

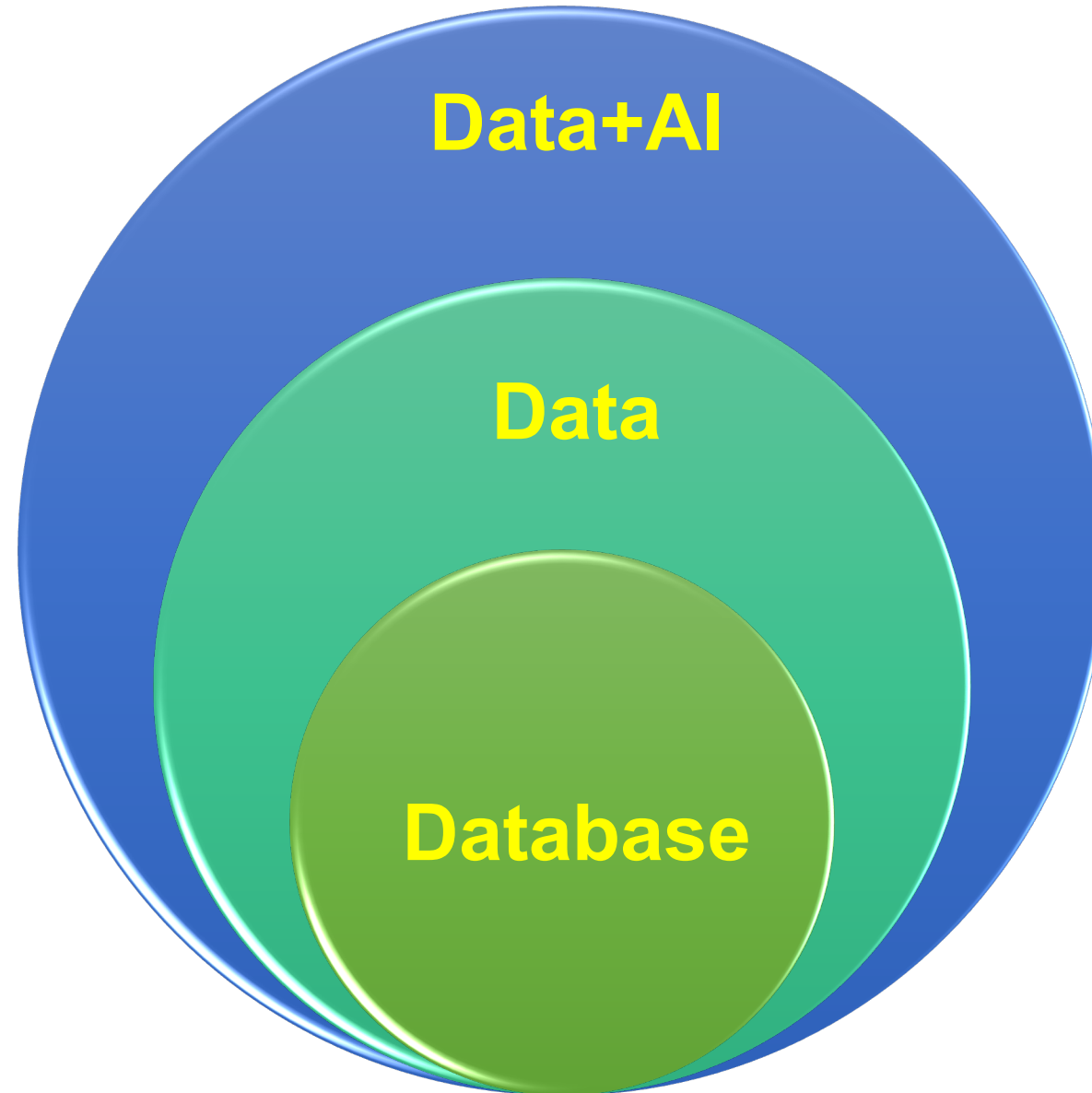


Data Agent: A Holistic Architecture for Orchestrating **Data**+**A**I Ecosystems

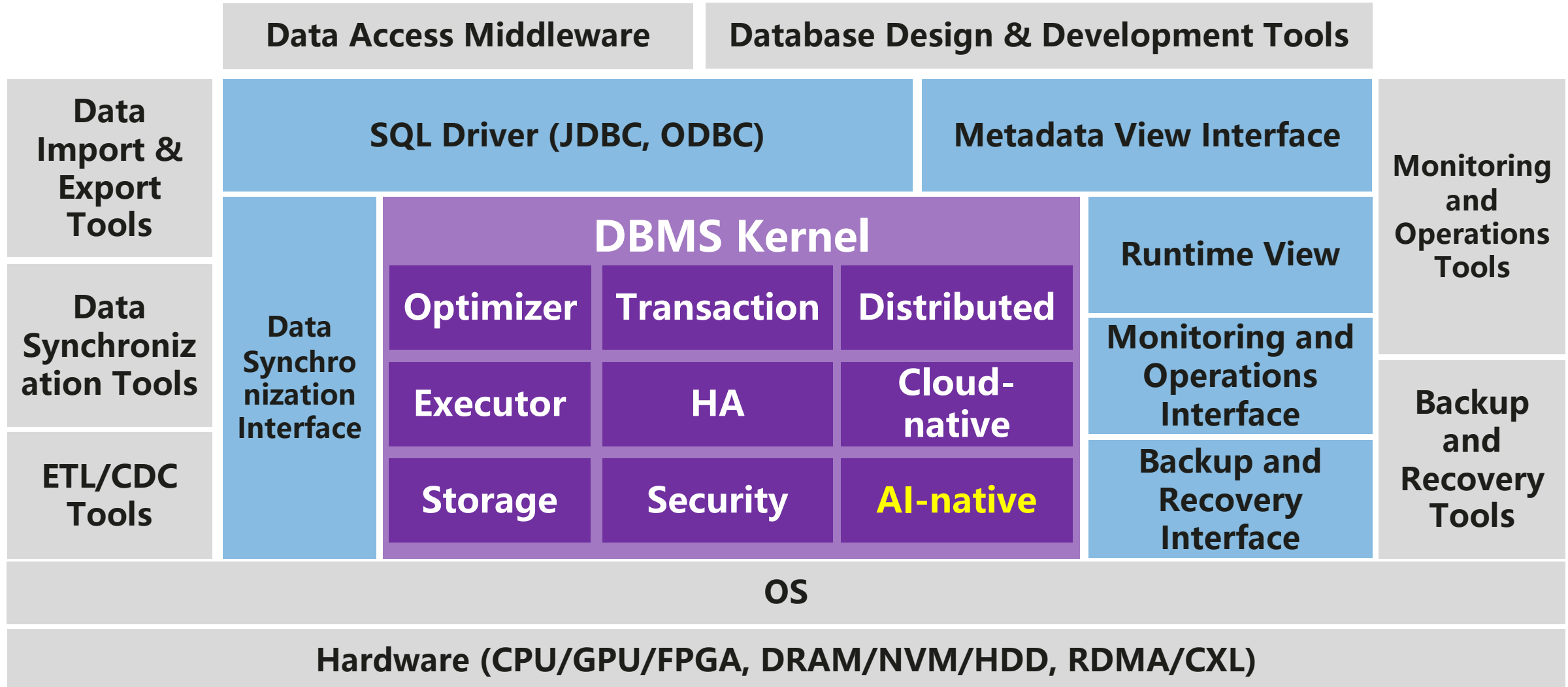
Guoliang Li (ACM/IEEE Fellow)

Department of Computer Science, Tsinghua University

Database → Data Systems → Data + AI

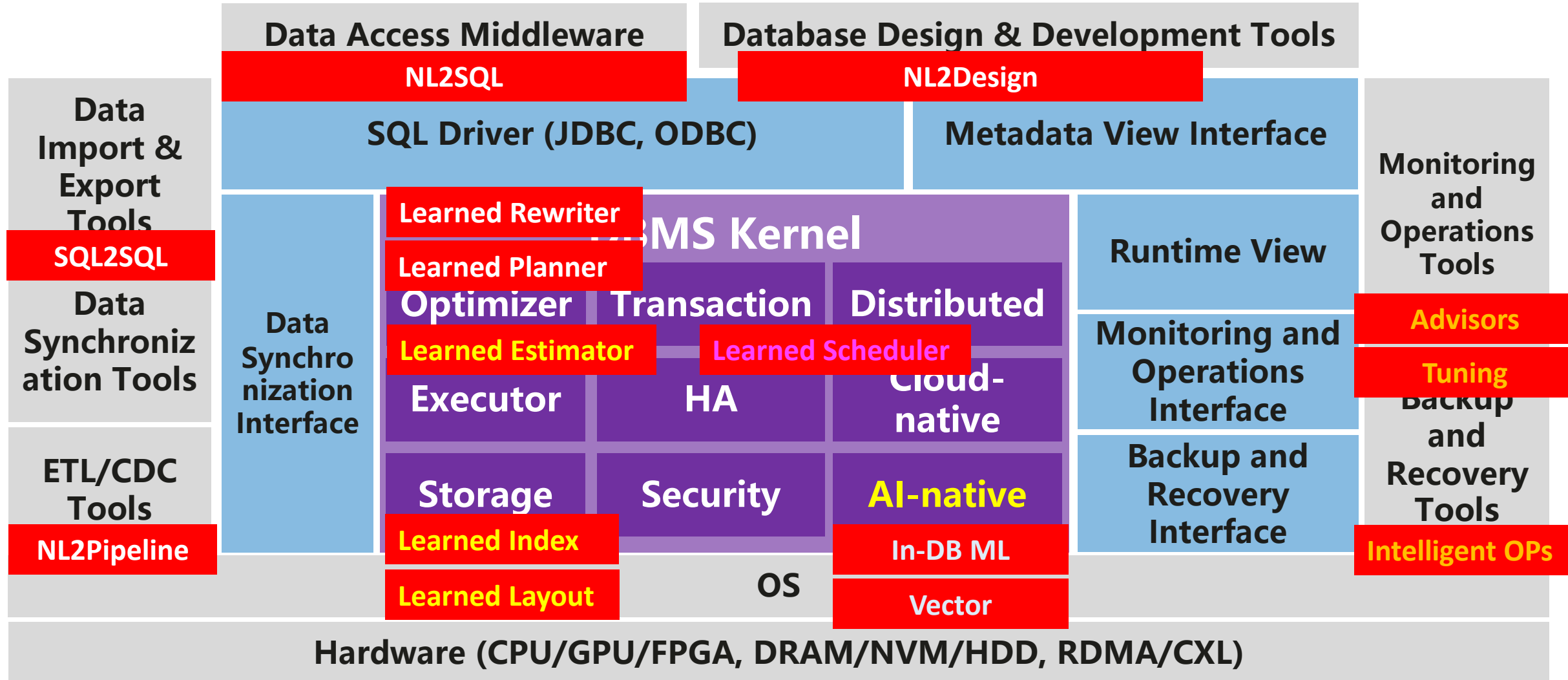


Database Ecosystem



Database Ecosystem

① Regression ② Planning (NP) ③ Prediction ④ Understanding ⑤ Embedded AI



What is an AI-Native Database?

AI-native databases seamlessly integrate AI techniques to automate data management, query processing, and resource scheduling.

① Embedded AI Capabilities

- Regression, planning, prediction
- NLP, understanding, semantics, reasoning

② Automated Optimization

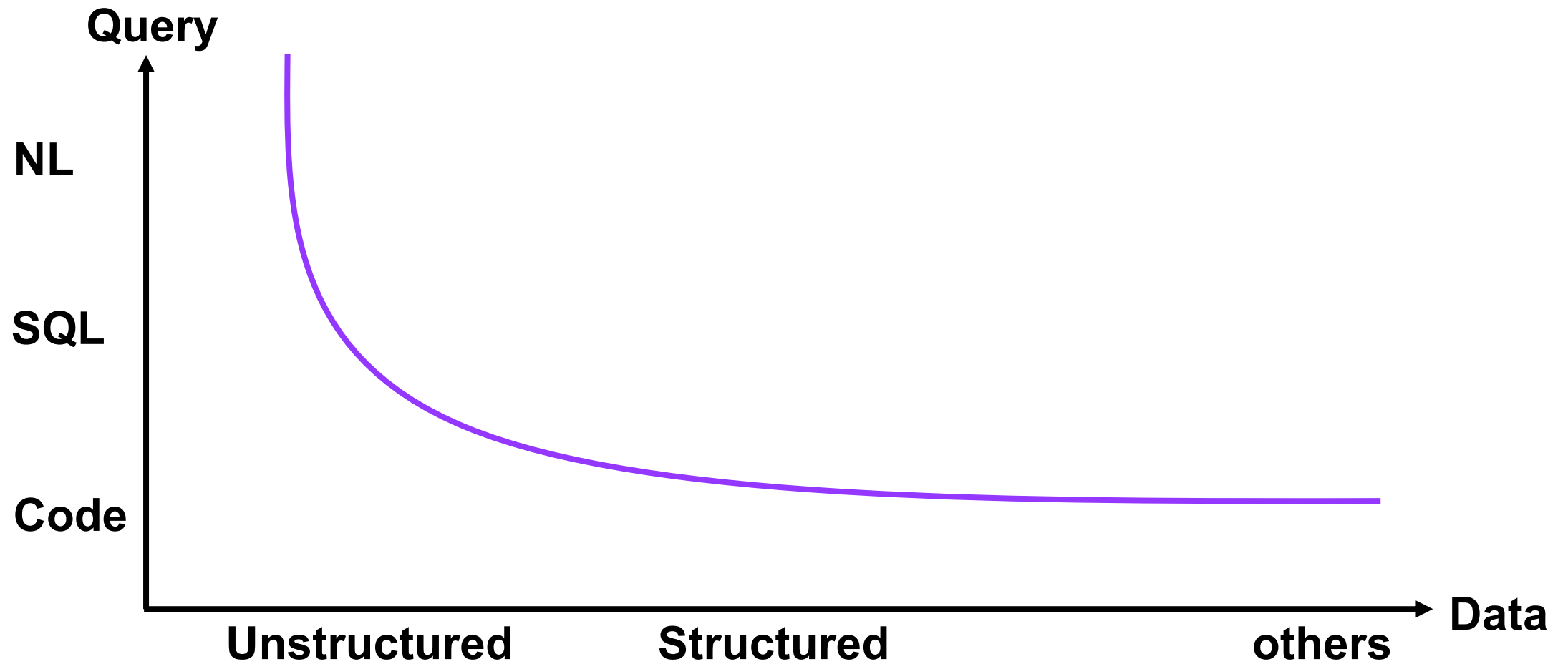
- Predictor, estimator, planner, advisor, scheduler

③ Adaptive Learning

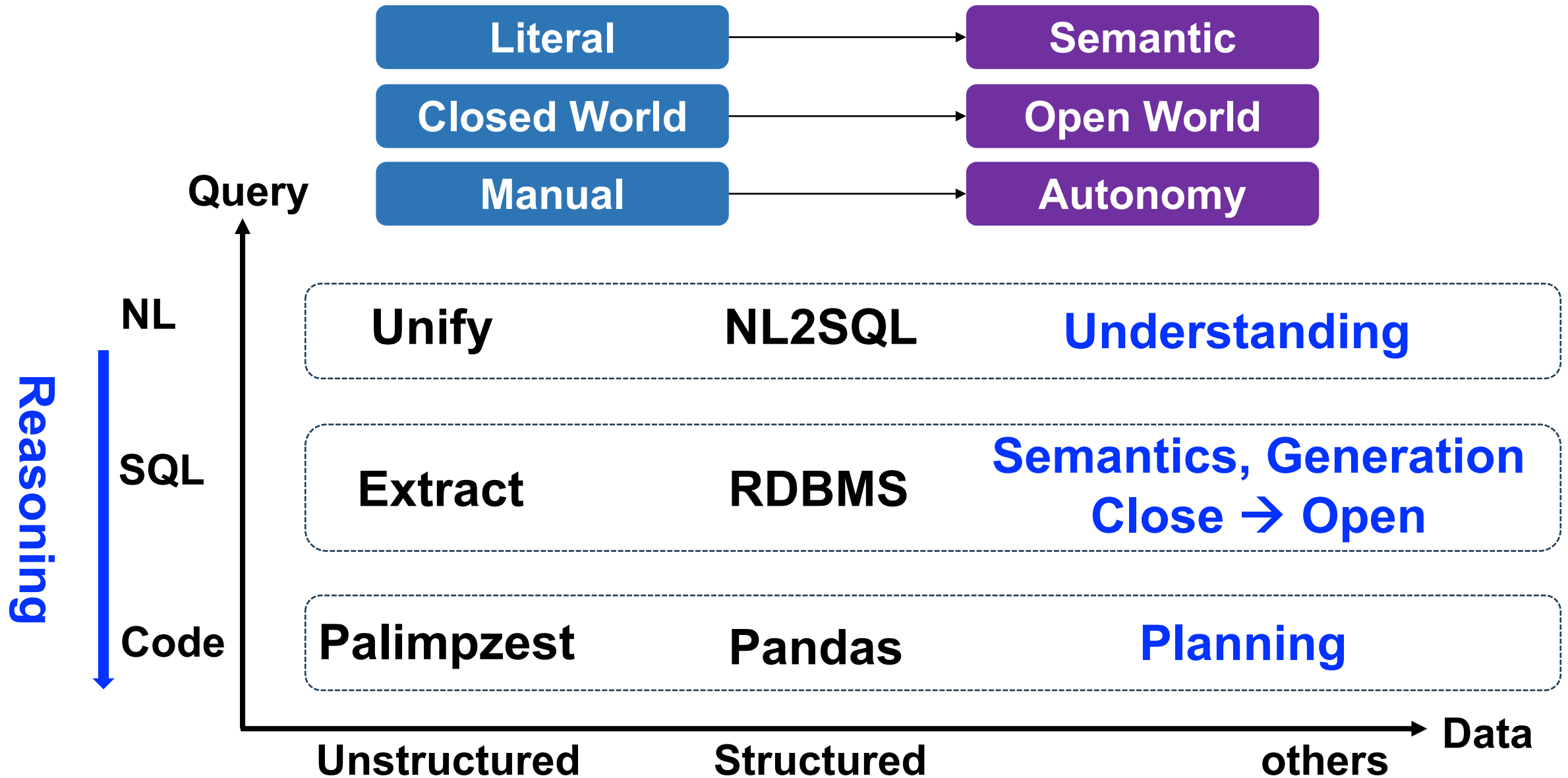
- Adapt to data, query, resource changes
- Self-reflection



Database → Data Systems



Database → Data Systems



Data Agent (AI-Native Data System)

Autonomously executes data tasks without human intervention, creating an E2E autonomous loop from raw data to business actions.

- Connotation

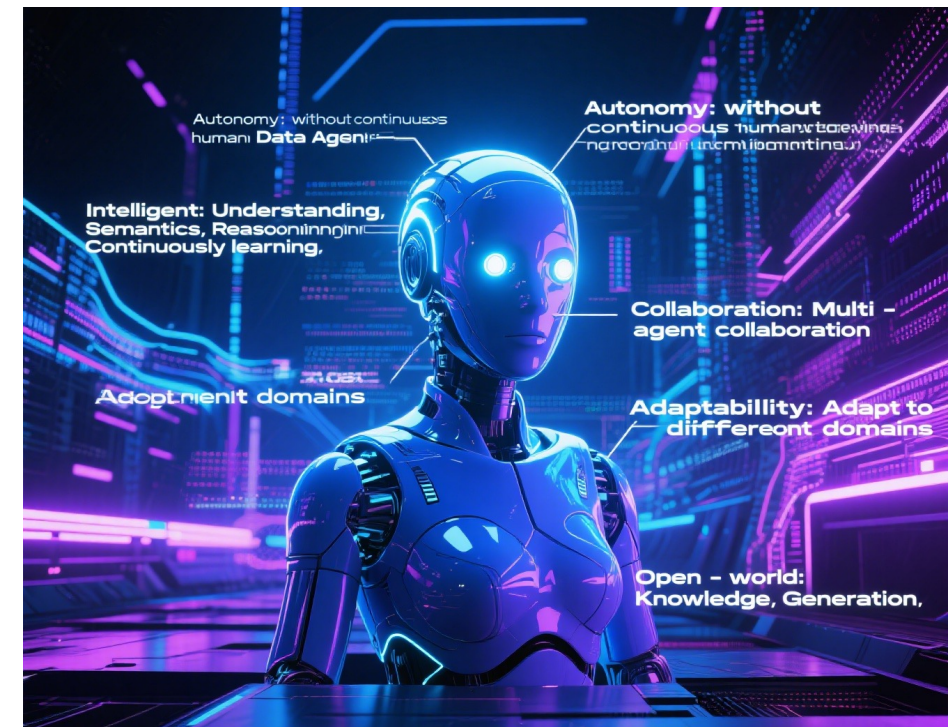
- Autonomy: without continuous human intervention

- Perception: Task & Environment Understanding
 - Orchestration: Task Decomposition
 - Reasoning & Planning: Optimization & Execution
 - Memory: Perception, Understanding, Semantics, Context
 - Self-reflection: Feedback
 - Multi-Agent Collaboration

- Continuous Learning: from new data & behavior

- Dynamic Adaptability: Adapt to different domains

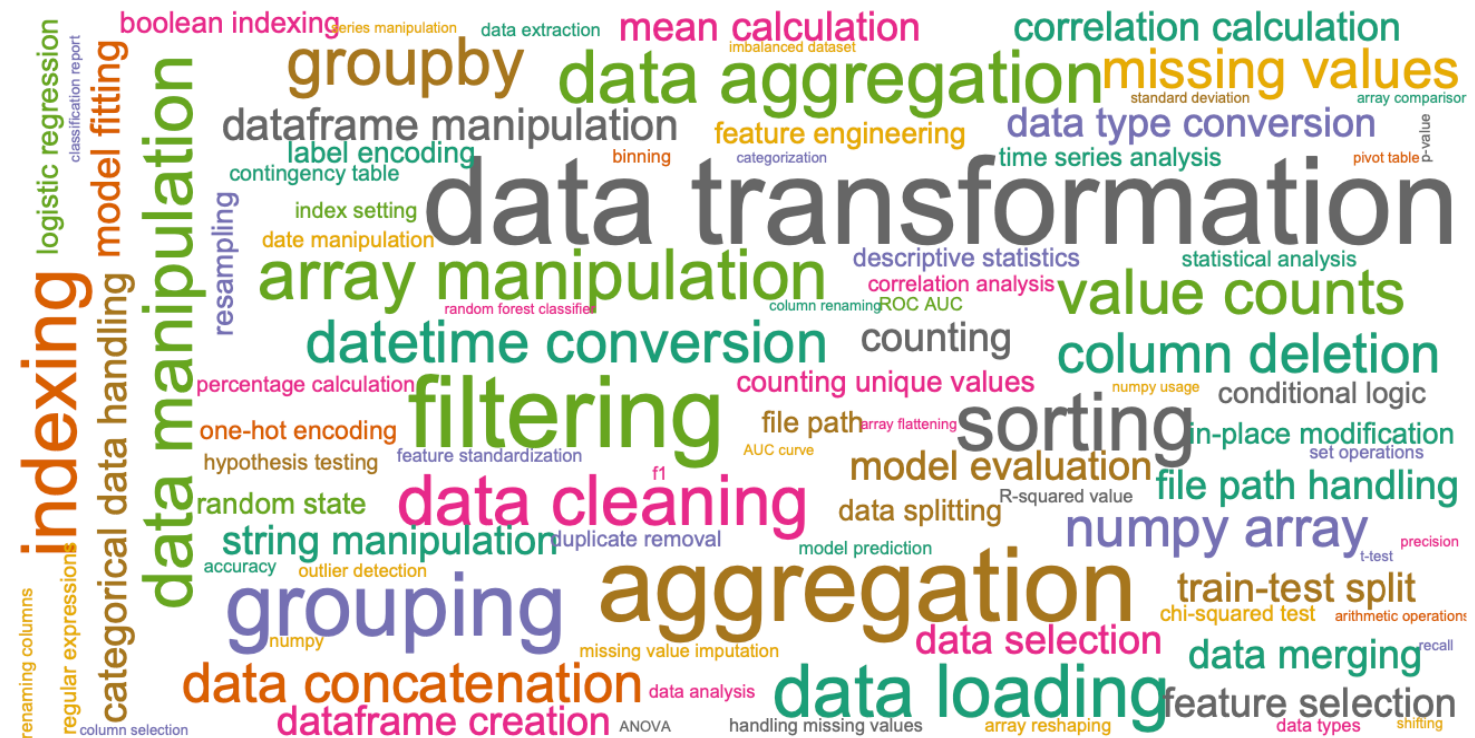
- Proactivity: Predict the future and take initiative



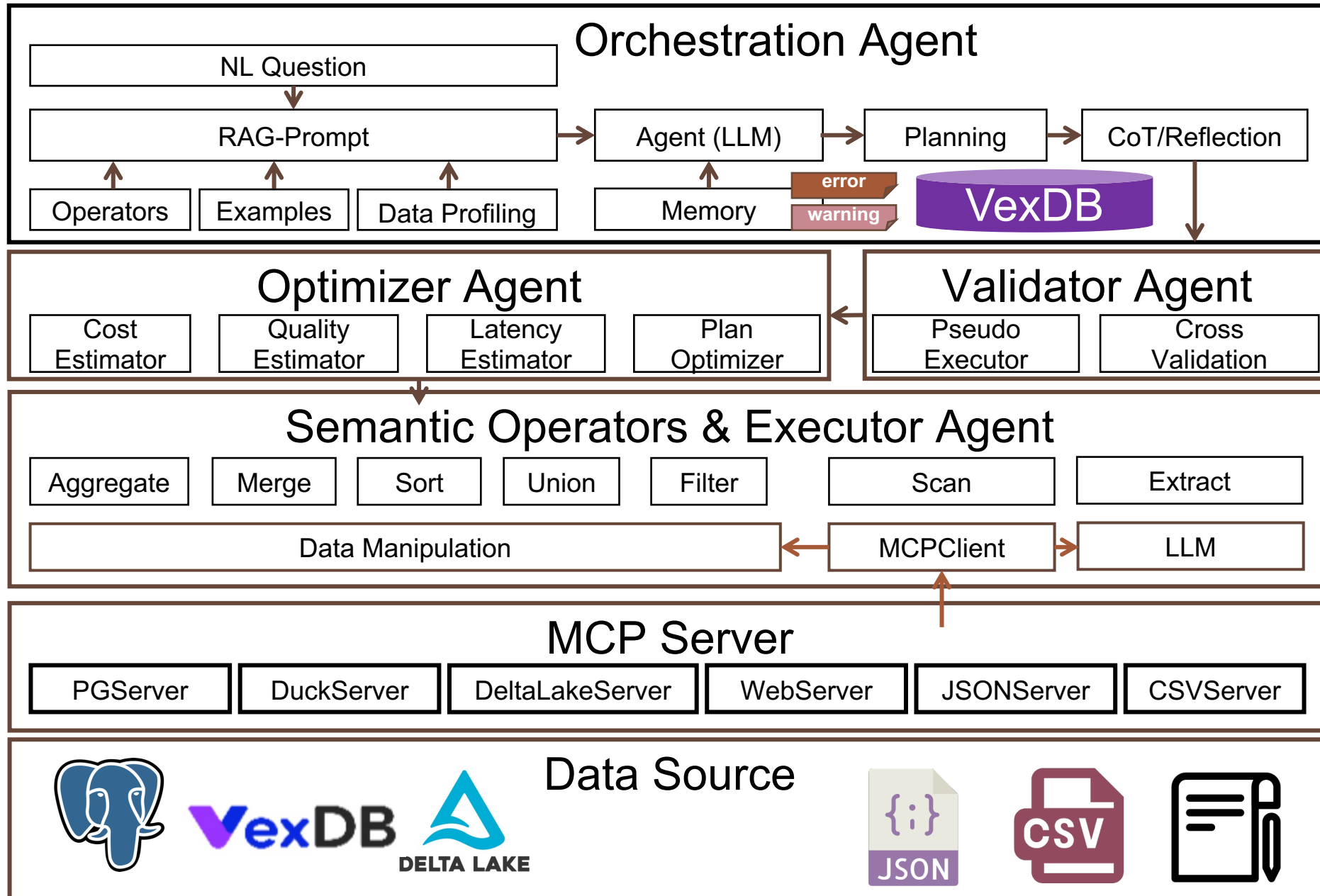
Data Agent (AI-Native Data System)

Autonomously executes data tasks without human intervention, creating an E2E autonomous loop from raw data to business actions.

- [illegible]

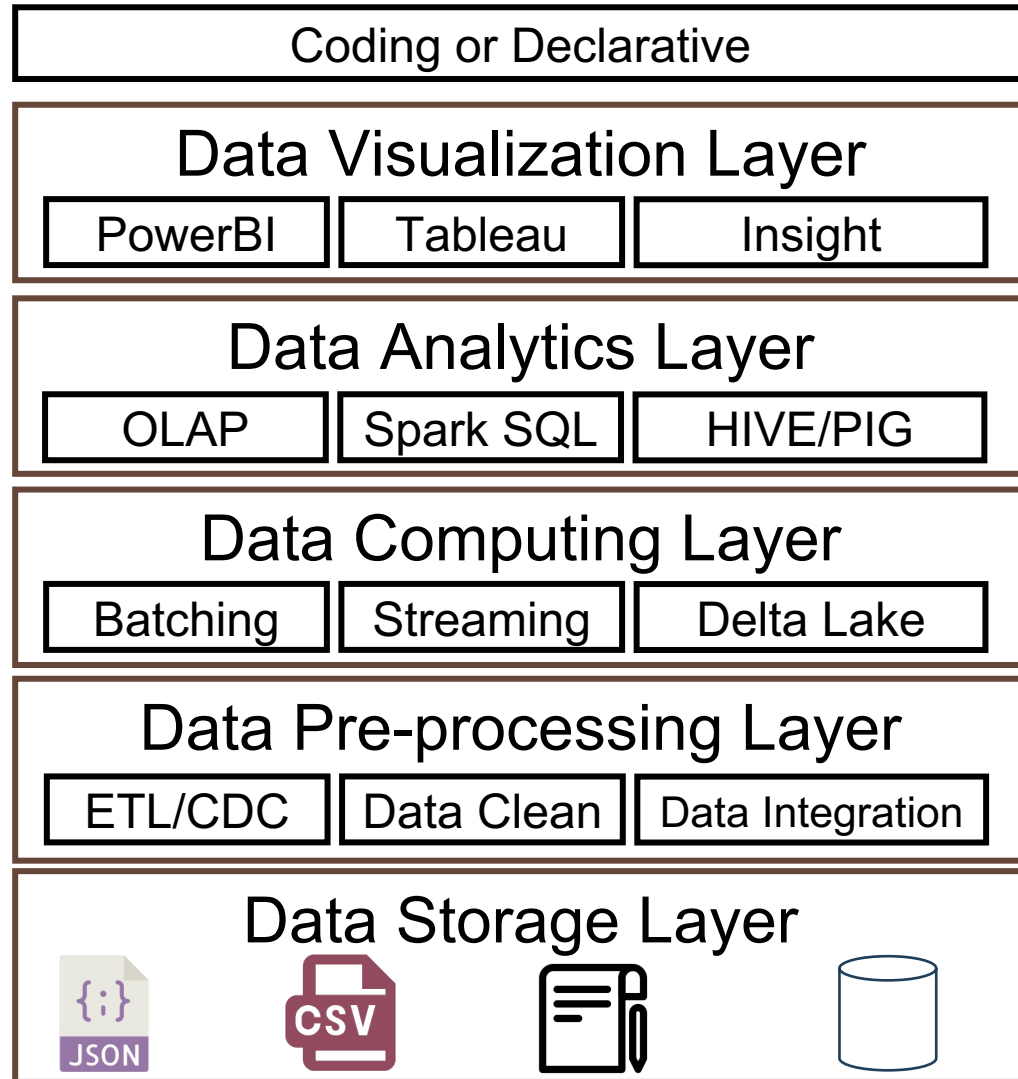


Data Agent: Architecture

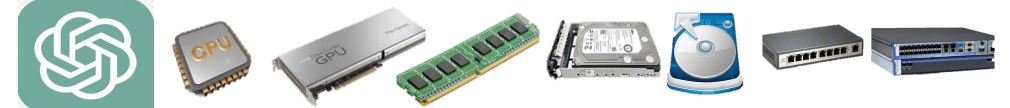
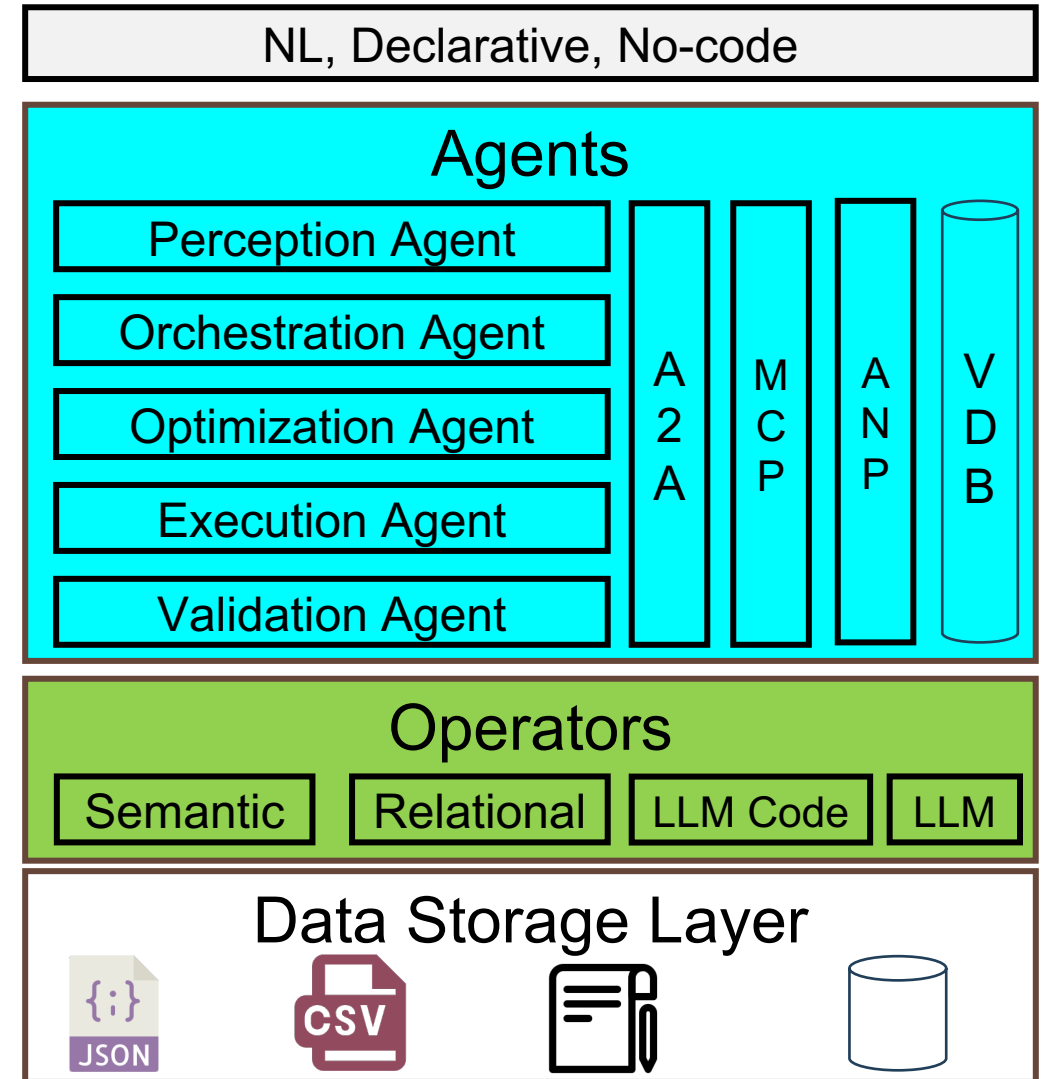


Next-Generation Data Analytics

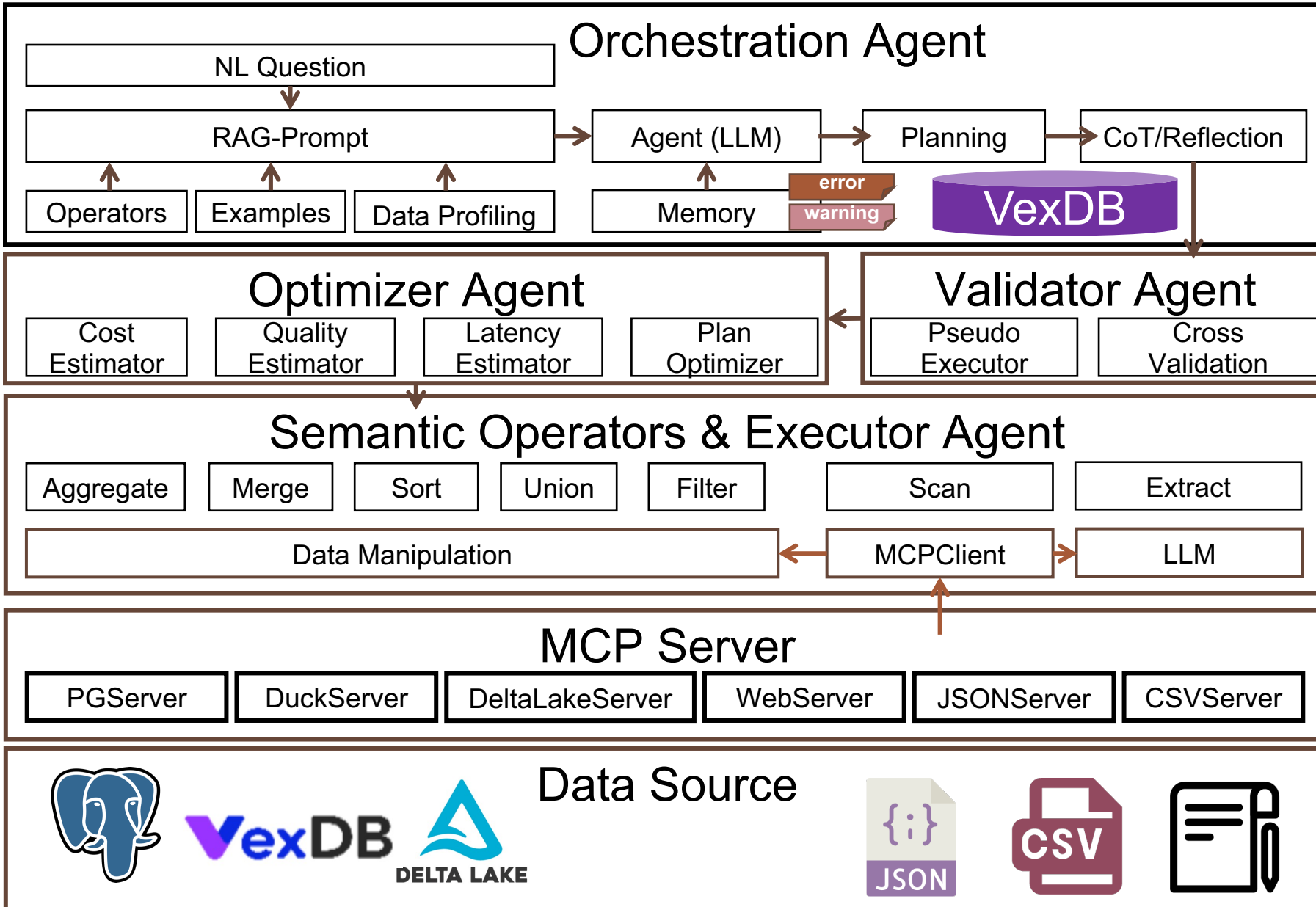
AS IS



TO BE



Data Agent: Challenges



②: How to **orchestrate** an effective/efficient pipeline?

③: How to effectively **design the memory**?

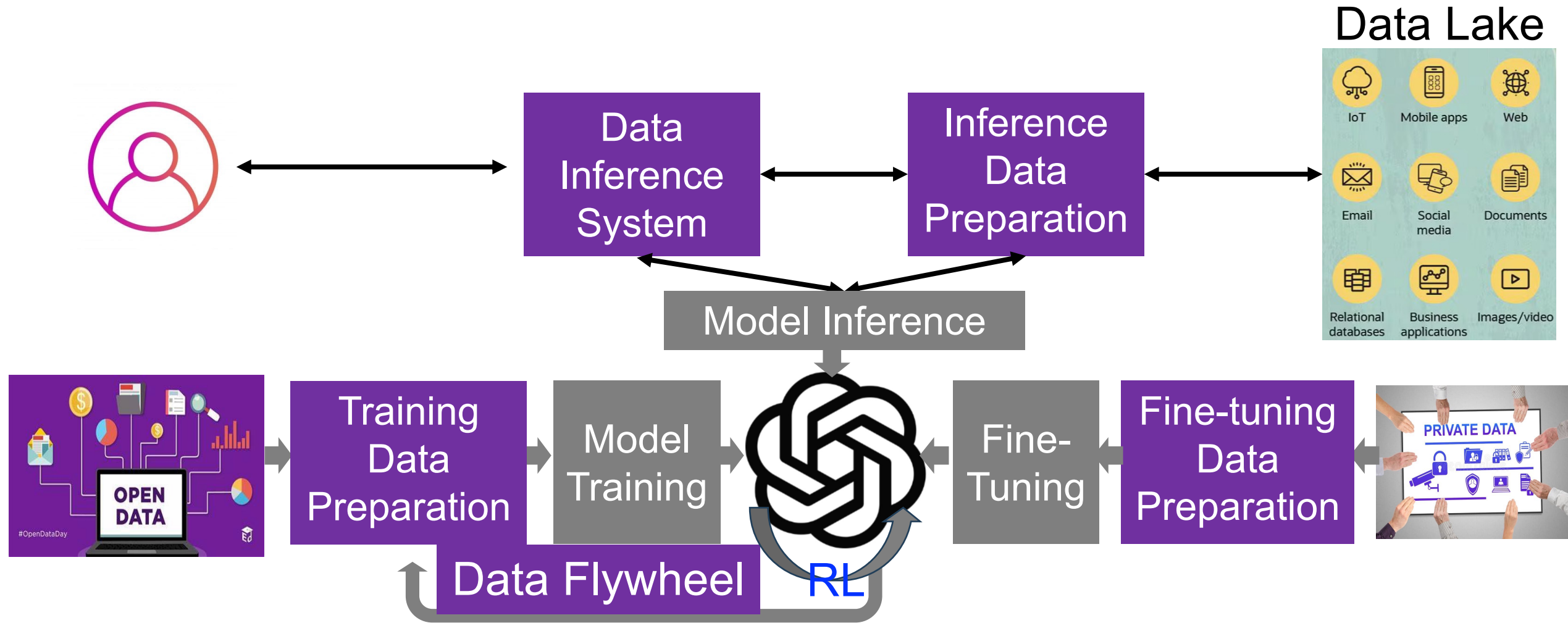
④: How to **optimize & execute** a pipeline?

⑤: How to **continuously enhance** pipeline quality?

⑥: How to **schedule and interact** agents and tools?

①: How to **understand** queries, data, agents, tools?

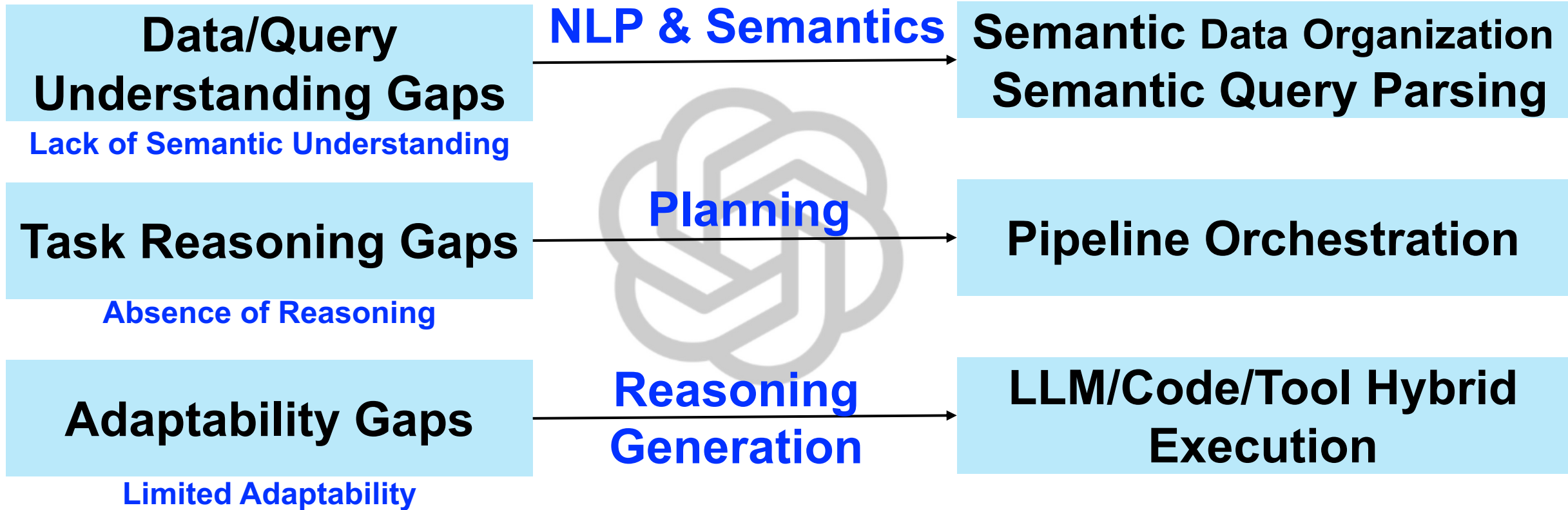
Data → Data + AI (Lifecycle)



Rely on labor-intensive coding, hard for Adaptability 😞
Manual Intervention → Autonomy

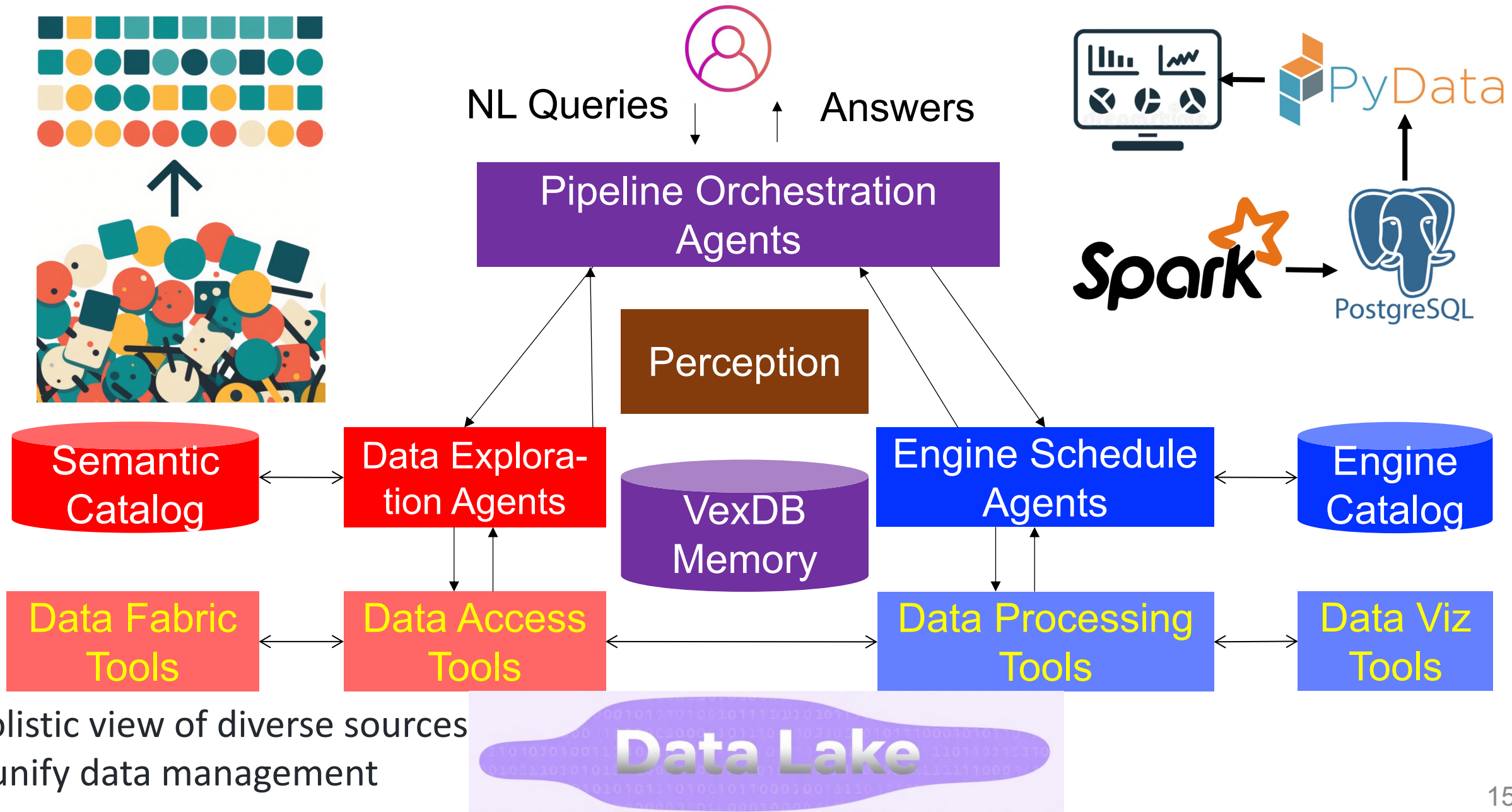
Data + AI Motivation

How to Bridge the Gaps?

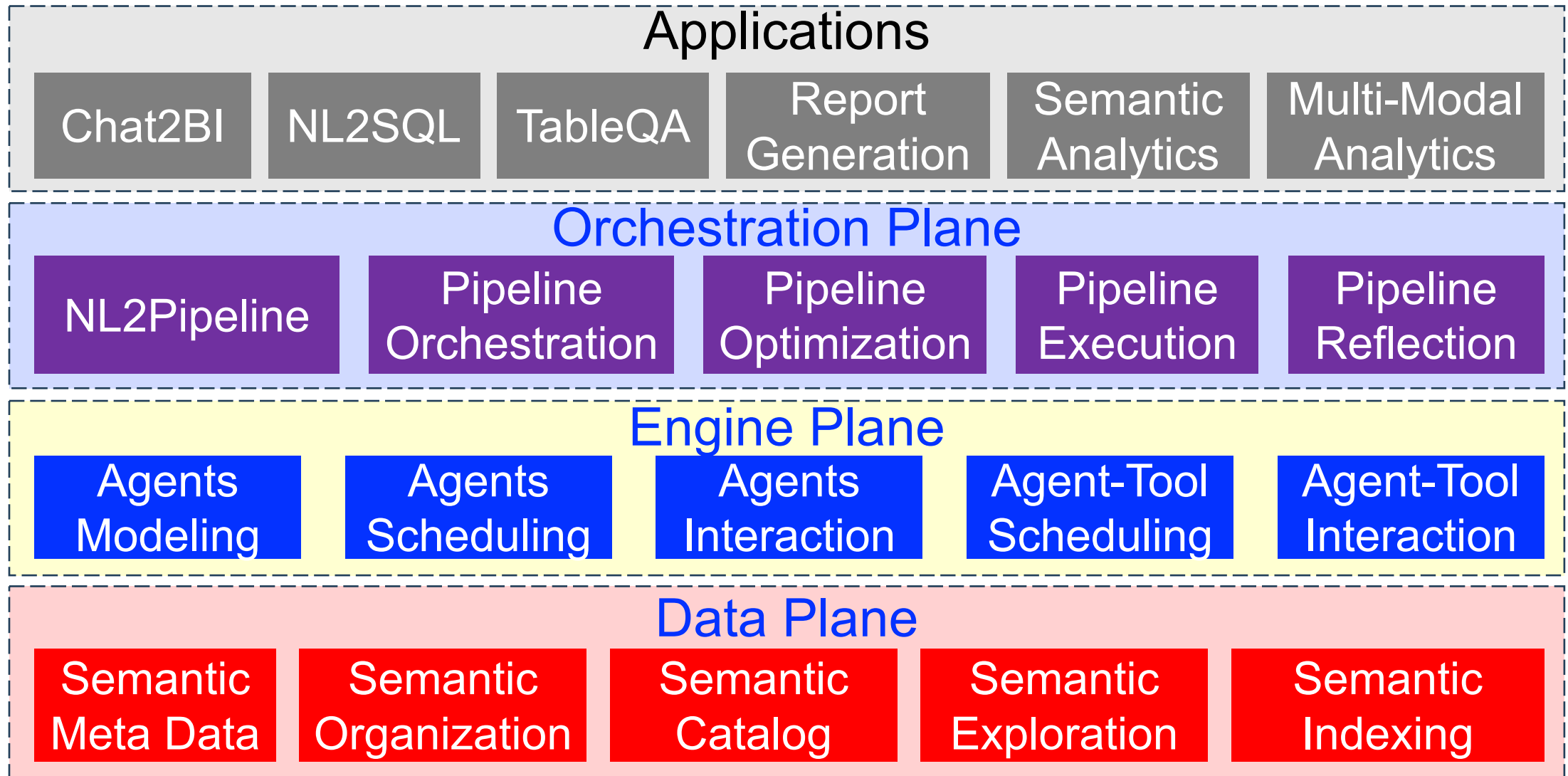


An autonomous system is crucial for Data+AI applications.

Data Agent: Framework



Data Agent: Modules

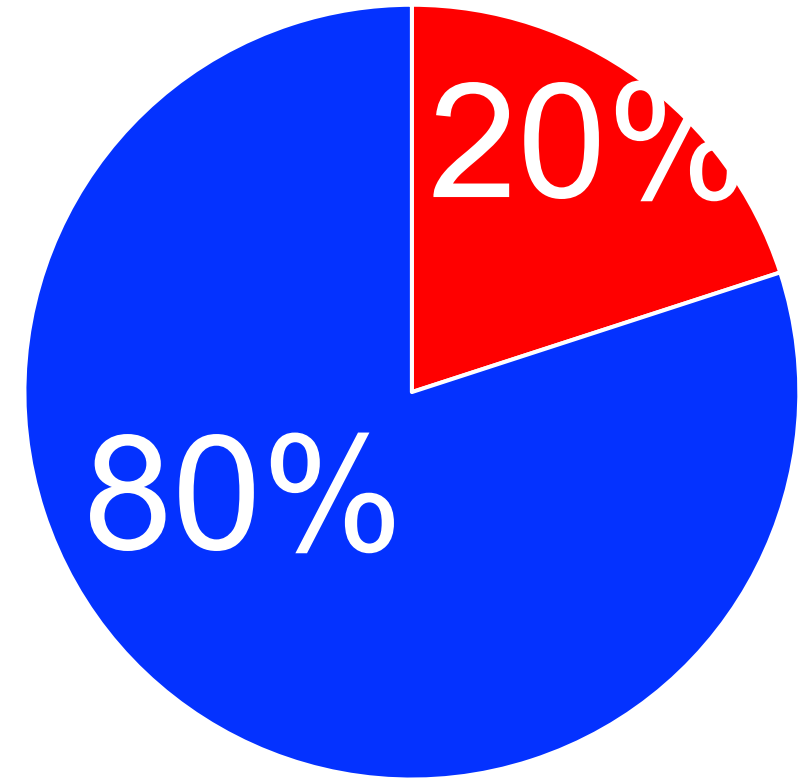




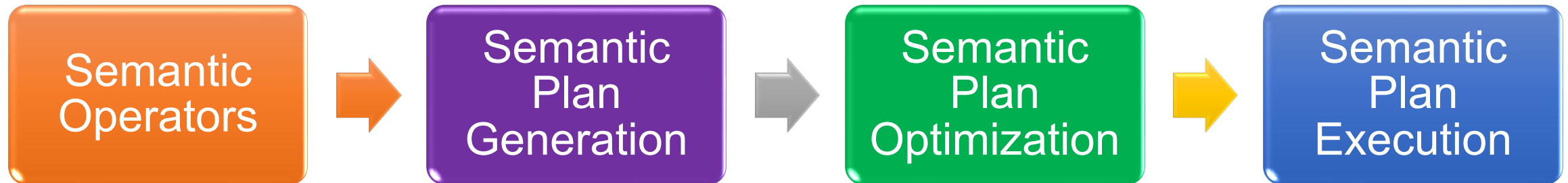
Data Agent for Unstructured Data Analytics

Motivation of Unstructured Data Analytics

- Hard to Analyze Unstructured Data!
 - How many VLDB25 papers are related to Data+AI?
- Challenges
 - Query Understanding
 - Data Organization
 - Semantic Query Processing
- Autonomous Unstructured Data Analytics Pipeline
Orchestration & Optimization



■ Structured ■ Unstructured



Semantic Data Analytics Example

Count the number of movies directed by Steven Spielberg that the number of positive reports is larger than the number of negative ones by their report sentiments.

- Step 1. **Semantic Extract** movies directed by Steven Spielberg
- Step 2. **Semantic Filter** the reports for the extracted movies
- Step 3. **Semantic Group** the reports by filtered movies
- Step 4. **Semantic Classify** the reports as Negative or Positive
- Step 5. **Count** the Positive reports and Negative reports for each group
- Step 6. **Compare** #Positive reports and #Negative reports
- Step 7. **Count** the number of groups with more #Positive reports

Unstructured Data Analytics Agent: NL2Pipeline

NL Query

Count the number of movies directed by Steven Spielberg that the number of positive reports is larger than the number of negative ones by their report sentiments.

Semantic
Extract

Semantic
Filter

Semantic
Group

Count

Semantic
Classify

Count

Compare

Count

Unstructured Data Analytics Agent: NL2Pipeline

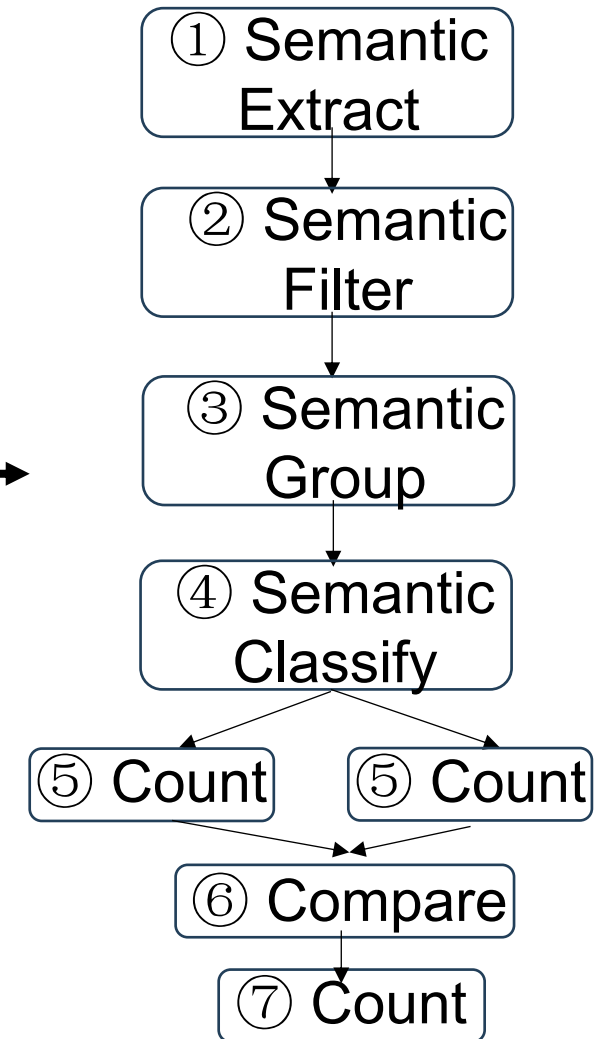
NL Query

Count the number of movies directed by Steven Spielberg that the number of positive reports is larger than the number of negative ones by their report sentiments.

LLM Capabilities



- NL Understanding
- Semantic Matching
- Reasoning
- Generation



Unstructured Data Analytics Agent: NL2Pipeline

Semantic Operators



Semantic Plan Generation



Semantic Plan Optimization



Semantic Plan Execution

NL Query

Count the number of movies directed by Steven Spielberg that the number of positive reports is larger than the number of negative ones by their report sentiments.

Challenges



- Semantic Operators
- Pipeline Orchestration
- Pipeline Optimization
- Pipeline Execution

① Semantic Extract

② Semantic Filter

③ Semantic Group

④ Semantic Classify

⑤ Count

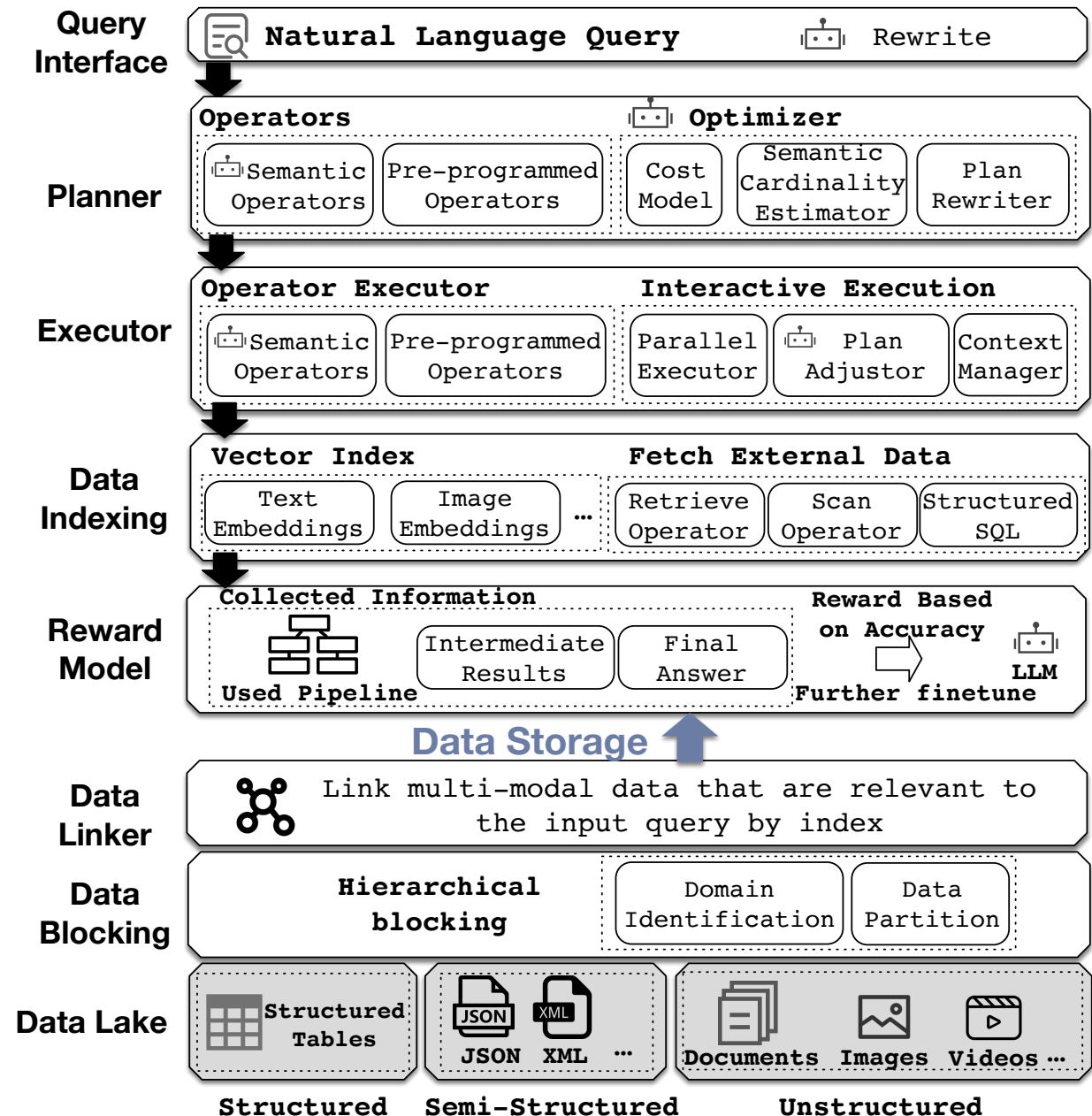
⑤ Count

⑥ Compare

⑦ Count

Unstructured Data Analytics Agent : Architecture

- Semantic Query: NL
- Semantic Operators:
 - LLM Semantic Operators
 - Programmed Operators
- Semantic Planning:
 - Pipeline orchestration
- Semantic Optimization & Execution
 - Multi-goal optimization
 - Interactive execution
- Self-reflection
 - COT, Reward model
- Semantic Indexing
 - Vector indexes



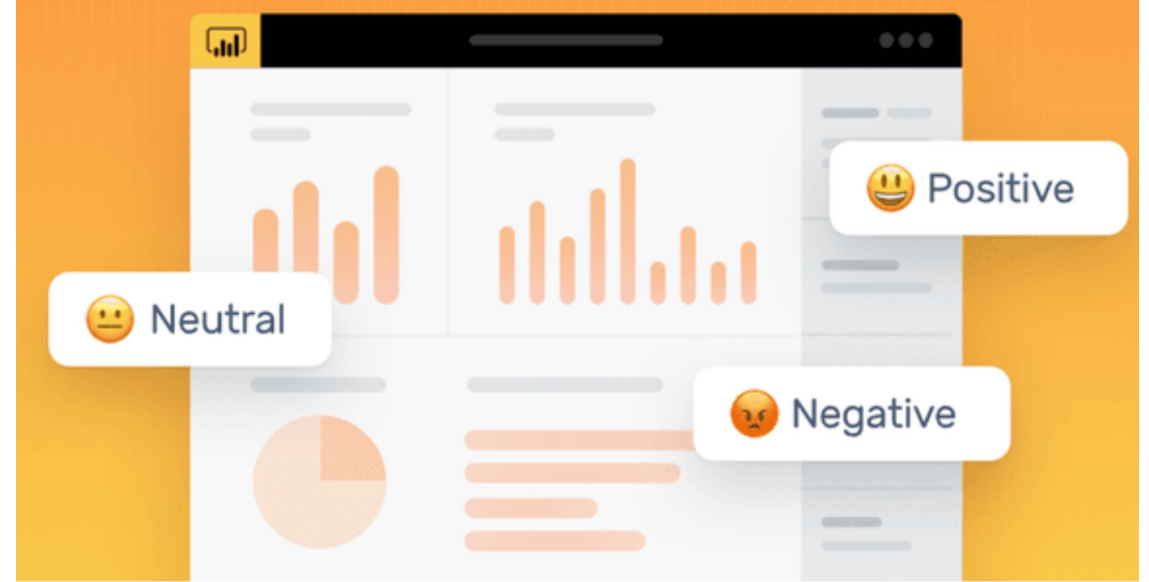
Unstructured Data Analytics Agent: Semantic Operators

- Extension from DB Operators

- Semantic filter
- Semantic projection
- Semantic join
- Semantic group-by
- Semantic order-by

- Unstructured Operators

- Semantic extract
- Semantic classify
- Semantic segmentation



Semantic Operators: Definition

- Relational operators are insufficient for unstructured data analytics
 - Lack semantic processing capabilities
- To address this, we manually identify a set of operators to support more comprehensive analytics, each defined with:
 - Input
 - Output
 - Predefined execution processes (physical operators)
 - A set of logical representations
- **A logical representation is a structured template that abstracts the semantic essence of NL expressions** into placeholders like Entity and Condition, capturing distinct semantic roles.
 - Filter: [Entity] satisfy [Condition]; [Entity] that [Condition]; [Entity] by [Condition]

Semantic Operators: Logical/Physical Implementation

- Logical operators: Filter, Extract, Compare, Group By, Classify, Order By ...
- Physical operators: Pre-programmed, LLM-based, LLM-based UDF

THE LOGICAL OPERATORS, THEIR INPUTS, OUTPUTS, CORRESPONDING PHYSICAL OPERATORS, AND EXAMPLE LOGICAL REPRESENTATIONS.

Operator	Input	Output	Pre-programmed Implementation	LLM-based Implementation	Example Logical Representation
Scan	List	List	Linear Scan, Index Scan	-	documents satisfy [Condition]
Filter	List	List	Exact condition filtering	Semantic filtering	[Entity] that [Condition]
Compare	A, B, Condition	A/B	Standard comparison, <i>e.g.</i> , >, <	Semantic comparison	larger in [Entity] and [Entity]
GroupBy		List of List	Grouping by exact attributes	Semantic grouping	aggregate [Entity] by [Attribute]
Count	List	Number	Standard aggregation (Count)	Semantic count	number of documents [Condition]
Sum	List	Number	Standard aggregation (Sum)	Semantic sum	the total sum of [Entity]
Max	List	Number	Standard aggregation (Max)	Semantic max	the maximum of [Entity]
Min	List	Number	Standard aggregation (Min)	Semantic min	the minimum of [Entity]
Average	List	Number	Standard aggregation (Average)	Semantic average	the mean of [Entity]
Median	List	Number	Standard aggregation (Median)	Semantic median	the median of [Entity]
Percentile	List	Number	Standard aggregation (Percentile)	Semantic percentile	the k-th percentile for [Entity]
OrderBy	List	List	Numerical/lexicographical sort	Semantic sorting	Sort [Entity] [Condition]
Classify	Text	Class	Rule-based/ML-based classification	Semantic classification	The type of [Entity]
Extract	Text	Text	Keyword/Regex extraction	Semantic extraction	get [Entity] from documents
TopK	List	List	Numeric ranking	Semantic ranking	the top [Number] [Entity]
Join	List, List	List	Join by key	Semantic join	[Entity] that also occurs in [Entity]
Union	Set, Set	Set	Standard set union	Semantic set union	set union of [Entity] and [Entity]
Intersection	Set, Set	Set	Standard set intersection	Semantic set intersection	in set [Entity] and in [Entity]
Complementary	Set, Set	Set	Standard set complementary	Semantic set complementary	in set [Entity] not in [Entity]
Compute	List	Number	Programmed Mathematical Equation	Semantic computation	sum of squares of [Entity]
Generate	Text	Text	-	LLM invocation	explain the result

Semantic Operators: Two Examples

✓ Semantic Filter

● Logical Representations:

- [Entity] satisfy [Condition]
- [Entity] that [Condition]

● Physical implementations:

- ① Keyword filtering
- ② Embedding-based filtering
- ③ LLM filtering

✓ Semantic GroupBy

● Logical Representations:

- Group [Entity] by [Attribute]
- For each [Entity], aggregate
- The number of [Entity]

● Physical implementations:

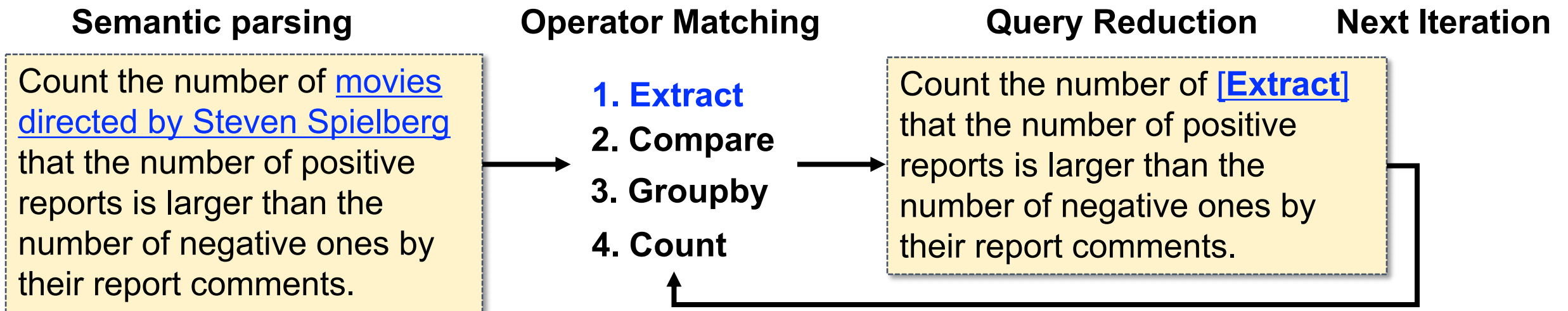
- ① Hash-based Groupby
- ② Embedding-based Groupby
- ③ LLM-based Groupby

Logical: Fixed Reserved Words → Semantic Words (for NL queries)

Physical: Fixed implementation → Semantic implementation

Pipeline Planning: Automatic Orchestration

- **Overview:** progressively identifying appropriate pre-defined logical operators and reducing the query with the operators.
 - ① Semantic Parsing: extract the logical representations from the query
 - ② Operator Matching: identify the matched logical operators
 - ③ Query Reduction: reduce the logical operators to generate a plan
 - ④ Error Handling: backtrack to the previous reduction



Pipeline Orchestration

Count the number of movies directed by Steven Spielberg that the number of positive reports is larger than the number of negative ones by their report sentiments.

Pipeline Orchestration

①

Count the number of movies directed by Steven Spielberg that the number of positive reports is larger than the number of negative ones by their report sentiments.

① Semantic
Extract

Step 1. **Semantic Extract** movies directed by Steven Spielberg

Pipeline Orchestration

①

Count the number of movies directed by Steven Spielberg that the number of positive reports is larger than the number of negative ones by their report sentiments.

②

Step 1. **Semantic Extract** movies directed by Steven Spielberg

Step 2. **Semantic Filter** the reports for the movies

① Semantic
Extract



② Semantic
Filter

Pipeline Orchestration

③

①

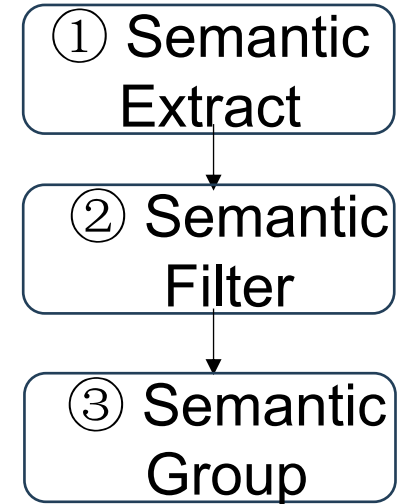
Count the number of movies directed by Steven Spielberg that the number of positive reports is larger than the number of negative ones by their report sentiments.

②

Step 1. **Semantic Extract** movies directed by Steven Spielberg

Step 2. **Semantic Filter** the reports for the movies

Step 3. **Semantic Group** the reports by movie



Pipeline Orchestration

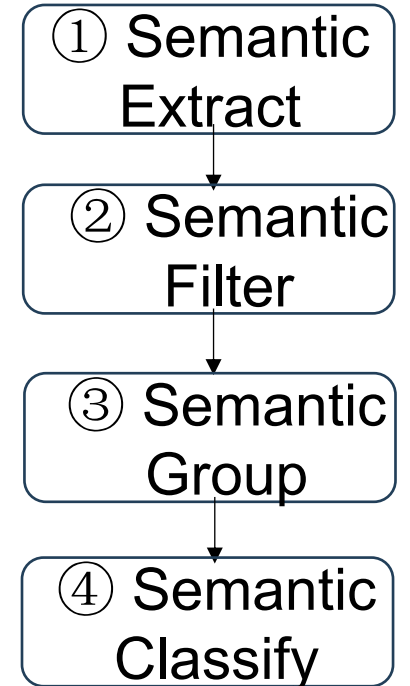
Count ^③the number^① of movies directed by Steven Spielberg that the number of ^④positive reports is larger than the number of negative ones by their report sentiments.

Step 1. **Semantic Extract** movies directed by Steven Spielberg

Step 2. **Semantic Filter** the reports for the movies

Step 3. **Semantic Group** the reports by movie

Step 4. **Semantic Classify** Negative or Positive



Pipeline Orchestration

Count ^③ the number of movies directed by Steven Spielberg ^① that the number of positive reports ^⑤ is larger than the number ^④ of negative ones by their report sentiments ^②.

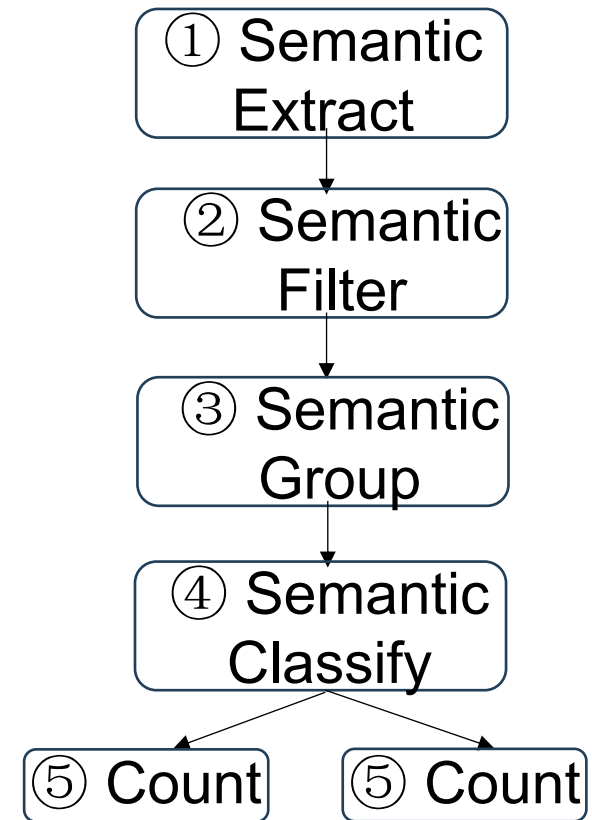
Step 1. **Semantic Extract** movies directed by Steven Spielberg

Step 2. **Semantic Filter** the reports for the movies

Step 3. **Semantic Group** the reports by movie

Step 4. **Semantic Classify** Negative or Positive

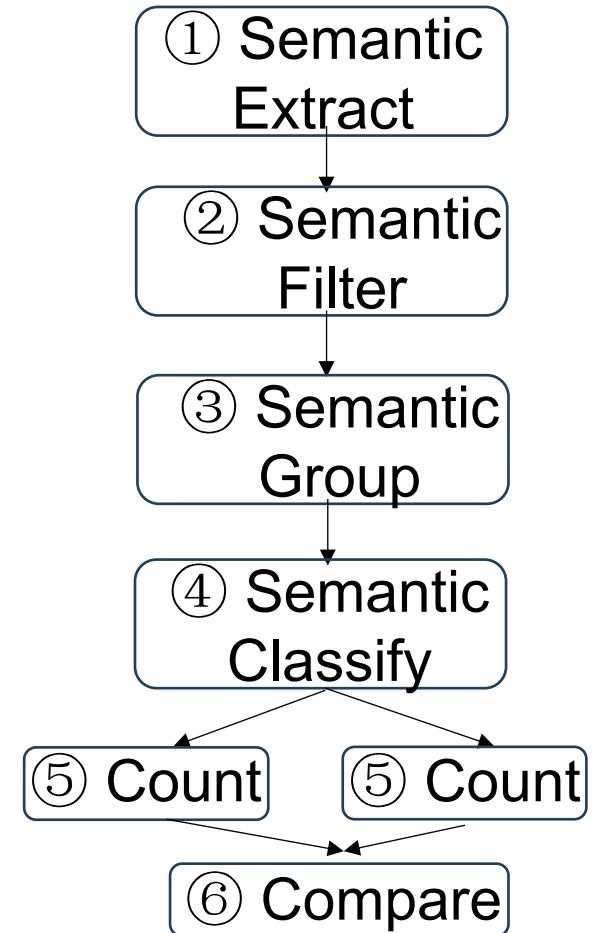
Step 5. **Count** Positive/Negative reports



Pipeline Orchestration

③ Count the number of movies directed by Steven Spielberg that the number of positive reports is larger than the number of negative ones by their report sentiments. ① ⑤ ④

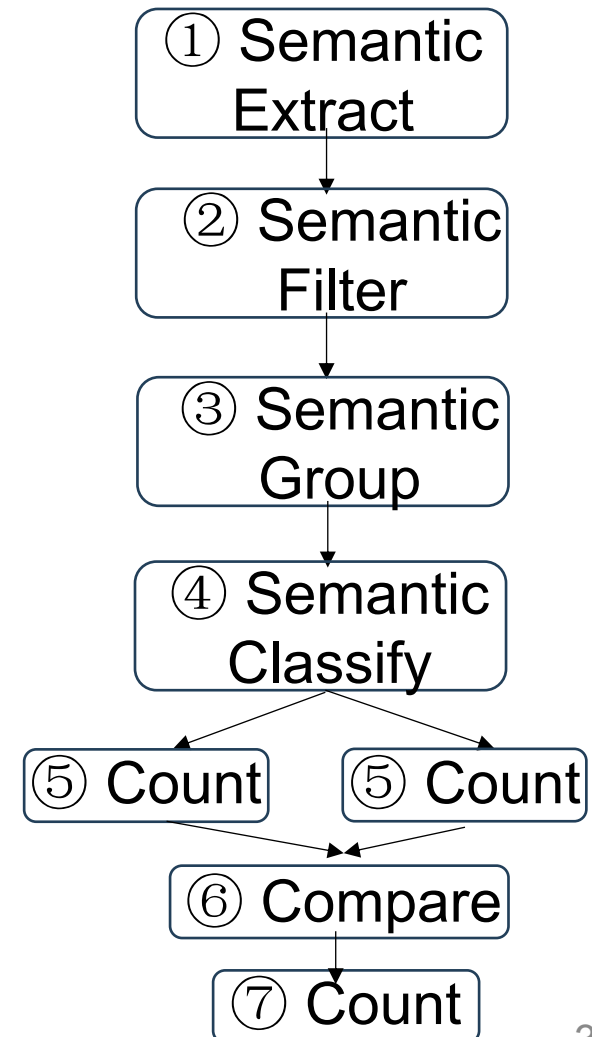
- ⑥
- ⑤
- ④
- ②
- Step 1. **Semantic Extract** movies directed by Steven Spielberg
- Step 2. **Semantic Filter** the reports for the movies
- Step 3. **Semantic Group** the reports by movie
- Step 4. **Semantic Classify** Negative or Positive
- Step 5. **Count** Positive/Negative reports
- Step 6. **Compare** #Positive and #Negative reports



Pipeline Orchestration

⑦ Count the number of movies directed by Steven Spielberg that the number of positive reports is larger than the number of negative ones by their report sentiments.

- ⑥
- ⑤
- ④
- ②
- Step 1. **Semantic Extract** movies directed by Steven Spielberg
- Step 2. **Semantic Filter** the reports for the movies
- Step 3. **Semantic Group** the reports by movie
- Step 4. **Semantic Classify** Negative or Positive
- Step 5. **Count** Positive/Negative reports
- Step 6. **Compare** #Positive and #Negative reports
- Step 7. **Count** the groups with more Positive reports

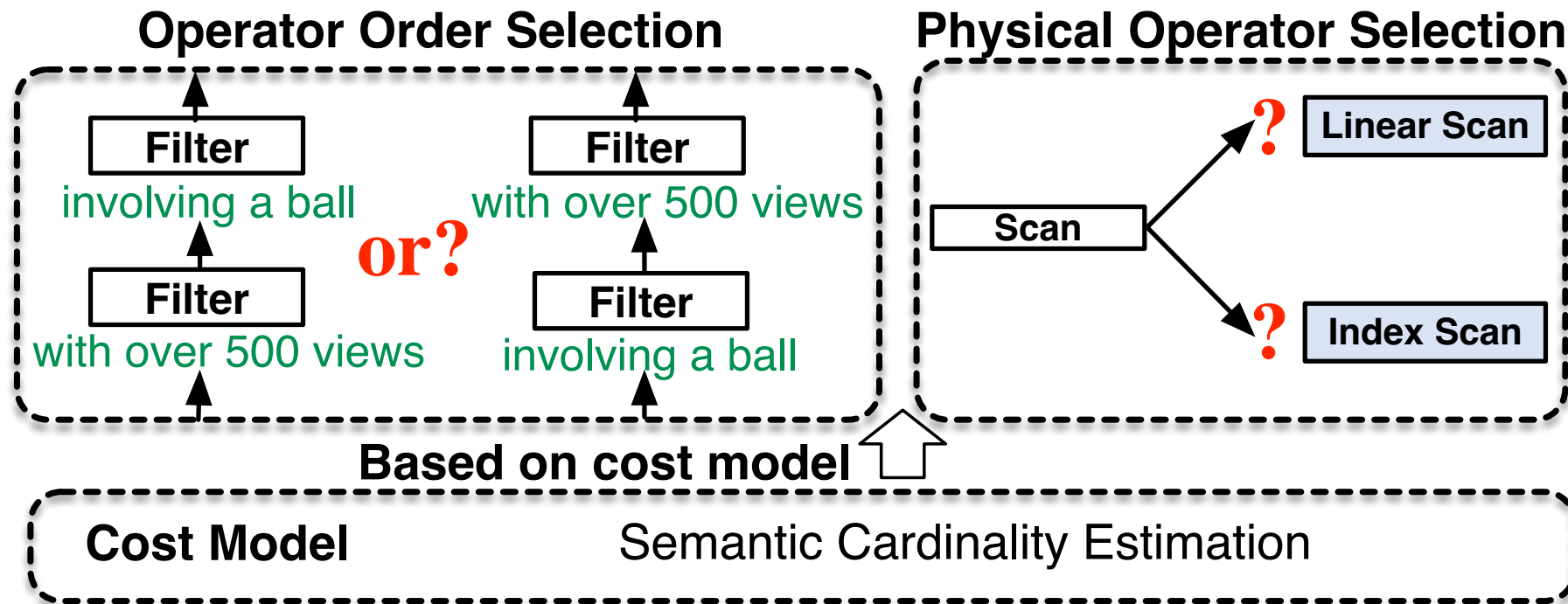


Pipeline Planning: Error Handling

- **Overview:** progressively identifying appropriate pre-defined logical operators and reducing the query with the operators.
 - ① **Semantic Parsing:** extract the logical representations from the query
 - ② **Operator Matching:** identify the matched logical operators
 - ③ **Query Reduction:** reduce the logical operators to generate a plan
 - ④ **Error Handling:** backtrack to the previous reduction
 - If all candidate operators cannot reduce the query:
 - backtracks to the query before the latest reduction
 - If no reduction path can fully decompose the query :
 - Append a Generate operator to produce an answer based on collected information via LLM (RAG).
 - Instruct the LLM to generate Python code for solving the remaining task (fallback to code generation).

Pipeline Optimization

- Physical Operator Selection: LLM, Pre-Programmed, LLM Coding
- Cost Model:
 - LLM-based: Costs depend on **input/output tokens** and **the number of LLM calls (Cardinality)**
 - Program-based: Costs depend on input size and computational complexity
 - **Semantic Cost Model**
- Multi-Goal Optimization: Pipeline Quality, LLM Cost, Query Latency



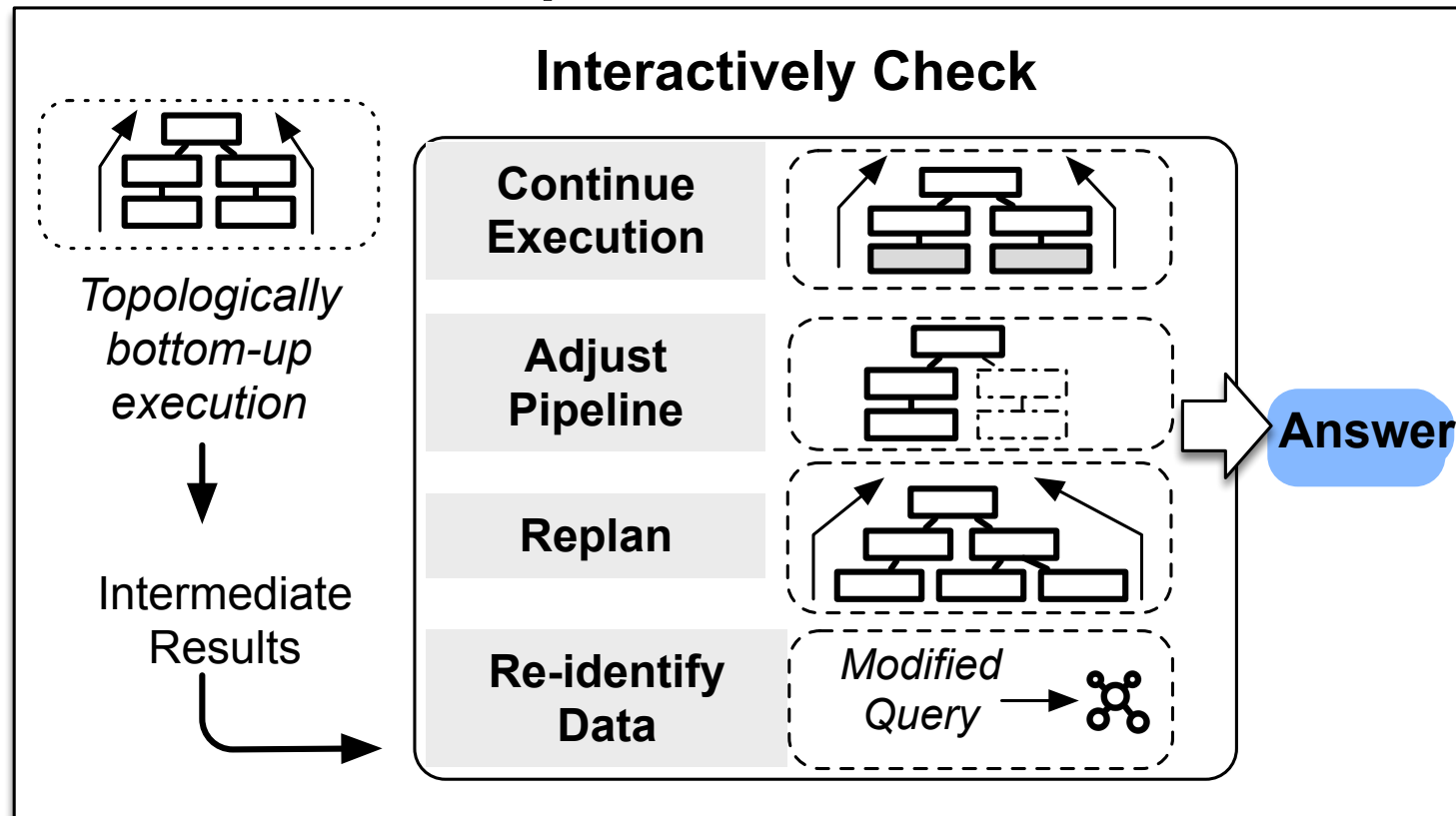
Pipeline Self-reflection

- Pipeline evaluation
 - Generate multiple pipelines, and select the best pipeline
 - Using samples to get some preliminary results and evaluating them
 - Cross validations
 - Use errors/warnings to provide feedbacks
- Reward Model
 - Evaluate the utility of a pipeline and give a reward
 - Compute the utility of the current local pipeline
 - Predict the utility of the following global pipeline

Pipeline Execution

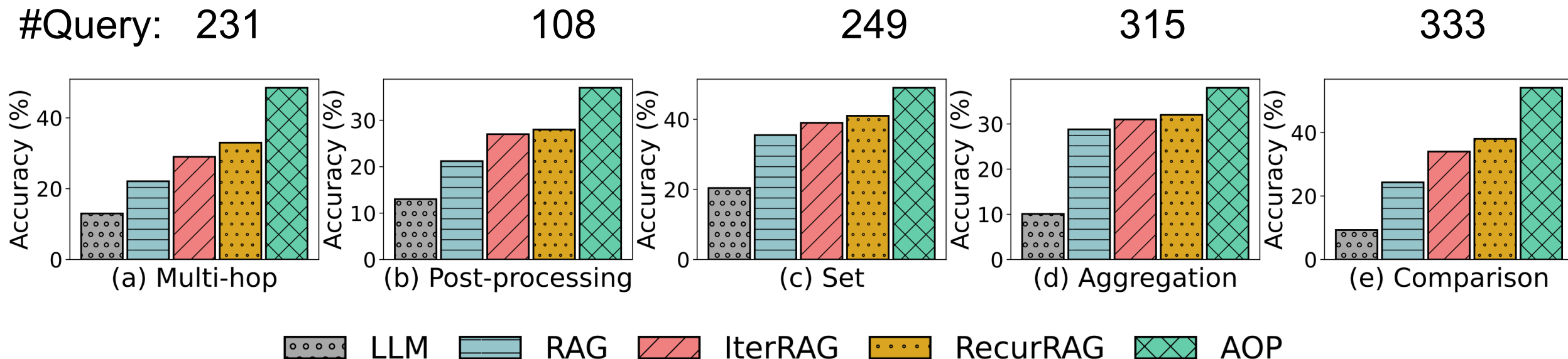
- **Interactive Plan Adjustment During Execution:** When operator execution fails, dynamically adjusts the plan by continuously replanning
- **Parallel Execution** for low latency
- **Multi-way Execution** for high accuracy (Vector & LLM, Merge operators)

Pipeline Execution



Experimental Results

Real datasets and real NL queries



30%-40% accuracy improvement against GPT-4

Real Use Case (DABstep Benchmark)

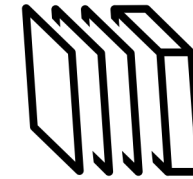
TASK

For account type H and the MCC description: Eating Places and Restaurants, what would be the average fee that the card scheme GlobalCard would charge for a transaction value of 10 EUR? Provide the answer in EUR and 6 decimals

Unstructured



Chunking



VexDB



Concept Understanding

Data Profiling

1. Understanding Payment Processing Fees

Payment Processing Fees depend on a number of characteristics. These characteristics belong to either the merchant or the transaction.

Merchant characteristics include:

* **ID**: identifier of the fee rule within the rule fee dataset

* **card_scheme**: string type. name of the card scheme or network that the fee applies to

2. Documentation for the payments.csv dataset

- **Description**: Synthetic dataset of payment transactions processed by the Payments Processor.


- **Columns**:

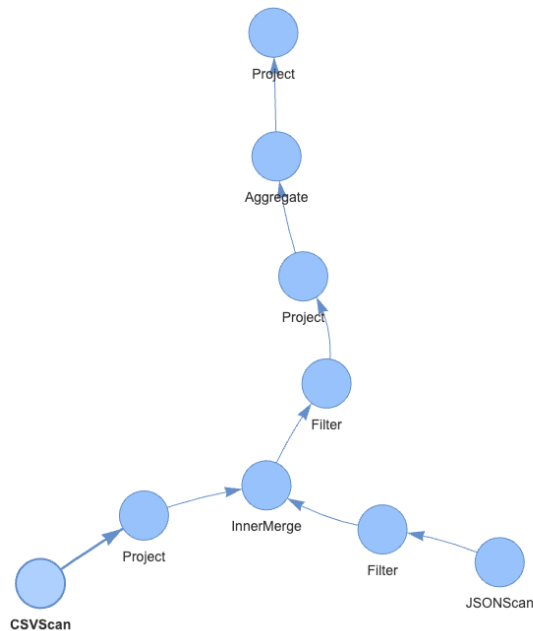
- `psp_reference`: Unique payment identifier (ID).

- `merchant`: Merchant name (Categorical), eg Starbucks or Netflix*.

- `card_scheme`: Card Scheme used (Categorical) - `[MasterCard, Visa, Amex, Other]`.*.

Real Use Case (DABstep Benchmark)

Pipeline  Feedback  Feedback  Refined Pipeline



Fix parsing error due to non-existent column

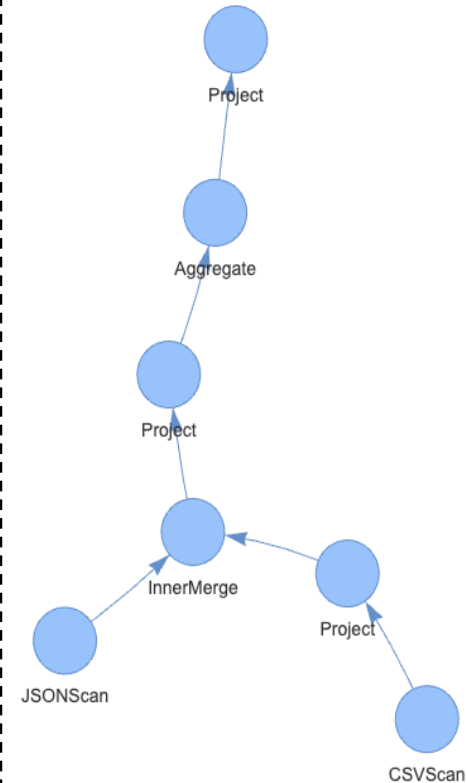
Error in plan execution:
invalid syntax (<string>, line 1)
Traceback (most recent call last):

File
"D:\lab\dataagent\queryprocess\
executor\aggregate.py", line 54
SyntaxError: invalid syntax

Remove the redundant Filter operators.

The plan incorrectly filters out fee rules by enforcing conditions on capture_delay

To fix the plan:
1. ****Remove the Filter node**** that enforces conditions on irrelevant fields.



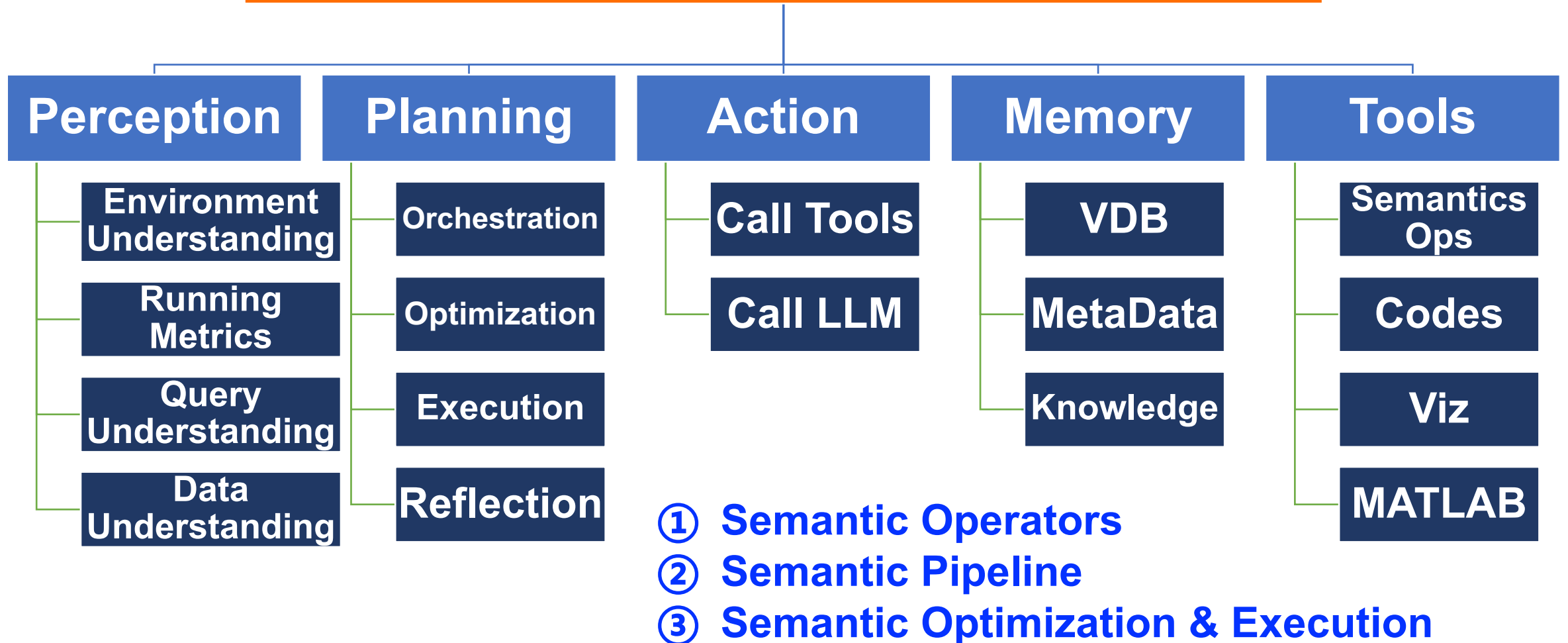
Unstructured Data Analytics Agent

- ✓ Standard semantic operators for pipeline orchestration
- ✓ Automated pipeline orchestration for semantic data analytics
- ✓ Pipeline optimization & execution techniques
- ✓ Automatic support for multi-hop and semantic analytics
- ✓ Extensive experiments demonstrate that our method achieves both high accuracy and high efficiency.

Open source: <https://github.com/TsinghuaDatabaseGroup/Unify>

Unstructured Data Analytics Agent

Unstructured Data Analytics Agent



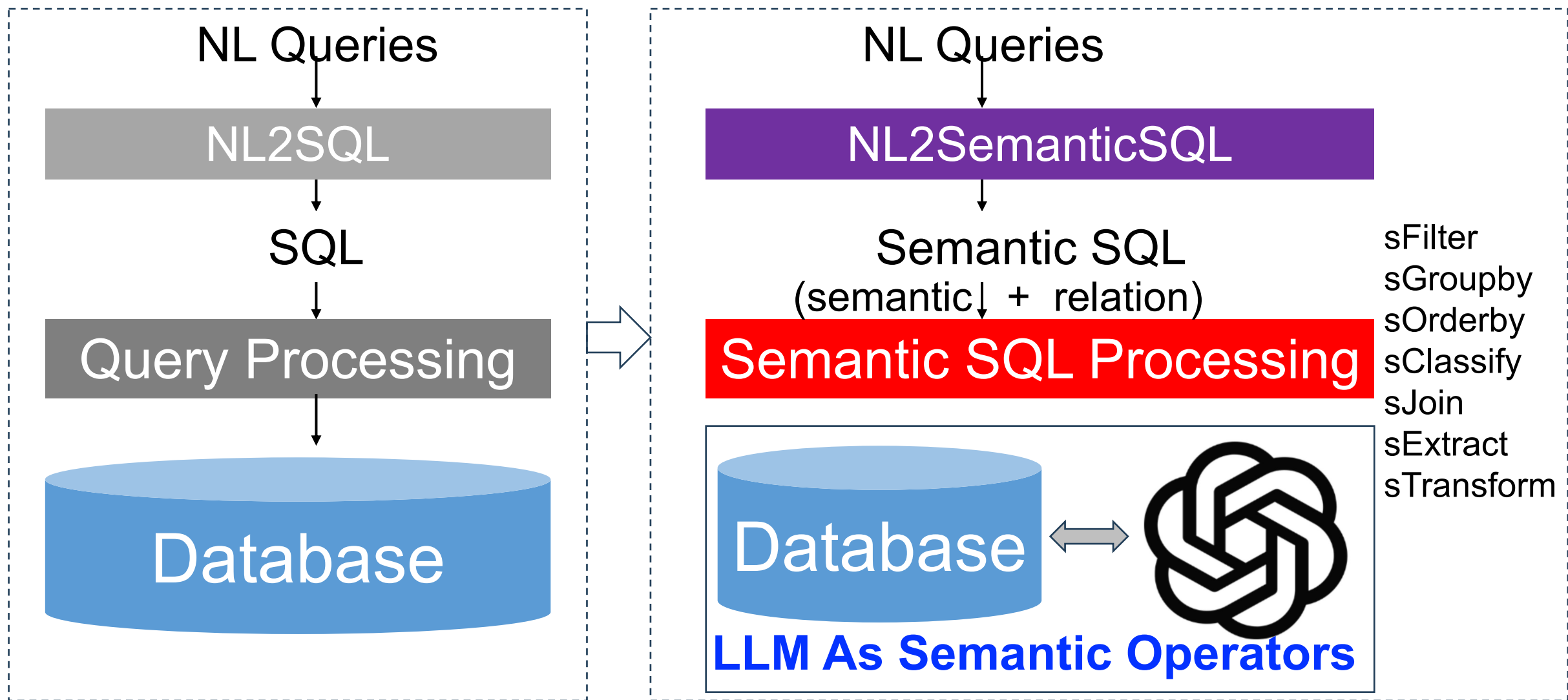


Data Agent for Semantic Structured Data Analytics

Semantic Structured Data Analytics Agent

AS IS

TO BE



Semantic Structured Data Analytics Agent

What are the number of positive and negative reviews for sci-fi movies given by users located in the capital of the US?

```
SELECT COUNT(Review)
FROM MovieReview
WHERE City = SemanticExtract(US)
AND
SemanticClassify(Movie)="sci-fi"
SemanticGroupBy(Review)
```

NL Queries

NL2SQL

Semantic SQL

Semantic Filter
Semantic Groupby
Semantic Orderby
Semantic Classify
Semantic Join
Semantic Extract
Semantic Transform

Semantic SQL Processing

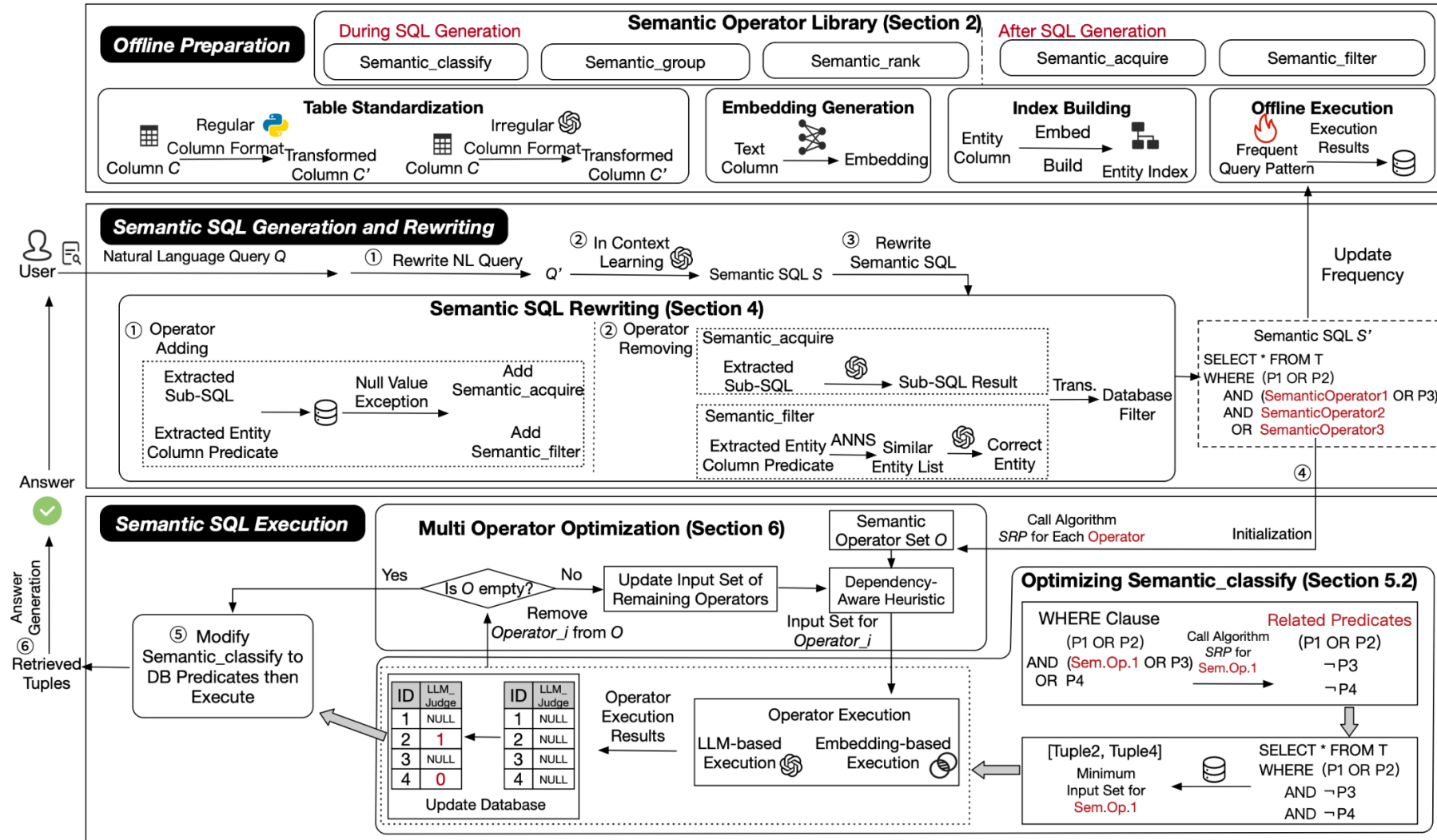


Hybrid LLM/DB Processing

LLM As
Semantic
Operators

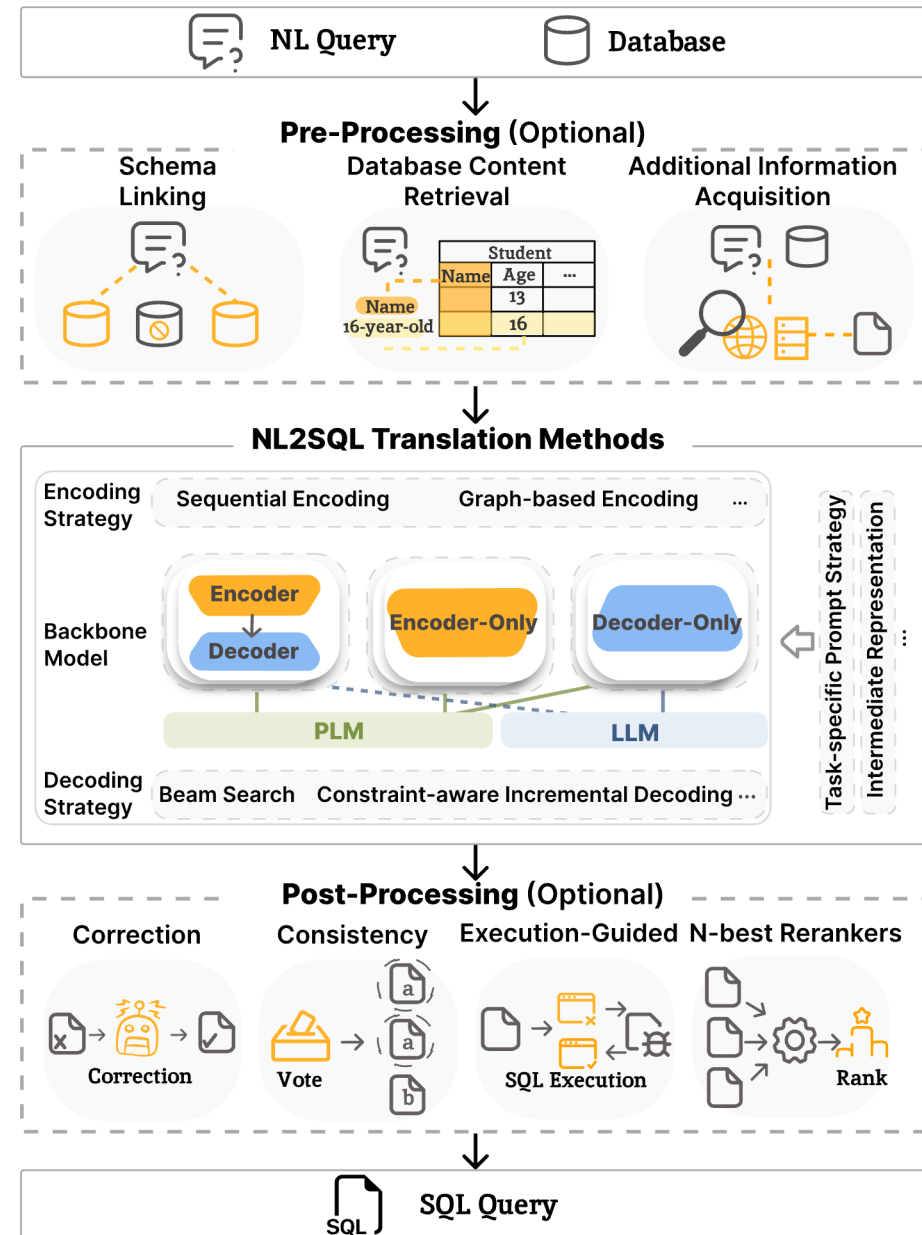
Semantic Structured Data Analytics Agent

- NL2SemanticSQL
- Semantic SQL rewrite
- Single Semantic OP Optimization
- Multiple Semantic OP Optimization



NL2SemanticSQL

- Query-Data Alignment
 - Query Rewrite
 - Query Augmentation
 - Schema Augmentation
 - Data Augmentation
 - Semantic Schema Linking
- NL2SemanticSQL Translation
 - Multiple SQL Generation
 - Best SQL Selection
 - SQL Reflection



Semantic Operator Rewrite

- Semantic operators are extensive.
- Reduce semantic operators to traditional operators
 - **Semantic Acquire:** report NBA players higher than Stephen Curry?
 - ① Acquire the height of Stephen Curry from LLM
 - ② Replace semantic acquire with height = 1.87m
 - **Semantic Filter:** report the conferences held in HK
 - ① Identify cities that are semantically equivalent to HK, such as Hong Kong, Pearl of the Orient, Heung Gong.
 - ② Replace semantic filter with “IN (HK, Hong Kong, Pearl of the Orient, Heung Gong)”

Semantic Plan Rewrite

- Invoking LLMs to execute semantic operators is expensive, it is vital to reduce LLM invocations

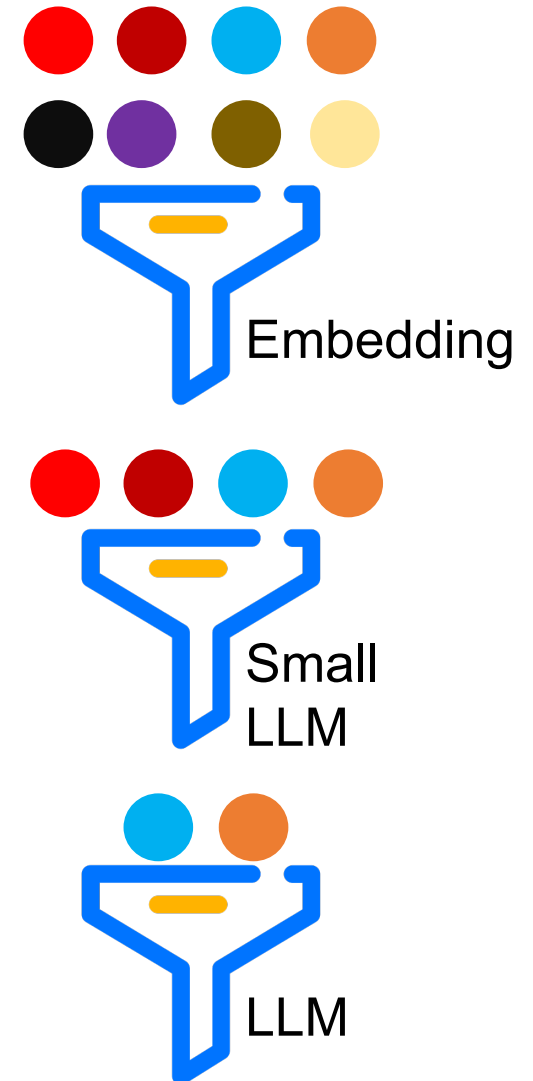
- LLM Bypass

- Using Embeddings/Vectors for Pruning
- Using Small LLMs (LLM cascade) for Pruning
- Using LLM Coding for Pruning (samples to verify the code)

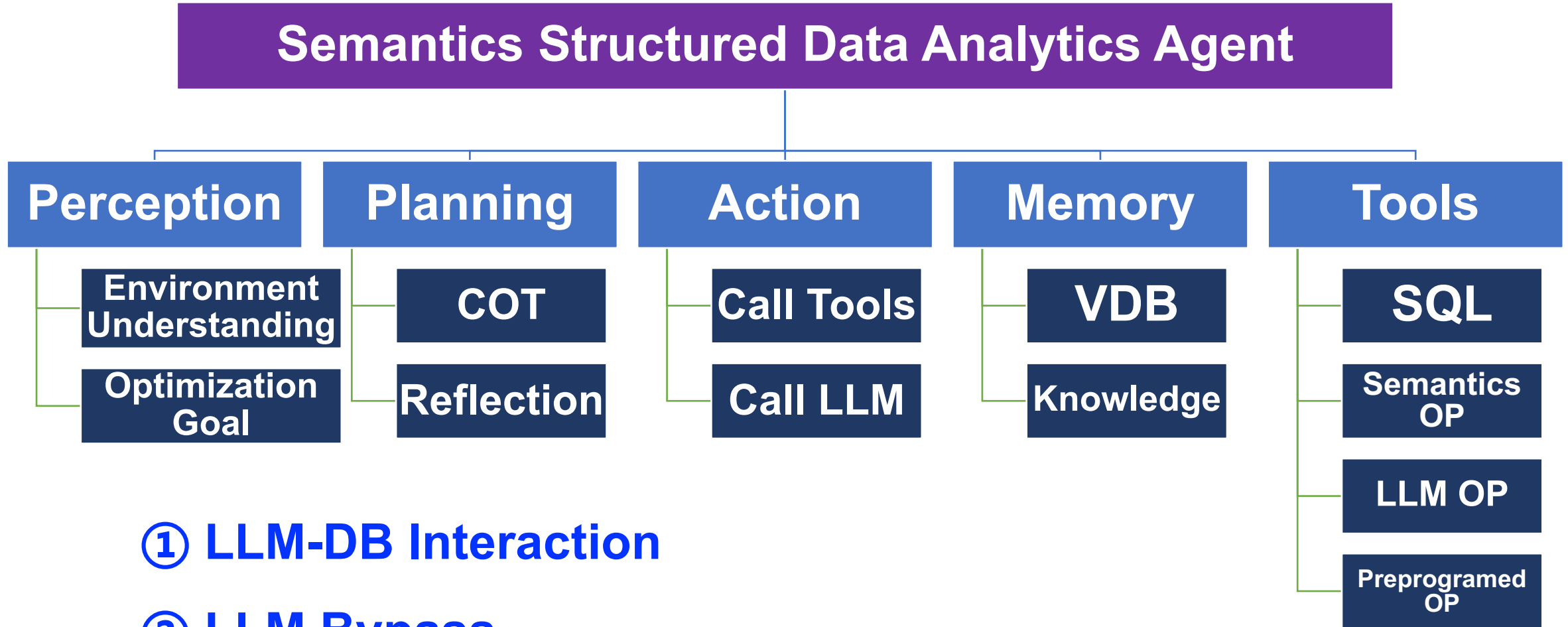
- Plan Rewrite

- Push down: First DB operators and then LLM operators
- Cost-based semantic operator order

- Multi-Goal Optimization (Quality, Latency, Cost)



Semantic Structured Data Analytics Agent



① LLM-DB Interaction

② LLM Bypass

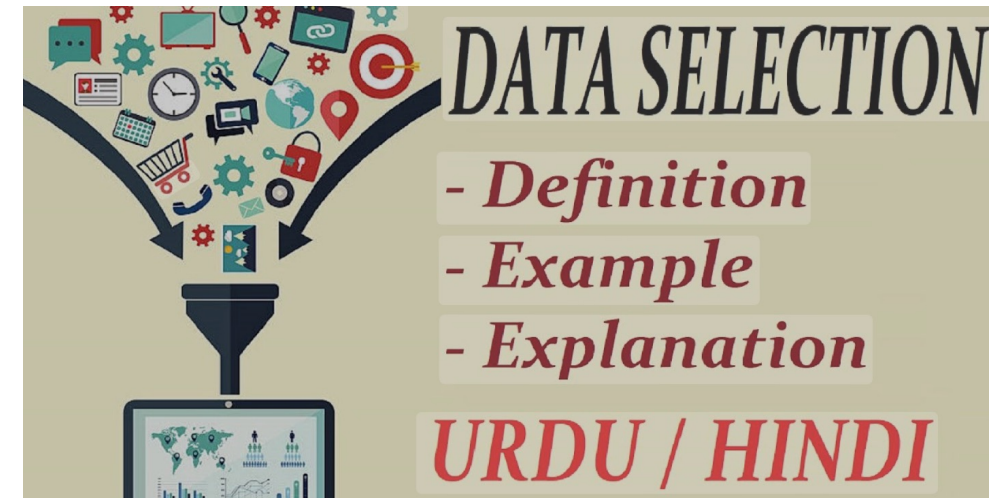
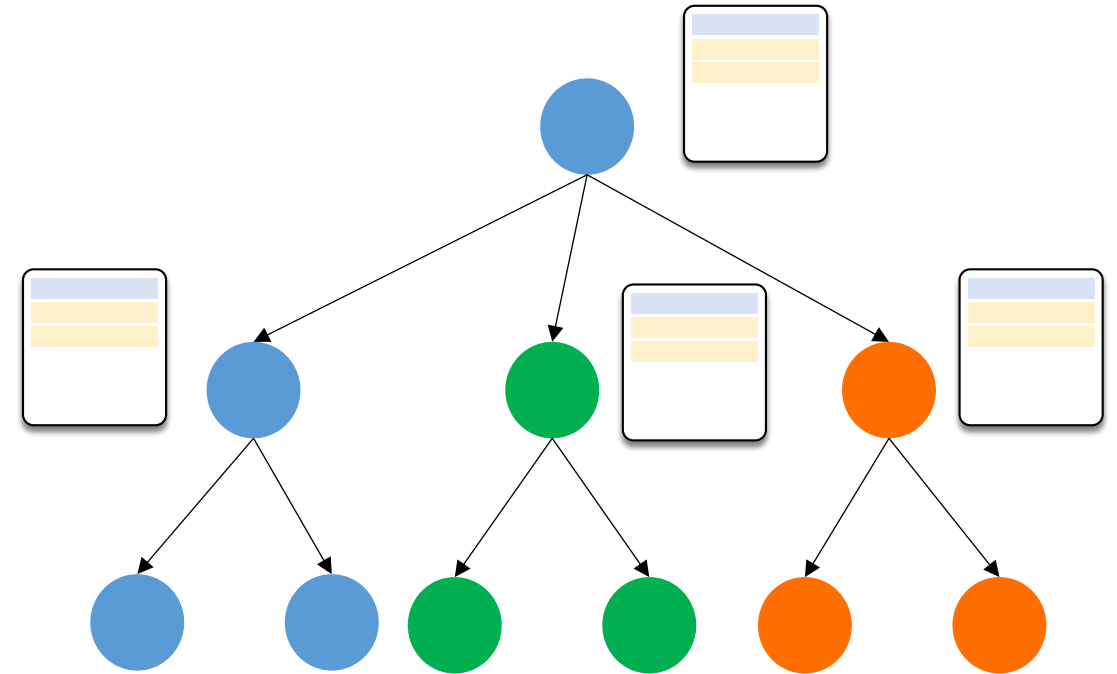
③ Semantic SQL Generation



Data Agent for Data Lake Analytics

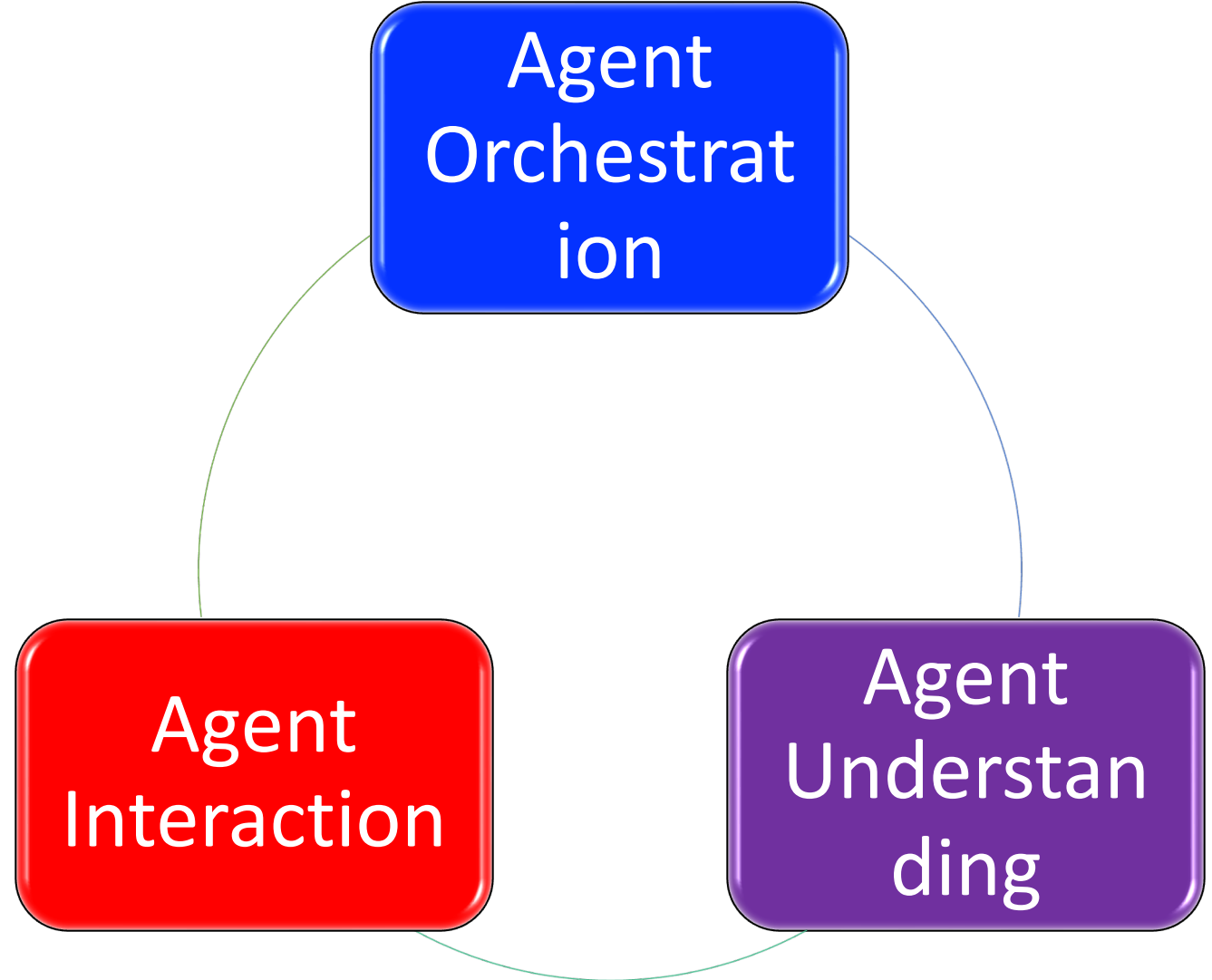
Data Lake Analytics Agent: Challenges

- Heterogenous Data
 - Data Linking
 - Data Embedding Model
 - OOD Indexing
- Semantic Source Selection
 - Unified Catalog
 - Semantic Data Organization
 - Hierarchical Data Exploration
- Pipeline Orchestration & Optimization
 - Agent Selection
 - Agent Interaction



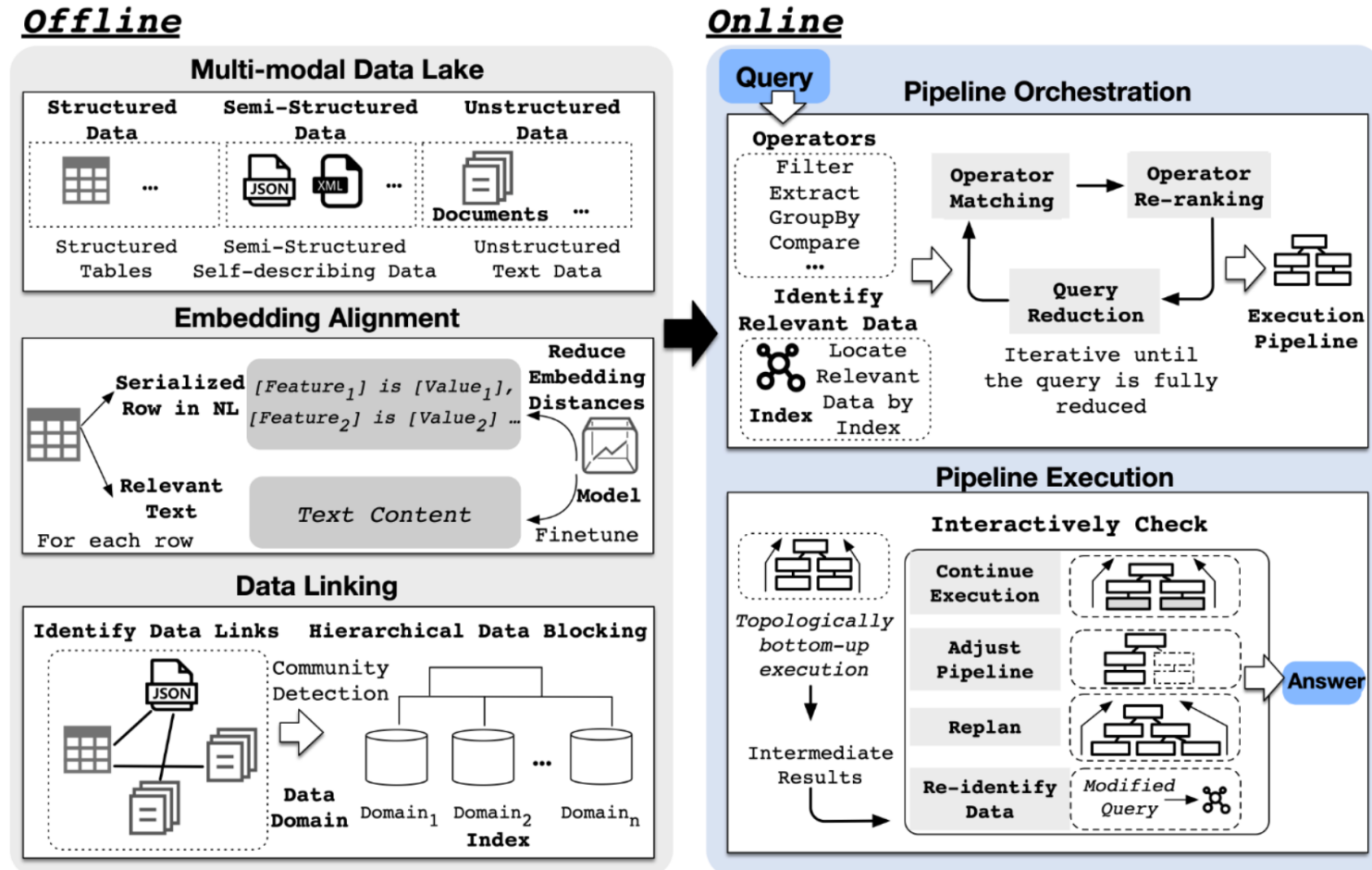
Data Lake Analytics Agent: Multi-Agents

- **Pipeline Orchestrator Agent**
- **Data Agents**
 - **Active Meta Data Management**
 - **Hierarchical Data Selection**
 - **Semantic Catalog**
 - **Semantic Data Statistics**
- **Analytics Agents**
 - **Unstructured Data Analytics**
 - **Structured Data Analytics**
- **Tool Agents**



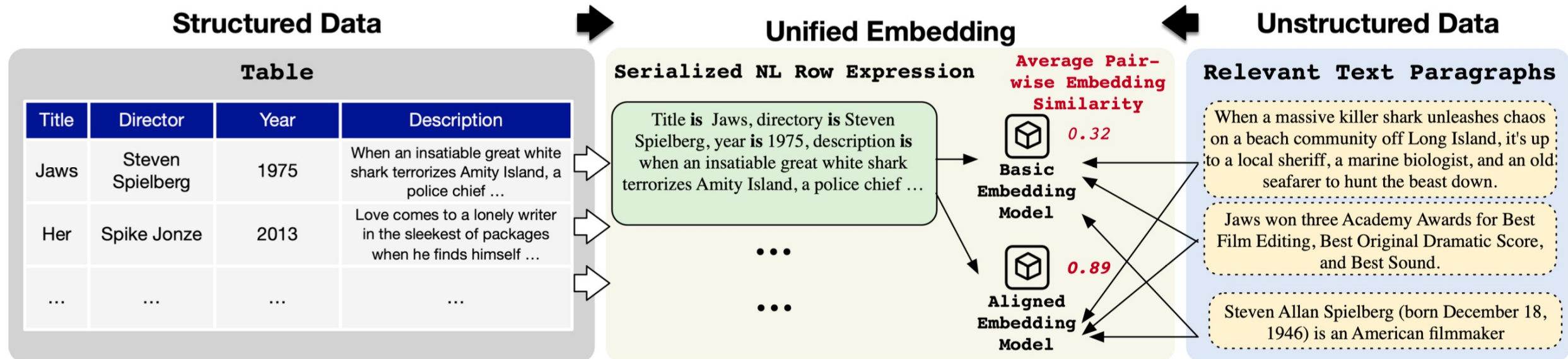
Data Lake Analytics Agent : Architecture

- Offline Data/Knowledge Preparation
 - Data Preparation for RAG and Prompt Engineering
 - Semantic Segmentation
 - Vector Indexing
 - Data Linking
 - Environment/Tool Understanding
 - Fine-tuning for alignment
 - Tool Calling
- Online Agent for optimization
 - Pipeline Orchestration
 - Pipeline Execution
 - Agent Interaction



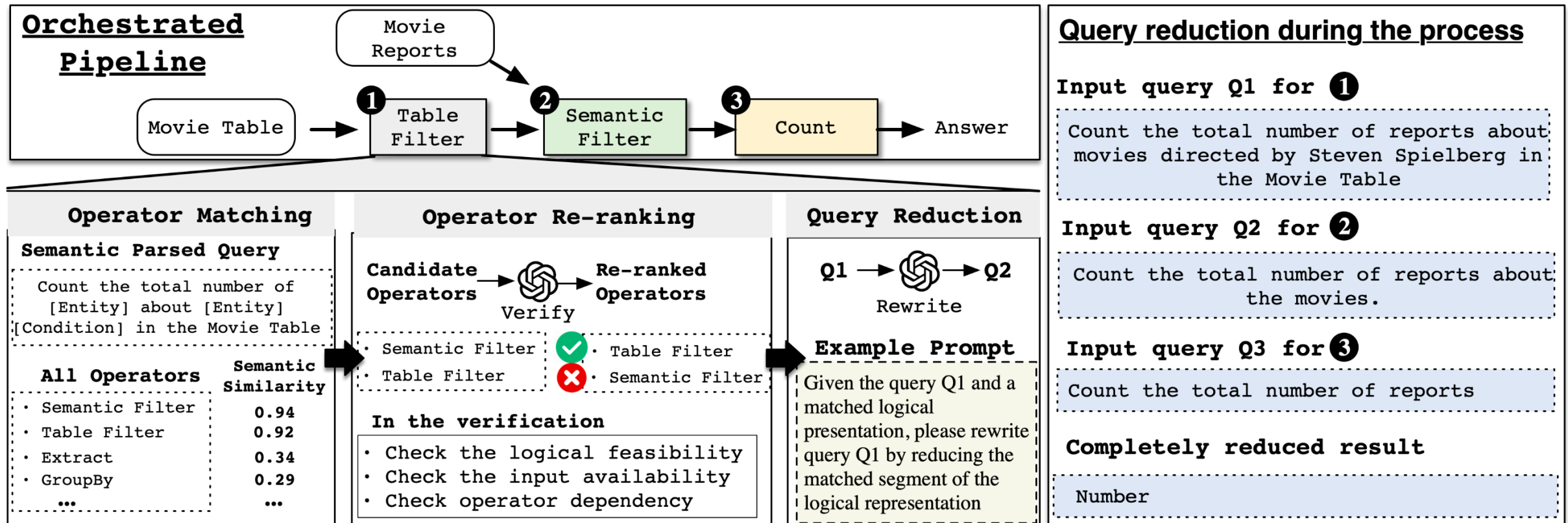
Data Lake Analytics Agent: Data Fabric

- Unified Data Access: Provides a single, consistent interface for accessing data, facilitates real-time data access and sharing across the organization.
- Semantic Catalog and Semantic Data Organization
- Unified Embedding
 - Transforms diverse data types into a unified embedding space
 - Data annotation and augmentation

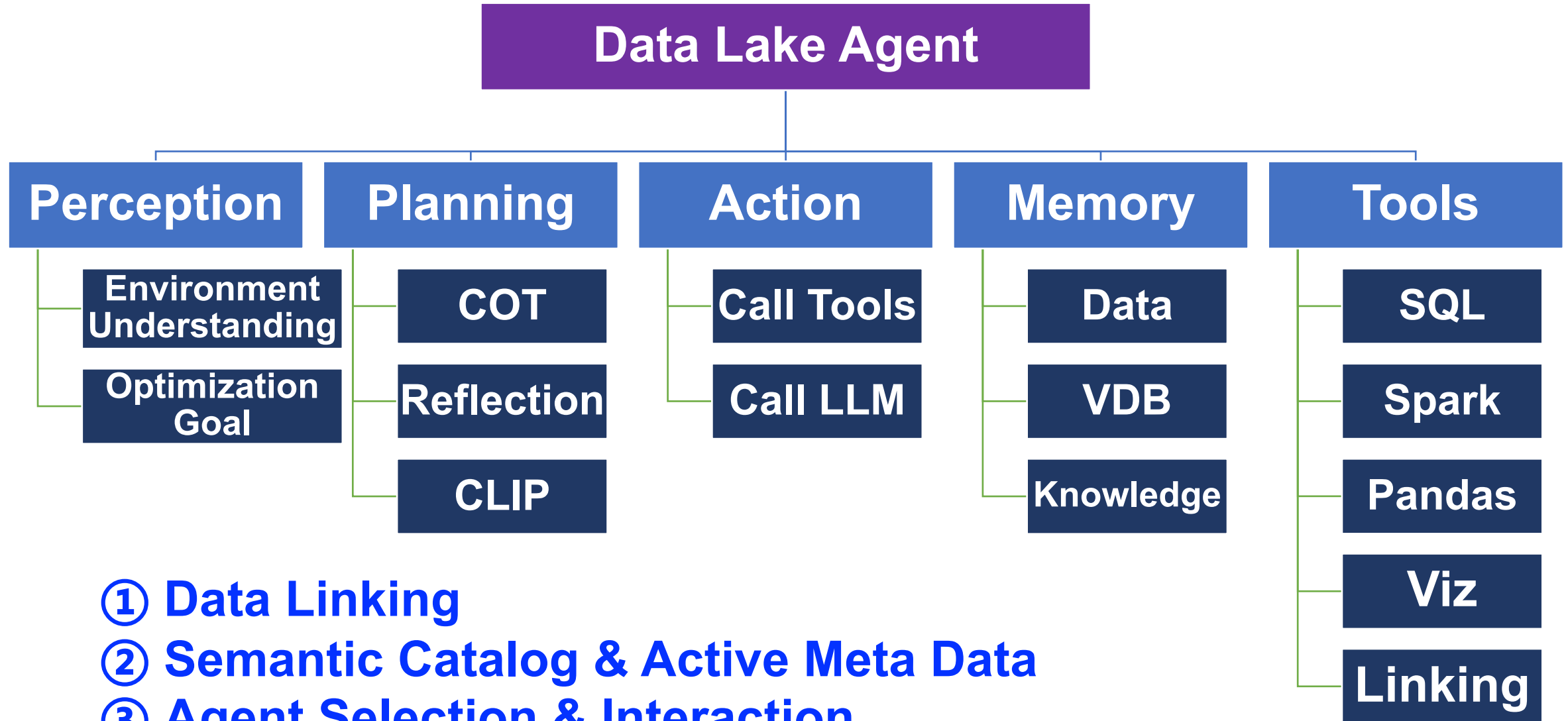


Data Lake Analytics Agent: Orchestration

- Pipeline Orchestration: Iterative Query Decomposition
 - Coarse-grained matching: identifying suitable agents
 - Fine-grained re-ranking: identifying suitable operators in agents
 - Reducing: applying a feasible operator to simplify the query



Data Lake Analytics Agent

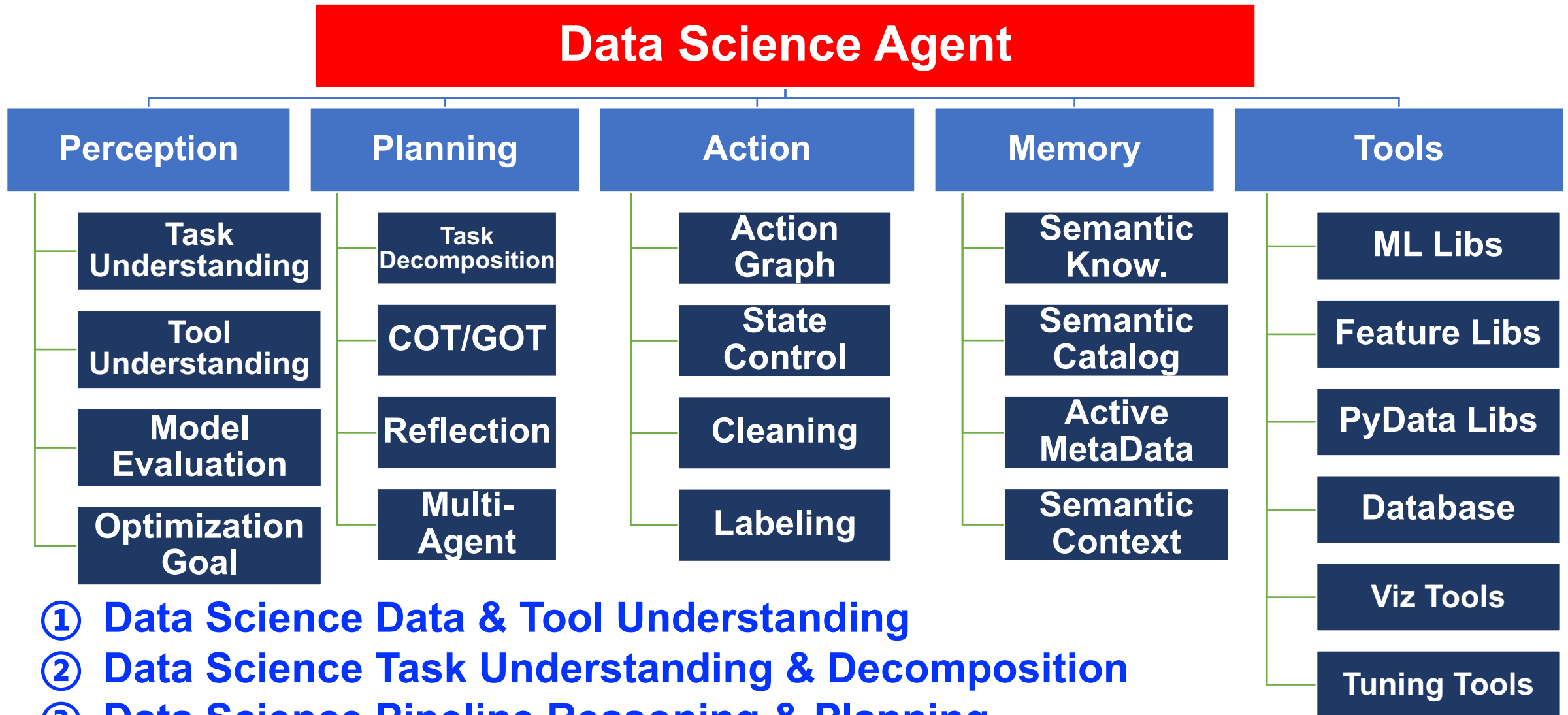


① Data Linking

② Semantic Catalog & Active Meta Data

③ Agent Selection & Interaction

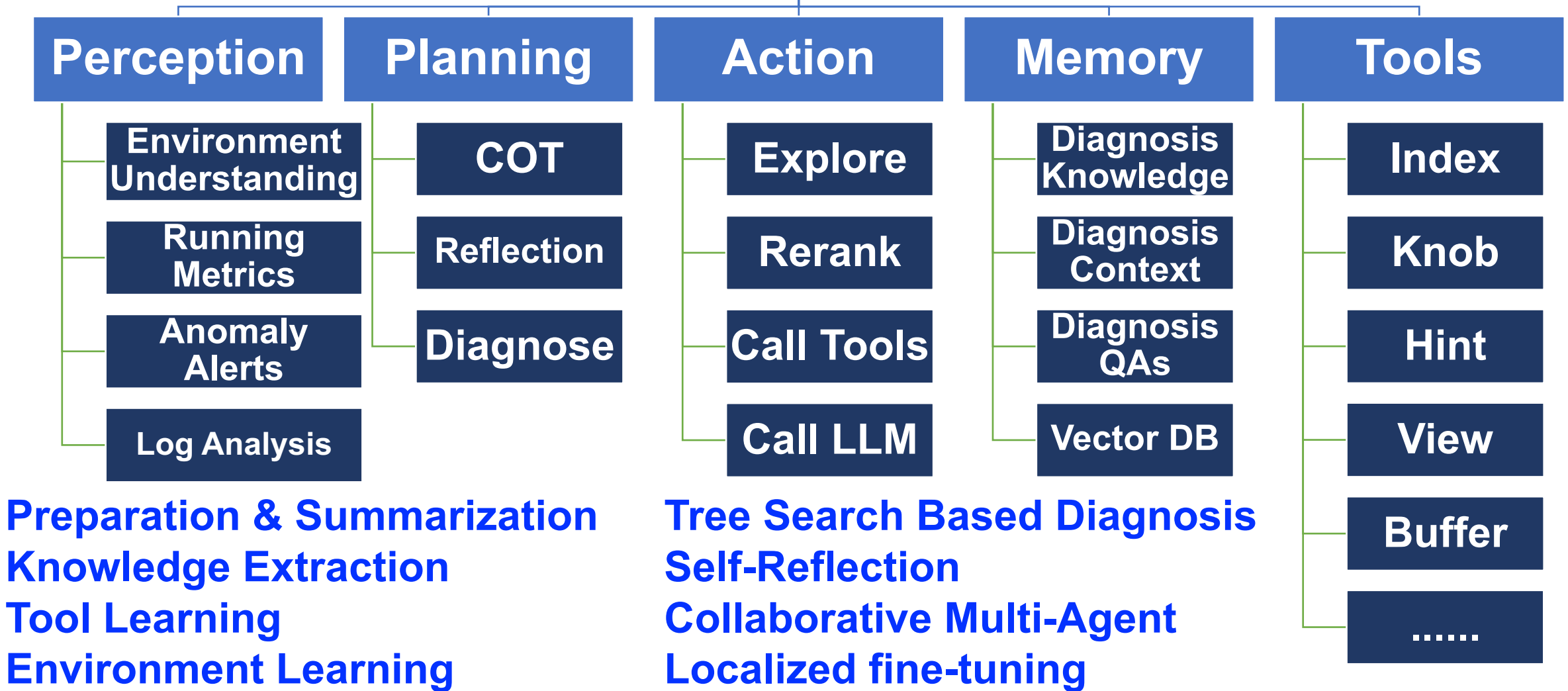
Data Science Agent



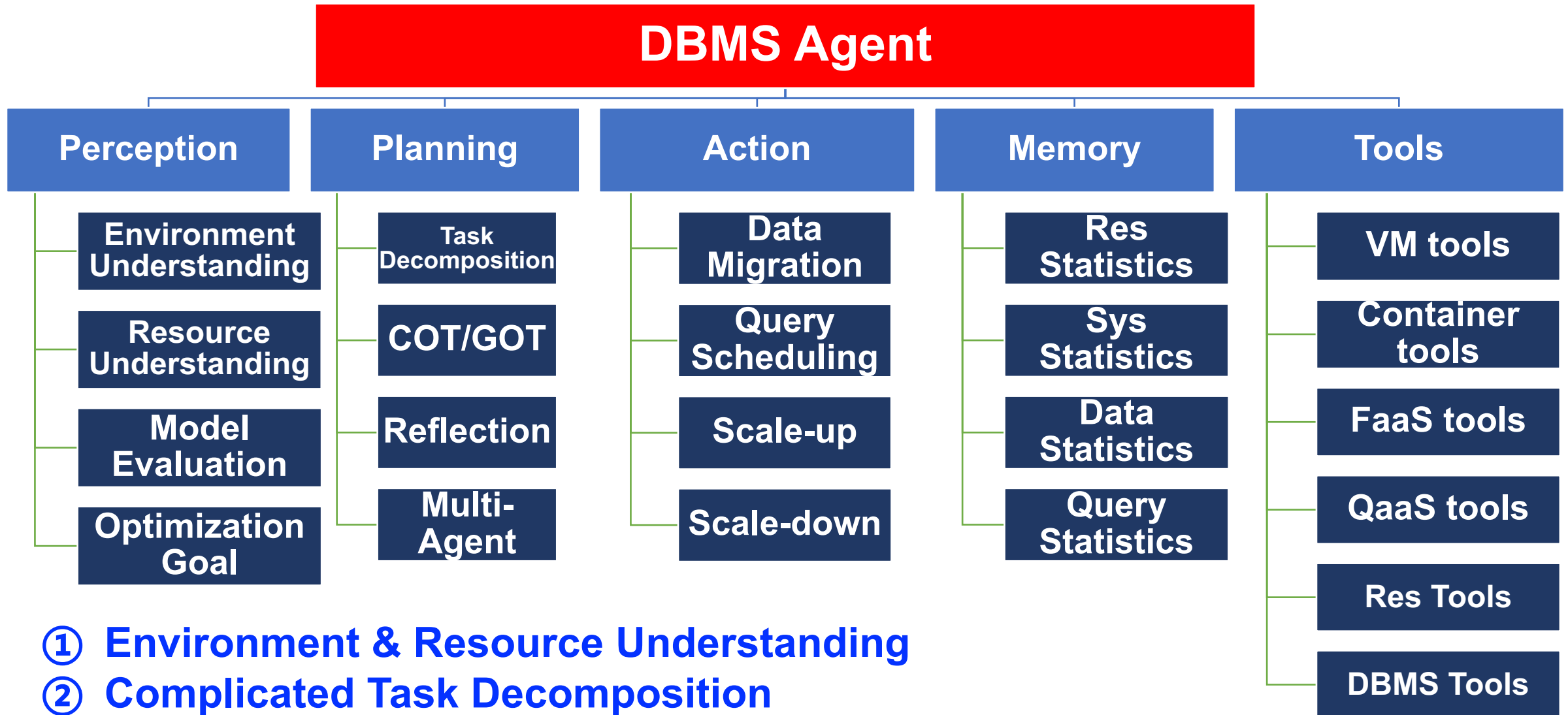
- ① Data Science Data & Tool Understanding
- ② Data Science Task Understanding & Decomposition
- ③ Data Science Pipeline Reasoning & Planning
- ④ Multiple Agents Collaboration & Interaction

Database Administrator Agent

DBA Agent

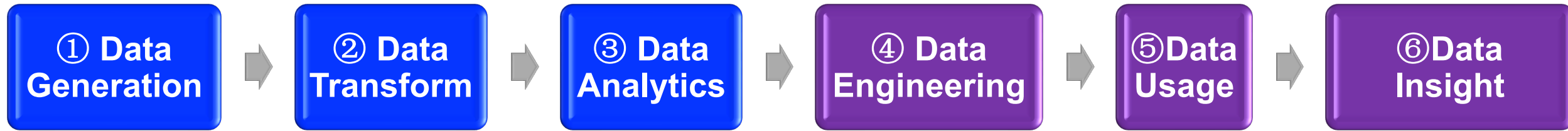


DBMS Agent



- ① Environment & Resource Understanding
- ② Complicated Task Decomposition
- ③ Resource Scheduling & Planning
- ④ Multiple Agents Collaboration

Data Agent Opportunities in Data Lifecycle Management



① OLTP

- Regression
 - Cardinality/Cost Estimation
- Online NP Optimization
 - Query Rewrite
 - Plan Selection
- Offline NP Optimization
 - Knob/Index/View Advisor
- Prediction
 - Workload
 - Resource
 - Data
- Database QAs
- Database Diagnosis
- SQL Dialect Translation

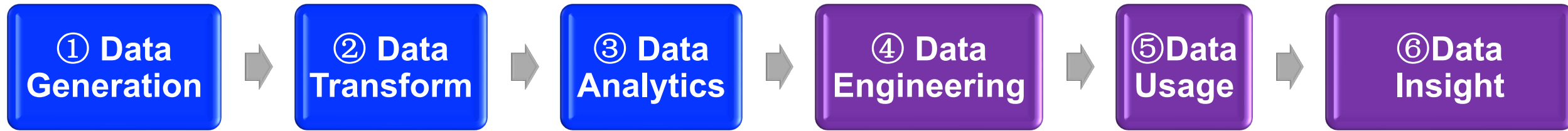
② Data Transformation

- Data Standardization
 - Automate Extract, Transform, Load
 - Auto schema mapping
 - Identifying patterns
 - Low code ETL
 - Predictive auto-scaling
 - Adaptive Transformation
- Multi-goal Optimization
 - Change data capture
 - Automate CDC
 - Predictive analytics
 - Reduce Human Cost

③ OLAP

- In-Database ML
 - In-Database Model
 - In-Database Vector
 - In-Database RAG
 - Anomaly Detection
 - Risk control analysis
- SQL+ML Analytics
 - TableQA
 - In-DB Semantics Analytics
- Autonomous Analytics
 - Autonomous workload management
 - Autonomous data format
 - Autonomous serverless

Data Agent Opportunities in Data Lifecycle Management



④ Data Engineering

- Data Preparation
 - Data Discovery
 - Data Selection
 - Data Cleaning
 - Data Transformation
 - Data Integration
 - Data Generation
 - Data Mixing
 - Data Extraction
 - Data Labeling
 - Meta Data Manag.
- Data Flywheel
- Data Fabric

⑤ Descriptive Data Usage

- Prompting
 - Automation/Examples
- RAG
 - Multi-hop RAG
 - Graph RAG
 - Agentic RAG
- LLM Inference
 - Prefill/Decoder disagg.
 - KV cache
 - Scheduling
 - Data/Model/Resources Parallism
 - Quantization

⑥ Proactive Data Interpretation

- Proactive Insights
 - Trends summarization
 - Insight discovery
 - Predictive decision making
 - Cognitive Analytics
- Proactive Data Agent
 - Chat2Data
 - Chat2BI
 - NL2Viz
 - In-DB Semantics Analytics
 - Unstructured data analytics
 - Multi-model data analytics on data lake

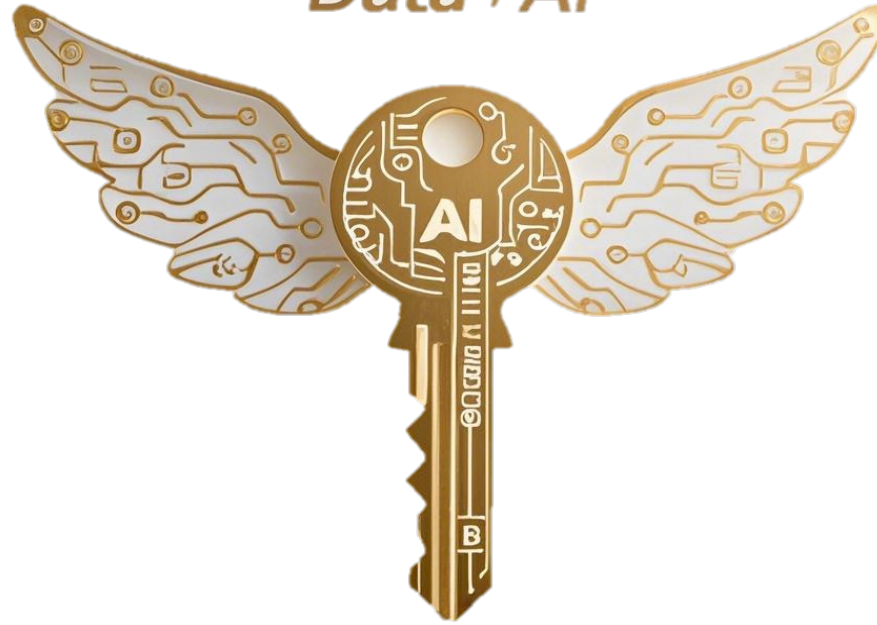
Conclusion

- **Data+AI is important for data management and analytics**
- **It urges the use of Data+AI techniques to revolutionize data systems**
 - **Data science, Data Analytics, Data Lake**
- **Data Agent is a promising direction for Database, Data, Data+AI**
 - **Agent Orchestration and Scheduling**
 - **Multi-Agents Interaction**
 - **Agent Memory**
 - **Proactive Data Management**
- **Open-source systems for Data Agent**



Data Agent

Data+AI



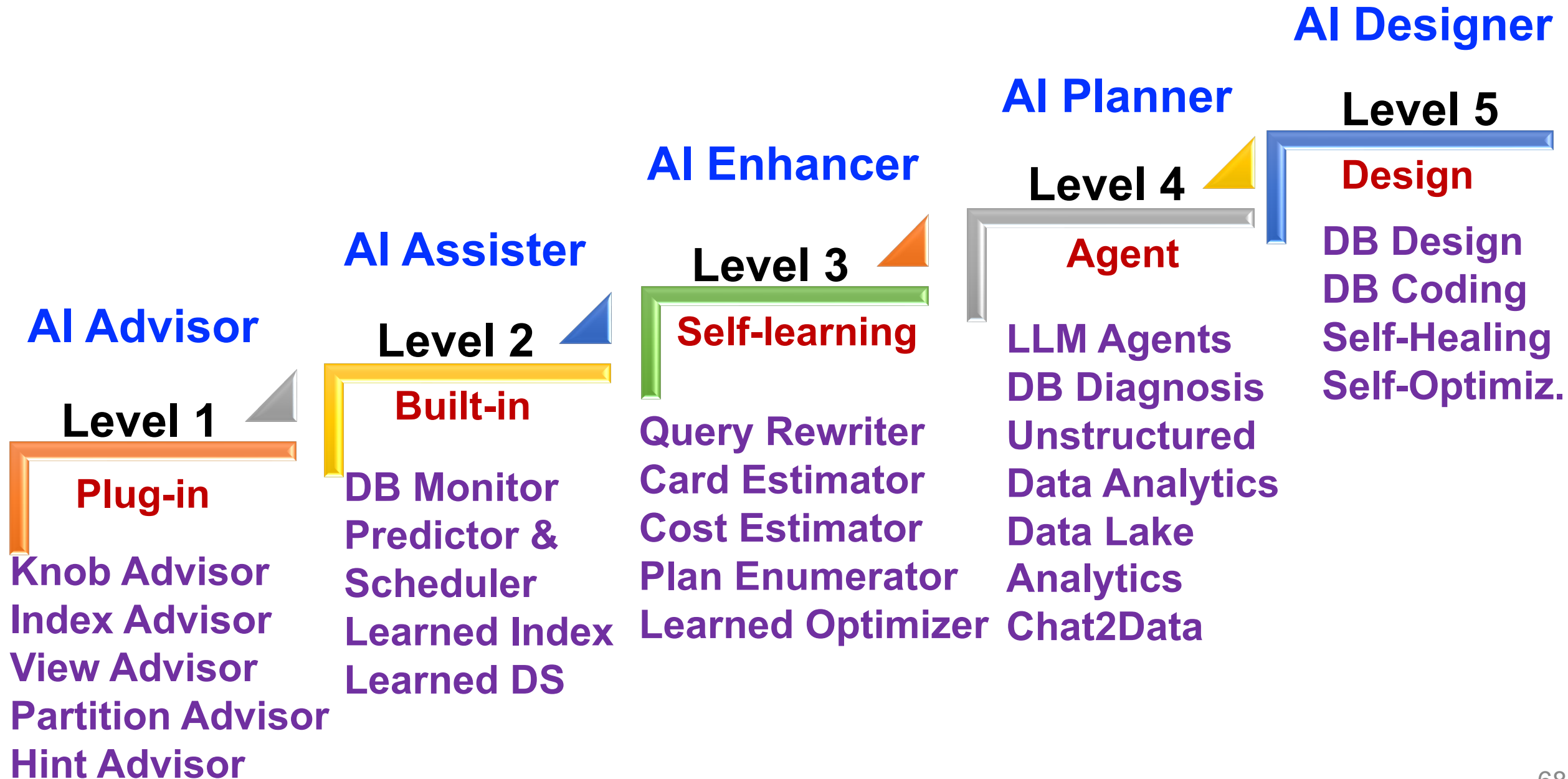
Thanks!

Slides: <https://dbgroun.cs.tsinghua.edu.cn/ligl/activities.html>

Data+AI Paper List: <https://github.com/code4DB/LLM4DB>

System: <https://github.com/TsinghuaDatabaseGroup/Unify>

Five Levels of AI4Data



Five Levels of Data4AI

