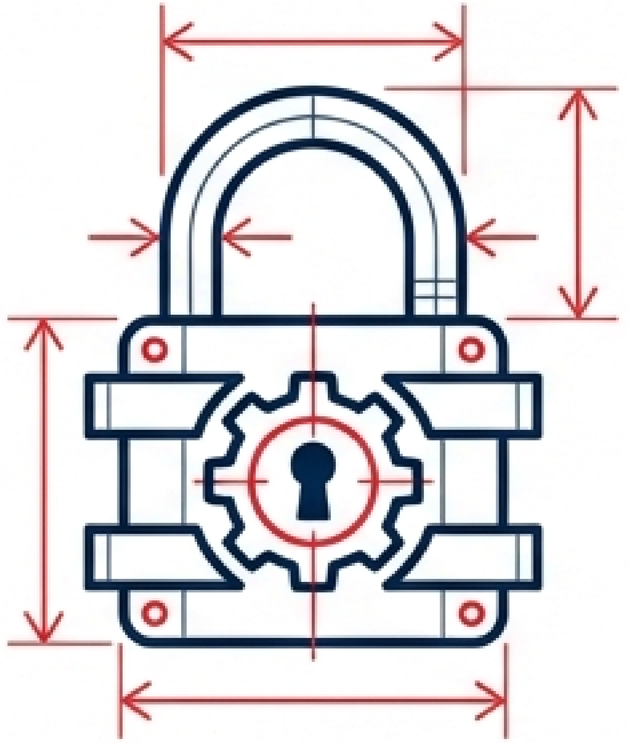
Visible Advantage: Bridging Strategy & Execution through Model Driven Architecture

From Strategic Planning Statements to Automated SQL Generation

	What (Data)	How (Function)	Where (Network)	Who (People)	When (Time)	Why (Motivation)
Scope (Planner)	Entity-relationship diagram	Business process flow	Global map with nodes nodes		Business Cycle	Business objectives
Business Model (Owner)	Semantic Data model	Business process model	Logistics network	Work flow model	Master schedule	Business plan
System Model (Designer)	Logical data model	Application architecture	Distributed system architecture	Human interface architecture	Processing structure	Business rule model
Technology Model (Builder)	Physical Database Schema	System design flow	Technology architecture	Presentation architecture	Control structure	Rule design
Detailed Representations (Subcontractor)	Data field definitions	Program control blocks	Network protocols	Security roles	Timing definition	Rule specification
Functioning Enterprise	Example Data	Example Function	Example Network	Example Organization	Example Schedule	Example Strategy

The Philosophy: One Logical, Many Physical.

Process-Driven.

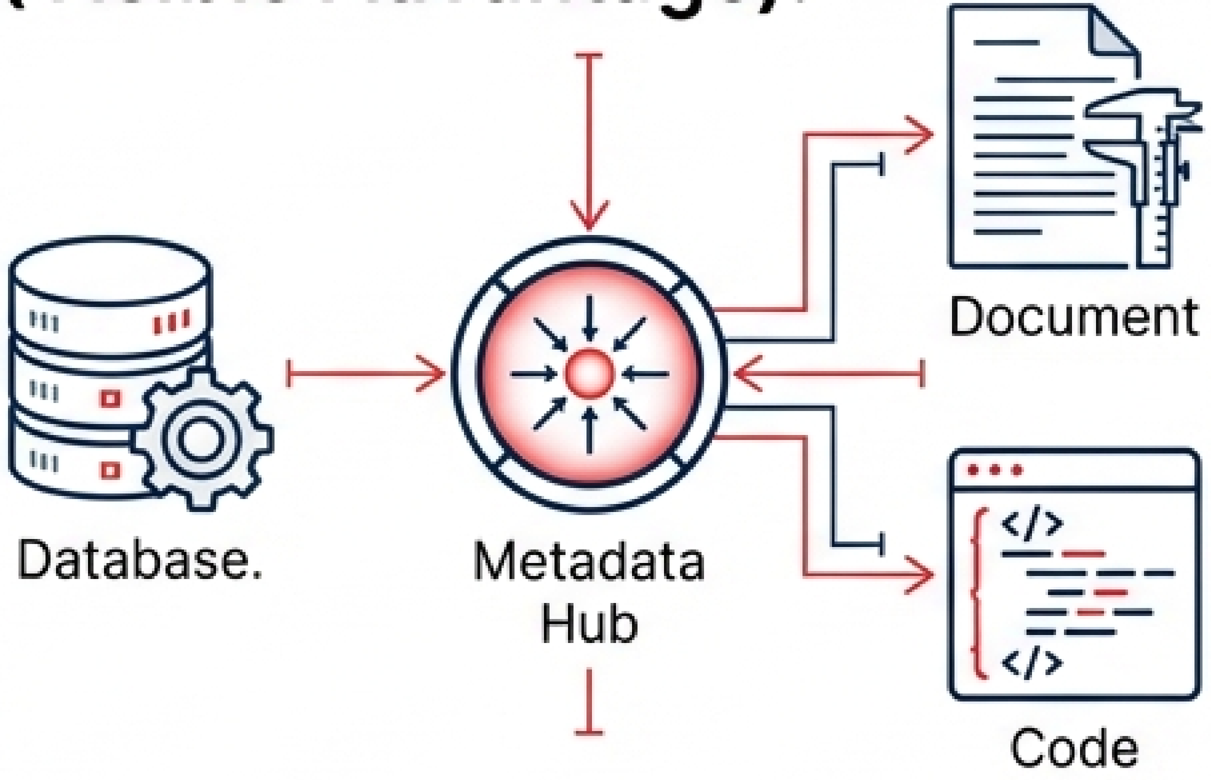


Rigorous but inflexible.
Rules hard-coded in software.

Procedure-Driven.

Flexible but lacks rigor.
Relies on users following manuals.

Model Driven Architecture (Visible Advantage).

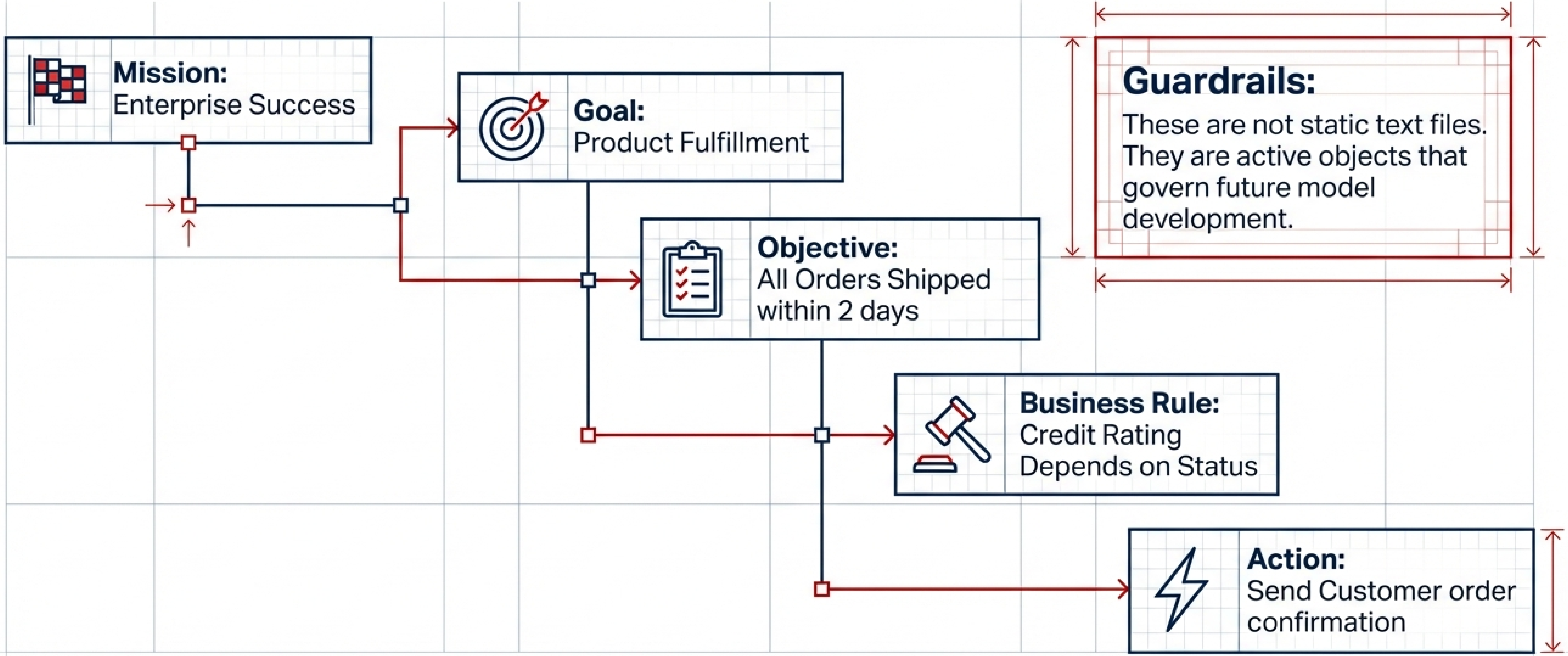


Rules documented ONCE as metadata.
Applied everywhere.

Benefit: Maximum flexibility, reduced maintenance, and systems that strictly follow business rules.

Establishing the Source of Truth.

Planning Statements as Repository Objects.



Mission: Enterprise Success

Goal: Product Fulfillment

Objective: All Orders Shipped within 2 days

Business Rule: Credit Rating Depends on Status

Action: Send Customer order confirmation

Guardrails: These are not static text files. They are active objects that govern future model development.

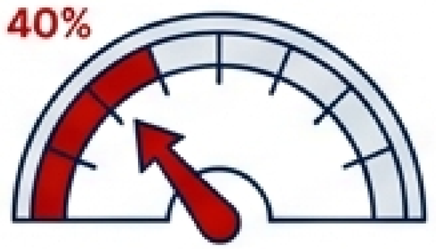
The Strategy Health Check

Objective Signals to Avoid Sunk Cost Fallacy

Market Fit

The 40% Rule

If <40% of users would be very disappointed without the product, fit is lacking.



Growth Efficiency

LTV:CAC Ratio

Target: 3:1. If CAC > LTV, the model is unsustainable.

Financial Runway

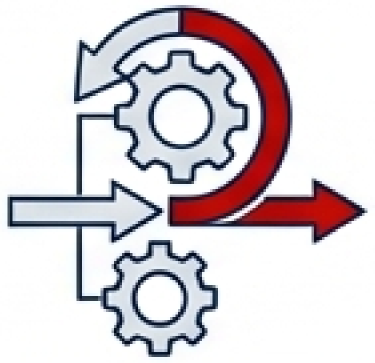
Burn Rate

If cash on hand < 6 months, immediate pivot required.

Internal Alignment

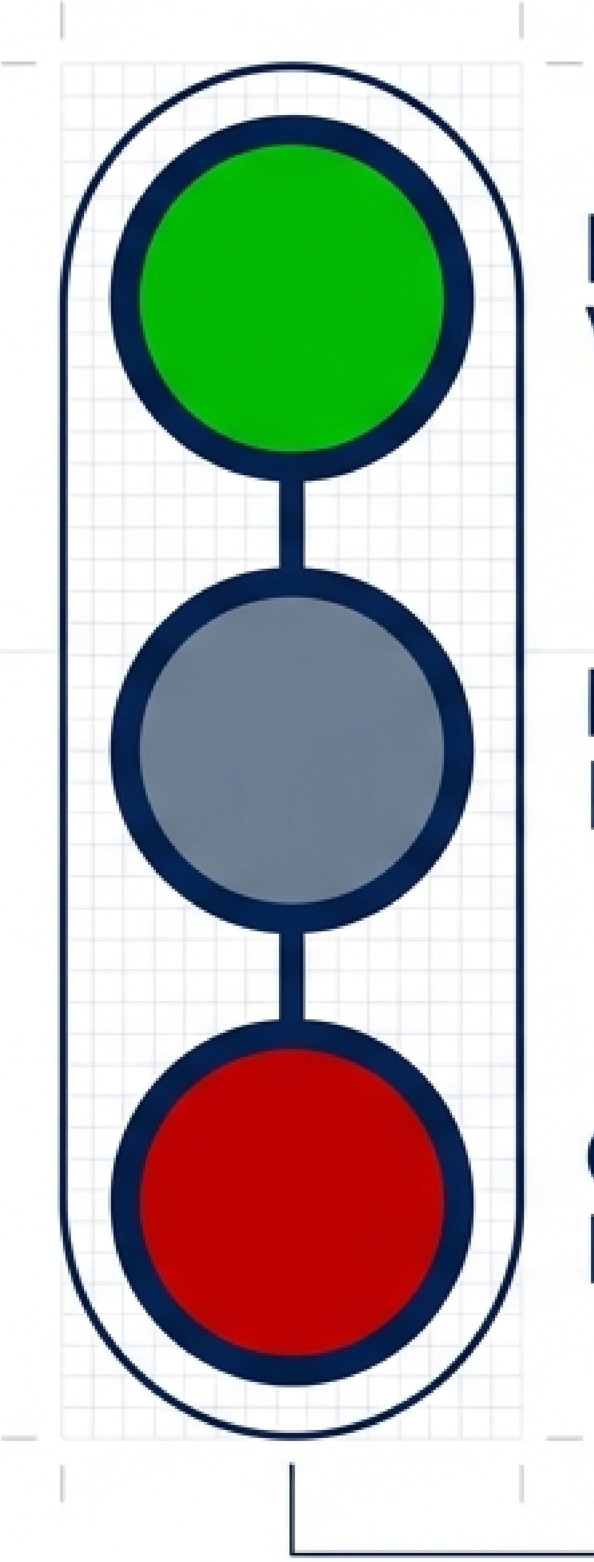
Shadow Strategy

Teams bypassing process indicates unworkable official strategy.



🟢 Execute 🟡 Root Cause Analysis 🔴 Pivot Required

The Pivot Framework.

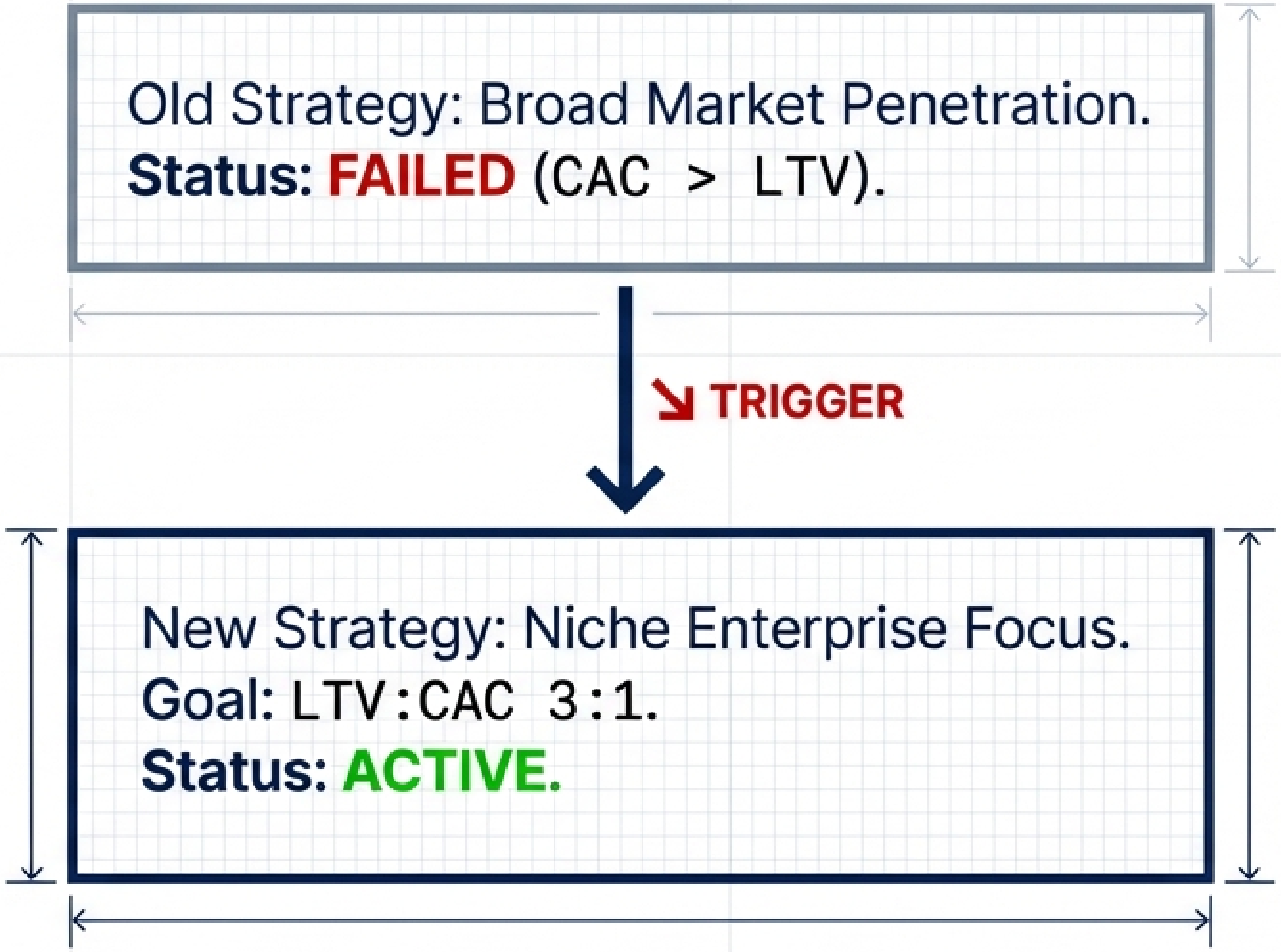


KEEP:
Valid Vision.

KILL:
Failing Tactics.

CHANGE:
New Direction.

Before & After of Planning Statement objects



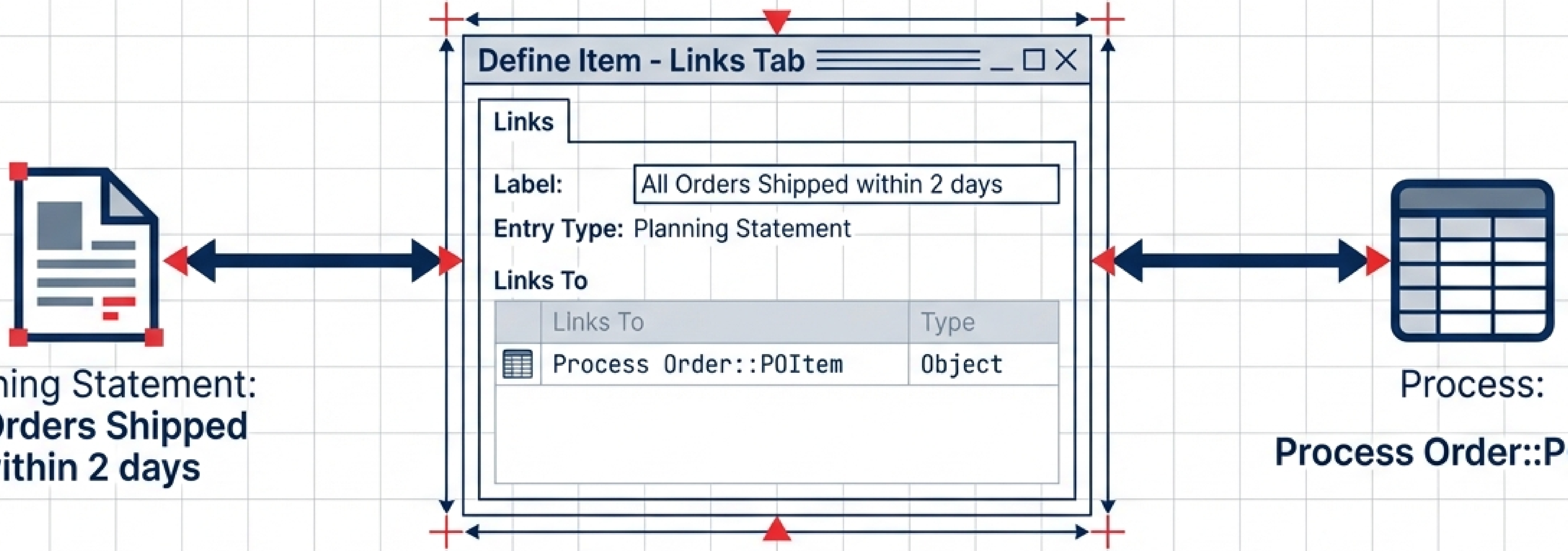
Old Strategy: Broad Market Penetration.
Status: FAILED (CAC > LTV).

↘ TRIGGER

New Strategy: Niche Enterprise Focus.
Goal: LTV:CAC 3:1.
Status: ACTIVE.

Pivot is documented in the Strategic Planning window first, aligning all downstream models.

Linking Vision to Reality: Traceability



Define Item - Links Tab

Links

Label: All Orders Shipped within 2 days

Entry Type: Planning Statement

Links To

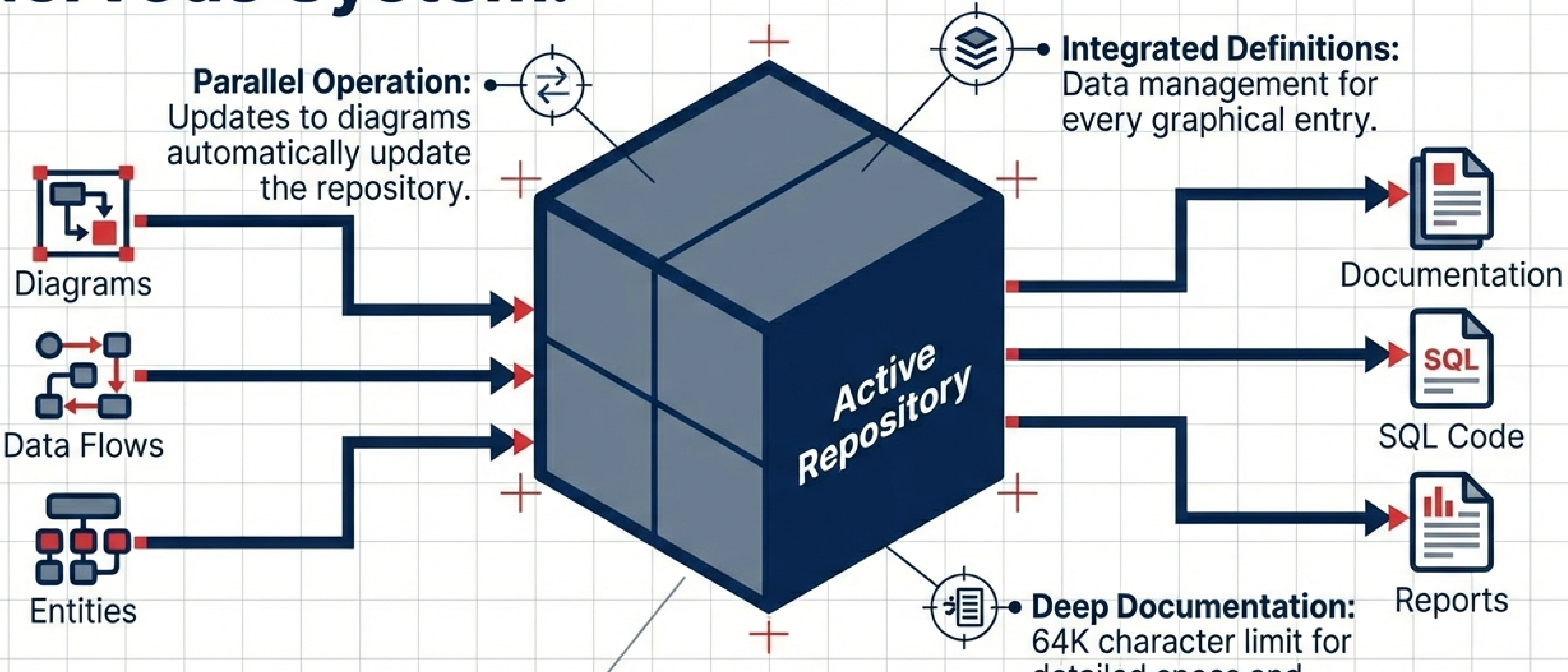
	Links To	Type
▦	Process Order::POItem	Object

Planning Statement:
All Orders Shipped within 2 days

Process:
Process Order::POItem

Traceability: If the strategy changes, we instantly know which software components must be updated.

The Repository: The Project's Central Nervous System.



Parallel Operation: Updates to diagrams automatically update the repository.

Integrated Definitions: Data management for every graphical entry.

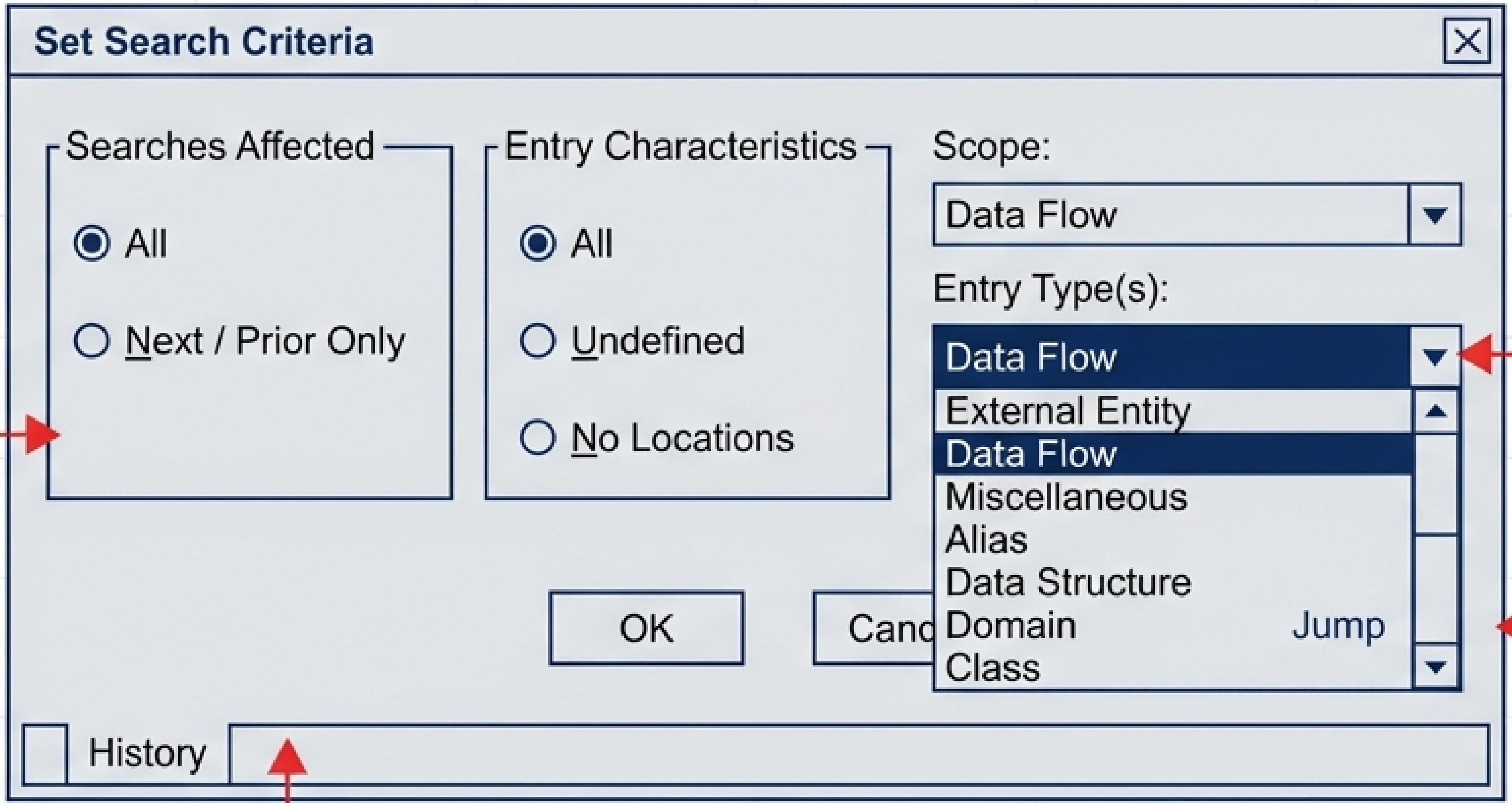
Deep Documentation: 64K character limit for detailed specs and hyperlinks.

Education versions: Add items to diagram first.

NotebookLM

Navigating Complexity

Search, Scope, and Jump



Set Search Criteria

Searches Affected
- ● All
- ○ Next / Prior Only

Entry Characteristics
- ● All
- ○ Undefined
- ○ No Locations

Scope: Data Flow

Entry Type(s): Data Flow

Search Criteria: 1 Cut through the noise by limiting scope (e.g., only Data Flows).

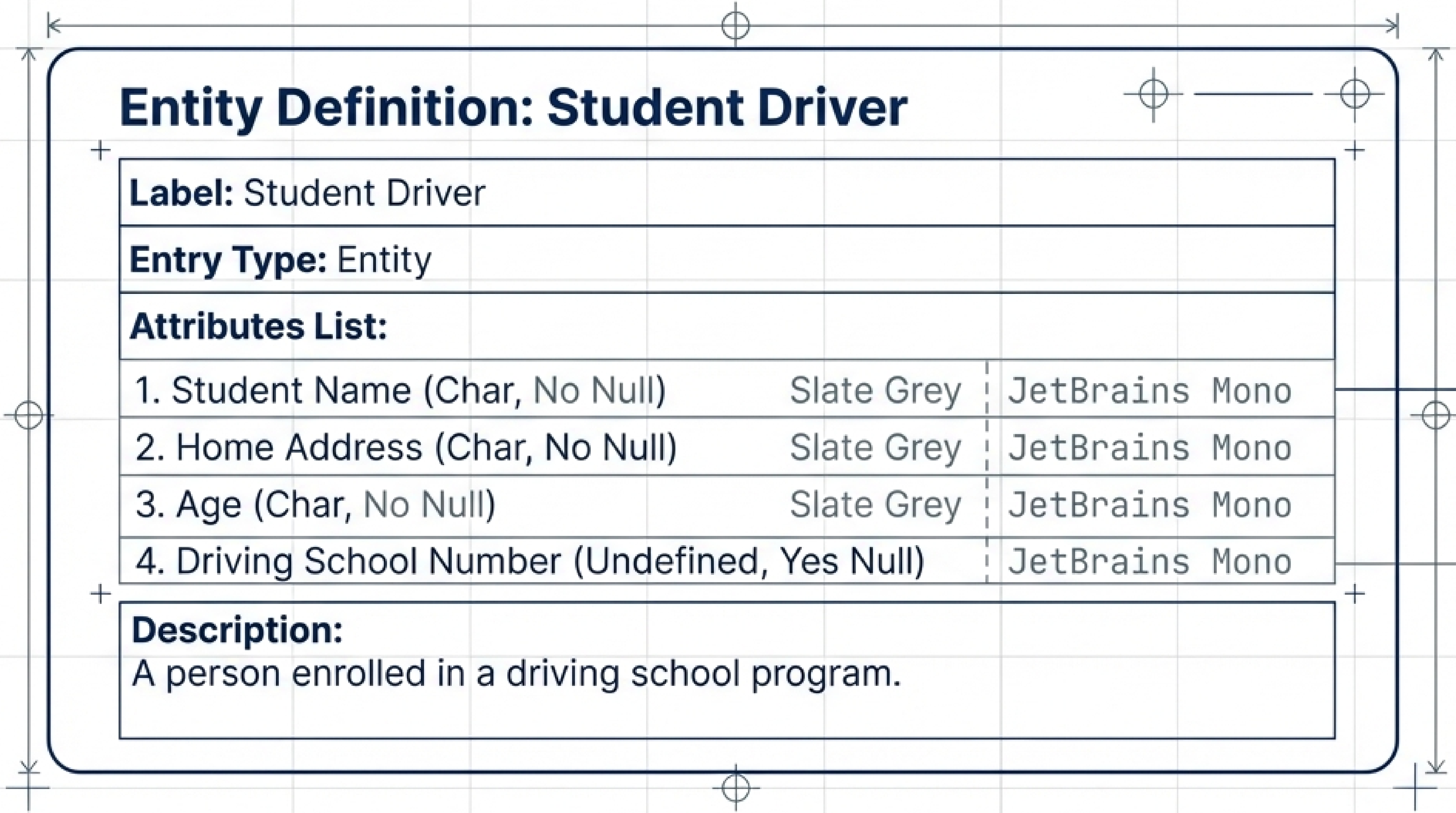
2 **Jump Button:** Hyper-navigation to instantly view related object definitions.

3 **History:** Tracks the last 500 objects for quick backtracking.

NotebookLM

Defining the Data

From Real-World Object to Repository Entity



Entity Definition: Student Driver

Label: Student Driver

Entry Type: Entity

Attributes List:

1. Student Name (Char, No Null)	Slate Grey	JetBrains Mono
2. Home Address (Char, No Null)	Slate Grey	JetBrains Mono
3. Age (Char, No Null)	Slate Grey	JetBrains Mono
4. Driving School Number (Undefined, Yes Null)		JetBrains Mono

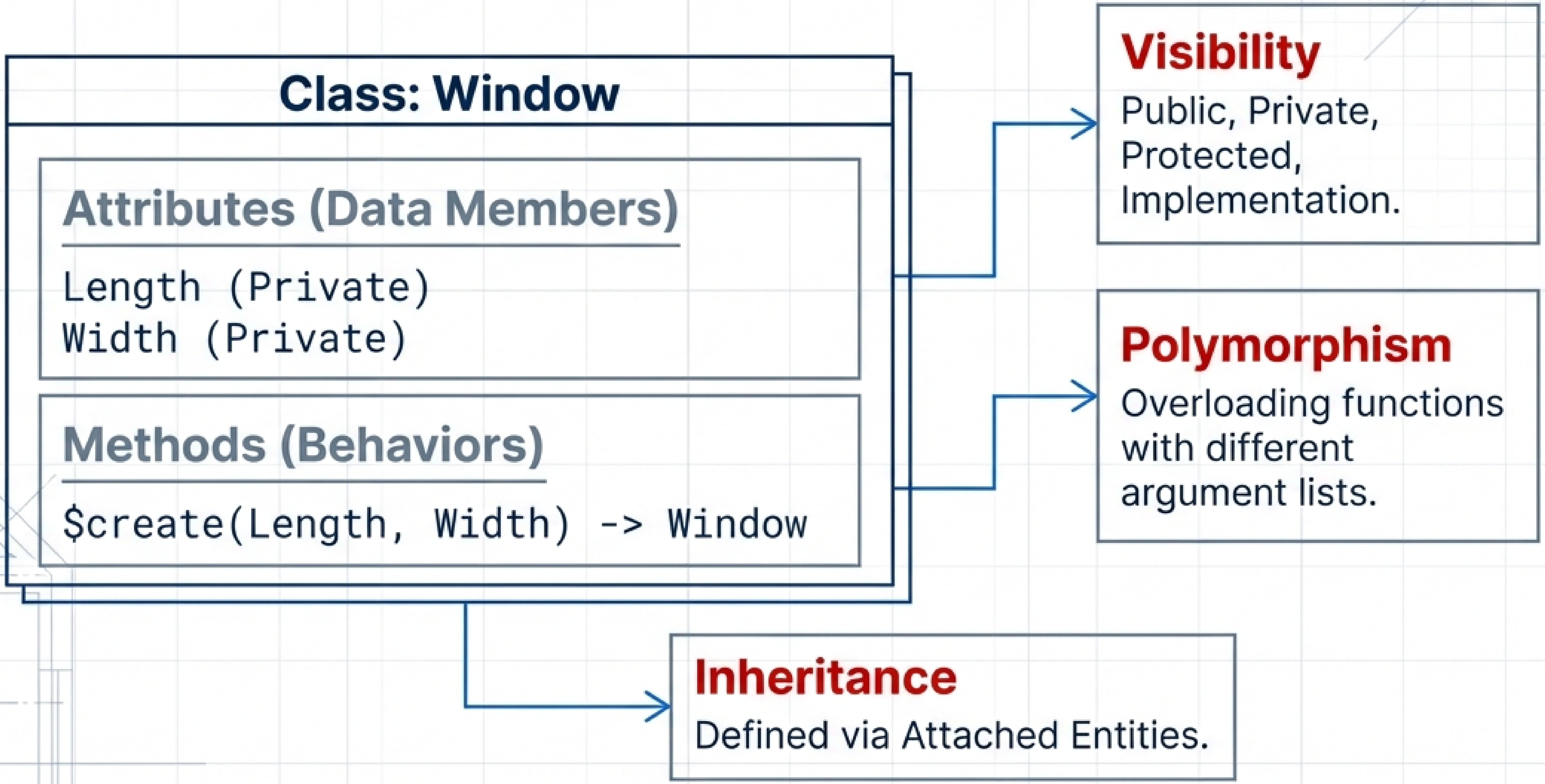
Description:
A person enrolled in a driving school program.

Granular Control: Define attributes, data types, and nullability.

Use ‘Values & Meanings' for business logic (up to 64K chars).

Object-Oriented Modeling

Encapsulating Logic with Classes & Methods



Class: Window

Attributes (Data Members)

```
Length (Private)
Width (Private)
```

Methods (Behaviors)

```
$create(Length, Width) -> Window
```

Visibility

Public, Private, Protected, Implementation.

Polymorphism

Overloading functions with different argument lists.

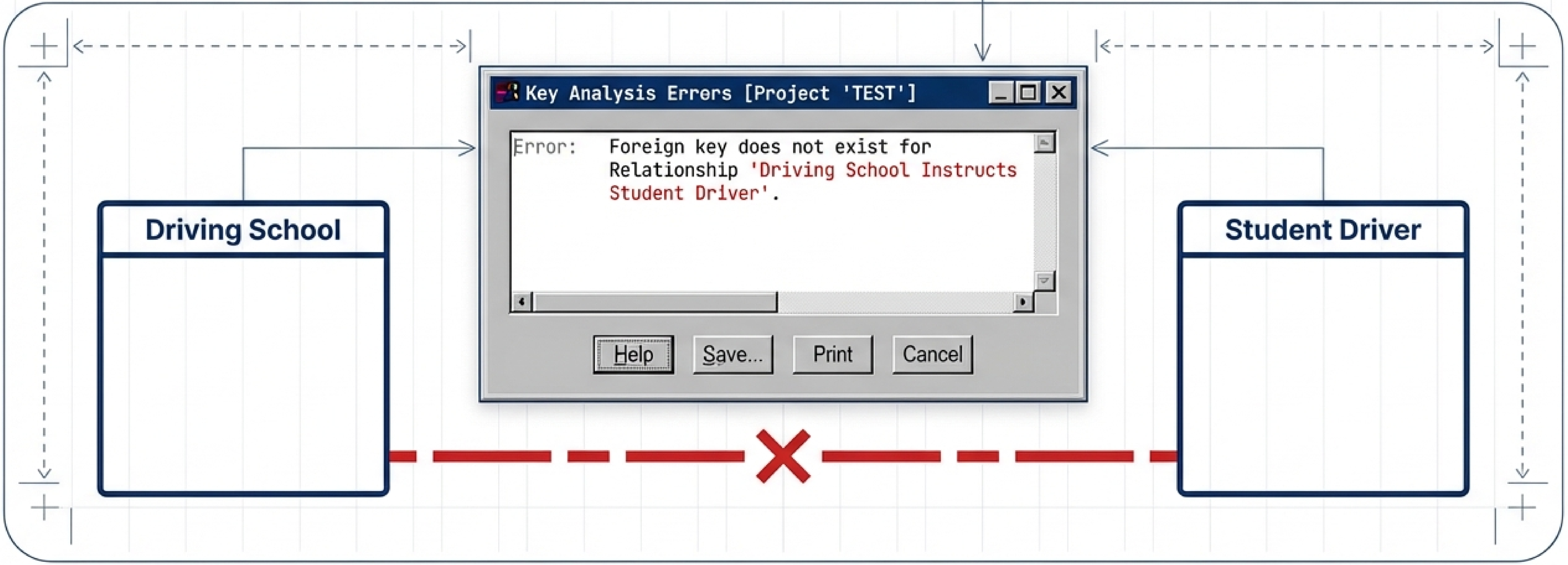
Inheritance

Defined via Attached Entities.

Ensuring Integrity: Automated Key Analysis.

The Sanity Check.

Before code generation, the tool scans the repository to flag disconnected relationships and missing keys.

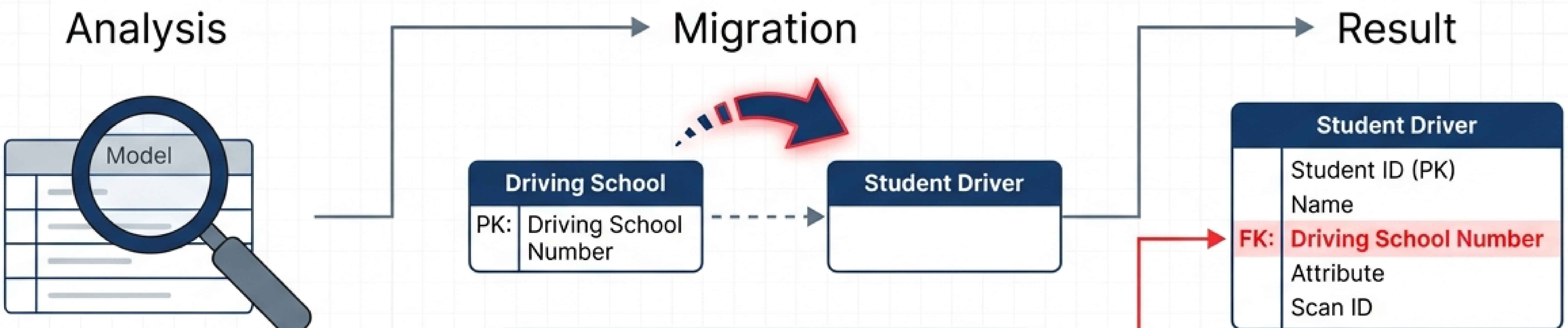


Key Analysis Errors [Project 'TEST']

```
Error:    Foreign key does not exist for
          Relationship 'Driving School Instructs
          Student Driver'.
```

Automating Consistency: Key Synchronization

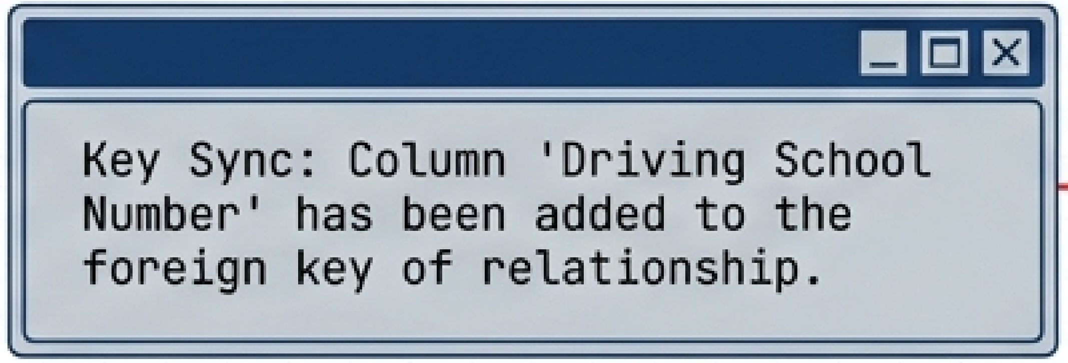
The Key Sync Process



Driving School	
PK:	Driving School Number

Student Driver

Student Driver	
	Student ID (PK)
	Name
FK:	Driving School Number
	Attribute
	Scan ID



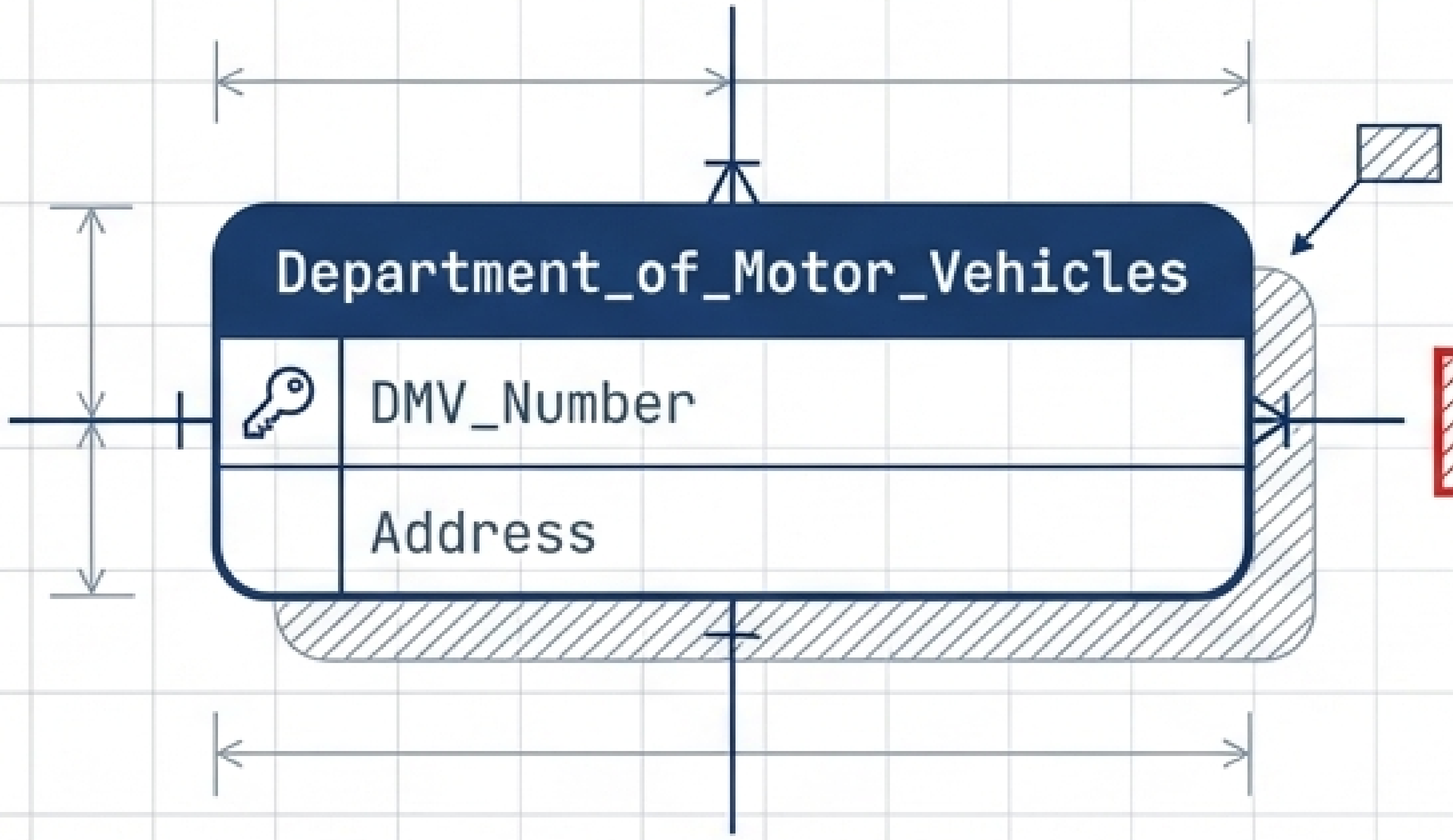
```
Key Sync: Column 'Driving School
Number' has been added to the
foreign key of relationship.
```

Productivity Booster: The system automatically migrates keys across relationships, saving manual effort.

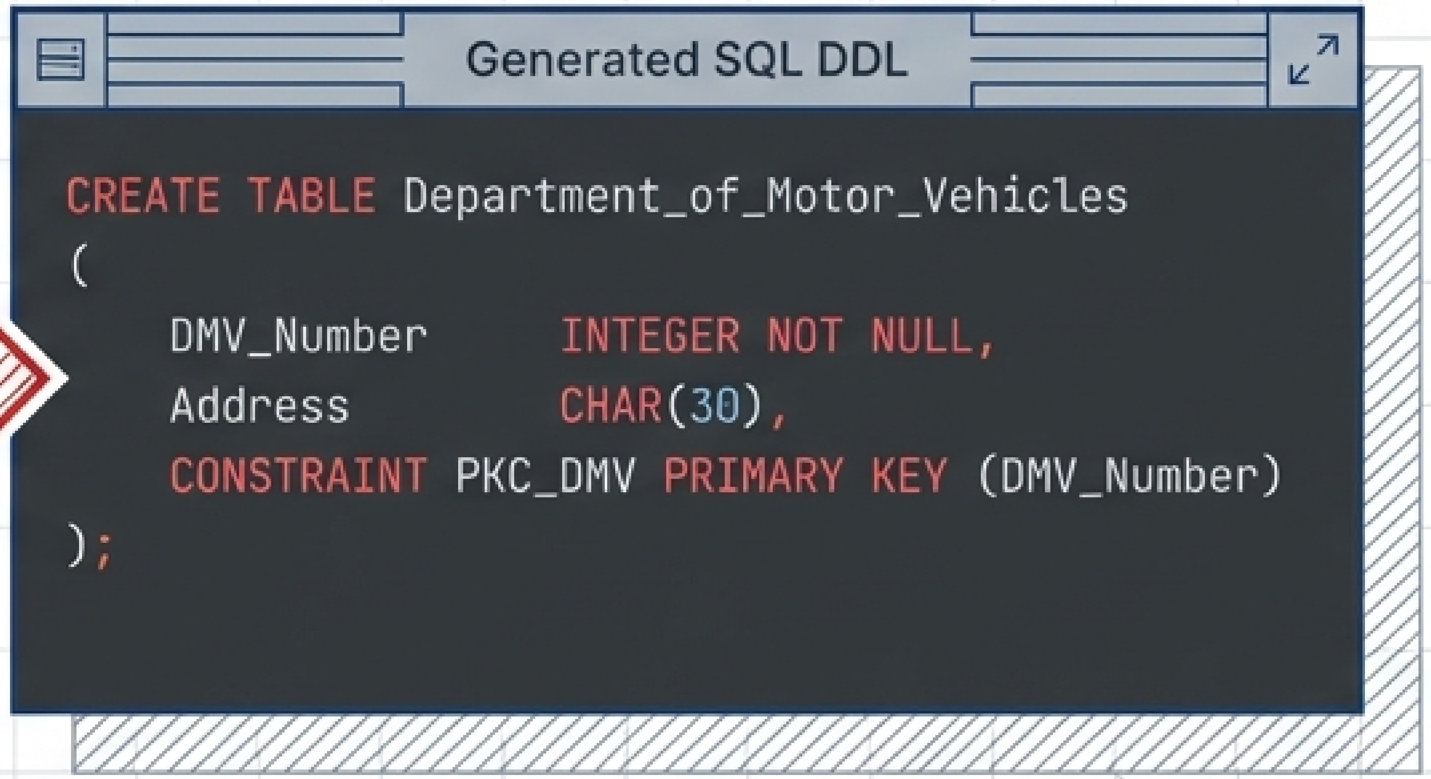
From Model to Code

Automated SQL DDL & XML Generation

THE MODEL



THE CODE

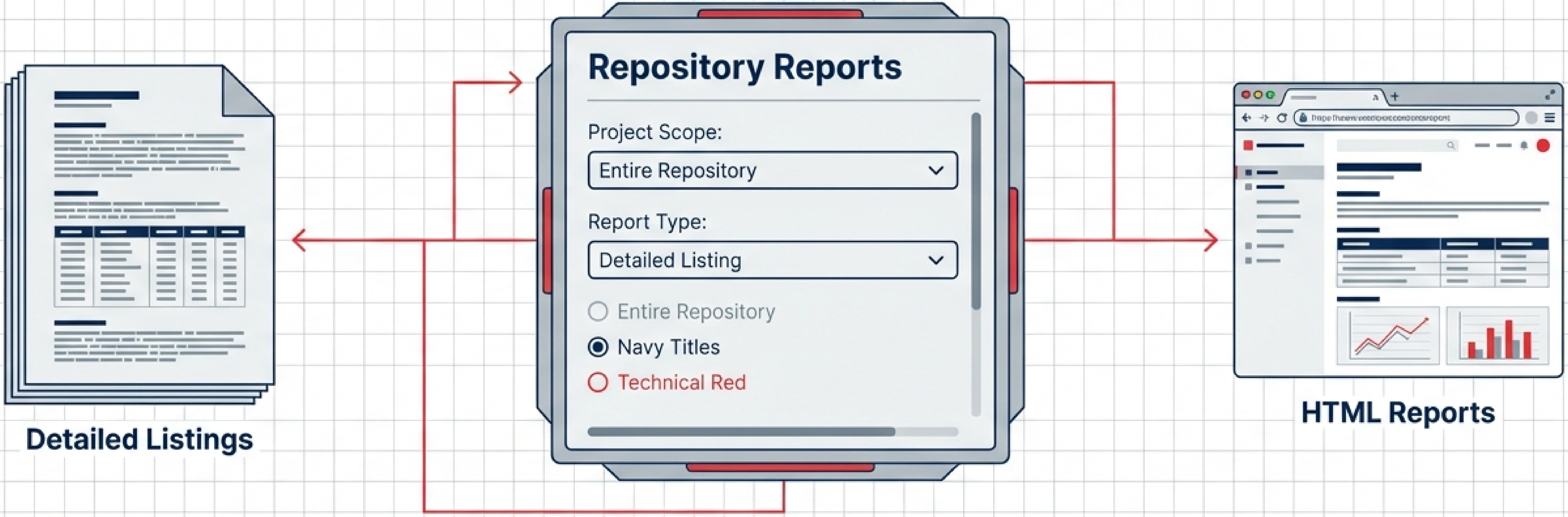


```
CREATE TABLE Department_of_Motor_Vehicles
(
    DMV_Number     INTEGER NOT NULL,
    Address        CHAR(30),
    CONSTRAINT PKC_DMV PRIMARY KEY (DMV_Number)
);
```

Supports Oracle, Access, and User-Defined Dialects.
Generates W3C standard XML Schemas.

Deliverables: Automated Documentation

Communication without Drudgery: Documentation is produced concurrently with design, ensuring stakeholders are always informed.



Repository Reports

Project Scope: Entire Repository

Report Type: Detailed Listing

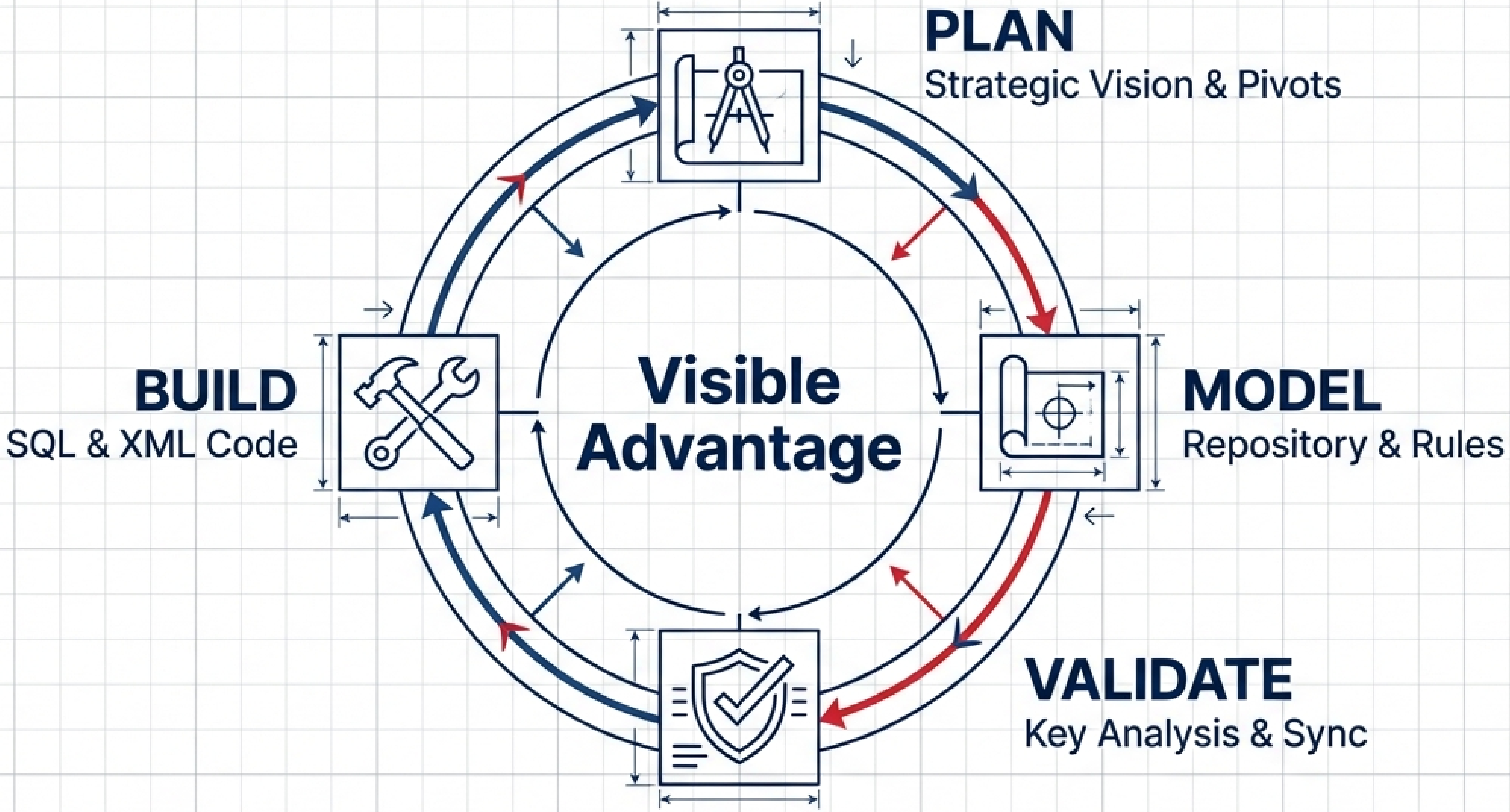
○ Entire Repository

● Navy Titles

○ Technical Red

Communication without Drudgery: Documentation is produced concurrently with design, ensuring stakeholders are always informed.

Conclusion: Putting MDA Within Reach



PLAN
Strategic Vision & Pivots

MODEL
Repository & Rules

VALIDATE
Key Analysis & Sync

BUILD
SQL & XML Code

Visible Advantage

Start small. Select a real-world project. Make the tool a seamless part of your everyday workflow.