LinkedIn

Coral: A SQL translation and rewrite engine for modern data lakes

Walaa Eldin Moustafa
Senior Staff Software Engineer

Modern Data Lake Architectures

Variety of query engines

trino

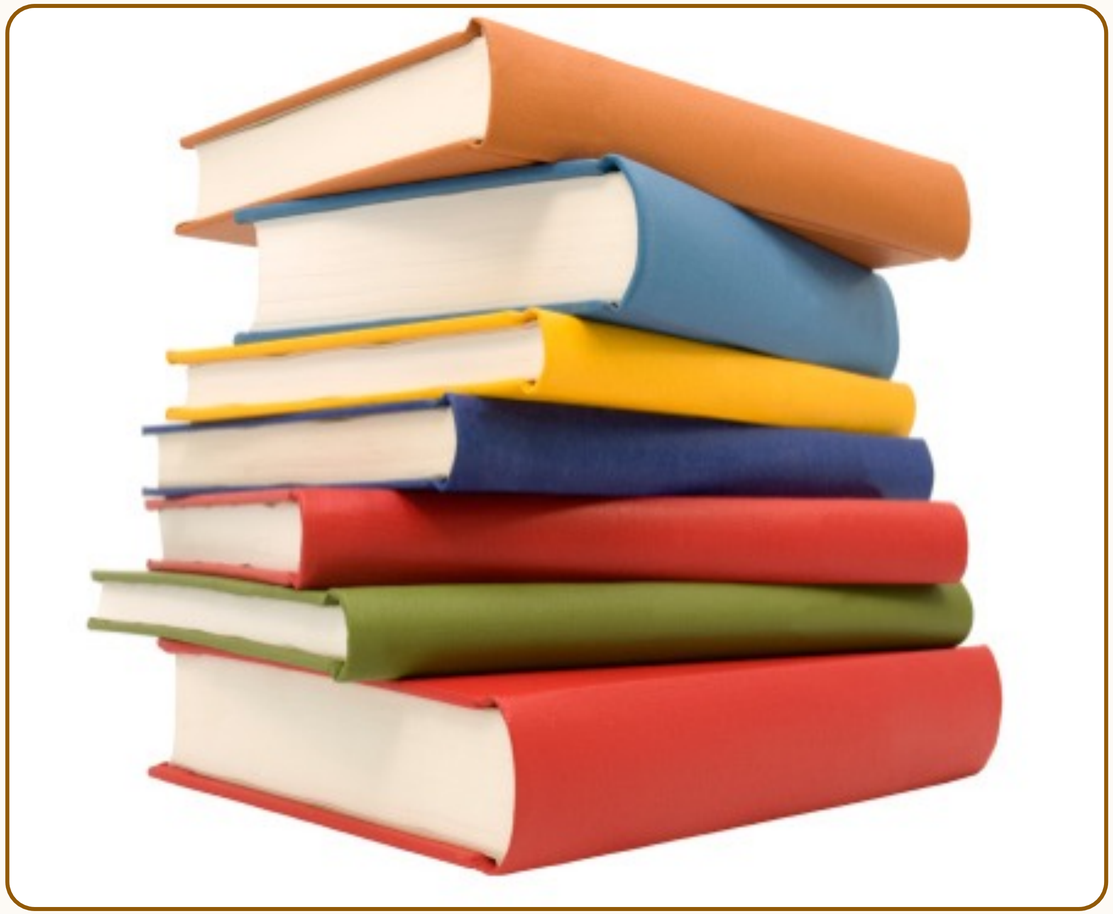
prestoDB

Flink

Modern Data Lake Architectures

Variety of query languages

- Spark SQL
- Hive QL
- Presto SQL
- Trino SQL
- Flink SQL
- Other: Gremlin, SPARQL, Spark Scala, PySpark



Modern Data Lake Architectures

Variety of data sources

Tables

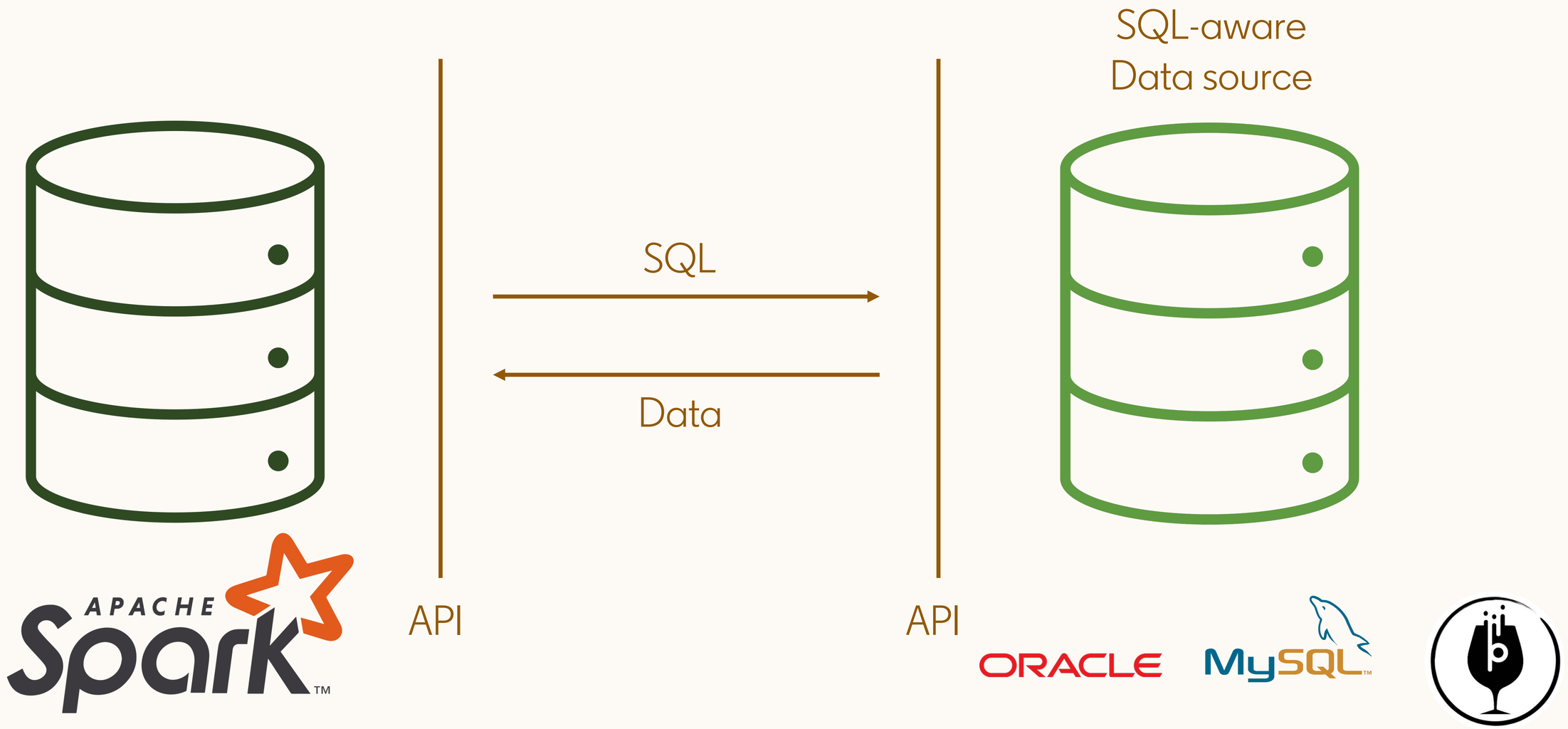
- Hive tables
- Delta Lake tables
- Iceberg tables
- Hudi tables
- Various file formats
  - Avro
  - ORC
  - Parquet

Views

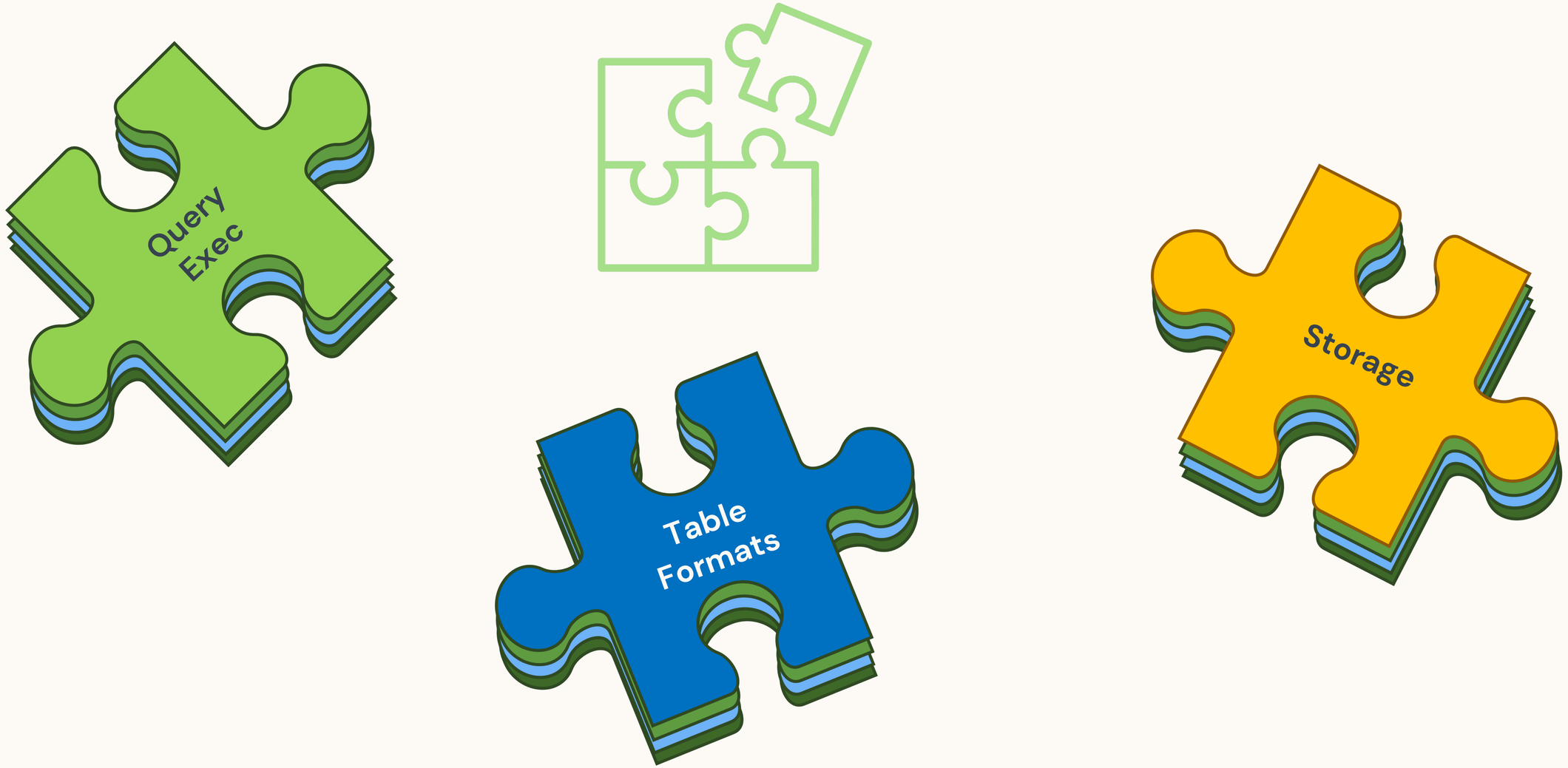
- Different query languages
- Different UDF APIs

Modern Data Lake Architectures

Even more data sources..

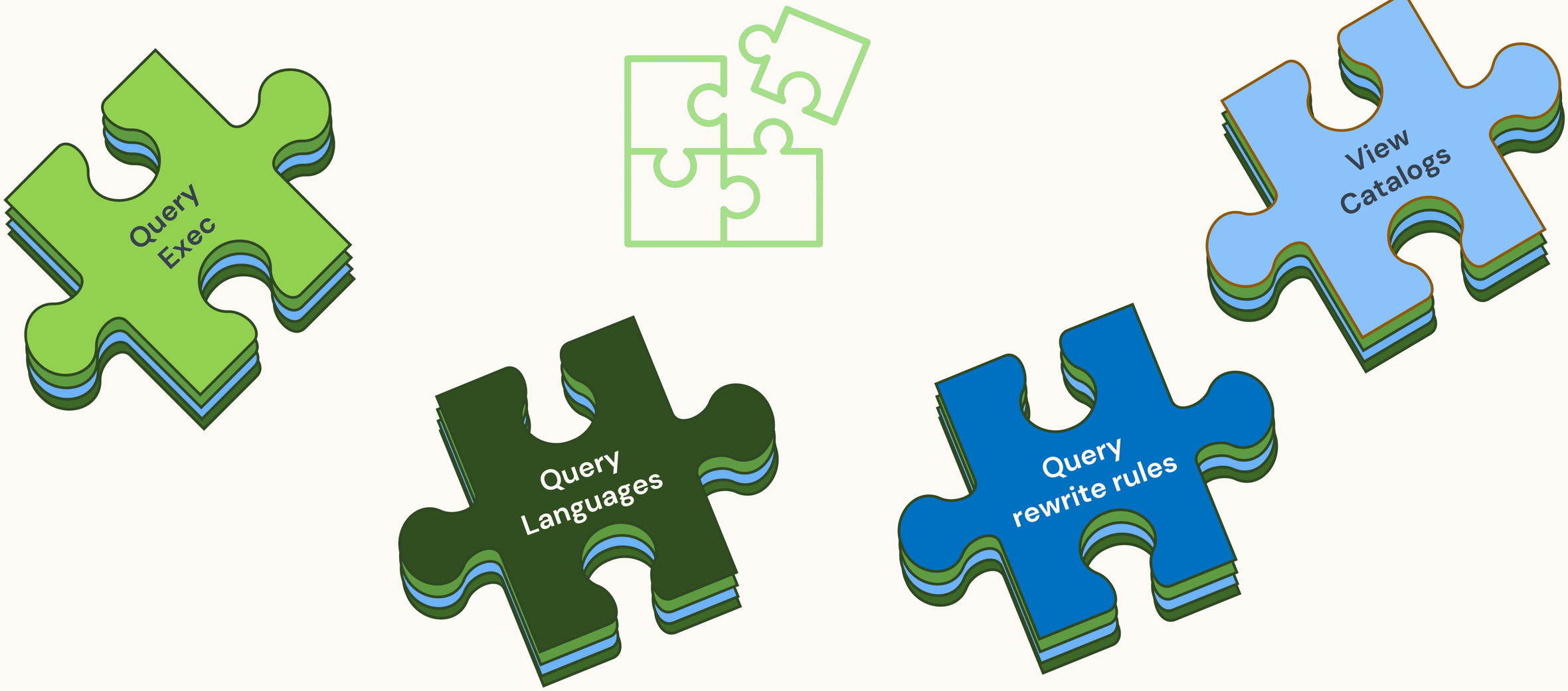


Composable Data Architectures



Composable Data Architectures

But not quite there yet..



Composable Data Architectures

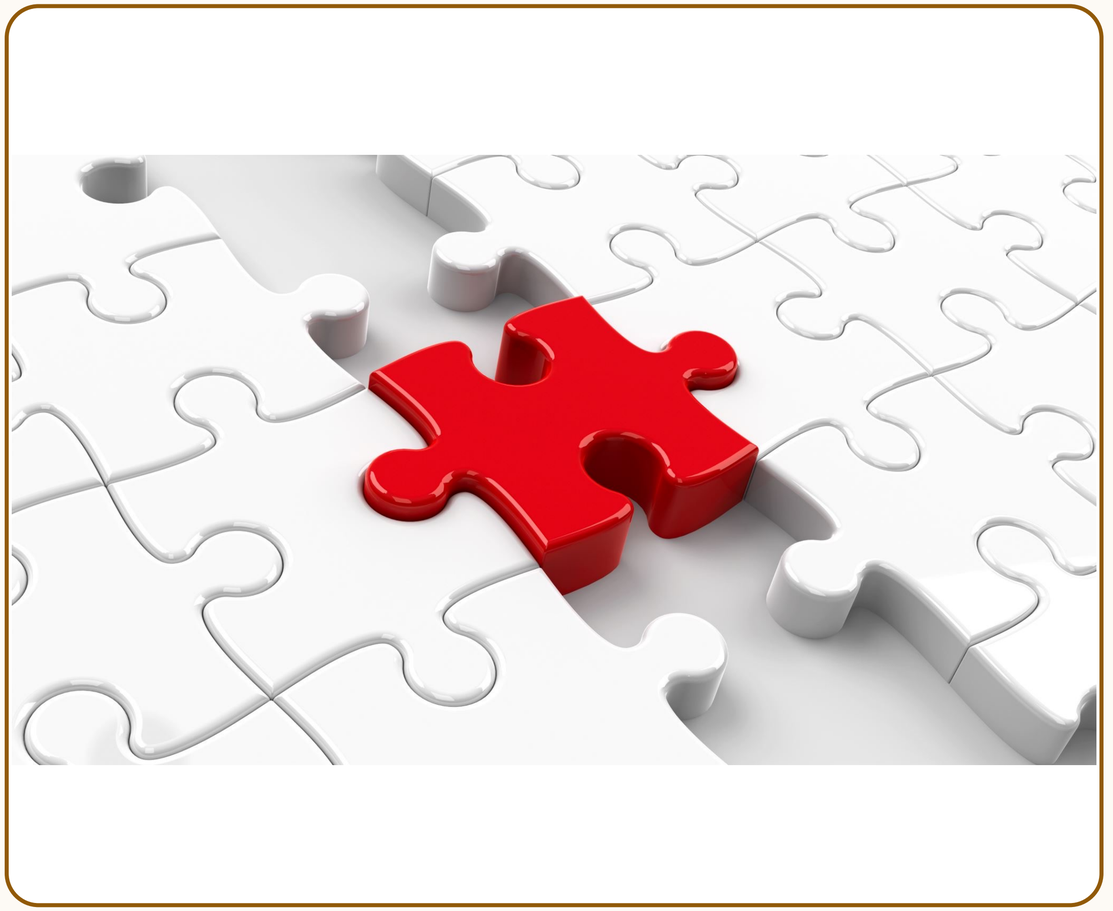
Logic interoperability

Common representation to capture

- Different SQL dialects
- View definitions
- Different engine plan representations
- SQL pushdown between engines
- Common query transformations

Adapters to transform

- From an input representation
- To an output representation



Composable Data Architectures

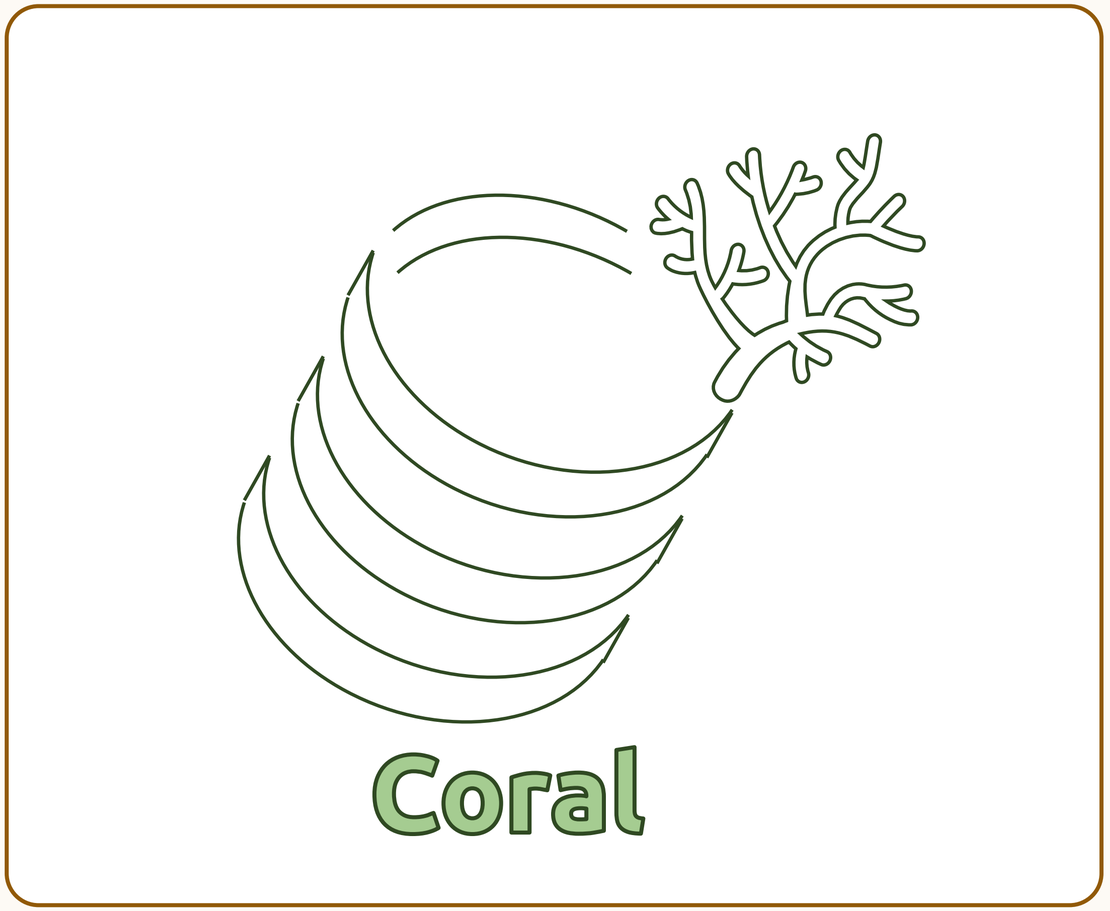
Coral

Common representation to capture

- Different SQL dialects
- View definitions
- Different engine plan representations
- SQL pushdown between engines
- Common query transformations

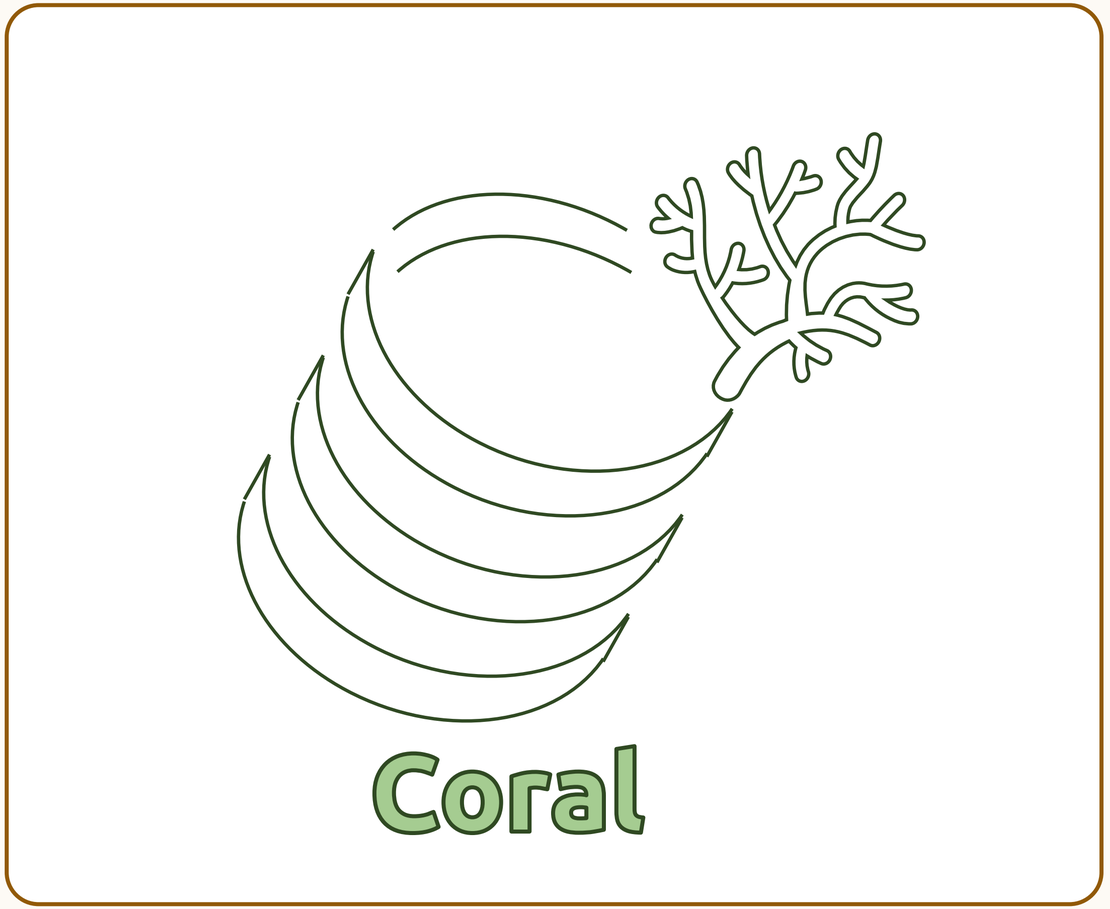
Adapters to transform

- From an input representation
- To an output representation



Coral

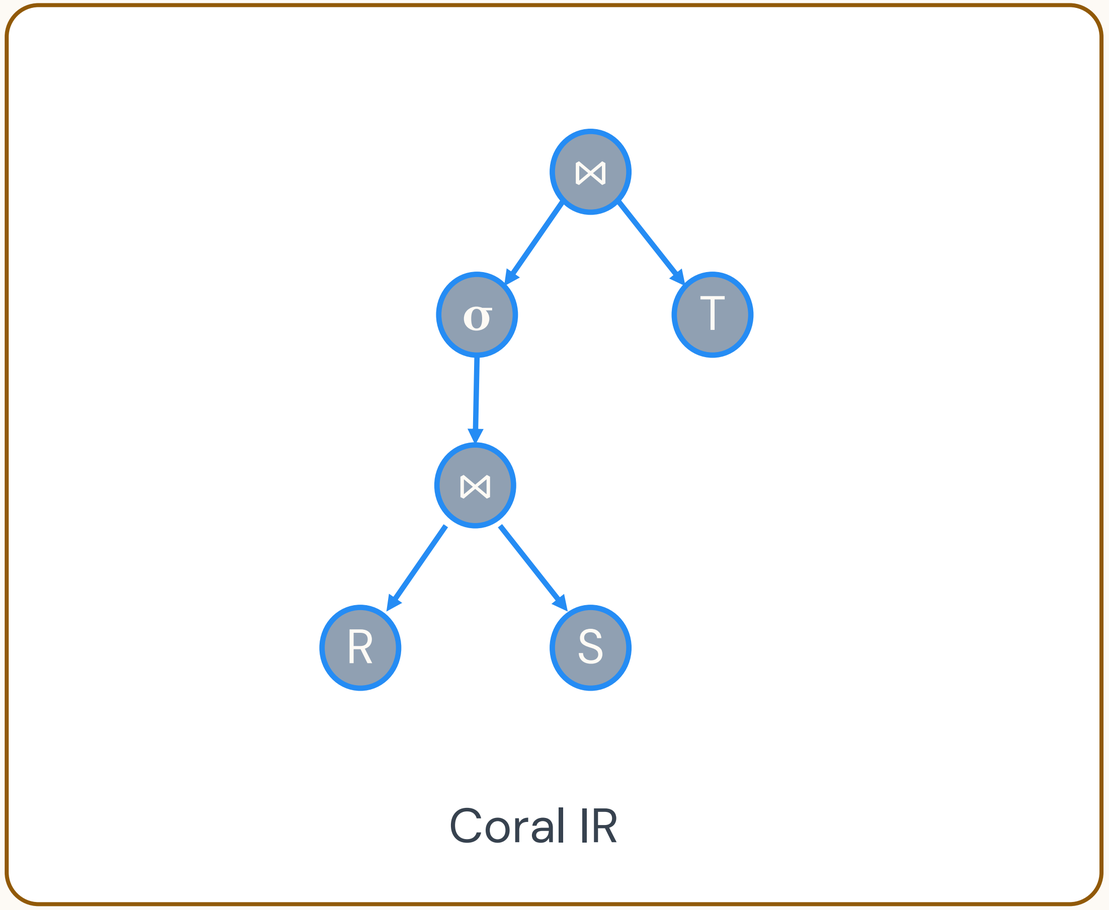
- Open-source project since 2020
- https://github.com/linkedin/coral
- Extends Calcite logical plan to represent logic
- Intermediate representation called Coral IR



Coral

IR, Transformations

- Coral IR captures query semantics using standard operators
- Supported Transformations
  - Hive QL (optionally Spark SQL) to Coral IR
  - Trino SQL to Coral IR (WIP)
  - Coral IR to Trino SQL
  - Coral IR to Spark SQL (optionally Hive QL)
  - Coral IR to Avro schema



Example

Spark SQL

Example Query

```
SELECT instr(R.x[0], 'foo')
FROM   R
WHERE  ! y
```

Operators

- instr(a, b): returns index of b in a
- x[i]: returns element i in array x, 0-based index
- ! y: negates y

Example

Trino SQL

Example Query

```
SELECT strpos(element_at(R.x, 1), 'foo')
FROM   R
WHERE  NOT y
```

Operators

- strpos(a, b): returns index of b in a
- element_at(x, i): returns element i in array x, 1-based index
- Not y: negates y

Transformations

Spark SQL to Coral IR conversion

Spark SQL		Coral IR
instr(x, y)	→	instr(x, y)
x[i]	→	x[i+1]
!x	→	NOT x

Transformations

Coral IR conversion to Trino SQL conversion

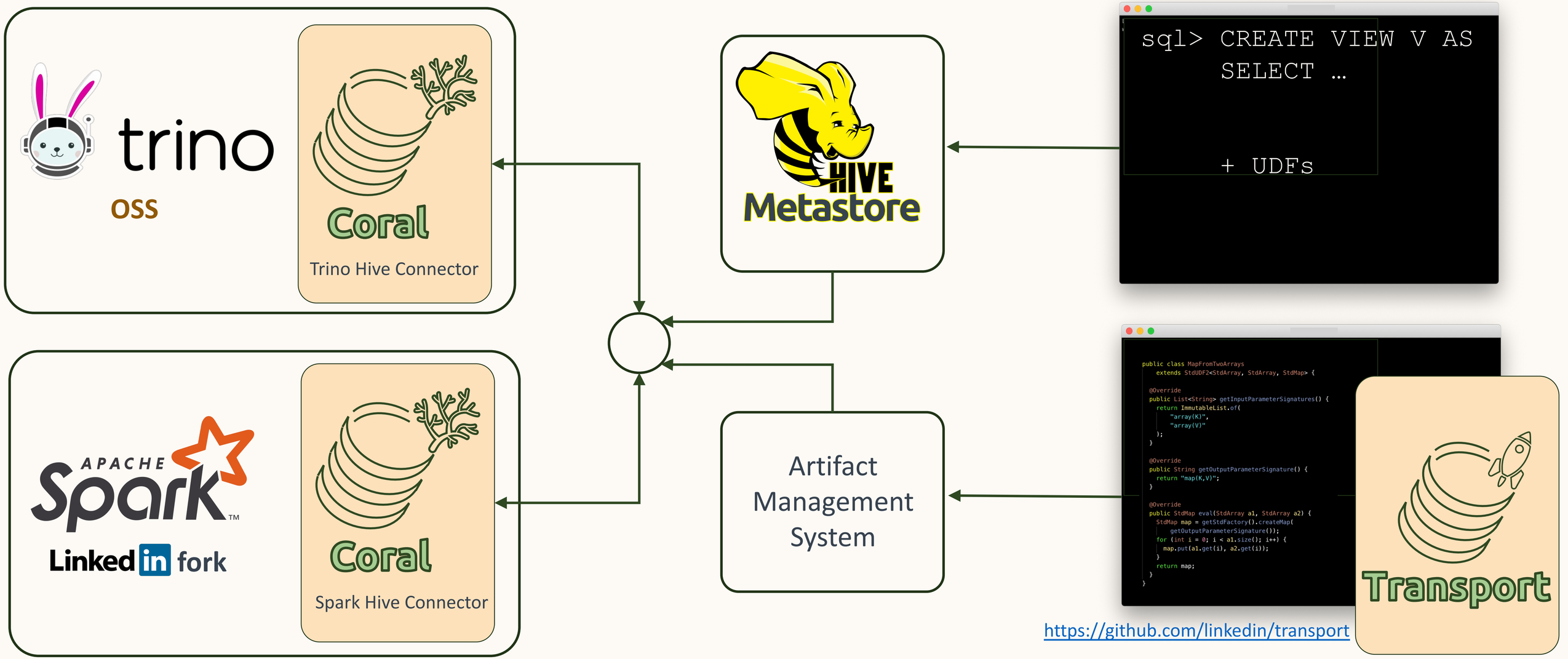
Coral IR		Trino SQL
instr(x, y)	→	strpos(x, y)
x[i]	→	element_at(x, i)
NOT x	→	NOT x

Transformations

More complex transformations

- Lateral view joins
- User defined table functions
- Window functions
- Common table expressions

Notable Integrations



https://github.com/linkedin/transport

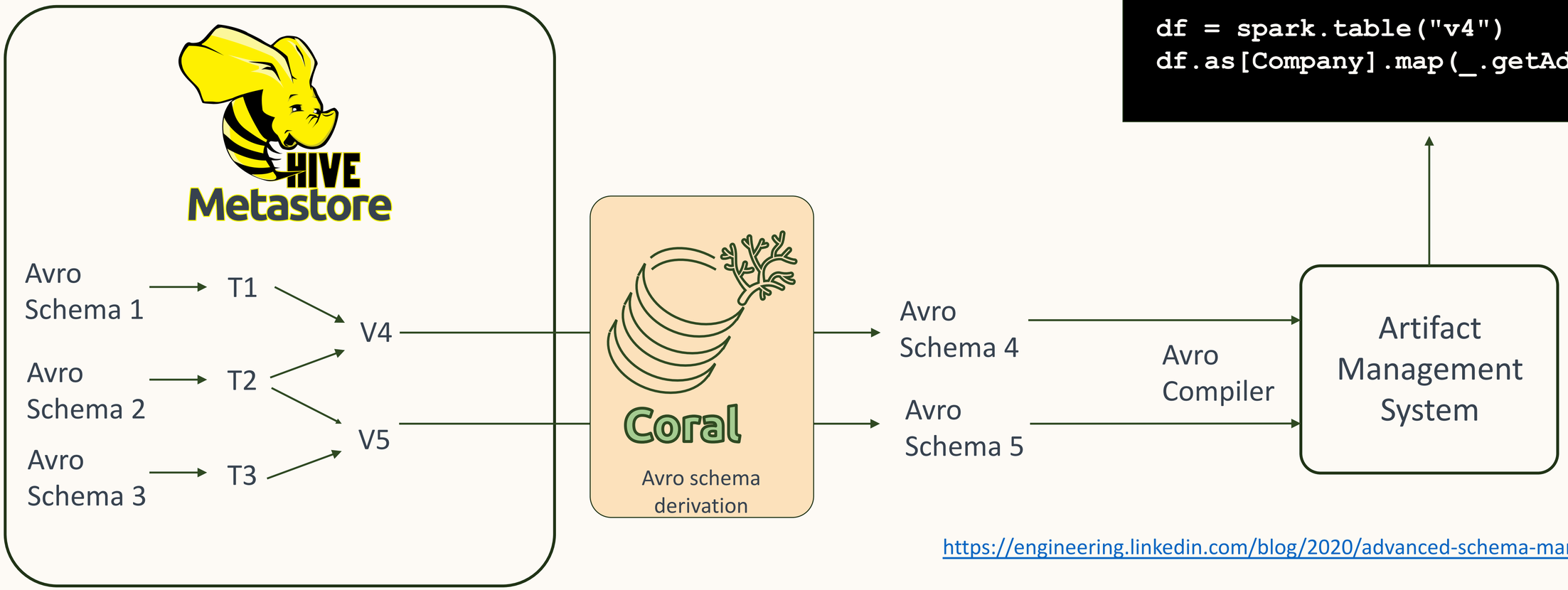
Notable Integrations

Type-safe programming in Spark


```
build.gradle:
compile com.linkedin.schema:v4:+

script.scala:
import my.company.example.*

df = spark.table("v4")
df.as[Company].map(_.getAddress())
```

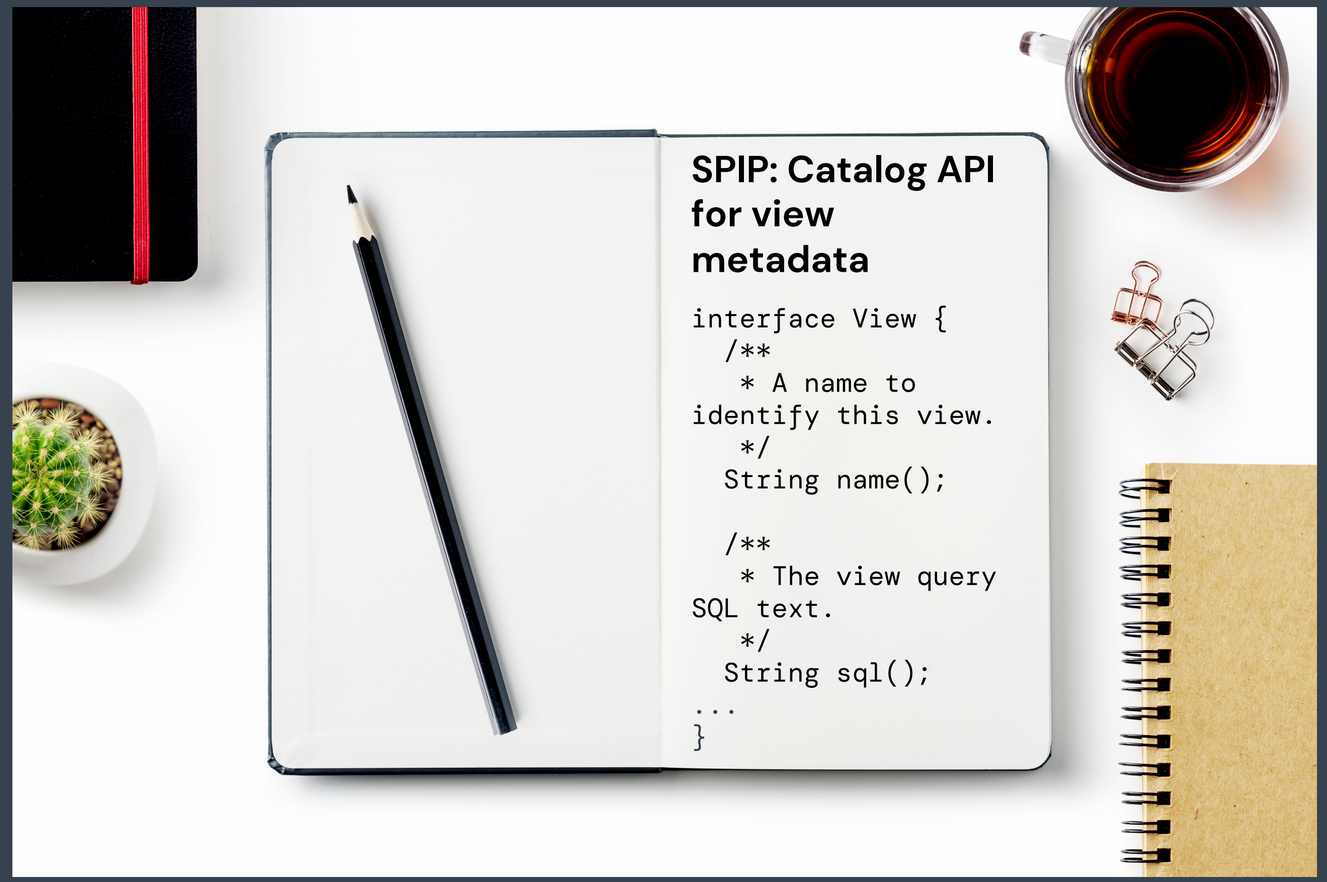


https://engineering.linkedin.com/blog/2020/advanced-schema-management-for-spark

Apache Spark Integration

SPARK-31357

- Spark improvement to introduce top-level view abstractions
  - ViewCatalog API
  - View API
- Enable custom implementations for view SQL and schema resolution
- Envision Coral integration to Apache Spark through this API



Standalone Mode

Coral-as-a-service

```
$ curl --header "Content-Type: application/json" \
  --request POST \
  --data '{
    "fromLanguage": "hive",
    "toLanguage": "trino",
    "query": "SELECT * FROM db1.airport"
  }' http://localhost:8080/api/translations/translate
```

Try it today! https://github.com/linkedin/coral

Future Extensions

- Spark catalyst plan to Coral IR
  - POC in Coral-Spark-Plan
  - Enables translation of all Spark APIs
    - Scala
    - Java
    - Python
- Common query rewrites
  - Materialized view substitution
  - Incremental view maintenance
  - Data governance (e.g., automatic obfuscation of PII)

Future Extensions

Engine Data Source Connector

- Engine data source integration
  - Push functions to data sources
    - Delta Lake
    - Iceberg
  - Push SQL expressions across SQL engines
    - Spark
    - Trino
    - Presto
    - Pinot

Thank you

https://github.com/linkedin/coral

https://coral-sql.slack.com