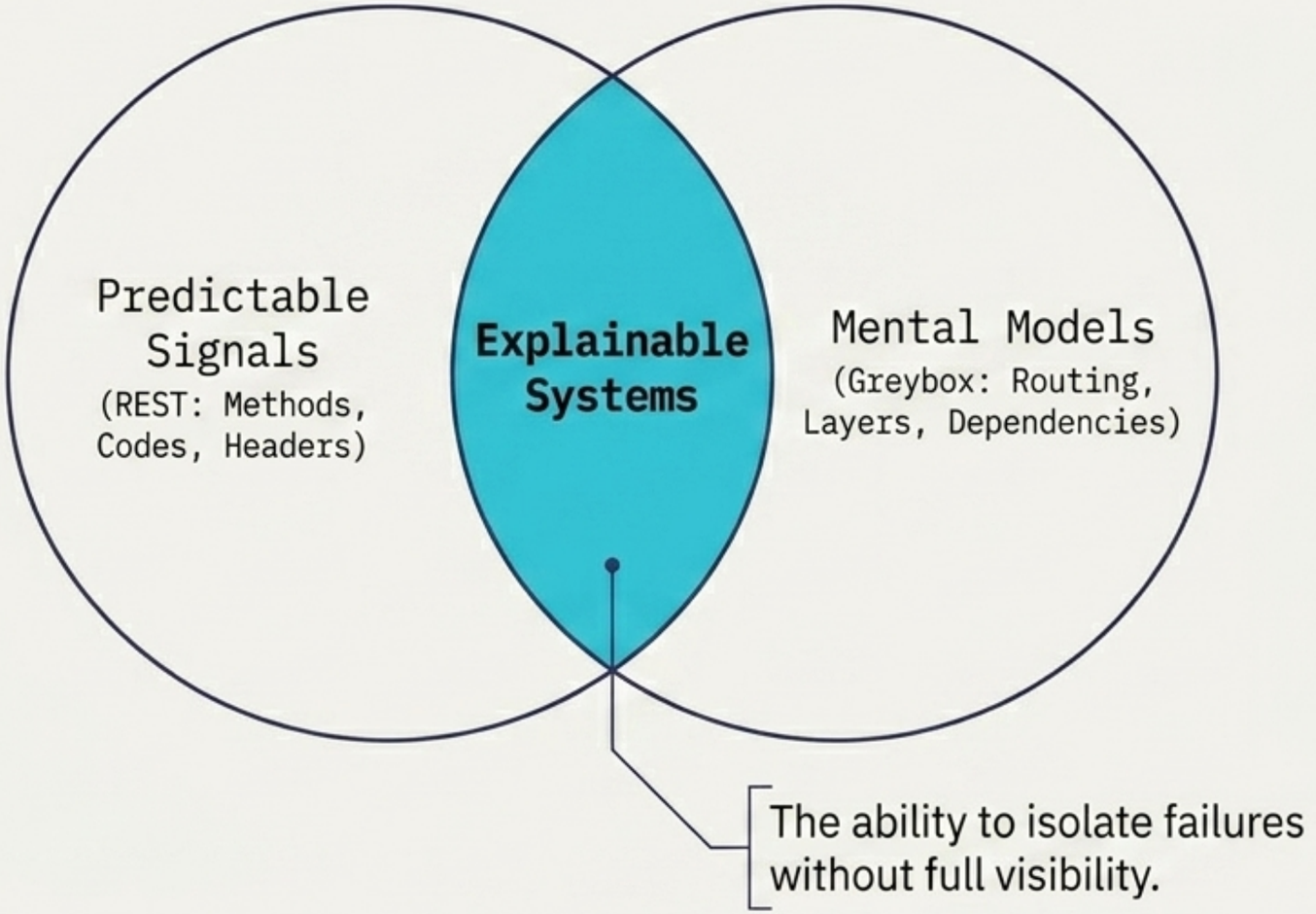
The Diagnostic Ecosystem

Greybox thinking is the superpower of anyone who operates at a system boundary without owning the kitchen.



The Predictability Framework

REST provides the predictable grammar. Greybox provides the deductive comprehension. Together, they transform opaque failures into explainable patterns.



The Greybox Manifesto

1. Treat boundaries as contracts. The URL is the target, the method is the intent.

2. Read responses as evidence. A failure code is a trace of exactly which hidden layer rejected you.

3. Stop at ‘it broke’. Infer where and why. Shift from observing symptoms to deducing mechanisms.