

Static Model Harness

Describe domain structure in models that can be checked and executed

Model contracts

Types, identity, ownership, relationships

Tool enforcement

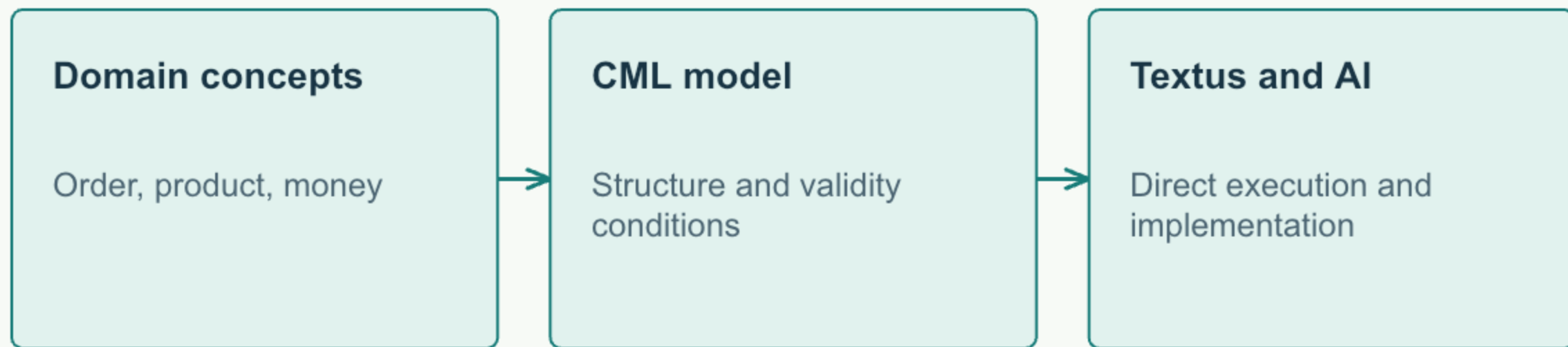
Detect violations at build and runtime

AI implementation

Generate and change within contracts

Models Close to Domain Knowledge

Express business concepts and rules in executable models



Entities and Value Objects

Distinguish object identity from equality by value

Entity: Order

The same object persists beyond execution

Value object: Money

Equality depends on amount and currency

Classification and Shared Properties

Use generalization as the basis, supplemented by powertypes and traits

Generalization

Concept hierarchy with
single inheritance

Powertype

Classification axes such as
customer type

Trait

Composition of shared
partial models

Classification, Ownership, and Associations

Start with kinds and whole–part structure, then describe other relationships

Is-a: Classification

General concepts and
specific kinds

Has-a: Whole and parts

Aggregation and
composition

Other associations

References to independent
objects

Aggregate, View, and Display Models

Each model defines a different grouping and role

Aggregate Model

Order and lines as a whole
with invariants

View Model

Overview or detail that
preserves meaning

Display Model

Presentation of headings,
values, actions

Entity Stereotypes

Capture structures and behavior suited to each domain role

Resource

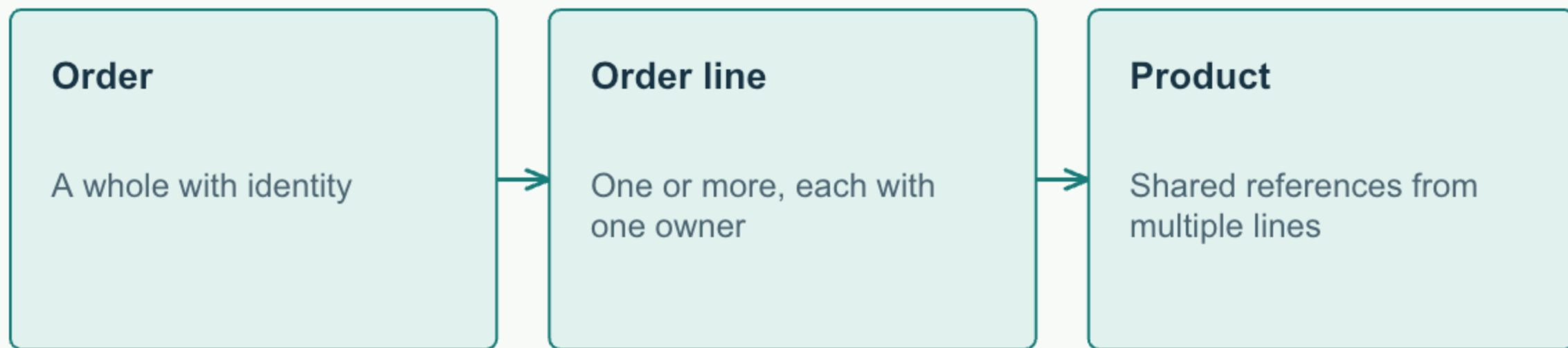
A resource with ownership and usage rules

Task

A unit of work with inputs and results

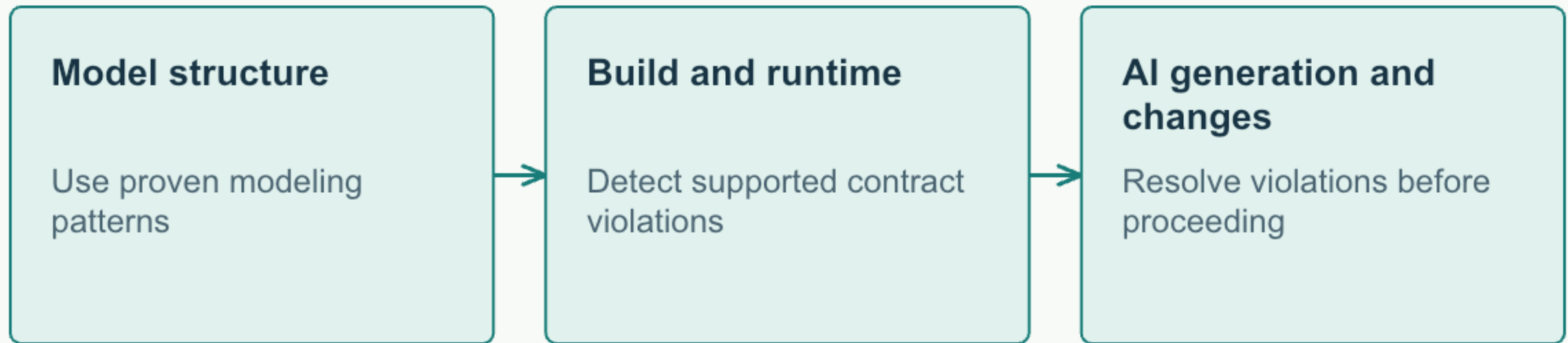
Ownership and Product References

Distinguish exclusive ownership of lines from shared product references



A Harness Through Checking and Execution

Turn proven concepts, components, and patterns into enforced constraints



Checks and Review

Combine mechanical checks with review of meaning

Static checks

Types, definitions,
references

Runtime checks

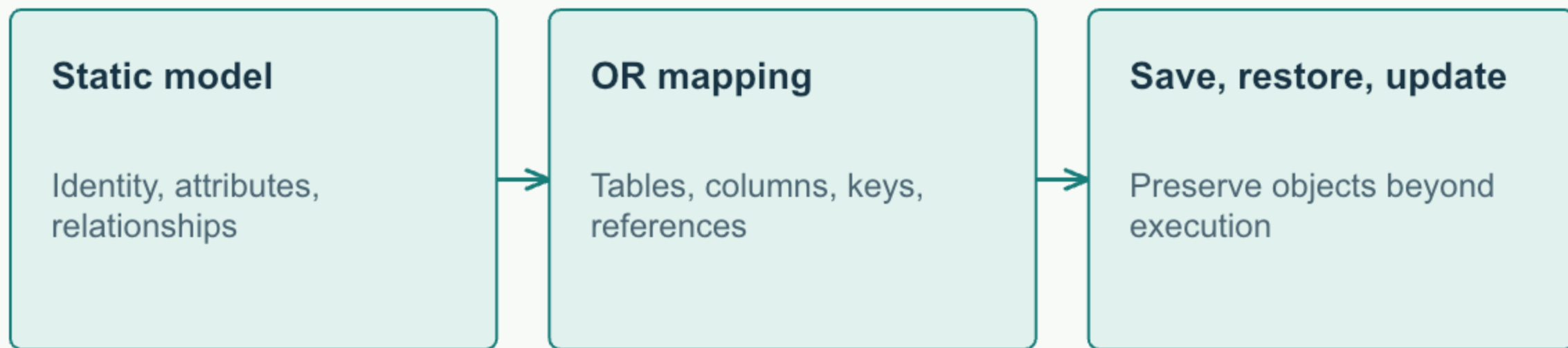
Values, multiplicity,
concurrent updates

Semantic review

AI assists, people judge
validity

Persistence and OR Mapping

Textus automates the mapping between static models and the database



Models Connected Through Terms

Shared glossary terms connect business explanations with model structure

Literate model

Business meaning and rationale

Glossary and RDF nodes

Names, definitions, identifiers

Object model

Types, attributes, relationships

Check term-to-model correspondence and prepare RDF knowledge for access through MCP

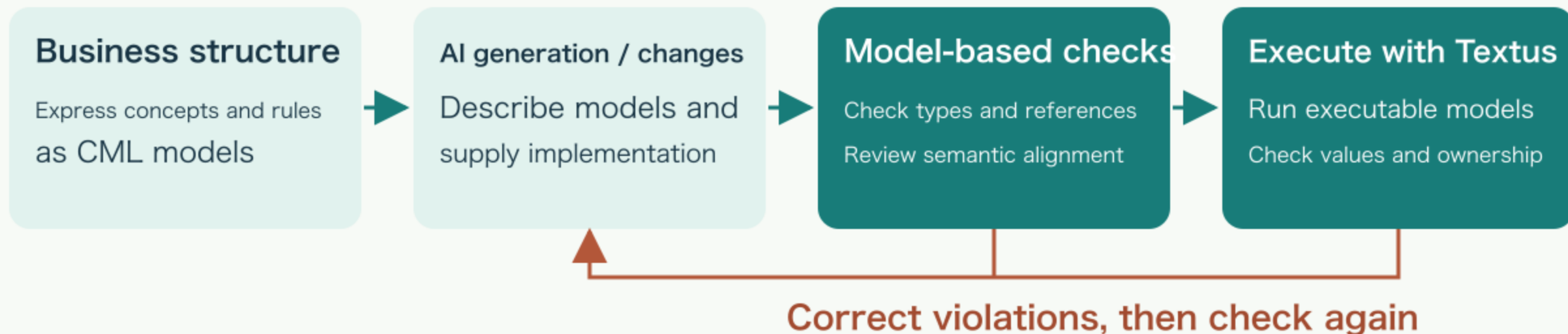
Model Changes and Their Impact

Use model meaning to assess effects on implementation and checks



Model contracts guide AI-generated work and changes

Software engineering foundations become checks and executable constraints



A modeling framework that makes constraints concrete

Types · Identity · Classification · Ownership · References · Multiplicity

Aggregate Model · View Model · Display Model / Persistence and glossary links

Constraints apply where supported by checks and execution. People also judge business fit.

From Static to Dynamic Models

The next episode adds models of permitted behavior to structural constraints

Part 3: Static models

Valid structures



Part 4: Dynamic models

States, transitions, behavior