

Application Modeling

Orchestrate domains to fulfill users' goals

Orchestrate multiple domains
Reuse a shared domain
Realize Use Case Scenarios

SIMPLEMODELING MODELING TECHNOLOGY SERIES · PART 9

Application Modeling

Orchestrate domains to fulfill users' goals

Basic structure: one application, several domains

Make needs, goals, and Use Cases concrete



Coordinate invocation, order, results, Events, and failures

Different needs

Different needs, shared domain



Use a shared domain from each application
Check domain updates' impact on applications

Application responsibilities

Define input/output, order, branches, and failures
Set Operation characteristics through use and properties
Stateless process flow guards and function capabilities

Domain responsibilities

Manage key data operations, rules, and invariants
Manage business state and StateMachines; expose Events
Account for shared understanding over a longer lifespan

From Use Cases to reviewable executable Models



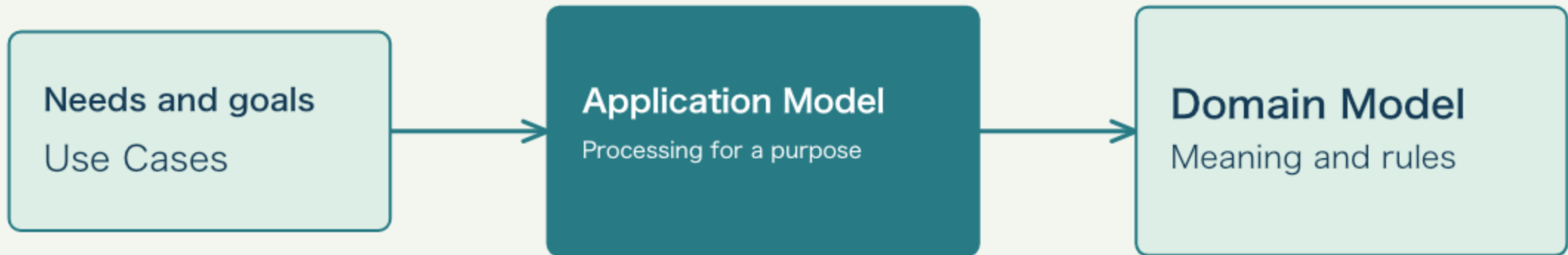
Trace requirements to responses; use feedback for the next Model update

CML: Causal Modeling Language | Interactions describe the realization examples; CML describes executable Models

SimpleModeling.org

Meet users' goals through domains

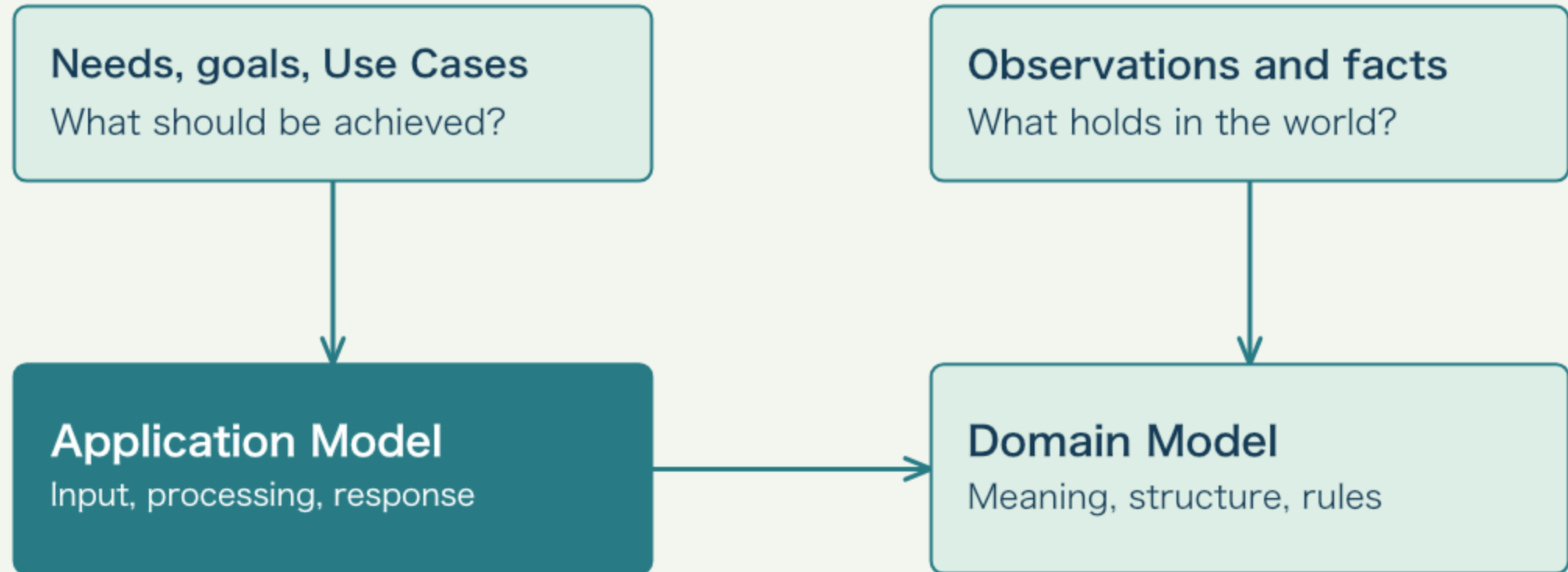
Application Models connect needs to domain functionality.



Both organize Object, Knowledge, and Literate Model elements for a purpose.

Different starting points and roles

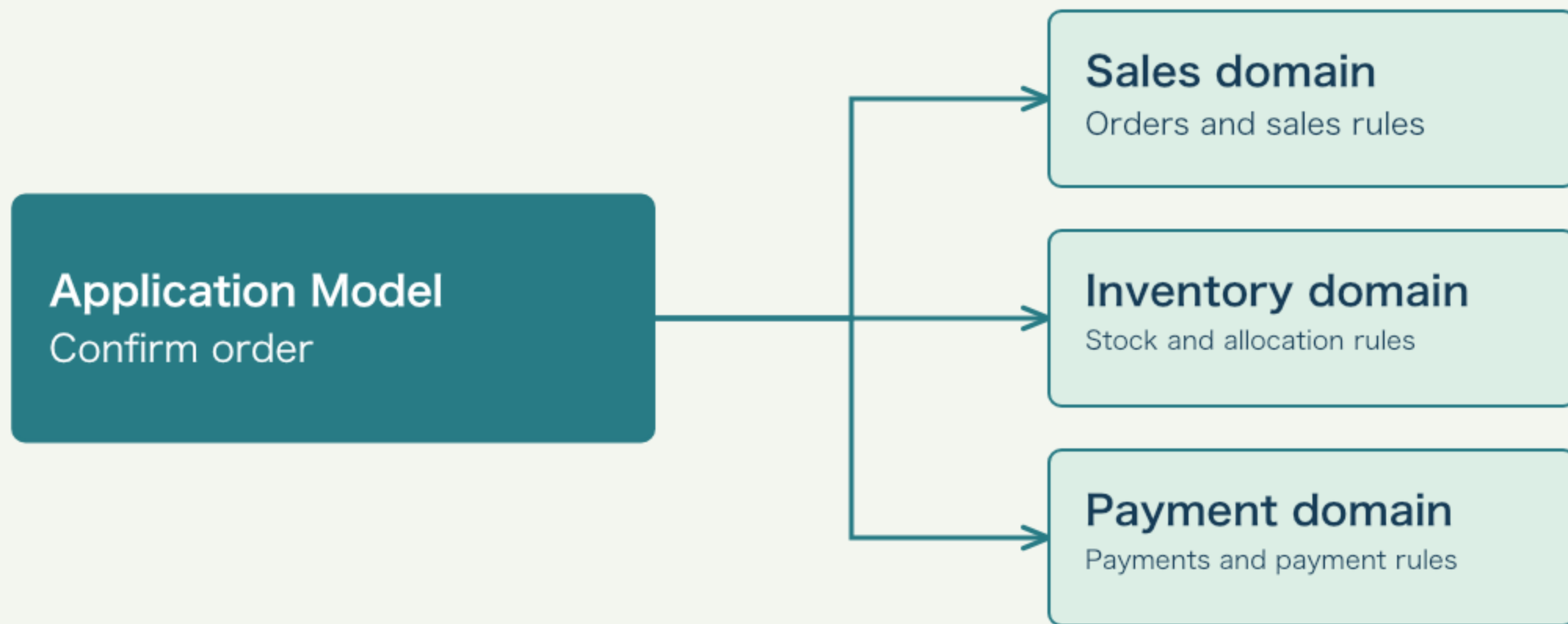
Applications realize goals; domains organize understanding.



Application → Domain: use the domain's meaning and functionality.

One application orchestrates several domains

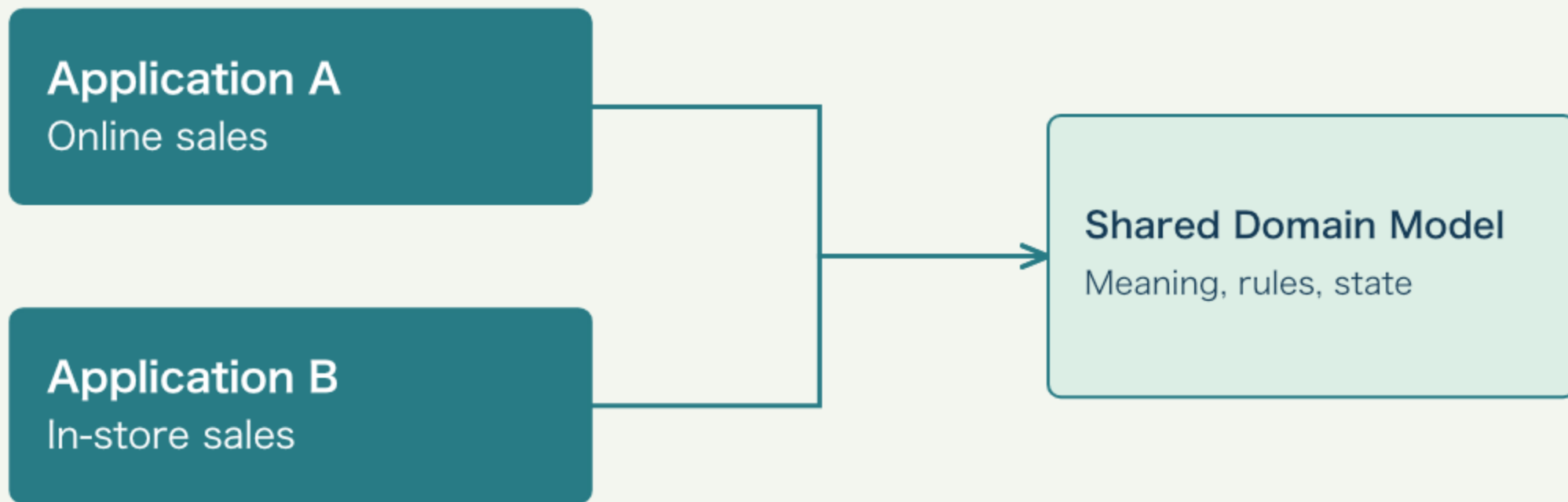
Combine each domain's functionality for the user's purpose.



Applications evolve with needs; each domain manages its own meaning, rules, and state.

Applications for different needs share a domain

Each Application Model serves its stakeholders' needs.



When updating a shared domain, check the impact on each application that uses it.

Map Use Case Scenarios to realization

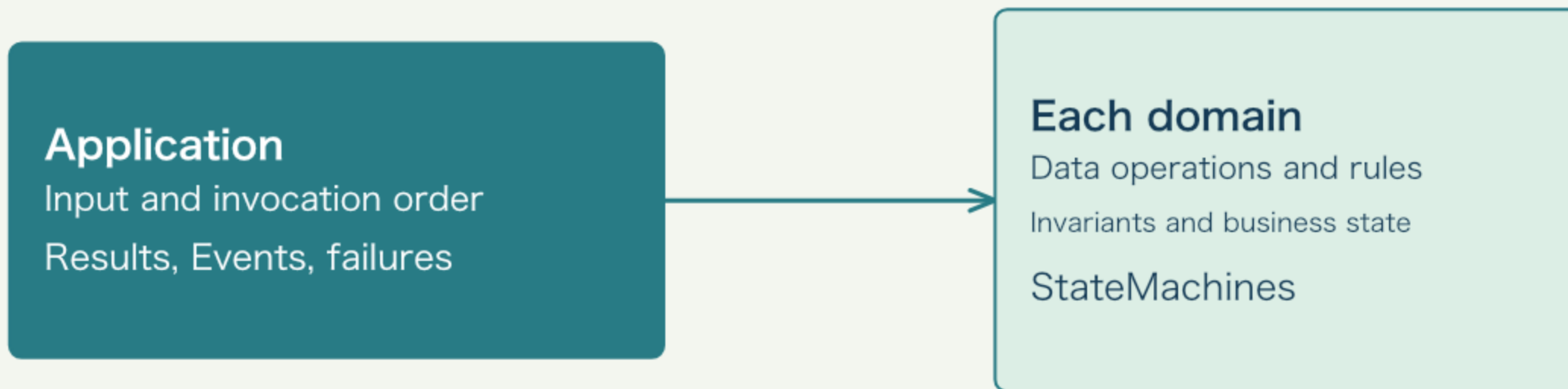
Connect user actions to UI, participants, and their interactions.



Review the mappings from each Scenario step to internal processing and responses.

The application is a thin orchestrator

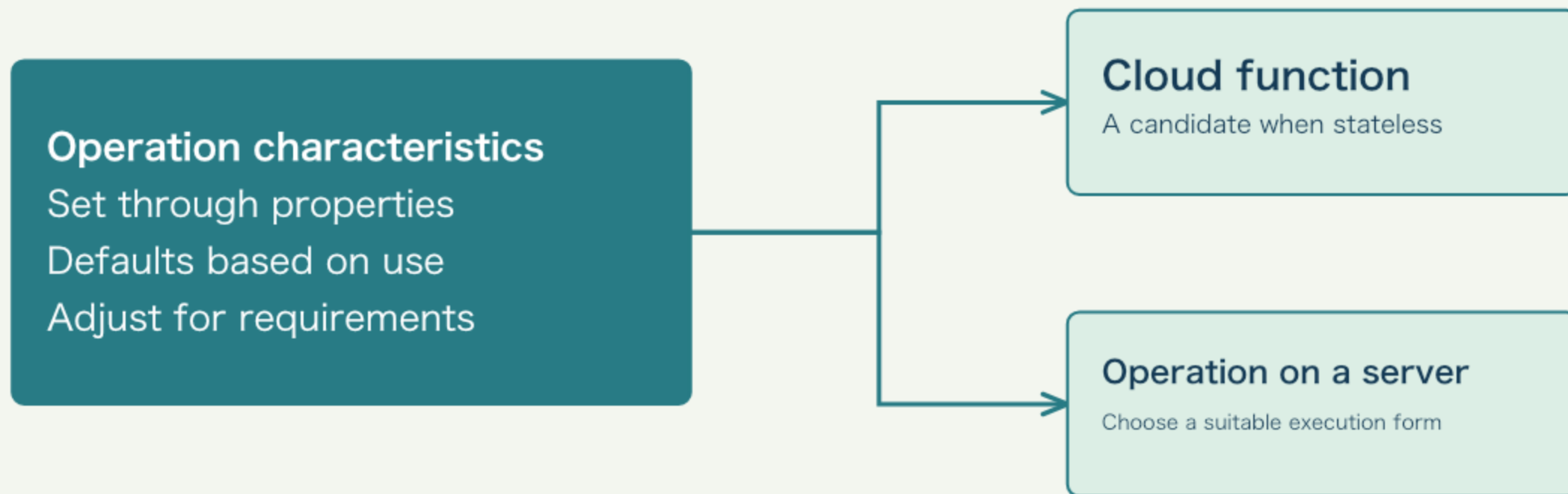
Coordinate order, results, and failures; domains do business operations.



Order confirmation: compose domain operations, result transfers, and failure responses.

Optimize realization through properties

Define Operation characteristics and select suitable execution forms.

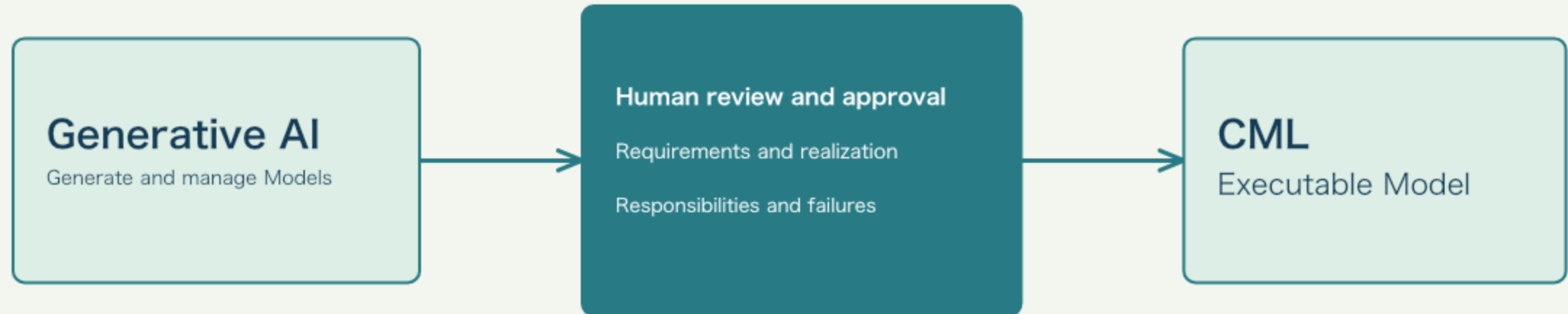


Example: cloud functions can substantially cut costs when usage and pricing fit.

Defaults: application use is stateless. Domain use is stateful. Check time and resources.

AI constructs Models; people review and approve

Trace requirements to realization and reflect approved Models in CML.



textus-cbd-support visualizes mappings, changes, and evidence; feedback informs updates.

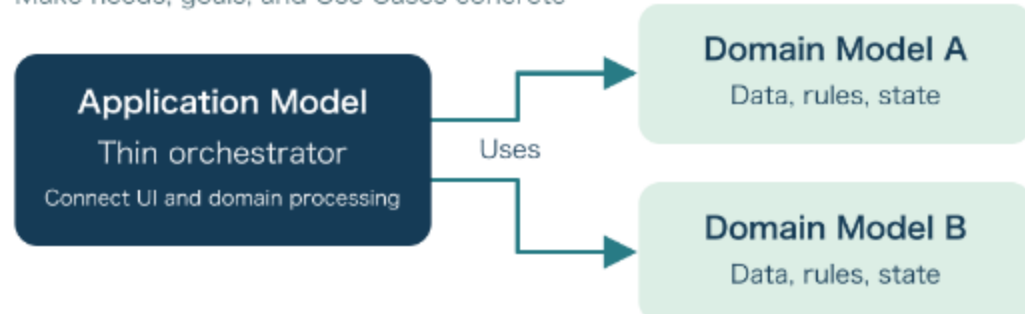
CML: Cozy Modeling Language. Execution-example Models such as Interactions are outside current CML.

Application Modeling

Orchestrate domains to fulfill users' goals

Basic structure: one application, several domains

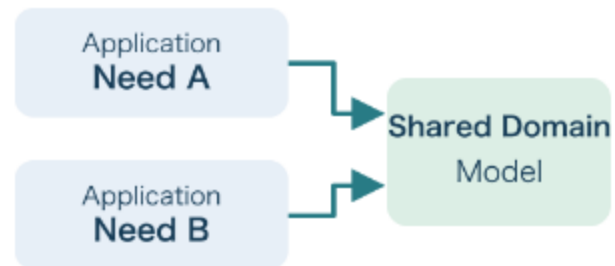
Make needs, goals, and Use Cases concrete



Coordinate invocation order, results, Events, and failures

Different needs

Different needs, shared domain



Use a shared domain from each application
Check domain updates' impact on applications

Application responsibilities

Define input/output, order, branches, and failures
Set Operation characteristics through use and properties
Stateless properties guide cloud-function candidates

Domain responsibilities

Manage key data operations, rules, and invariants
Manage business state and StateMachines; expose Events
Accumulate world understanding over a longer lifecycle

From Use Cases to reviewable executable Models



Trace requirements to responses; use feedback for the next Model update

CML: Cozy Modeling Language | Interactions describe execution examples; CML describes executable Models

SimpleModeling.org

References and credits

SimpleModeling.org

<https://www.simplemodeling.org/>

Part 8: Domain Modeling

<https://www.simplemodeling.org/en/development-process/domain-modeling.html>

Part 7: Literate Modeling

<https://www.simplemodeling.org/en/development-process/literate-modeling.html>

Part 6: Knowledge Modeling

<https://www.simplemodeling.org/en/development-process/knowledge-modeling.html>

Part 5: Object Modeling

<https://www.simplemodeling.org/en/development-process/object-modeling.html>

Created by SimpleModeling

Reimu and Marisa: Kitsune Yukkuri by Kitsune. Touhou Project derivative work.

Voices: macOS say — Samantha (Reimu), Karen (Marisa)

Character source: <https://ci-en.net/creator/34363/article/1770533>