Domain-Driven Design Representation of Monolith Candidate Decompositions

Miguel Levezinho - INESC-ID, University of Lisbon Instituto Superior Técnico
Stefan Kapferer - OST Eastern Switzerland University of Applied Sciences
Olaf Zimmermann - OST Eastern Switzerland University of Applied Sciences
António Rito Silva - INESC-ID, University of Lisbon Instituto Superior Técnico

11 September 2024

Table of Contents

I Motivation

When does the need arise to migrate Monoliths to Microservices?

What is DDD and why should it be used in Microservice design?

Why migrate to Microservices?

Monoliths tend to be harder to manage as size and complexity grows

- Increased maintainability cost
- Lack of scalability for Cloud deployment
- Hinders agile development

Why migrate to Microservices?

Modular structure with **strong boundaries** between services.

Speed up production with independent **agile development.**

Different **scalability options** with **service tailored infrastructures.**

Why use Domain-Driven Design?

Design software with a **focus on the business** domain.

Use **tactical** and **strategic design patterns** to **model** complex domains.

II Problem Statement

Approach

Mono2Micro

Modular and extensible for the **identification of microservices** in monolith systems

Focuses on **identifying transactional contexts** in the monolithic code, based on entity accesses

Context Mapper

Contains a robust **DSL for representing DDD** concepts, called CML

Provides many peripheral modules, which are designed to **facilitate architects in refactoring** activities

Research Questions

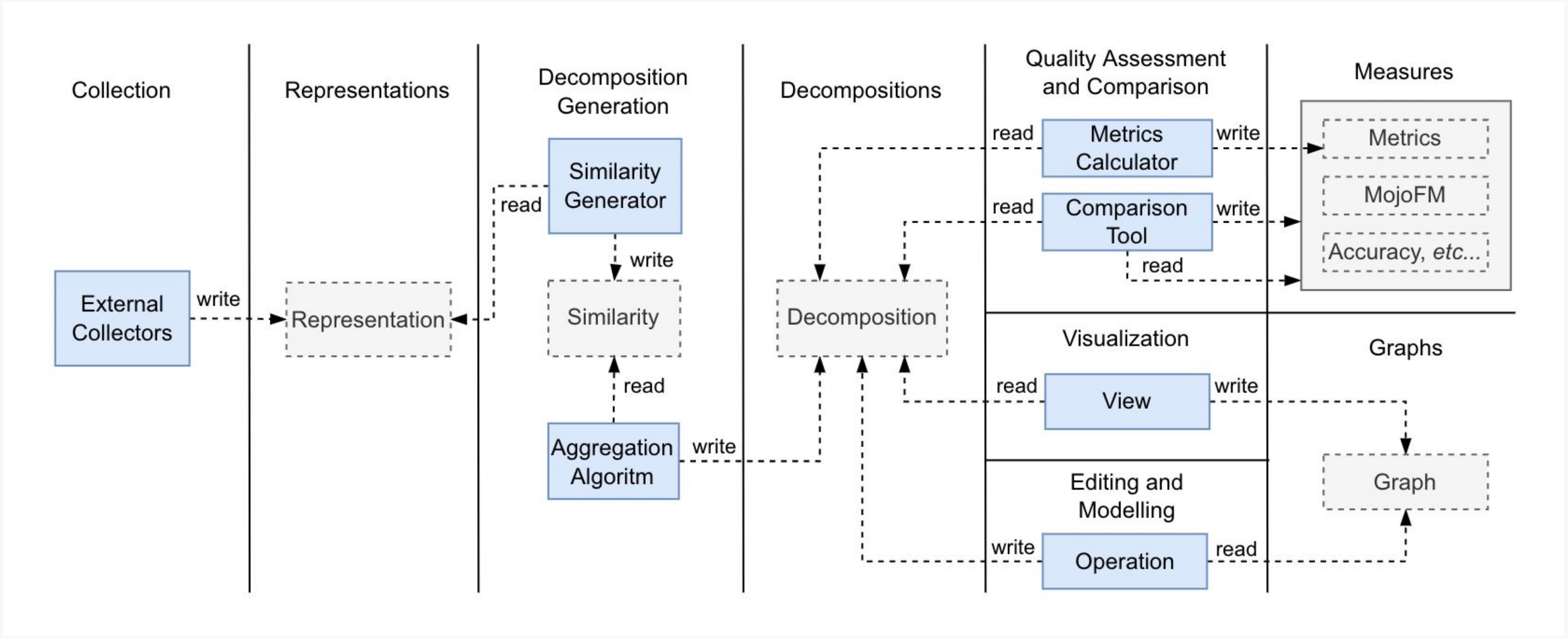
RQ1: How can current approaches to the identification of microservices in monolith systems be extended to include DDD.

RQ2: Can the results of a candidate decomposition based on entity accesses be represented in terms of DDD?

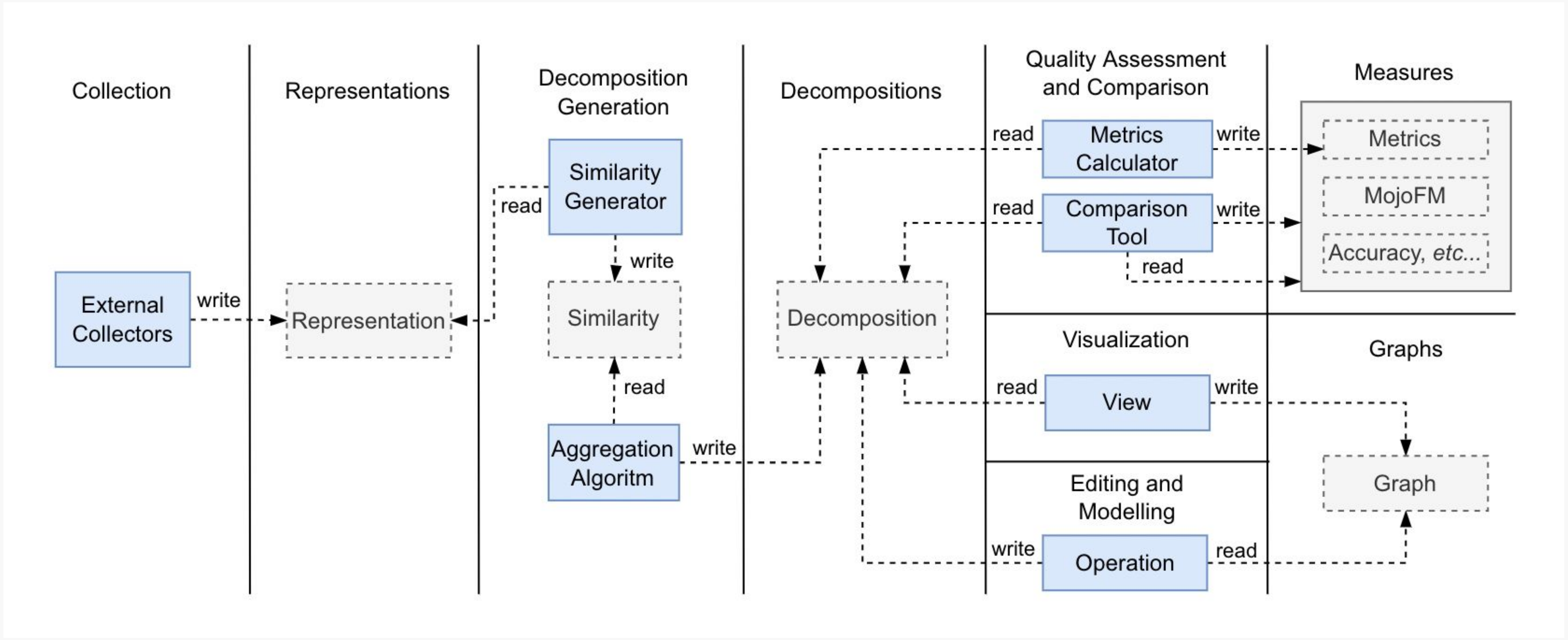
RQ3: Can an architect benefit from the use of a tool that integrates DDD when analyzing and working on a candidate decomposition?

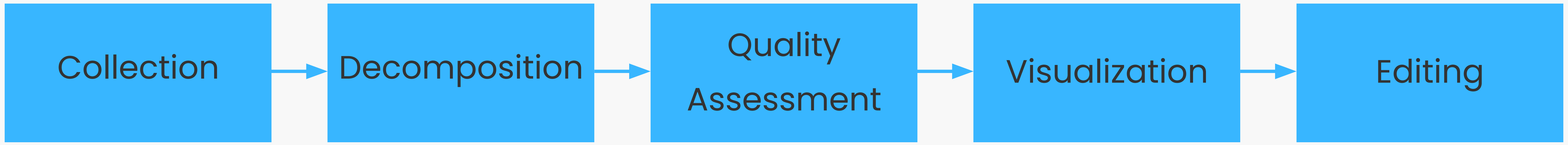
III Solution Architecture

Mono2Micro - Pipeline Architecture

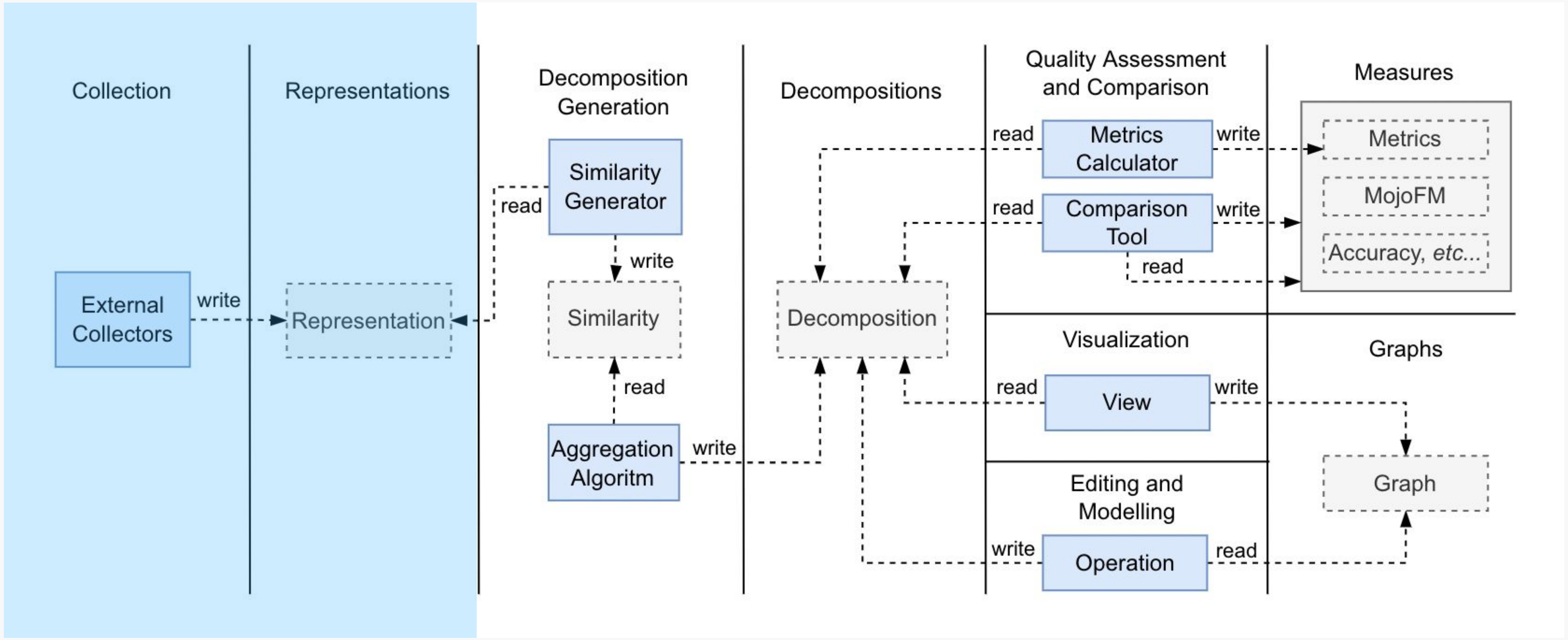


Mono2Micro - Pipeline Architecture



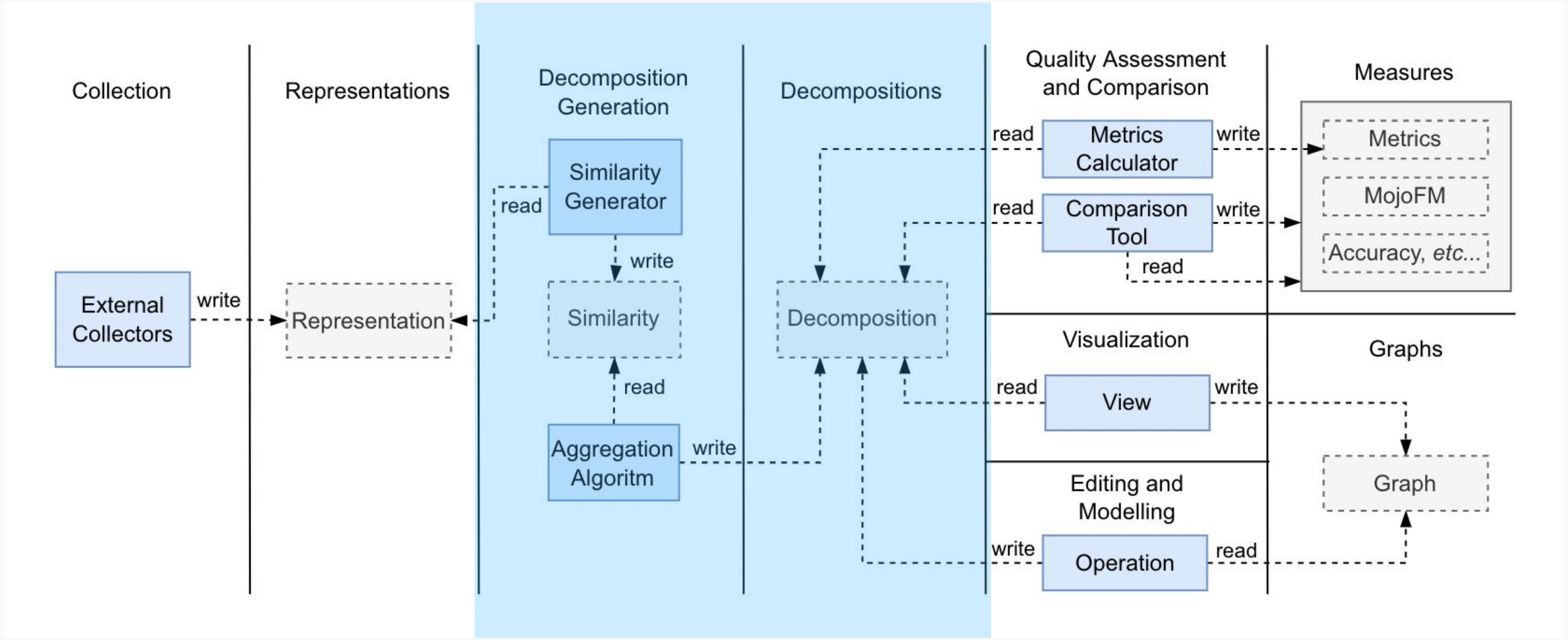


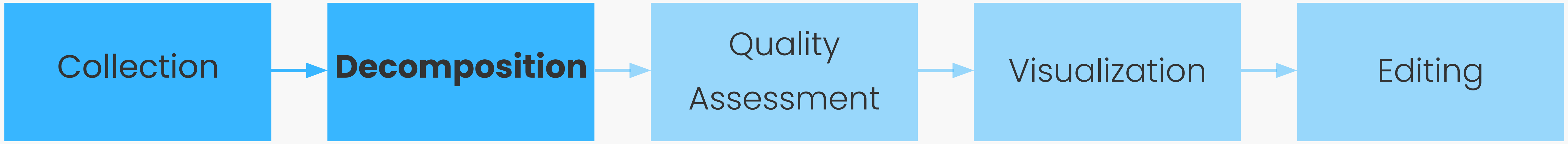
Mono2Micro - Pipeline Architecture



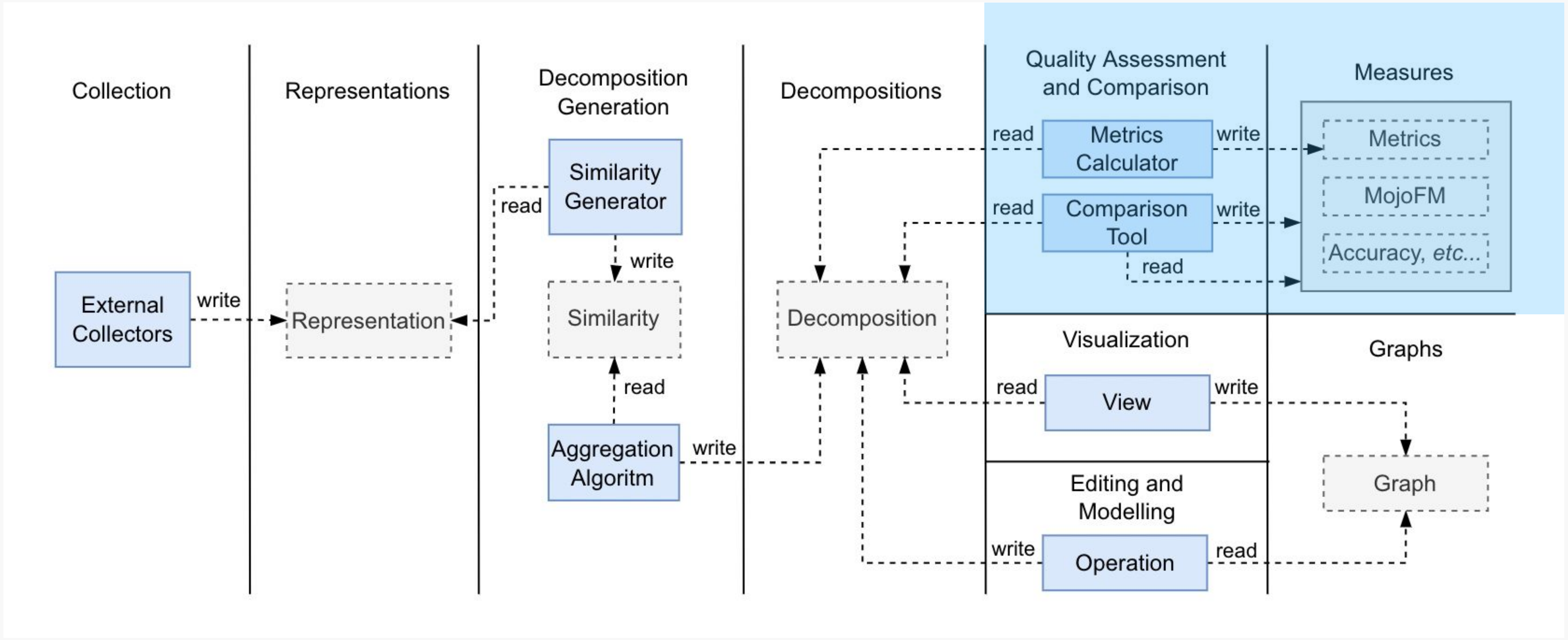
Collection → Decomposition → Quality Assessment → Visualization → Editing

Mono2Micro - Pipeline Architecture



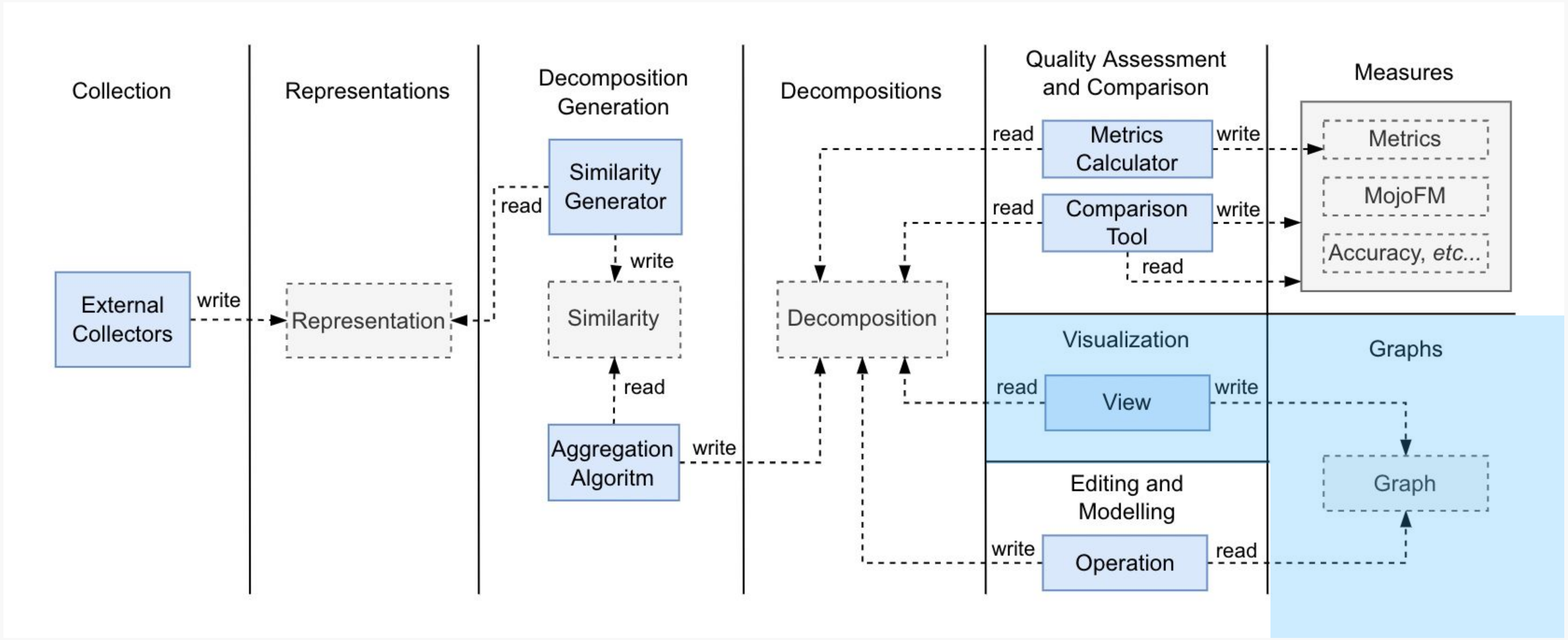


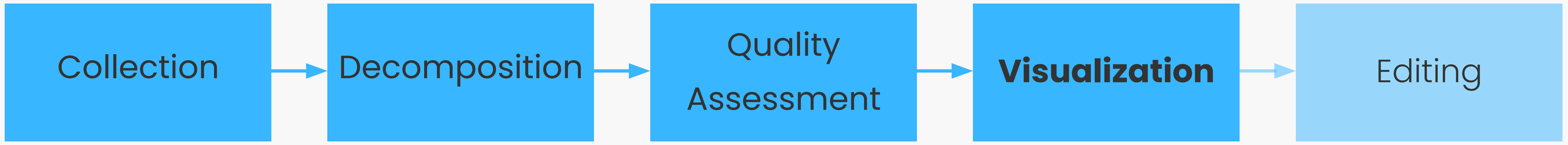
Mono2Micro - Pipeline Architecture



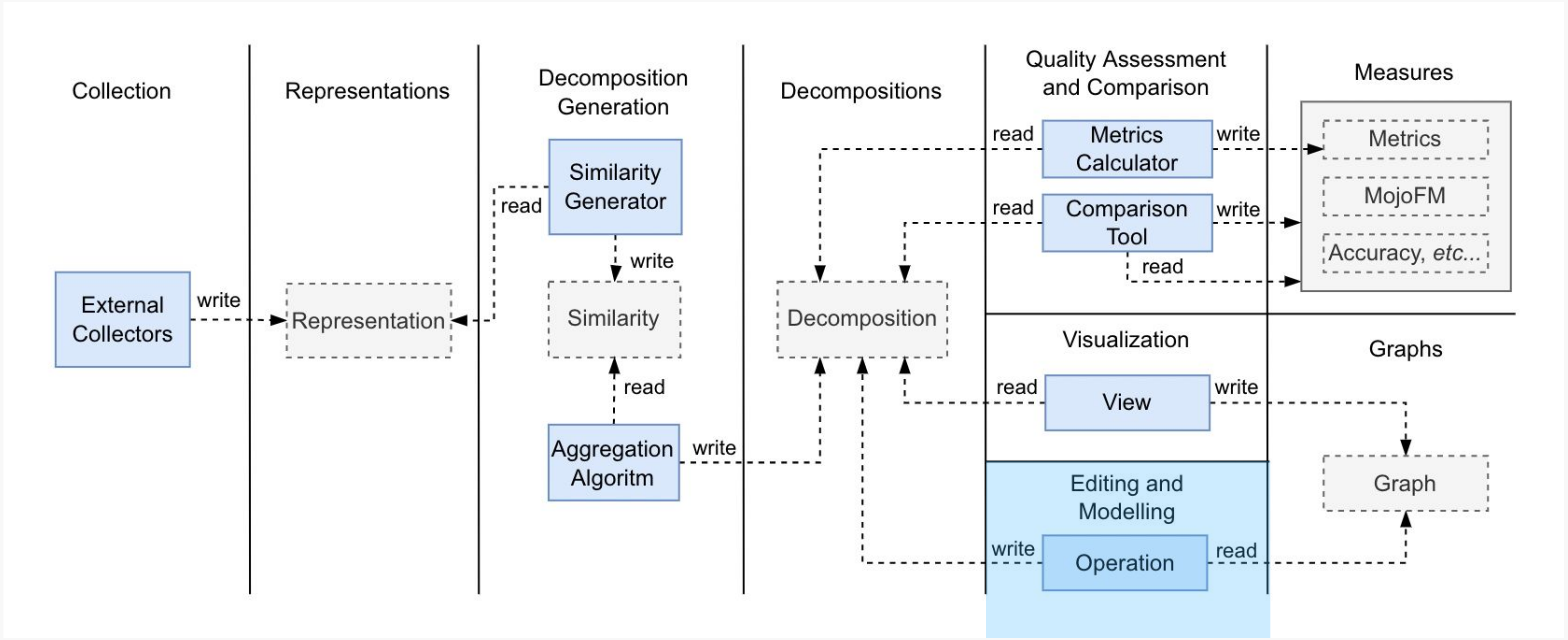
Collection → Decomposition → **Quality Assessment** → Visualization → Editing

Mono2Micro - Pipeline Architecture



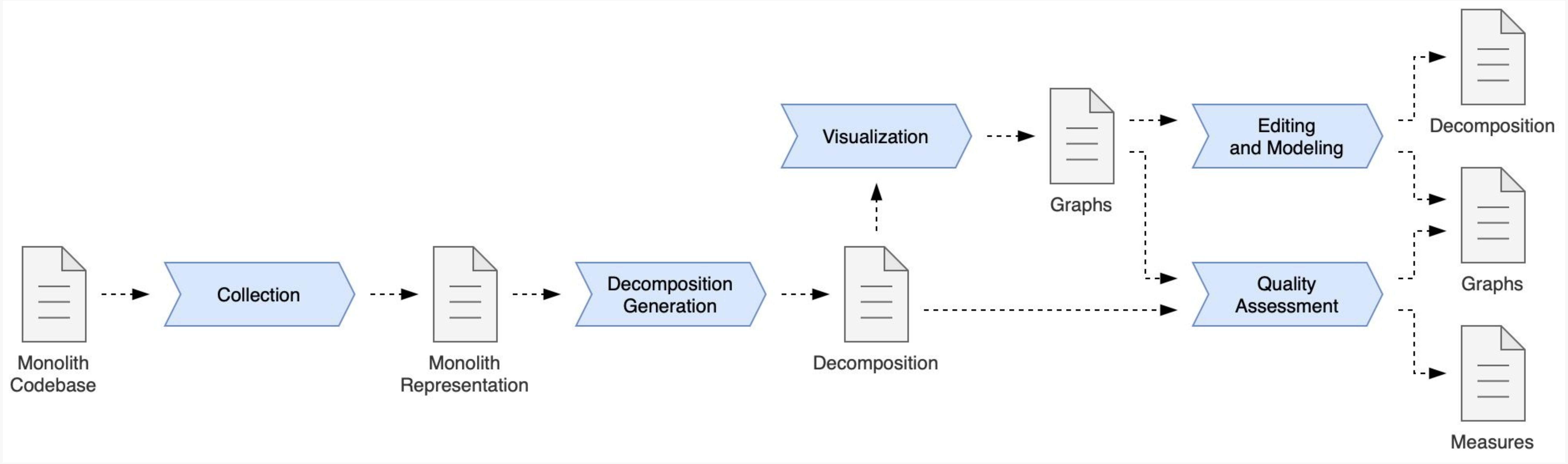


Mono2Micro - Pipeline Architecture

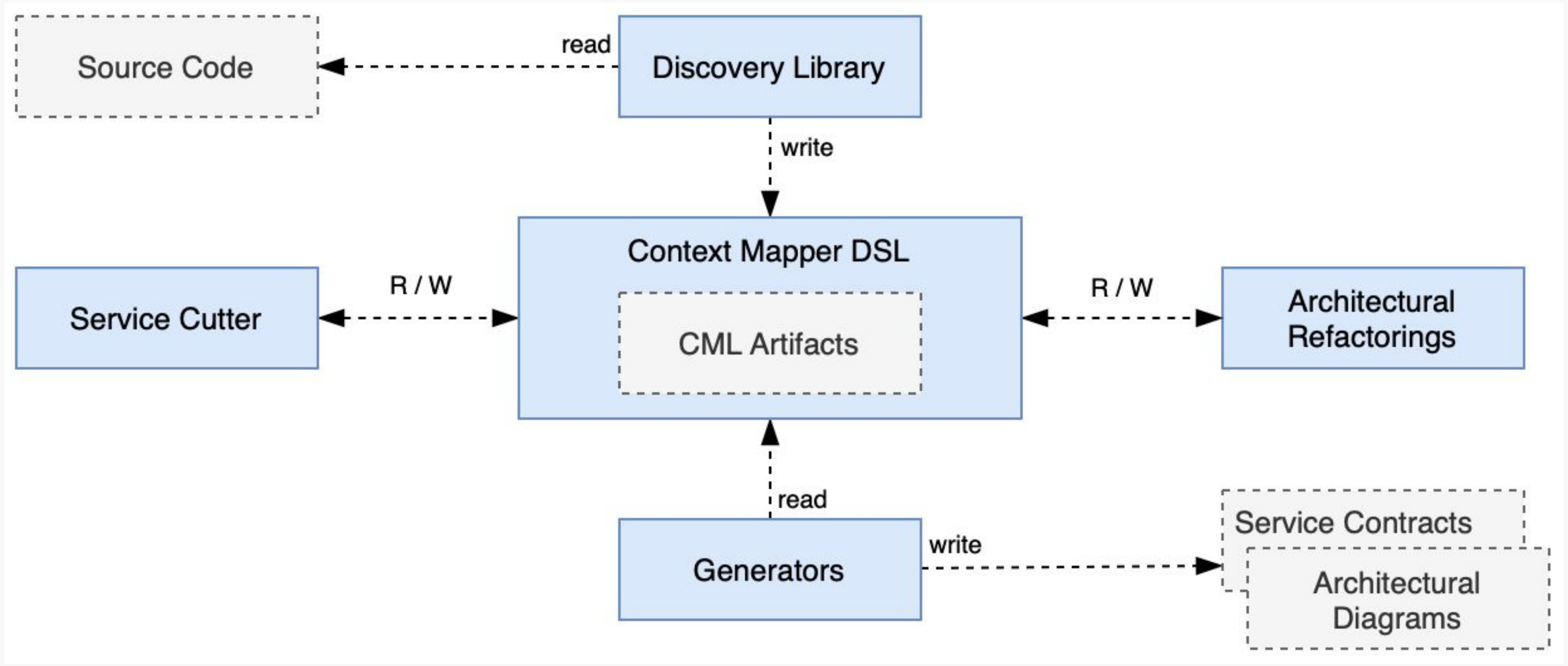


Collection → Decomposition → Quality Assessment → Visualization → **Editing**

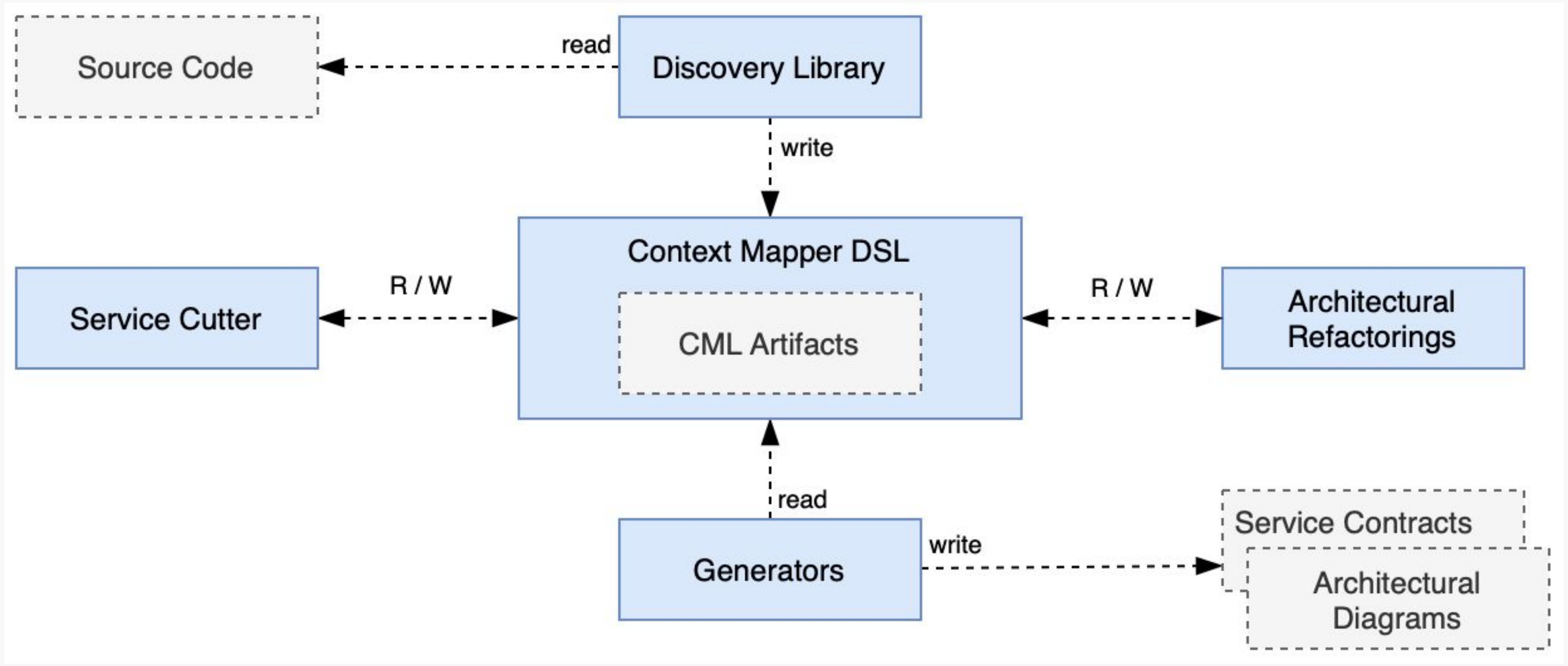
Mono2Micro - Pipeline Architecture



Context Mapper - Hub and Spoke Architecture



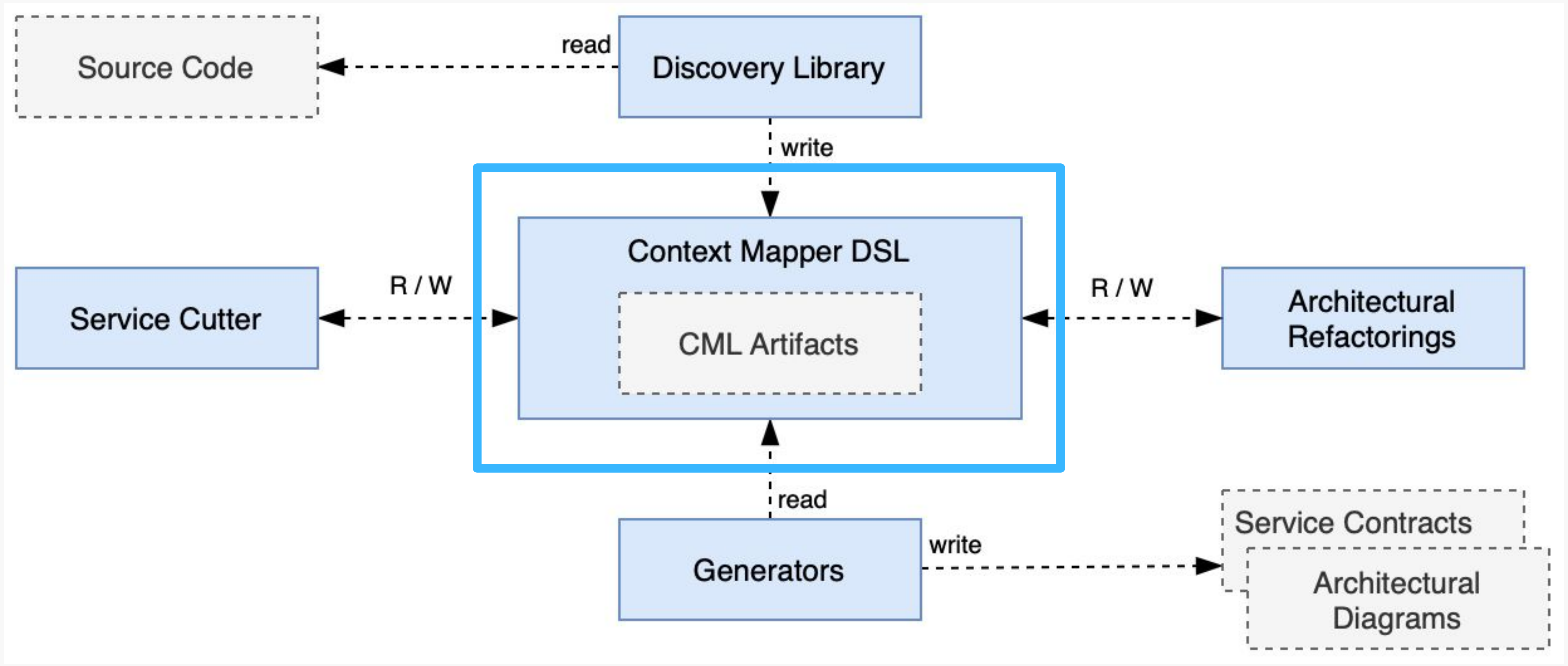
Context Mapper - Hub and Spoke Architecture



CML

Discovery Library

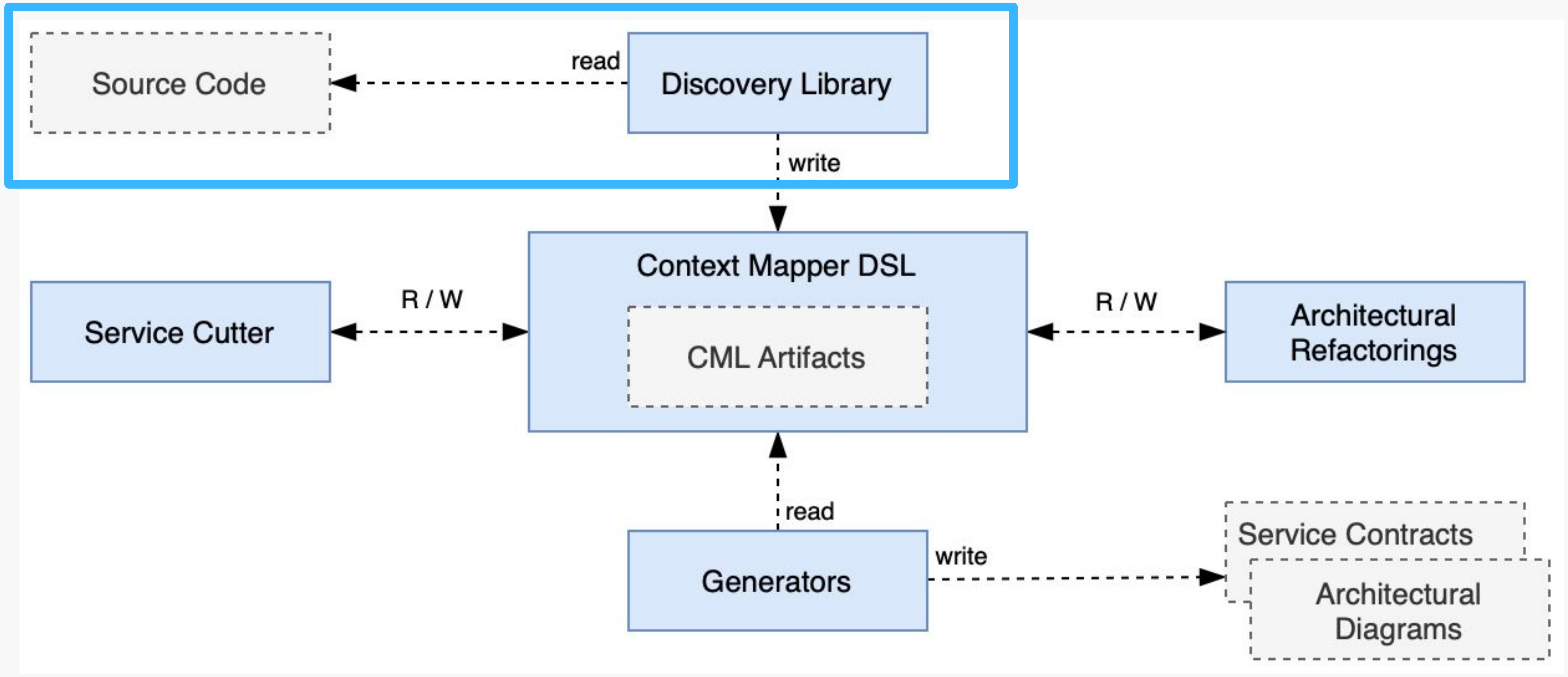
Context Mapper - Hub and Spoke Architecture



CML

Discovery Library

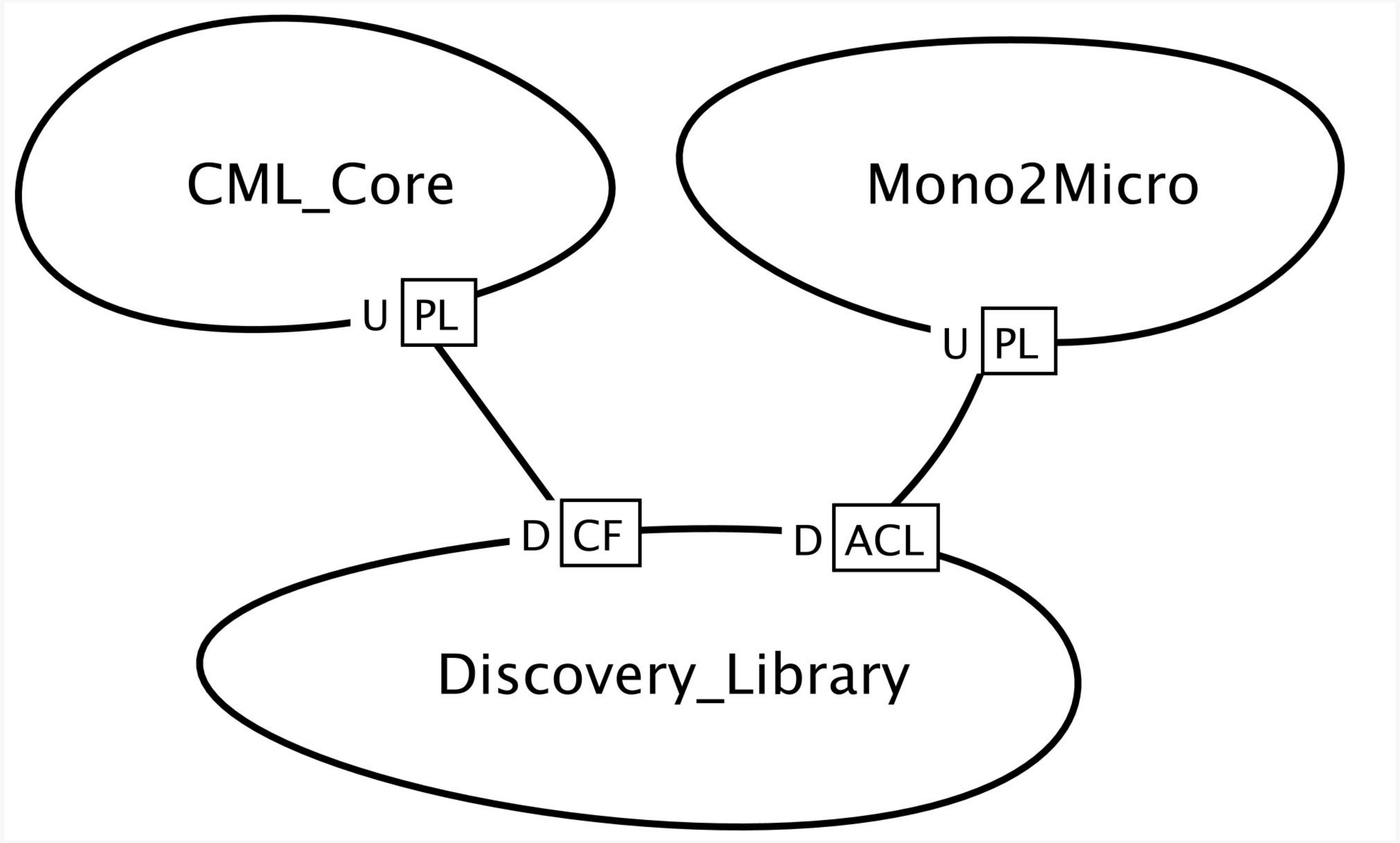
Context Mapper - Hub and Spoke Architecture



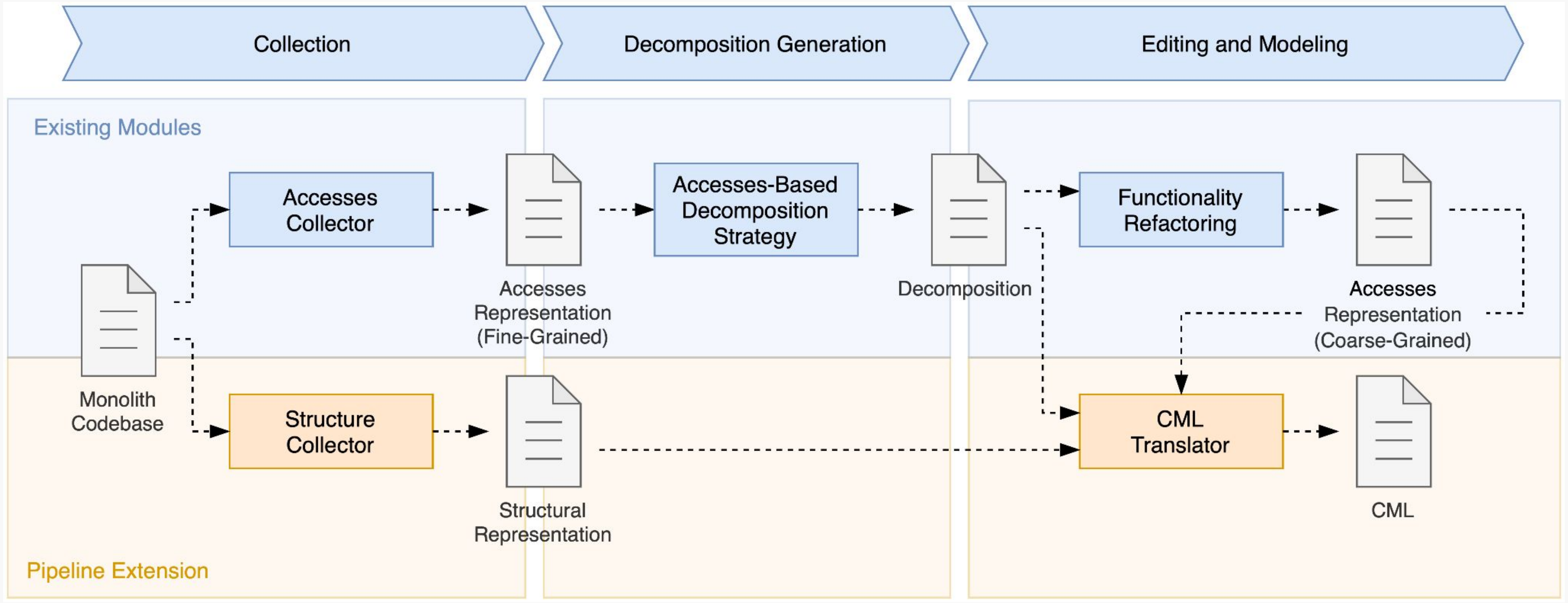
CML

Discovery Library

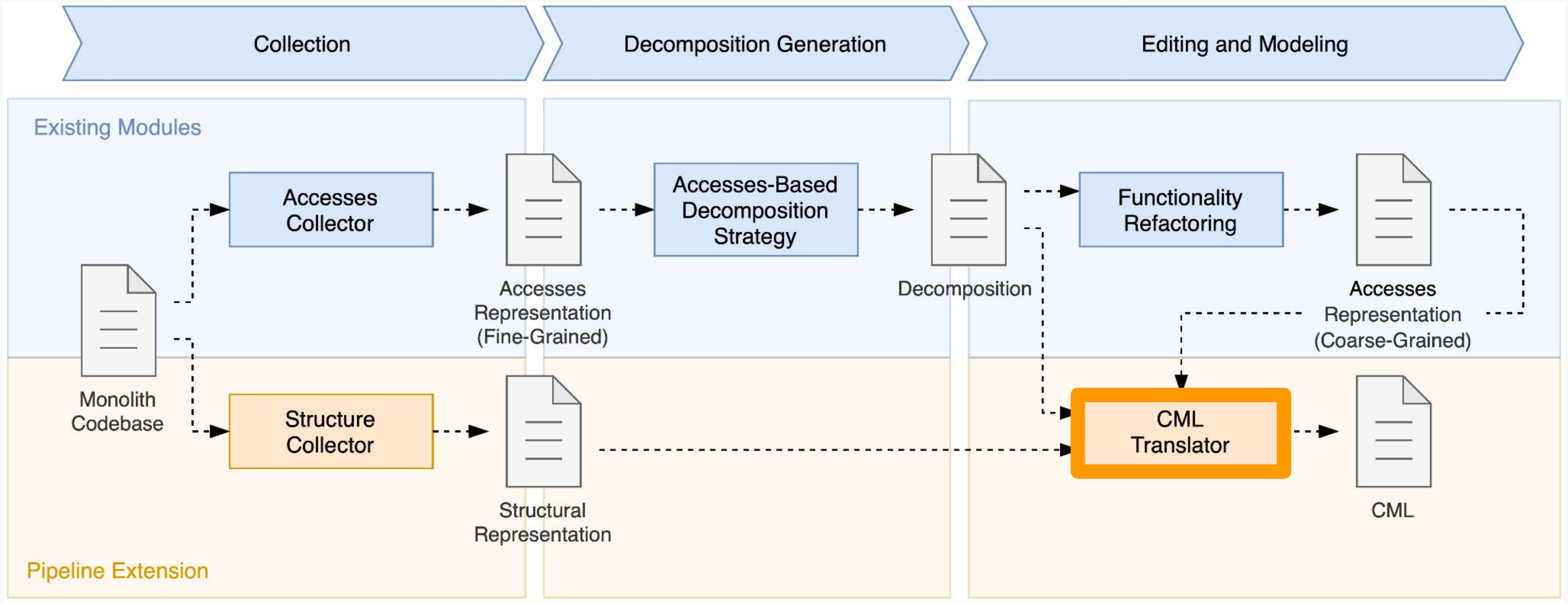
Integration Strategy in a Context Map Diagram



Mono2Micro Pipeline Extension



Mono2Micro Pipeline Extension



Mapping Strategy

A decomposition can be represented by three main concepts:

Entities

Represent domain classes in the source code

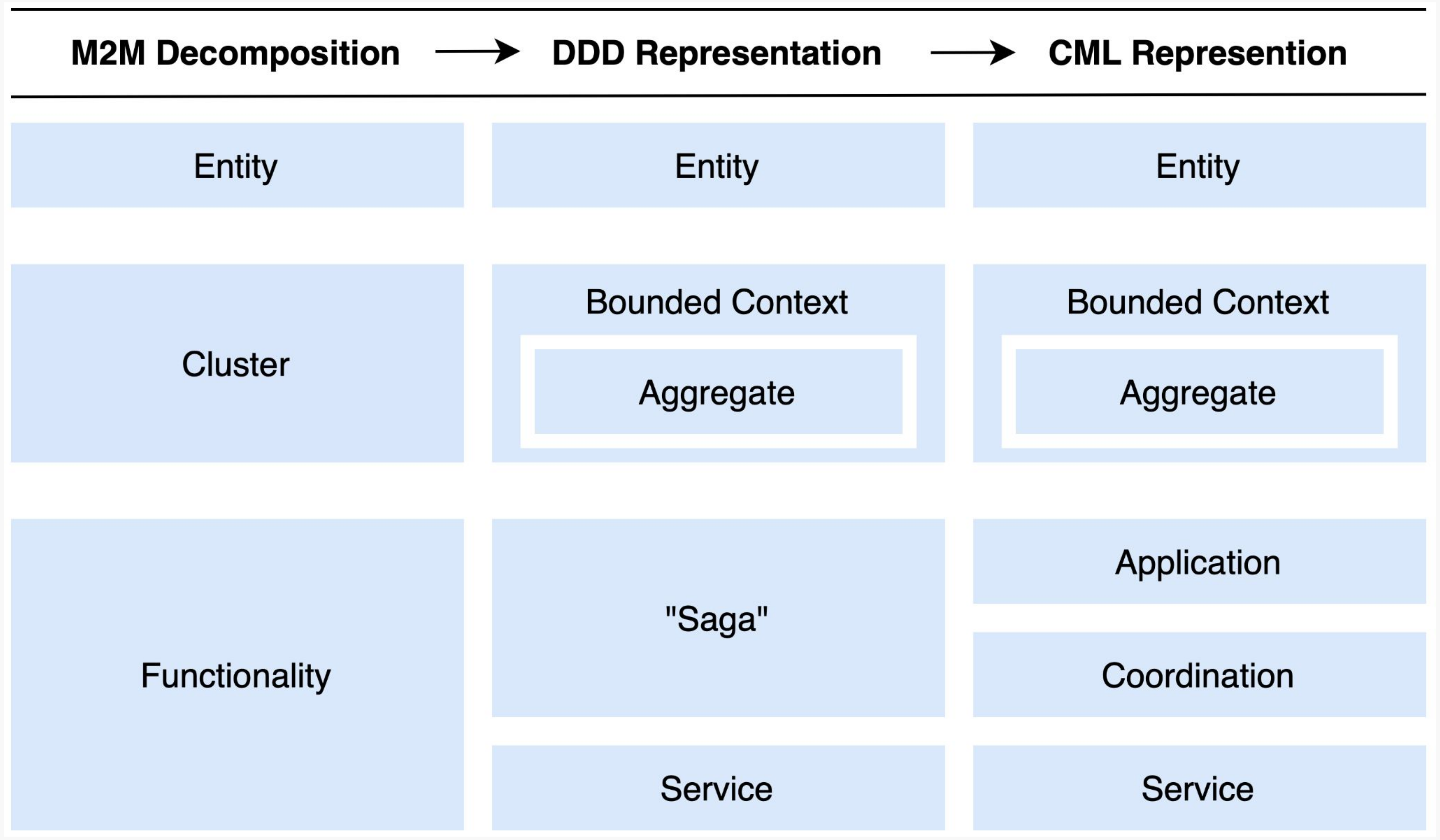
Clusters

Represent a set of entities grouped by similarity criteria

Functionalities

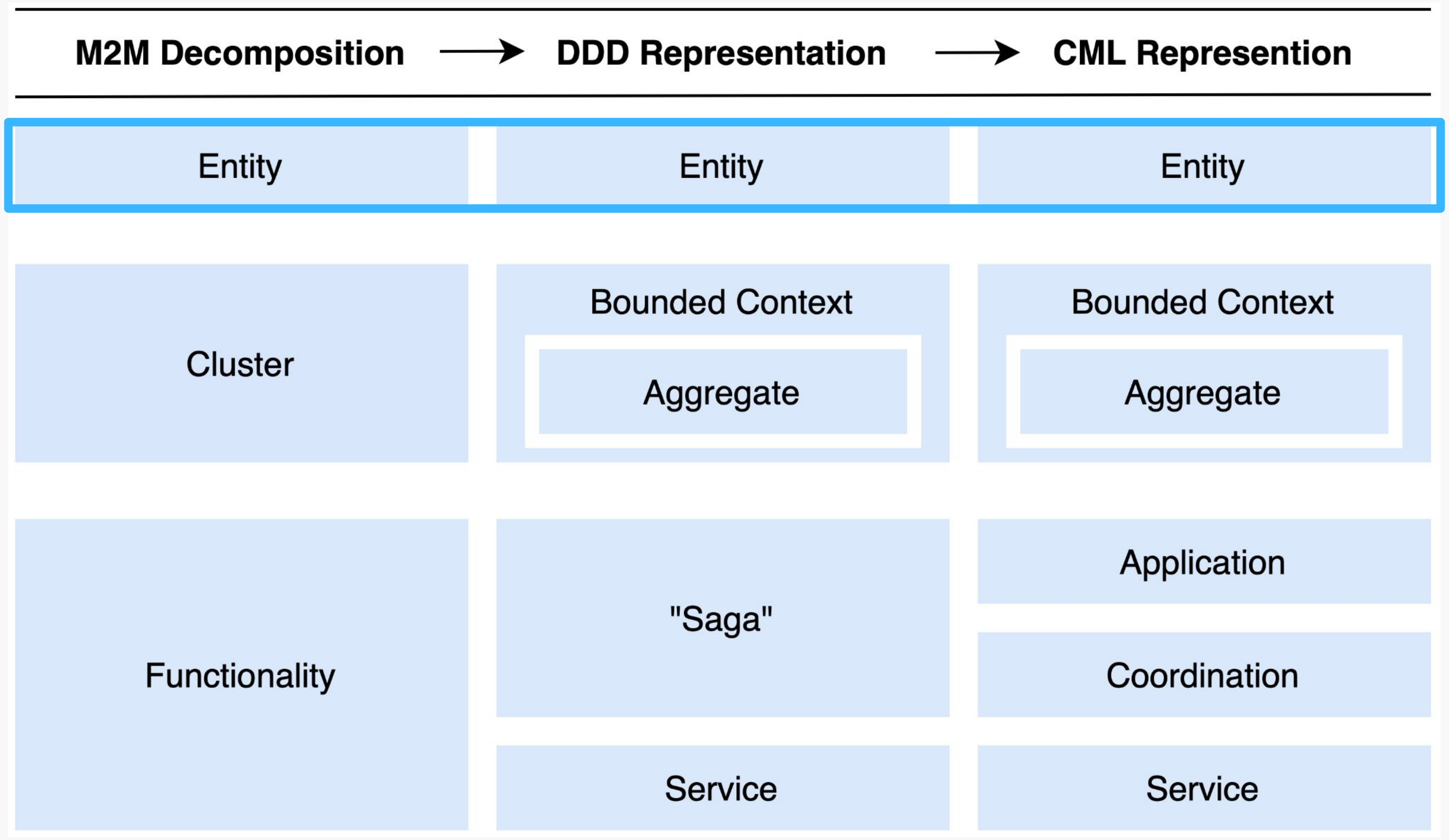
Represent a sequence of read/write accesses to entities in one or more clusters

Mapping Strategy



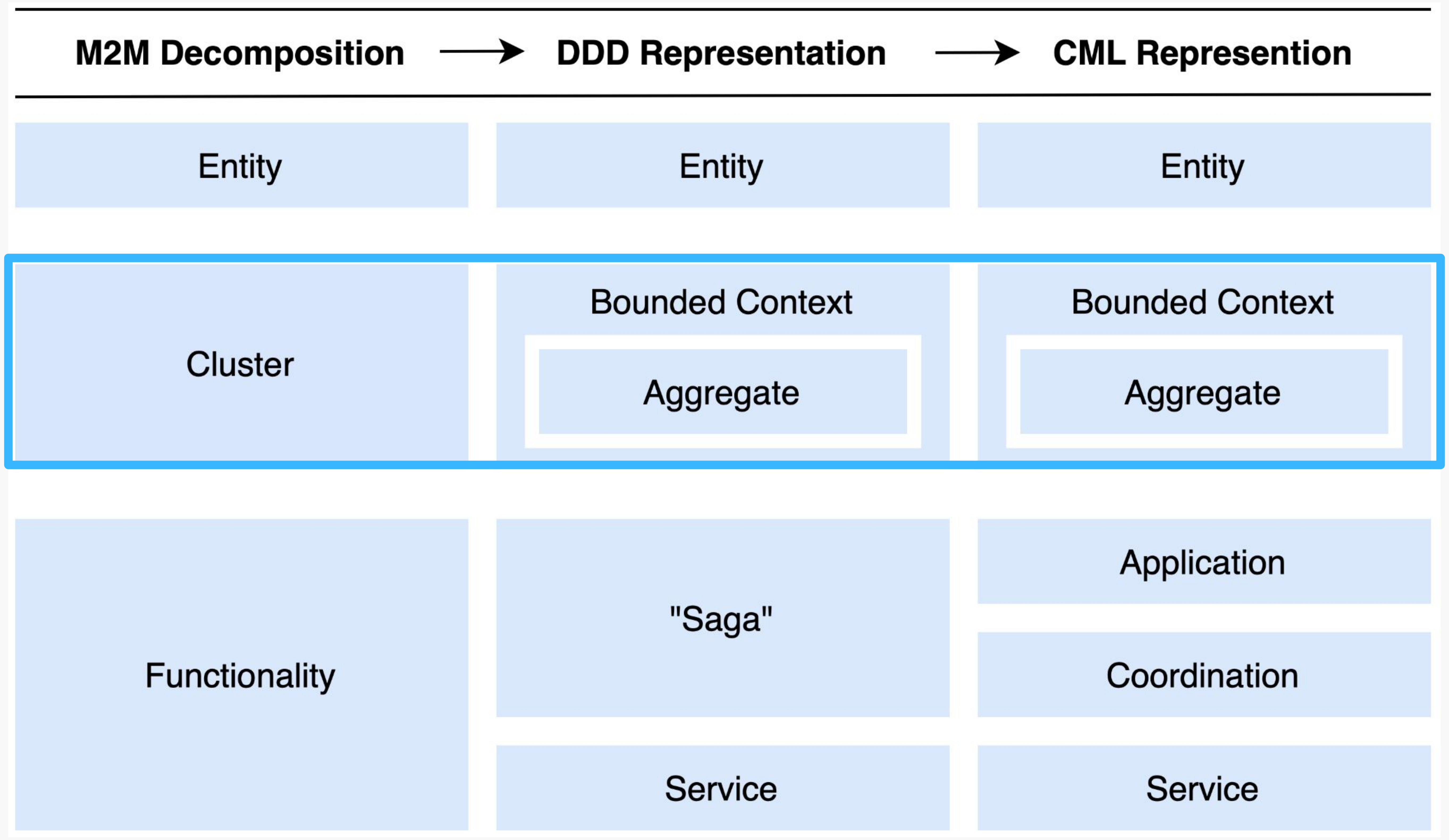
M2M Decomposition →	DDD Representation →	CML Represention
Entity	Entity	Entity
Cluster	Bounded Context: Aggregate	Bounded Context: Aggregate
Functionality	"Saga"	Application
		Coordination
	Service	Service

Mapping Strategy



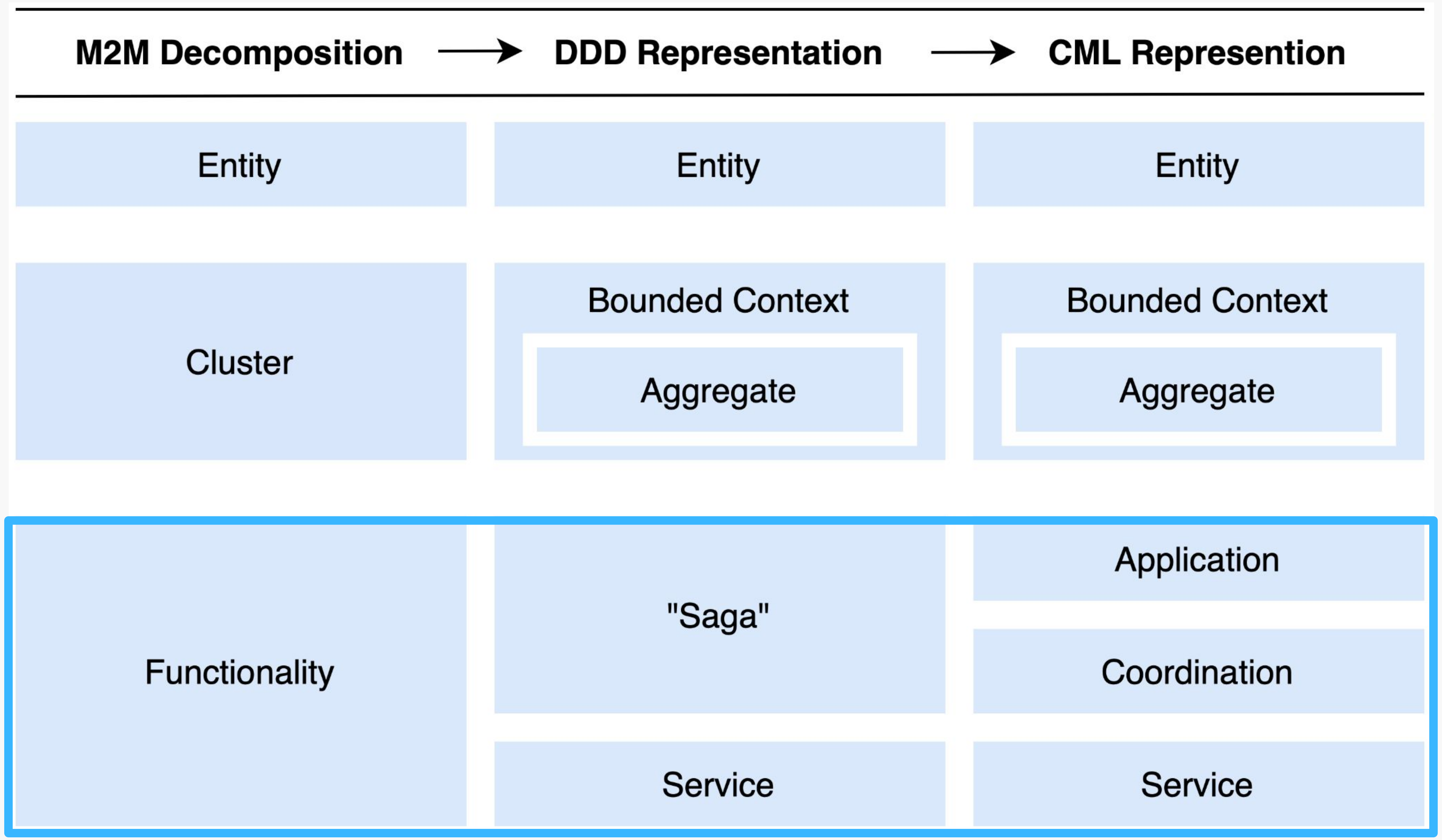
M2M Decomposition	DDD Representation	CML Represention
Entity	Entity	Entity
Cluster	Bounded Context Aggregate	Bounded Context Aggregate
Functionality	"Saga"	Application Coordination
	Service	Service

Mapping Strategy



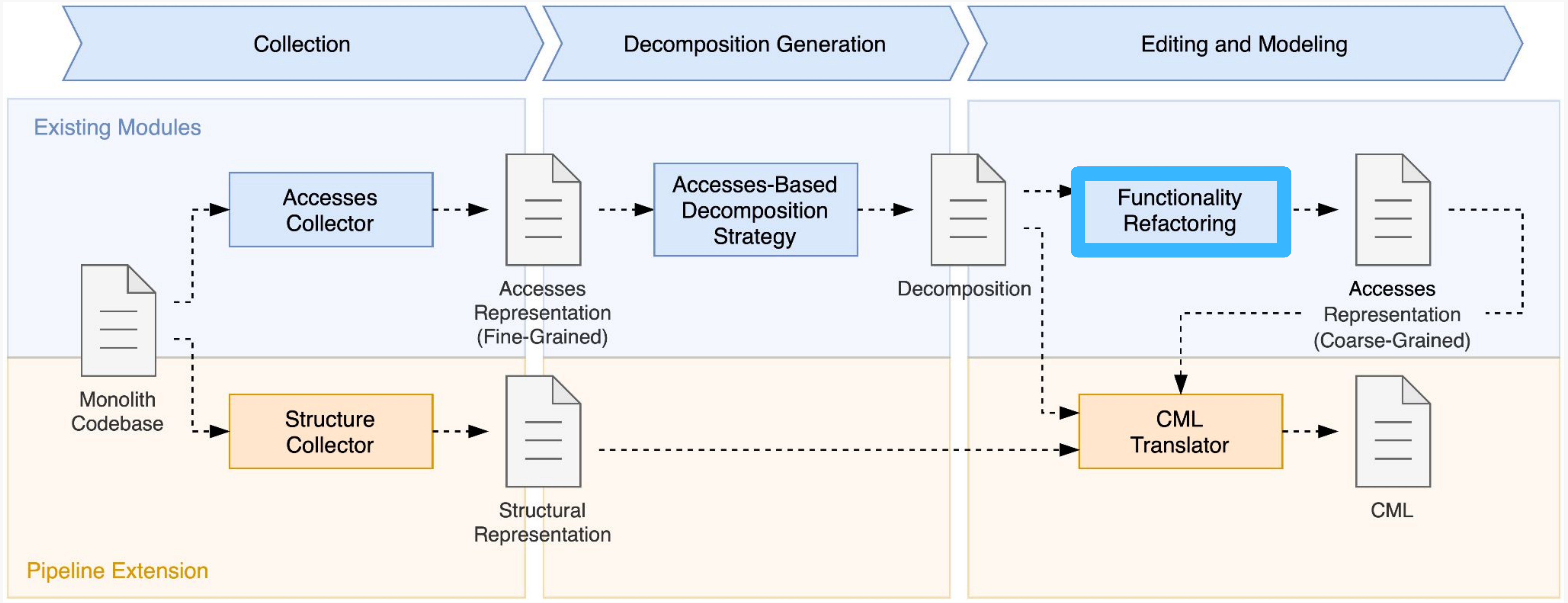
M2M Decomposition	→	DDD Representation	→	CML Represention
Entity		Entity		Entity
Cluster		Bounded Context / Aggregate		Bounded Context / Aggregate
Functionality		"Saga"		Application
				Coordination
		Service		Service

Mapping Strategy



M2M Decomposition →	DDD Representation →	CML Represention
Entity	Entity	Entity
Cluster	Bounded Context: Aggregate	Bounded Context: Aggregate
Functionality	"Saga"	Application
		Coordination
	Service	Service

Mono2Micro Pipeline Extension



IV Evaluation

Case Study with Quizzes-Tutor

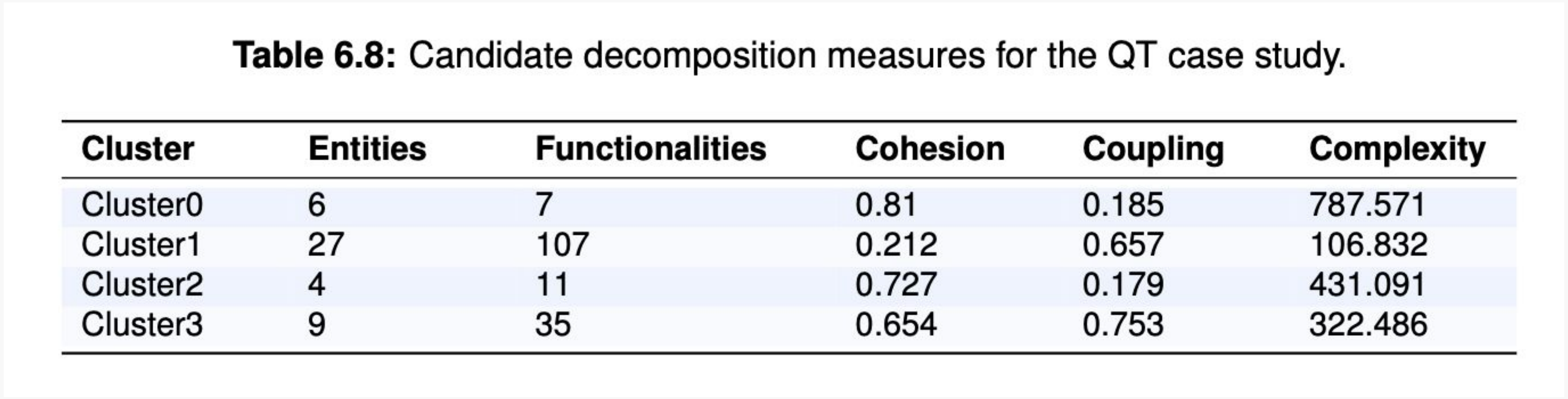
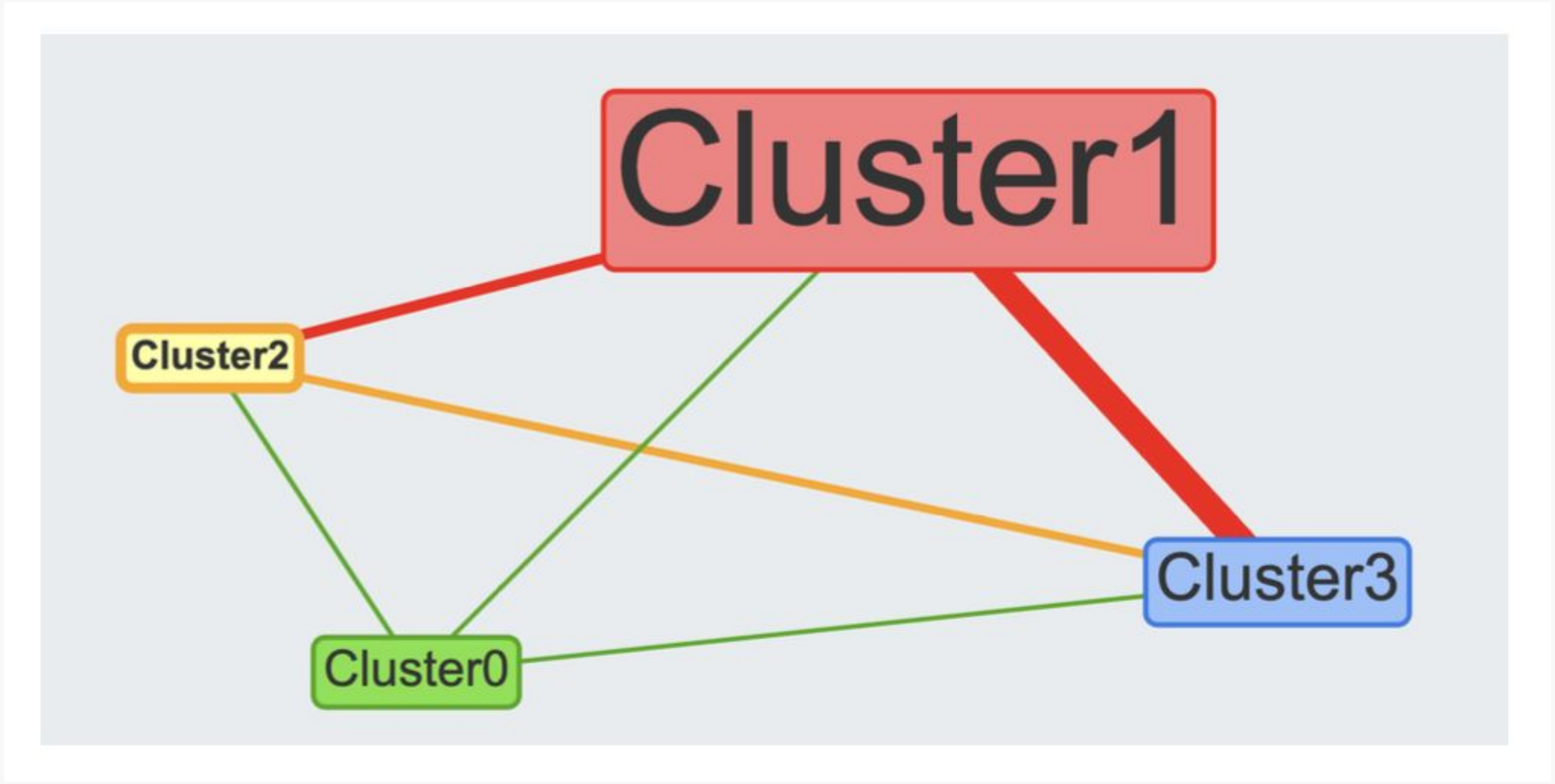


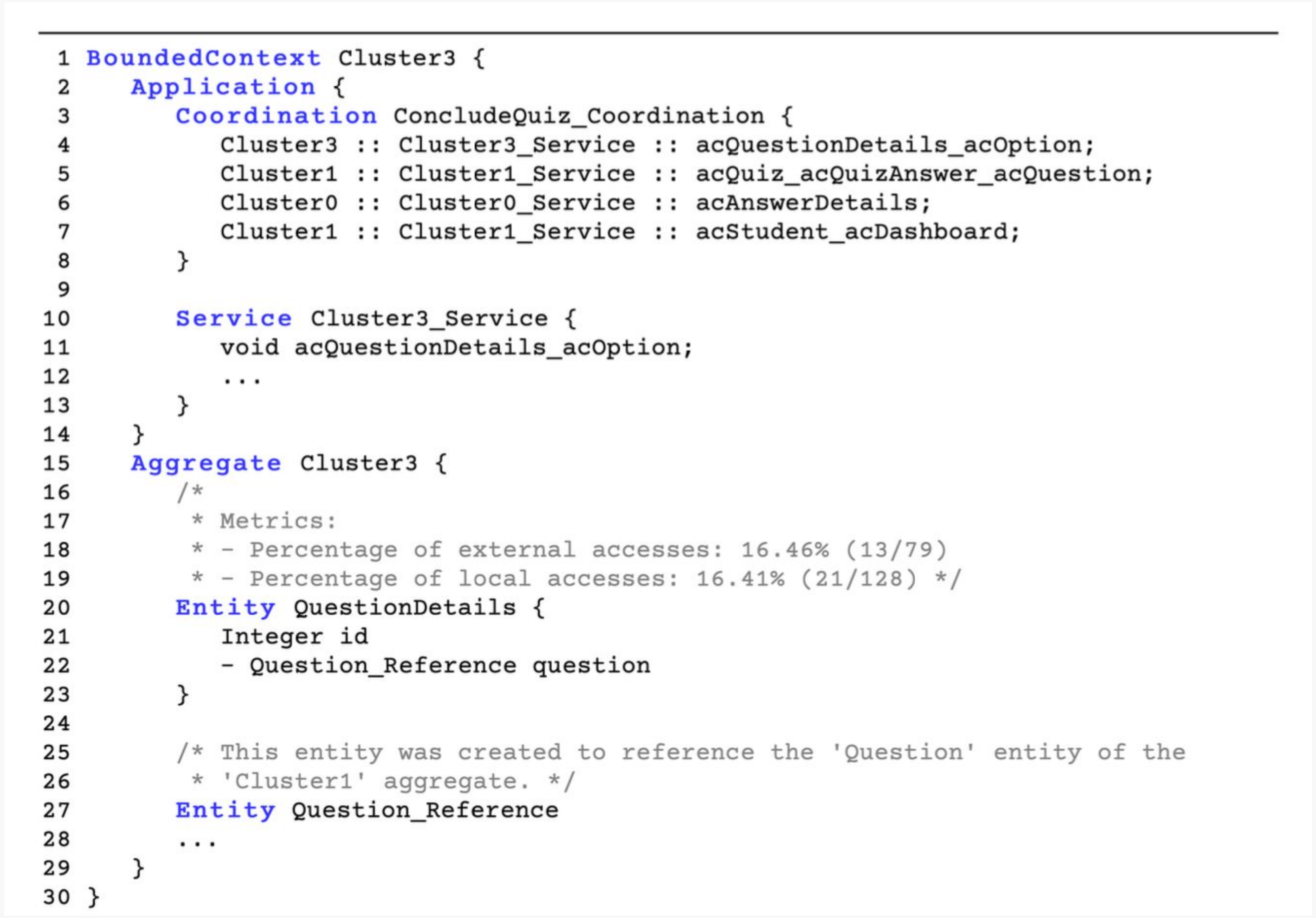
Table 6.8: Candidate decomposition measures for the QT case study.

Cluster	Entities	Functionalities	Cohesion	Coupling	Complexity
Cluster0	6	7	0.81	0.185	787.571
Cluster1	27	107	0.212	0.657	106.832
Cluster2	4	11	0.727	0.179	431.091
Cluster3	9	35	0.654	0.753	322.486

Case Study with Quizzes-Tutor



Case Study with Quizzes-Tutor



```
BoundedContext Cluster3 {
   Application {
      Coordination ConcludeQuiz_Coordination {
         Cluster3 :: Cluster3_Service :: acQuestionDetails_acOption;
         Cluster1 :: Cluster1_Service :: acQuiz_acQuizAnswer_acQuestion;
         Cluster0 :: Cluster0_Service :: acAnswerDetails;
         Cluster1 :: Cluster1_Service :: acStudent_acDashboard;
      }

      Service Cluster3_Service {
         void acQuestionDetails_acOption;
         ...
      }
   }
   Aggregate Cluster3 {
      /*
       * Metrics:
       * - Percentage of external accesses: 16.46% (13/79)
       * - Percentage of local accesses: 16.41% (21/128) */
      Entity QuestionDetails {
         Integer id
         - Question_Reference question
      }

      /* This entity was created to reference the 'Question' entity of the
       * 'Cluster1' aggregate. */
      Entity Question_Reference
      ...
   }
}
```

Answers to Research Questions

RQ1: By following and respecting both tools models, and utilizing the Discovery Library module as the integration point.

RQ2: By defining a decomposition in Mono2Micro and establishing a mapping between the concepts to DDD patterns in CML.

RQ3: An architect benefits by having a complete semi-automatic pipeline to model decompositions in a DDD environment.

V Conclusions

Conclusions

- Almost no migration tools incorporate **DDD editing**.
- **Integration** of Mono2Micro and Context Mapper as a solution.
- Defining a **mapping of concepts** between tools so that DDD can be used.

Contributions

- A monolith decomposition tool based on **DDD modeling**
- A new **data collector** on the side of Mono2Micro
- A new **contract** between Mono2Micro and Context Mapper
- New Mono2Micro decomposition **discovery strategies**
- New **syntax rules** in CML on the side of Context Mapper

Questions & Discussion

References

- Lopes, T.D., Silva, A.R.: Monolith Microservices Identification: An Extensible Multiple Strategy Tool Examination Committee. Master’s thesis, Instituto Superior Técnico, University of Lisbon (2022)
- Nunes, L., Santos, N., Rito Silva, A.: From a monolith to a microservices architecture: An approach based on transactional contexts. In: Software Architecture: 13th European Conference, ECSA 2019, Paris, France, September 9–13, 2019, Proceedings. pp. 37–52 (2019). https://doi.org/10.1007/978-3-030-29983-5 3
- Correia, J., Rito Silva, A.: Identification of monolith functionality refactorings for microservices migration. Software: Practice and Experience 52(12), 2664–2683 (2022). https://doi.org/10.1002/spe.3141
- S. Kapferer. and O. Zimmermann., “Domain-specific language and tools for strategic domain-driven design, context mapping and bounded context modeling,” in *Proceedings of the 8th International Conference on Model-Driven Engineering and Software Development - MODELSWARD,*, INSTICC. SciTePress, 2020, pp. 299–306
- Kapferer, S., Zimmermann, O.: Domain-driven service design. In: Dustdar, S. (ed.) Service-Oriented Computing. pp. 189–208. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-64846-6 11
- Shaw, M.: Writing good software engineering research papers. In: 25th International Conference on Software Engineering, 2003. Proceedings. pp. 726–736 (2003). https://doi.org/10.1109/ICSE.2003.1201262