ICDE'18

Cross-Platform Data Processing

- Use Cases & Challenges -

Zoi Kaoudi
Jorge Quiané
Qatar Computing Research Institute (QCRI)

Nature Inspired

Going to Race?

Going to Race?

Going to the Desert?

Going to the Desert?

Going to the Forest?

Going to the Forest?

Going to the Everest?

Going to the Everest?

Going to Fly?

Going to Fly?

FACT 1

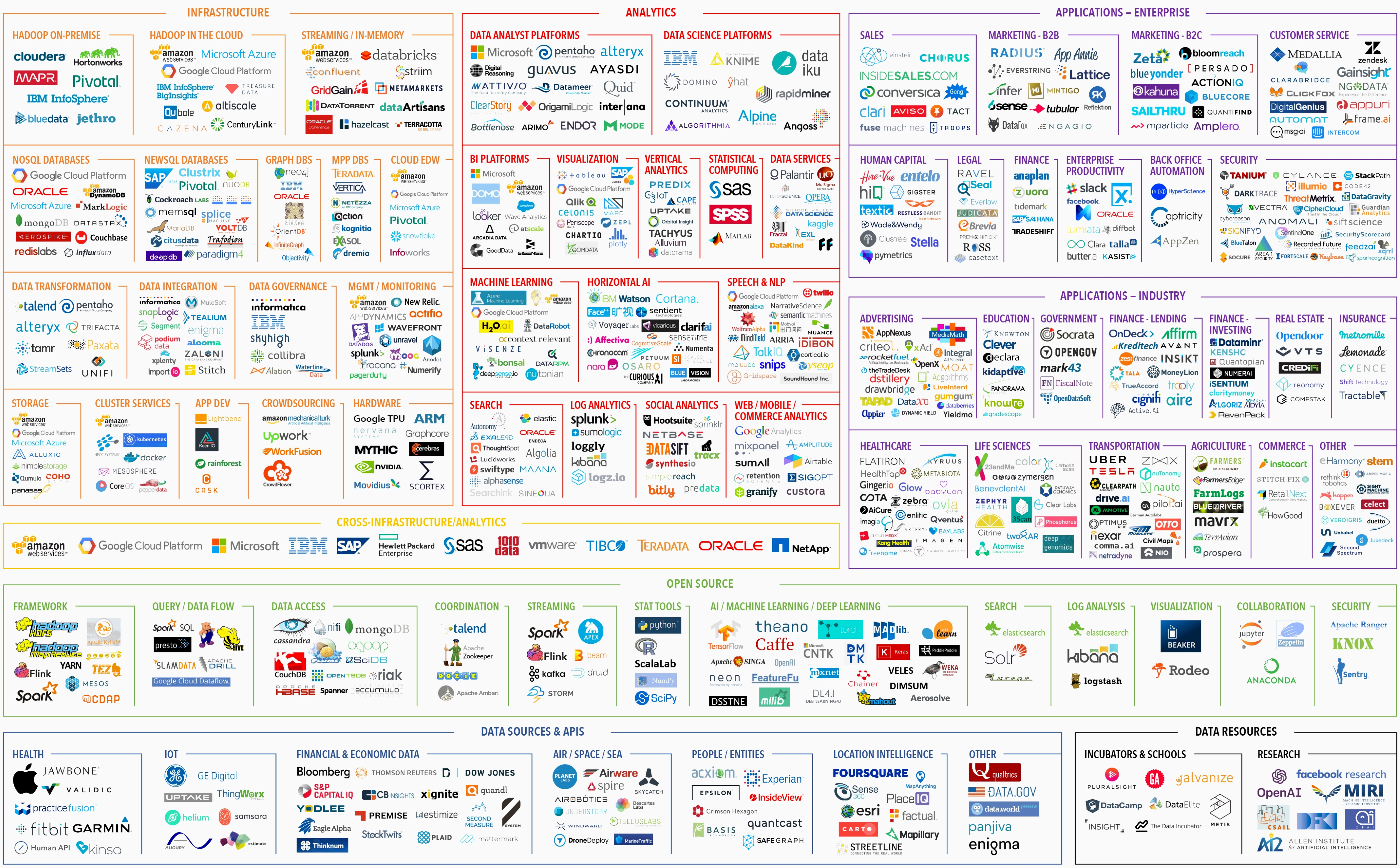
— One Size Does Not Fit All —

Big Data Landscape 2014

dave@vcdave.com

blogs.forbes.com/davefeinleib

Big Data Landscape 2017



FACT 2

— Zoo of Systems —

Current Data Analytics

SPJ Queries

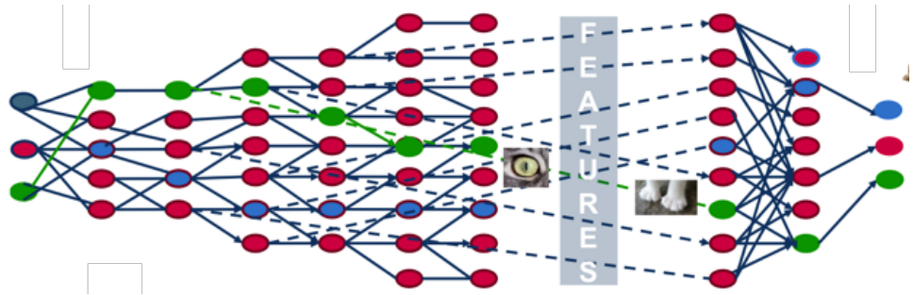
```
SELECT name
FROM employee e
INNER JOIN Person p
ON p.firstname = e.name
```

Current Data Analytics

SPJ Queries

```
SELECT name
FROM employee e
INNER JOIN Person p
ON p.firstname = e.name
```

Deep Learning

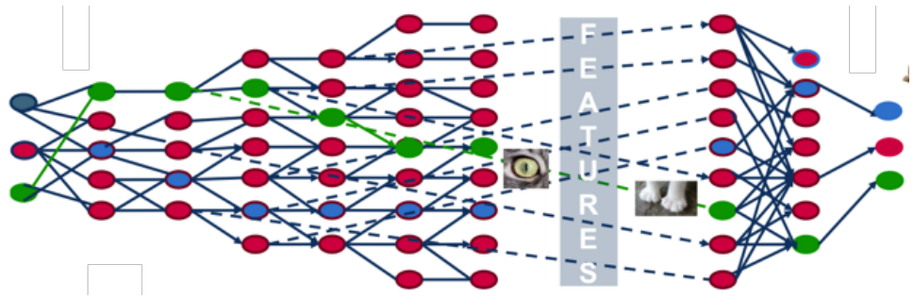


Current Data Analytics

SPJ Queries

```
SELECT name
FROM employee e
INNER JOIN Person p
ON p.firstname = e.name
```

Deep Learning



Relational Data

Activity Code	Activity Name
23	Patching
24	Overlay
25	Crack Sealing

Key = 24

Activity Code	Date	Route No.
24	01/12/01	I-95
24	02/08/01	I-66

Date	Activity Code	Route No.
01/12/01	24	I-95
01/15/01	23	I-495
02/08/01	24	I-66

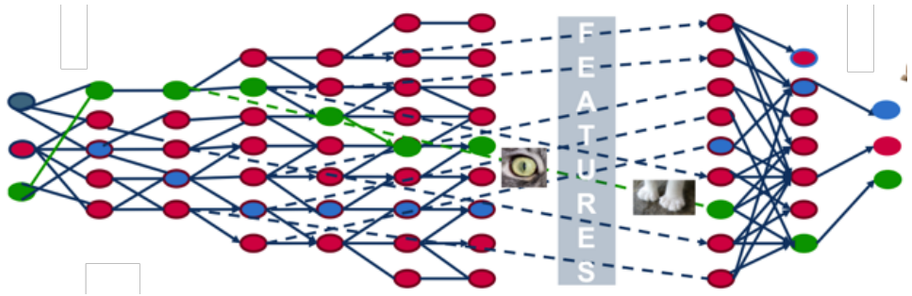
Graph Data

Current Data Analytics

SPJ Queries

```
SELECT name
FROM employee e
INNER JOIN Person p
ON p.firstname = e.name
```

Deep Learning



— Queries Beyond a Single Platform —

Q1: unstructured + structured data processing

Q2: Join Postgres data with Spark data

Q3: Run machine learning tasks over graph data

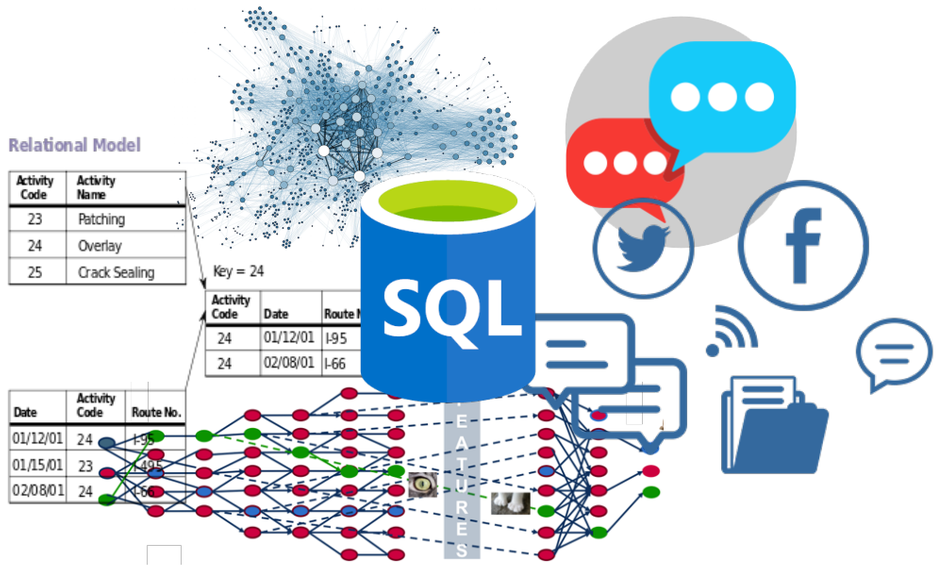
Code	Date	Route No.
24	01/12/01	I-95
24	02/08/01	I-66

Date	Activity Code	Route No.
01/12/01	24	I-95
01/15/01	23	I-495
02/08/01	24	I-66

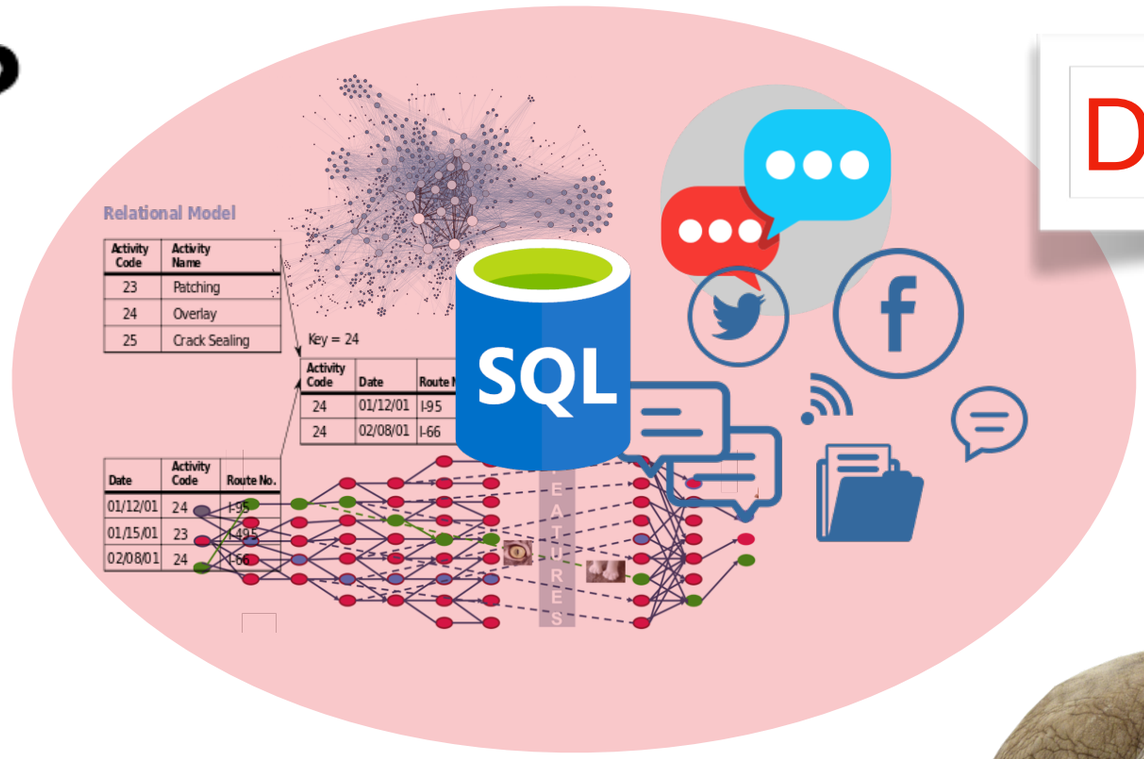
FACT 3

— Variety of Data Analytics —

What to Do?

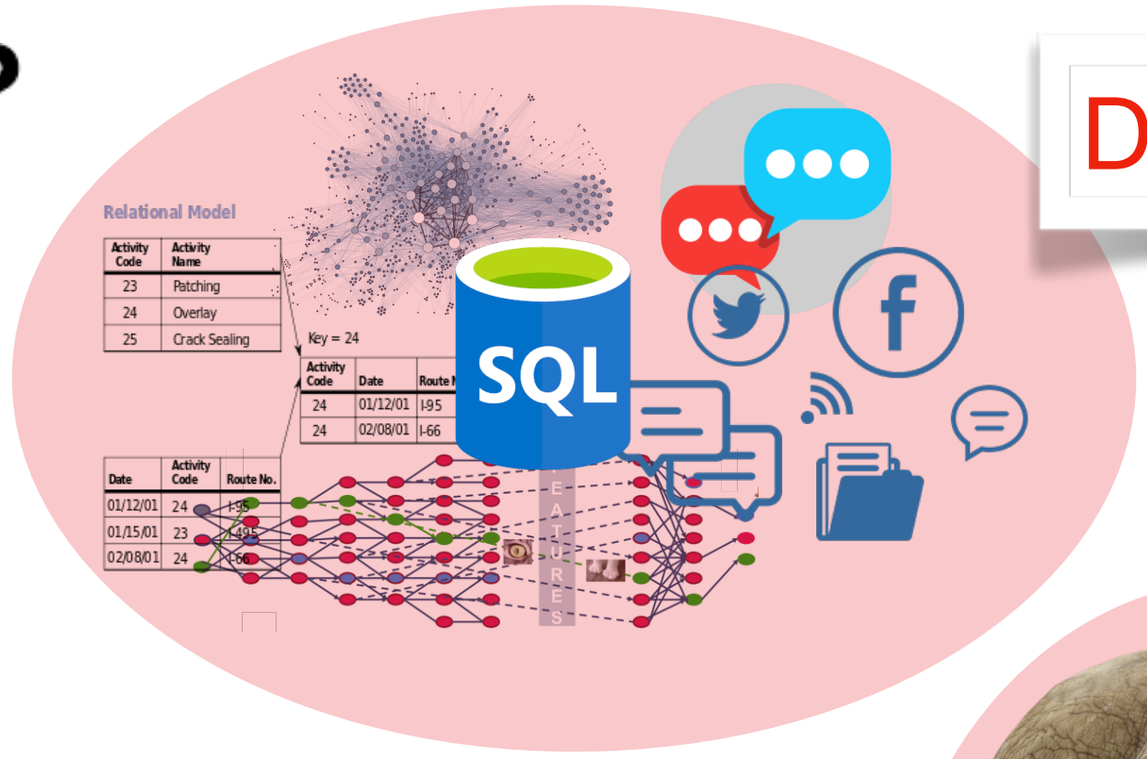

Relational Model
Activity Code
Activity Name
23
Patching
24
Overlay
25
Crack Sealing
Key = 24
Activity Code
Date
24
01/12/01
I-95
24
02/08/01
I-66
Date
Activity Code
Route No.
01/12/01
24
01/15/01
23
02/08/01
24
SQL
FEATURES

What to Do?



Diverse Data Analytics

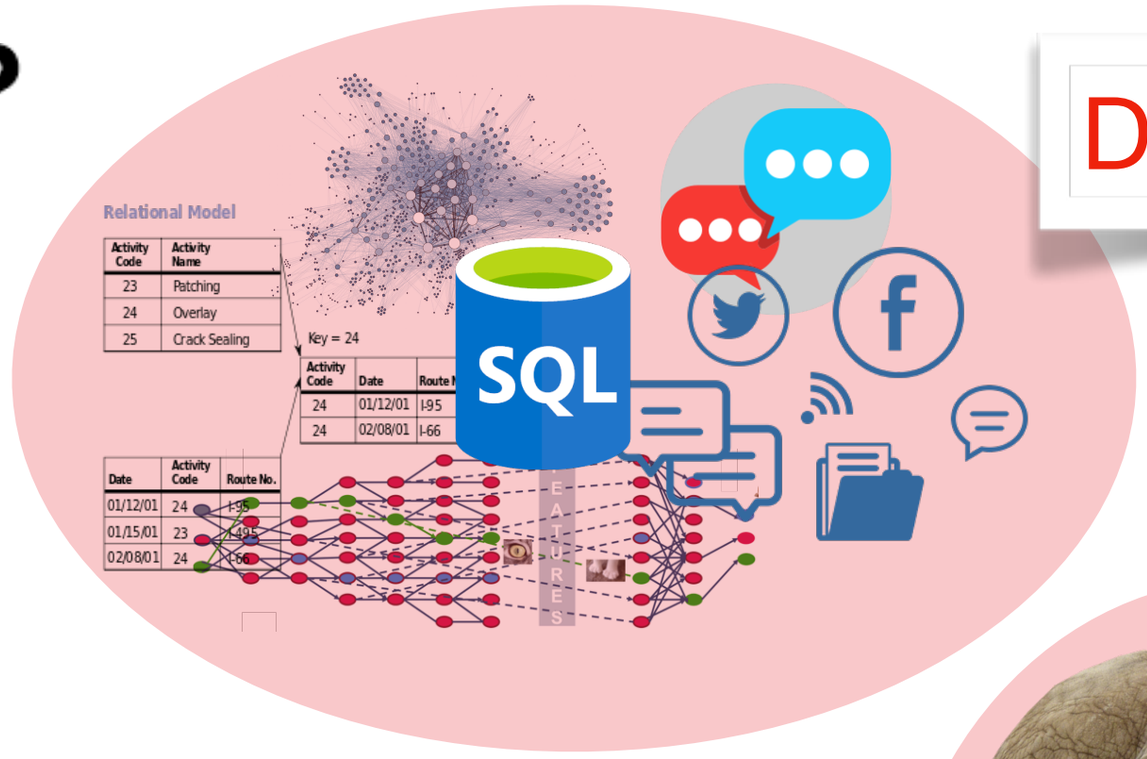
What to Do?



Diverse Data Analytics

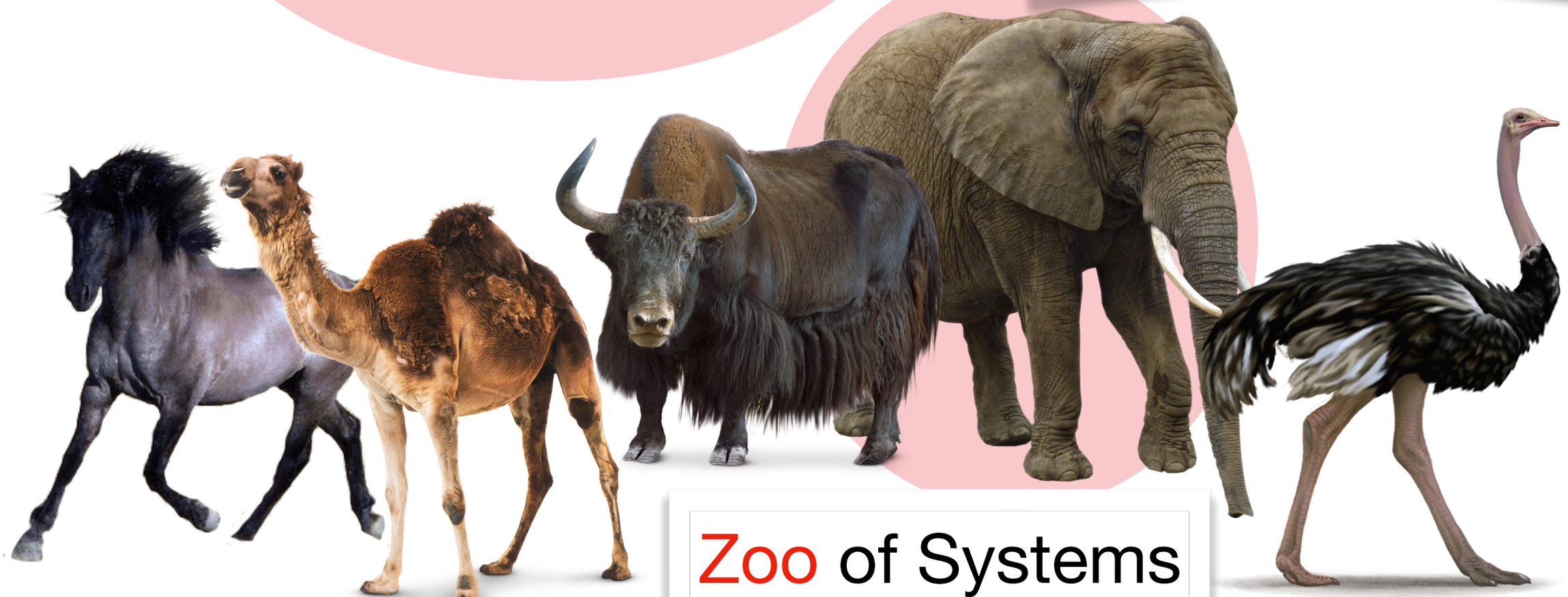
One Cannot Fit All

What to Do?



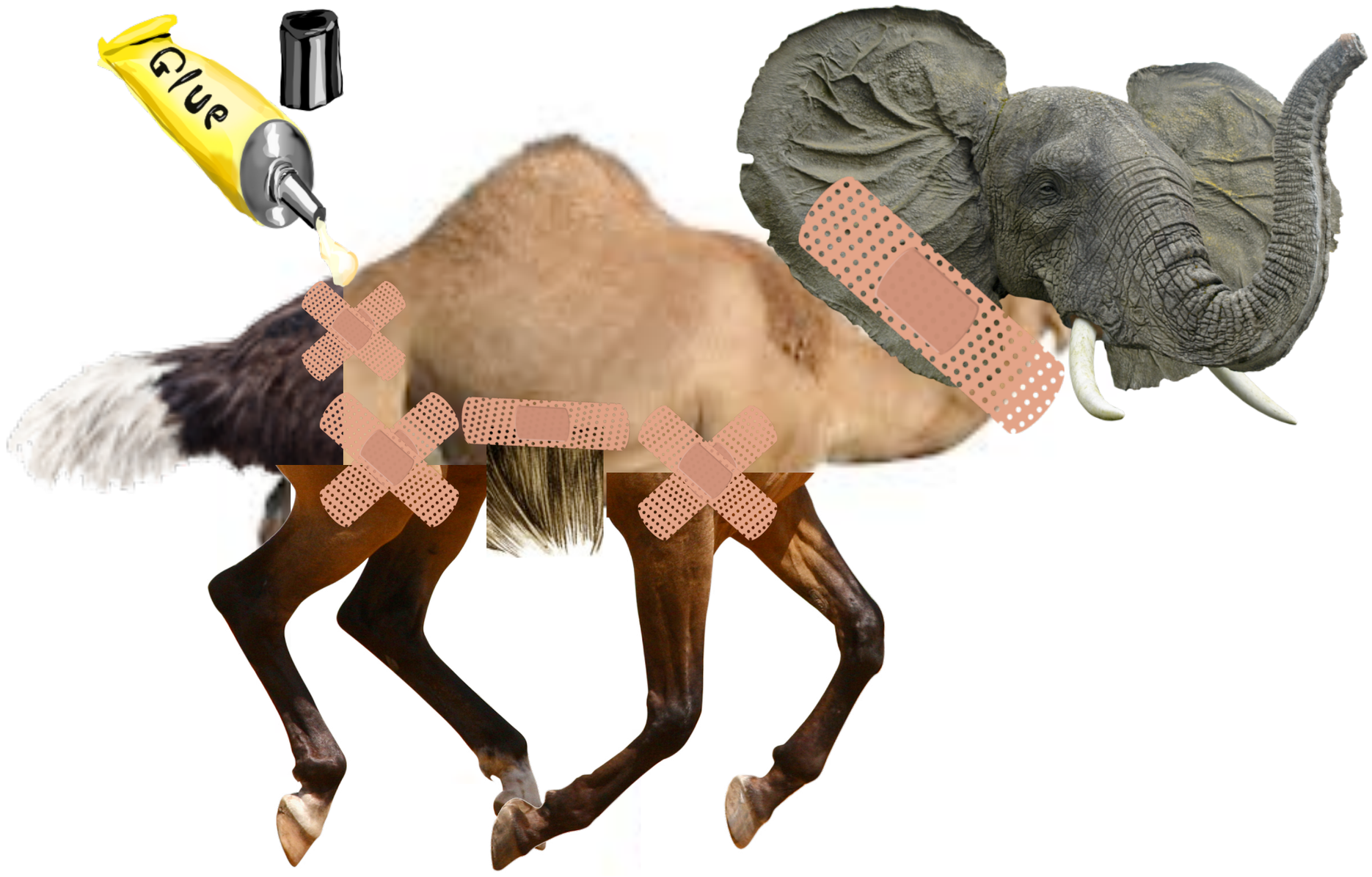
Diverse Data Analytics

One Cannot Fit All

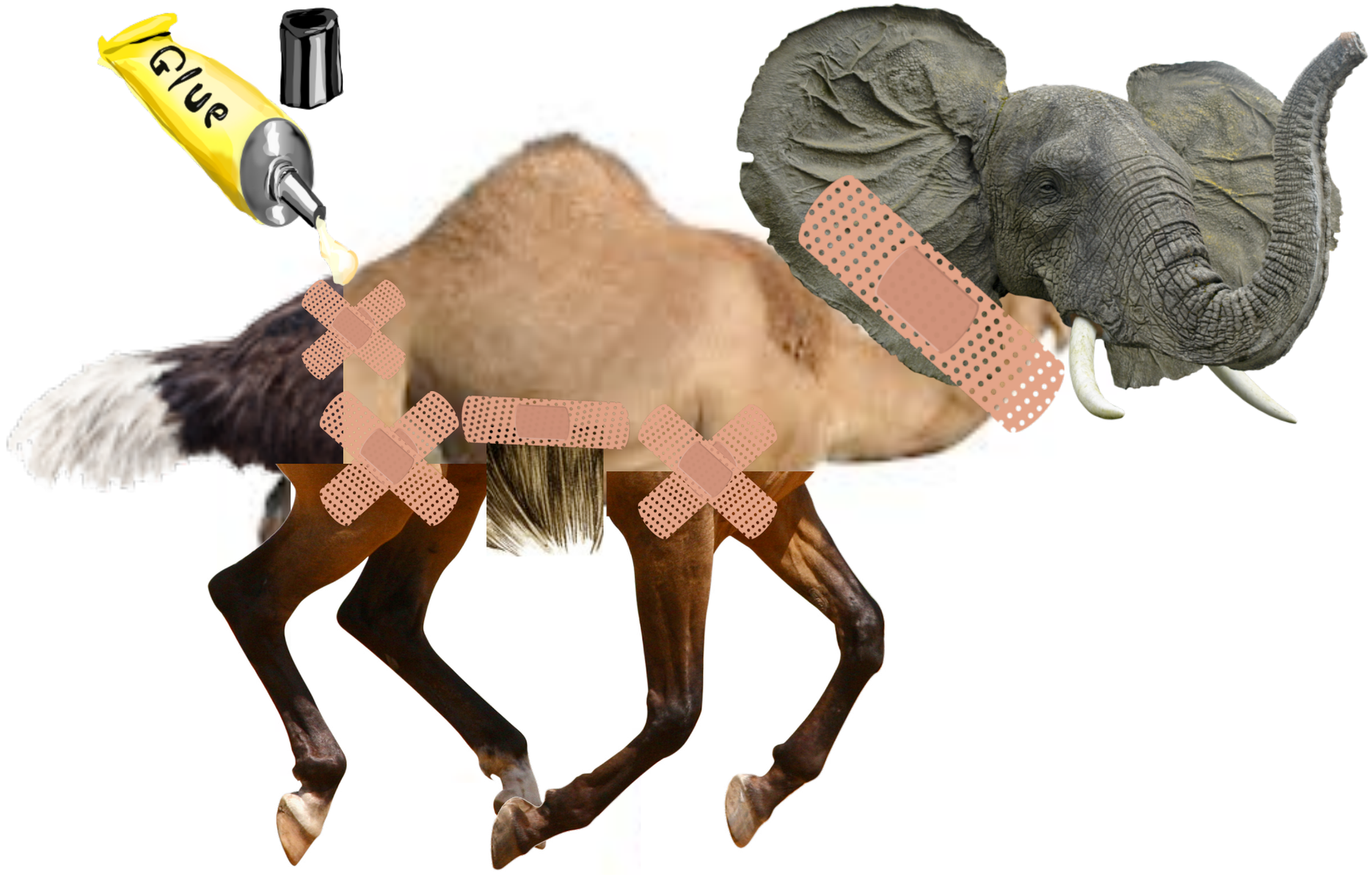


Zoo of Systems

Hybrid Systems

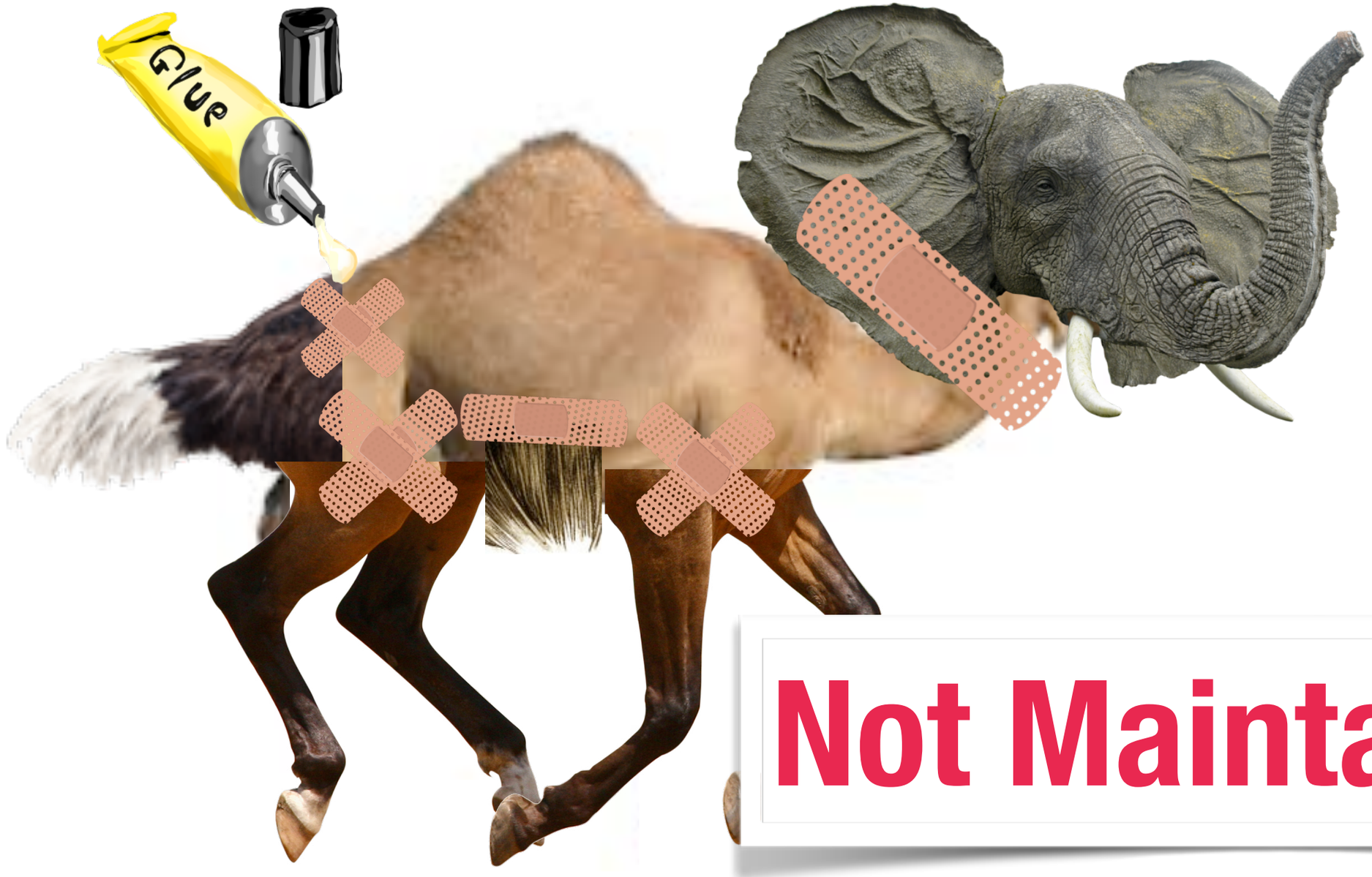


Hybrid Systems



Examples: HadoopDB, SystemML, FlumeJava …

Hybrid Systems

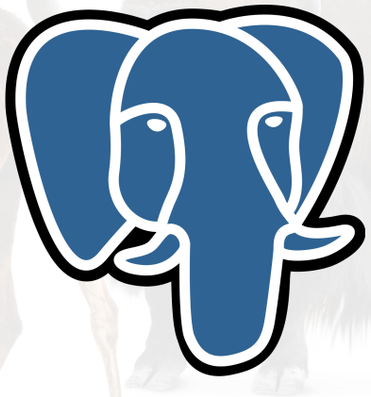


Not Maintainable!

Examples: HadoopDB, SystemML, FlumeJava …

Taming Them to Live Together

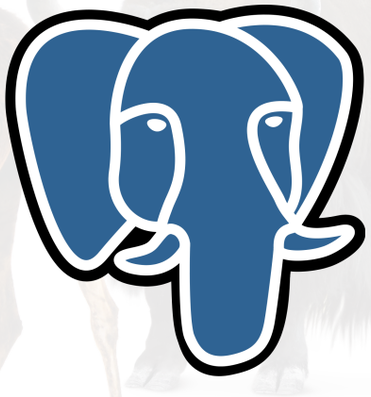
Taming Them to Live Together



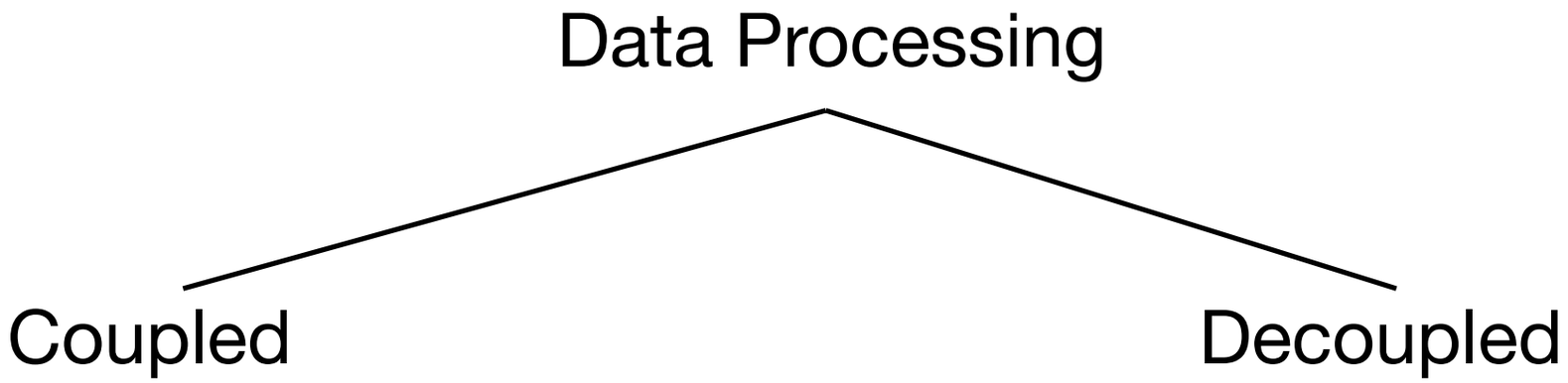
Taming Them to Live Together

use of multiple data processing platforms for query processing (single run or across several runs)

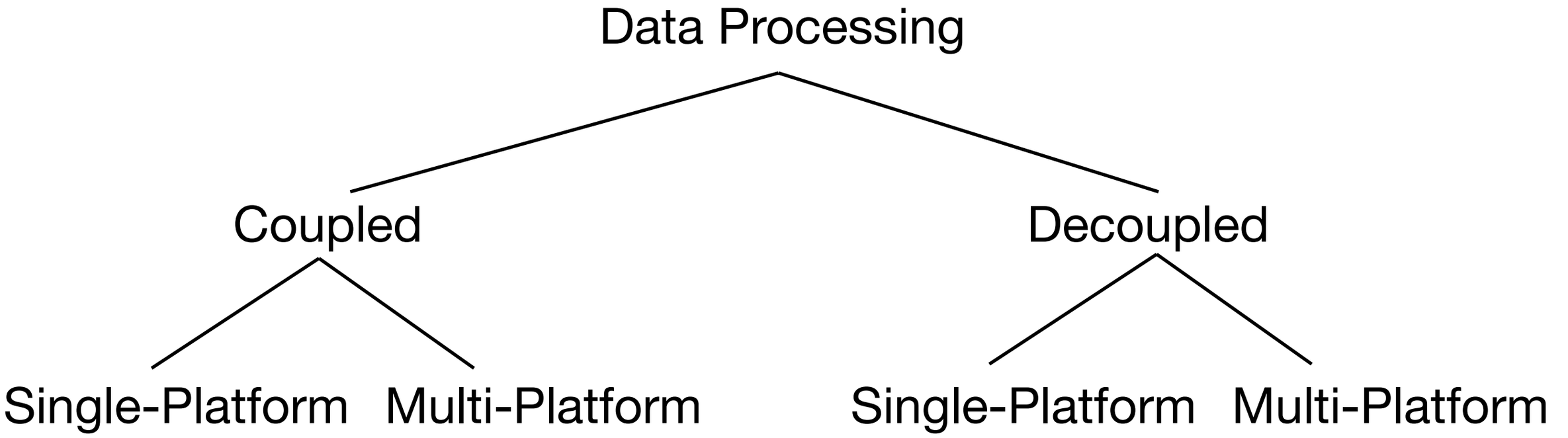
Cross-Platform Data Processing



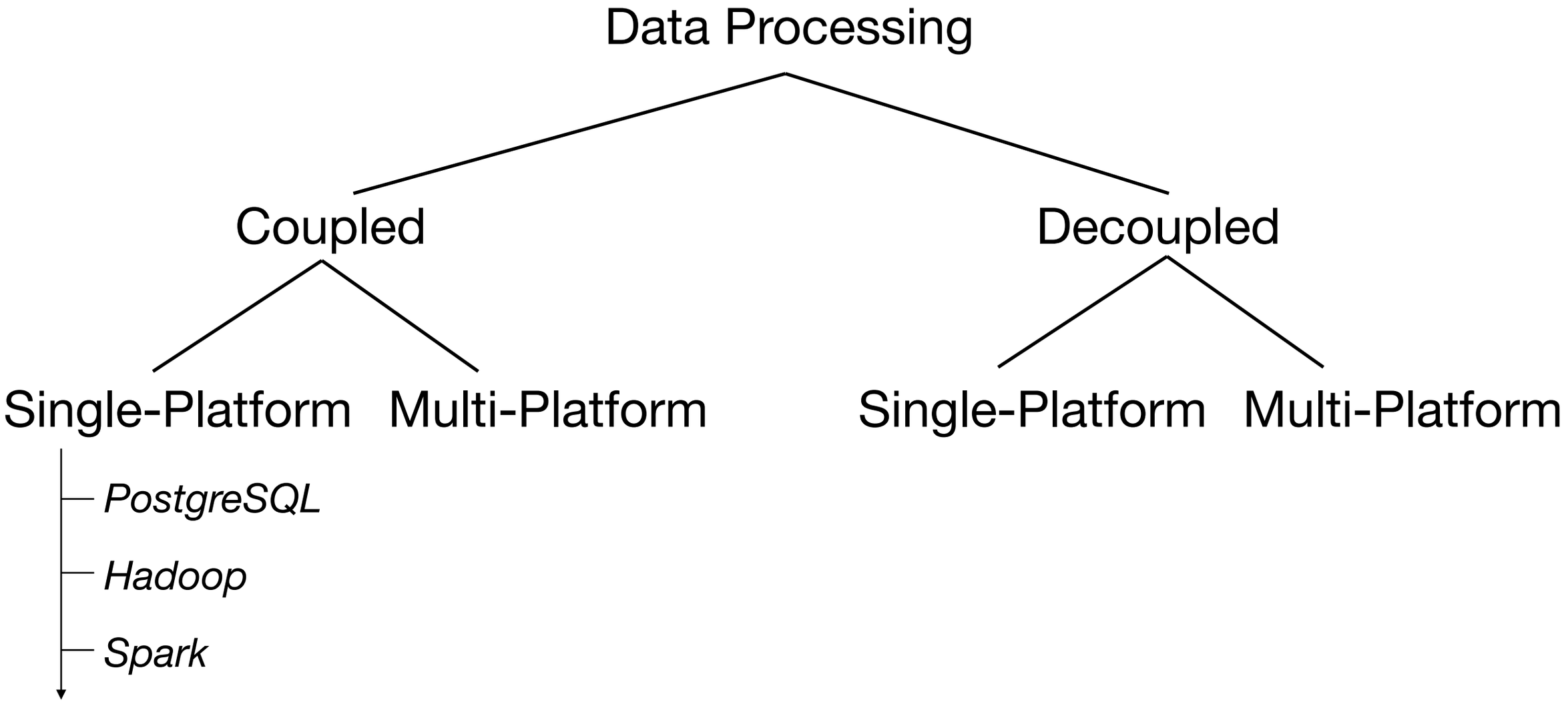
Data Processing Taxonomy



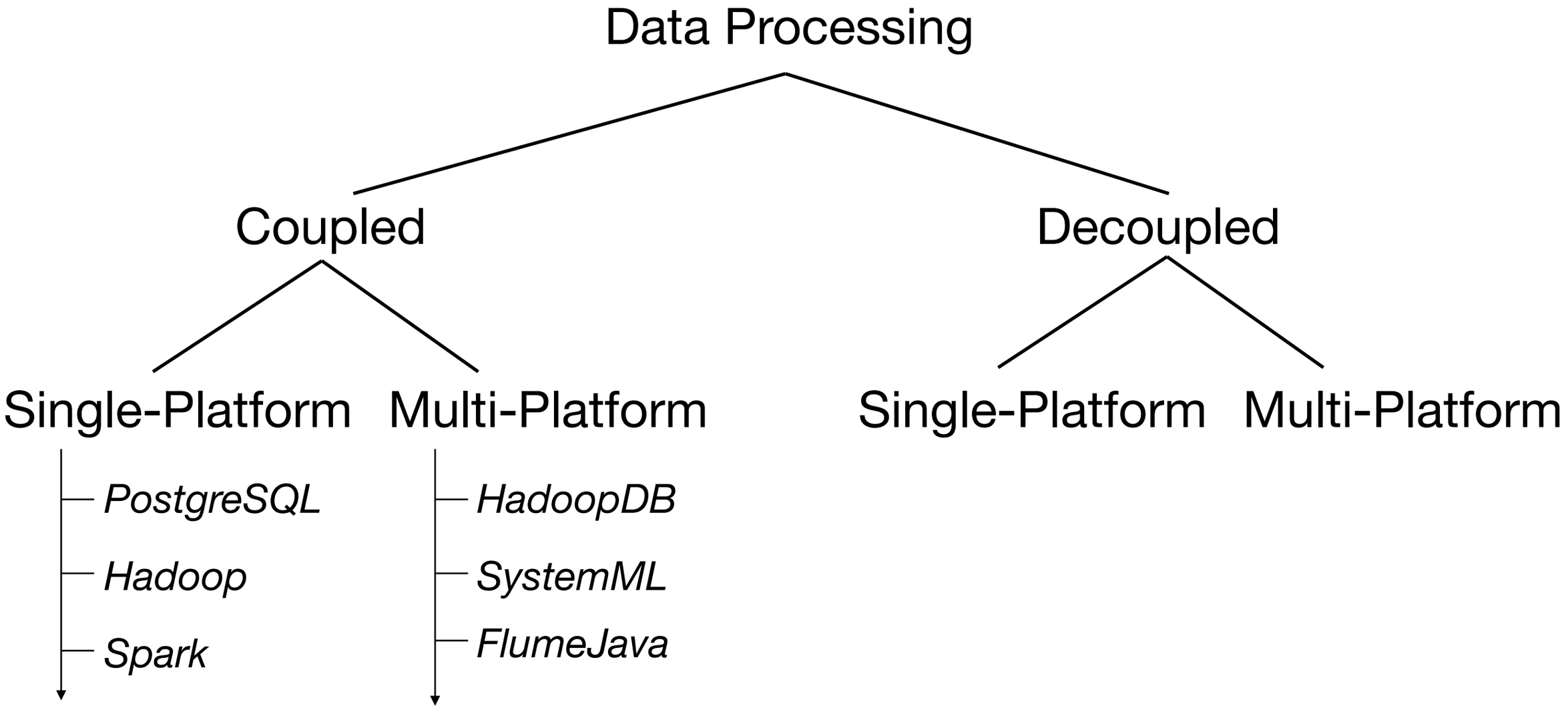
Data Processing Taxonomy



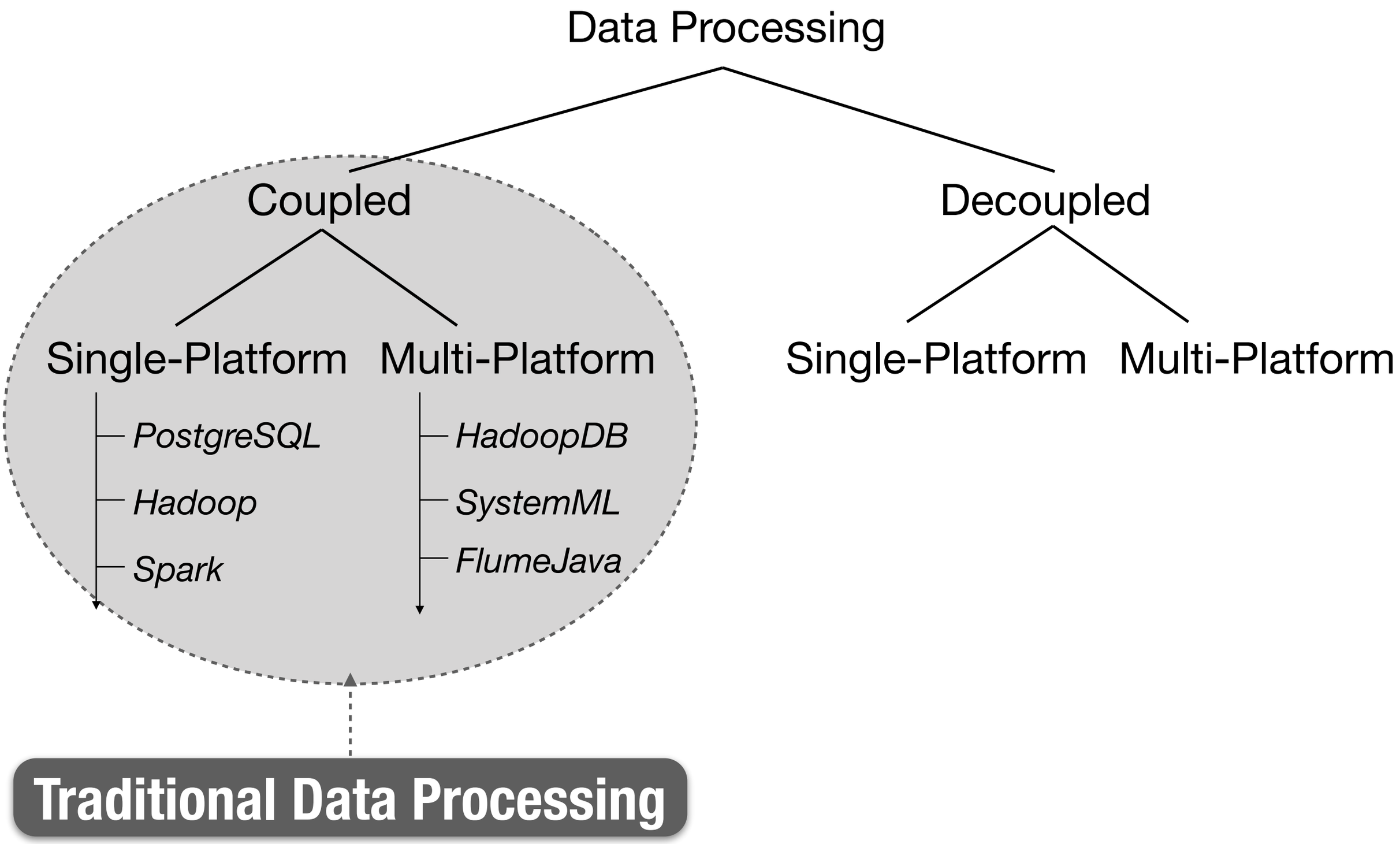
Data Processing Taxonomy



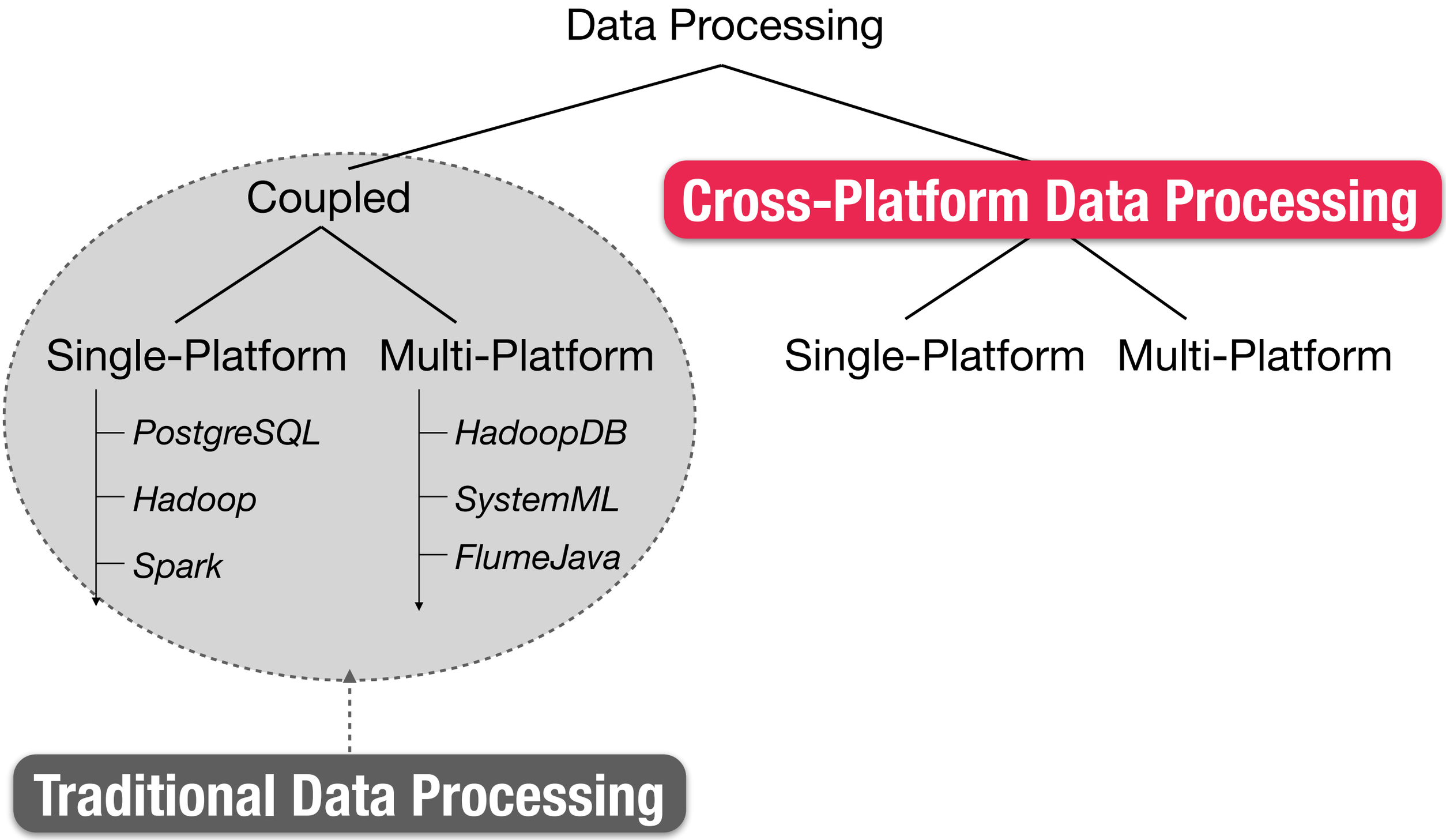
Data Processing Taxonomy



Data Processing Taxonomy



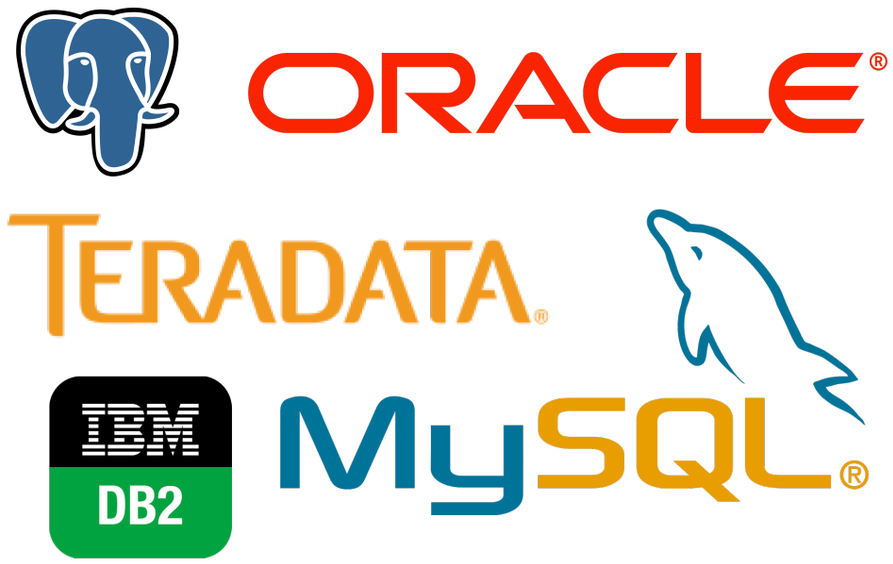
Data Processing Taxonomy



Federated DBs vs Cross-Platform

Single Data Model

Federated DB System



Disparate Data Models

Cross-Platform System

Cross-Platform Data Processing

Motivation

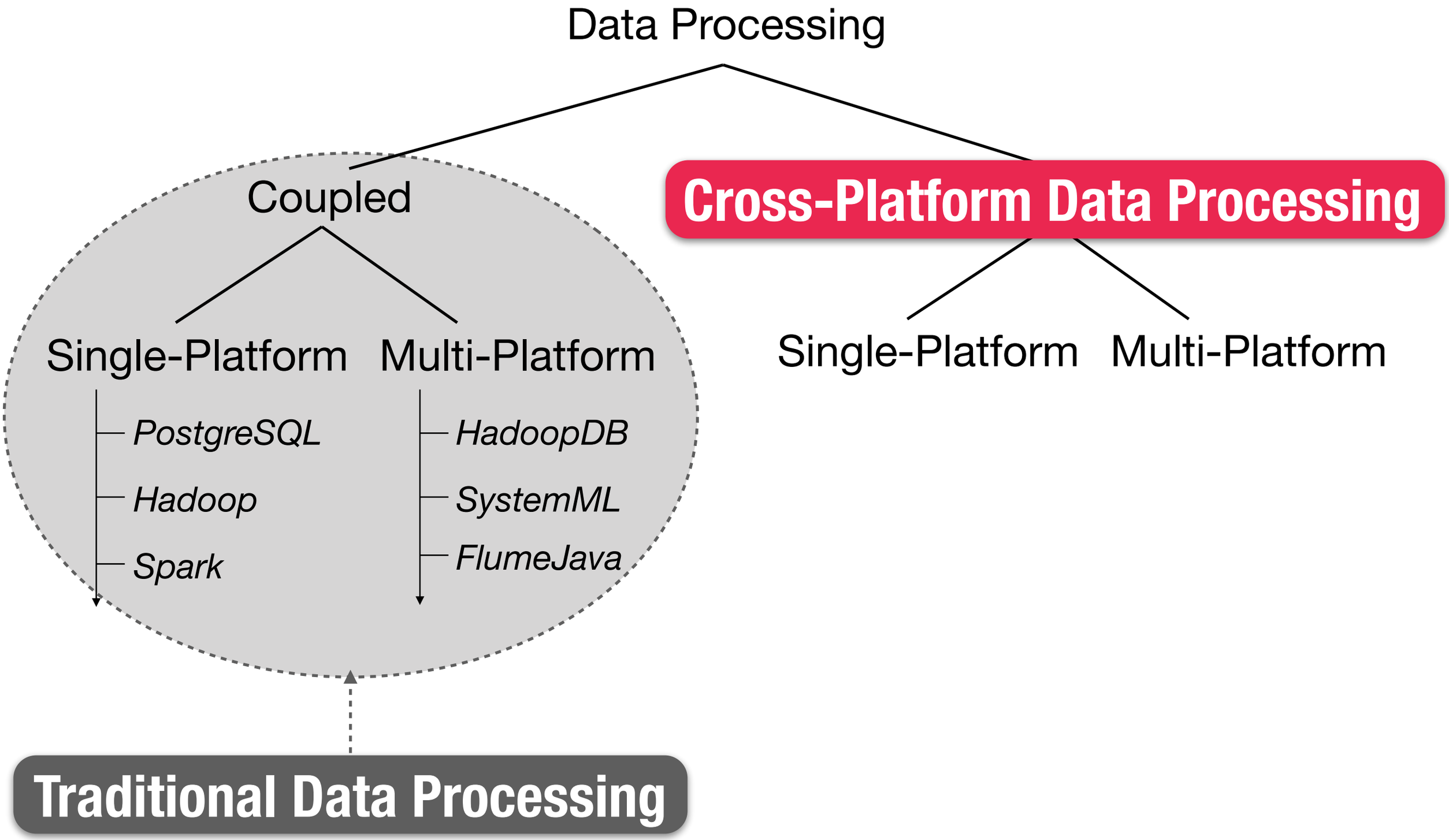
Use Cases

Challenges

Current Efforts

Summary

Data Processing Taxonomy



Cross-Platform Use Cases

Data Processing

Cross-Platform Data Processing

Single-Platform

Multi-Platform

Cross-Platform Use Cases

Data Processing

Cross-Platform Data Processing

Decoupled Single-Platform

processing a query on ***any single*** data processing platform

Multi-Platform

Cross-Platform Use Cases

Data Processing

Cross-Platform Data Processing

Decoupled Single-Platform

Multi-Platform

processing ***one single*** query on
several data processing platforms…

Cross-Platform Use Cases

Data Processing

Cross-Platform Data Processing

Decoupled Single-Platform

Multi-Platform

processing ***one single*** query on ***several*** data processing platforms…

Opportunistic

… to reduce the total cost of the input query

Cross-Platform Use Cases

Data Processing

Cross-Platform Data Processing

Decoupled Single-Platform

Multi-Platform

processing ***one single*** query on ***several*** data processing platforms…

Opportunistic

Mandatory

… because the platform where the data is cannot run it entirely

Cross-Platform Use Cases

Data Processing

Cross-Platform Data Processing

Decoupled Single-Platform

Multi-Platform

processing ***one single*** query on ***several*** data processing platforms…

Opportunistic

Mandatory

Polystore

… because the data is stored across several store engines

Cross-Platform Use Cases

Data Processing

Cross-Platform Data Processing

Decoupled Single-Platform

Multi-Platform

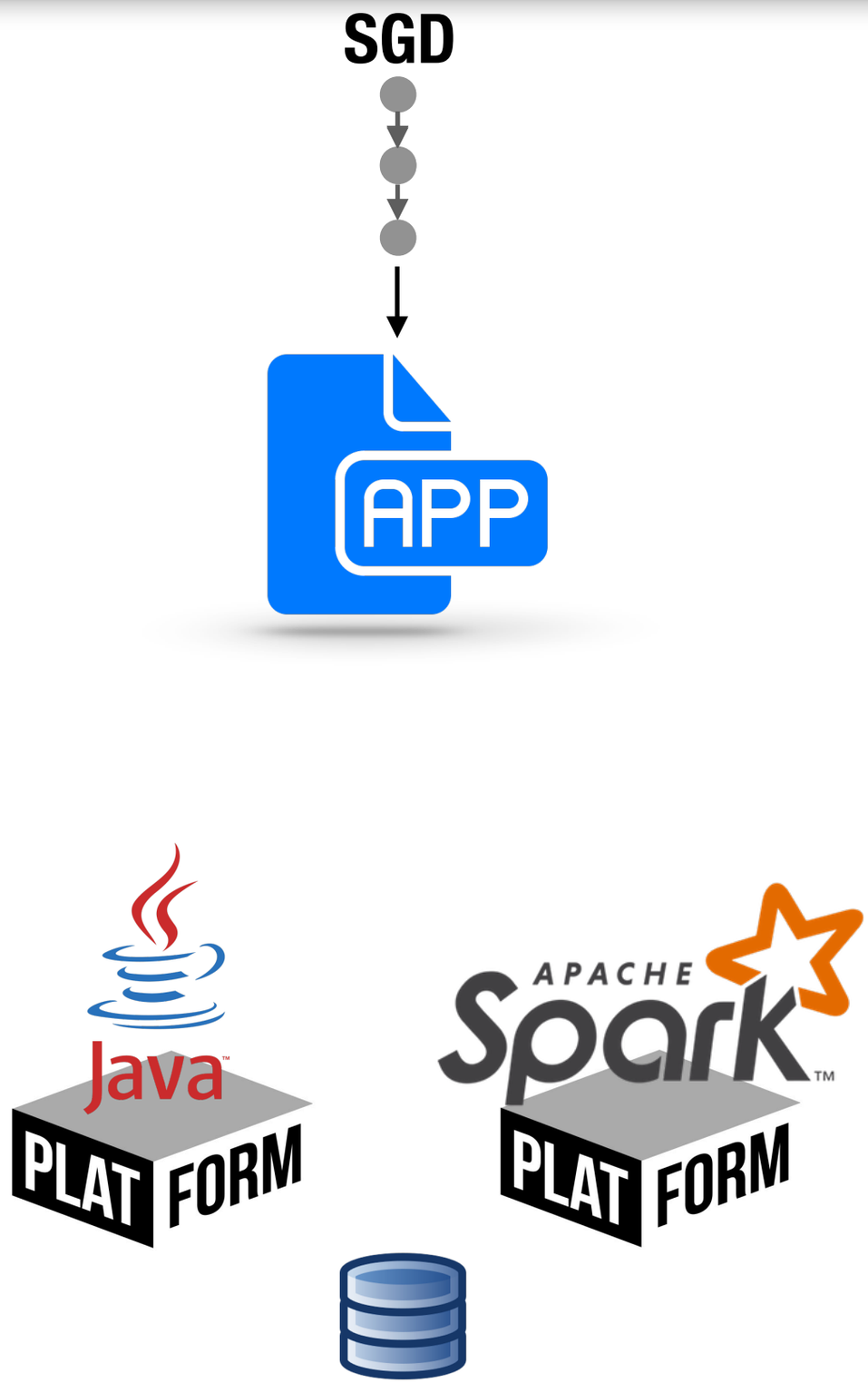
Opportunistic

Mandatory

Polystore

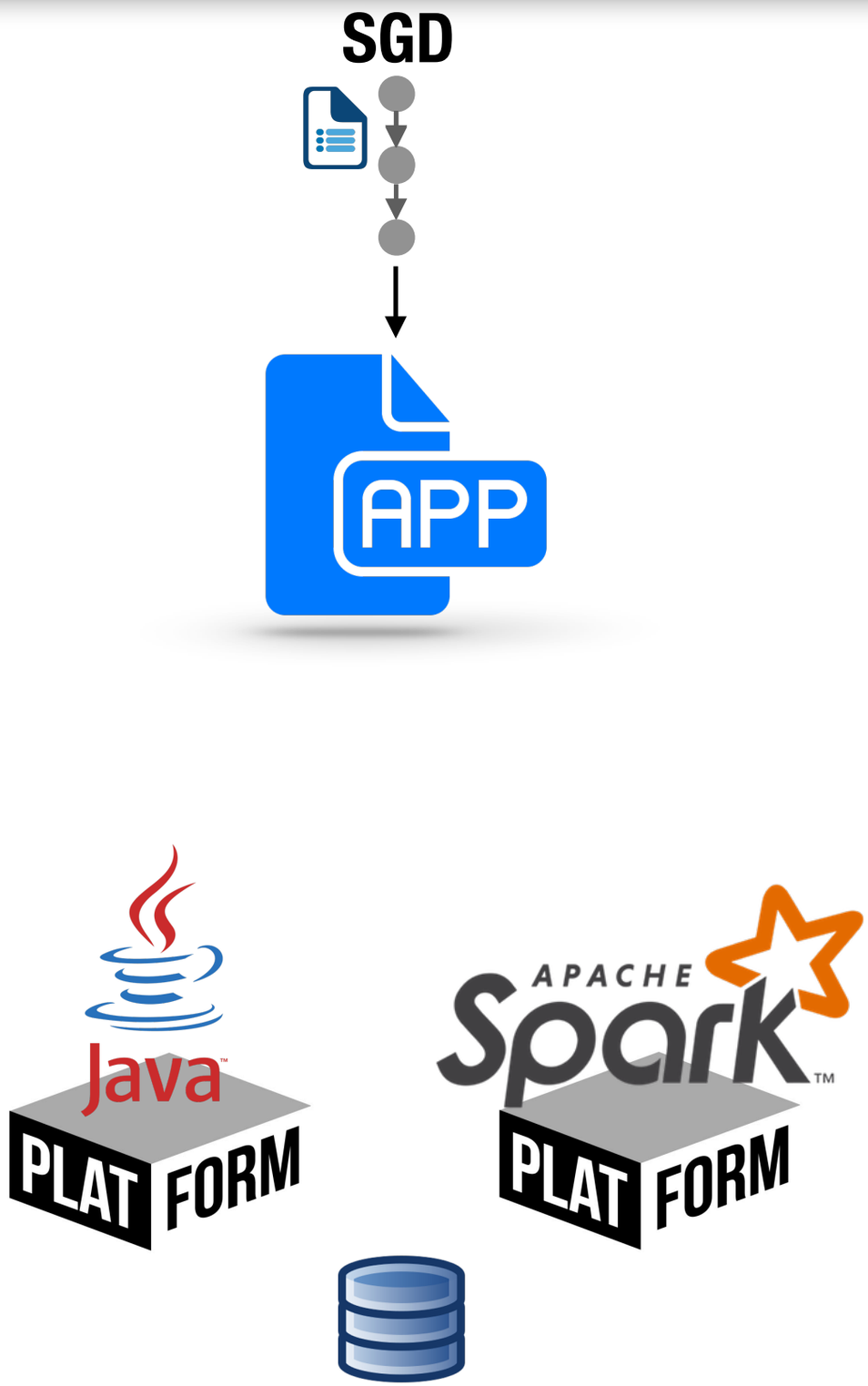
Decoupled Single-Platform

using ***any single*** data processing platform to process a query



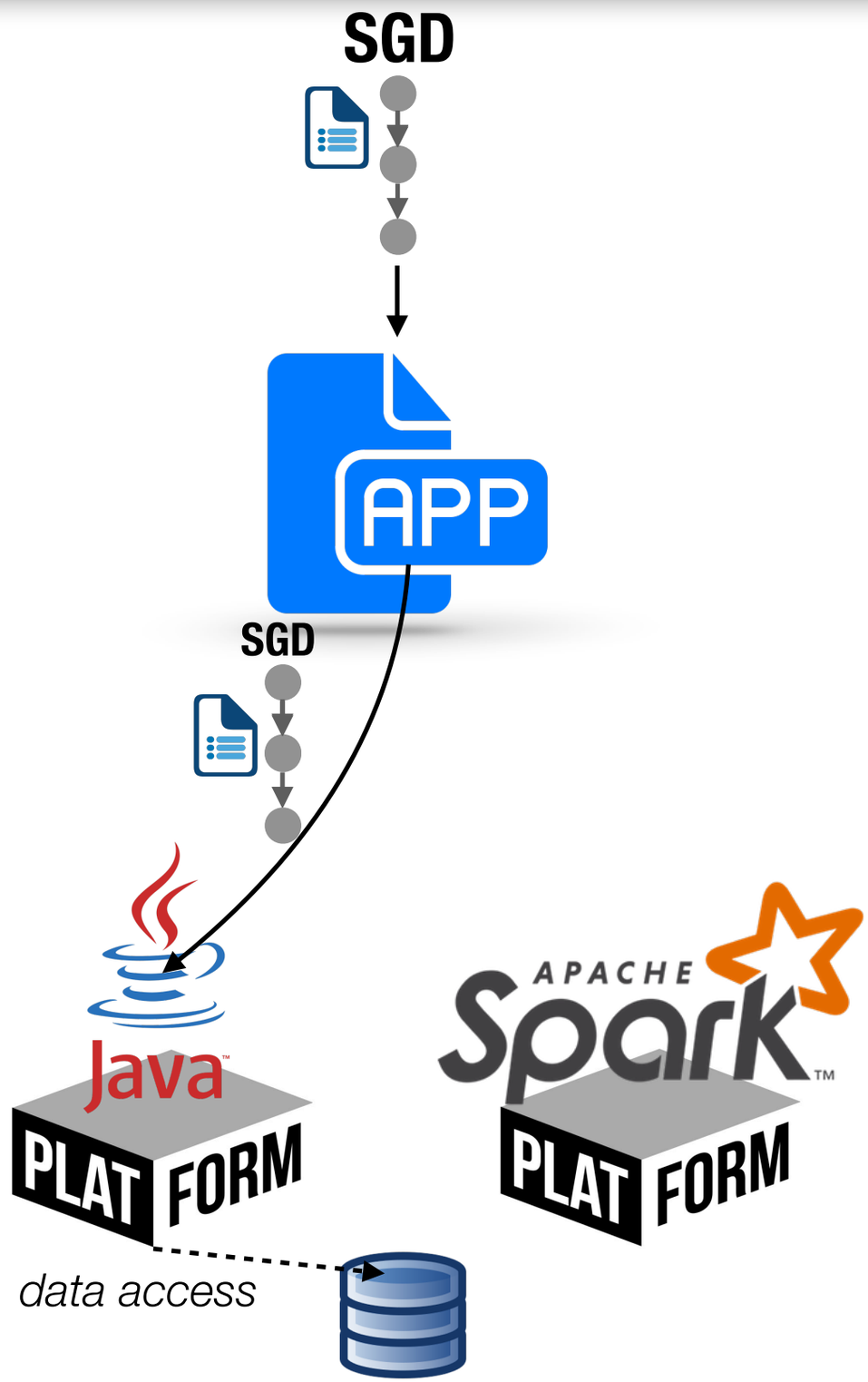
Decoupled Single-Platform

using ***any single*** data processing platform to process a query



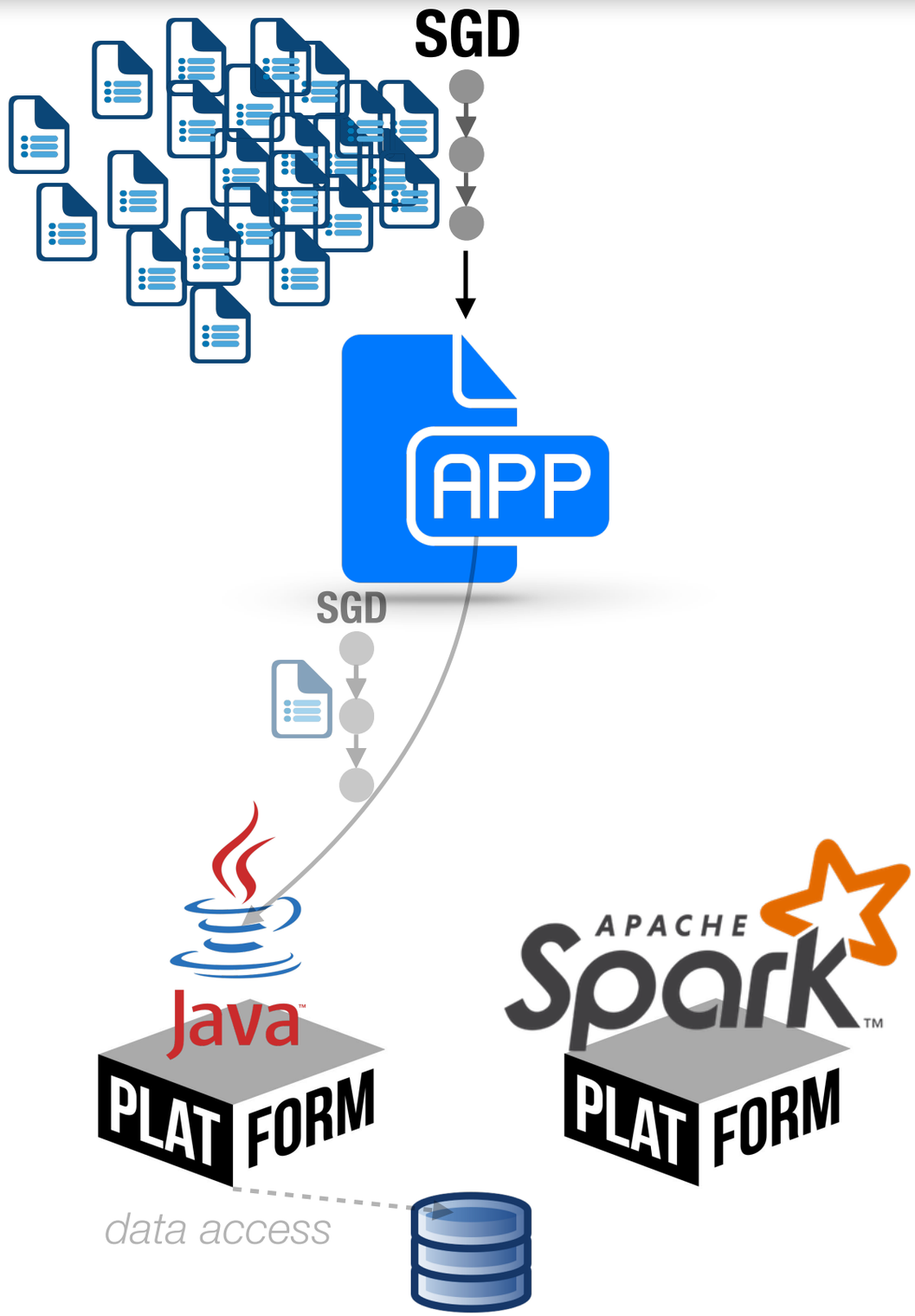
Decoupled Single-Platform

using ***any single*** data processing platform to process a query



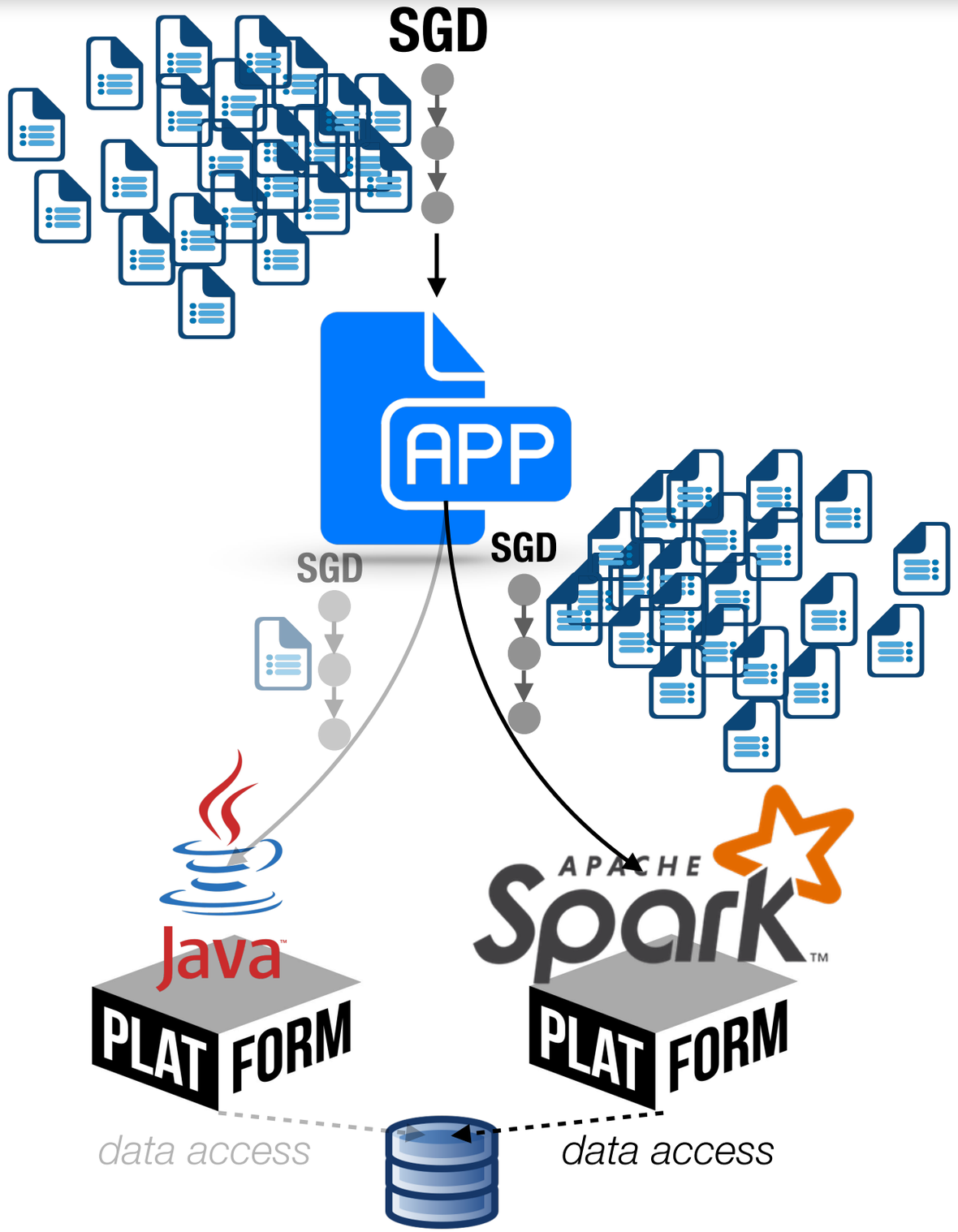
Decoupled Single-Platform

using ***any single*** data processing platform to process a query



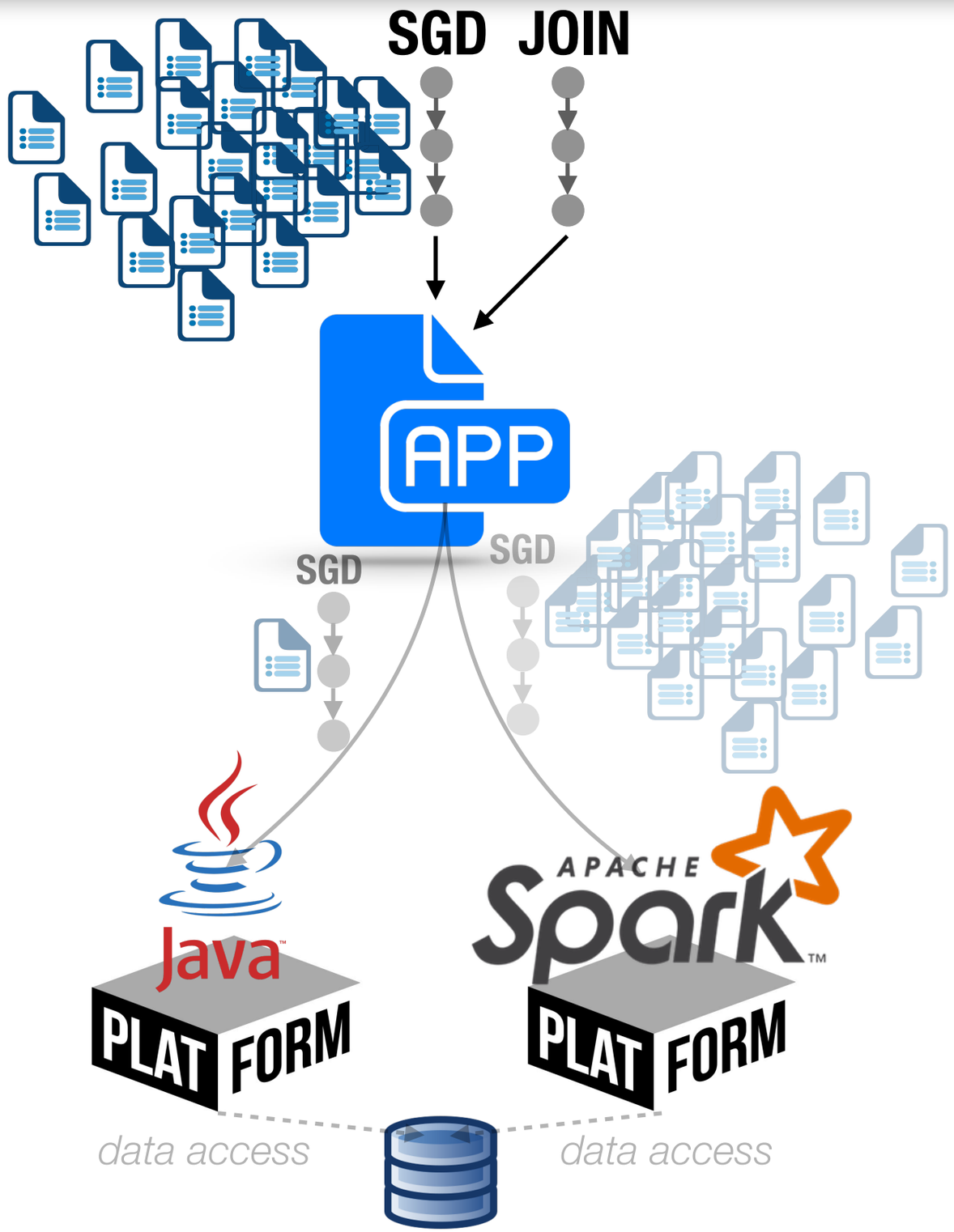
Decoupled Single-Platform

using ***any single*** data processing platform to process a query



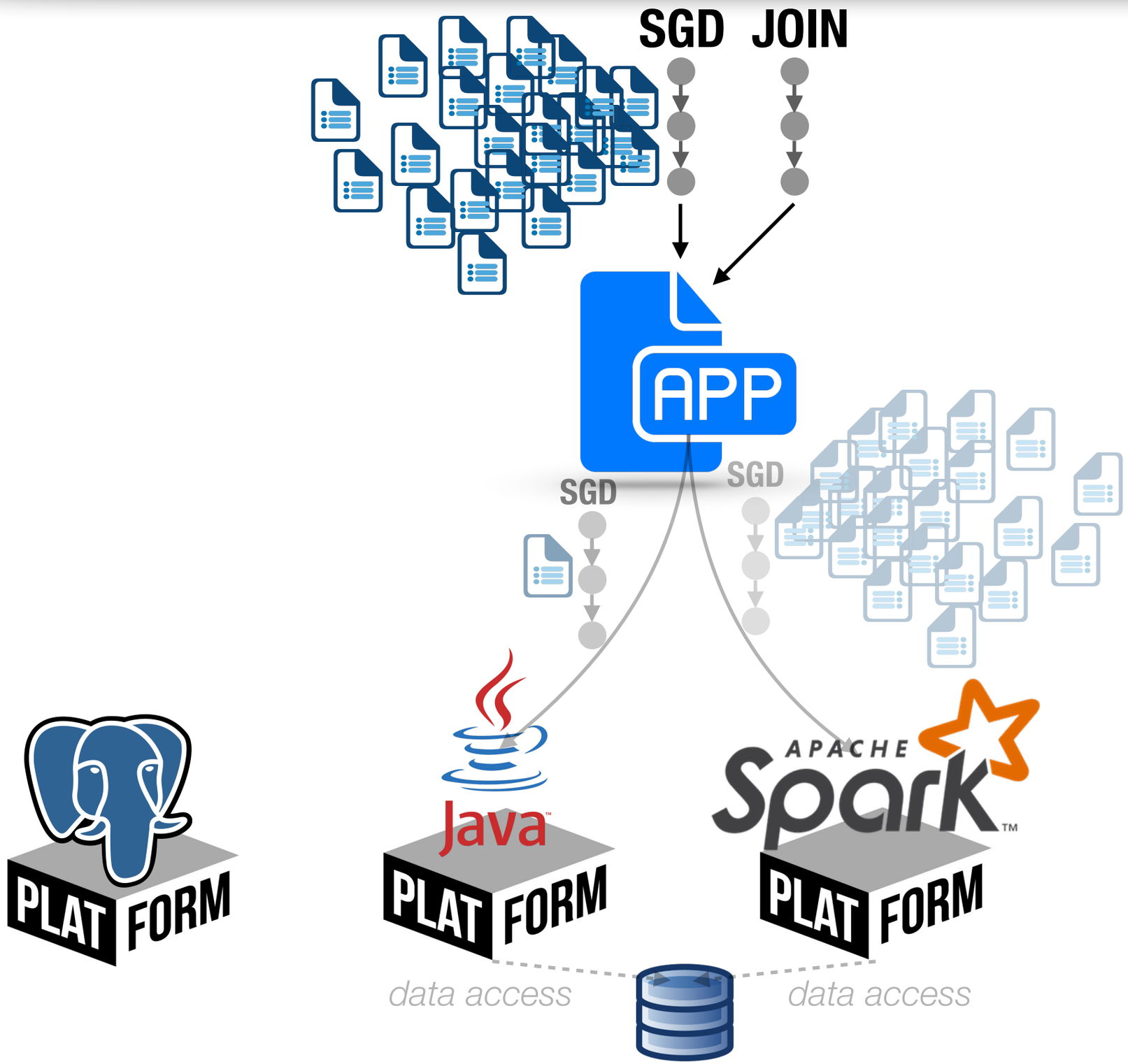
Decoupled Single-Platform

using ***any single*** data processing platform to process a query



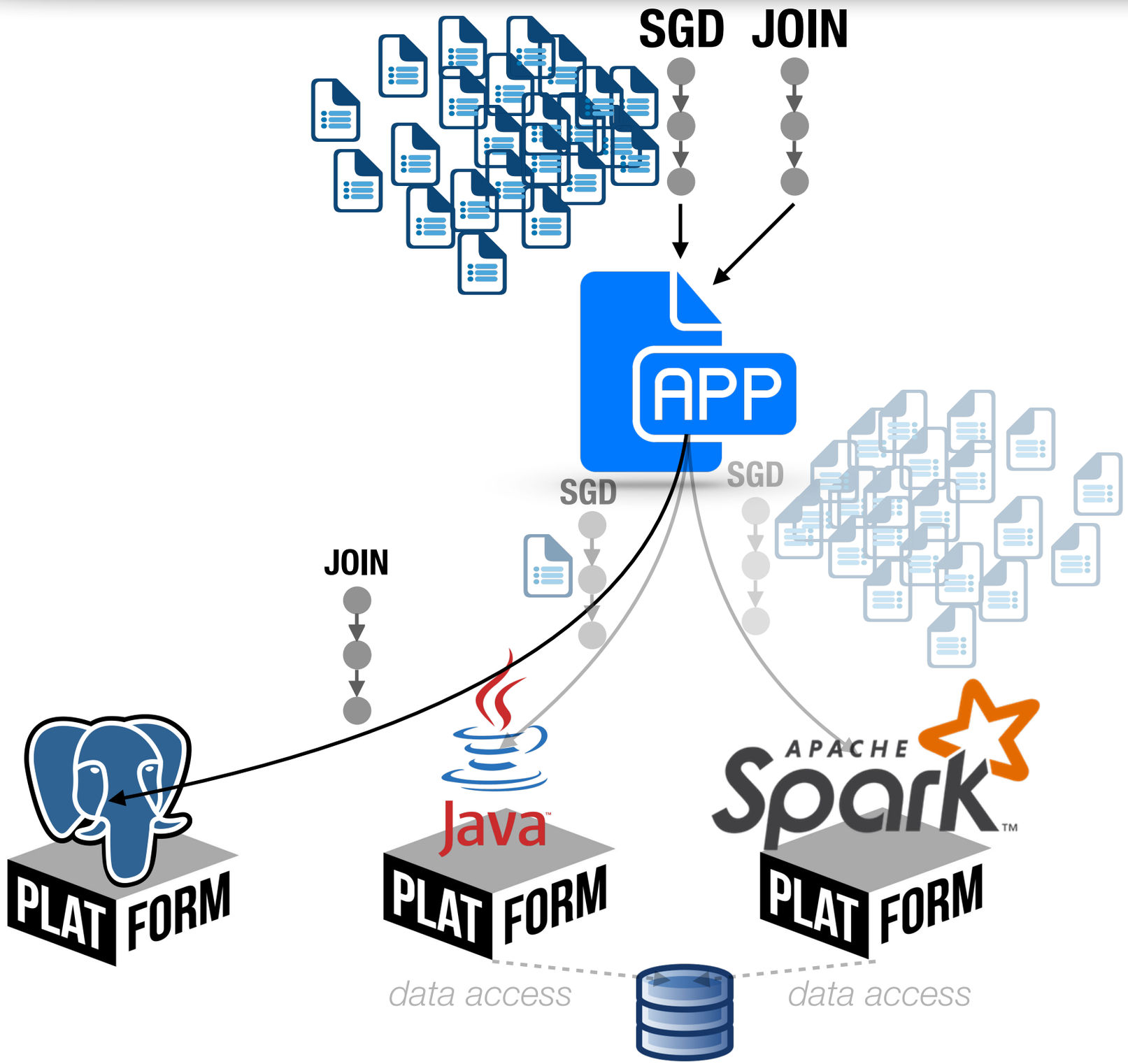
Decoupled Single-Platform

using ***any single*** data processing platform to process a query



Decoupled Single-Platform

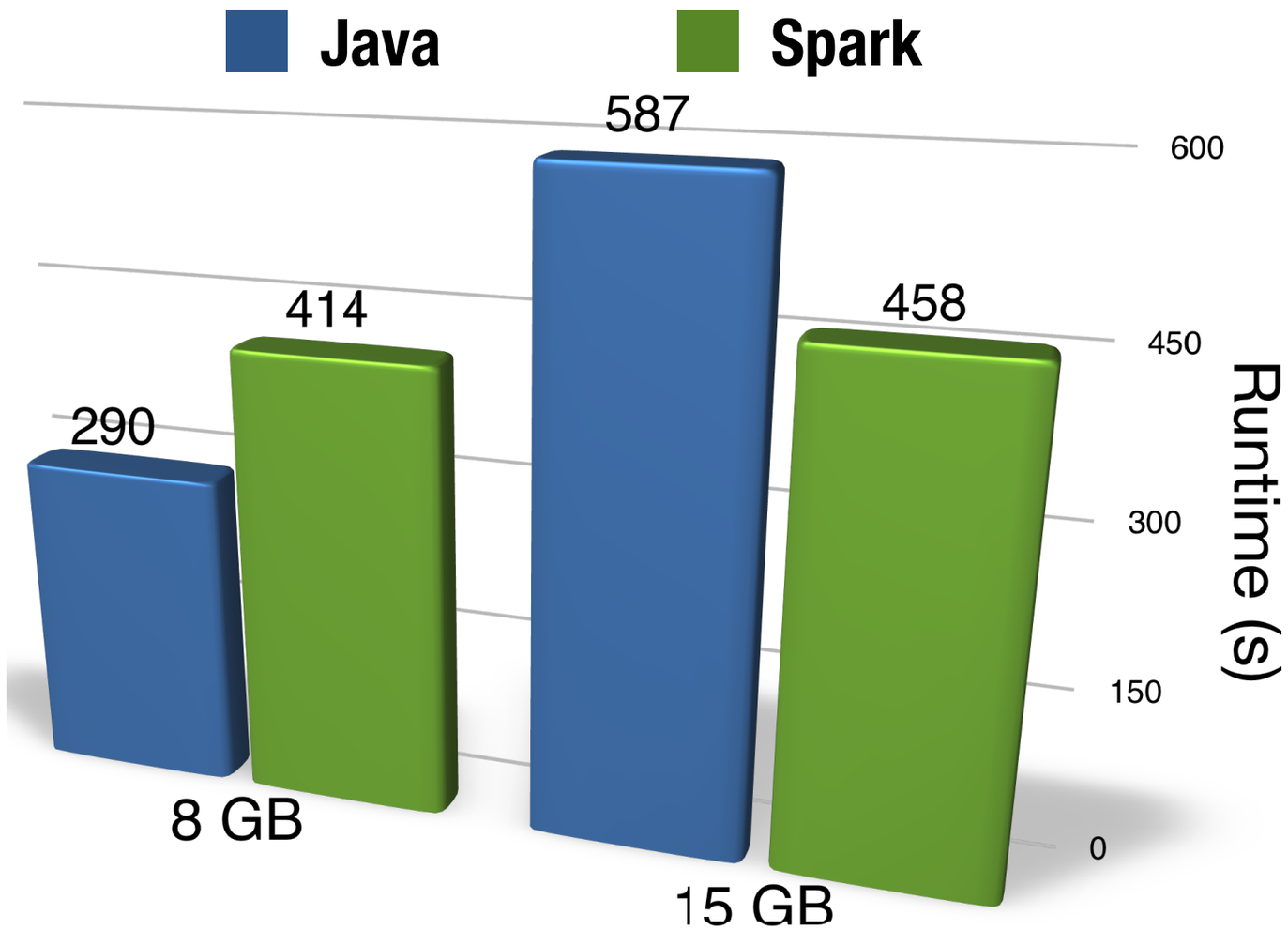
using ***any single*** data processing platform to process a query



Decoupled Single-Platform

using ***any single*** data processing platform to process a query

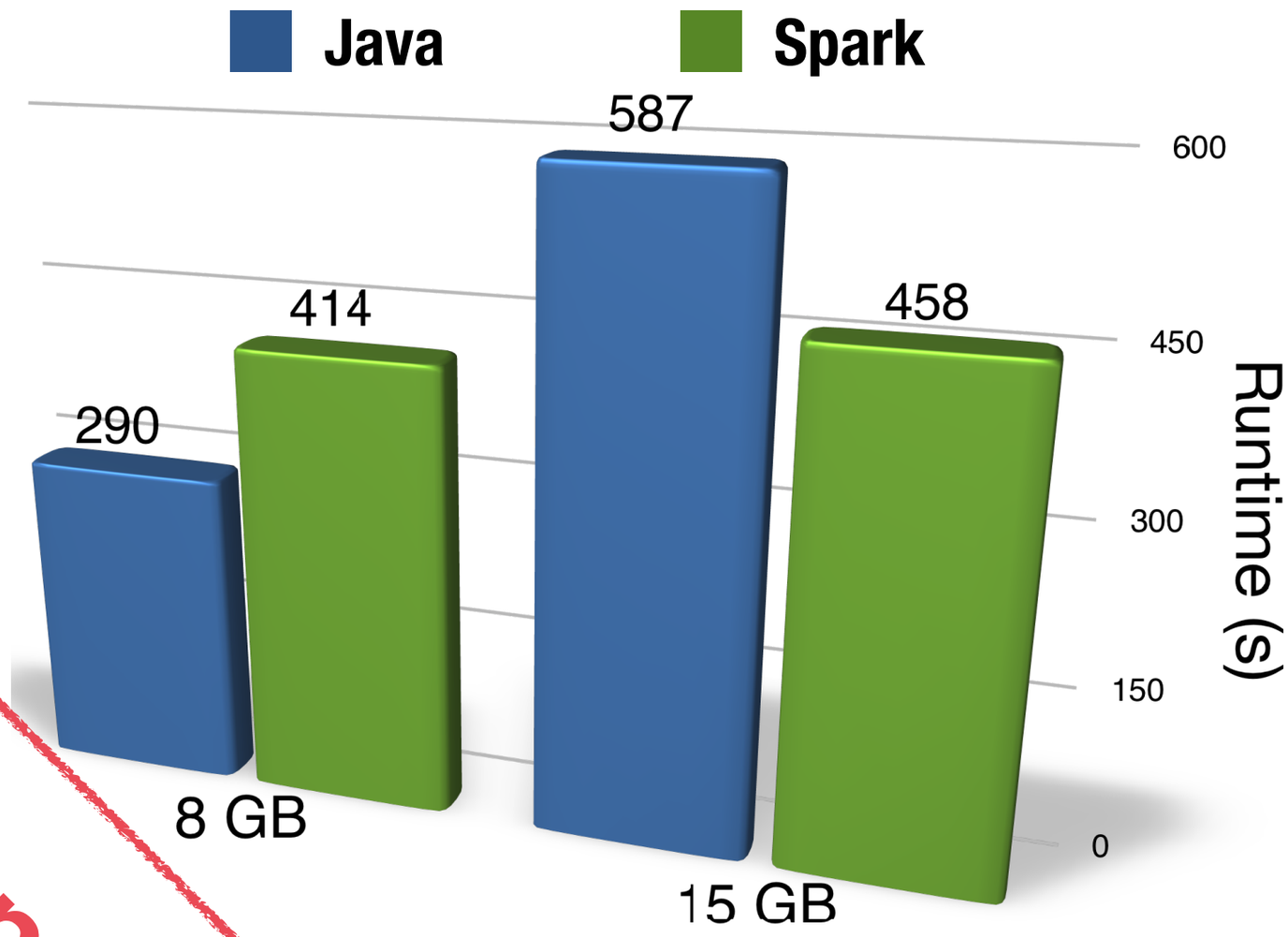
SGD



Decoupled Single-Platform

using ***any single*** data processing platform to process a query

SGD

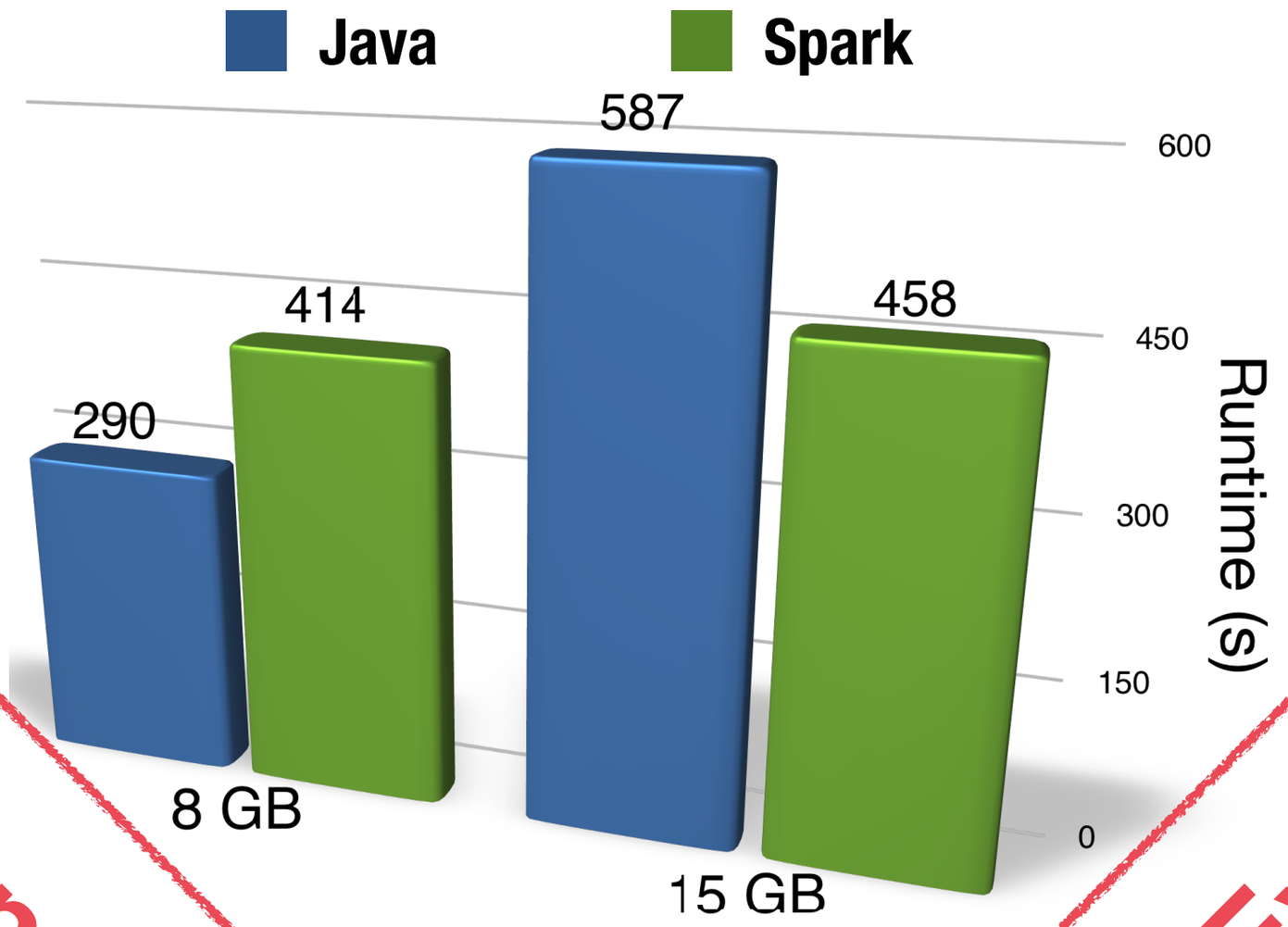


Performance

Decoupled Single-Platform

using ***any single*** data processing platform to process a query

SGD



Performance

Evolution

System-Store Quadrant

	Monoplatform	Polyplatform
Monostore	Decoupled Single Platform	
Polystore		

Cross-Platform Use Cases

Data Processing

Cross-Platform Data Processing

Decoupled Single-Platform

Multi-Platform

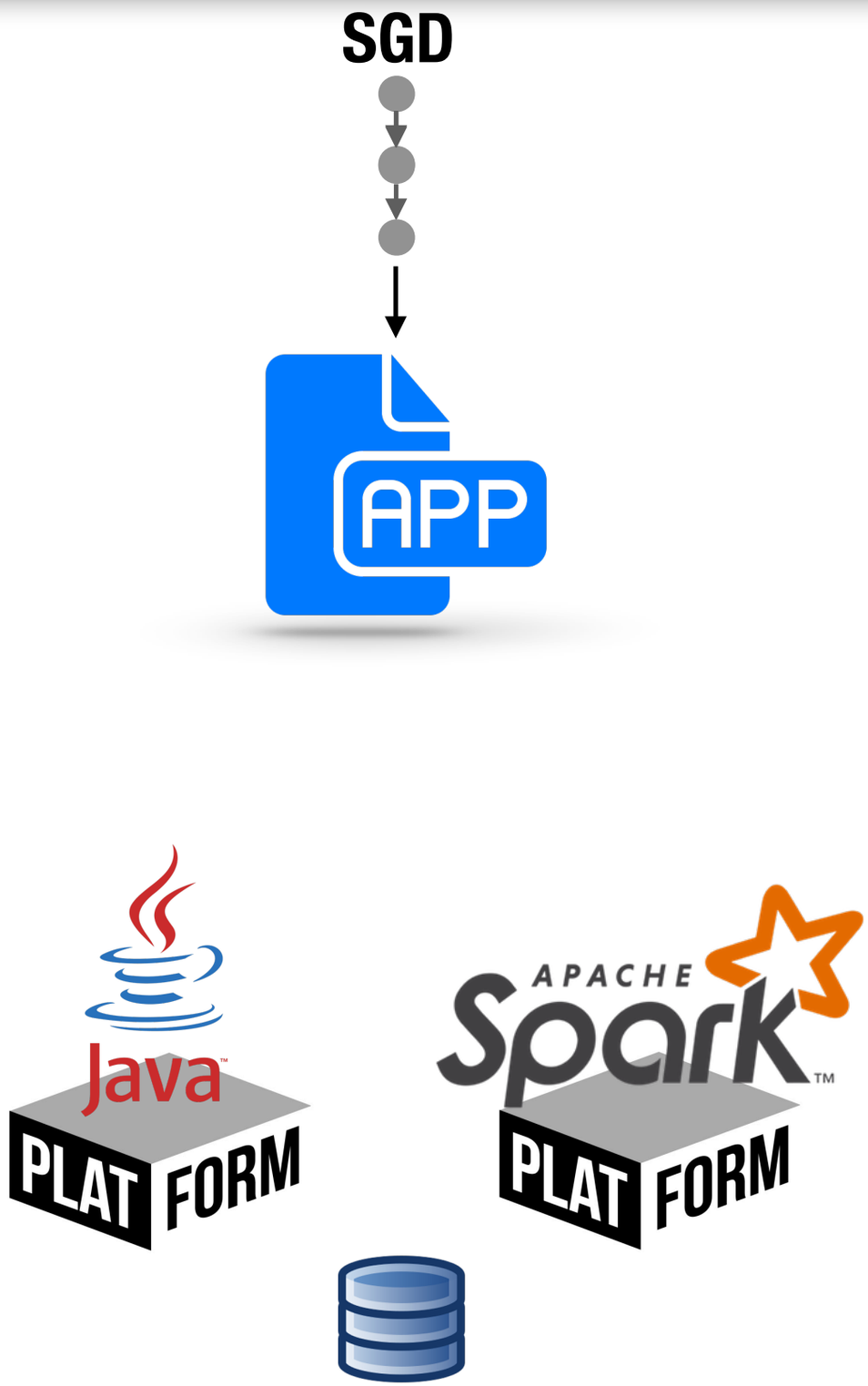
Opportunistic

Mandatory

Polystore

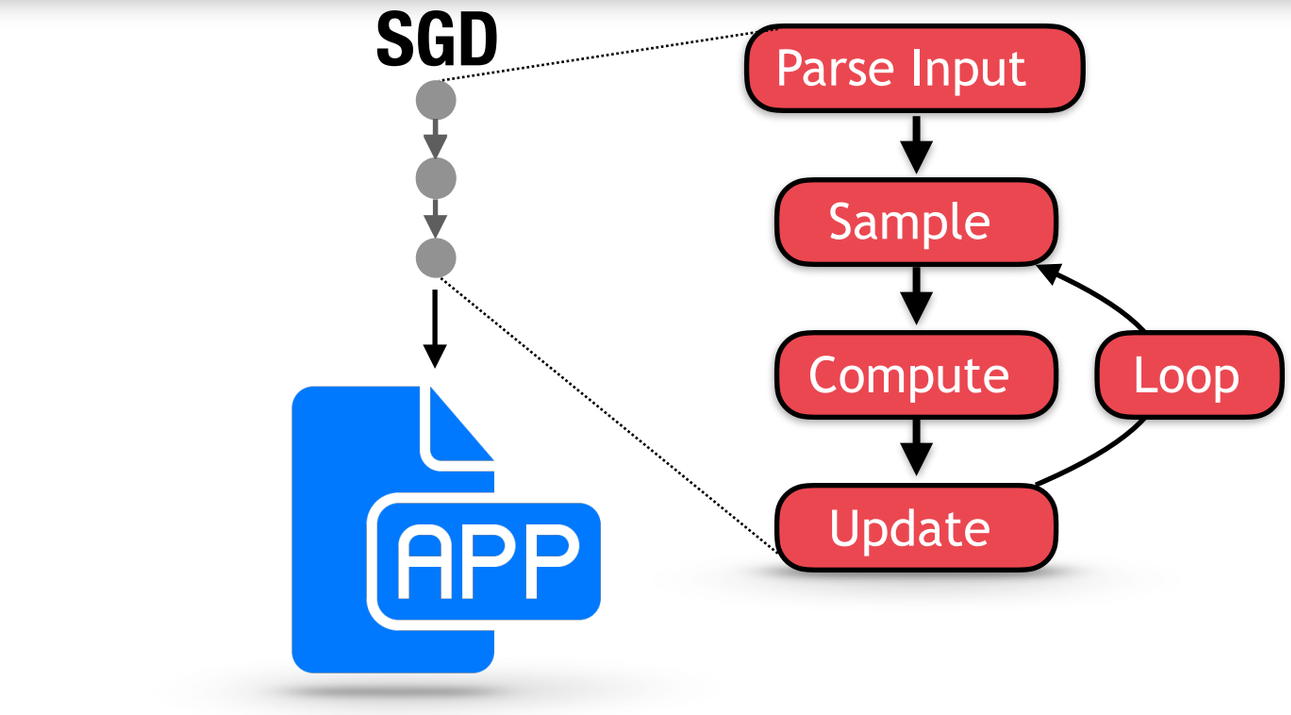
Opportunistic Cross-Platform

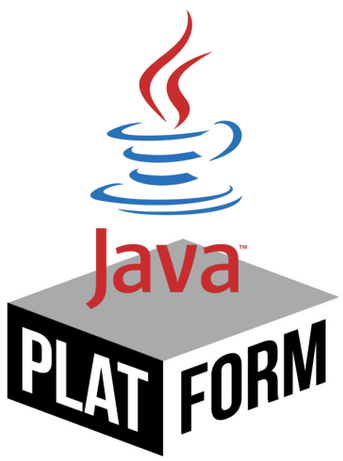
using ***several*** data processing platforms to reduce the cost of a query

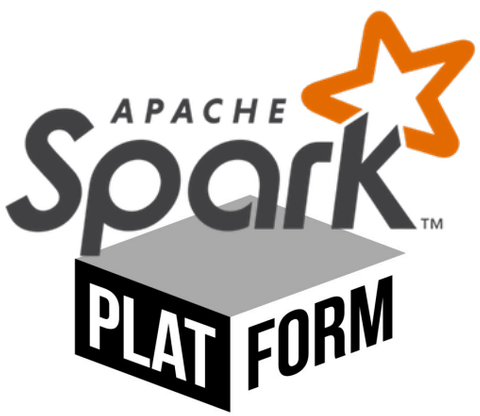


Opportunistic Cross-Platform

using ***several*** data processing platforms to reduce the cost of a query

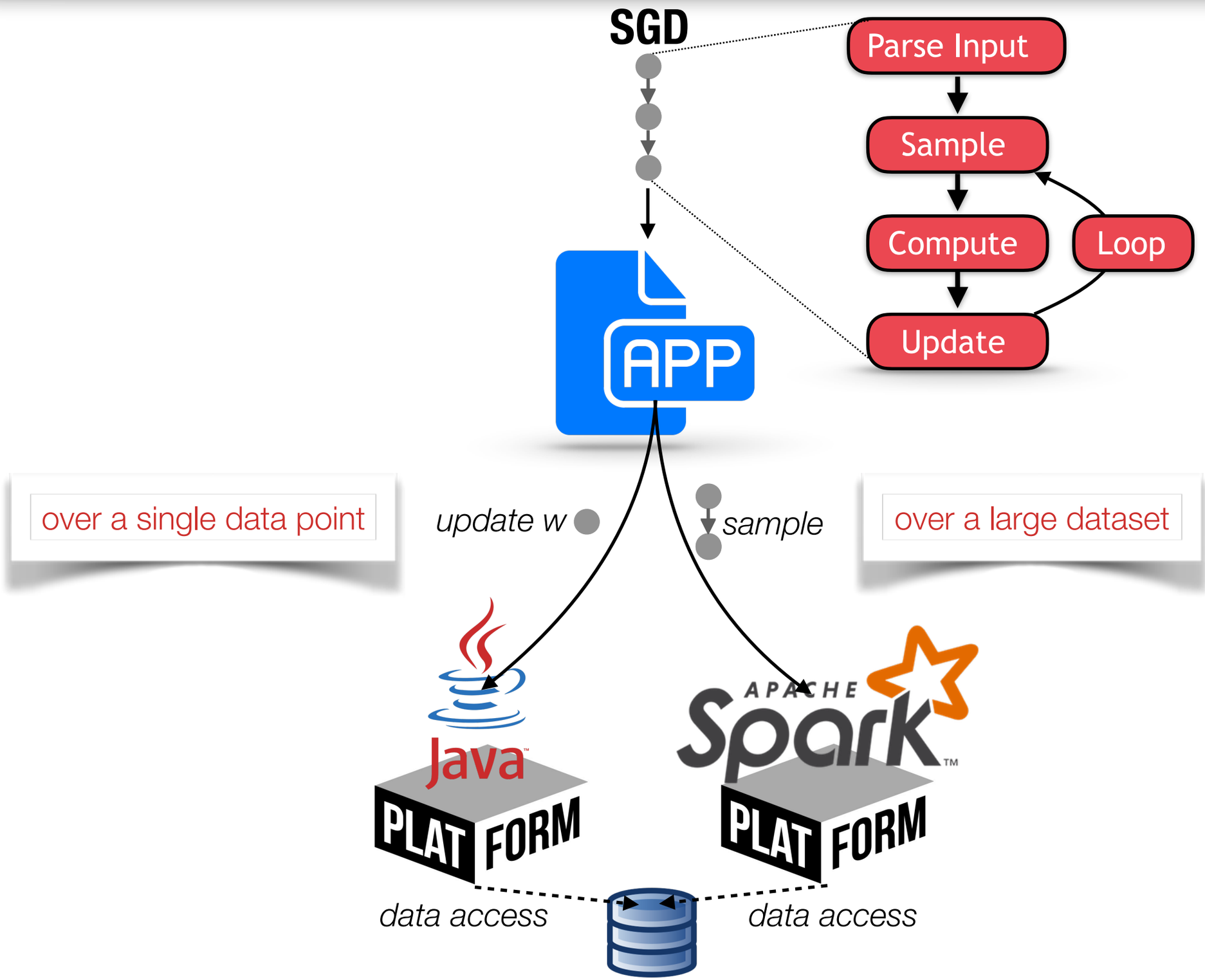






Opportunistic Cross-Platform

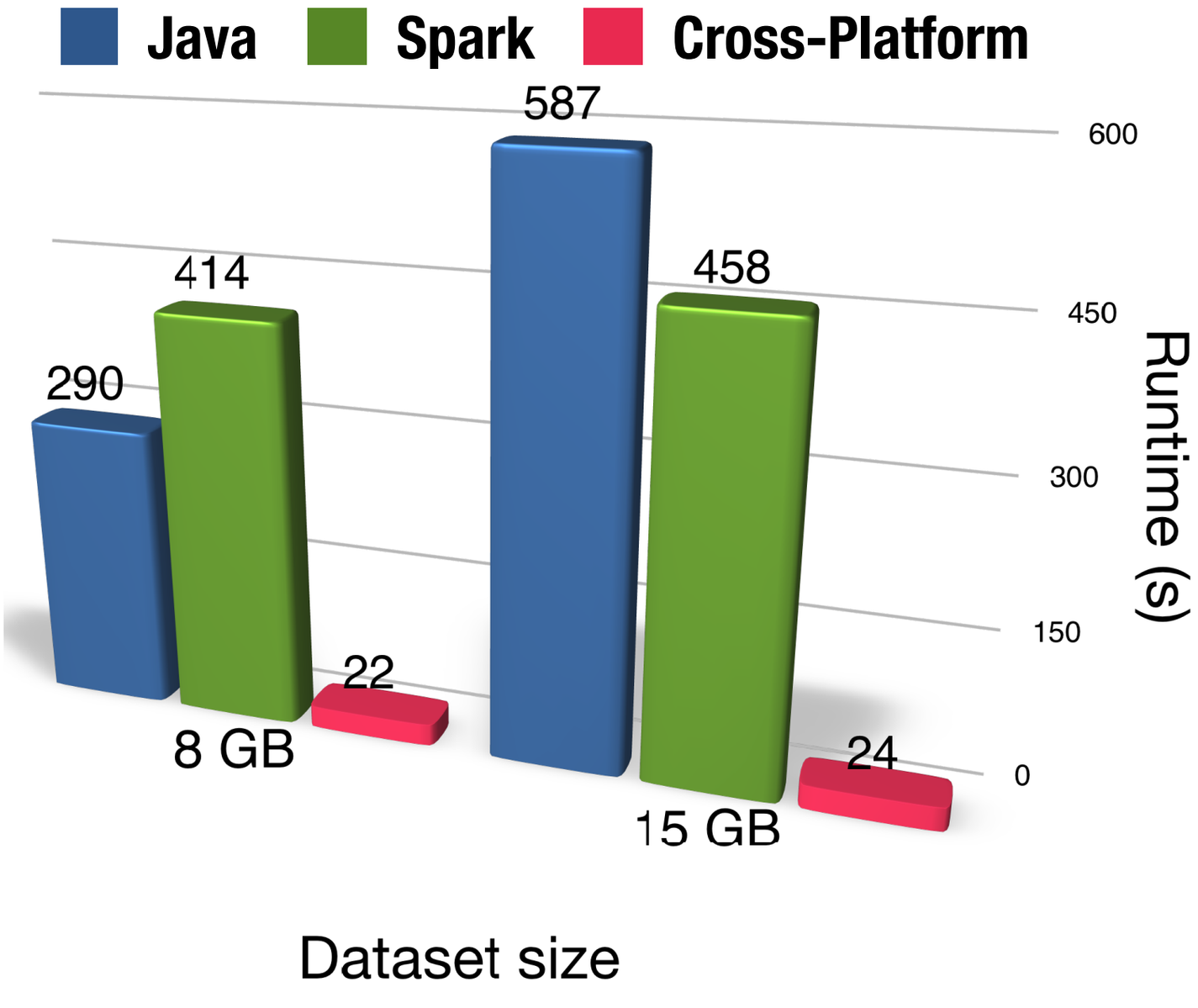
using ***several*** data processing platforms to reduce the cost of a query



Opportunistic Cross-Platform

using ***several*** data processing platforms to reduce the cost of a query

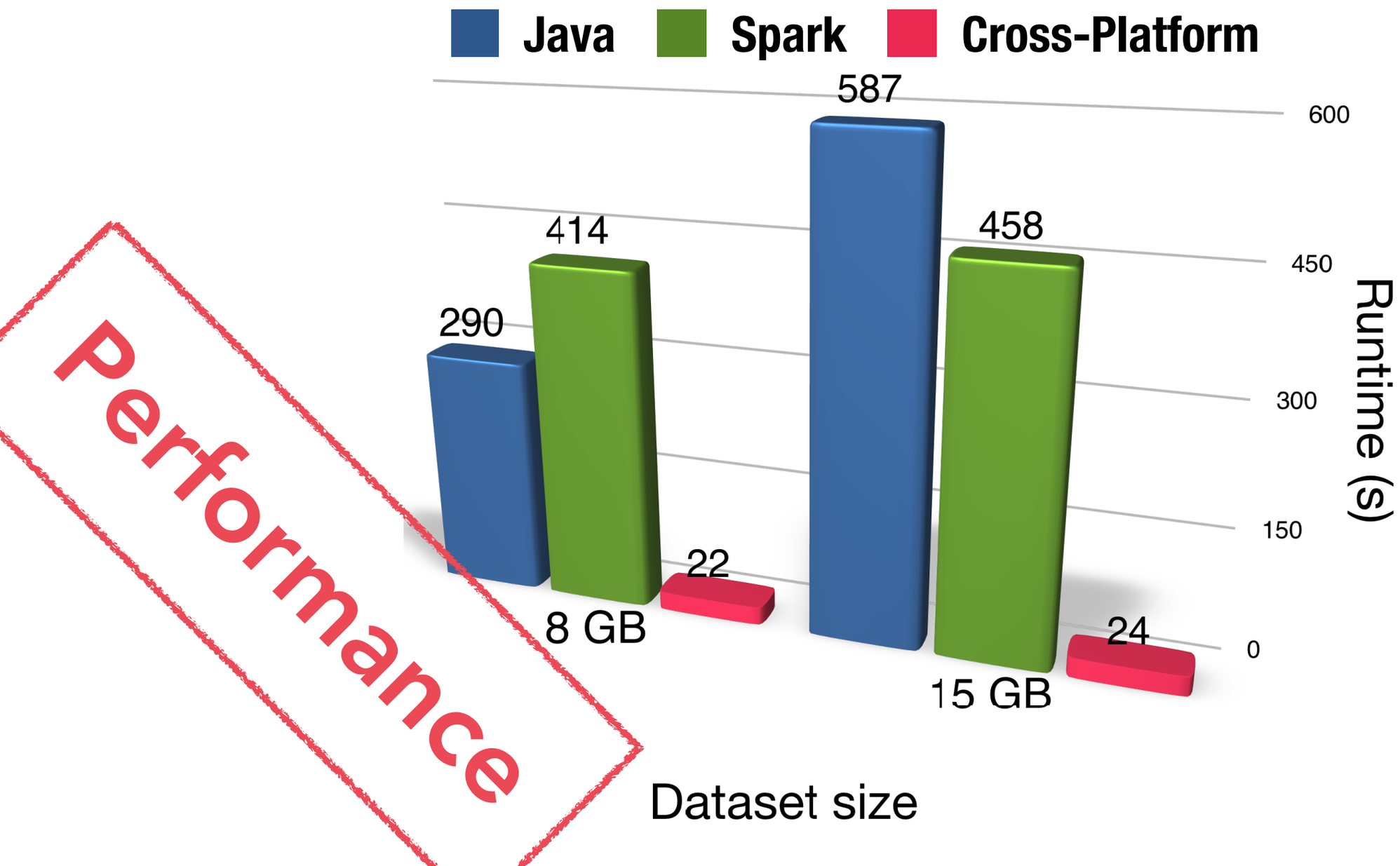
SGD



Opportunistic Cross-Platform

using ***several*** data processing platforms to reduce the cost of a query

SGD



Performance

System-Store Quadrant

	Monoplatform	Polyplatform
Monostore	Decoupled Single Platform	Opportunistic CP
Polystore		

Cross-Platform Use Cases

Data Processing

Cross-Platform Data Processing

Decoupled Single-Platform

Multi-Platform

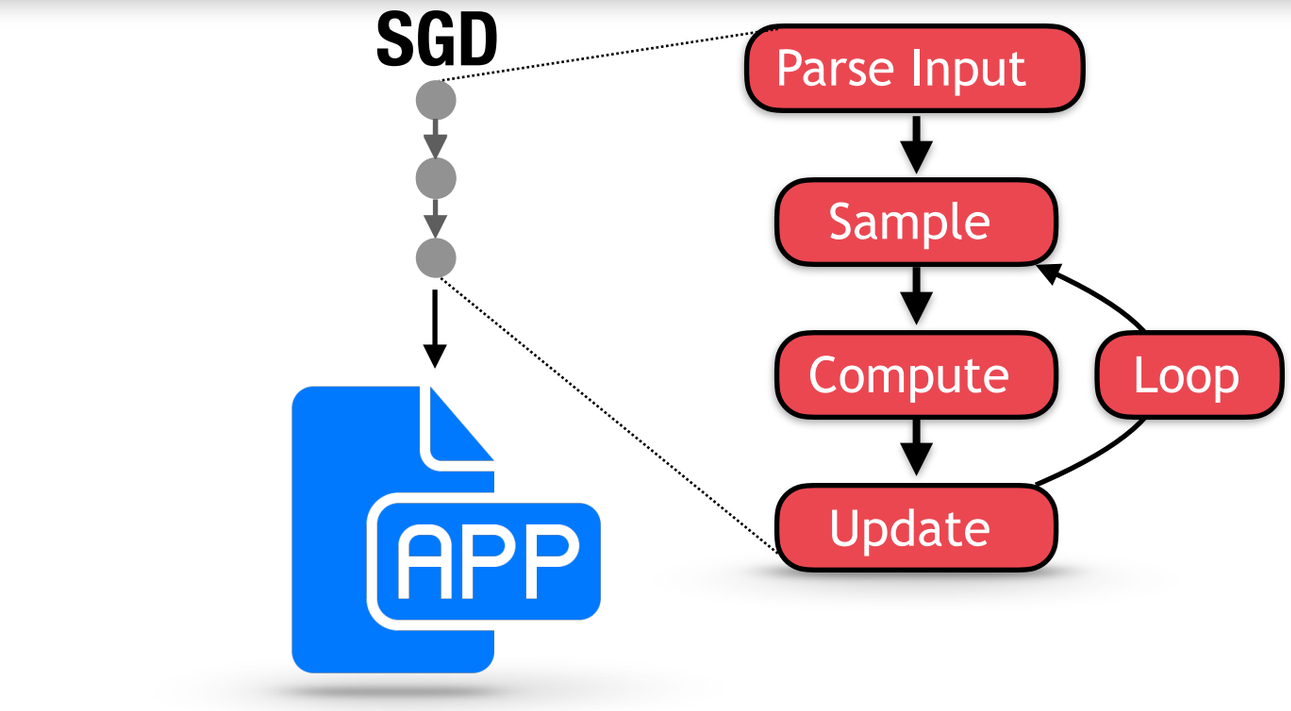
Opportunistic

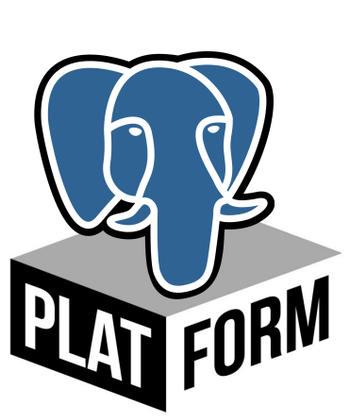
Mandatory

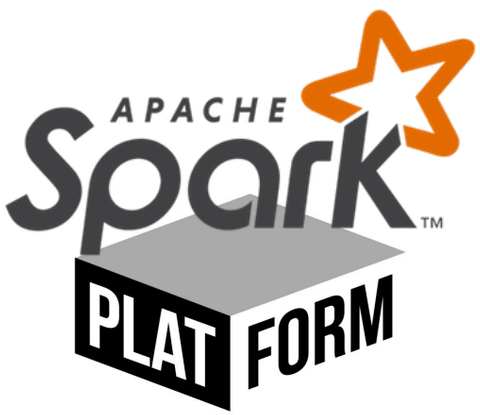
Polystore

Mandatory Cross-Platform

using ***several*** data processing platforms to be able to process a query

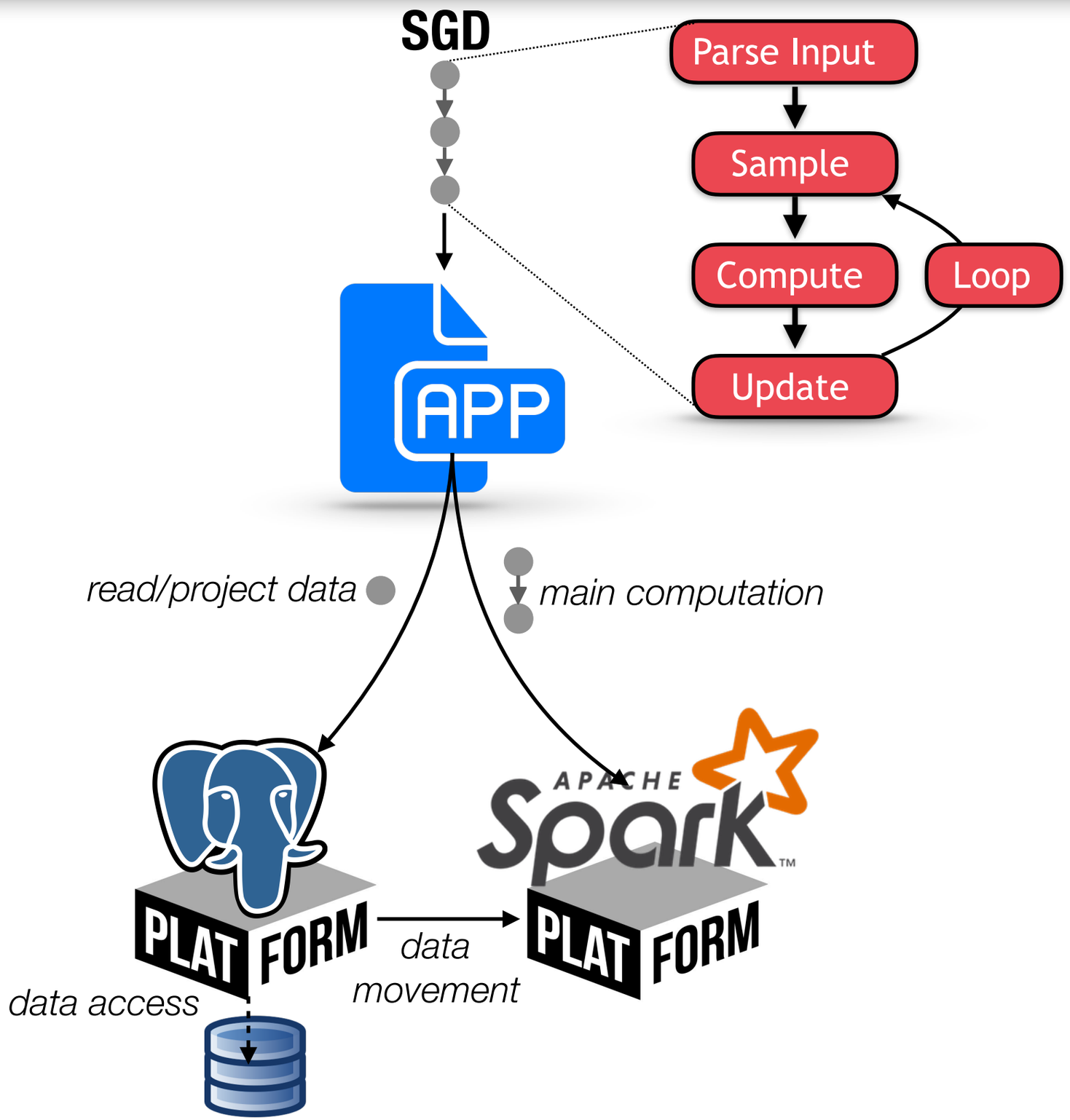






Mandatory Cross-Platform

using ***several*** data processing platforms to be able to process a query



System-Store Quadrant

	Monoplatform	Polyplatform
Monostore	Decoupled Single Platform	Opportunistic CP Mandatory CP
Polystore		

Cross-Platform Use Cases

Data Processing

Cross-Platform Data Processing

Decoupled Single-Platform

Multi-Platform

Opportunistic

Mandatory

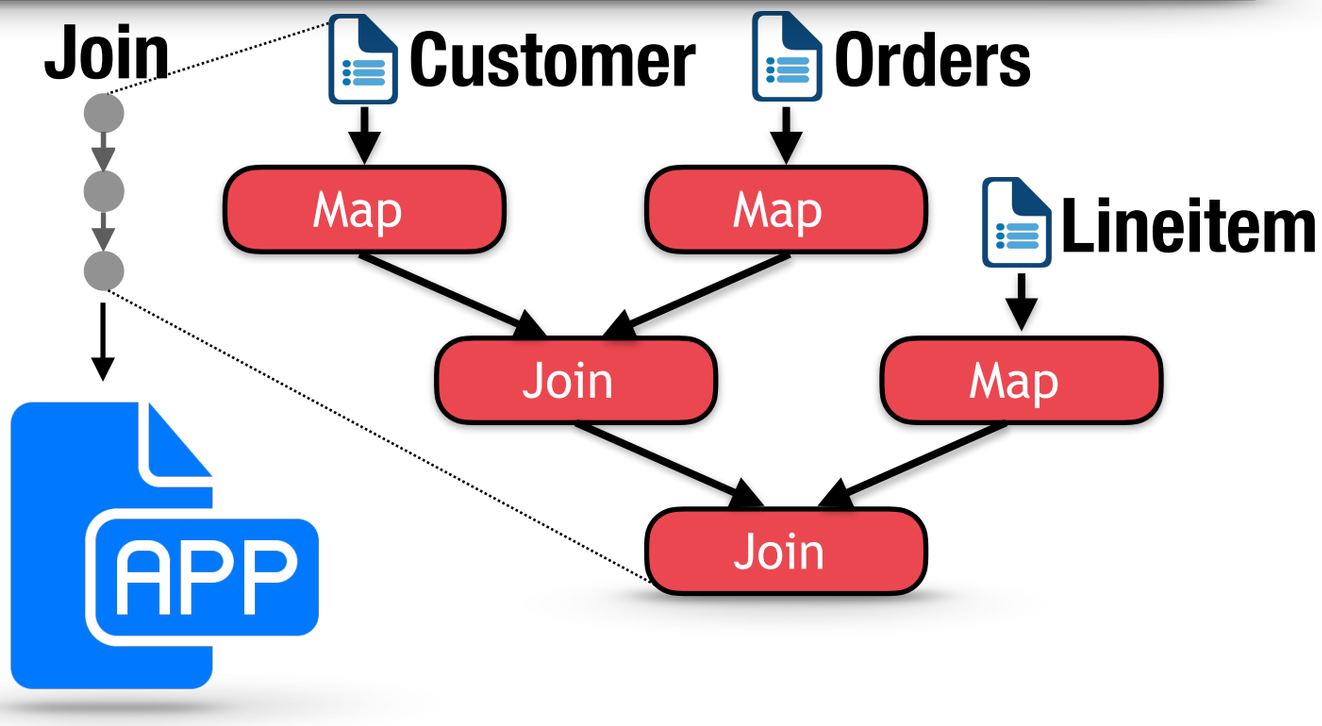
Polystore

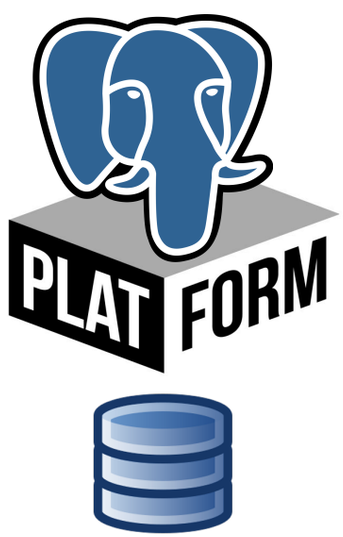
Polystore

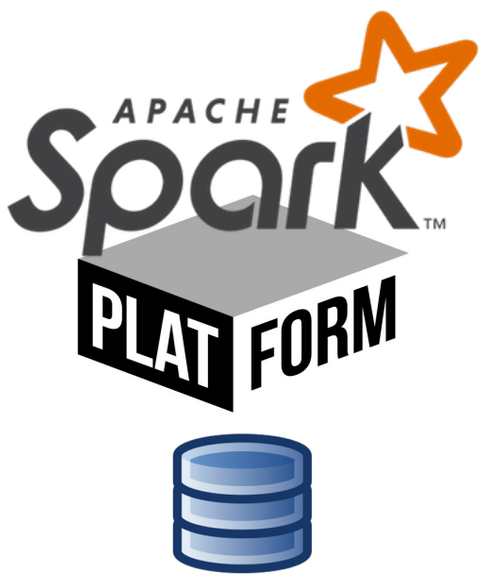
using ***several*** data processing platforms because data is spread

Polystore

using ***several*** data processing platforms because data is spread

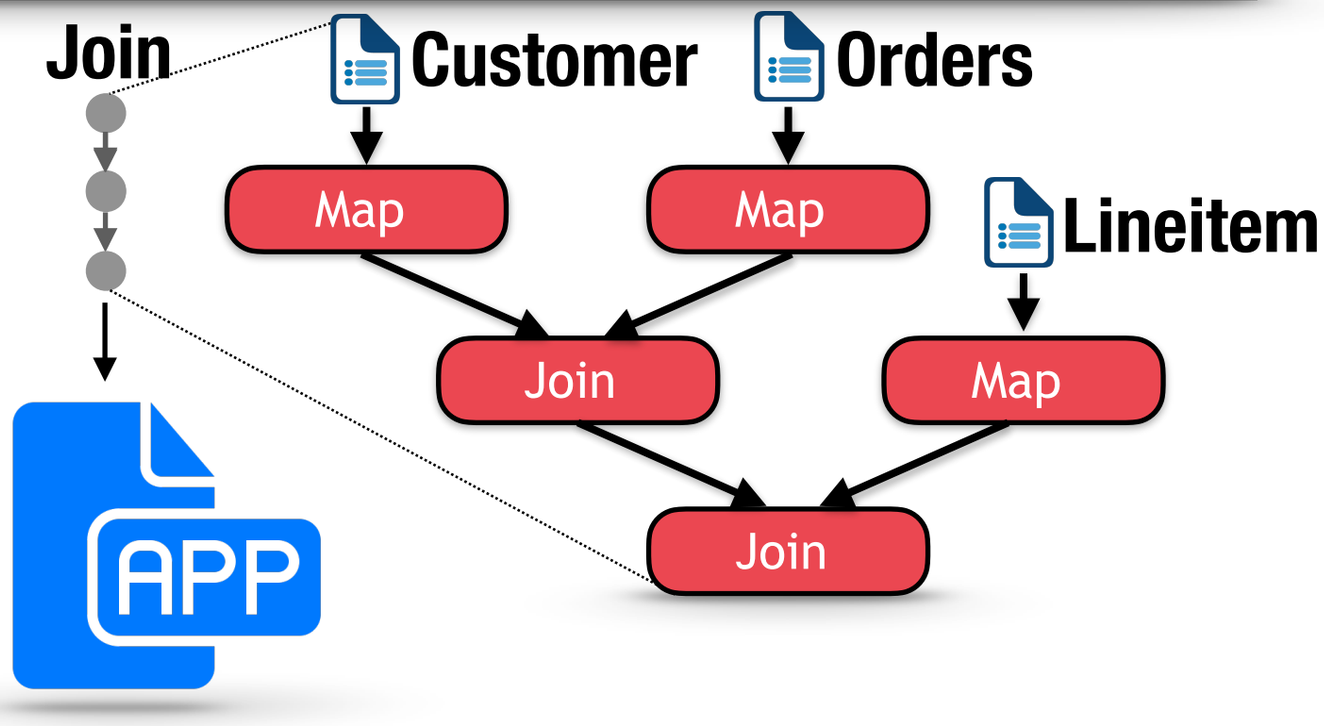


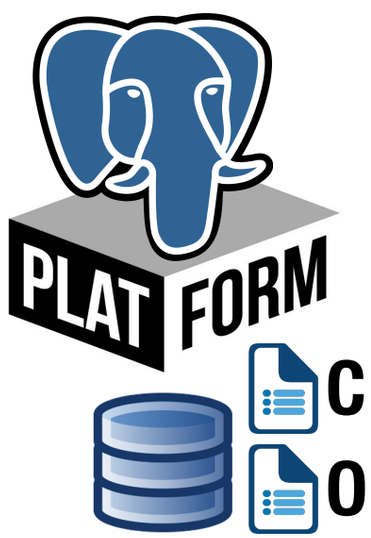




Polystore

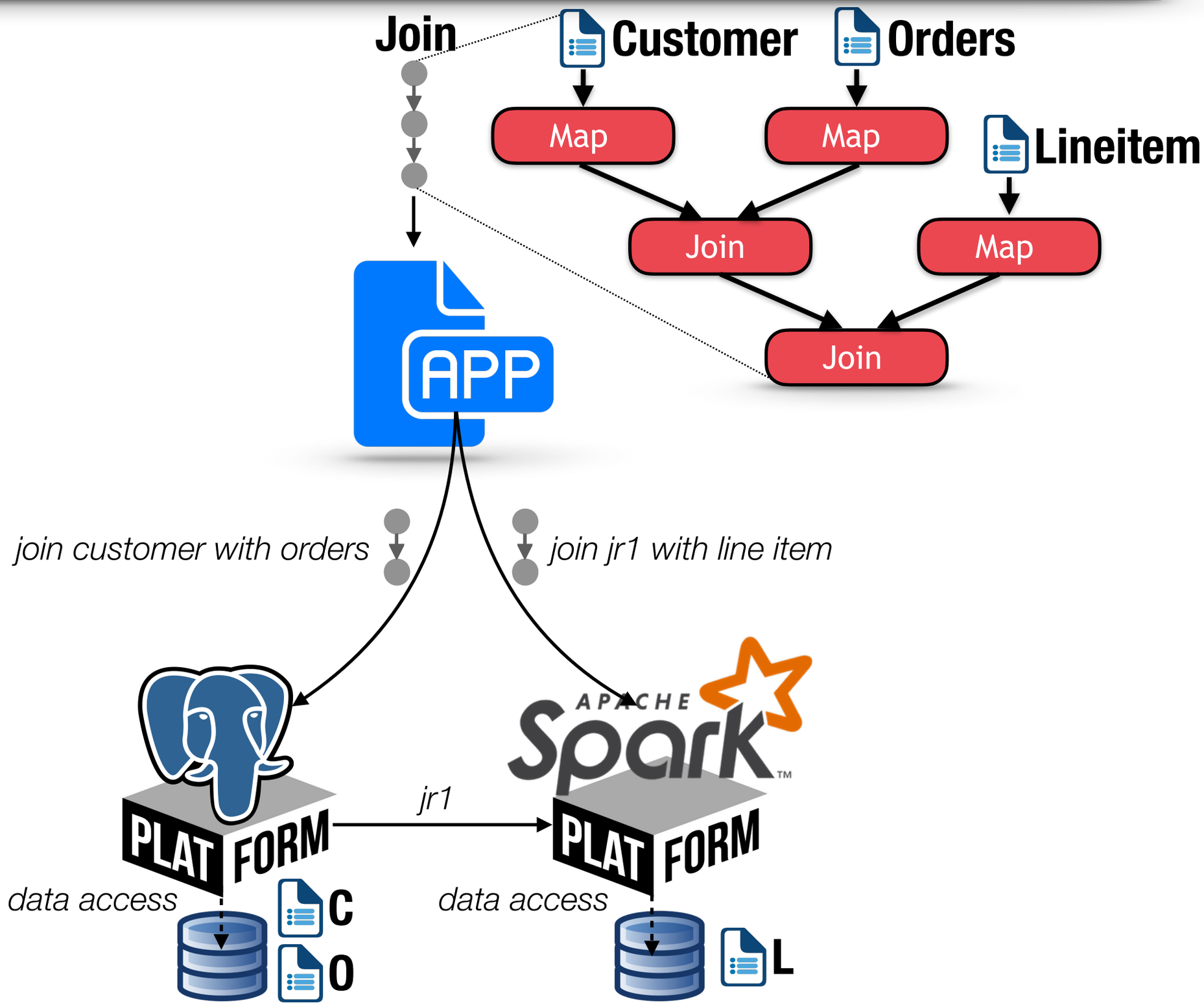
using ***several*** data processing platforms because data is spread





Polystore

using ***several*** data processing platforms because data is spread



System-Store Quadrant

	Monoplatform	Polyplatform
Monostore	Decoupled Single Platform	Opportunistic CP Mandatory CP
Polystore		Polystore CP

System-Store Quadrant

	Monoplatform	Polyplatform
Monostore	Decoupled Single Platform	Opportunistic CP Mandatory CP
Polystore	Decoupled Single Platform	Polystore CP

Federated DBs vs Cross-Platform (II)

Polystore

Decoupled Single Platform
Opportunistic CP
Mandatory CP
Polystore CP

Single Data Model

Disparate Data Models

Federated DB System

Cross-Platform System

Cross-Platform Data Processing

Motivation

Use Cases

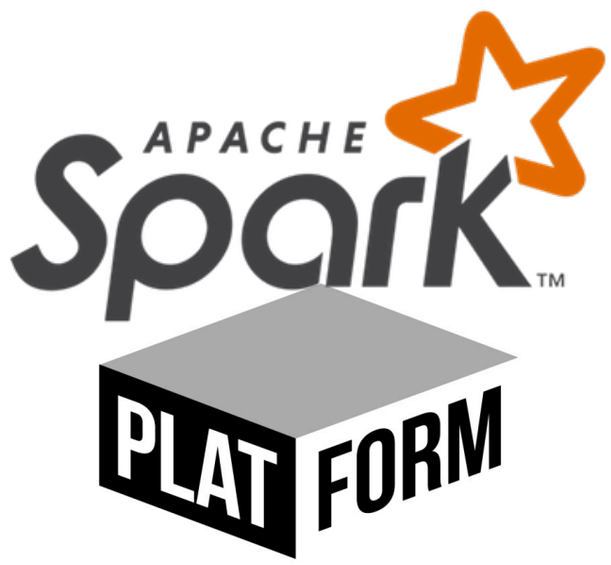
Challenges

Current Efforts

Summary

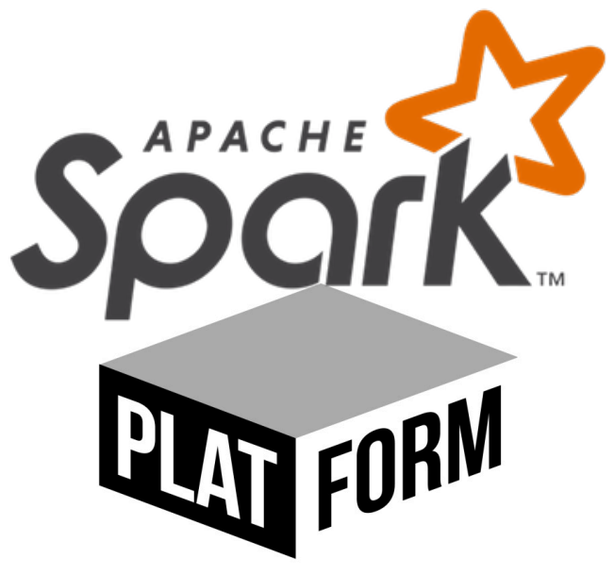
Challenges

Challenges



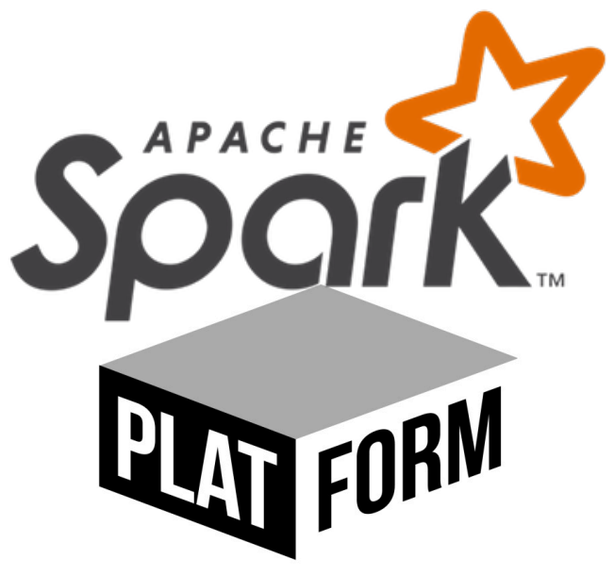
Challenges

1 Decoupling Applications



Challenges

1 Decoupling Applications

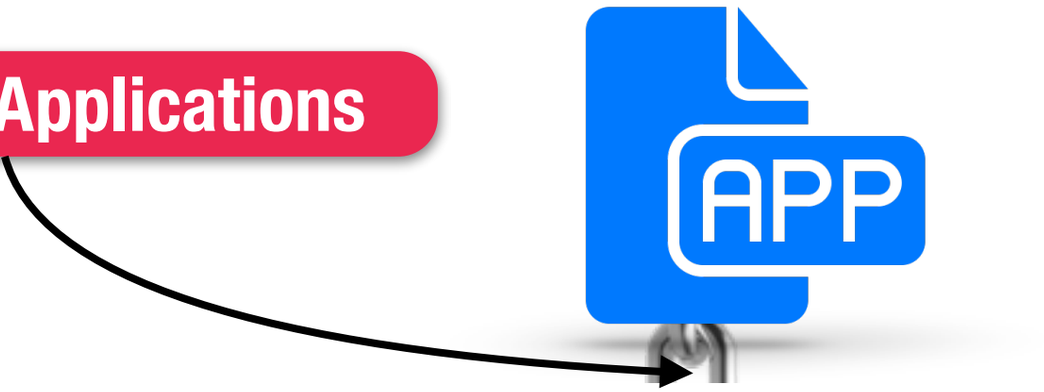


Challenges

1 Decoupling Applications

Challenges

1 Decoupling Applications



2 Data Movement

Challenges

1 Decoupling Applications

3

Automatic Cross-Platform Data Processing

2 Data Movement

Challenges

1 **Decoupling Applications**

3

Automatic Cross-Platform Data Processing

2 **Data Movement**

Challenges

1 **Decoupling Applications**

3

Automatic Cross-Platform Data Processing

2 **Data Movement**

Challenges

1 Decoupling Applications

4 Extensibility

3

Automatic Cross-Platform Data Processing

2 Data Movement

Cross-Platform Data Processing

Motivation

Use Cases

Challenges

Current Efforts

Summary

Challenges

1 Decoupling Applications

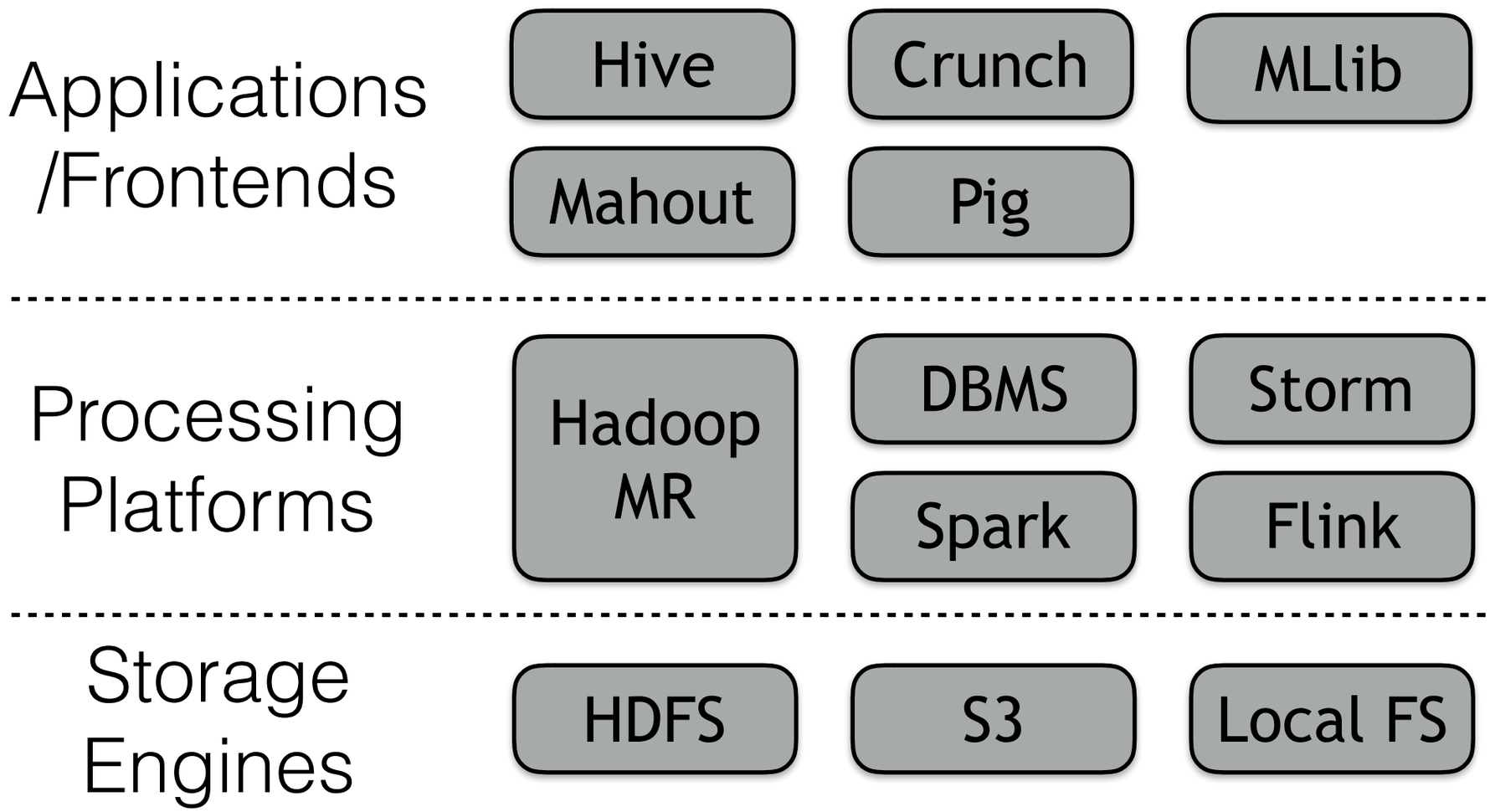
4 Extensibility

2

Automatic Cross-Platform Data Processing

3 Data Movement

Decoupling Applications



Decoupling Applications

Applications /Frontends

Hive | Crunch | MLlib | Mahout | Pig

Processing Platforms

Hadoop MR | DBMS | Storm | Spark | Flink

Storage Engines

HDFS | S3 | Local FS

Decoupling Applications

Applications /Frontends

Hive | Crunch | MLlib | Mahout | Pig

Cross-Platform System

Processing Platforms

Hadoop MR | DBMS | Storm | Spark | Flink

Storage Engines

HDFS | S3 | Local FS

Decoupling Applications

Applications /Frontends

Hive | Crunch | MLlib | Mahout | Pig

Musketeer **Apache Beam**

Rheem **QoX**

BigDawg **Ires** **DBMS+**

Processing Platforms

Hadoop MR | DBMS | Storm | Spark | Flink

Storage Engines

HDFS | S3 | Local FS

Decoupling Applications

Applications /Frontends

Hive | Crunch | MLlib | Mahout | Pig

Musketeer | **Apache Beam** | **Rheem** | **QoX** | **BigDawg** | **Ires** | **DBMS+**

Processing Platforms

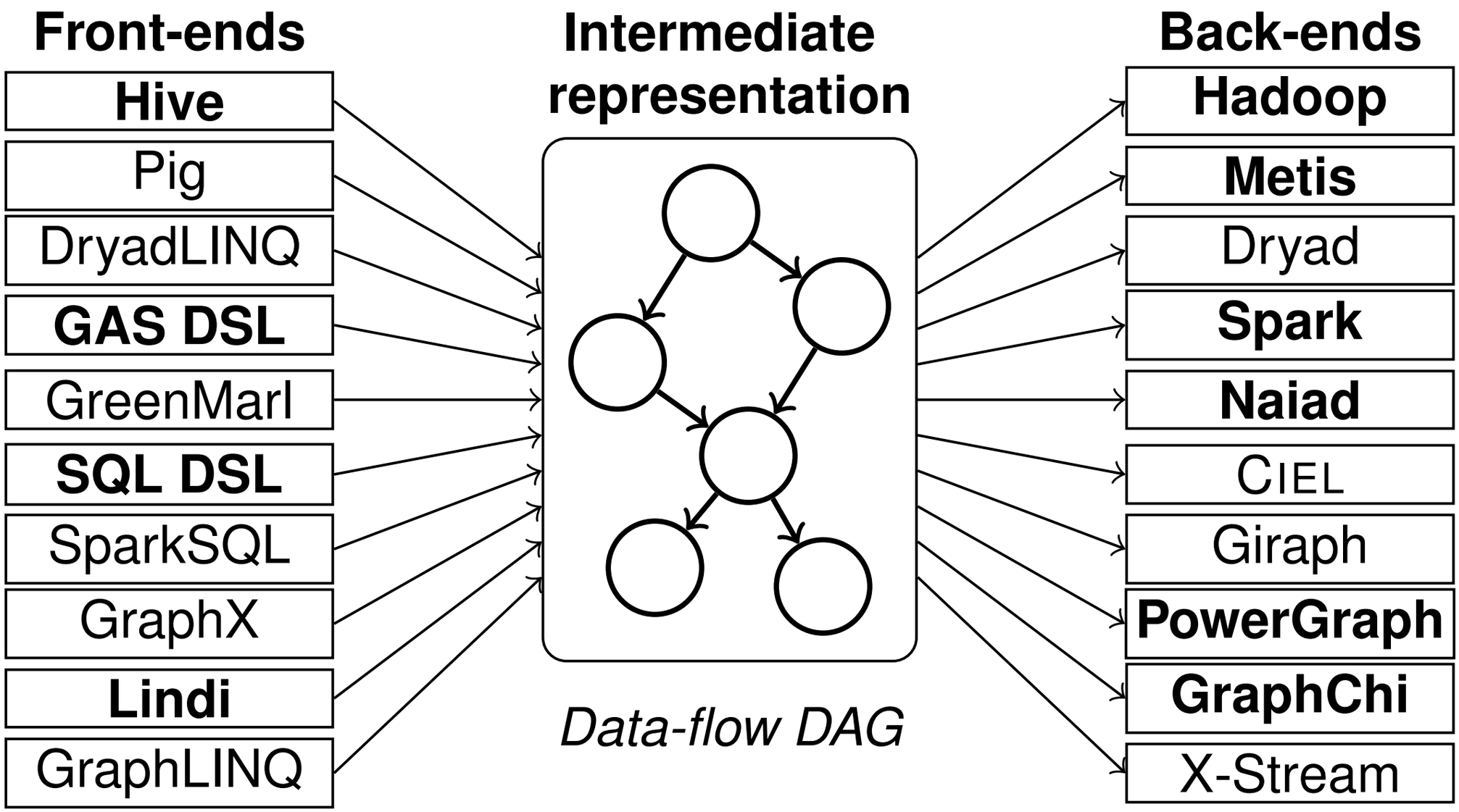
Hadoop MR | DBMS | Storm | Spark | Flink

Storage Engines

HDFS | S3 | Local FS

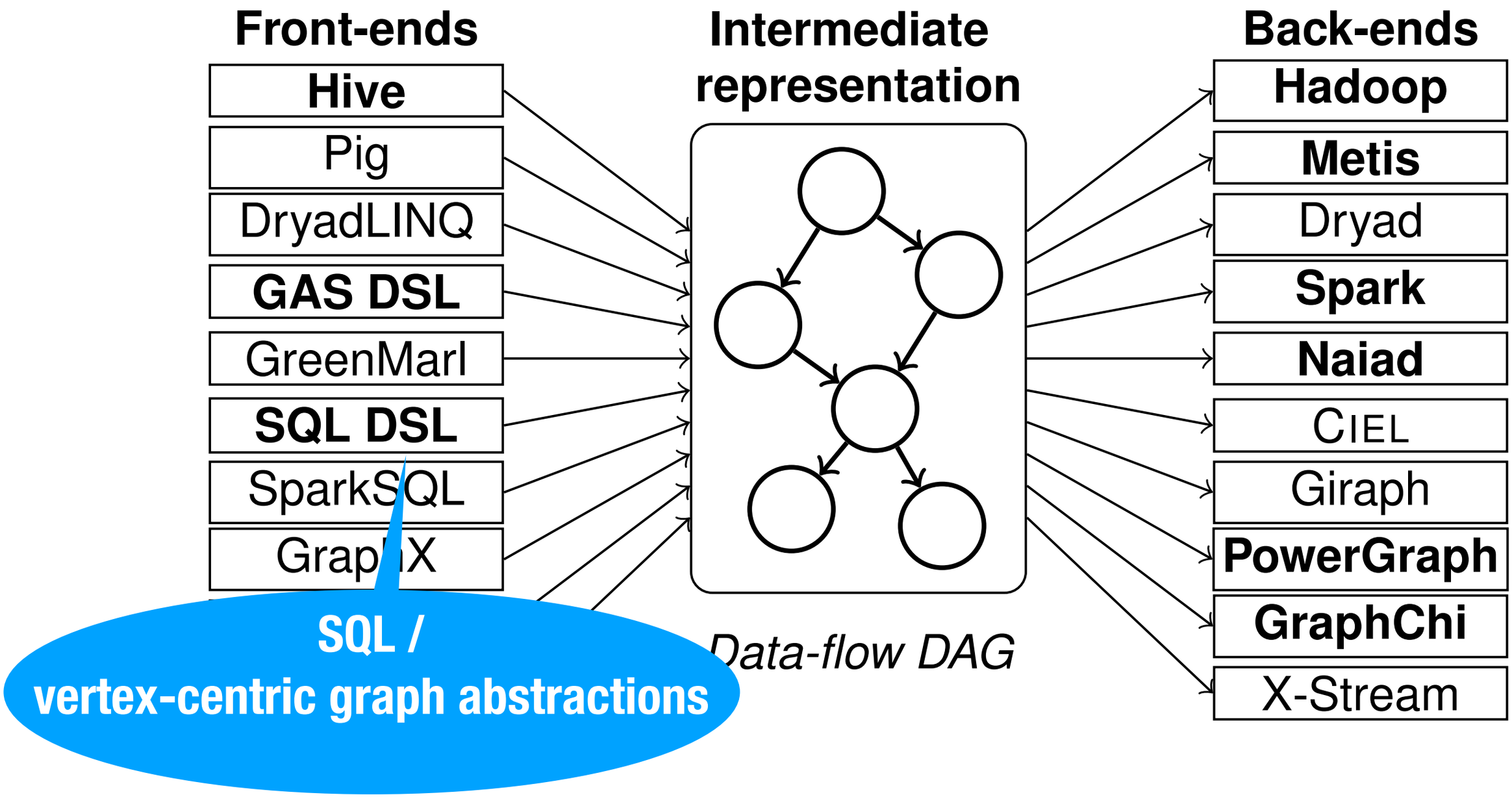
Musketeer

DAG-based ***intermediate representation***



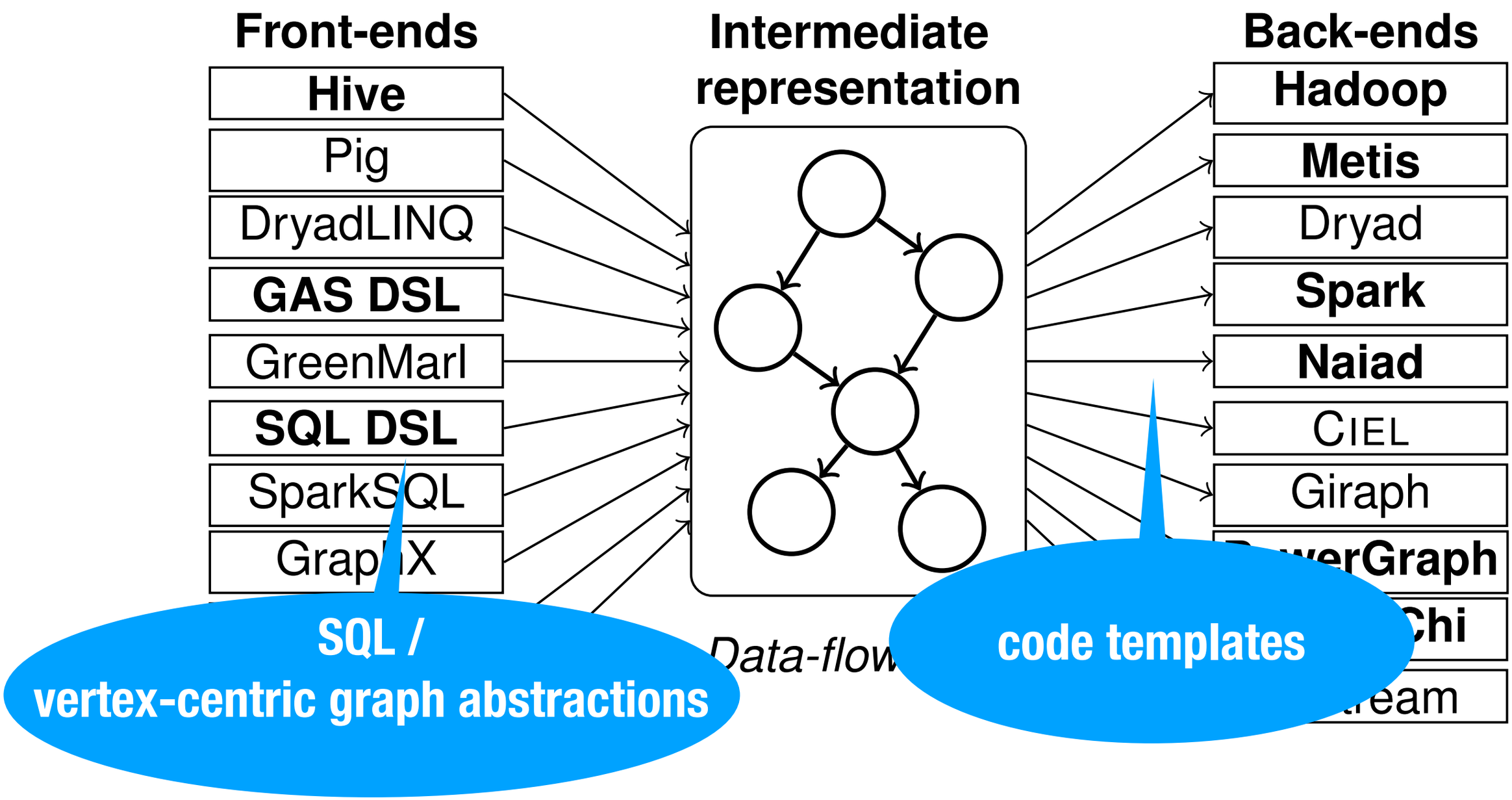
Musketeer

DAG-based ***intermediate representation***



Musketeer

DAG-based ***intermediate representation***



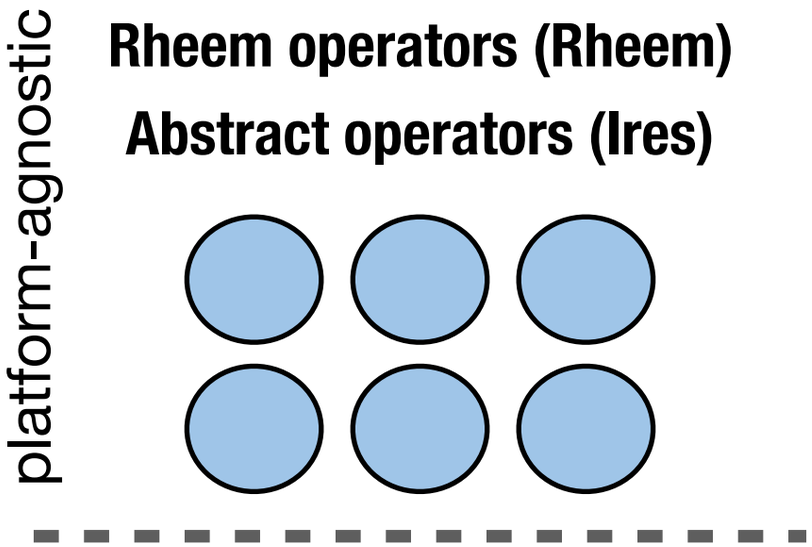
Musketeer: all for one, one for all in data processing systems. EuroSys'15.

Decoupling Features

	Processing Model	CP Granularity	Mappings
Musketeer	Dataflow DAG + Loop	Operator	Code templates

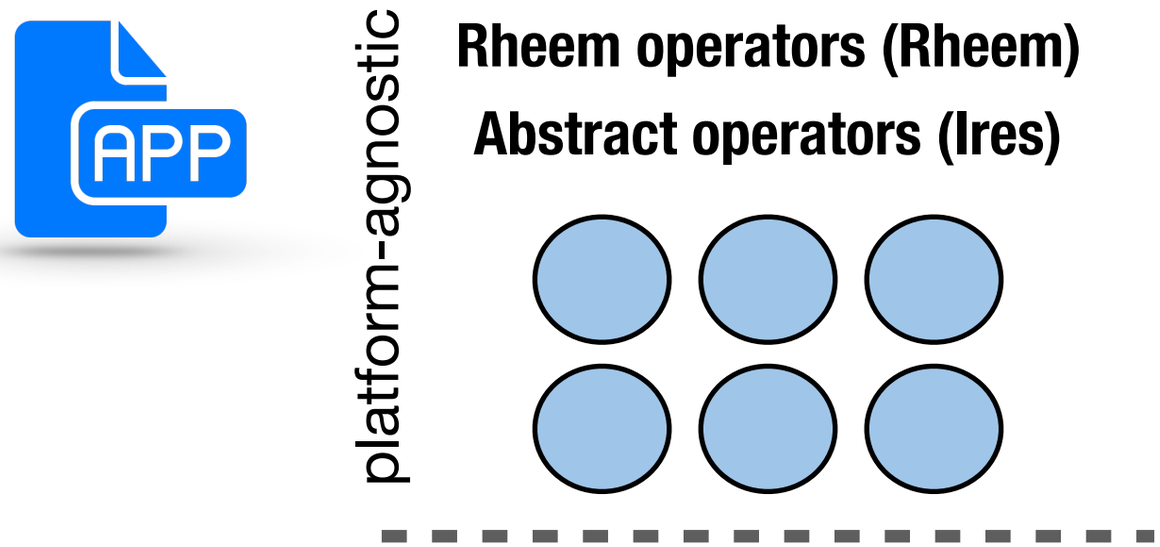
Rheem & Ires

2 layers of operators



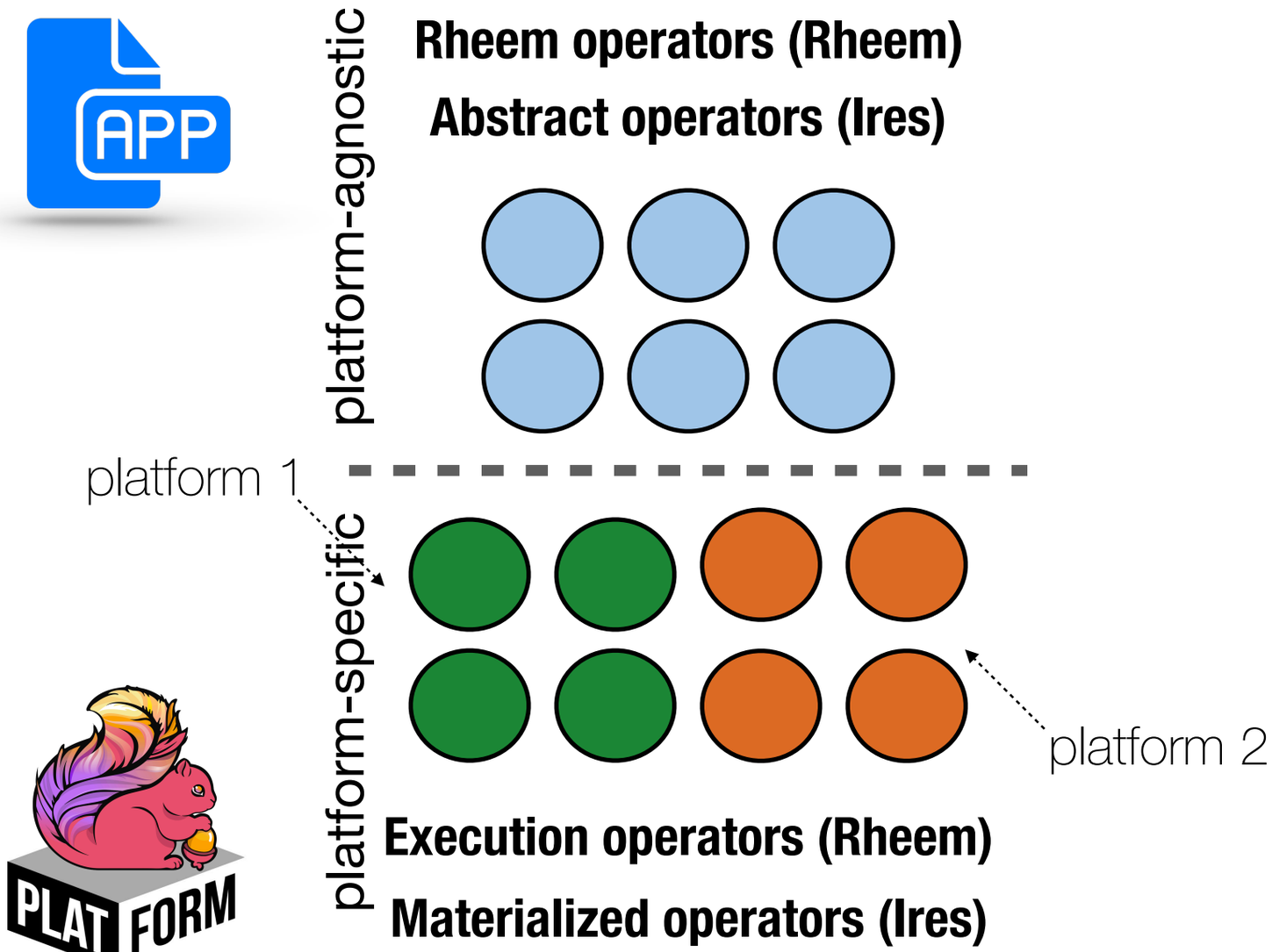
Rheem & Ires

2 layers of operators



Rheem & Ires

2 layers of operators

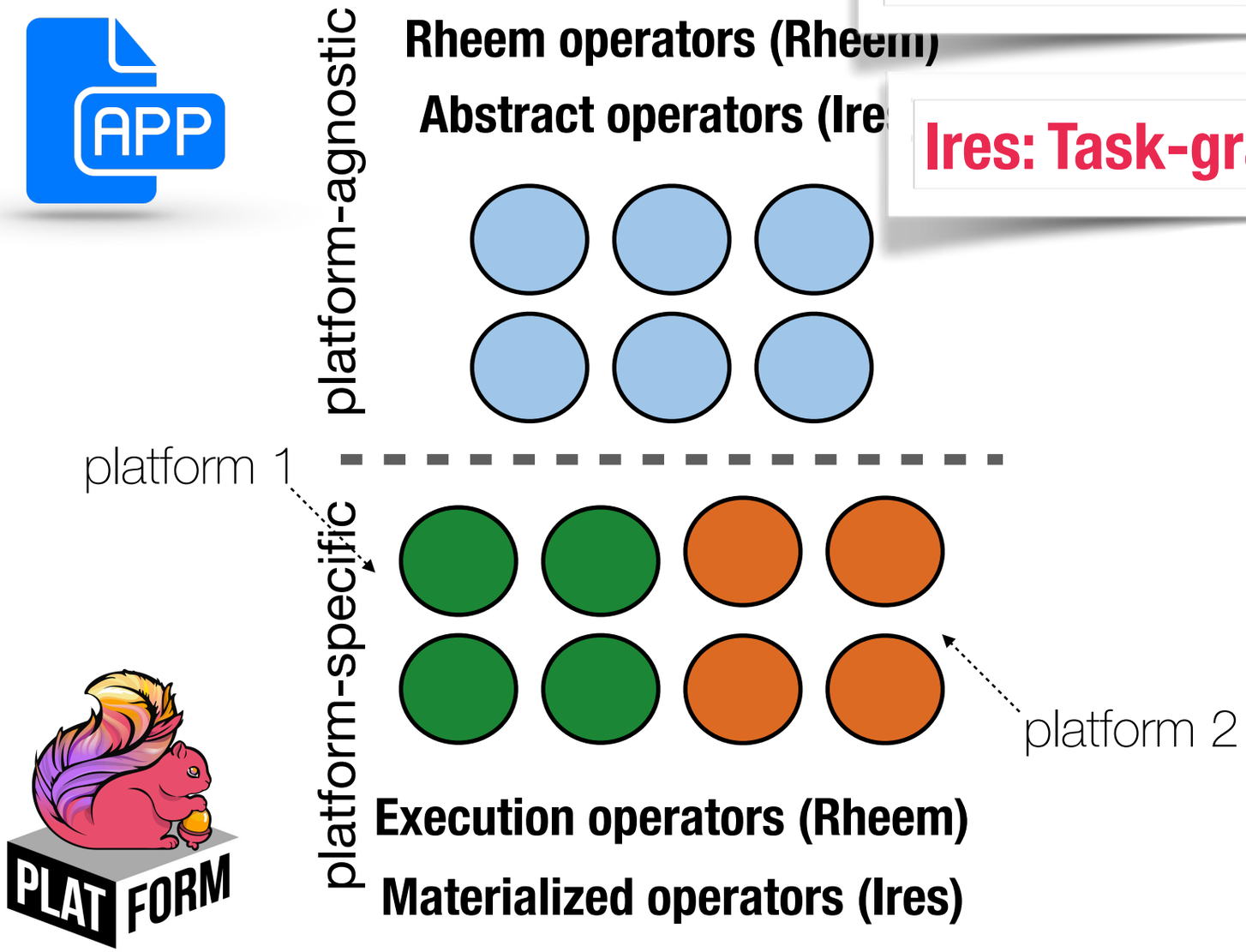


Rheem & Ires

2 layers of operators

Rheem: fine-granular operators

Ires: Task-granular operators

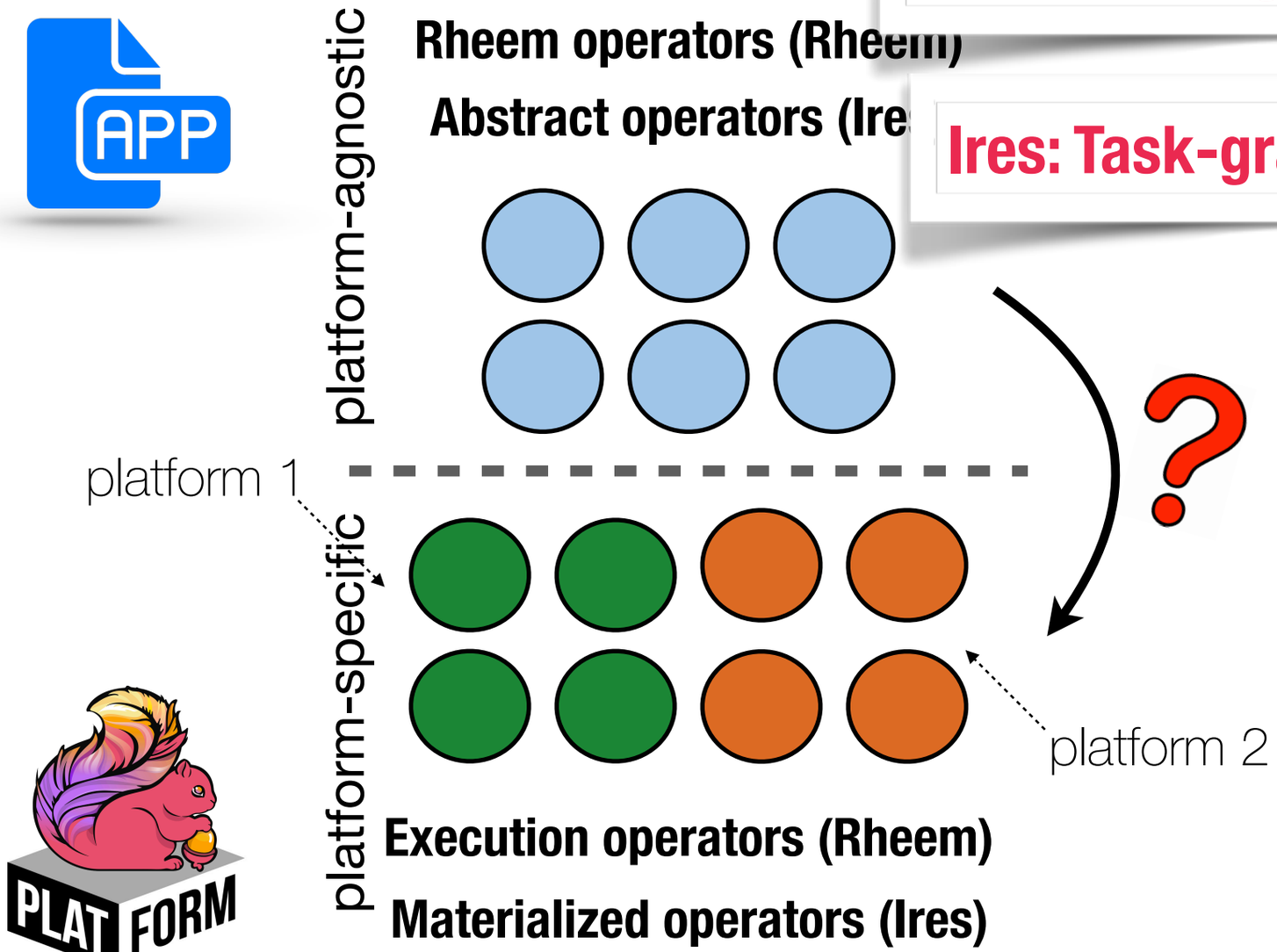


Rheem & Ires

2 layers of operators

Rheem: fine-granular operators

Ires: Task-granular operators



Rheem & Ires

Graph-based mappings

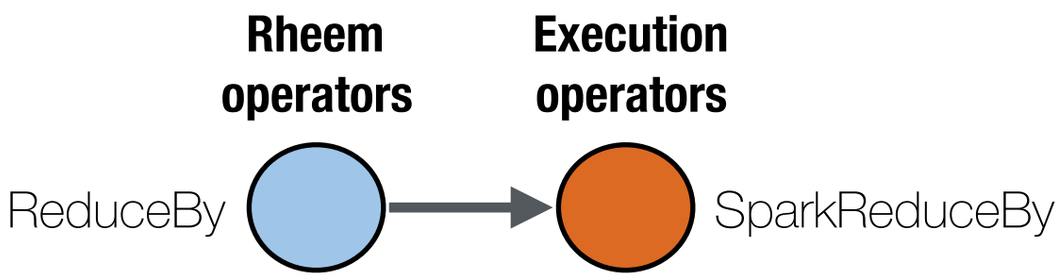
Metadata-based mappings

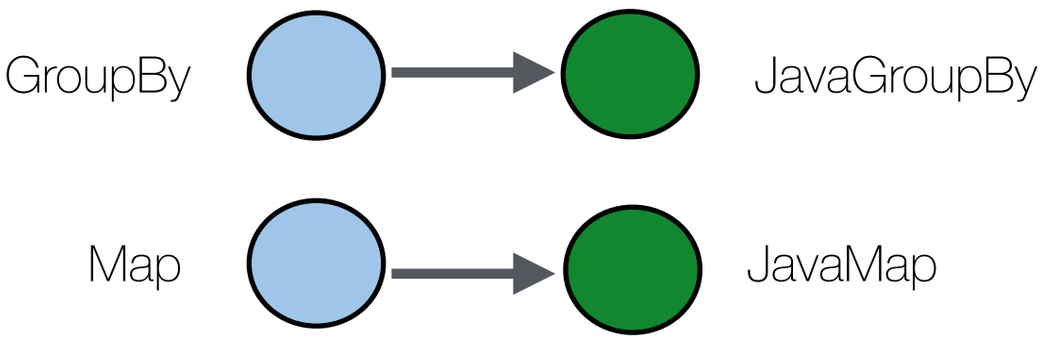
Road to Freedom in Big Data Analytics. EDBT 2016

Mix ’n’ Match Multi-Engine Analytics. IEEE BigData 2017

Rheem & Ires

Graph-based mappings





Metadata-based mappings

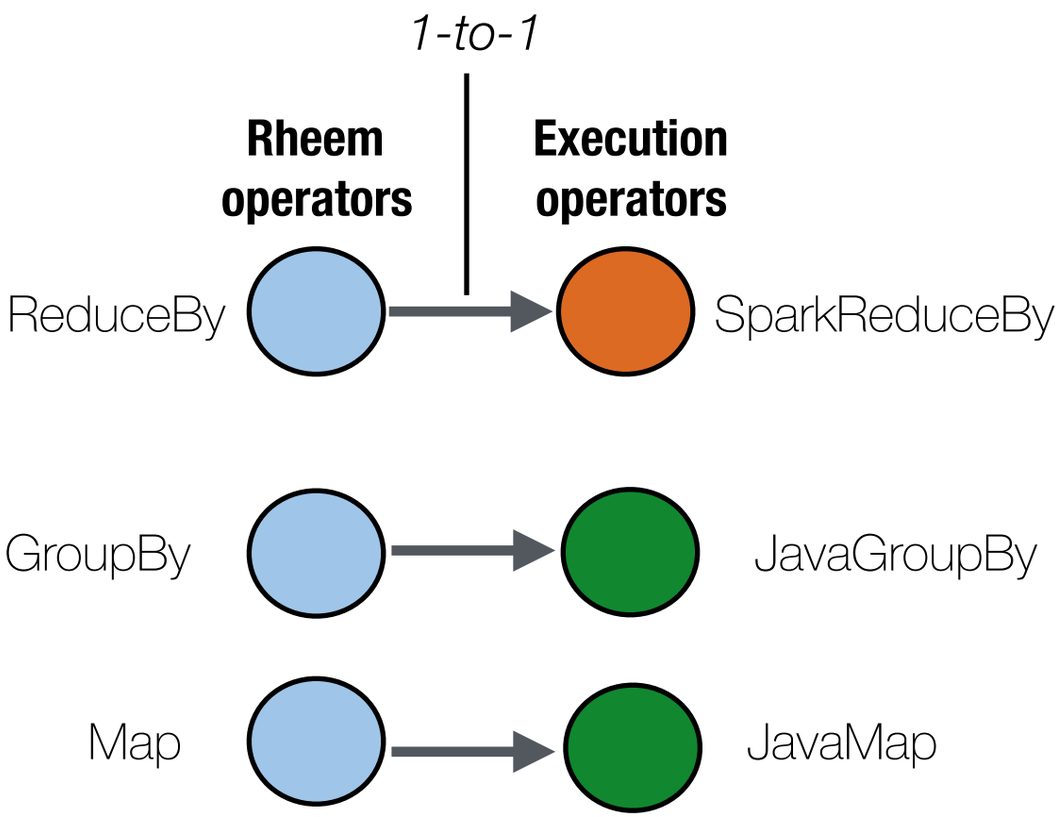
Road to Freedom in Big Data Analytics. EDBT 2016

Mix 'n' Match Multi-Engine Analytics. IEEE BigData 2017

Rheem & Ires

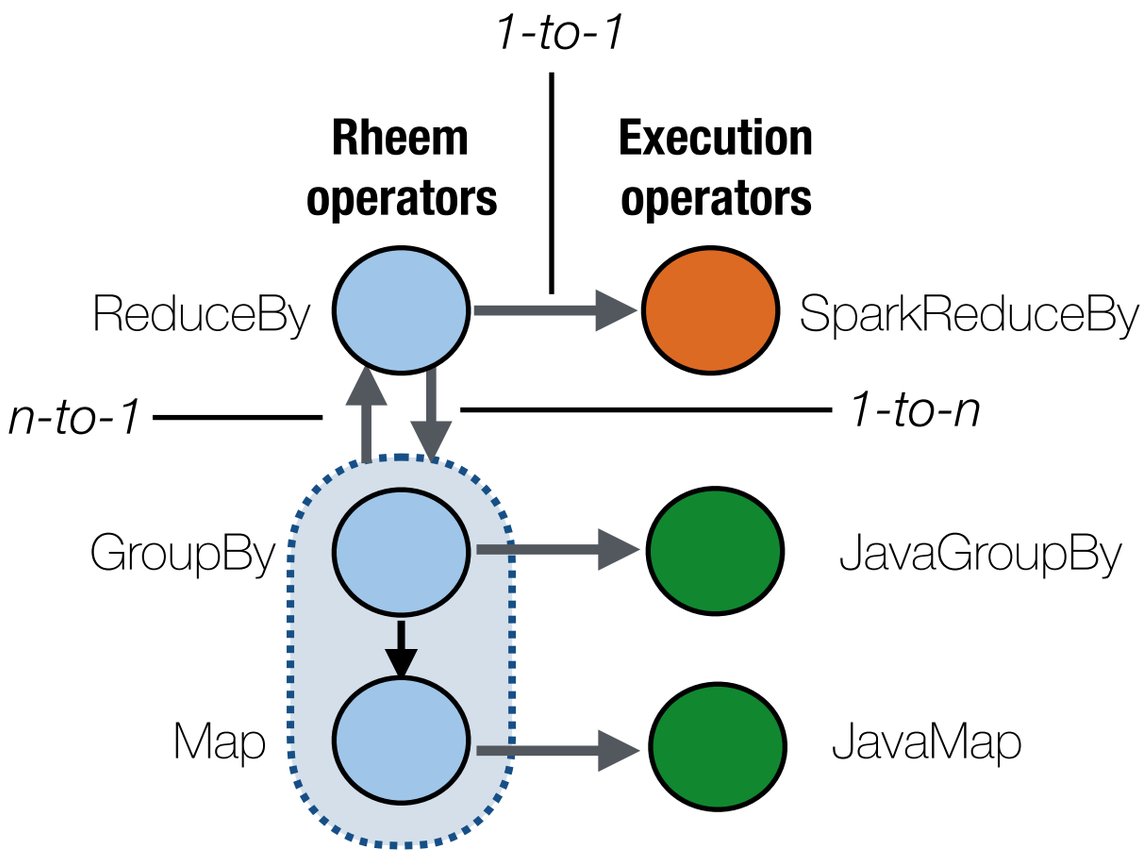
Graph-based mappings

Metadata-based mappings



Rheem & Ires

Graph-based mappings



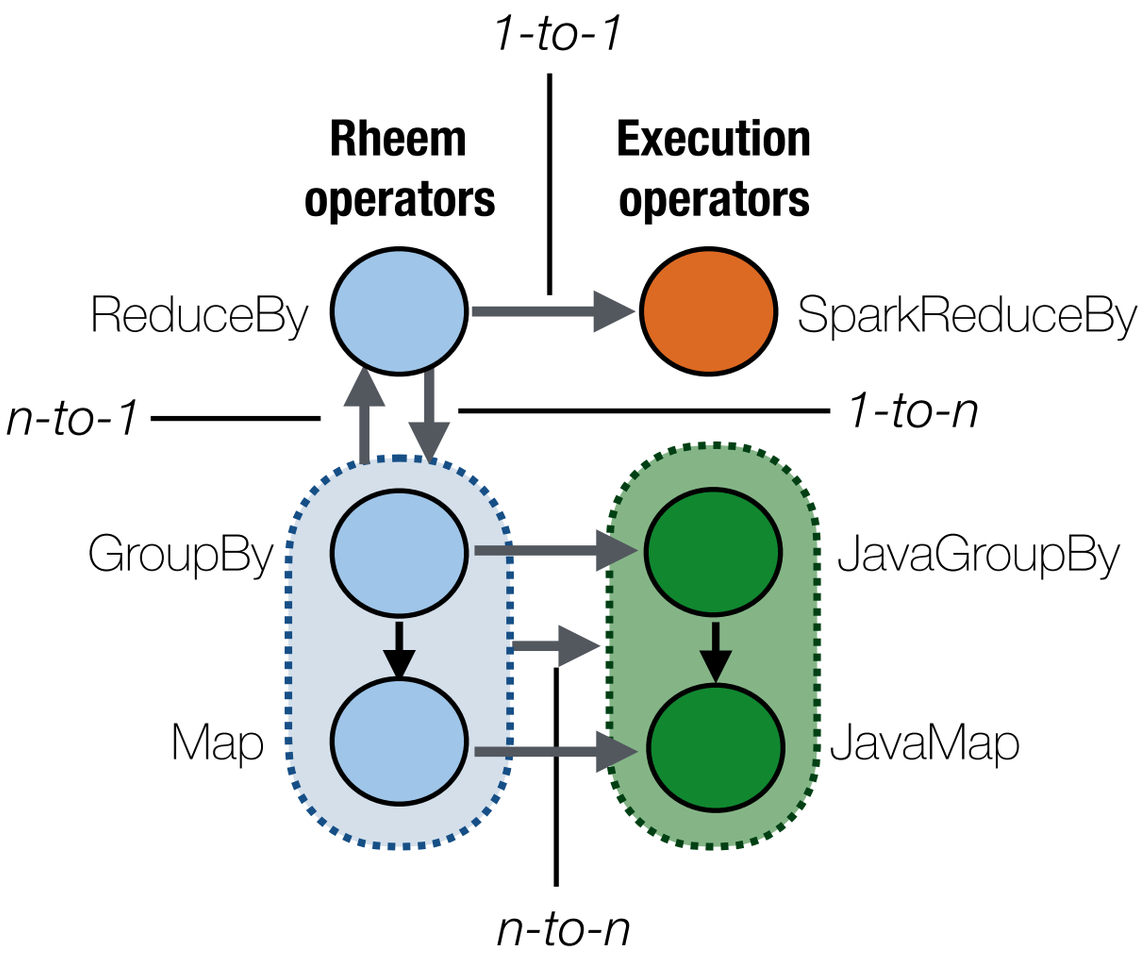
Metadata-based mappings

Road to Freedom in Big Data Analytics. EDBT 2016

Mix 'n' Match Multi-Engine Analytics. IEEE BigData 2017

Rheem & Ires

Graph-based mappings



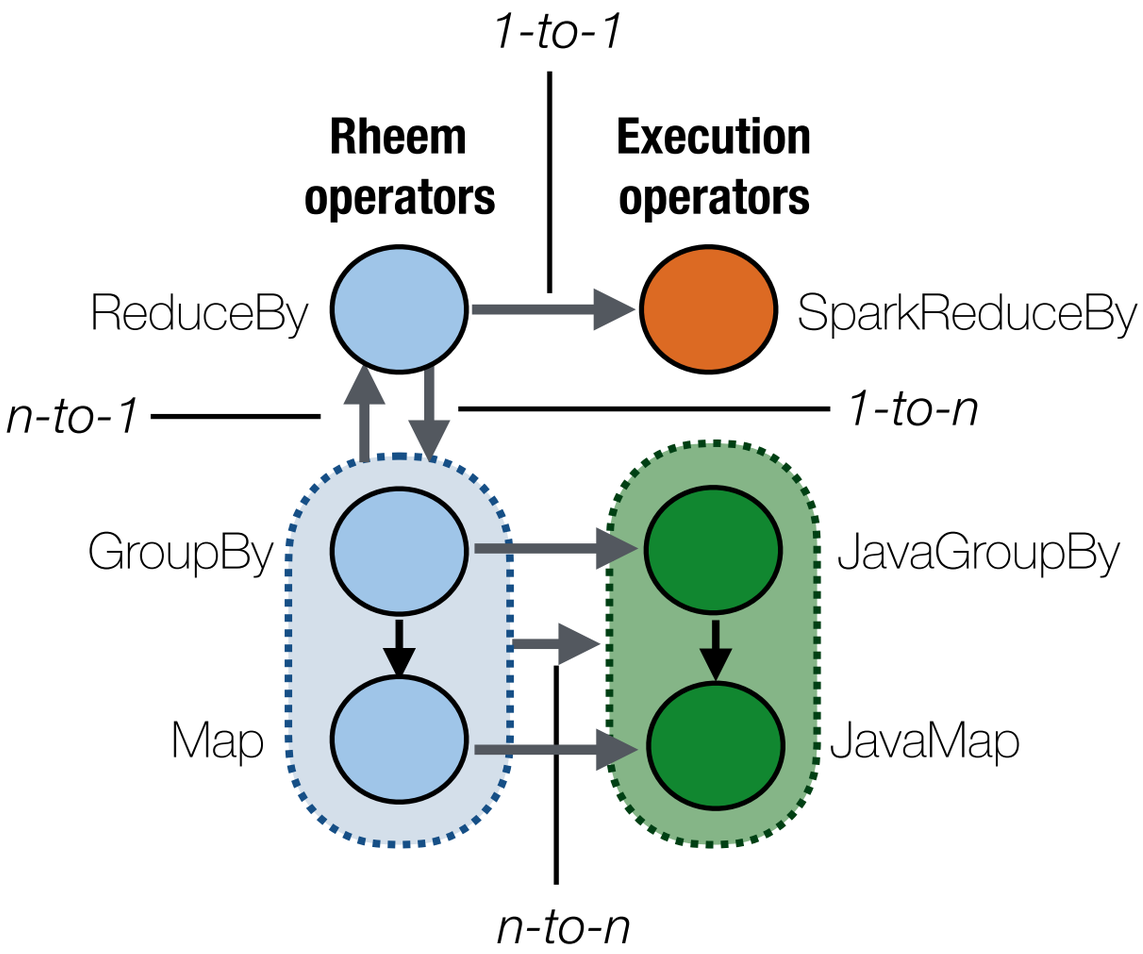
Metadata-based mappings

Road to Freedom in Big Data Analytics. EDBT 2016

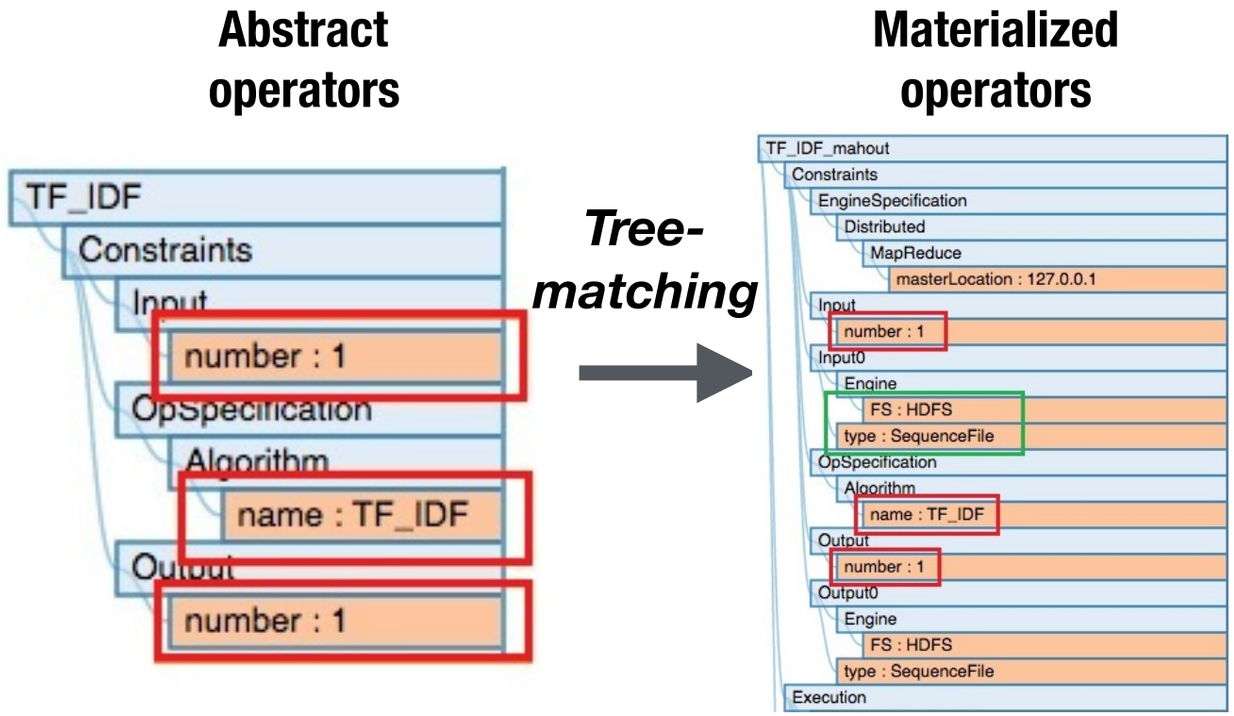
Mix 'n' Match Multi-Engine Analytics. IEEE BigData 2017

Rheem & Ires

Graph-based mappings



Metadata-based mappings



Road to Freedom in Big Data Analytics. EDBT 2016

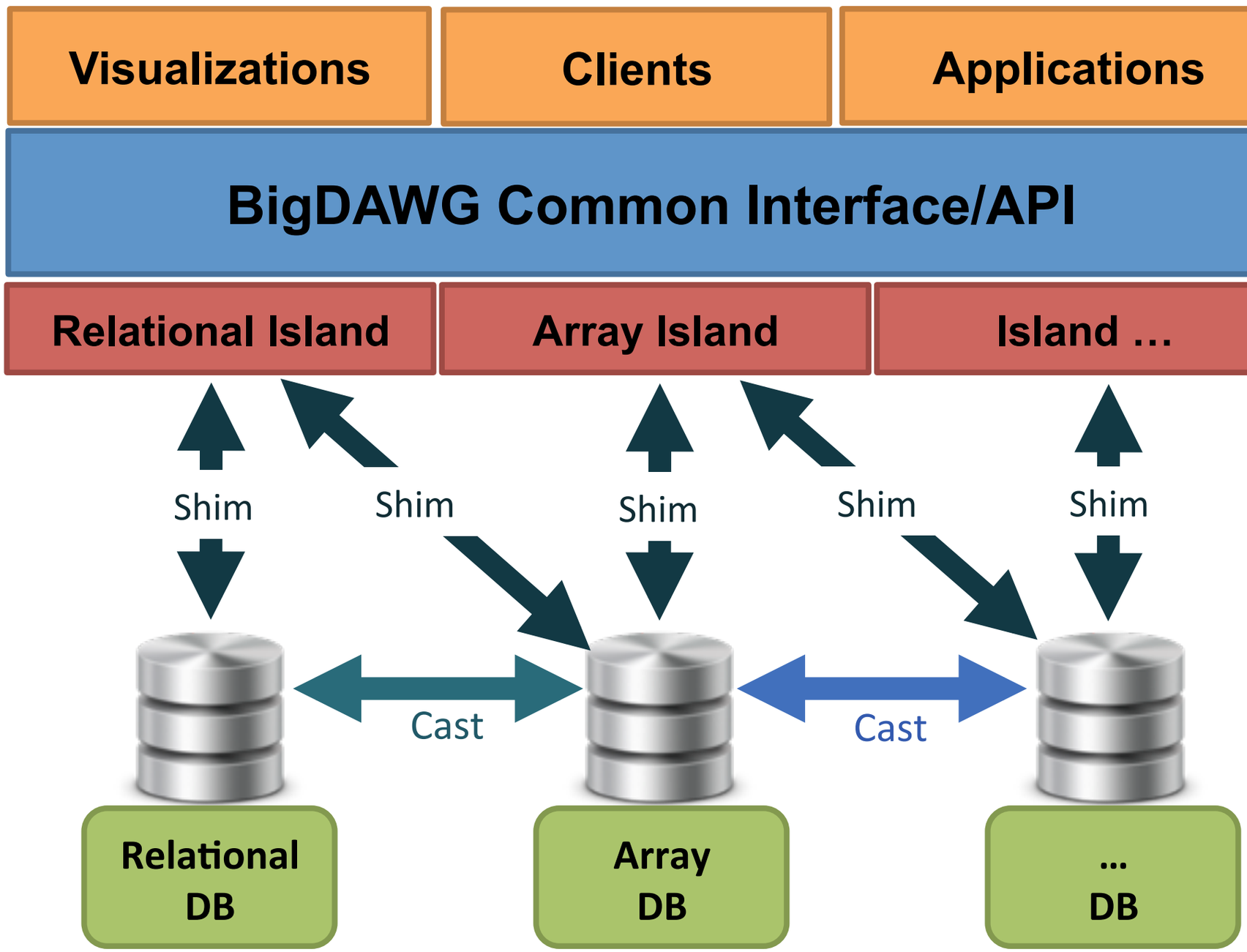
Mix 'n' Match Multi-Engine Analytics. IEEE BigData 2017

Decoupling Features

	Processing Model	CP Granularity	Mappings
Musketeer	Dataflow DAG + Loop	Operator	Code templates
Rheem	Dataflow DAG + Loop	Operator	Graph-based
Ires	Workflow graph	Task	Metadata-based

BigDawg

Two-tier decoupling



The BigDawg Polystore System and Architecture. IEEE High Performance Extreme Computing 2016.

Decoupling Features

	Processing Model	CP Granularity	Mappings
Musketeer	Dataflow DAG + Loop	Operator	Code templates
Rheem	Dataflow DAG + Loop	Operator	Graph-based
Ires	Workflow graph	Task	Metadata-based
BigDawg	Query Plan Tree	Subquery	Shims

Decoupling Features

	Processing Model	CP Granularity	Mappings
Musketeer	Dataflow DAG + Loop	Operator	Code templates
Rheem	Dataflow DAG + Loop	Operator	Graph-based
Ires	Workflow graph	Task	Metadata-based
BigDawg	Query Plan Tree	Subquery	Shims
Myria	Extended relational	Operator	Rewrite rules
QoX	Dataflow graph	Task	?
Cyclops	Windowed aggregation	Operator	?
Apache Beam	Dataflow DAG	Pipeline	?
CloudMdsQL	Query Plan Tree	Subquery	

Challenges

1 Decoupling Applications

4 Extensibility

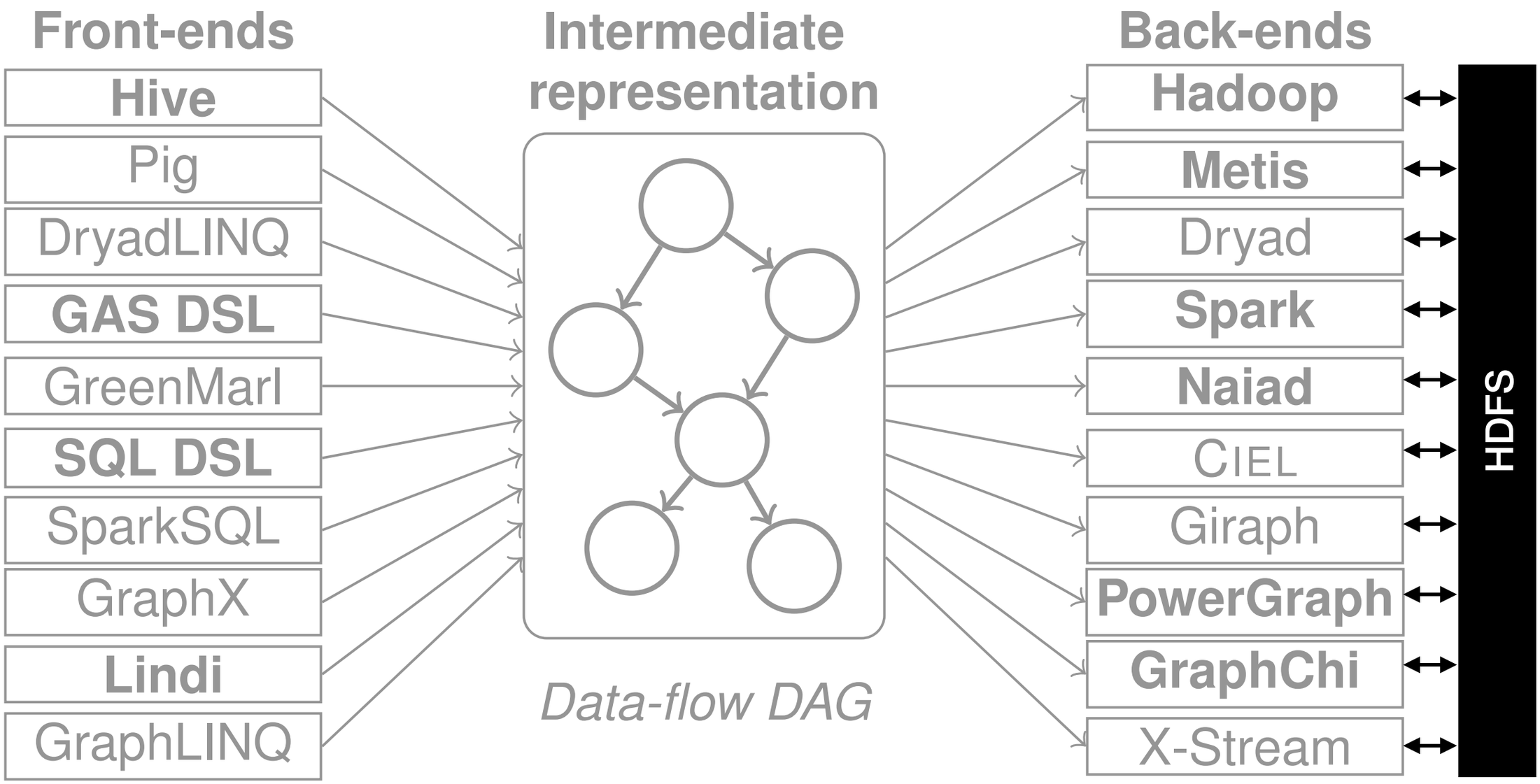
3

Automatic Cross-Platform Data Processing

2 Data Movement

Musketeer

Communication ***always*** via ***HDFS***

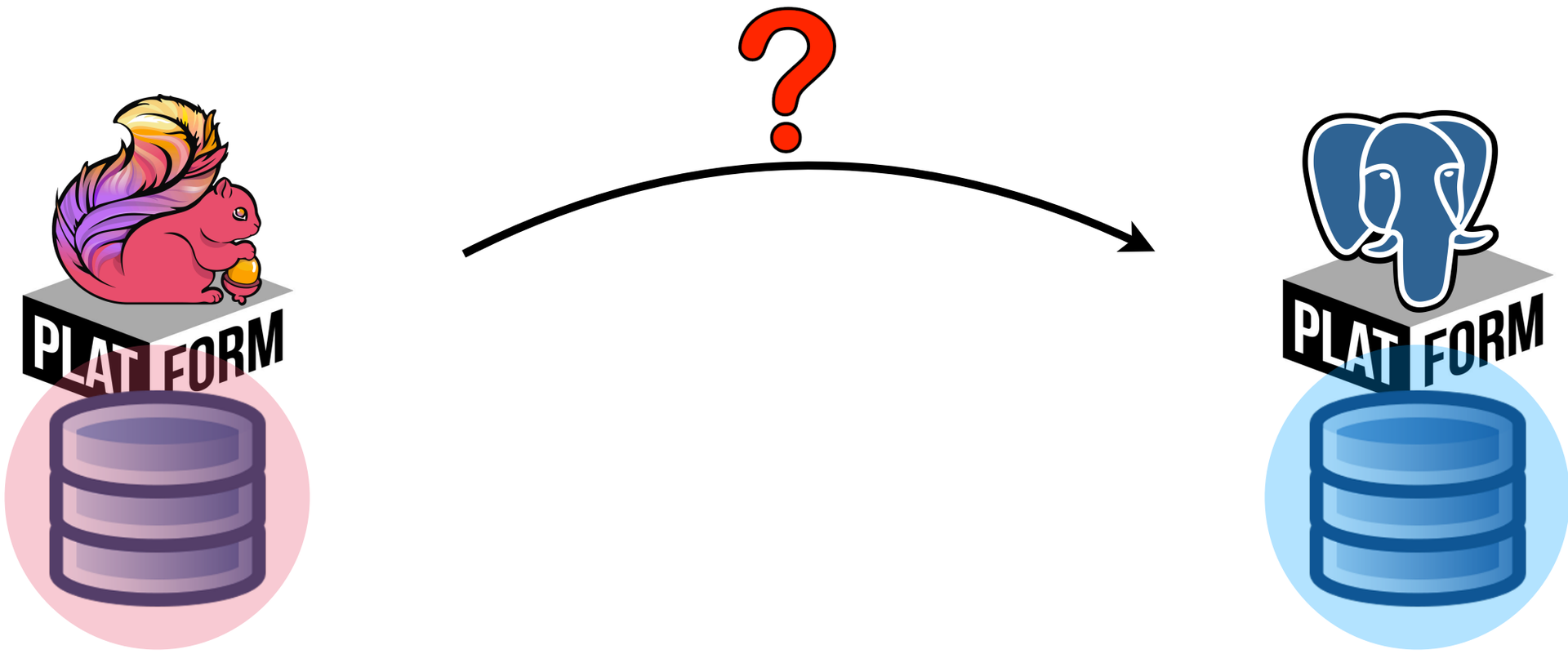


Musketeer: all for one, one for all in data processing systems. EuroSys'15.

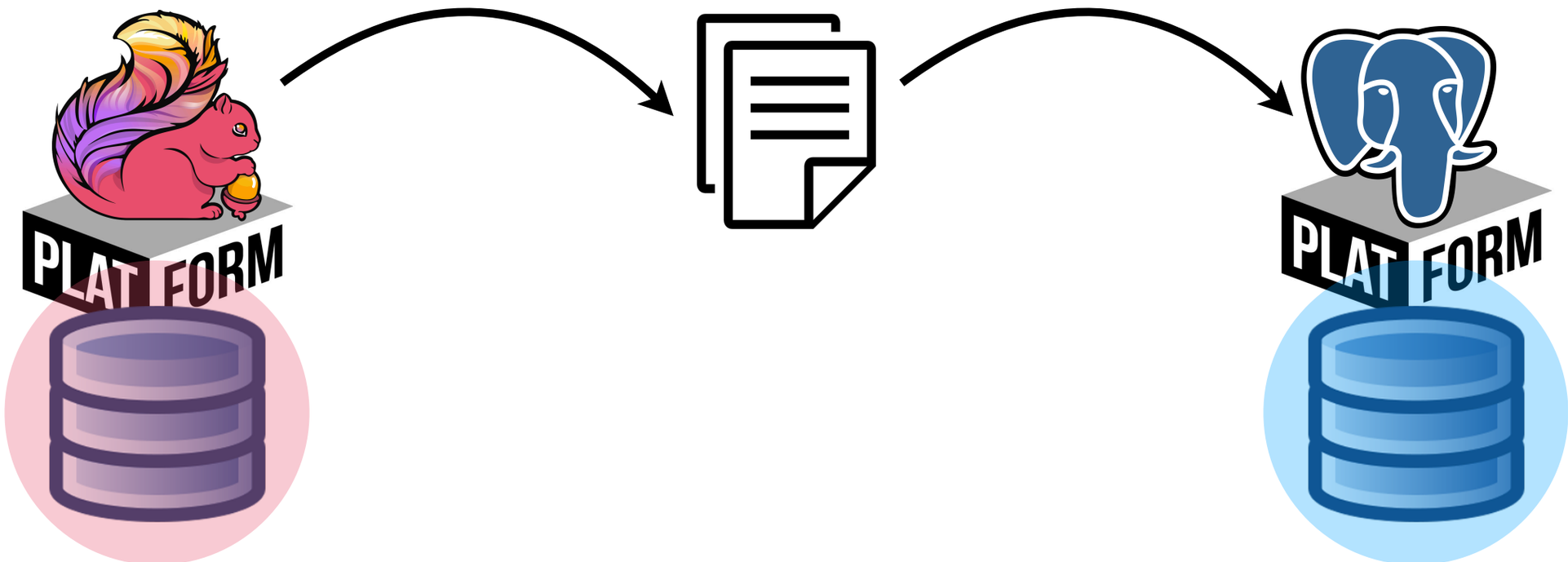
Data Movement Features

	Connection	Transfer Means
Musketeer	Direct	HDFS Files

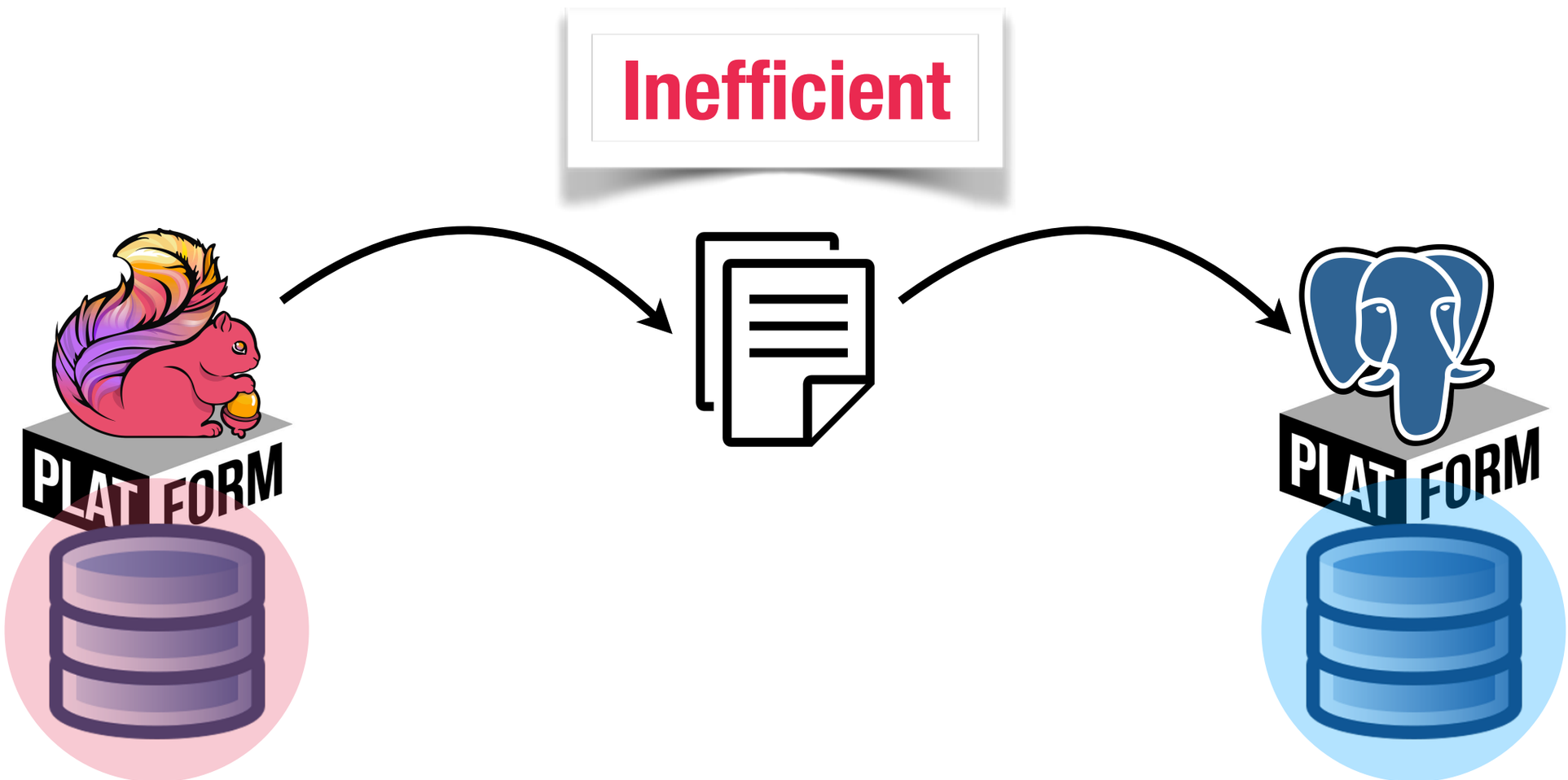
Connecting two platforms



Connecting two platforms

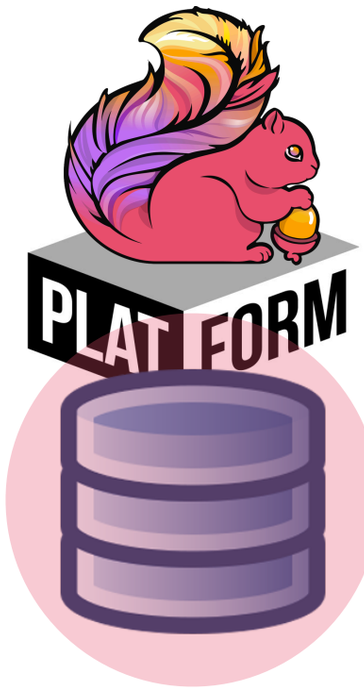


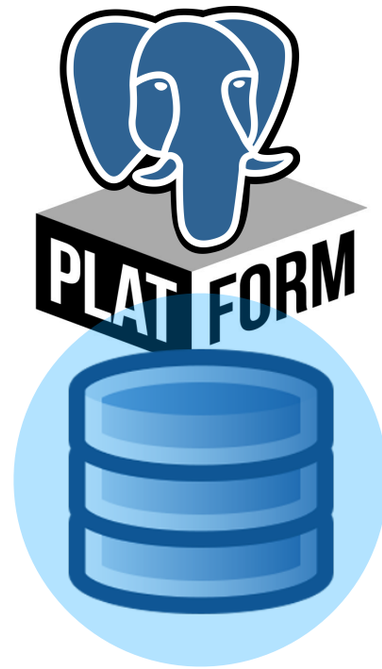
Connecting two platforms



PipeGen by Myria

Binary data transfer via generated ***data pipes***

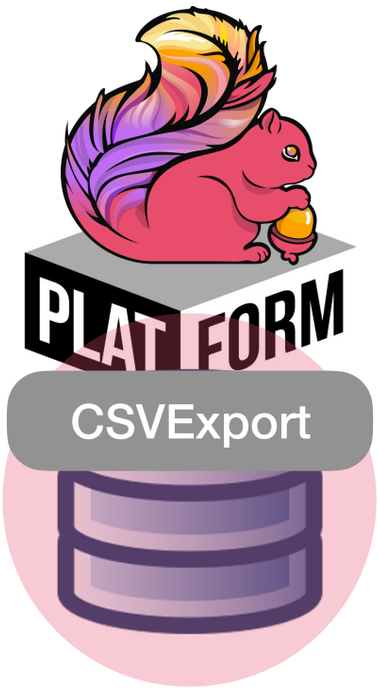


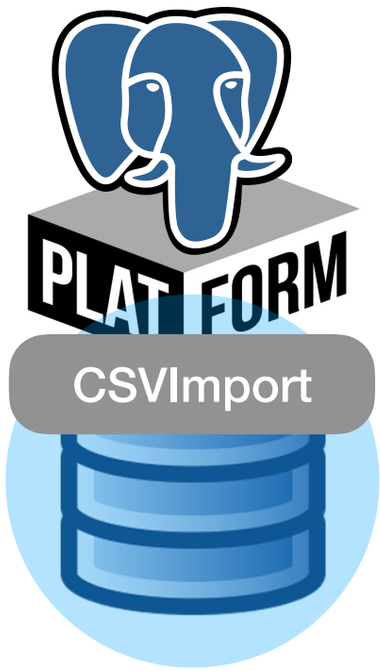


PipeGen by Myria

Binary data transfer via generated ***data pipes***

Same Import/Export Capabilities



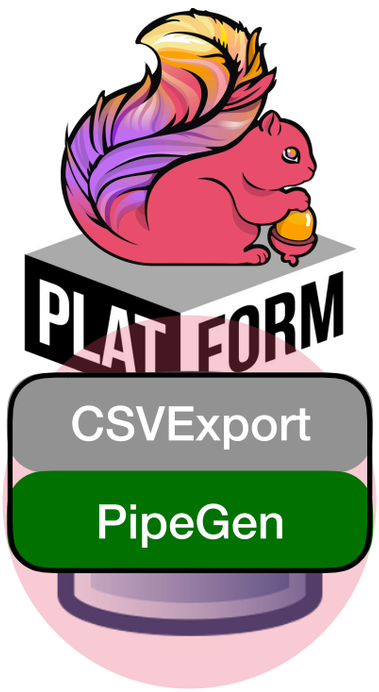


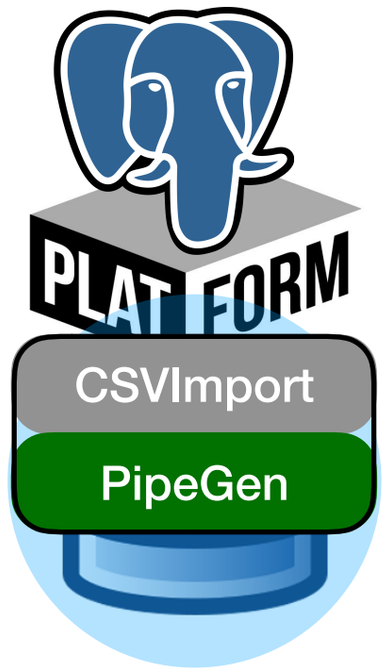
PipeGen: Data Pipe Generator for Hybrid Analytics. SoCC 2016.

PipeGen by Myria

Binary data transfer via generated ***data pipes***

Same Import/Export Capabilities





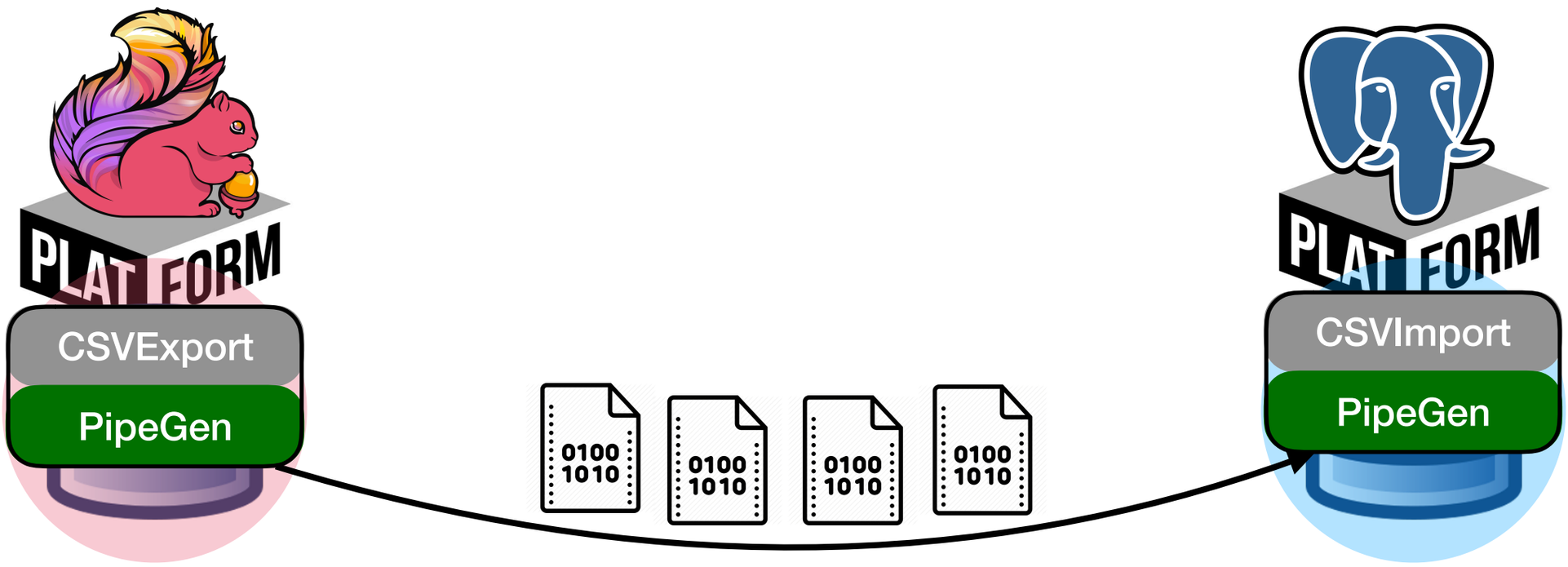
Program Analysis

PipeGen: Data Pipe Generator for Hybrid Analytics. SoCC 2016.

PipeGen by Myria

Binary data transfer via generated ***data pipes***

Same Import/Export Capabilities



Program Analysis

PipeGen: Data Pipe Generator for Hybrid Analytics. SoCC 2016.

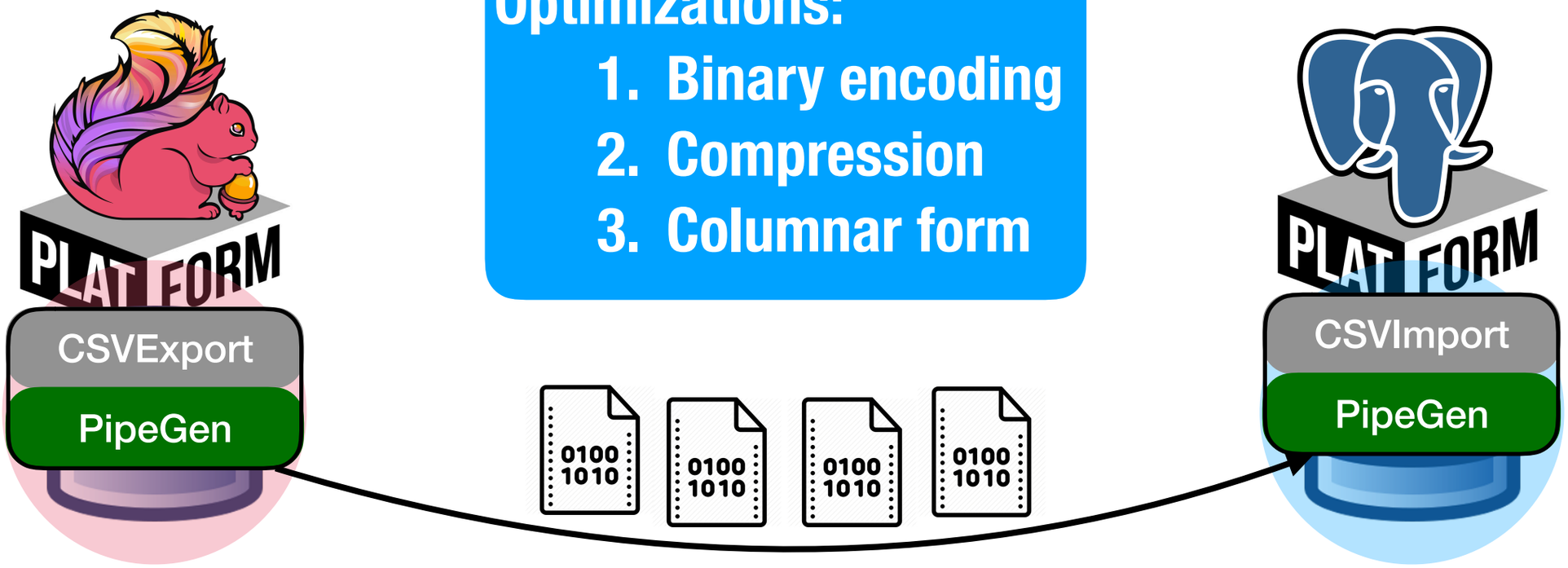
PipeGen by Myria

Binary data transfer via generated ***data pipes***

Same Import/Export Capabilities

Optimizations:

1. **Binary encoding**
2. **Compression**
3. **Columnar form**



Program Analysis

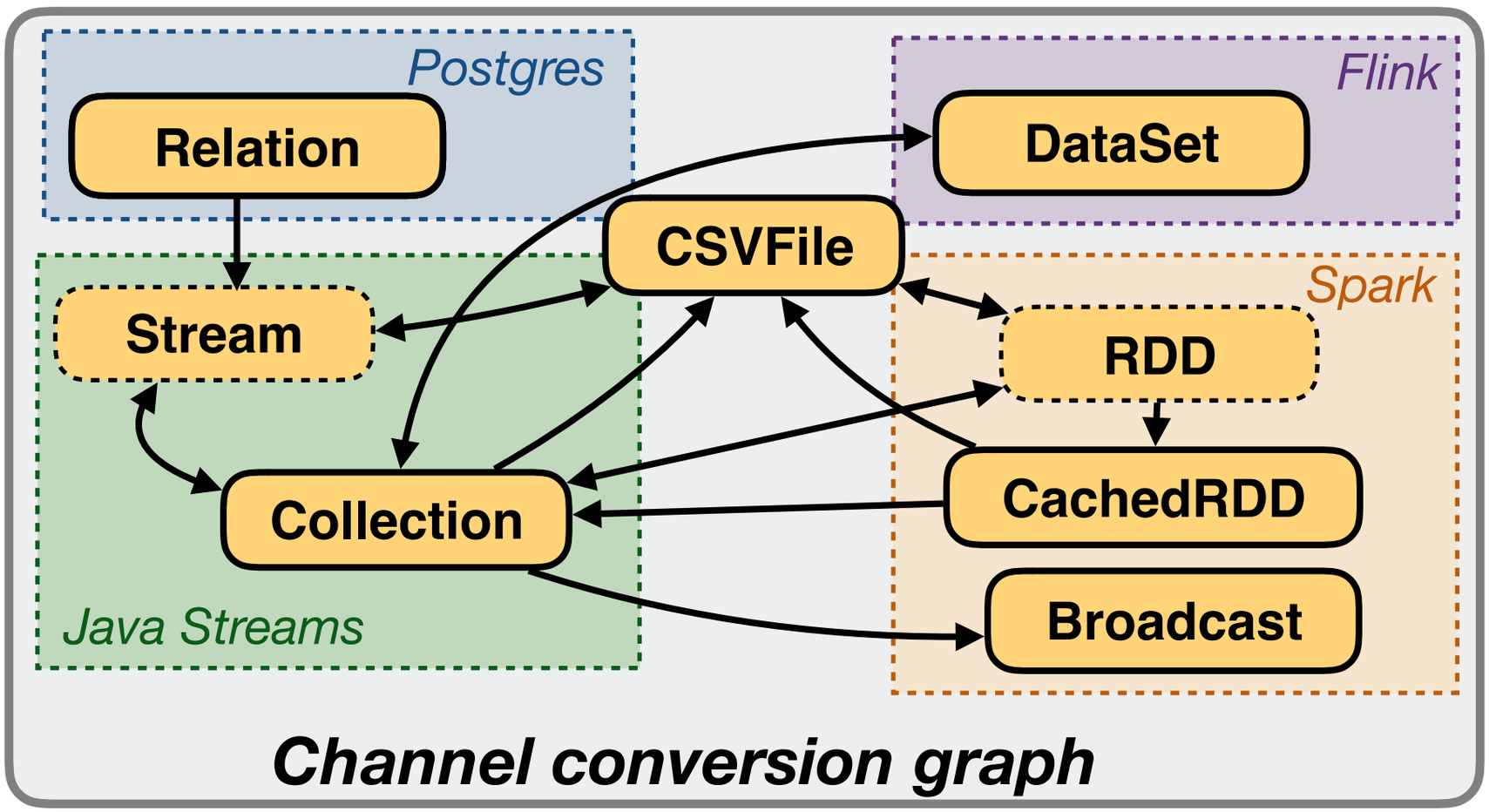
PipeGen: Data Pipe Generator for Hybrid Analytics. SoCC 2016.

Data Movement Features

	Connection	Transfer Means
Musketeer	Direct	HDFS Files
Myria	Direct	PipeGen

Channels Graph by Rheem

Graph-based data transfer representation

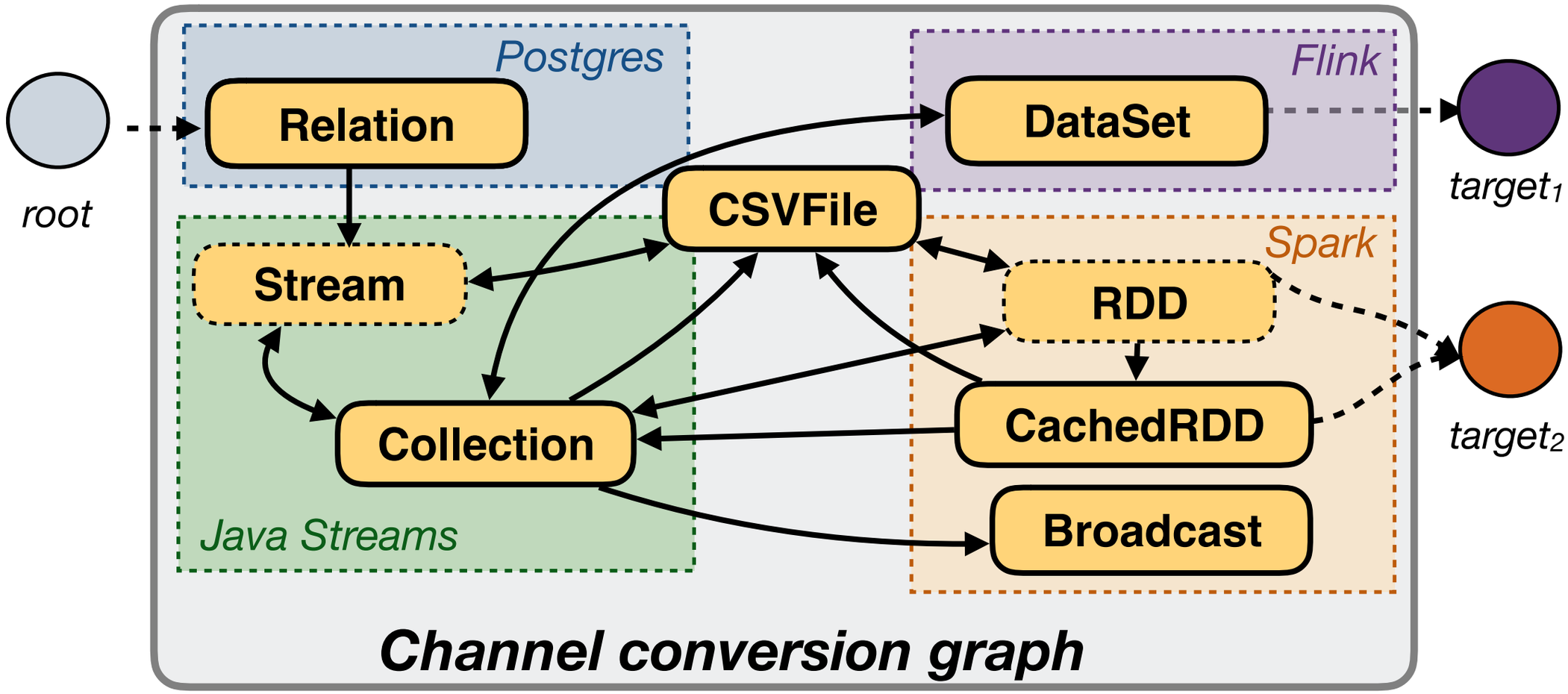


reusable channel

non-reusable channel

Channels Graph by Rheem

Graph-based data transfer representation

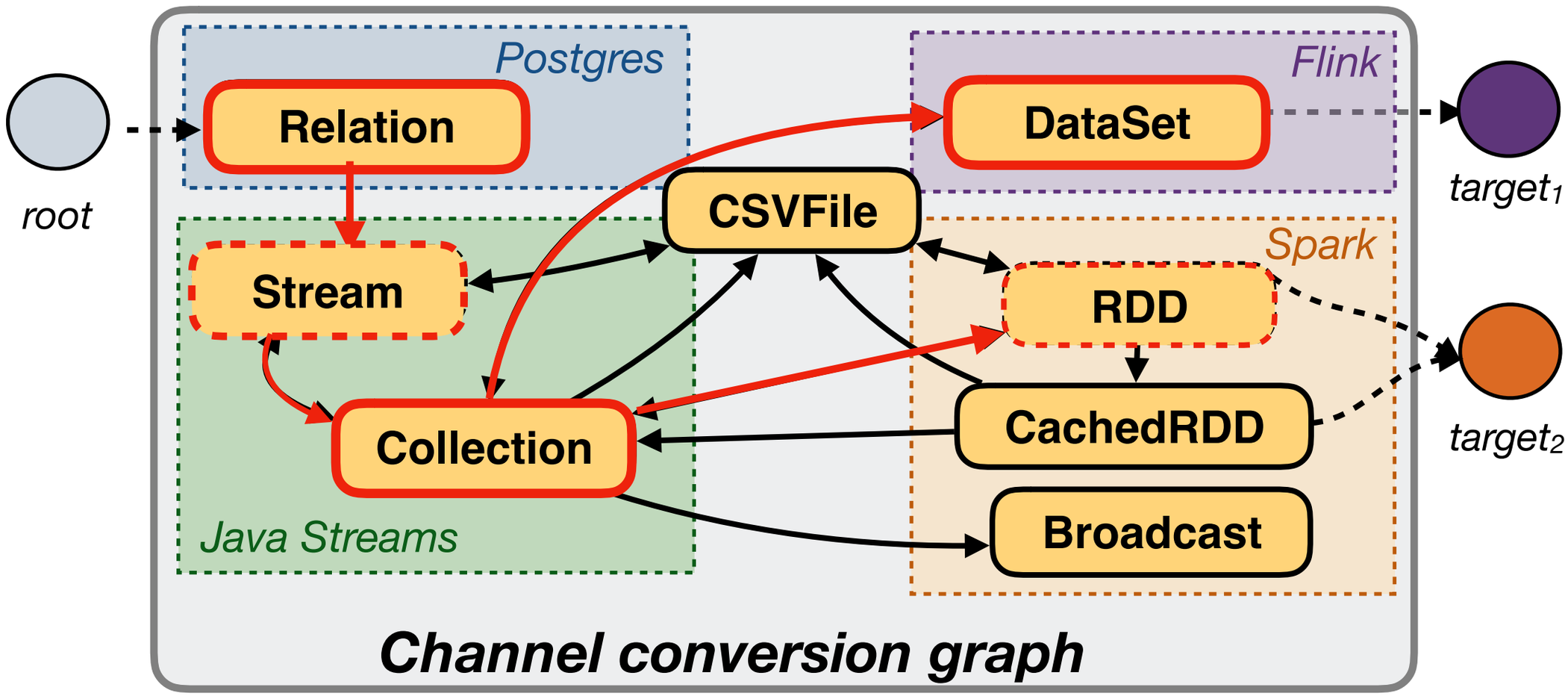


reusable channel

non-reusable channel

Channels Graph by Rheem

Graph-based data transfer representation



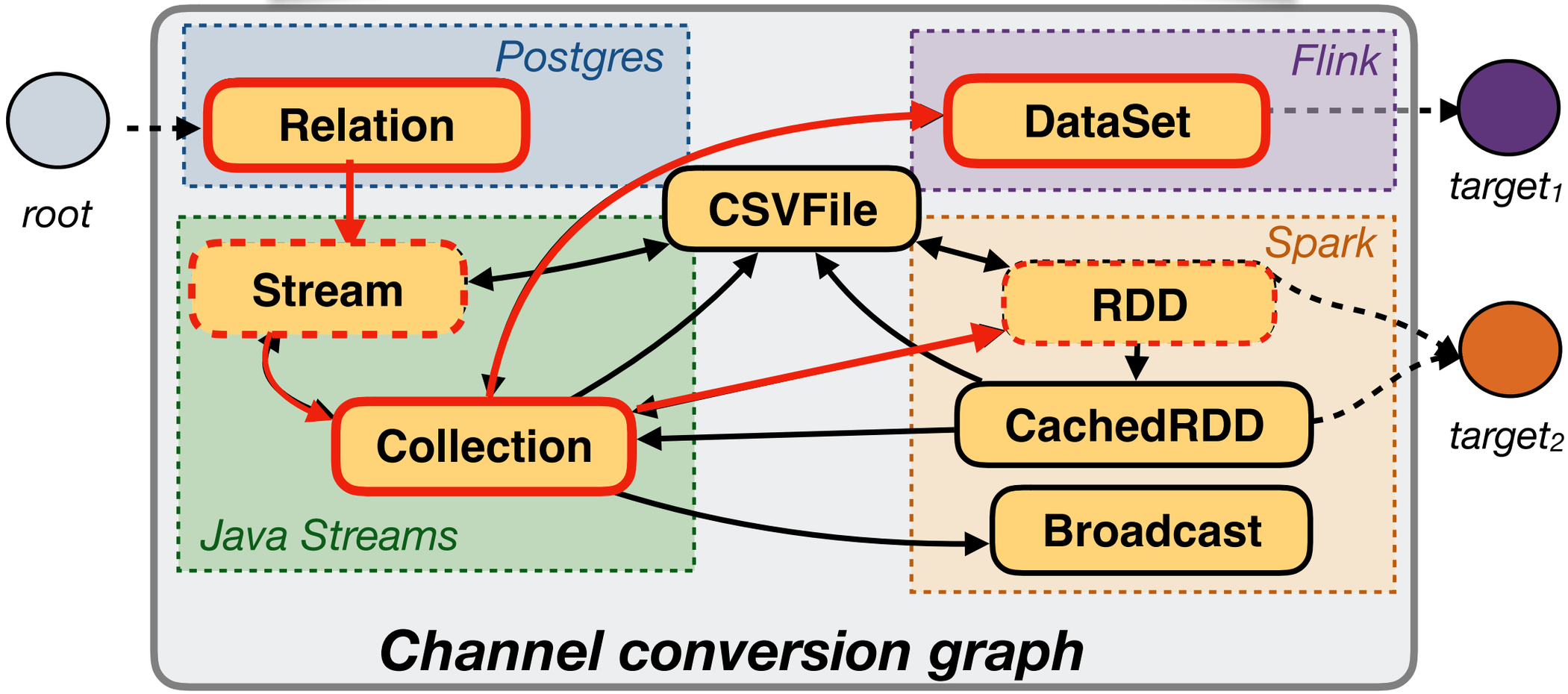
reusable channel

non-reusable channel

Channels Graph by Rheem

Graph-based data transfer representation

Group Steiner Tree Problem



reusable channel

non-reusable channel

Data Movement Features

	Connection	Transfer Means
Musketeer	Direct	HDFS Files
Myria	Direct	PipeGen
Rheem	Graph	Any

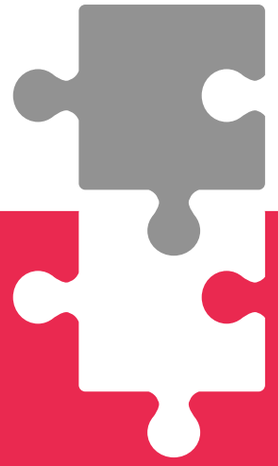
Data Movement Features

	Connection	Transfer Means
Musketeer	Direct	HDFS Files
Myria	Direct	PipeGen
Rheem	Graph	Any
Ires	Direct	Any
QoX	Graph	?
DBMS+	Direct	HDFS Files

Challenges

1 Decoupling Applications

4 Extensibility



3

Automatic Cross-Platform Data Processing

2 Data Movement

Decoupling Applications

Applications /Frontends

Hive | Crunch | MLlib | Mahout | Pig

Cross-Platform System

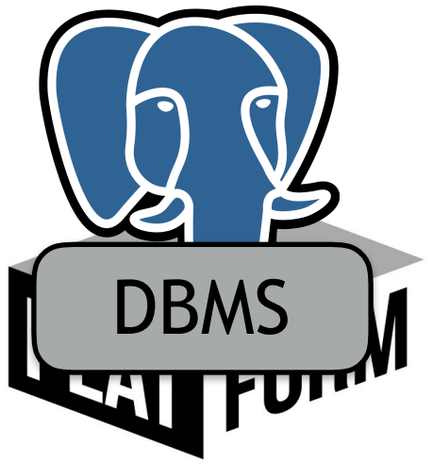
Processing Platforms

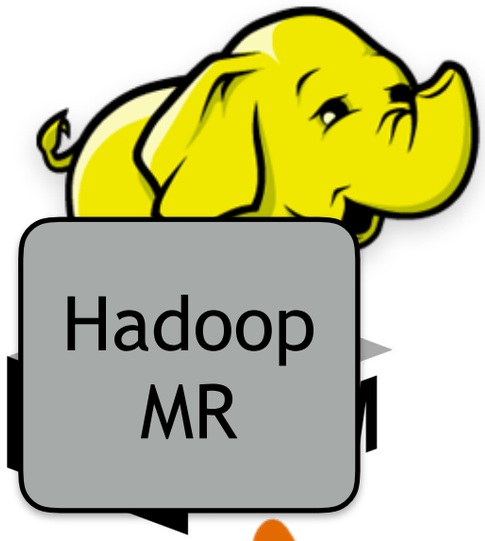
Hadoop MR | DBMS | Storm | Spark | Flink

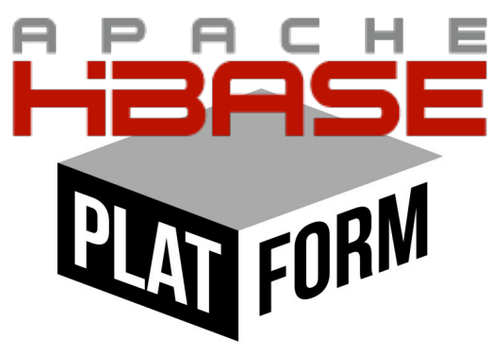
Storage Engines

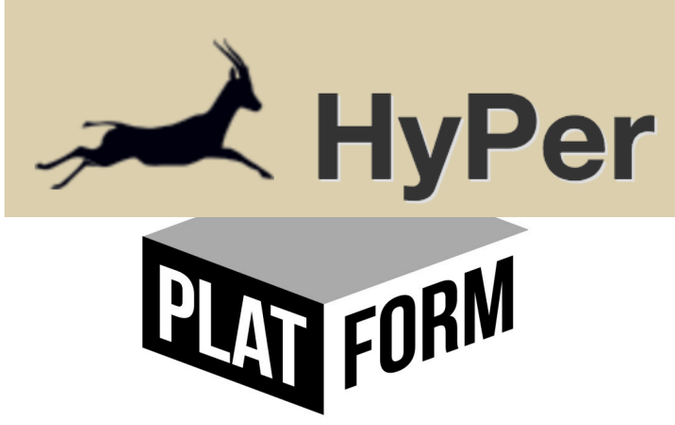
HDFS | S3 | Local FS

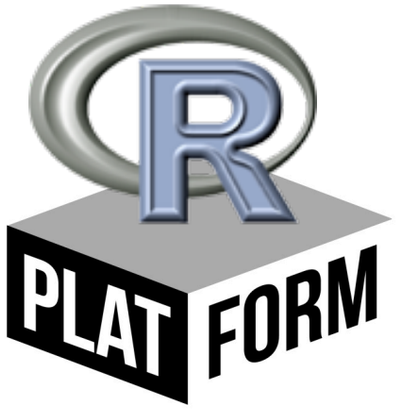
Which Platforms to Use?

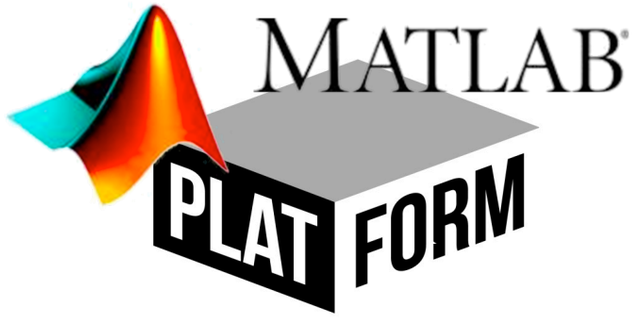


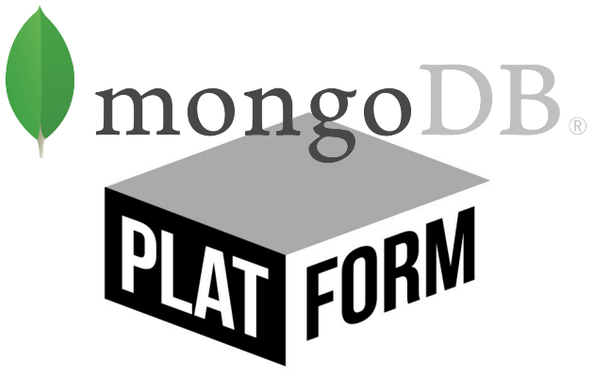












Which Platforms to Use?

PLAT FORM

PLAT FORM

PLAT FORM

APACHE HBASE

PLAT FORM

APACHE Spark™

PLAT FORM

HyPer

PLAT FORM

R

PLAT FORM

MATLAB

PLAT FORM

mongoDB.

PLAT FORM

Cross-Platform Optimization

Cost-based

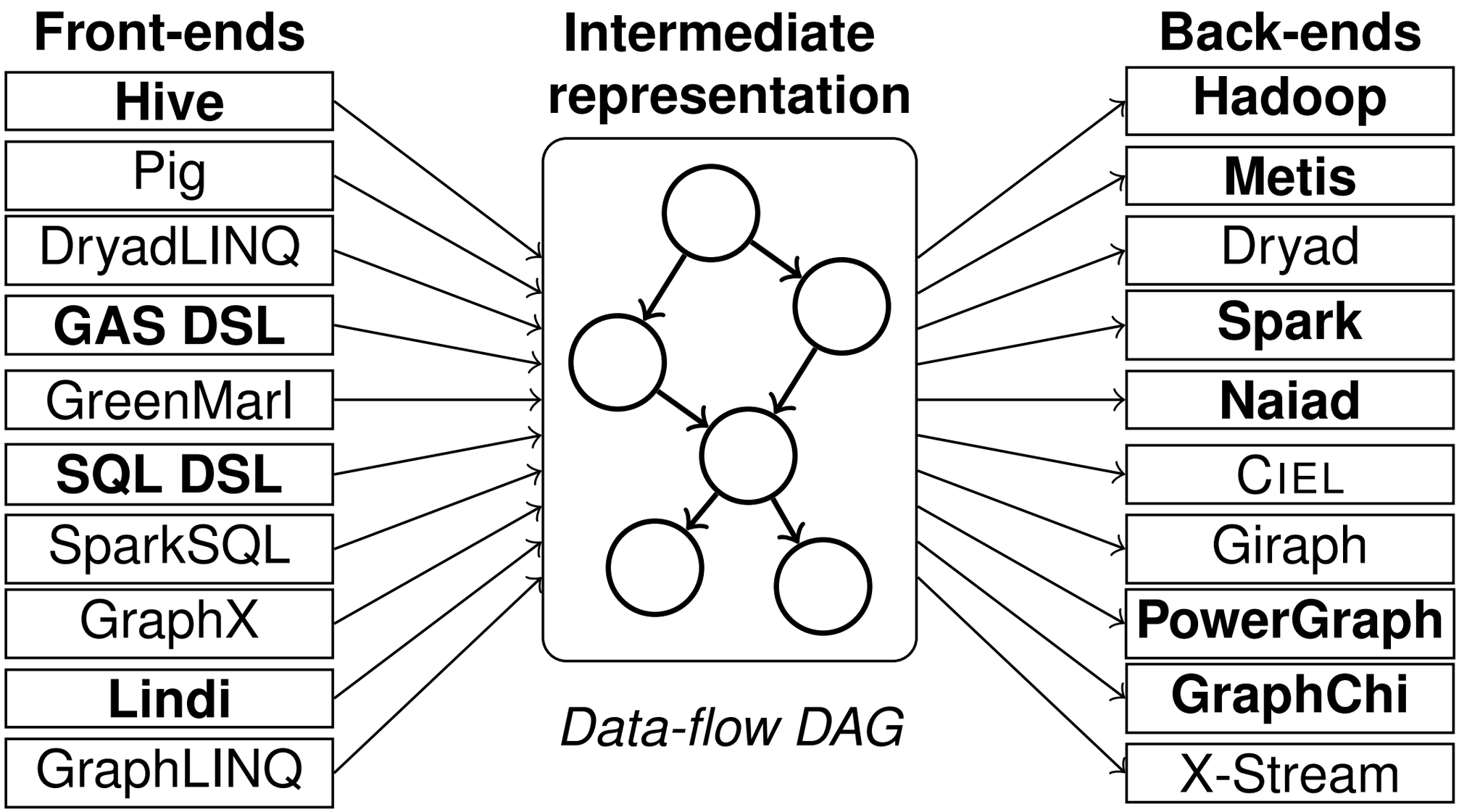
Rule-based

Cross-Platform Optimization

Cost-based

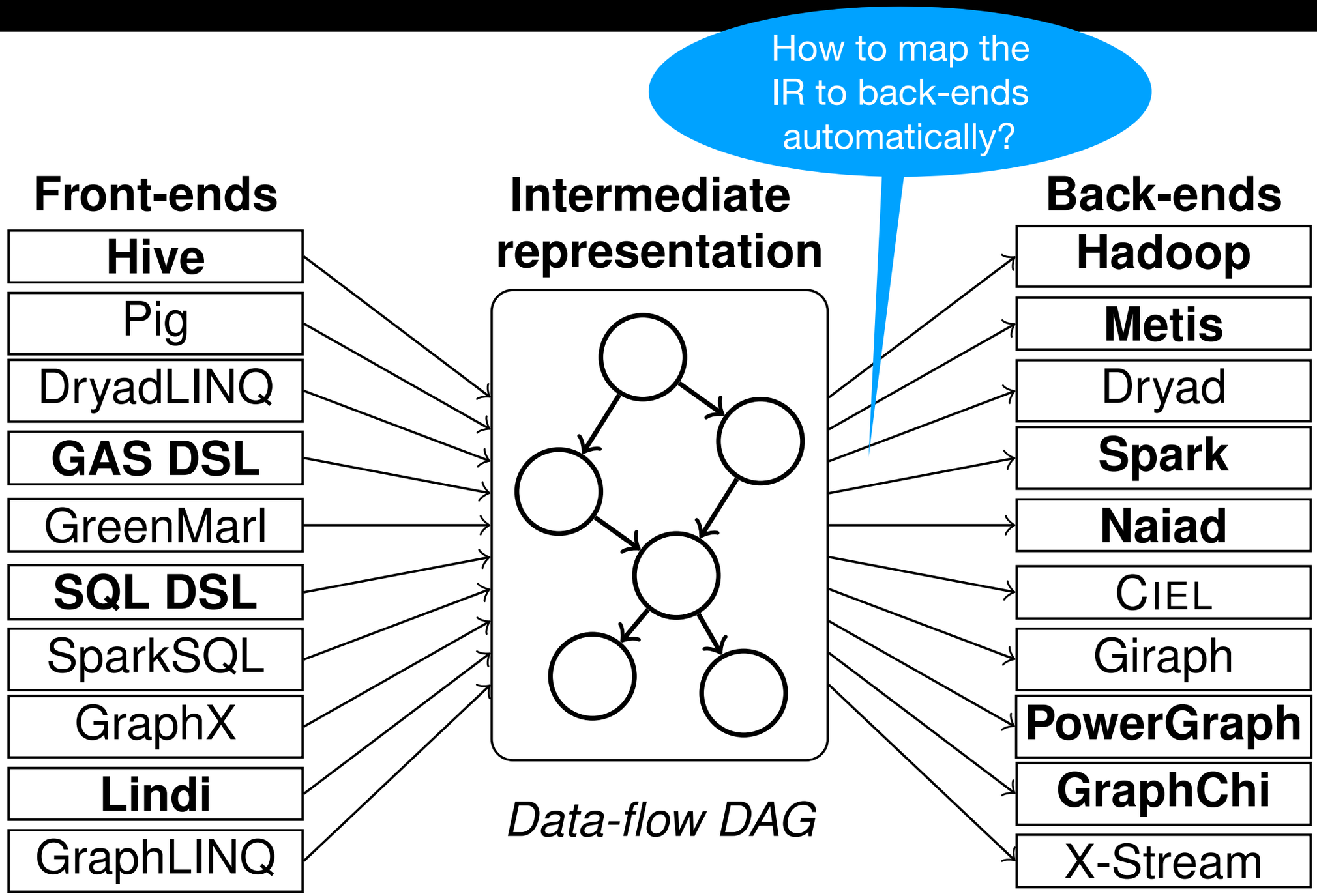
Rule-based

Musketeer



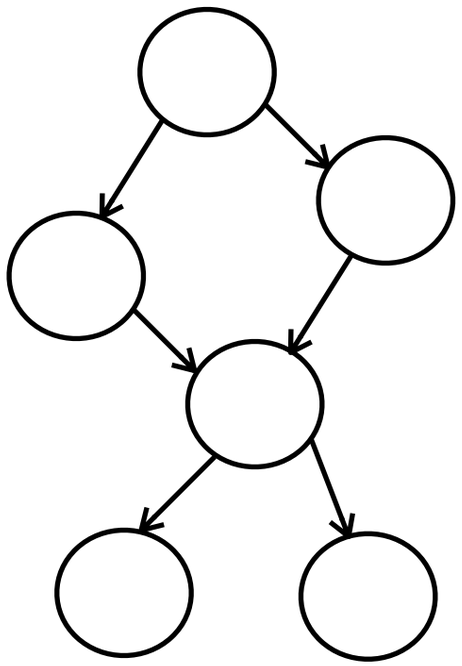
Musketeer: all for one, one for all in data processing systems. EuroSys'15.

Musketeer



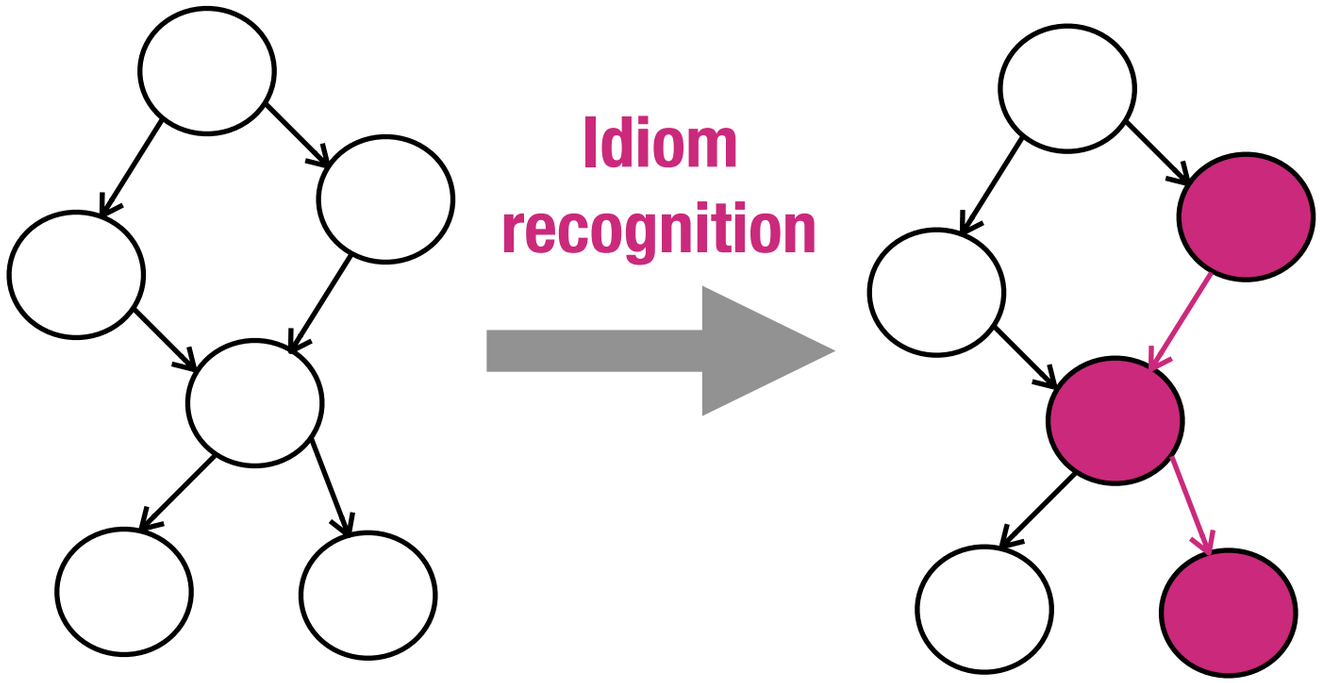
Musketeer: all for one, one for all in data processing systems. EuroSys'15.

Musketeer



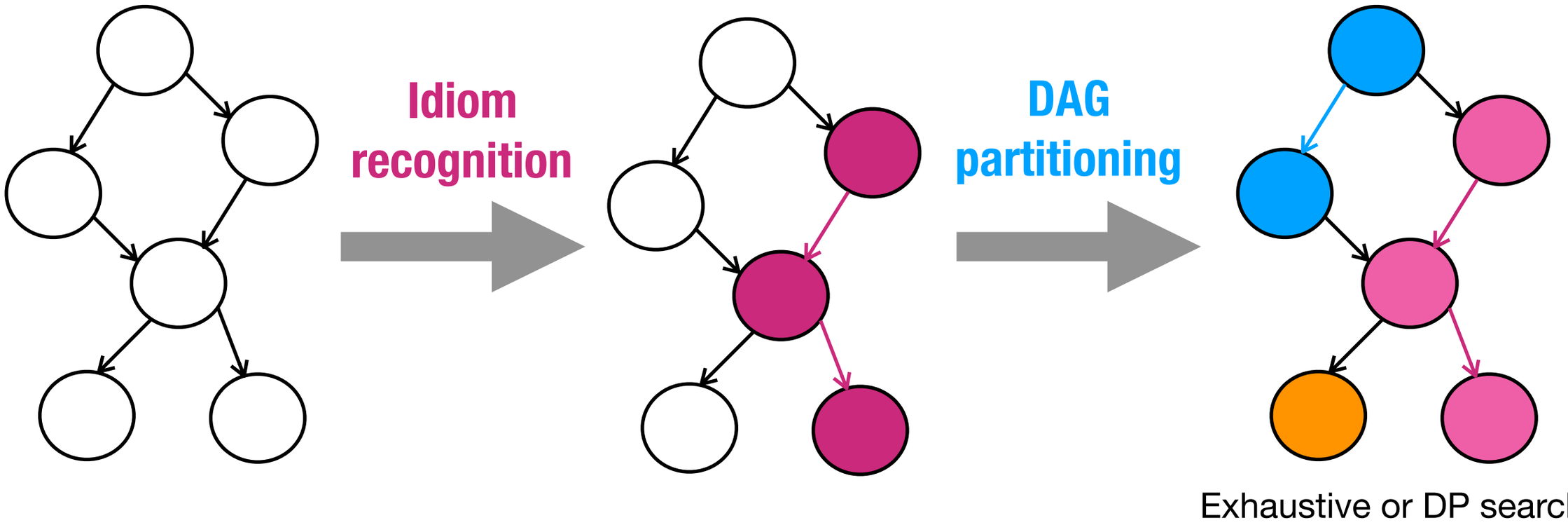
Musketeer: all for one, one for all in data processing systems. EuroSys'15.

Musketeer



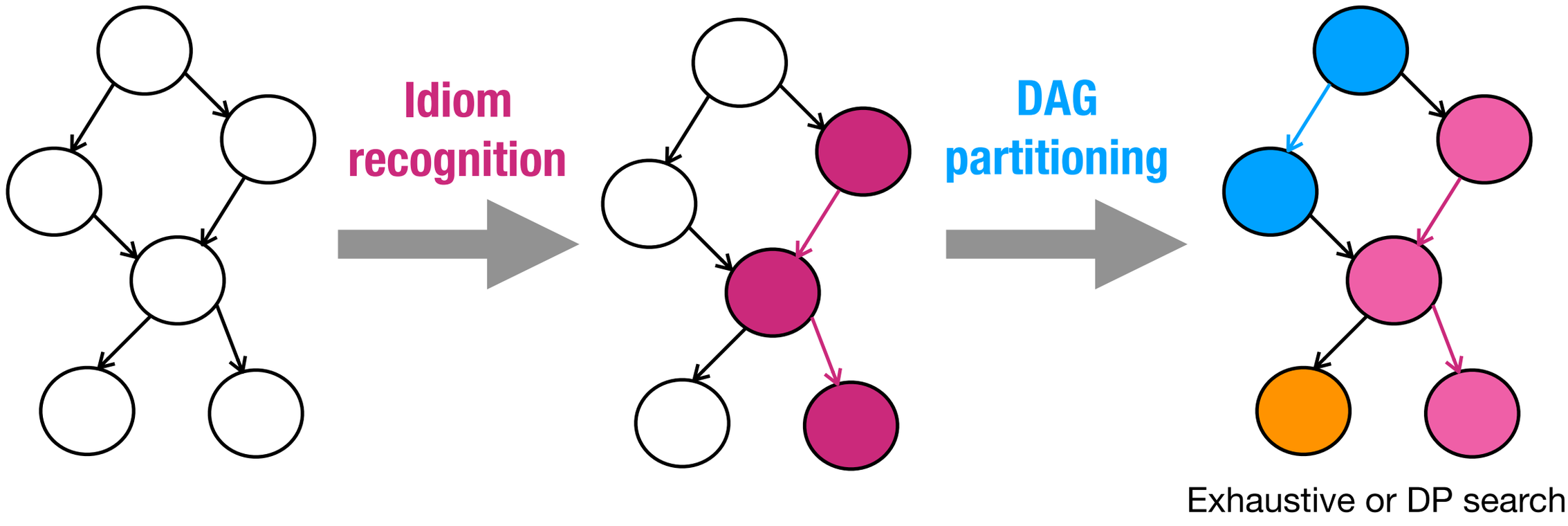
Musketeer: all for one, one for all in data processing systems. EuroSys'15.

Musketeer



Musketeer: all for one, one for all in data processing systems. EuroSys'15.

Musketeer



Analytical Cost Model

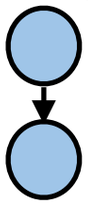
Musketeer: all for one, one for all in data processing systems. EuroSys'15.

CP Optimization Features

	Method	Enumeration	Cost Model	Data Movement Cost	User-Specified Platform
Musketeer	Cost-based	Exhaustive/DP	Analytical	No	No

Rheem

RHEEM plan
(platform-agnostic)



Rheem's optimizer

Rheem: enabling cross-platform data processing. TR 2018.

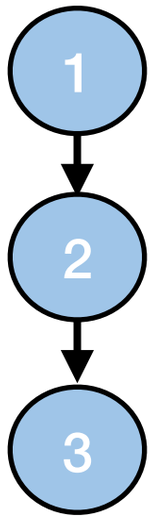
Rheem

Rheem: enabling cross-platform data processing. TR 2018.

Rheem

Plan Inflation

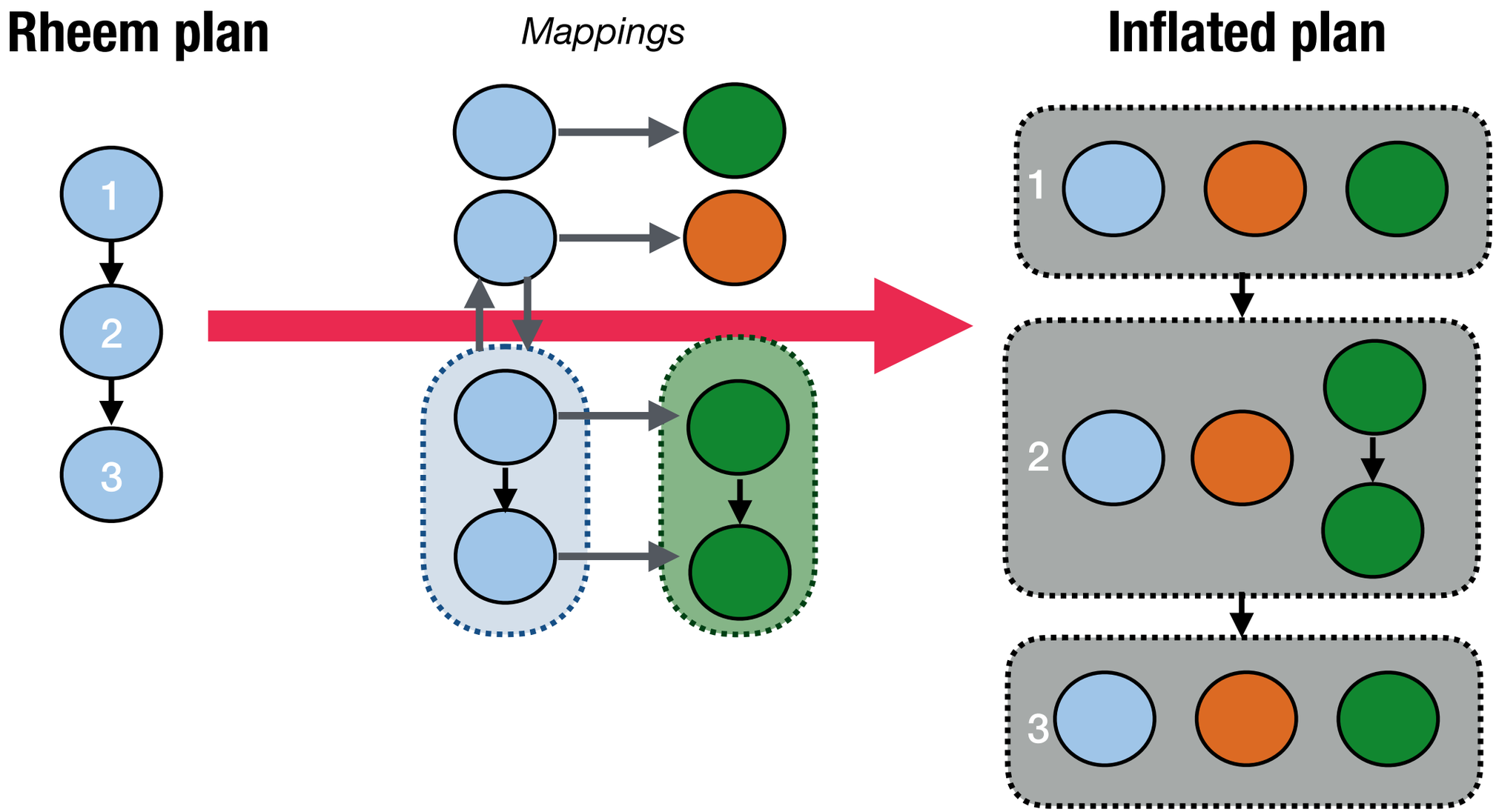
Rheem plan



Rheem: enabling cross-platform data processing. TR 2018.

Rheem

Plan Inflation

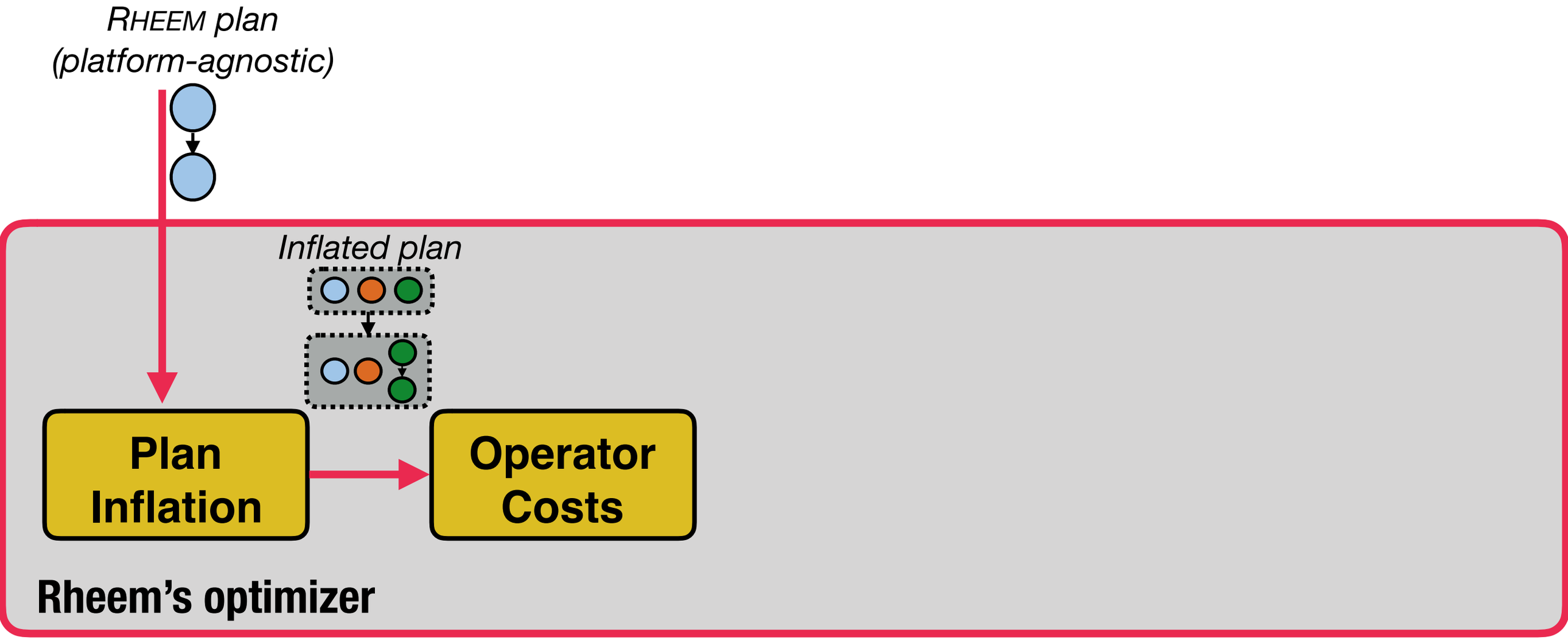


Rheem: enabling cross-platform data processing. TR 2018.

Rheem

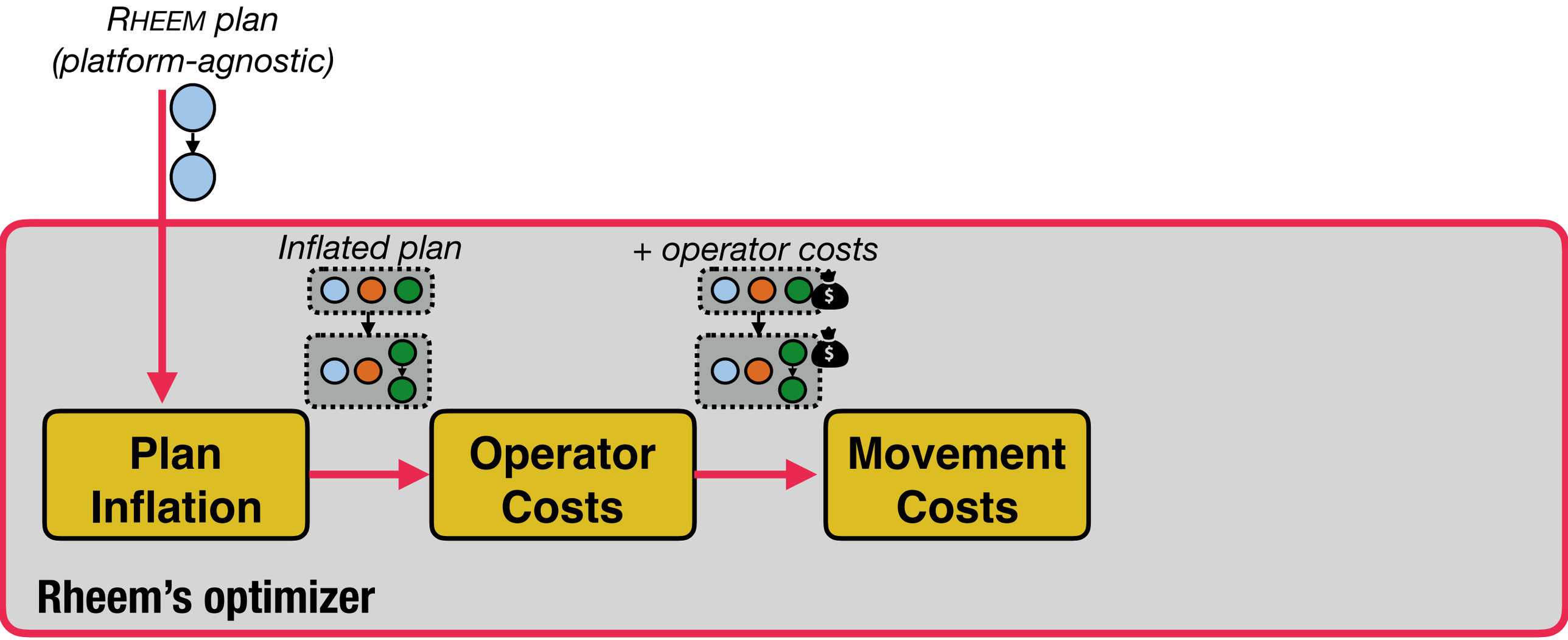
Rheem: enabling cross-platform data processing. TR 2018.

Rheem



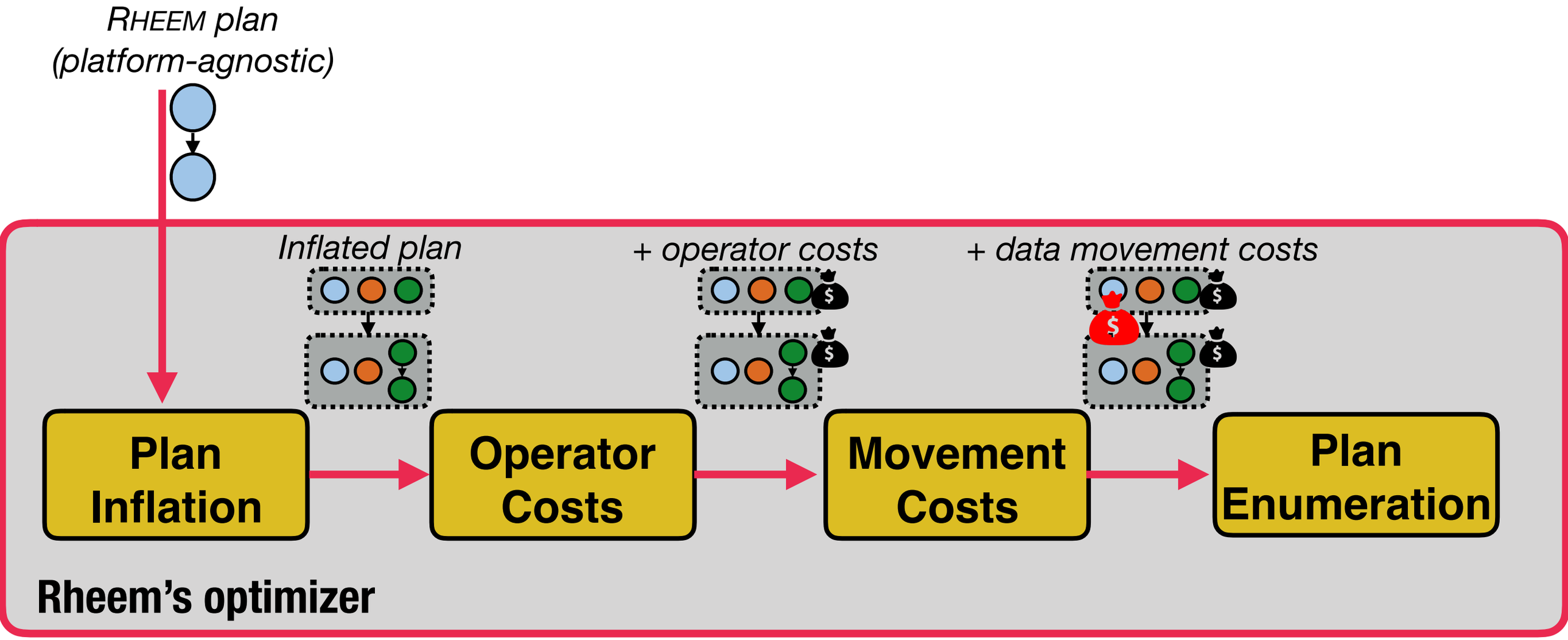
Rheem: enabling cross-platform data processing. TR 2018.

Rheem



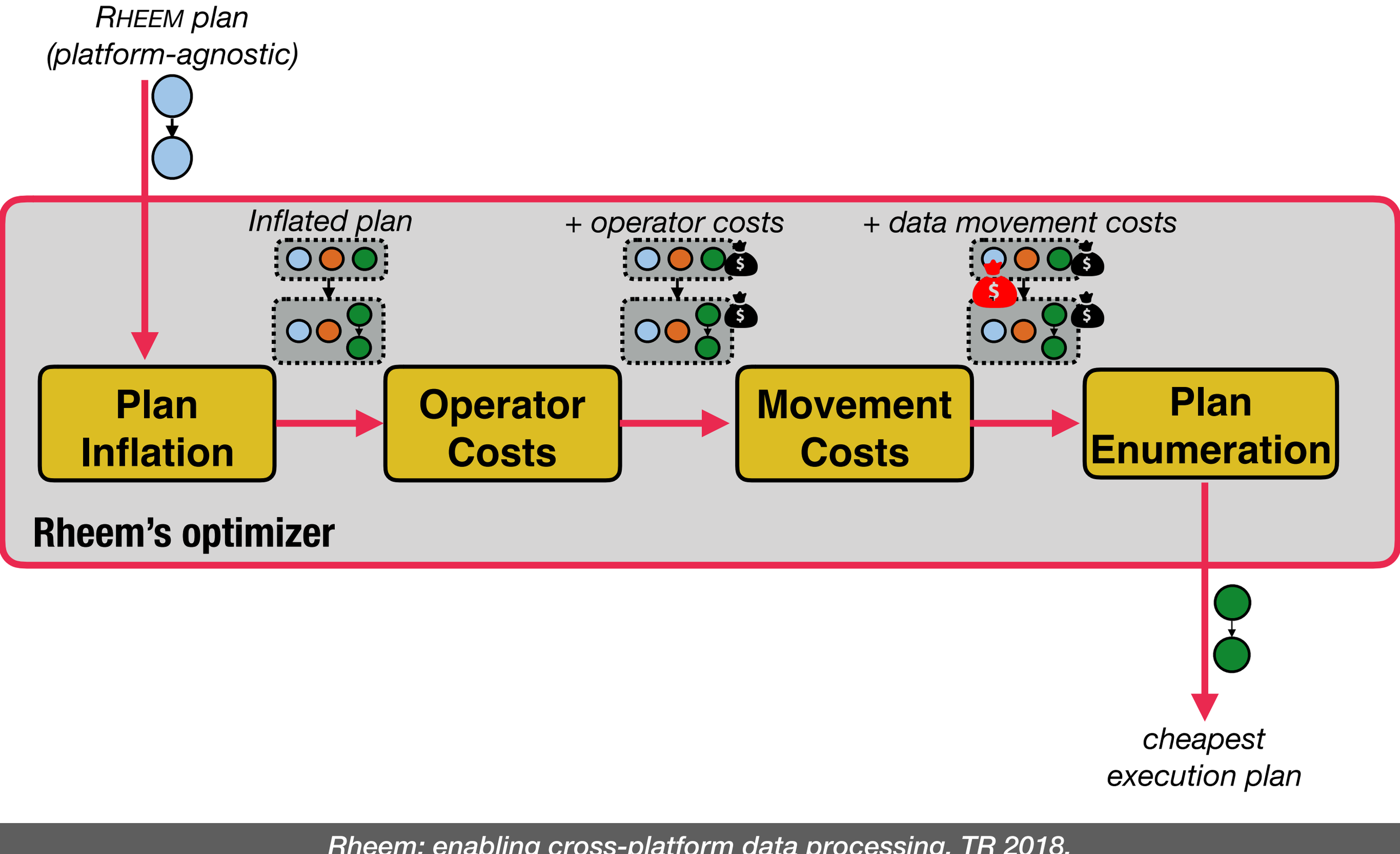
Rheem: enabling cross-platform data processing. TR 2018.

Rheem



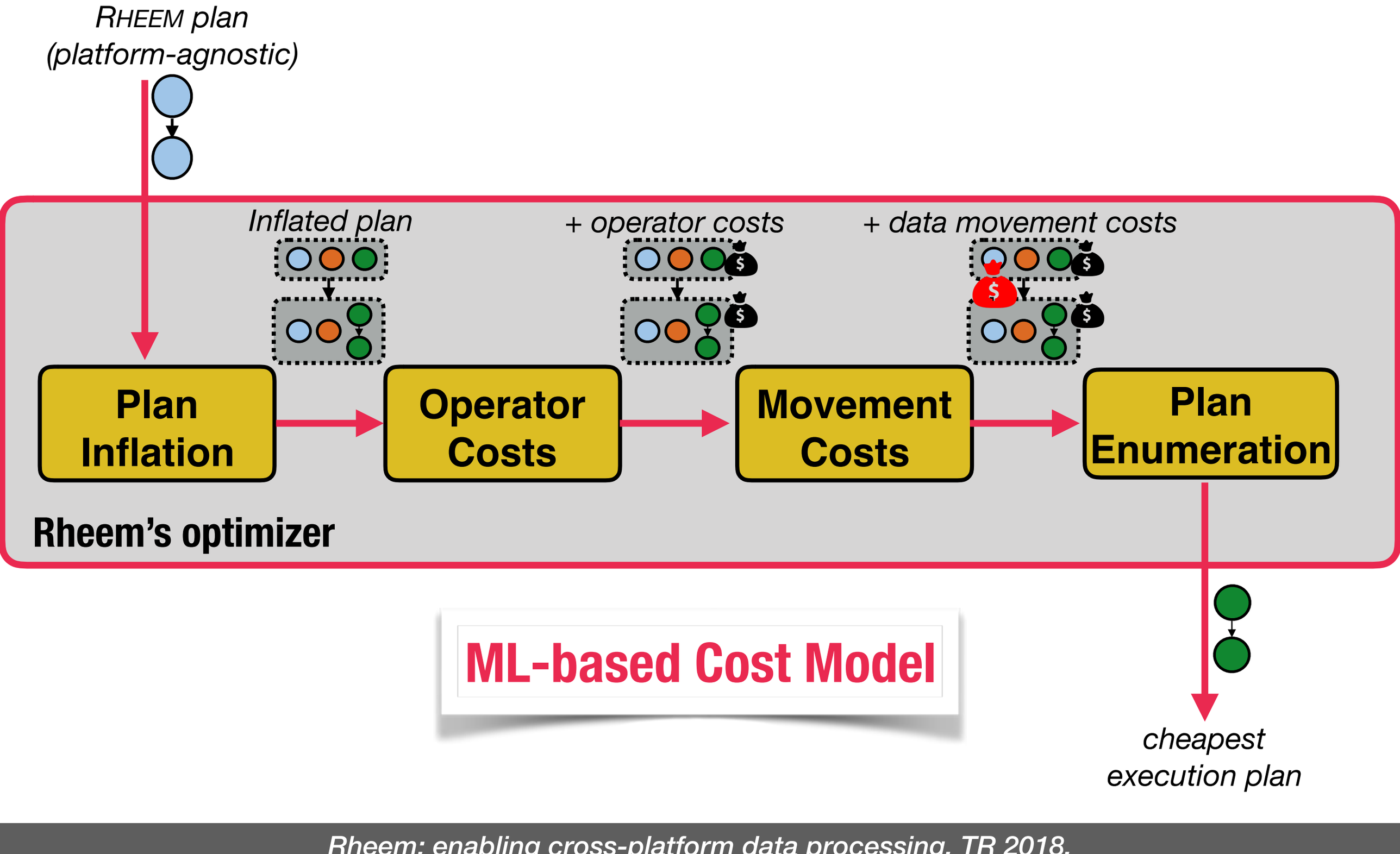
Rheem: enabling cross-platform data processing. TR 2018.

Rheem



Rheem: enabling cross-platform data processing. TR 2018.

Rheem



Rheem: enabling cross-platform data processing. TR 2018.

CP Optimization Features

	Method	Enumeration	Cost Model	Data Movement Cost	User-Specified Platform
Musketeer	Cost-based	Exhaustive/DP	Analytical	No	No
Rheem	Cost-based	Enumeration algebra	ML-based	Yes	Operator level

CP Optimization Features

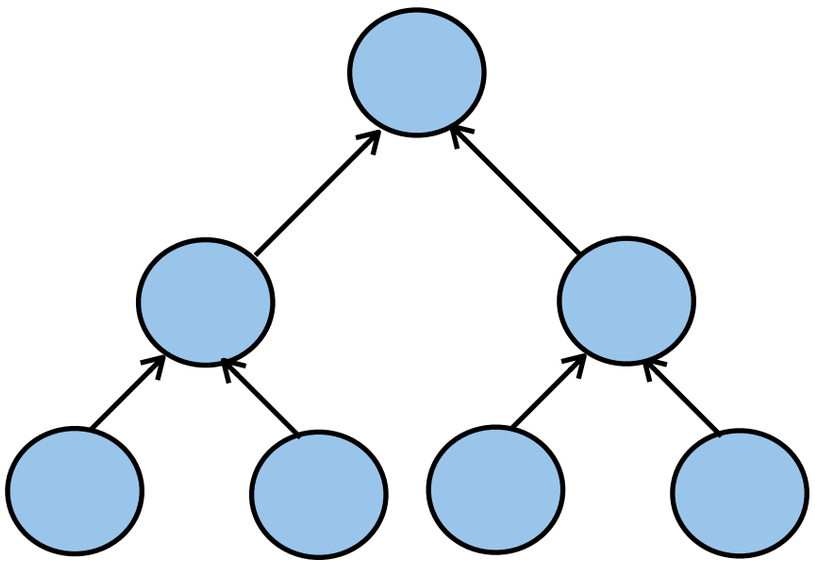
	Method	Enumeration	Cost Model	Data Movement Cost	User-Specified Platform
Musketeer	Cost-based	Exhaustive/DP	Analytical	No	No
Rheem	Cost-based	Enumeration algebra	ML-based	Yes	Operator level
Ires	Cost-based	DP	ML-based	Yes	Operator level
QoX	Cost-based	Greedy	Analytical	Yes	No
Cyclops	Cost-based	?	ML-based	No	Task level

Cross-Platform Optimization

Cost-based

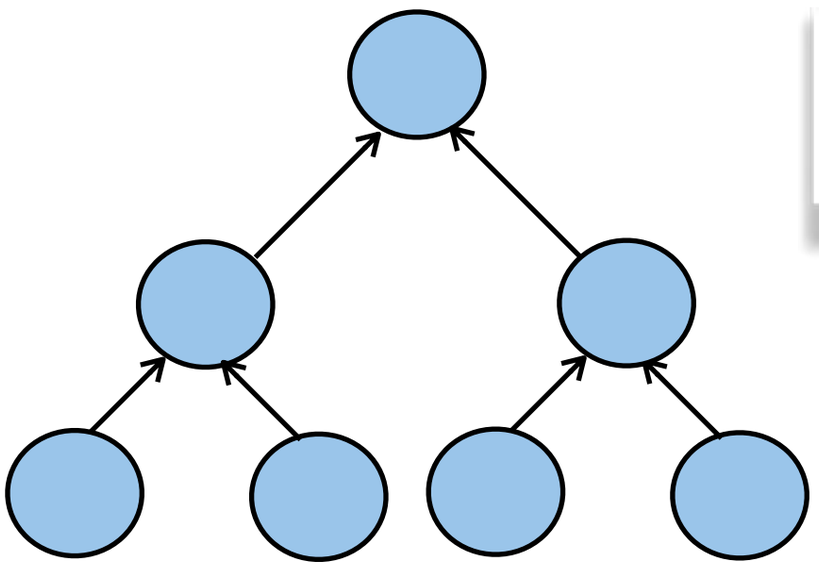
Rule-based

Myria



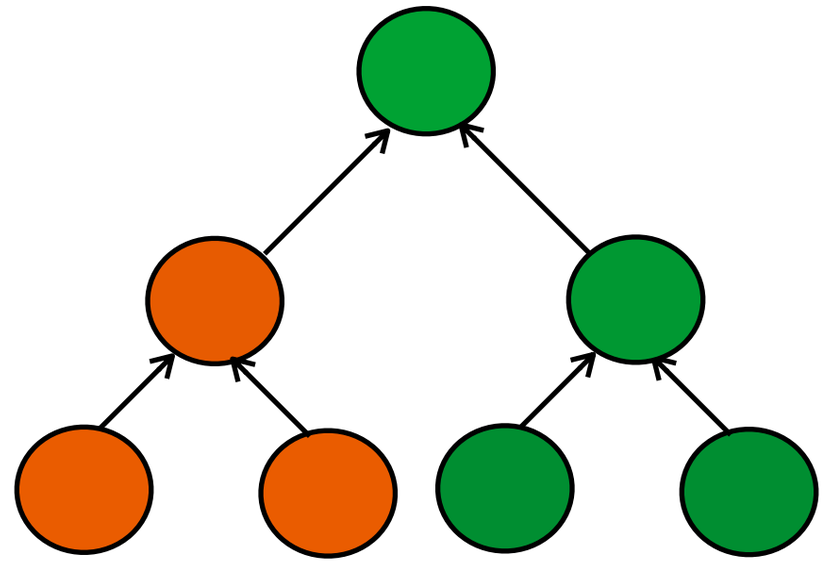
Logical plan

Myria



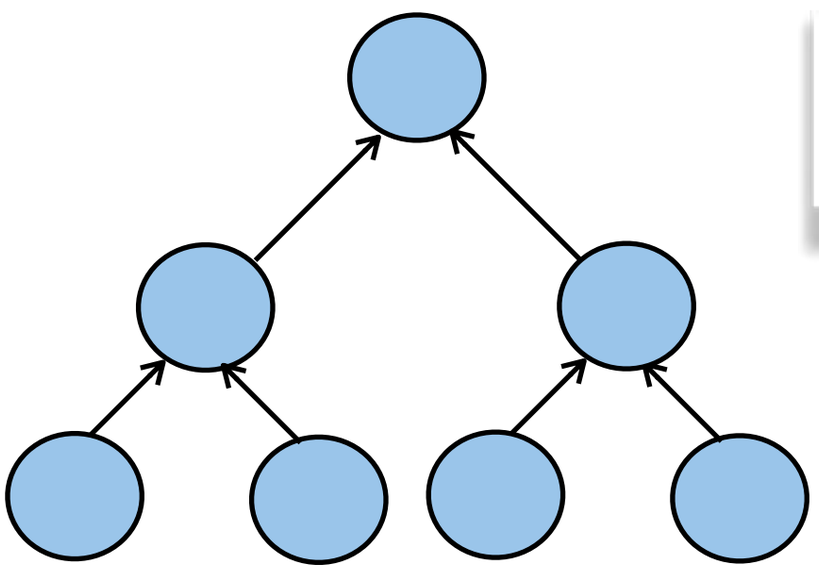
Logical plan

Rewrite rules



Physical plan

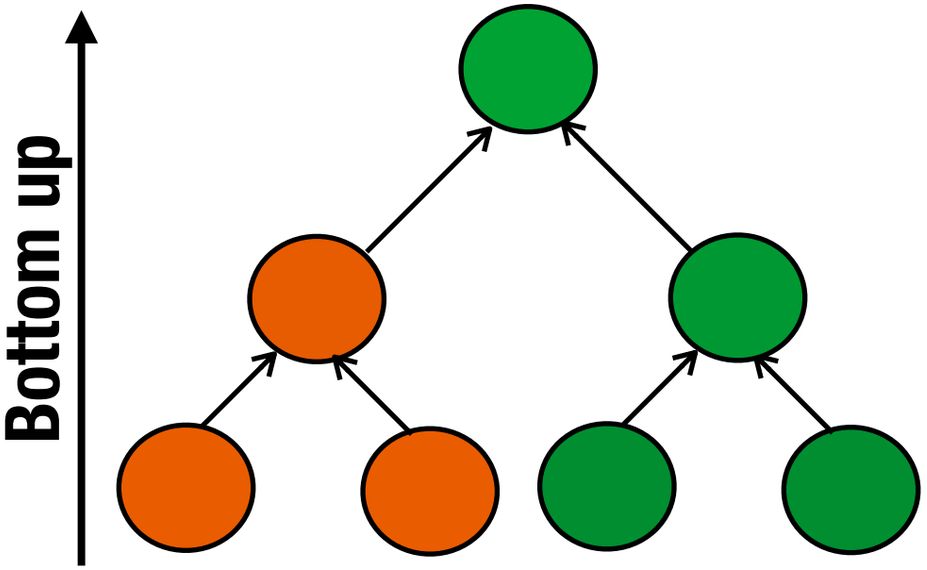
Myria



Logical plan

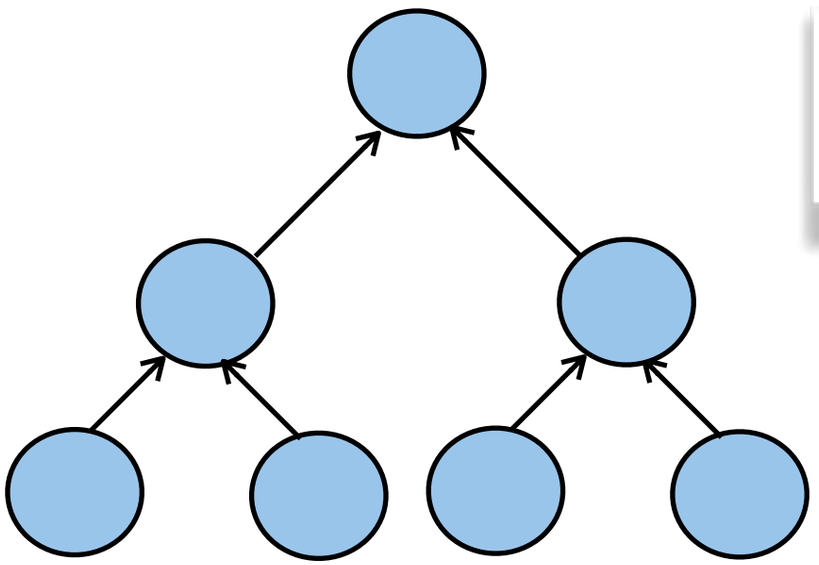
Rewrite rules

Bottom up



Physical plan

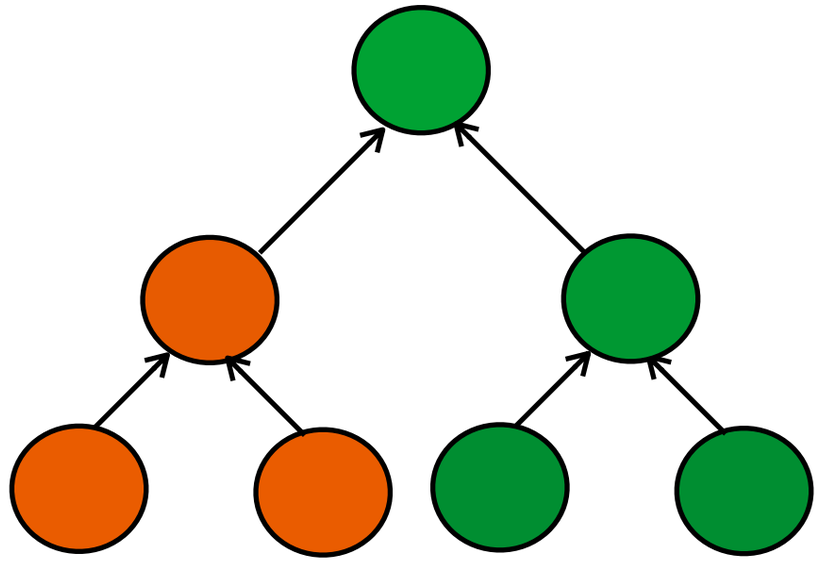
Myria



Logical plan

Rewrite rules

Bottom up



Physical plan

Leaf nodes executed where the data resides

CP Optimization Features

	Method	Enumeration	Cost Model	Data Movement Cost	User-Specified Platform
Musketeer	Cost-based	Exhaustive/DP	Analytical	No	No
Rheem	Cost-based	Enumeration algebra	ML-based	Yes	Operator level
Ires	Cost-based	DP	ML-based	Yes	Operator level
QoX	Cost-based	Greedy	Analytical	Yes	No
Cyclops	Cost-based	?	ML-based	No	Task level
Myria	Rule-based	Bottom up	NA	NA	No

Challenges

1 Decoupling Applications

4 Extensibility

3

Automatic Cross-Platform Data Processing

2 Data Movement

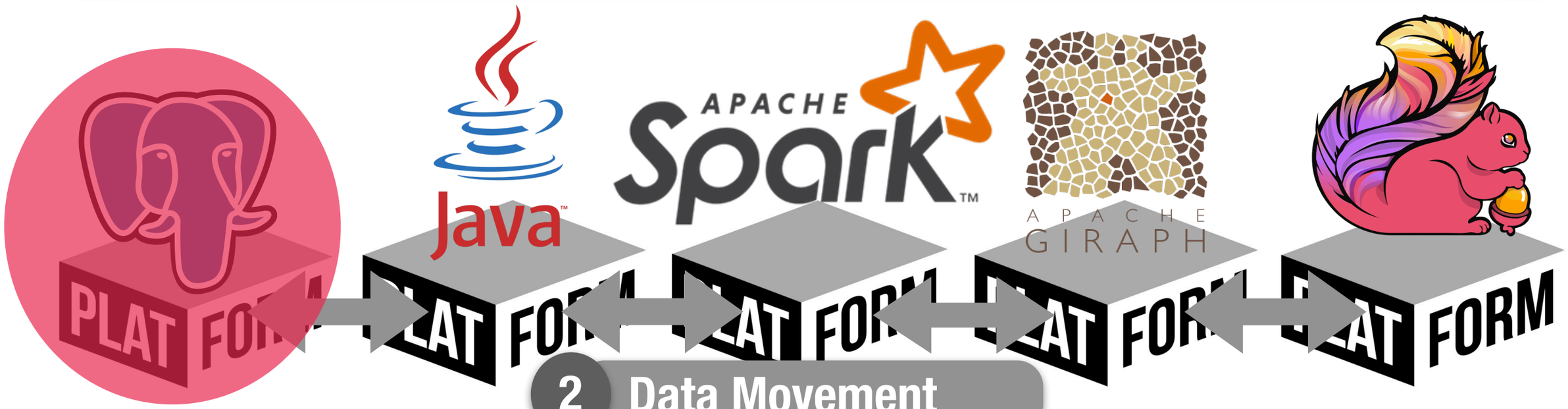
Adding/Modifying a Platform

1 Decoupling Applications

4 Extensibility

3

Automatic Cross-Platform Data Processing



2 Data Movement

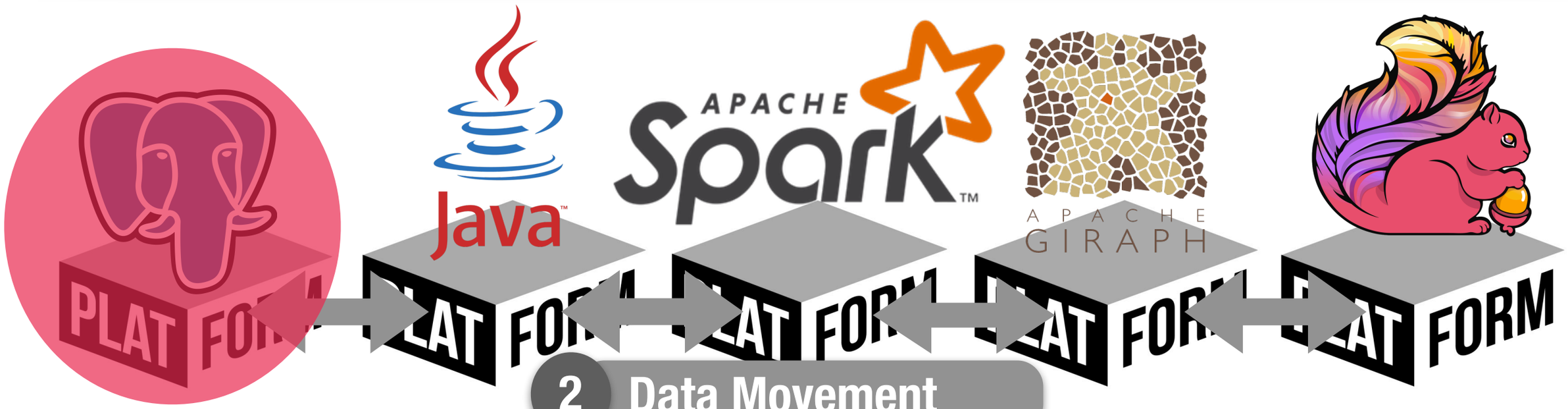
Adding/Modifying a Platform

1 Decoupling Applications

4 Extensibility

3

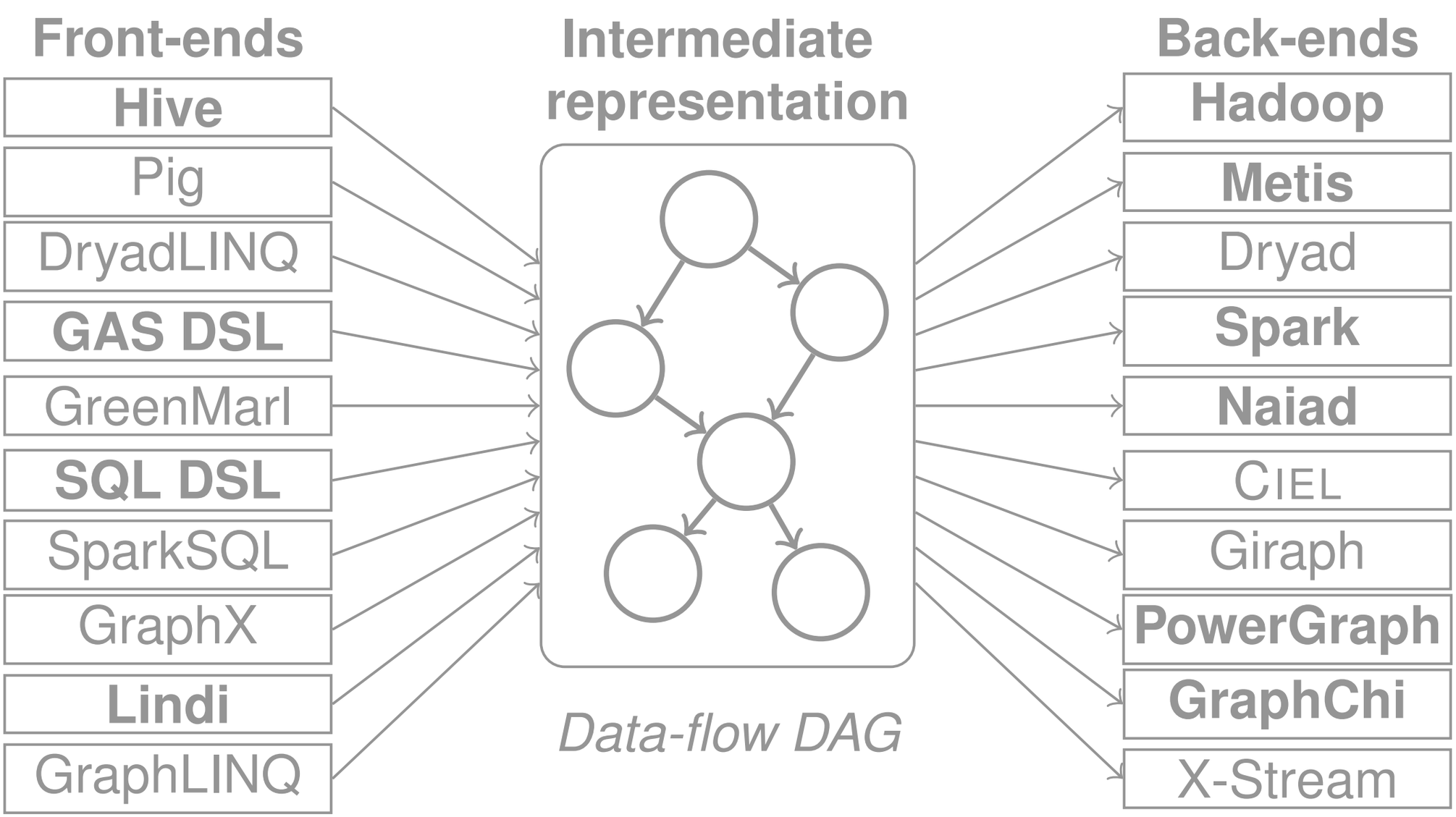
Automatic Cross-Platform Data Processing



2 Data Movement

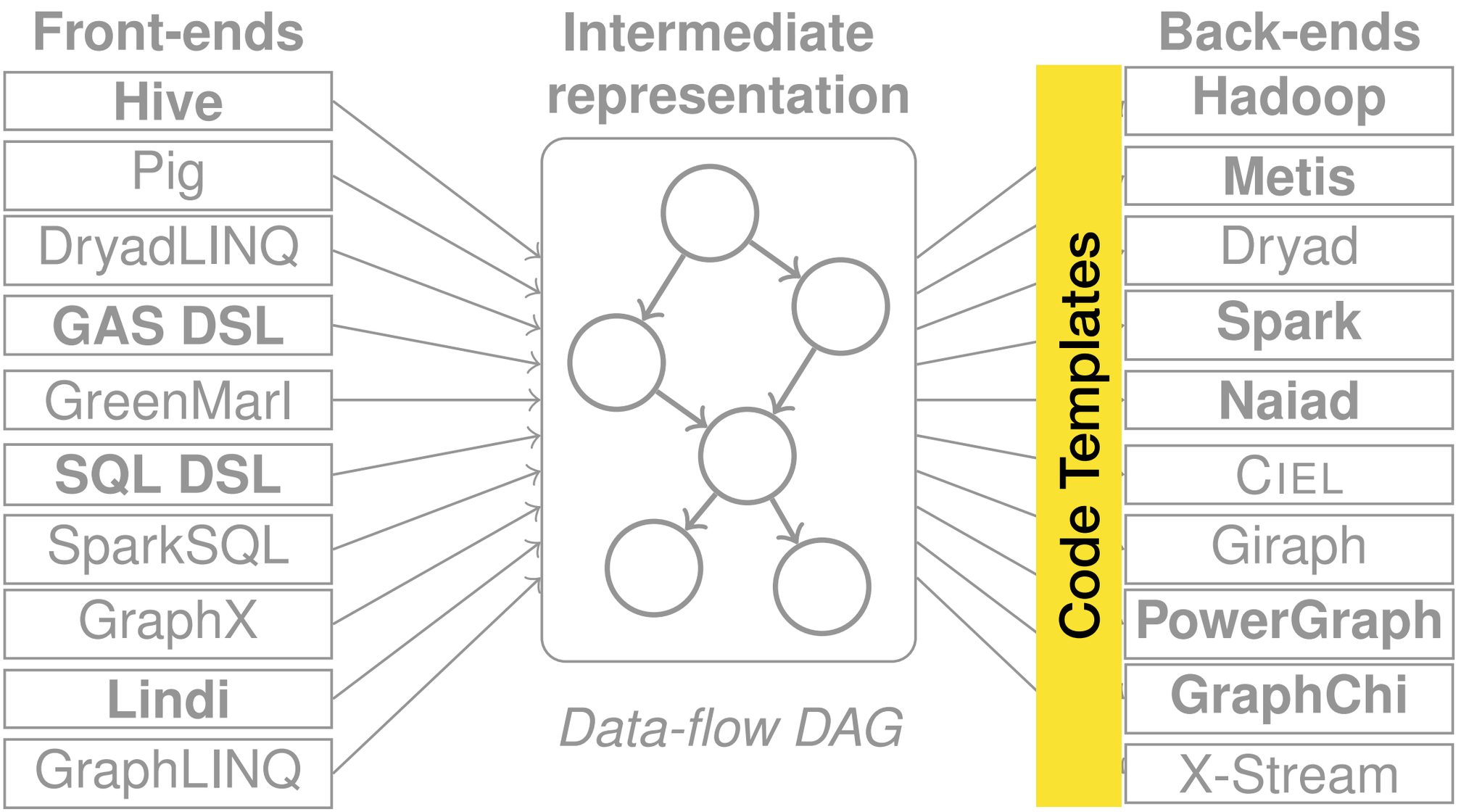
Musketeer

Code templates describing a platform



Musketeer

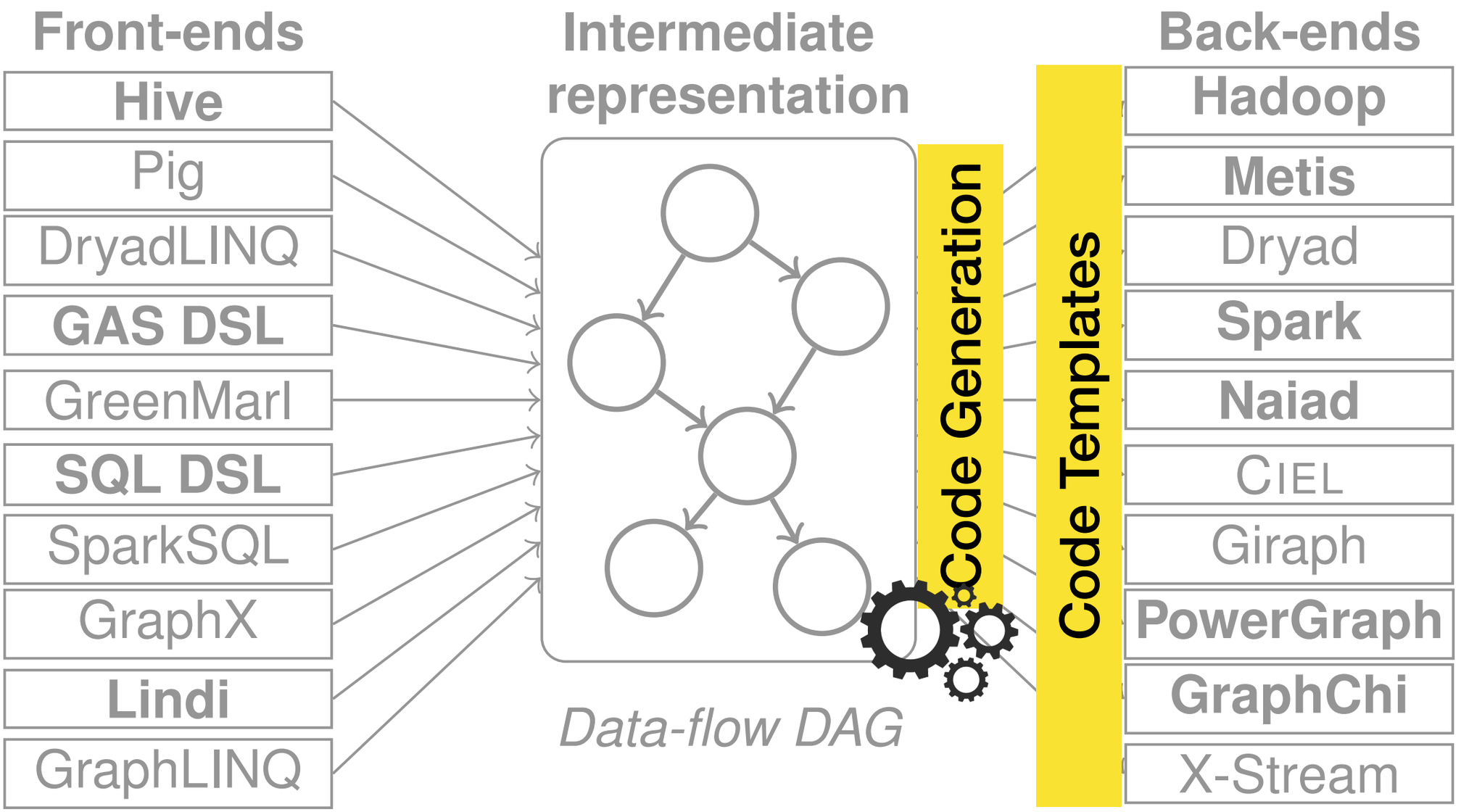
Code templates describing a platform



Musketeer: all for one, one for all in data processing systems. EuroSys'15.

Musketeer

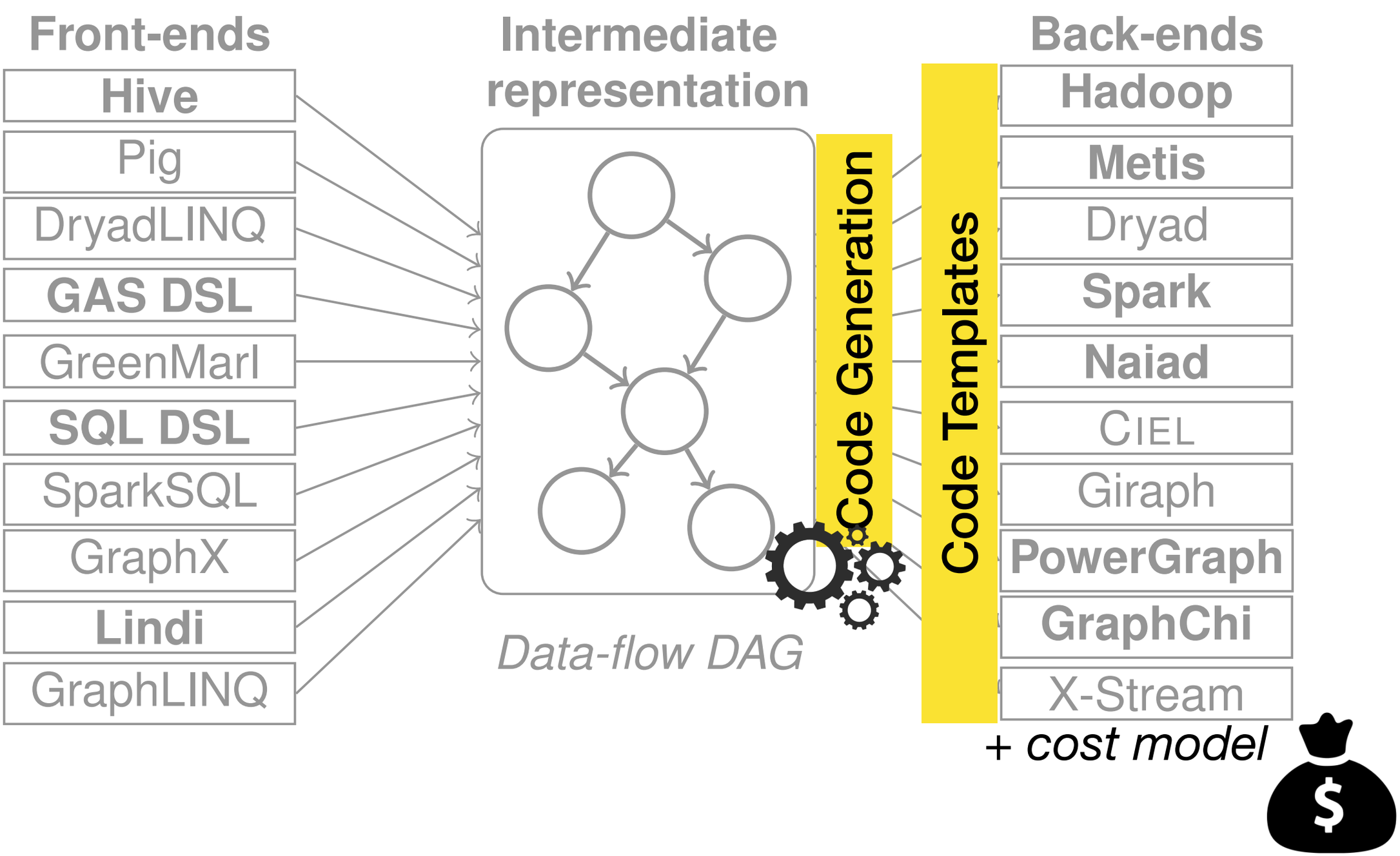
Code templates describing a platform



Musketeer: all for one, one for all in data processing systems. EuroSys'15.

Musketeer

Code templates describing a platform

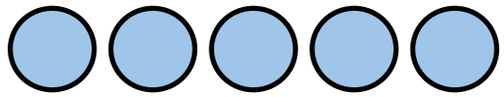


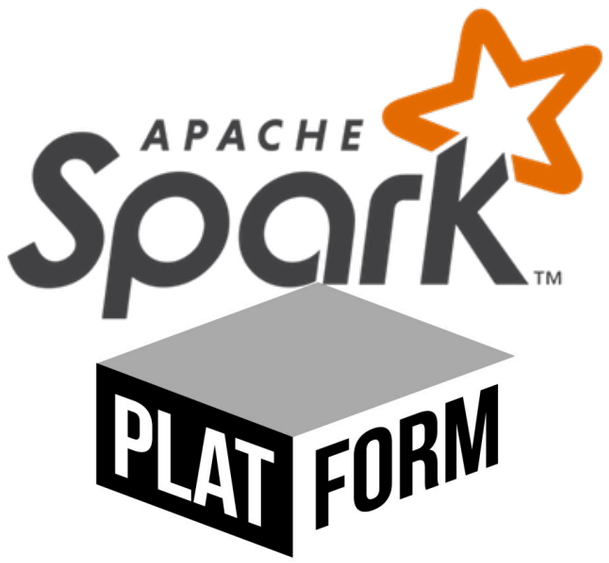
Musketeer: all for one, one for all in data processing systems. EuroSys'15.

Rheem

Execution operators describing a platform

Rheem Operators



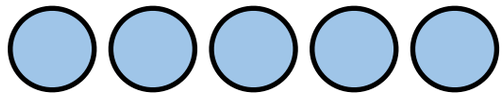


Rheem: enabling cross-platform data processing. TR 2018.

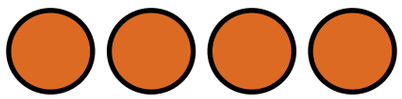
Rheem

Execution operators describing a platform

Rheem Operators



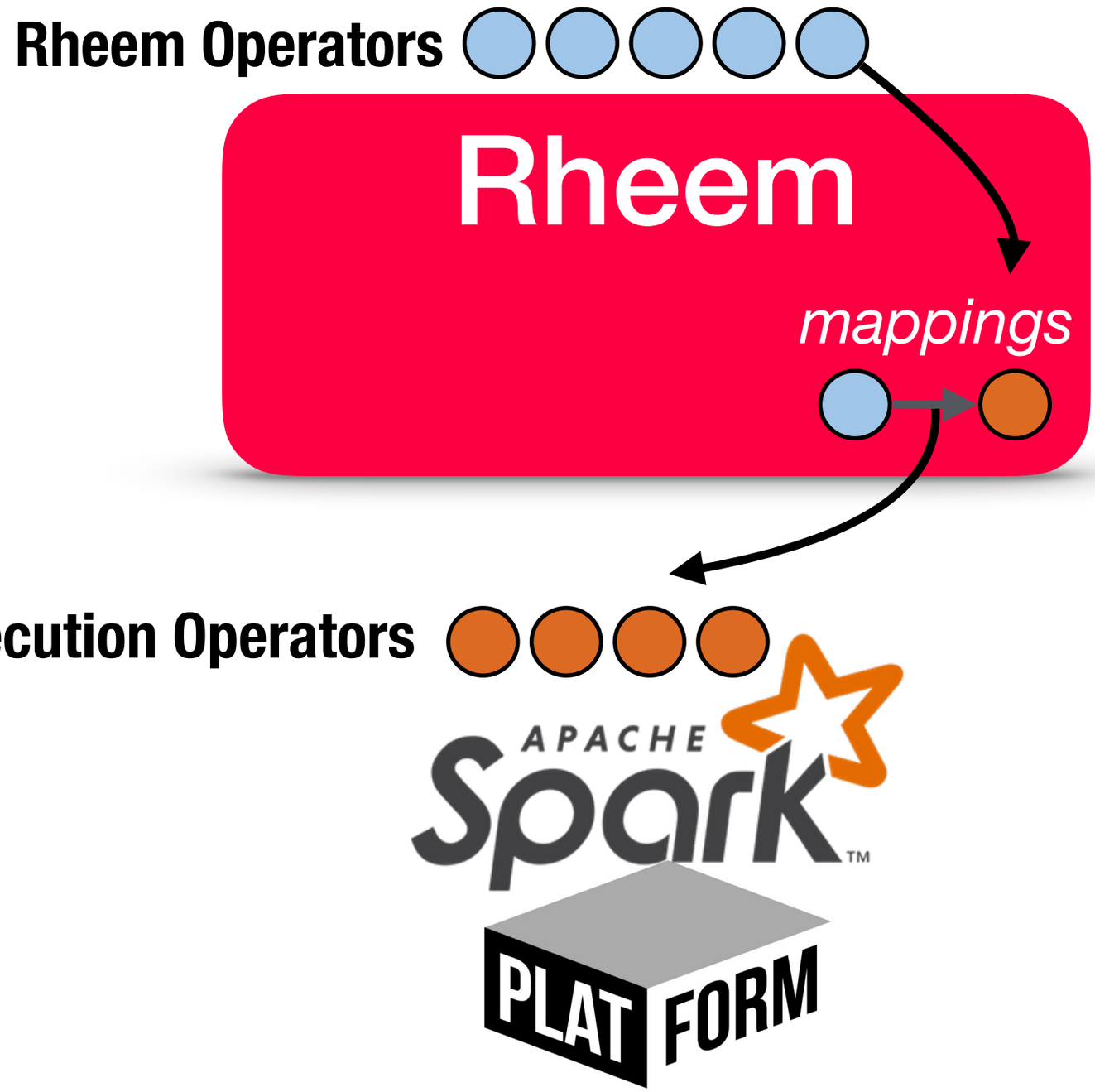
Execution Operators



Rheem: enabling cross-platform data processing. TR 2018.

Rheem

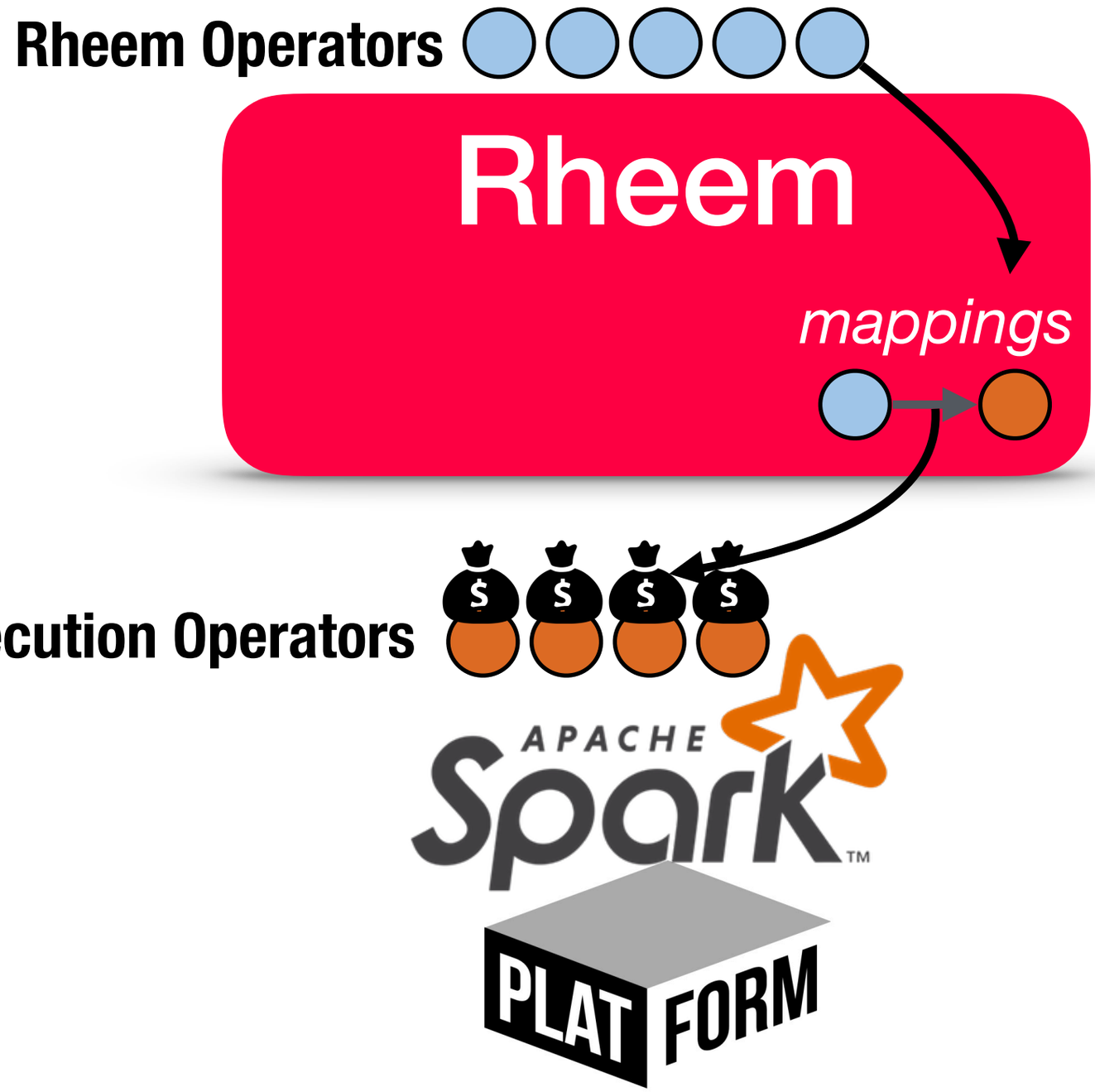
Execution operators describing a platform



Rheem: enabling cross-platform data processing. TR 2018.

Rheem

Execution operators describing a platform



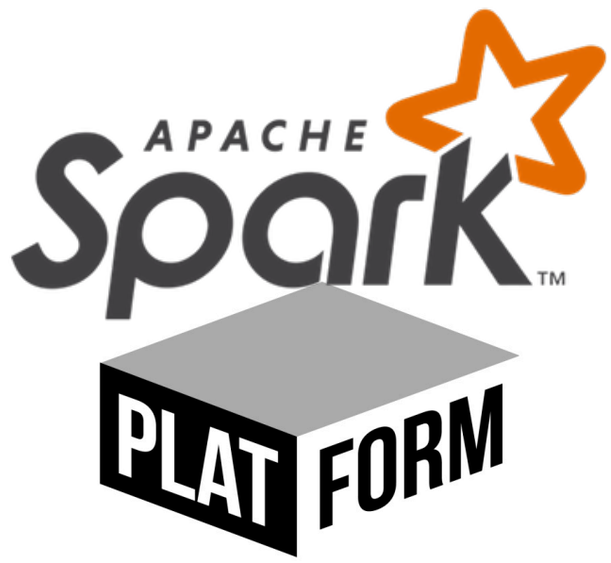
Rheem: enabling cross-platform data processing. TR 2018.

Myria

AST describing a platform

MyriaL

Myria



The Myria Big Data Management and Analytics System and Cloud Service. CIDR 2017

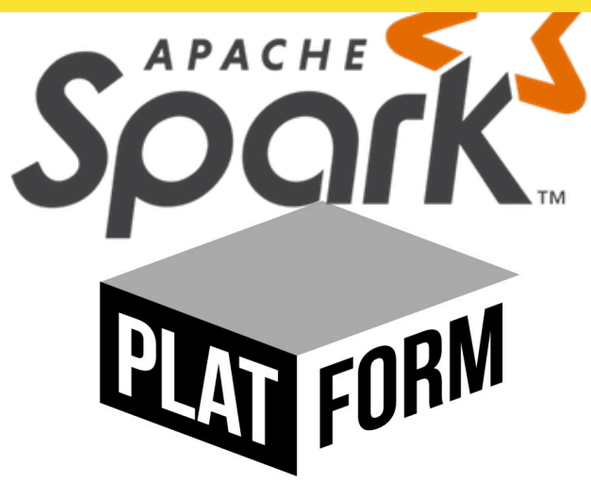
Myria

AST describing a platform

MyriaL

Myria

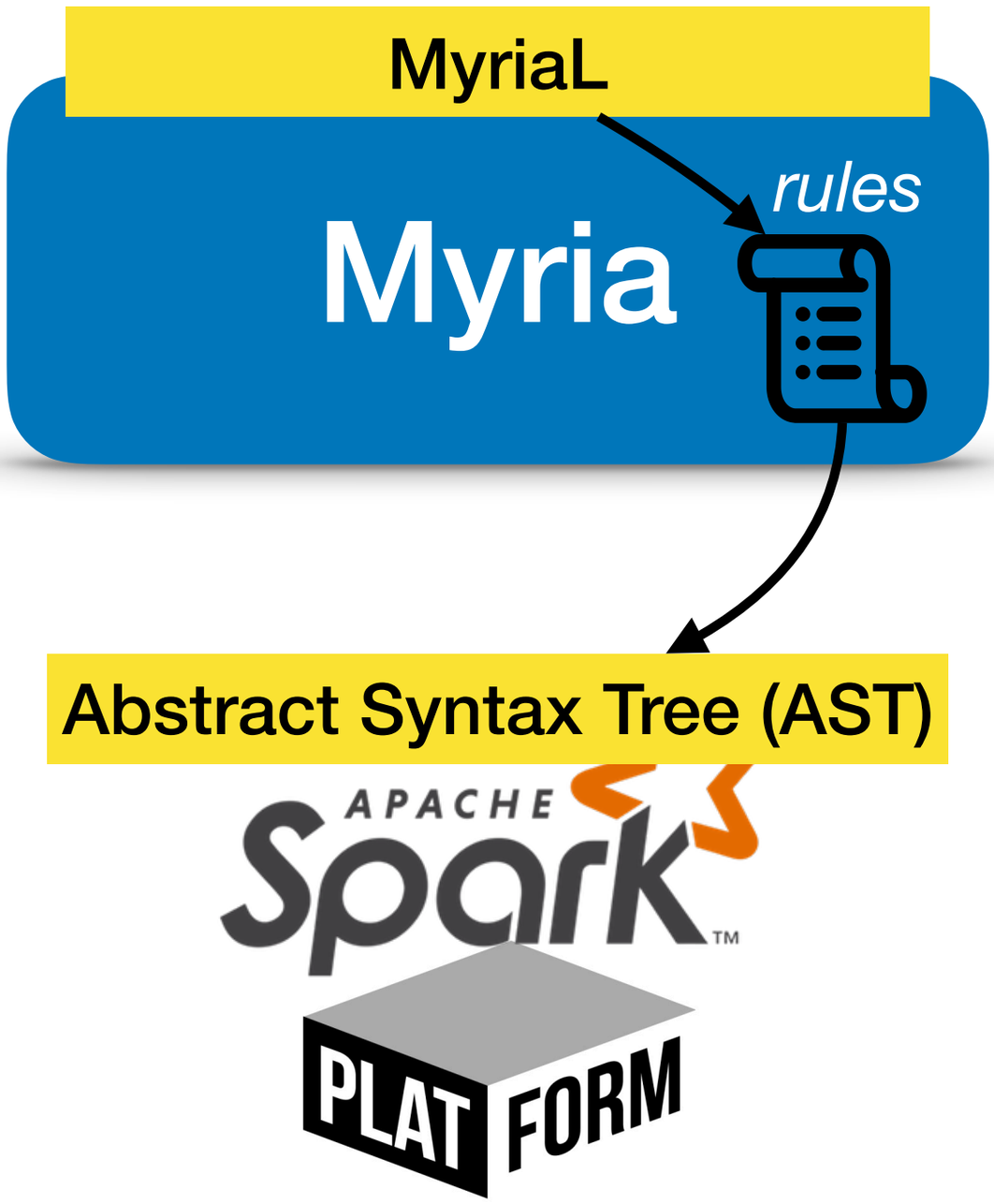
Abstract Syntax Tree (AST)



The Myria Big Data Management and Analytics System and Cloud Service. CIDR 2017

Myria

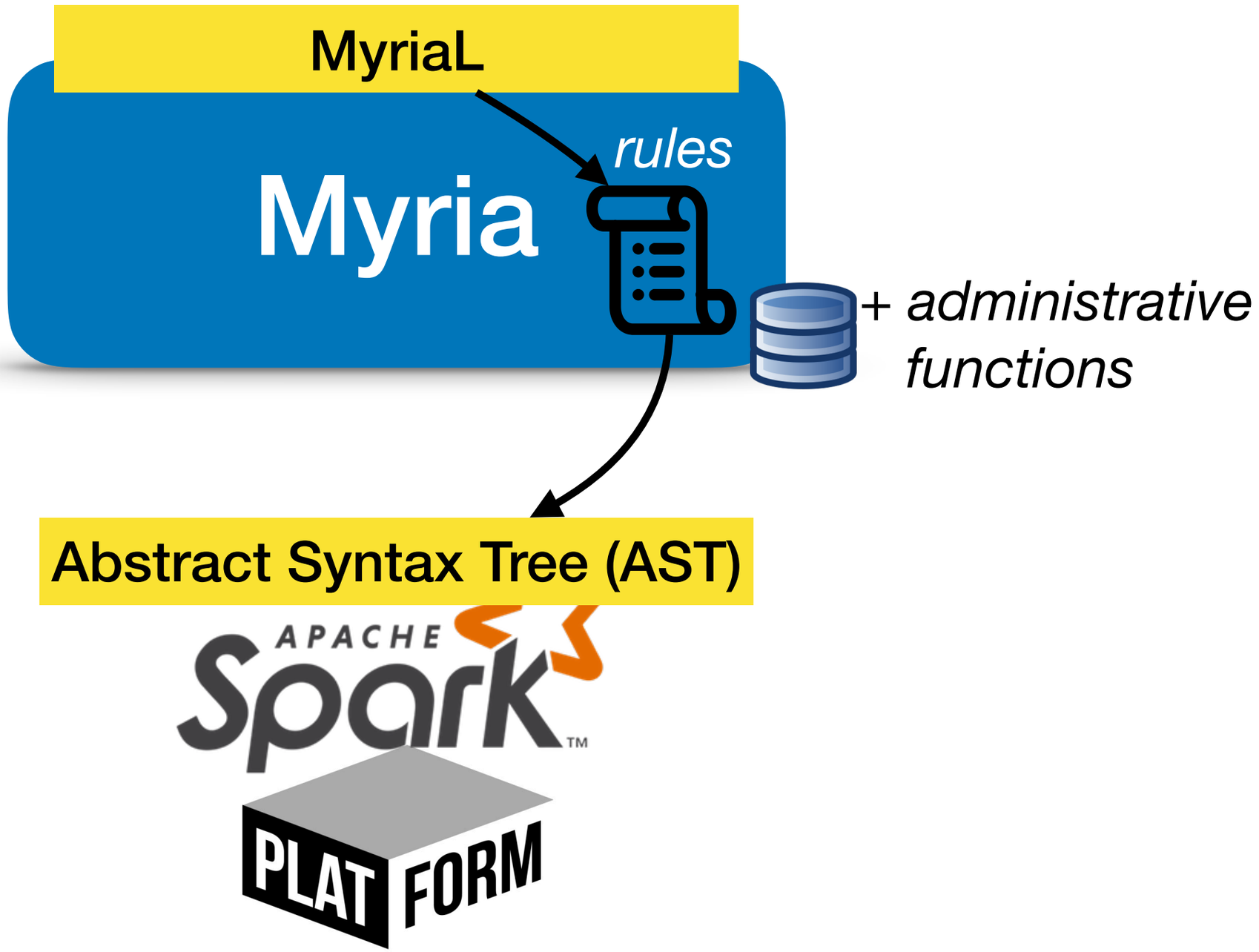
AST describing a platform



The Myria Big Data Management and Analytics System and Cloud Service. CIDR 2017

Myria

AST describing a platform



Extending the Processing Model

1 Decoupling Applications

4 Extensibility

3

Automatic Cross-Platform Data Processing

2 Data Movement

Extending the Processing Model

1 Decoupling Applications

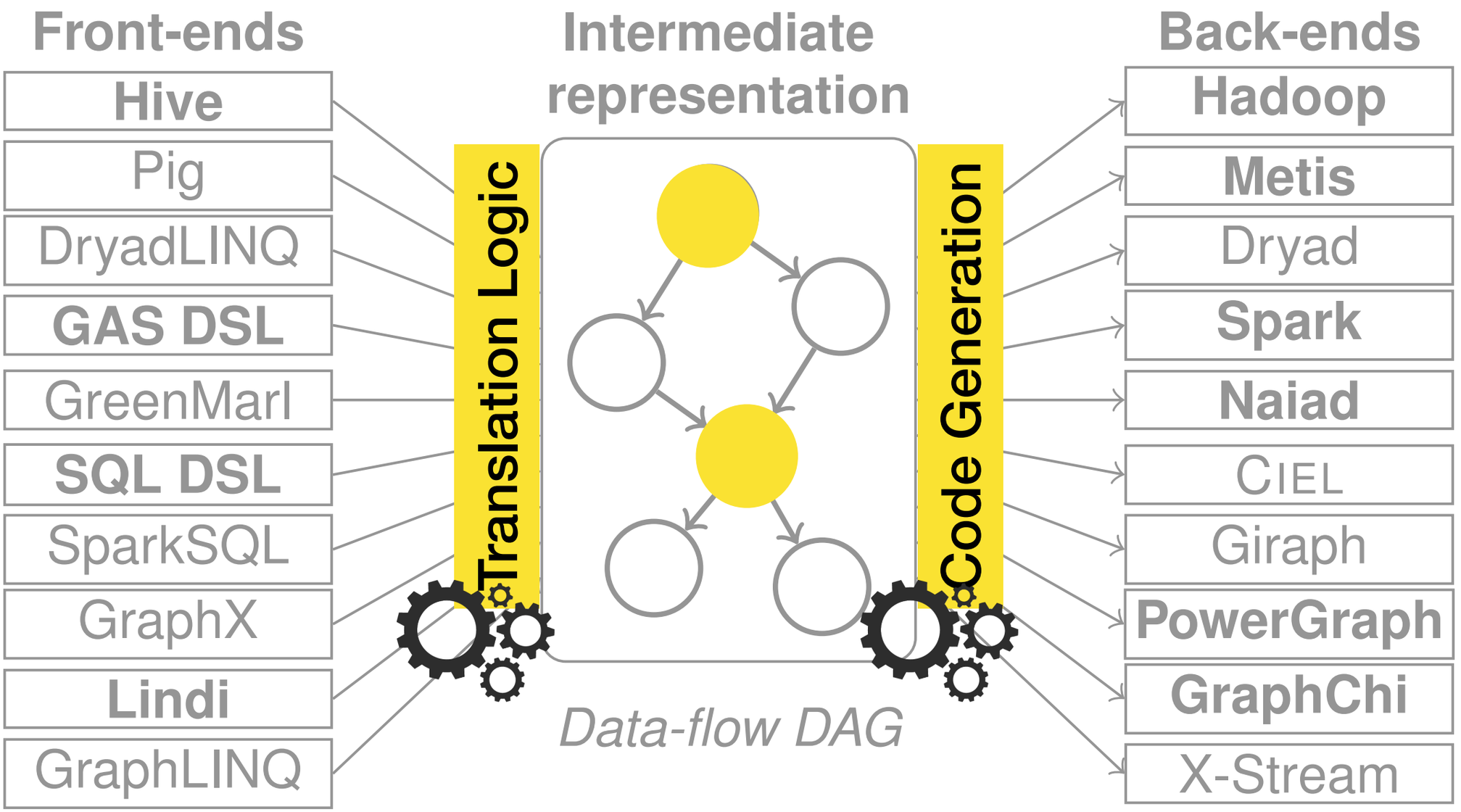
4 Extensibility

3

Automatic Cross-Platform Data Processing

2 Data Movement

Musketeer

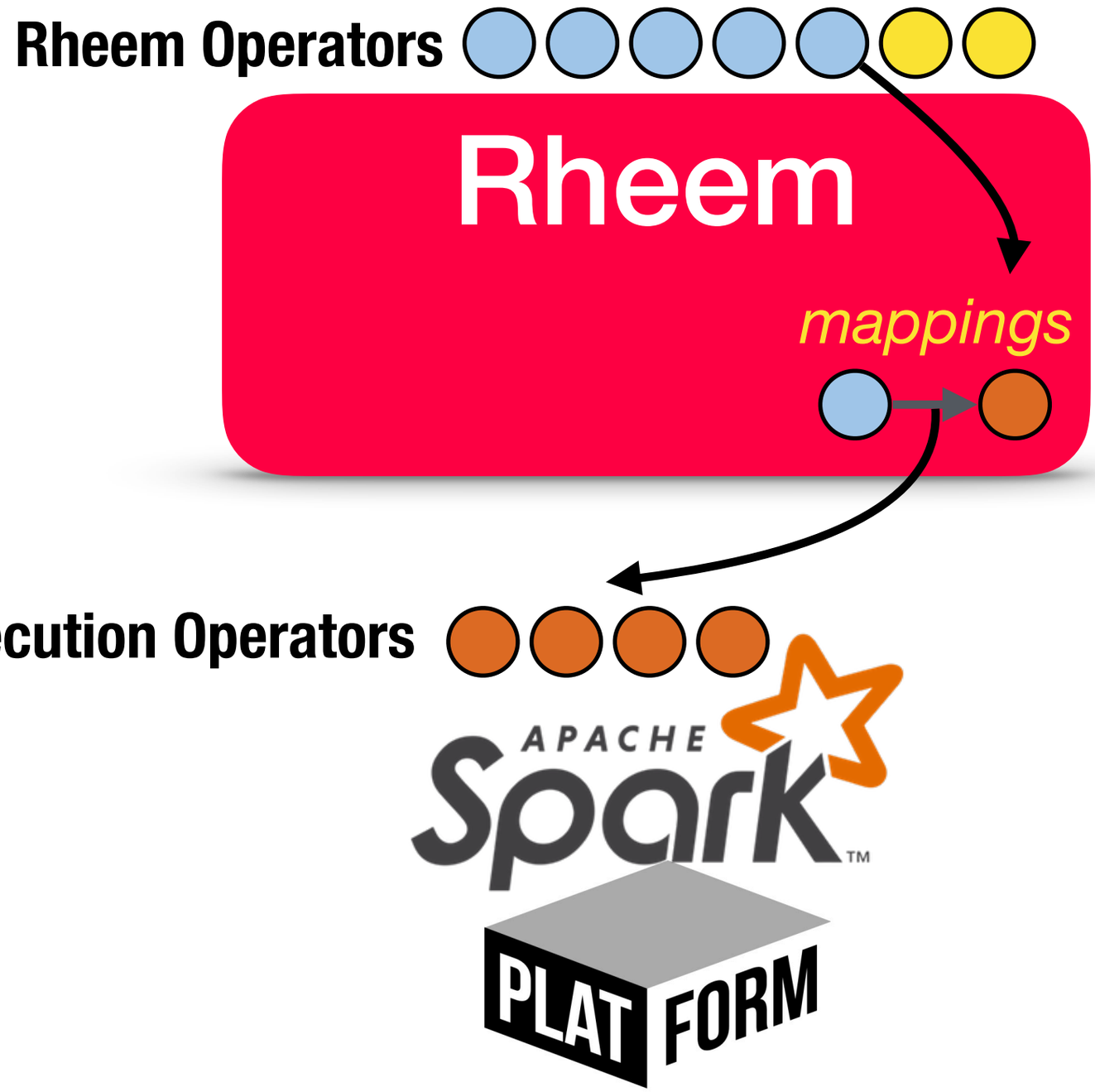


Musketeer: all for one, one for all in data processing systems. EuroSys'15.

Extensibility Features

	Platform Description	IR Extensibility	Difficulty
Musketeer	Code Templates	Internal Code Changes	High
Rheem	Execution Operators		
Myria	AST		

Rheem



Rheem: enabling cross-platform data processing. TR 2018.

Extensibility Features

	Platform Description	IR Extensibility	Difficulty
Musketeer	Code Templates	Internal Code Changes	High
Rheem	Execution Operators	Rheem Operators	Medium
Myria	AST		

Extensibility Features

	Platform Description	IR Extensibility	Difficulty
Musketeer	Code Templates	Internal Code Changes	High
Rheem	Execution Operators	Rheem Operators	Medium
Myria	AST	–	Medium
BigDawg	Language Translators	–	High
Cyclops	?	–	High
QoX	Internal Code Changes	–	High

Cross-Platform Data Processing

Motivation

Use Cases

Challenges

Current Efforts

Summary

Recap

	Processing Model	CP Optimizer	Data Movement Cost	Extensibility
Musketeer	Dataflow DAG + Loop	Yes	**No**	High
Rheem	Dataflow DAG + Loop	Yes	Yes	Medium
Ires	Workflow graph	Yes	Yes	Medium
BigDawg	Query Plan Tree	**No**	NA	High
Myria	Extended relational	Yes	**No**	High
QoX	Dataflow graph	Yes	Yes	High
Cyclops	Windowed aggregation	Yes	Yes	High
Apache Beam	Dataflow DAG	**No**	NA	High
CloudMdsQL	Query Plan Tree	**No**	NA	High

Recap

	Processing Model	CP Optimizer	Data Movement Cost	Extensibility
Musketeer	Dataflow DAG + Loop	Yes	**No**	High
Rheem	Dataflow DAG + Loop	Yes	Yes	Medium
			Yes	Medium
BigDawg	Query Plan Tree	**No**	NA	High
Myria	Extended relational	Yes	**No**	High
QoX	Dataflow graph	Yes	Yes	High
Cyclops	Windowed aggregation	Yes	Yes	High
Apache Beam	Dataflow DAG	**No**	NA	High
CloudMdsQL	Query Plan Tree	**No**	NA	High

Fixed Processing Model

Recap

	Processing Model	CP Optimizer	Data Movement Cost	Extensibility
Musketeer	Dataflow DAG + Loop	Yes	**No**	High
Rheem	Dataflow DAG + Loop	Yes	Yes	Medium
			Yes	Medium
BigDawg	Query Plan Tree	**No**	NA	High
Myria	Extended			High
QoX	Dataflo			High
Cyclops	Windowed aggregation	Yes	Yes	High
Apache Beam	Dataflow DAG	**No**	NA	High
CloudMdsQL	Query Plan Tree	**No**	NA	High

Fixed Processing Model

Time-Consuming & Hard Cost Tuning

Recap

	Processing Model	CP Optimizer	Data Movement Cost	Extensibility
Musketeer	Dataflow DAG + Loop	Yes	**No**	High
Rheem	Dataflow DAG + Loop	Yes	Yes	Medium
			Yes	Medium
BigDawg	Query Plan Tree	**No**	NA	High
Myria	Extended			High
QoX	Dataflo			High
Cyclops	Windowed aggregation	Yes	Yes	High
Apache Beam	Dataflow DAG	**No**	NA	High
CloudMdsQL	Query Plan Tree	**No**	NA	High

Fixed Processing Model

Time-Consuming & Hard Cost Tuning

Open Problems

Flexible Processing Model

How to support arbitrary processing models?

Open Problems

Flexible Processing Model

How to support arbitrary processing models?

Fully Automatic Cost Tuning

- How to match real workloads without knowing them?
- Weak control over the underlying platforms

Open Problems

Flexible Processing Model

How to support arbitrary processing models?

Fully Automatic Cost Tuning

- How to match real workloads without knowing them?
- Weak control over the underlying platforms

Platform Integration

How to integrate a platform fully automatically?

Open Problems

Flexible Processing Model

How to support arbitrary processing models?

Fully Automatic Cost Tuning

- How to match real workloads without knowing them?
- Weak control over the underlying platforms

Platform Integration

How to integrate a platform fully automatically?

Fast Data Transfer

What is the right intermediate data representation?

Open Problems

Flexible Processing Model

How to support arbitrary processing models?

Fully Automatic Cost Tuning

- How to match real workloads without knowing them?
- Weak control over the underlying platforms

Platform Integration

How to integrate a platform fully automatically?

Fast Data Transfer

What is the right intermediate data representation?

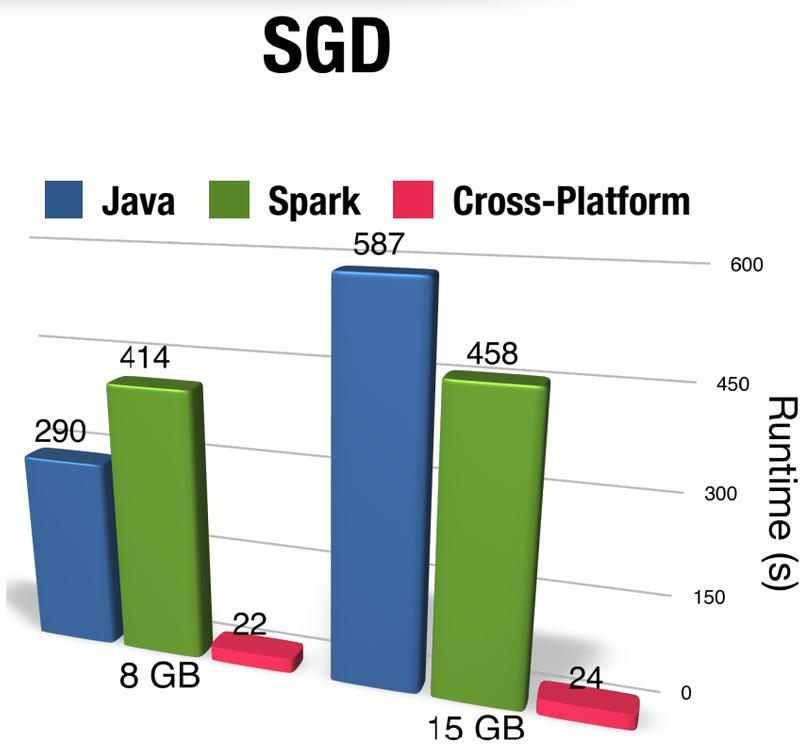
Flexible High-Level Languages

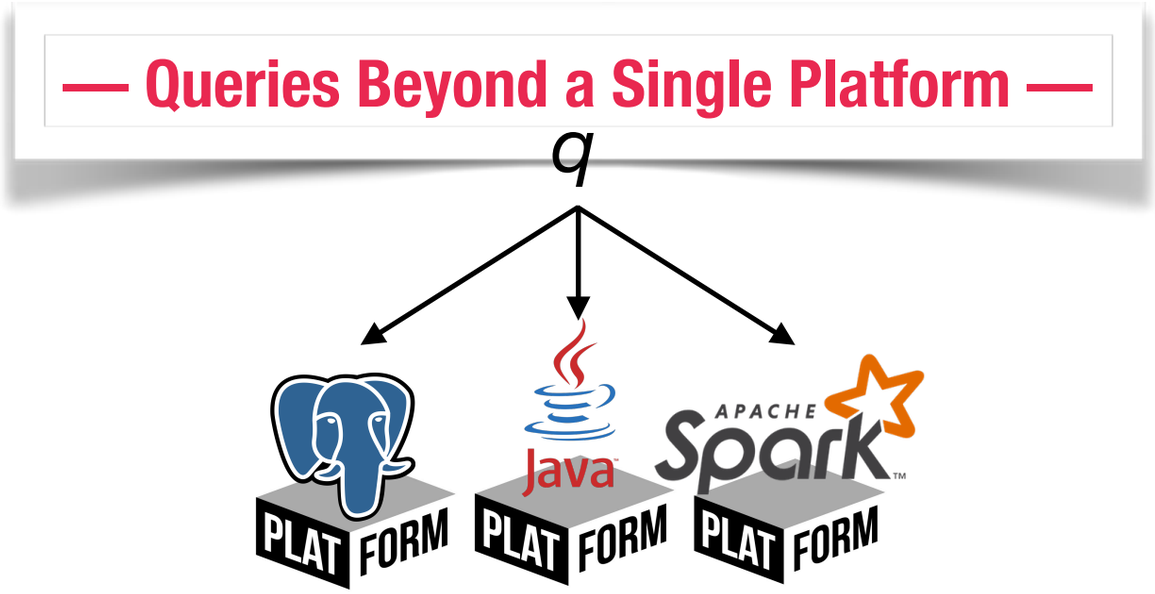
How to adapt to the needs of applications?

Take Away

real need for ***data processing independence***

	Monoplatform	Polyplatform
Monostore	Decoupled Single Platform	Opportunistic CP Mandatory CP
Polystore	Decoupled Single Platform	Polystore CP





	Processing Model	CP Optimizer	Data Movement Cost	Extensibility
Musketeer	Dataflow DAG + Loop	Yes	No	High
Myria	Extended	Yes	Yes	High
QoX	Dataflow graph	Yes	Yes	High
Cyclops	Windowed aggregation	Yes	Yes	High
Apache Beam	Dataflow DAG	No	NA	High

No Perfect Current Solution

Time-consuming & Hard Cost Tuning

OPEN PROBLEMS

1. Fully Automatic Cost Tuning
2. Flexible Processing Model
3. Platform Integration
4. Fast Data Movement
5. Flexible High-Level Languages

References

- I. Gog et al. *Musketeer: all for one, one for all in data processing systems*. In EuroSys, 2015.
- D. Agrawal et al. *Rheem: Enabling Multi-Platform Task Execution*. In SIGMOD, 2016.
- D. Agrawal et al. *Road to Freedom in Big Data Analytics*. In EDBT, 2016.
- K. Doka et al. *Mix ’n’ match multi-engine analytics*. In IEEE BigData, 2016.
- J. Wang et al. *The Myria Big Data Management and Analytics System and Cloud Services*. In CIDR, 2017.
- A. Simitsis et al. *Optimizing Analytic Data Flows for Multiple Execution Engines*. In SIGMOD, 2012.
- V. Gadepally et al. The BigDawg Polystore System and Architecture. IEEE High Performance Extreme Computing 2016.
- A. J. Elmore et al. *A Demonstration of the BigDAWG Polystore System*. PVLDB, 8(12):1908–1911, 2015.
- H. Lim et al. *How to Fit when No One Size Fits*. In CIDR, 2013.
- B. Haynes et al. *PipeGen: Data Pipe Generator for Hybrid Analytics*. In SoCC, 2016.
- S. Palkar et al. *Weld: A Common Runtime for High Performance Data Analysis*. In CIDR, 2017.
- M. Boehm et al. *SystemML: Declarative Machine Learning on Spark*. PVLDB, 9(13):1425–1436, 2016.
- B. Kolev et al. *CloudMdsQL: querying heterogeneous cloud data stores with a common language*. Journal of Distributed and Parallel Databases, 2016.
- C. Chambers et al. *FlumeJava: Easy, Efficient Data-Parallel Pipelines*. SIGPlan 2010.