

# The robust way to connect AI to your legacy business logic.

## DETERMINISTIC SYSTEM-INTELLIGENCE

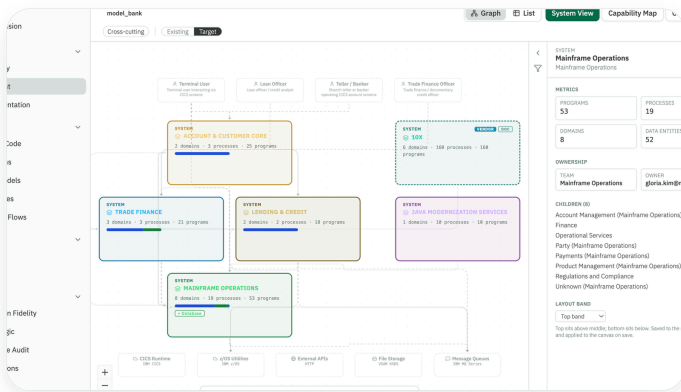
You can't replace 40 years of business logic - and you shouldn't have to. One deterministic blueprint of every program, data flow and business process - so engineering teams build agentic workflows directly on the core, with source-grounded control.

We don't guess. We determine.

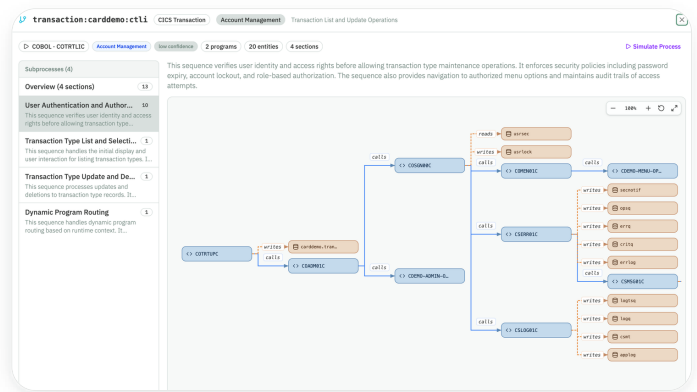
## TWEEZR'S CORE OFFERING

Deep system understanding, mapped from the code up.

**SYSTEM EXPLORER** Every architectural system, code-evidenced



**PROCESS FLOWS** Dependency and blast-radius graph



**AGENT SESSION** Source-grounded answers, every claim linked to the line of code

The Agent Session interface shows a user query: "how does billing work". The system provides a structured answer titled "Billing Process Overview". The answer explains that the billing system is managed by the "Billing and General Ledger Posting" process, which generates billing transactions and posts them to the general ledger. The core billing program is "COBIL00C", which handles bill payment entry, validation, and execution.

**How Billing Works**

- User Interaction and Validation**
  - When a user initiates a bill payment through the CICS transaction CB00, COBIL00C performs a series of validations:
    - Account Input Validation**: Ensures the account ID is not empty or invalid
    - Confirmation Validation**: Validates user confirmation (Y/N) for the payment
    - Account Status Validation**: Checks if the account status allows payments by verifying the account is not:
      - Closed

The right-hand side of the interface shows the "Card Demo System Administration" entry point, listing 19 data entities, 4 sequences, and 9 steps, with a detailed breakdown of each step.

Fig. 01 - Deterministic system intelligence, regenerated from source. No LLM in the analysis path; every node traceable to the line.

75%

TIME SAVED

Feature design, reverse-engineering, incident resolution.

2 weeks

TIME TO VALUE

Pilot to first answer you can trust.

1,800,000

ANALYZED DAILY

Non-trivial code dependencies, and counting.

## THE PLATFORM Three layers of system intelligence, built from your code up.

### LAYER 01 · IMPACT GRAPH

Walk into any change knowing exactly what it touches.

**Bottom-up from source** - every CICS transaction, batch job and program mapped into entry points, execution flows and sub-flows. Estimate blast radius, surface every caller, triage what to migrate first.

impact analysis   dependency mapping  
dead-code detection   migration triage

**Blast radius known before the change.** →

### LAYER 02 · ARCHITECTURE

Read your estate as a business architecture, not a code dump.

**Mapped to the business** - technical reality resolved into systems, domains, processes and data entities. Onboard engineers in days, document the estate for regulators, rationalise overlapping functionality.

architecture docs   engineer onboarding  
rule discovery   domain rationalisation

**One architecture, regenerated from source.** →

### LAYER 03 · AGENTIC ACCESS

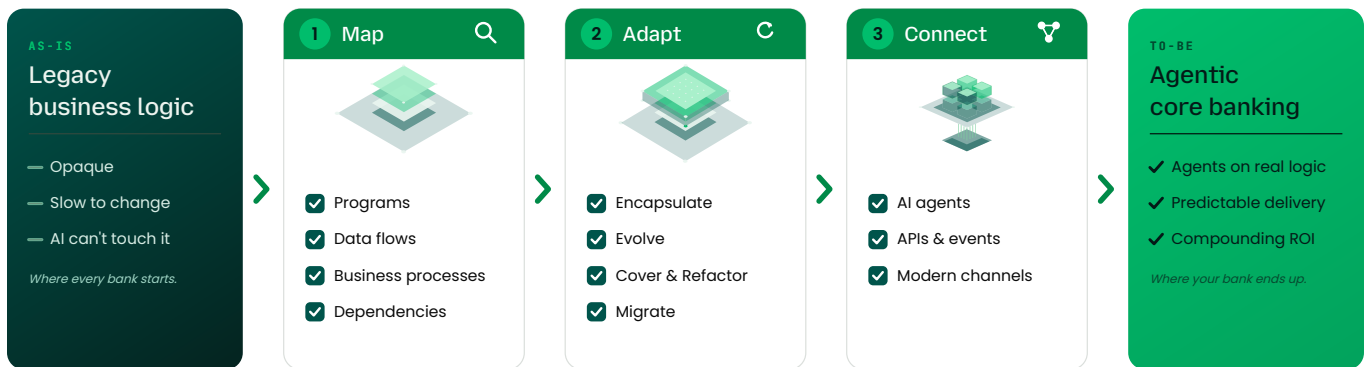
Send AI agents into the codebase, and trust the answers.

**Grounded in the blueprint** - chat sessions and agent skills resolve incidents in minutes, design features against ground truth, plan migrations with measured blast radius. Plug in Copilot, Claude Code or Q via MCP - every answer linked to source.

incident resolution   feature design  
migration planning   audit prep

**Every agent answer linked to source.** →

## THE JOURNEY From legacy business logic to agentic core banking - three deterministic moves.



"Best of show for the second year running."

AURÉLIE L'HOSTIS, FORRESTER · FINOVATEUROPE 2026

### GET STARTED

## See your own core mapped.

[Book a 30-min session →](#)

ON-PREMISE · NO DATA EGRESS

MAINFRAME Z/OS · IBM I (AS/400) · HERITAGE PLATFORMS · THIRD-PARTY BANKING CORES · MODERN LEGACY

**TWEEZR** · ON-PREMISE · AIR-GAPPED · NO DATA EGRESS