



CDM Business Cases

TRADE EXECUTION

CHRIS.RAYNER@ISLAEMEA.ORG

1	INTRODUCTION	3
2	EXECUTIVE SUMMARY	3
3	BUSINESS PROBLEM	4
4	ALTERNATIVE SOLUTIONS	5
4.1	VENDOR SOLUTIONS	5
4.2	BUILDING INHOUSE	6
5	PREFERRED SOLUTION	6
6	SCOPE	9
6.1	HARDWARE	9
6.2	SOFTWARE.....	9
6.3	TECHNICAL RESOURCE	9
7	APPENDIX	10

1 Introduction

The purpose of this document is to present a business case on how the Common Domain Model (CDM) can be used to resolve some of the issues found in the trade execution processes used today.

The CDM is a freely available Open Source model maintained by the Fintech Open Source Foundation (FINOS). It can be used to model financial products and events, supplying business logic and functions that manage a trade's lifecycle as it transitions from one state to another. The model has been developed by trade associations and industry professionals for the benefit of all market participants, harmonising and standardising data structures and processing.

The CDM has been written to be language and infrastructure agnostic, one of its guiding principles being to provide objects that can be distributed across multiple applications and technologies. This reusability and interoperability places the CDM firmly at the forefront of the market in our drive towards a digital future.

2 Executive Summary

At trade execution all the components of a new securities loan are finalised. The economic terms, delivery schedules, clearing details, agreement type etc. all need to be stamped into a structure that represents the trade.

Each party on a trade uses a variety of internal and external systems and applications to perform the execution. A variety of messaging and data standards can also be employed to facilitate the representation and communication of the terms of the trade between parties. This invariably results in frequent mismatching of critical information, most notably causing reconciliation and settlement failures.

The CDM provides an industry standardised structure for a trade, along with the events required to execute it. This means that all applications that use the CDM for trade execution will generate objects that will exactly match; this object can then become the common representation of the trade across all parties involved in it.

This document proposes the use of the CDM as the central component of any application that performs the execution of securities financing trades. With all parties using the structures and functions provided by this Open Source model the likelihood of data disparity across participants will be eliminated.

3 Business Problem

All market participants require a plethora of different systems and infrastructure to manage their different businesses. Whether vendor supplied, self-built, running on a physical server or as a service in the cloud, each offering has its own complexities and nuances – even when a number of them are all attempting to solve the same problem.

Communication between these systems is also fraught with problems. Message formats and standards abound, with translation layers and transformation models implemented across divisions and business lines in order to facilitate the necessary data comprehension.

This is especially true in the case of trade execution, where securities financing details exchanged between parties often differ, resulting in the need for costly trade matching and reconciliation services. The ongoing impact of these trading breaks can then be felt in settlement, with it being generally recognised that failure rates are only fractionally improving, and thus still resulting in significant fines from financial regulators¹.

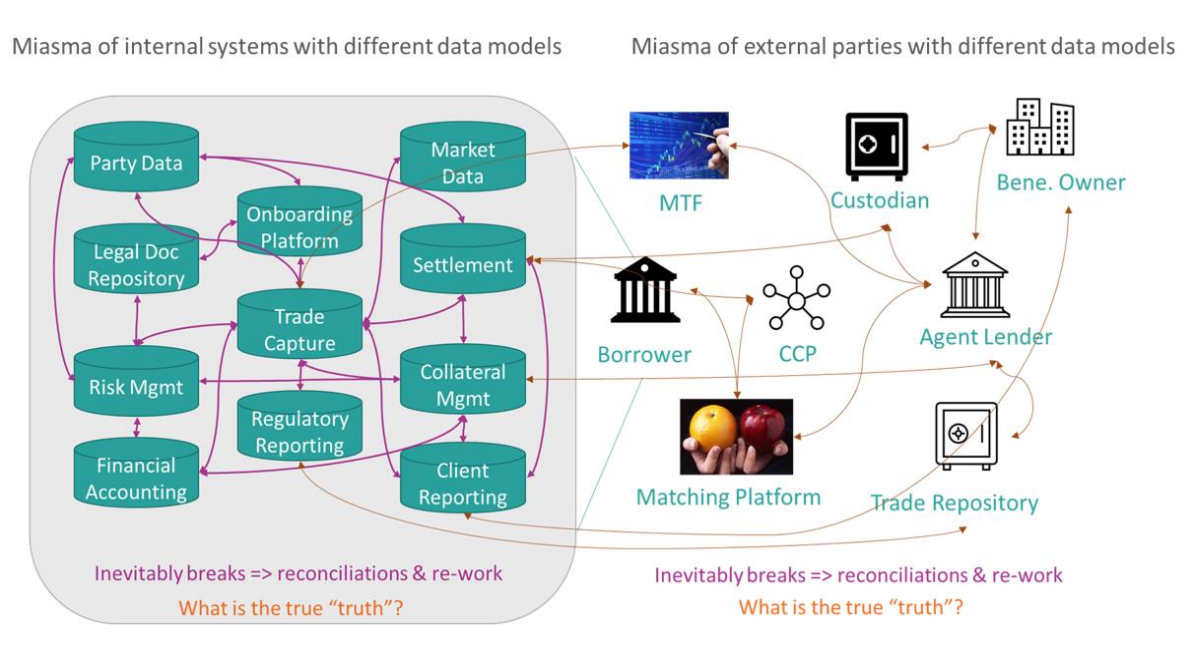


Diagram 1: The Data Model Mess. The difference in booking models inevitably leads to problems like settlement failure, collateral disputes or reconciliation breaks.

Getting the details right at trade execution, the start of the post trade cycle, will effectively reduce the probability of breaks occurring further on during the trade lifecycle. Thus, a solution that can be used by all parties to generate a common representation of a trade is required.

The technological benefits of having a standardised object representation of a trade are not limited to reducing reconciliation efforts. Standardisation improves the efficiency of the entire execution process, allowing systems to be streamlined for a single structure and logic;

¹ Survey carried out by ISLA Market Practice group determined that the settlement rate for new loans in 2021 was 94% against 96% in 2022. Closed loans improved from 87% in 2021 to just under 88% in 2022.

automation of these processes is also enhanced, which in turn provides better throughput and faster turnaround of associated transactions. Indeed, as the work undertaken by ISLA to standardise the pre-trade negotiation workflow has shown, there is considerable scope for improving the first steps in the pre and post trade cycles.

A lot of resources are currently employed on a daily basis resolving issues created by failed, malformed or mismatched trade executions. Operations personnel and systems have to manage and address issues in numerous areas, a few of which would be:

- Settlement – resources required to add missing delivery details, pre-match settlement etc.
- Contract compare – resources to execute and manage comparison with counterparties
- Margins – resources to address margin calls and price renegotiation due to poor executions
- Recalls – resource to manage recall/returns if execution is invalid
- Billing Compare – agreeing misaligned fees/rebates across the entire book

If we can get the trade execution “*right first time*” then a large proportion of these resources would be able to be freed up to engage in more productive work. Using a standard framework for trade execution, as defined in the CDM, will thus provide a real benefit in terms of costs and resources.

4 Alternative Solutions

Before presenting the preferred solution, it is worth briefly describing some of the other avenues that could be adopted to alleviate some of the difficulties seen with trade executions.

There are two main types of solution for this business problem; it’s the age-old question of “build” versus “buy”, i.e. building something in house or buying an existing vendor solution. The pros and cons of each approach are described in this section in relation to trade executions.

4.1 Vendor solutions

Buying an out of the box solution from a vendor is often the quickest way to get up and running. Every vendor solution will have its own way of representing trade entry and hence trade execution. The data required by different vendor applications will differ too, probably requiring several different systems to all work together to perform the execution.

As the number of vendor applications deployed increases, the complexity of system integration and the opportunity for errors also increases. This leads to additional applications being required to monitor and correct issues that arise from poorly formed executions, most notably being services like contract compare and reconciliation.

Although the need for these services will probably never be entirely eliminated, the volume of issues encountered could be significantly reduced if all vendor offerings were using the same data and processing logic.

4.2 Building inhouse

An inhouse solution is often attractive as you are not tied into any particular vendor's idea of how a trade execution should be undertaken. This also allows you to add functionality that is bespoke to your firm, potentially giving you a competitive advantage.

The actual core data in any trade execution across the market should actually be identical though; a company's unique and proprietary data would not be transferred across to other parties. Thus, the economic and legal terms of an execution, which essentially make up the core of any trade execution, should be the same regardless of how an application is built.

5 Preferred Solution

The CDM, with its product, event and legal documentation dimensions, is well recognised as the key to unravelling the data model mess illustrated earlier (in Diagram 1). With an entire community actively enhancing and extending it, the CDM must be considered as a core component of any securities financing solution.

For these reasons, the best solution would be to:

- Either build an inhouse solution that has the CDM at its heart,
- Or use a vendor solution that has native links to the CDM, or has the CDM integrated into its core

The current application and service infrastructure employed to service trade executions and the passing of these details to the post trade systems is complicated and requires a lot of systems all using their own messages and protocols.

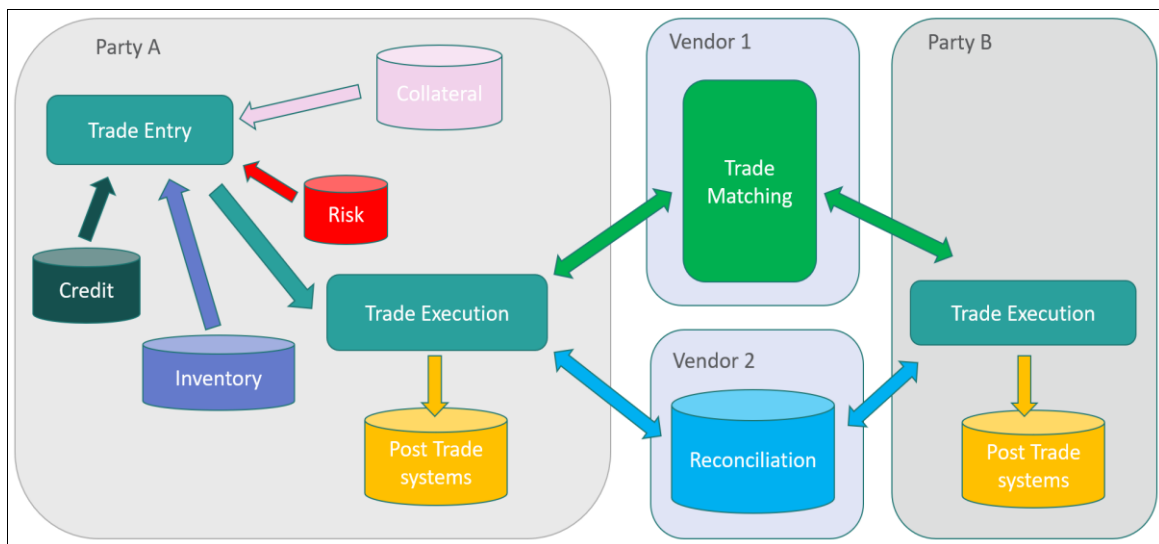


Diagram 2: A basic representation of the application and service infrastructure requirements to perform a bilateral trade execution. Note that each system has its own messaging and data requirements, and there is a necessity for matching and reconciliation services.

With the CDM at the heart of an application, service architecture or trading environment, the need for additional data transformations between systems is reduced. This allows an iterative approach to implementation, as existing systems can keep using their own messages and data, with the CDM initially facilitating the generation of the standardised trade execution object.

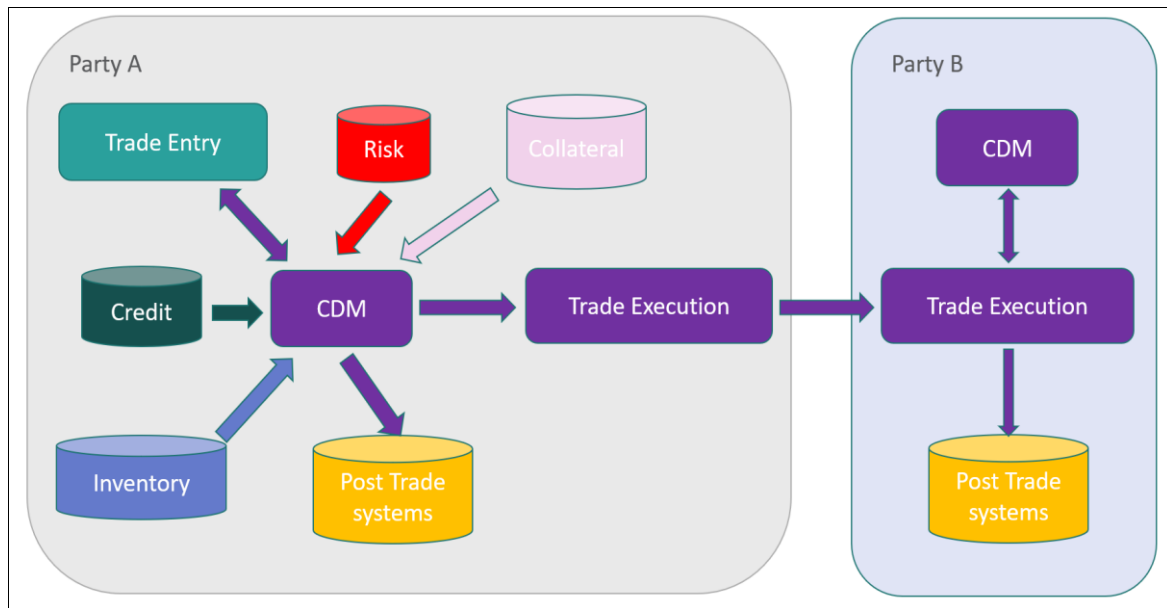


Diagram 3: The CDM can be incrementally introduced to a system's architecture. In this initial phase the applications and services still use their own data formats, with the CDM creating a standard trade execution object that can be passed to other internal or external systems.

The end goal would be to have all systems involved in the process able to communicate natively with the CDM, delivering a streamlined, efficient and transparent process.

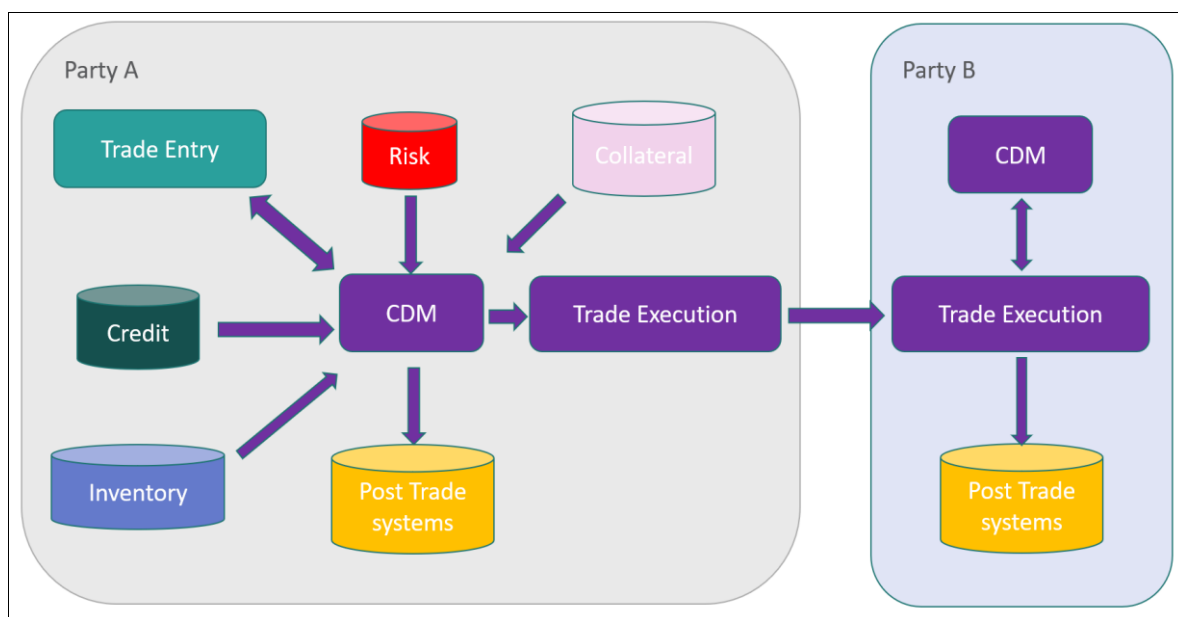


Diagram 4: In this environment all the artifacts involved in the trade execution can natively communicate with the CDM, allowing a seamless transfer of data in a single format from location to location. Note also that the necessity for intermediary applications like trade matching or contract compare platforms has been eliminated, as have post trade reconciliation services.

Often mistakenly believed to be “just another data standard”, the CDM goes far beyond this, not only allowing complex financial products to be represented in a standard format, but also standardising the business logic used to manage the events that can affect a trade throughout the duration of its lifecycle.

Furthermore, with the imminent introduction of the legal terms of the trade into the model this year, the CDM will be able to describe both the economic and the legal terms of the trade.

The CDM is also preparing us for the use of technologies like smart contracts and blockchain. If the trade execution includes the legal terms of the trade then the object created by the CDM can define how to react in the event of specific corporate actions for example. This will allow smart contracts to be built that will automatically react to market transactions covered in the trade’s contractual terms.

Once a trade has been executed the CDM will have produced an object that represents that trade. This object can be hashed and put onto a blockchain. If all parties on the trade are using the CDM to manage the trade execution, then it would be a simple task to compare the object or its hash to confirm that the object created is identical across all parties. This object hash can then become the common representation of the trade and used for validation, reconciliation or regulatory reporting purposes.

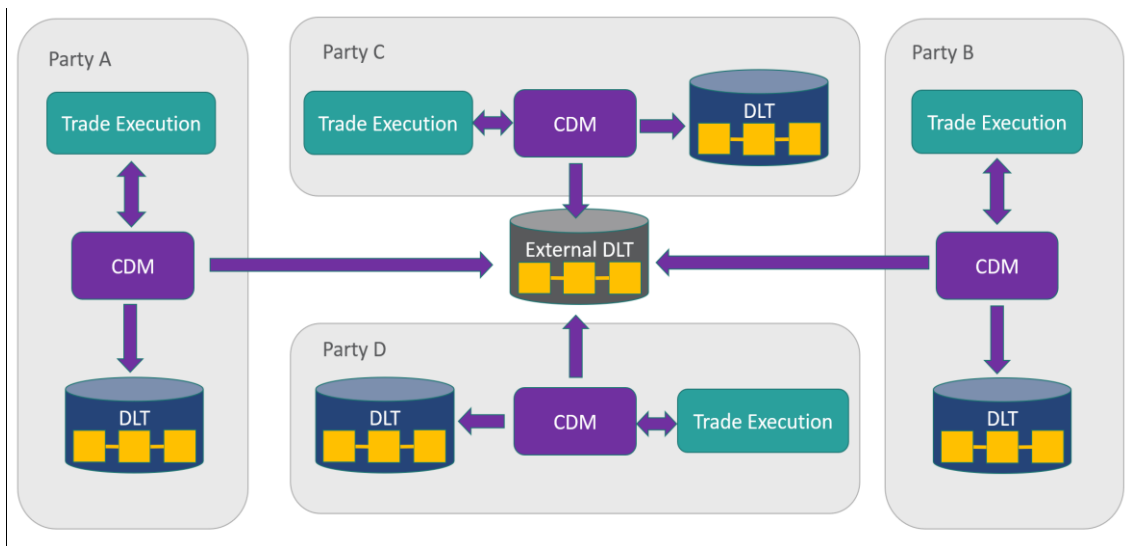


Diagram 5: The hash of the CDM object can be put onto a local or external blockchain. This immutable data can become the common representation for a trade across all parties. An instance of the CDM could be used by each party, as depicted above, or a single instance could be run as a service alongside the external DLT platform.

Mappings to existing data standards are also available out of the box, with more being extrapolated and defined as the model continues to evolve. Standards that deal with trade executions or confirmations (SWIFT, FpML, FIX) are thus able to be consumed by the CDM.

Aside from the functional and technological advantages of the CDM it is also worth reiterating that the model is freely available, production ready, and being continually updated by market professionals. The cost benefits of having a community driven project of this scale are tremendous, with adopters of the CDM being able to realise a potential 50 to 70% increase in efficiency² by restructuring their data to support it.

² The 50 to 70% increase in efficiency was stated by Dr Lee Braine at the CDM Showcase on February 21st 2023. One of the specific examples he used was that restructuring the data for swaps to use the CDM would result in a more than 50% improvement in efficiency.

6 Scope

Every site will have different requirements when considering an installation of the CDM. Thus, the scope of each project involving the CDM will be different. There are 3 main areas to consider when scoping any project – hardware, software and technical resources.

6.1 Hardware

The choice of hardware and where to house an instance of the CDM is independent of the CDM itself. The CDM can be run on physical or virtual servers and can be easily run as a service in the cloud. The decision as to what infrastructure to use will primarily be down to the organisation themselves, as the CDM can be deployed across all mainstream environments and architectures.

6.2 Software

The CDM is provided as a series of java libraries that can be embedded into new or existing applications, or deployed as a service layer. Some basic code generators are also provided that expose aspects of the model in other programming languages too, like C# or Typescript. Members of the community are also encouraged to write and contribute code generators for other languages like Python for example.

This means that the scope of any software development is again down to the implementation. Deployment in java applications or as a service will be the simplest, with more work required if the CDM is required in other programming languages.

6.3 Technical resource

Assuming the necessary hardware and software expertise has already been determined, the one remaining resource consideration is technical personnel, especially with experience of trade execution either with the CDM or in the market in general.

An understanding of the CDM and how trade executions are modelled within it can be obtained by consulting the CDM Resource library available at FINOS. Example objects and function calls are also available from the same site, and within the Rosetta application provided by REGnosys.

FINOS and the trading associations (ISLA, ISDA and ICMA) host working groups where modelling concepts and structures are discussed which would also be beneficial. FINOS groups also exist where questions on CDM technology, implementation and architecture are discussed.

In summary, each project looking to use the CDM will have different resourcing requirements. It is also important to realise that only a small portion of the model needs to be utilised to support trade executions. This allows an iterative approach to be used when developing with the CDM, as only the data/functions specific to the business event that an application is looking to use need to be implemented.



7 Appendix

For more information about the CDM please go to the Common Domain Model microsite hosted by FINOS here: <https://www.finos.org/common-domain-model>

For more information about the work being done by ISLA please go to the main International Securities Lending Association website here: <https://www.islaemea.org/>