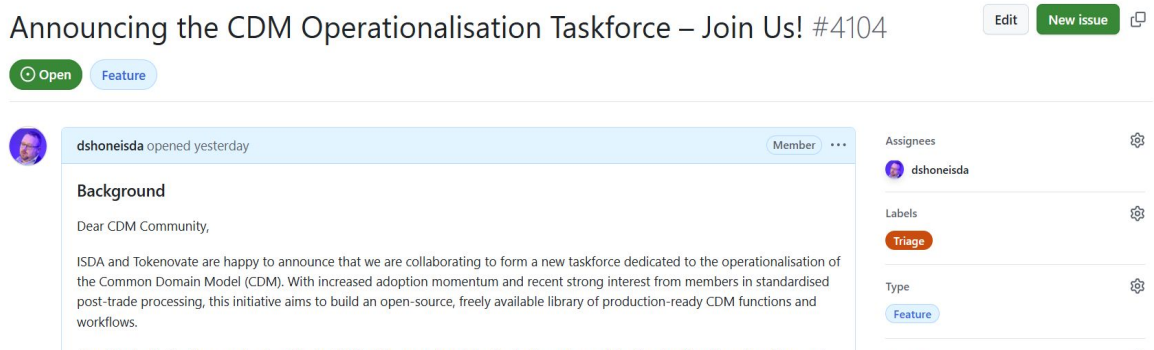
Common Domain Model - Smart Contracts Overview

2026

Modelling tokenised securities in CDM

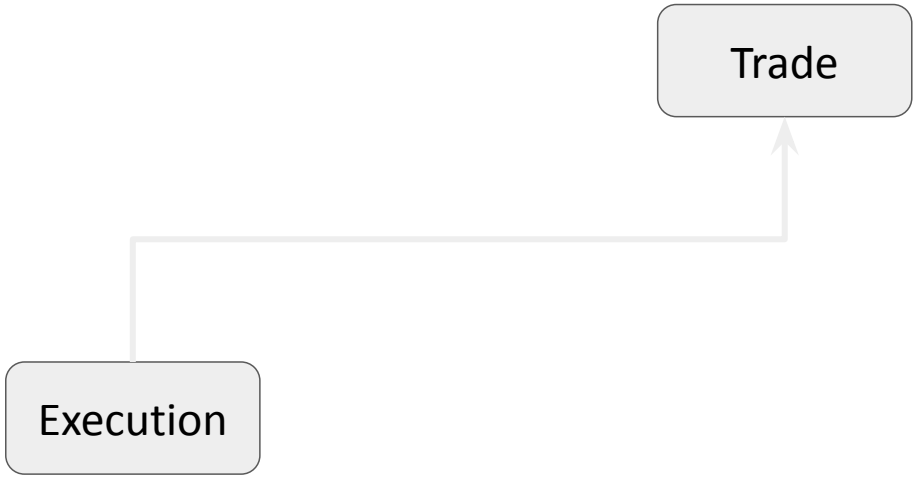
CDM
COMMON DOMAIN MODEL

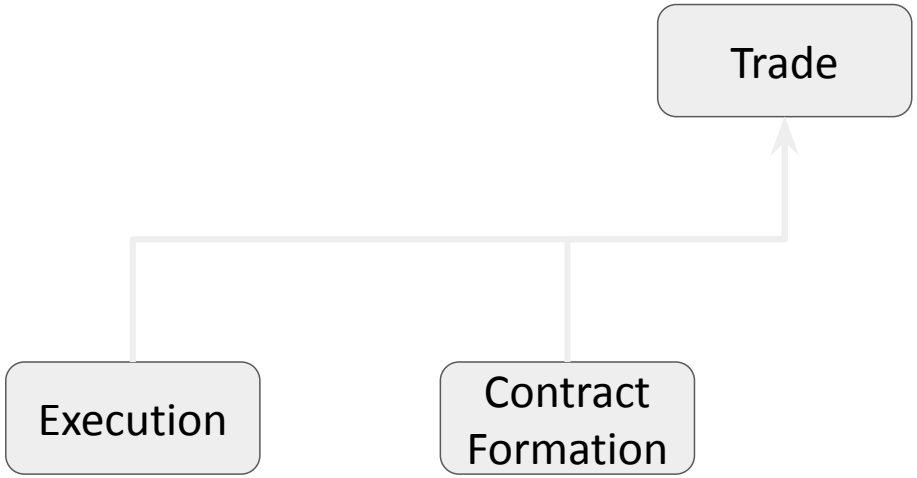
Problem statement	**Increased adoption momentum has seen more firms actively engaging with CDM. This has highlighted:** **1) Strong interest from members in standardized post-trade processing using CDM** **2) Knowingly under-developed workflow dimension can lead to orchestration of CDM functions in slightly different ways by implementers**
Objective	**Build an open-source and freely available library of standardized production-ready CDM functions and workflows to allow firms to automate post-trade events using CDM logic directly in their infrastructure, beginning with a focus on resets.** **This will be achieved by workshops with members and development, project managed and OS code delivered by Tokenovate on behalf of ISDA**
In-scope	• **66% of clauses in ISDA 2021 IRD Definitions** • Reset process including: observation, business day adjustments, observation lags (lookback, lockout, shift), compounding/averaging/interpolation methodologies, caps and floors • Fallback logic • Cashflow calculation • Integration with ISDA FRO matrix (access to values will still require relevant membership/IP purchasing through MyLibrary)
Benefits	• **Added value to CDM/DRR adopters** through consistent industry tooling for a wider set of post-trade use cases • **Reduced risk** of fragmentation at the implementation stage • Foundation for tokenized workflows and **T+1 settlement** • Ready-made libraries for use in **smart contracts**, also **unlocking AI-based modelling** • **Scalable to other asset classes** and products e.g. Eq/FX Swaps, any product with averaging, compounding components

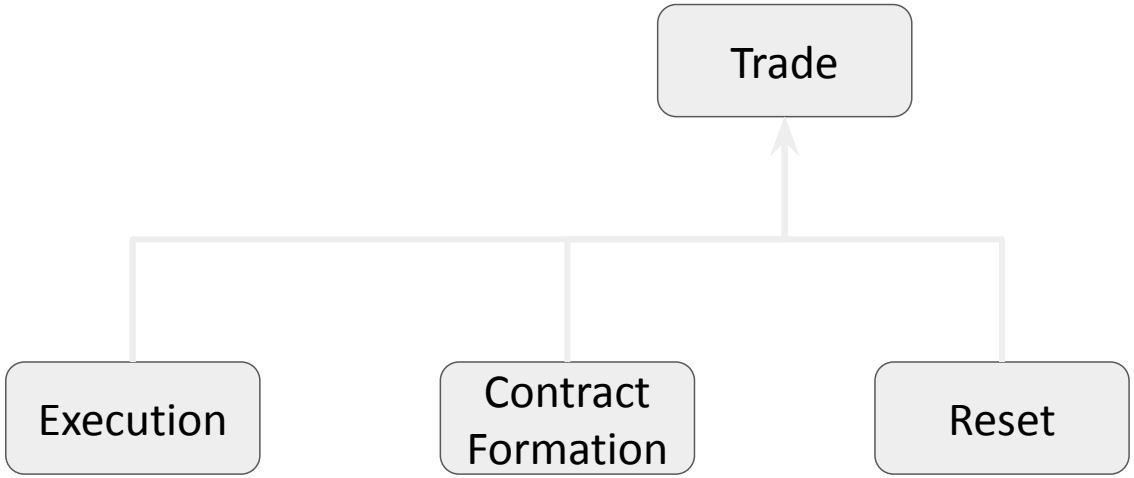


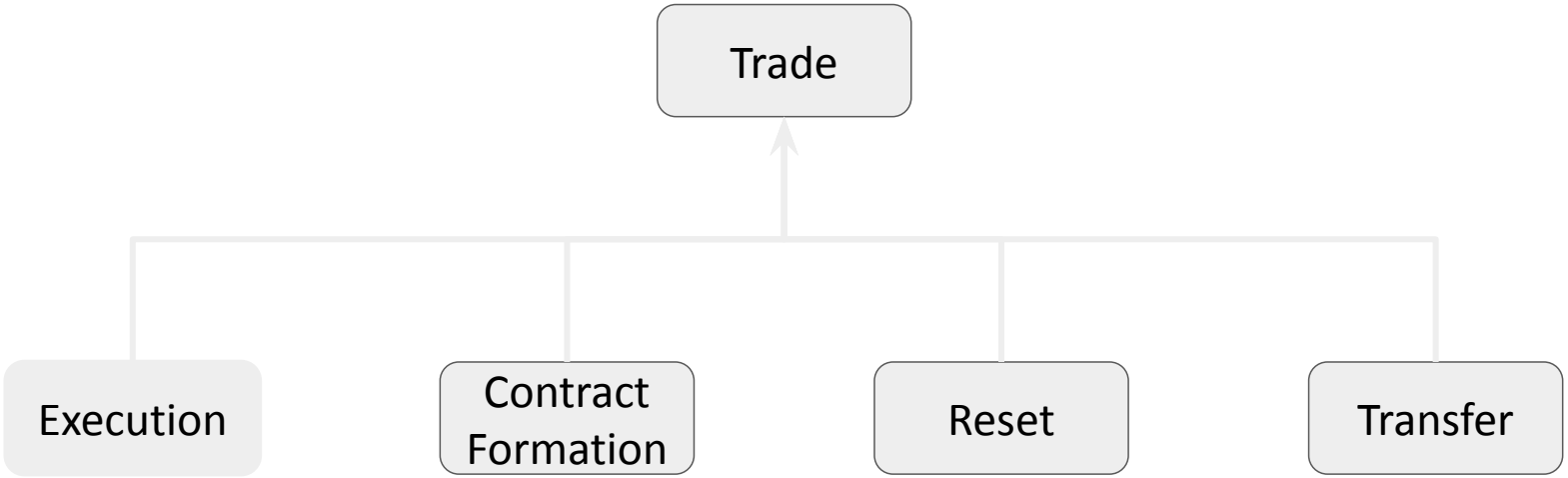
TOKENOVATE

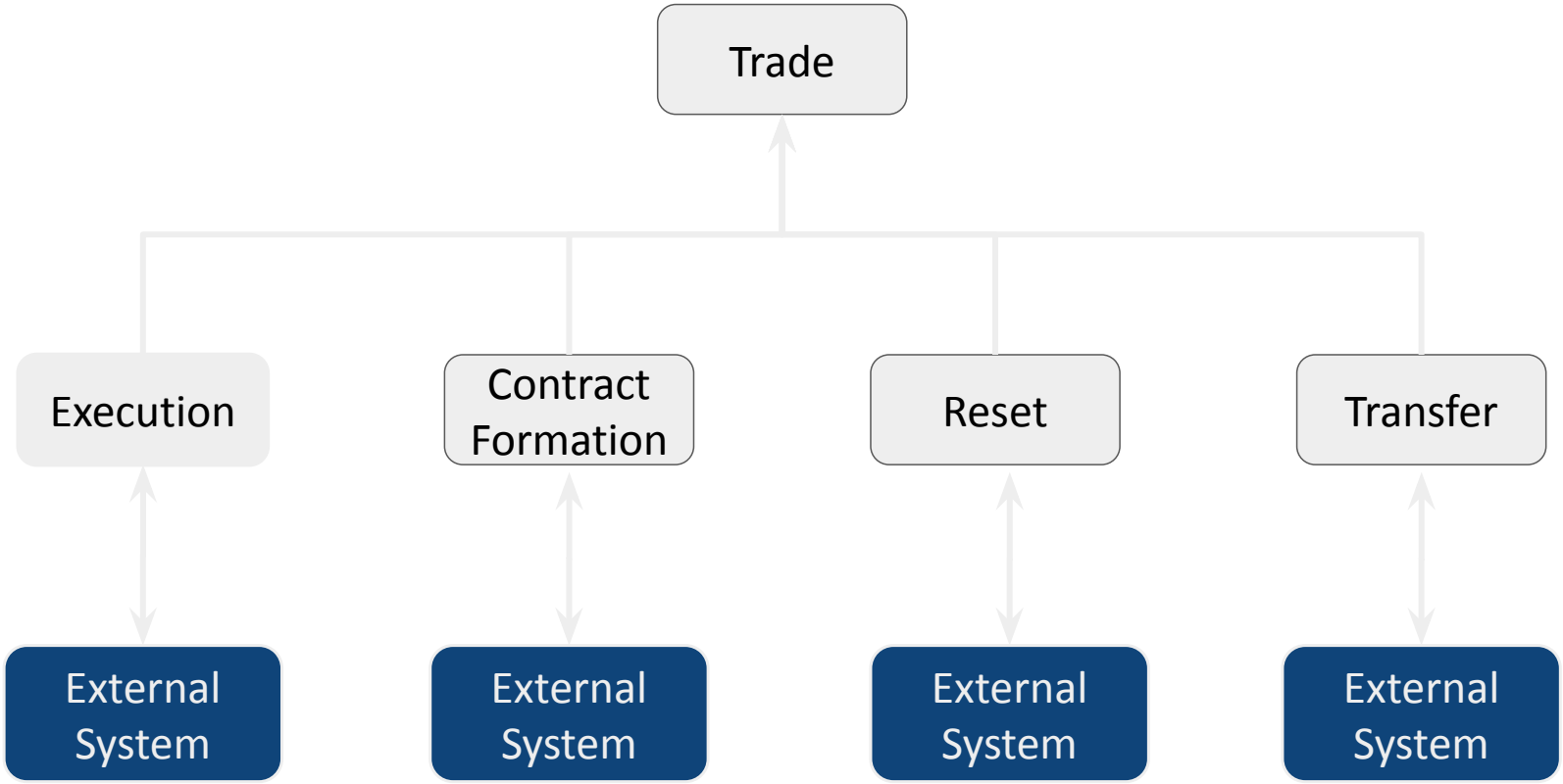
Trade

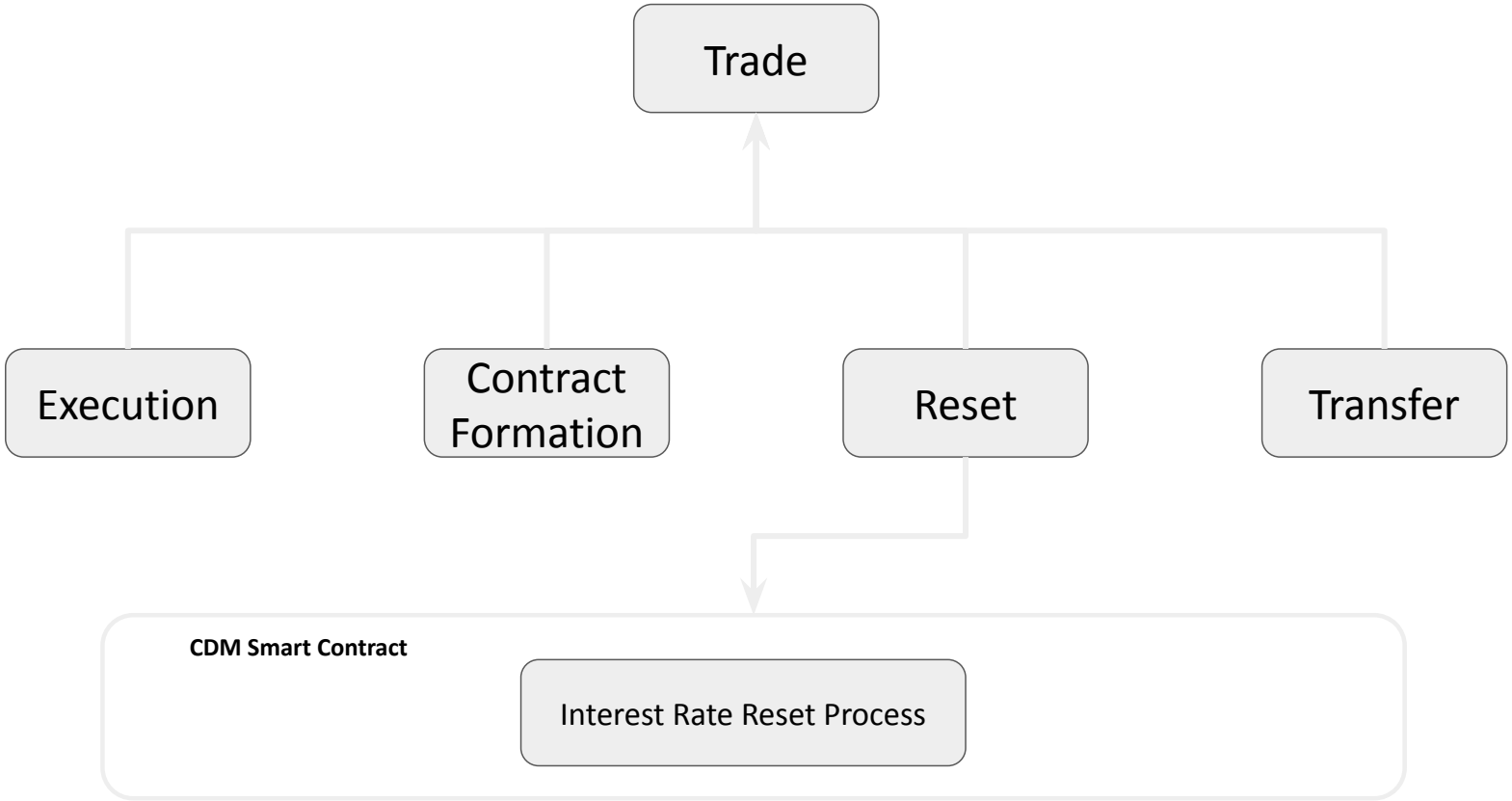

Trade
Execution


Trade
Execution
Contract Formation


Trade
Execution
Contract Formation
Reset

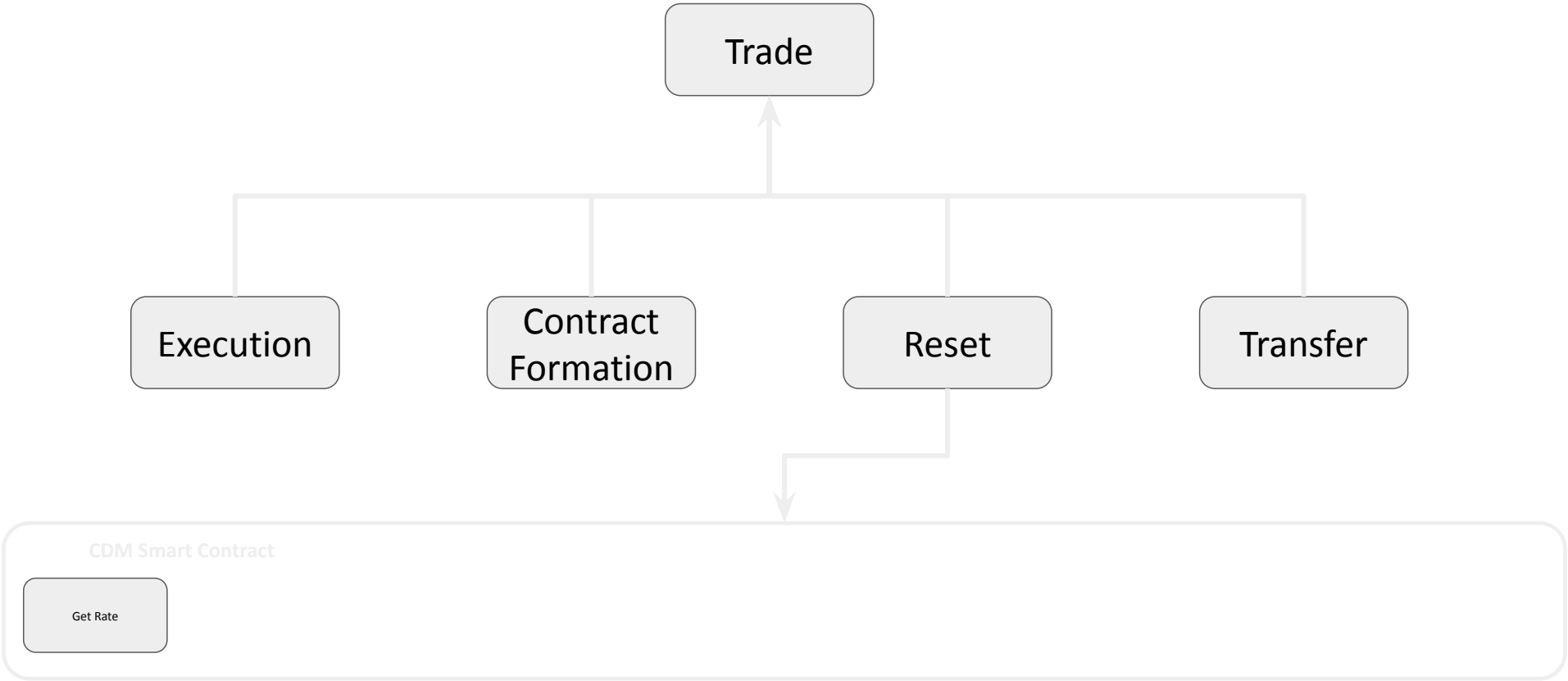

Trade
Execution
Contract Formation
Reset
Transfer

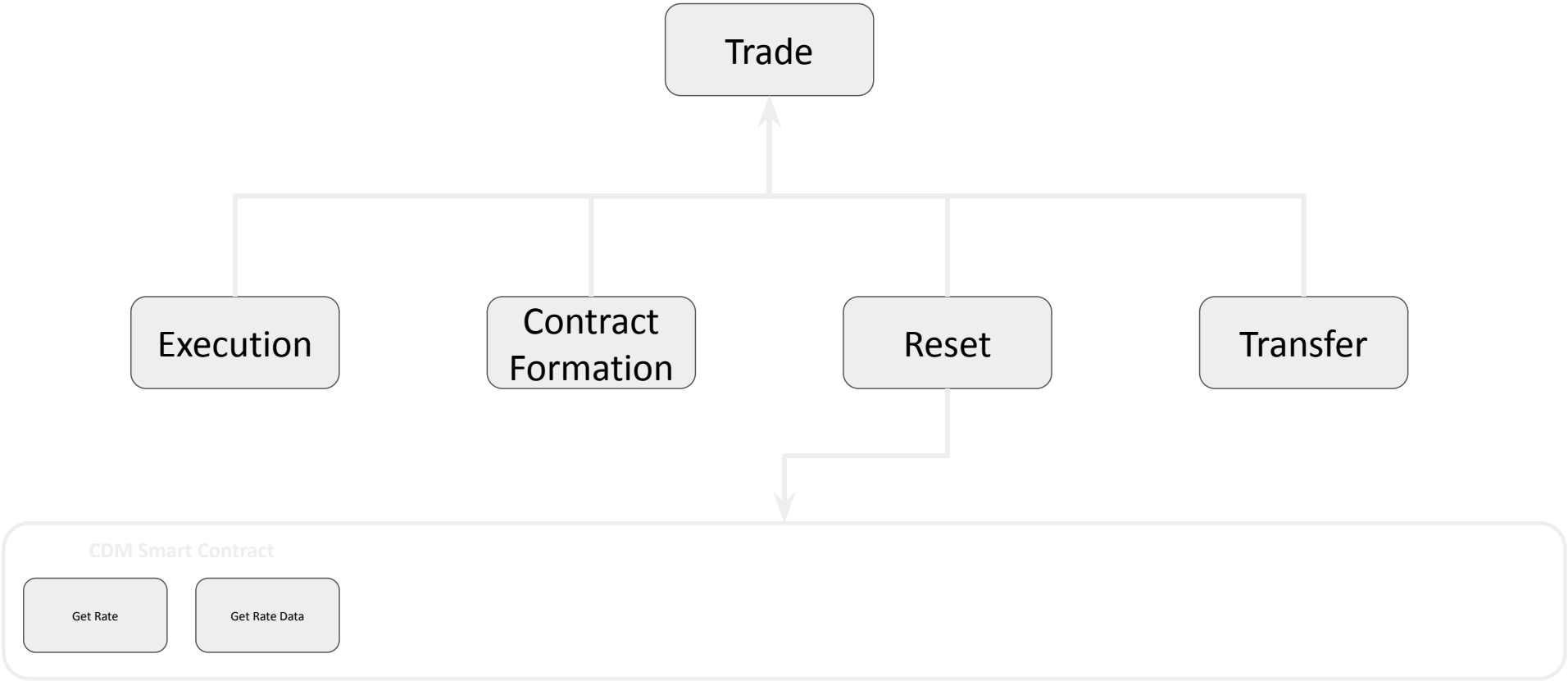

Trade
Execution
Contract Formation
Reset
Transfer
External System
External System
External System
External System

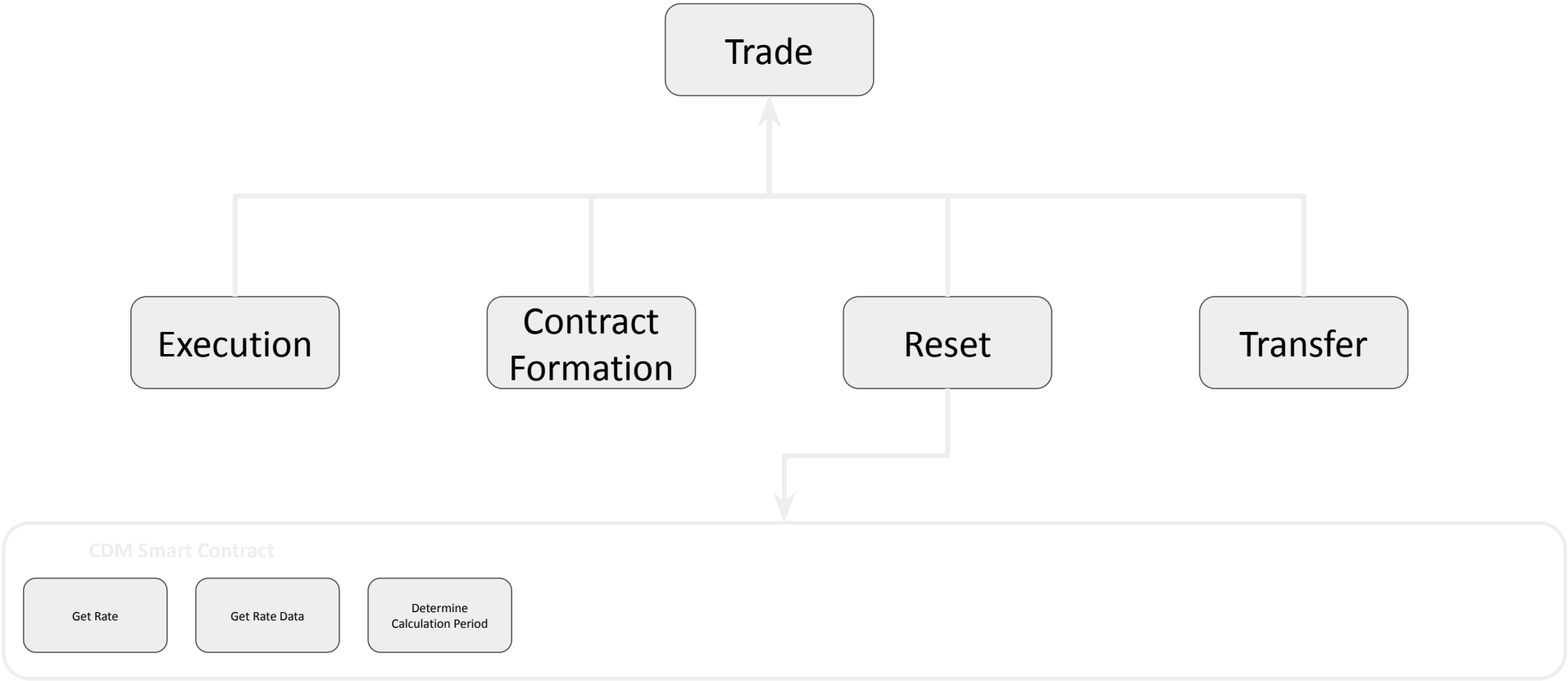

Trade
Execution
Contract Formation
Reset
Transfer
CDM Smart Contract
Interest Rate Reset Process

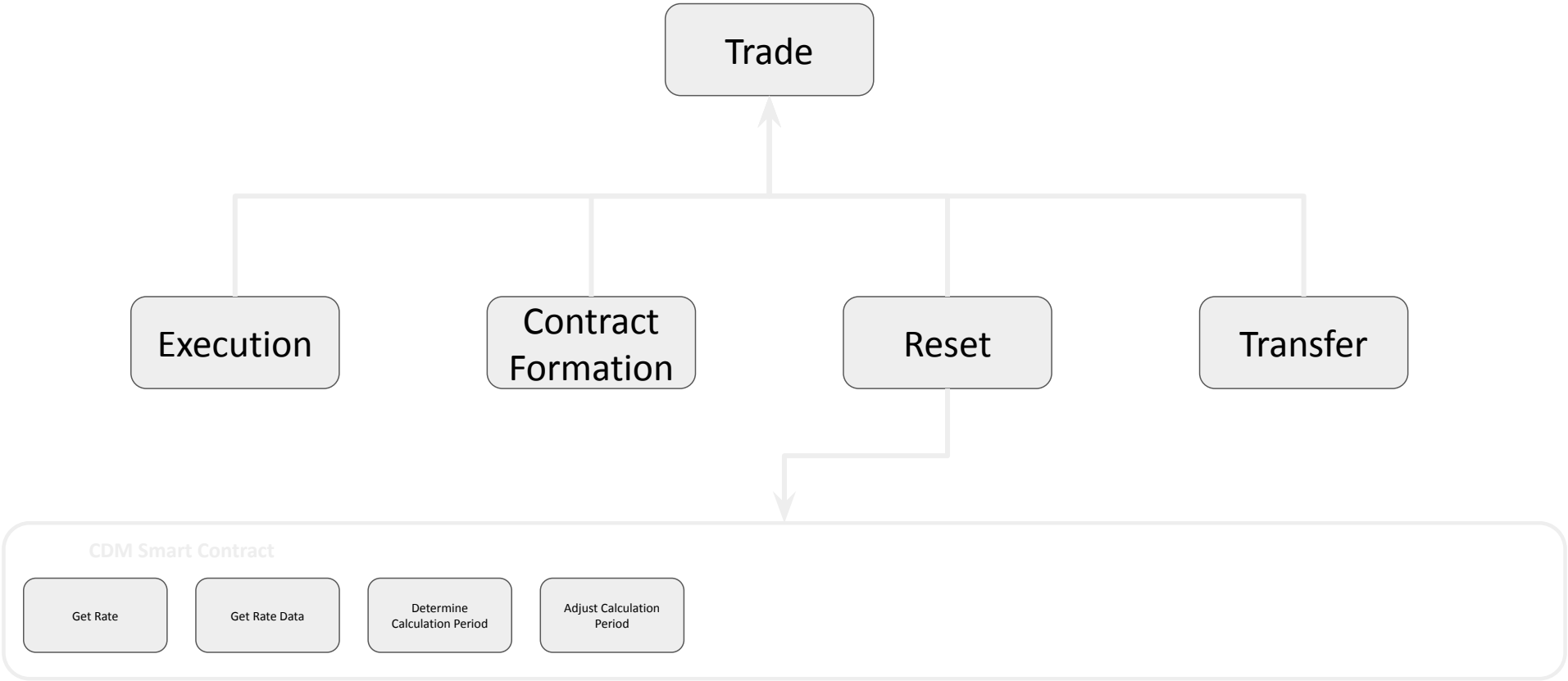
Design Principles

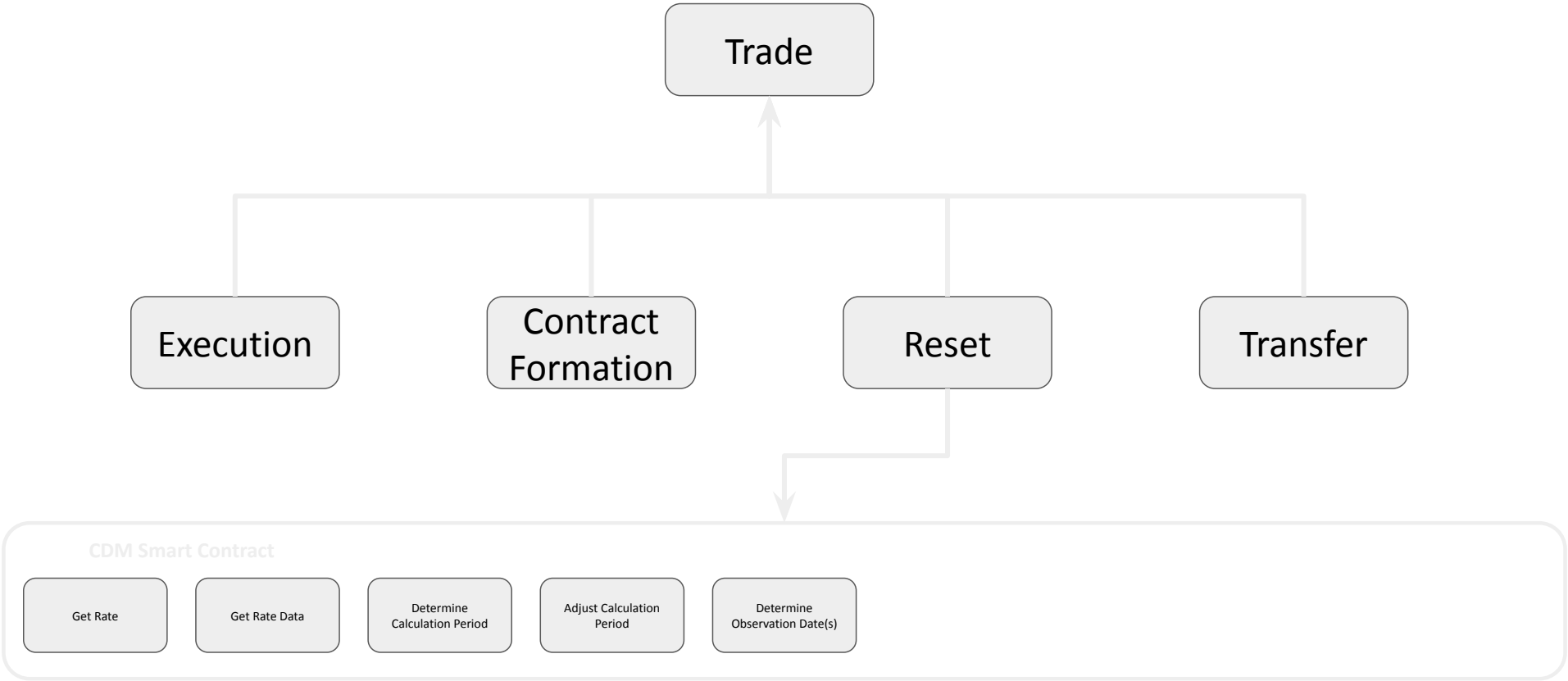
1. **Single-function, reusable steps:** Each Smart Contract Step performs one defined operation (for example date adjustment or rate observation) and can be reused across products and lifecycle events.
2. **Composable workflows:** Products and market conventions are implemented by sequencing steps. Functionality is changed by reordering or extending the sequence, not by rewriting logic.
3. **Stateful execution with traceability:** A Smart Contract History object records all intermediate inputs and outputs, enabling auditability, incremental enrichment, and deterministic replay.
4. **Isolated execution and recovery:** Each step commits its output independently. Failures affect only the relevant step, allowing targeted retries without rolling back the full workflow.

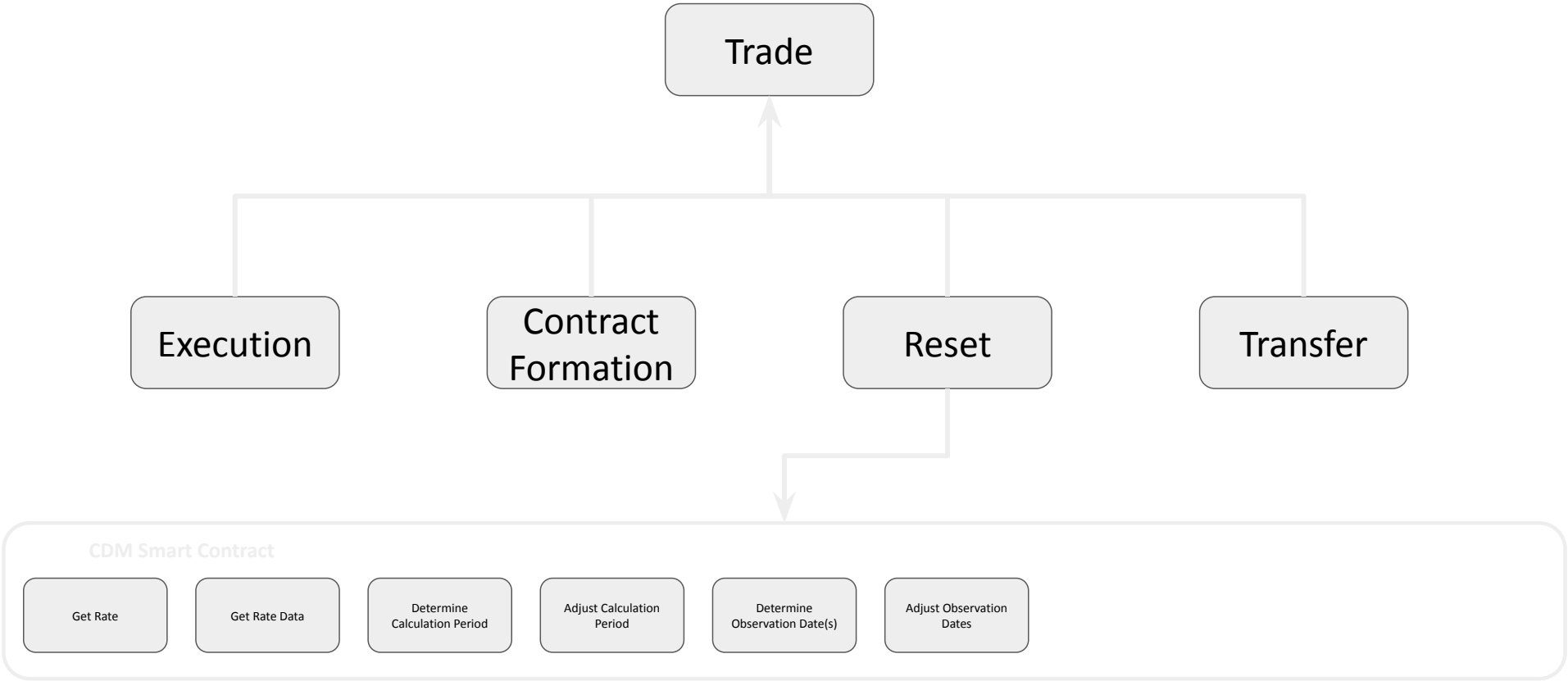

Trade
Execution
Contract Formation
Reset
Transfer
CDM Smart Contract
Get Rate

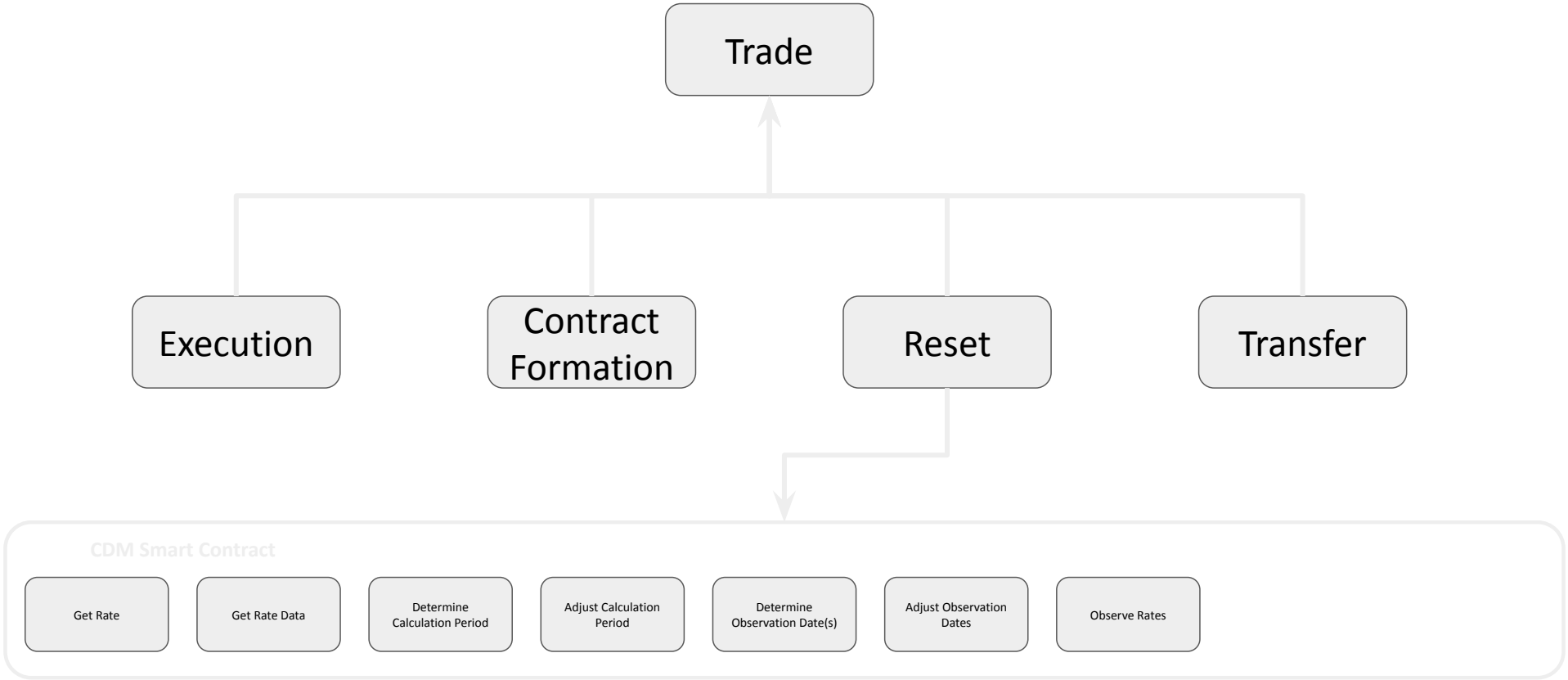

Trade
Execution
Contract Formation
Reset
Transfer
CDM Smart Contract
Get Rate
Get Rate Data

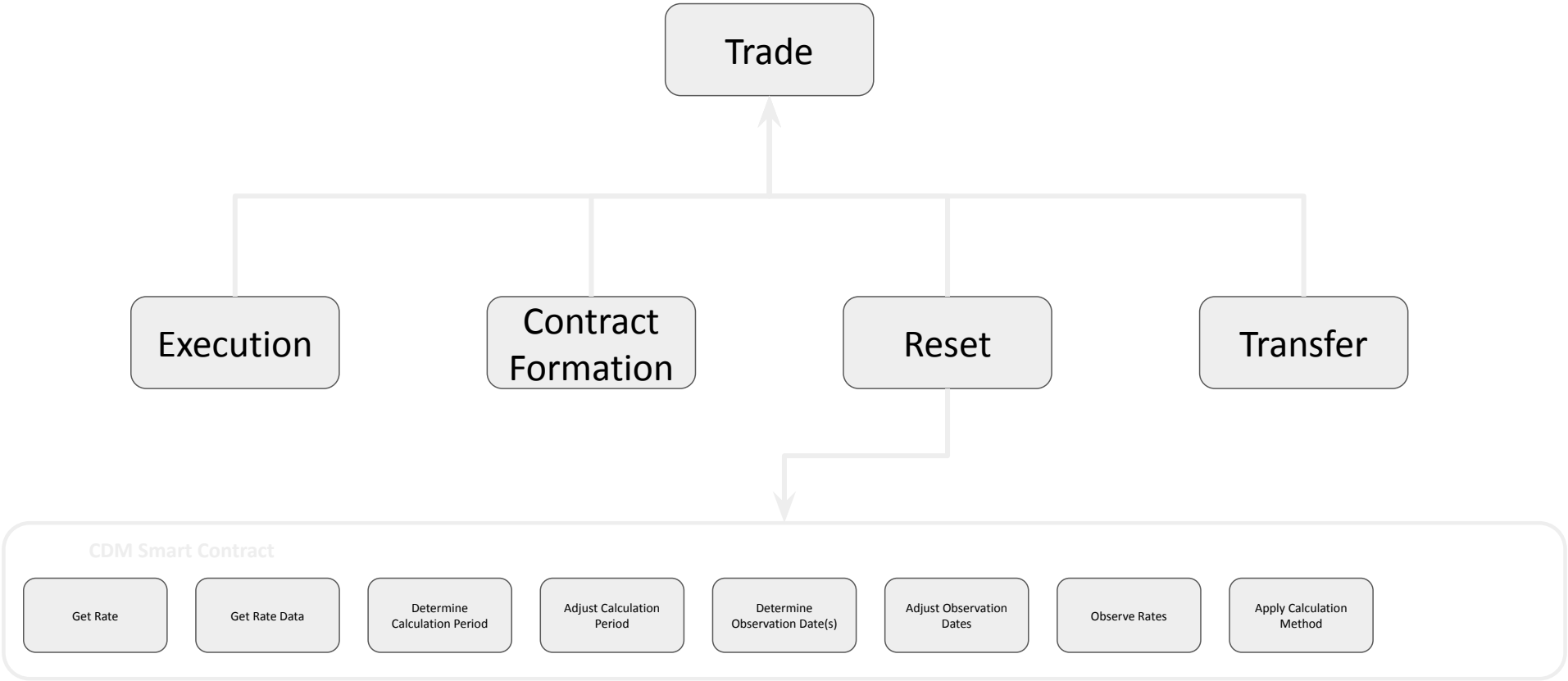

Trade
Execution
Contract Formation
Reset
Transfer
CDM Smart Contract
Get Rate
Get Rate Data
Determine Calculation Period

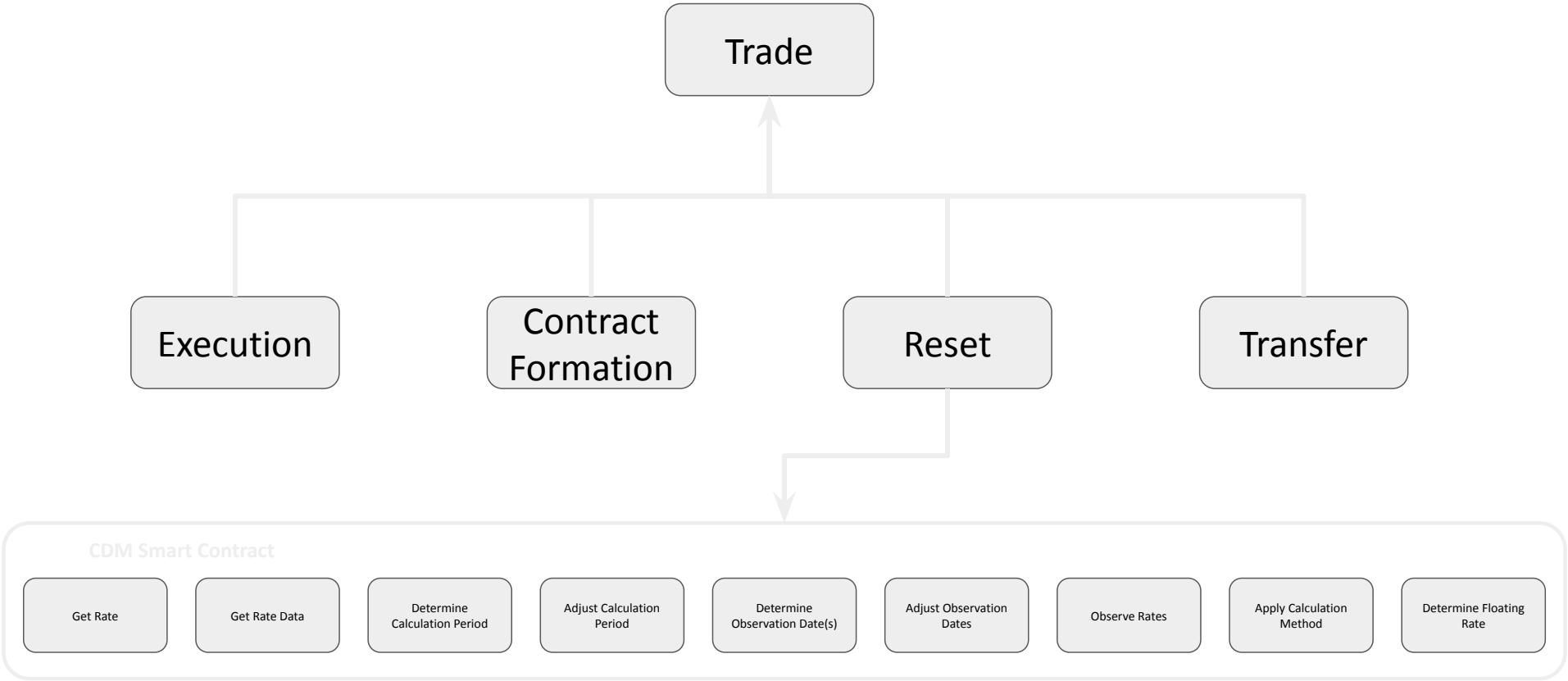

Trade
Execution
Contract Formation
Reset
Transfer
CDM Smart Contract
Get Rate
Get Rate Data
Determine Calculation Period
Adjust Calculation Period

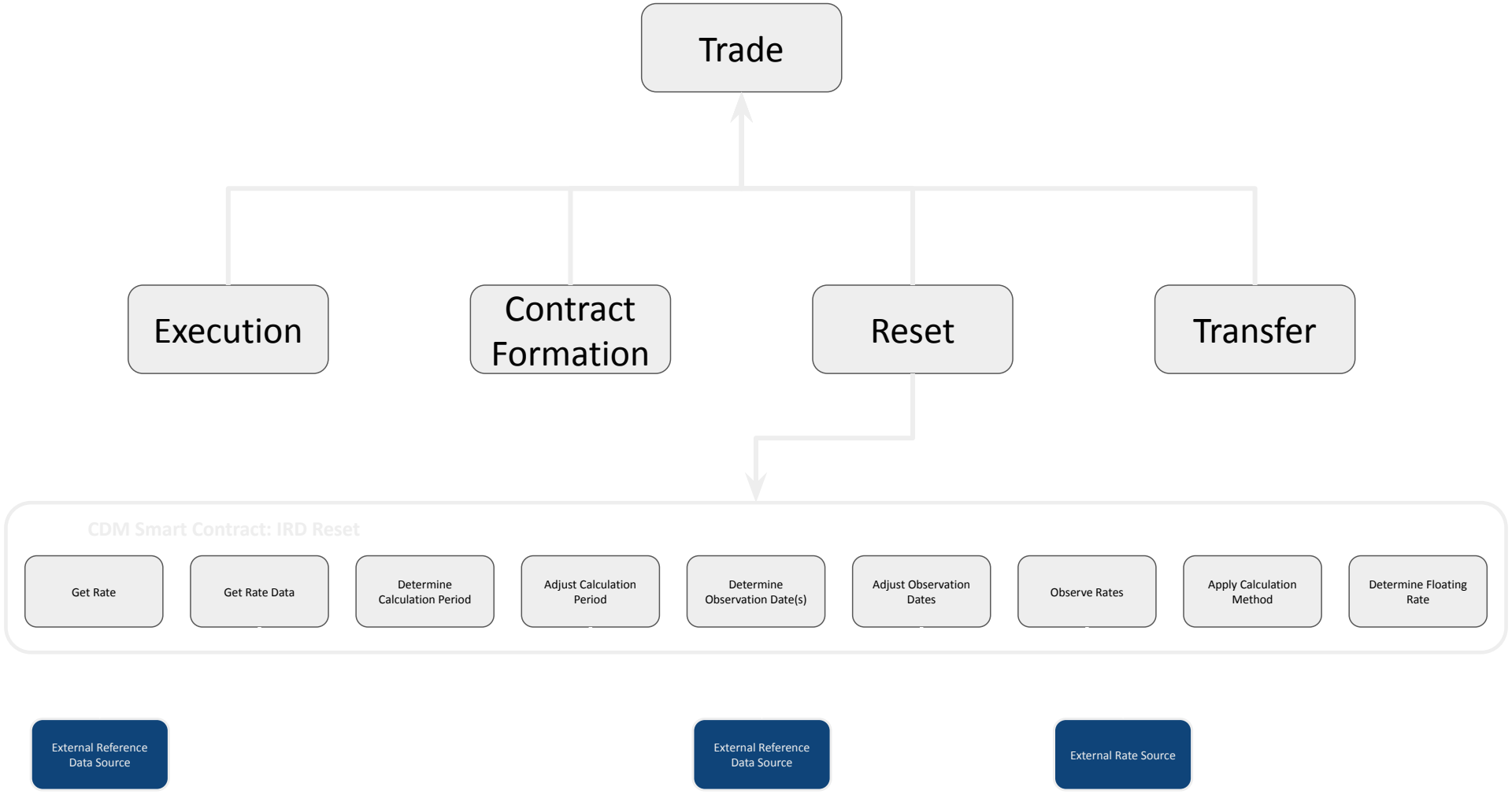

Trade
Execution
Contract Formation
Reset
Transfer
CDM Smart Contract
Get Rate
Get Rate Data
Determine Calculation Period
Adjust Calculation Period
Determine Observation Date(s)


Trade
Execution
Contract Formation
Reset
Transfer
CDM Smart Contract
Get Rate
Get Rate Data
Determine Calculation Period
Adjust Calculation Period
Determine Observation Date(s)
Adjust Observation Dates

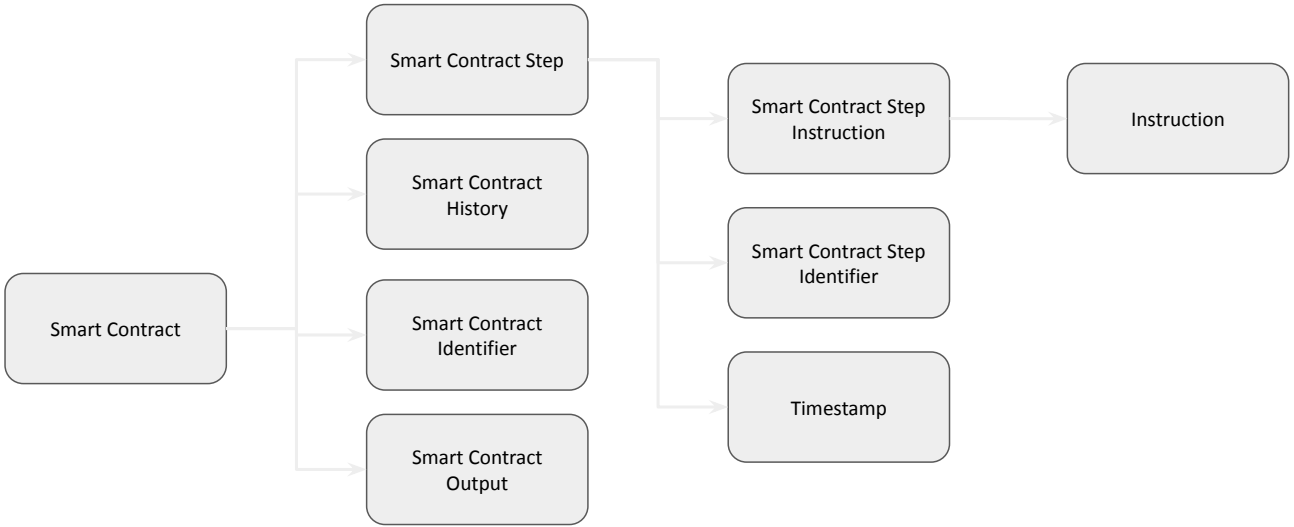

Trade
Execution
Contract Formation
Reset
Transfer
CDM Smart Contract
Get Rate
Get Rate Data
Determine Calculation Period
Adjust Calculation Period
Determine Observation Date(s)
Adjust Observation Dates
Observe Rates


Trade
Execution
Contract Formation
Reset
Transfer
CDM Smart Contract
Get Rate
Get Rate Data
Determine Calculation Period
Adjust Calculation Period
Determine Observation Date(s)
Adjust Observation Dates
Observe Rates
Apply Calculation Method

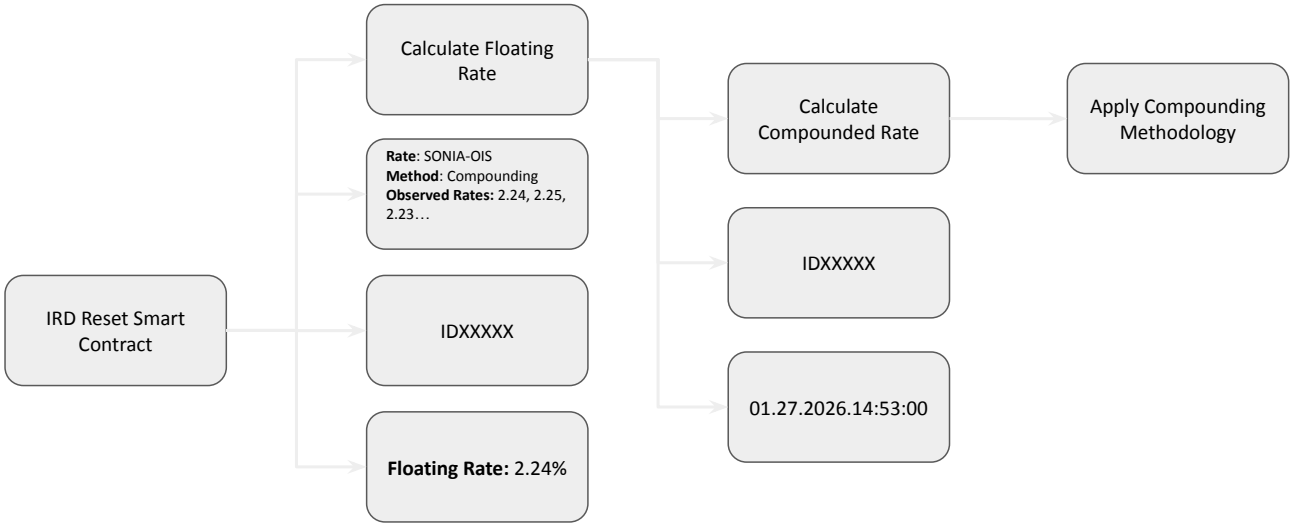

Trade
Execution
Contract Formation
Reset
Transfer
CDM Smart Contract
Get Rate
Get Rate Data
Determine Calculation Period
Adjust Calculation Period
Determine Observation Date(s)
Adjust Observation Dates
Observe Rates
Apply Calculation Method
Determine Floating Rate


Trade
Execution
Contract Formation
Reset
Transfer
CDM Smart Contract: IRD Reset
Get Rate
Get Rate Data
Determine Calculation Period
Adjust Calculation Period
Determine Observation Date(s)
Adjust Observation Dates
Observe Rates
Apply Calculation Method
Determine Floating Rate
External Reference Data Source
External Reference Data Source
External Rate Source

CDM Smart Contract



CDM Smart Contract



Smart Contract Comparison

Blockchain Smart Contracts

- Typically blockchain code that executes simple conditional logic e.g. *“if X, then transfer Y.”*
- Treats code as the contract, often detached from underlying legal documentation.
- Designed for token transfers and self-contained execution on a specific ledger.
- Limited context, poor auditability, and difficult to amend once deployed.

The CDM Smart Contract Model

- Starts from legal and market standards (eg. ISDA Definitions).
- Treats law as code, faithfully expressing contractual terms as modular, executable logic.
- Each Smart Contract Step performs a defined legal function.
- Outputs remain CDM-native, ensuring interoperability across institutions, systems, and ledgers.