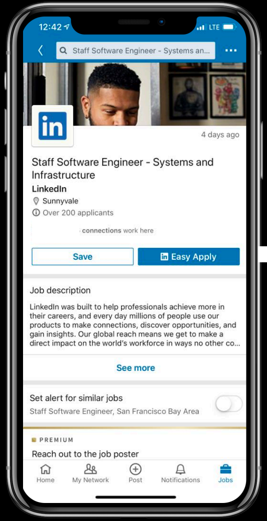
Coral & Transport UDFs

Building Blocks of a Postmodern Data Warehouse

Walaa Eldin Moustafa

Staff Software Engineer @ LinkedIn

The lifecycle of a data application at LinkedIn *[Simplified]*



12:42 LTE

Staff Software Engineer - Systems an...

4 days ago

Staff Software Engineer - Systems and Infrastructure

LinkedIn

Sunnyvale

Over 200 applicants

connections work here

Save | Easy Apply

Job description

LinkedIn was built to help professionals achieve more in their careers, and every day millions of people use our products to make connections, discover opportunities, and gain insights. Our global reach means we get to make a direct impact on the world's workforce in ways no other co...

See more

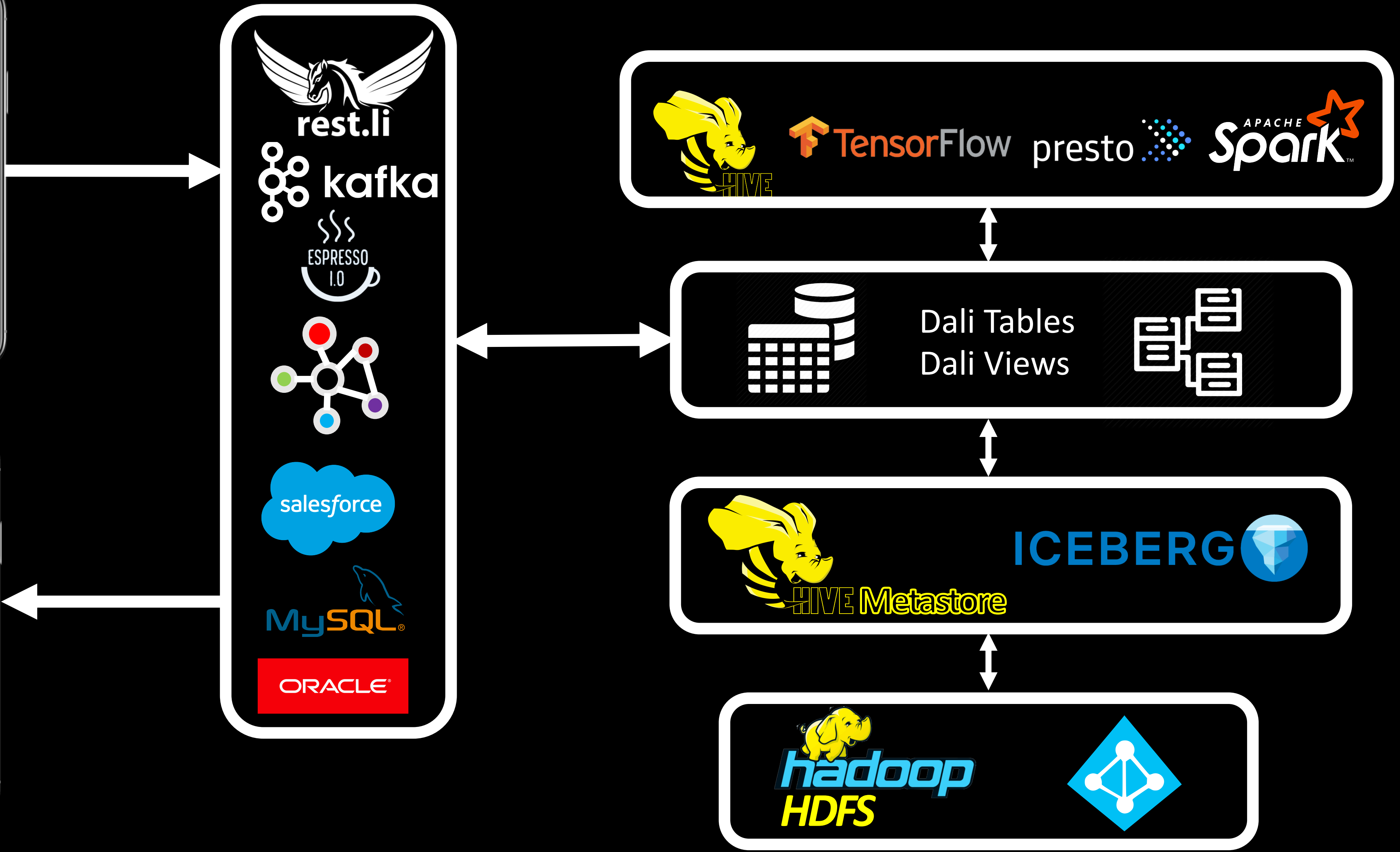
Set alert for similar jobs

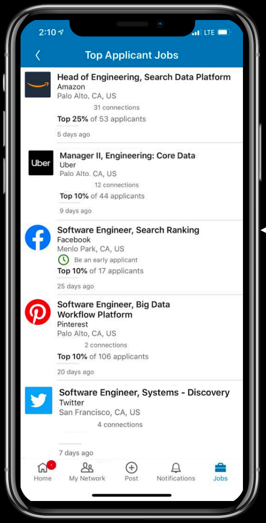
Staff Software Engineer, San Francisco Bay Area

PREMIUM

Reach out to the job poster

Home | My Network | Post | Notifications | Jobs





2:10 LTE

Top Applicant Jobs

Head of Engineering, Search Data Platform
Amazon
Palo Alto, CA, US
31 connections
Top 25% of 53 applicants
5 days ago

Manager II, Engineering: Core Data
Uber
Palo Alto, CA, US
12 connections
Top 10% of 44 applicants
9 days ago

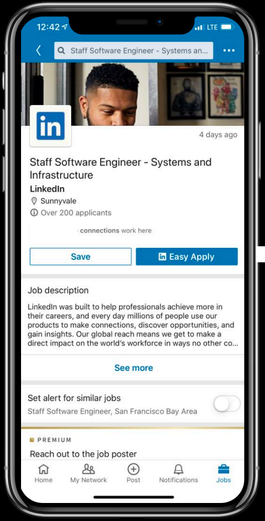
Software Engineer, Search Ranking
Facebook
Menlo Park, CA, US
Be an early applicant
Top 10% of 17 applicants
25 days ago

Software Engineer, Big Data Workflow Platform
Pinterest
Palo Alto, CA, US
2 connections
Top 10% of 106 applicants
20 days ago

Software Engineer, Systems - Discovery
Twitter
San Francisco, CA, US
4 connections
7 days ago

Home | My Network | Post | Notifications | Jobs

The lifecycle of a data application at LinkedIn *[Simplified]*



12:42 LTE

Staff Software Engineer - Systems an...

4 days ago

Staff Software Engineer - Systems and Infrastructure

LinkedIn

Sunnyvale

Over 200 applicants

connections work here

Save | Easy Apply

Job description

LinkedIn was built to help professionals achieve more in their careers, and every day millions of people use our products to make connections, discover opportunities, and gain insights. Our global reach means we get to make a direct impact on the world's workforce in ways no other co...

See more

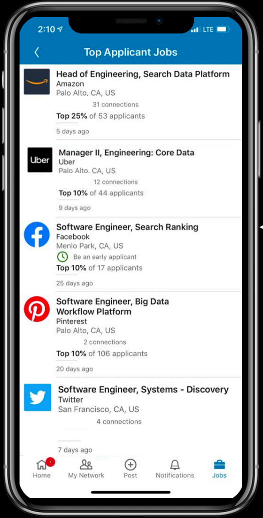
Set alert for similar jobs

Staff Software Engineer, San Francisco Bay Area

PREMIUM

Reach out to the job poster

Home | My Network | Post | Notifications | Jobs



2:10 LTE

Top Applicant Jobs

Head of Engineering, Search Data Platform
Amazon
Palo Alto, CA, US
31 connections
Top 25% of 53 applicants
5 days ago

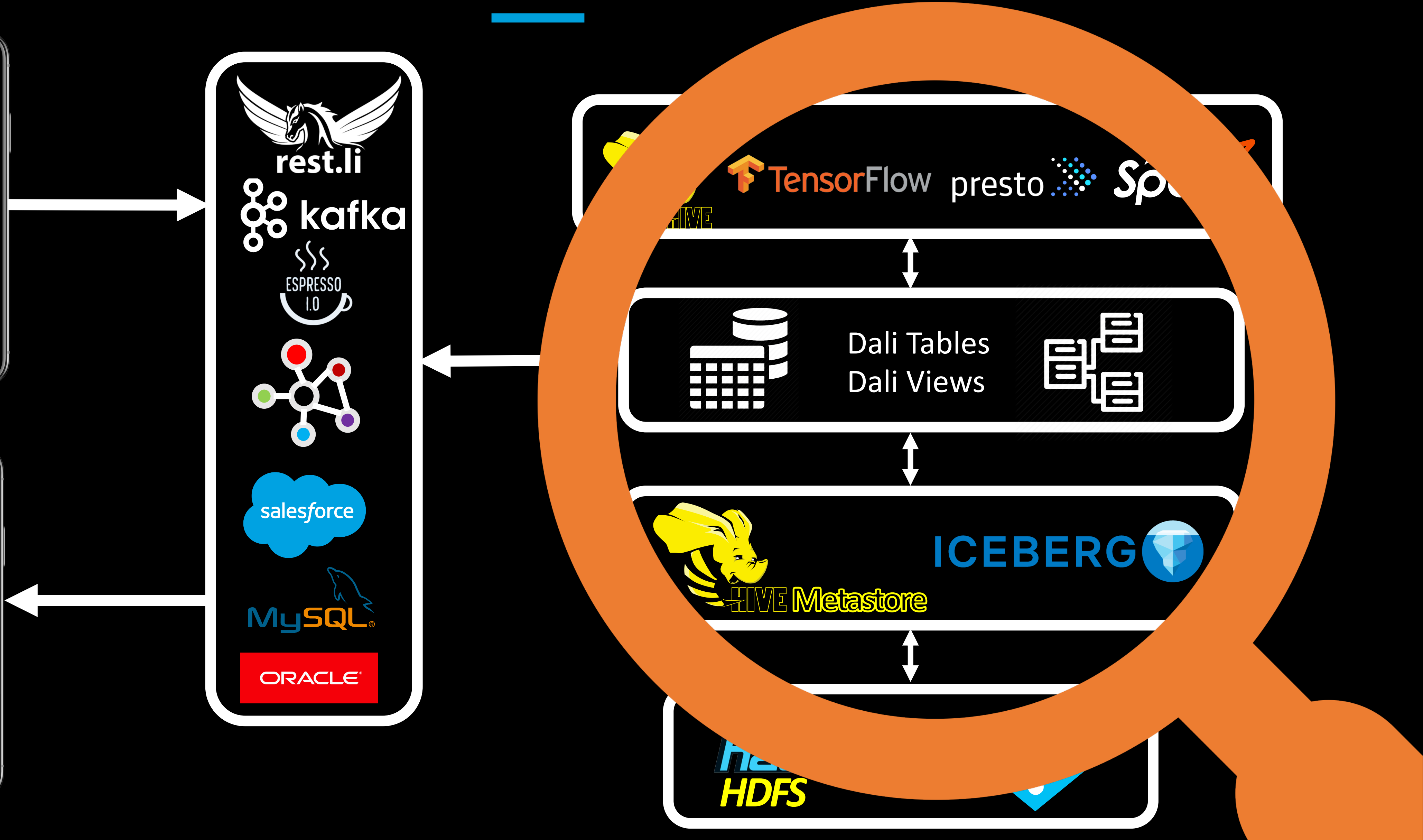
Manager II, Engineering: Core Data
Uber
Palo Alto, CA, US
12 connections
Top 10% of 44 applicants
9 days ago

Software Engineer, Search Ranking
Facebook
Menlo Park, CA, US
Be an early applicant
Top 10% of 17 applicants
25 days ago

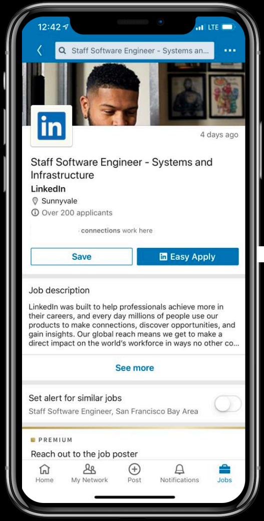
Software Engineer, Big Data Workflow Platform
Pinterest
Palo Alto, CA, US
2 connections
Top 10% of 106 applicants
20 days ago

Software Engineer, Systems - Discovery
Twitter
San Francisco, CA, US
4 connections
7 days ago

Home | My Network | Post | Notifications | Jobs



The lifecycle of a data application at LinkedIn *[Simplified]*



12:42 LTE

Staff Software Engineer - Systems an...

4 days ago

Staff Software Engineer - Systems and Infrastructure

LinkedIn

Sunnyvale

Over 200 applicants

connections work here

Save | Easy Apply

Job description

LinkedIn was built to help professionals achieve more in their careers, and every day millions of people use our products to make connections, discover opportunities, and gain insights. Our global reach means we get to make a direct impact on the world's workforce in ways no other co...

See more

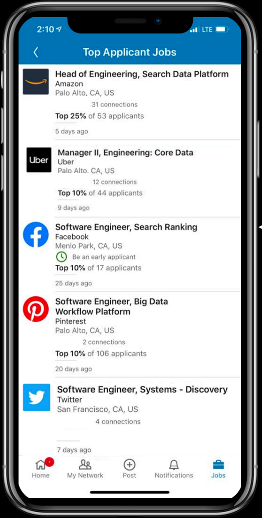
Set alert for similar jobs

Staff Software Engineer, San Francisco Bay Area

PREMIUM

Reach out to the job poster

Home | My Network | Post | Notifications | Jobs



2:10 LTE

Top Applicant Jobs

Head of Engineering, Search Data Platform
Amazon
Palo Alto, CA, US
31 connections
Top 25% of 53 applicants
5 days ago

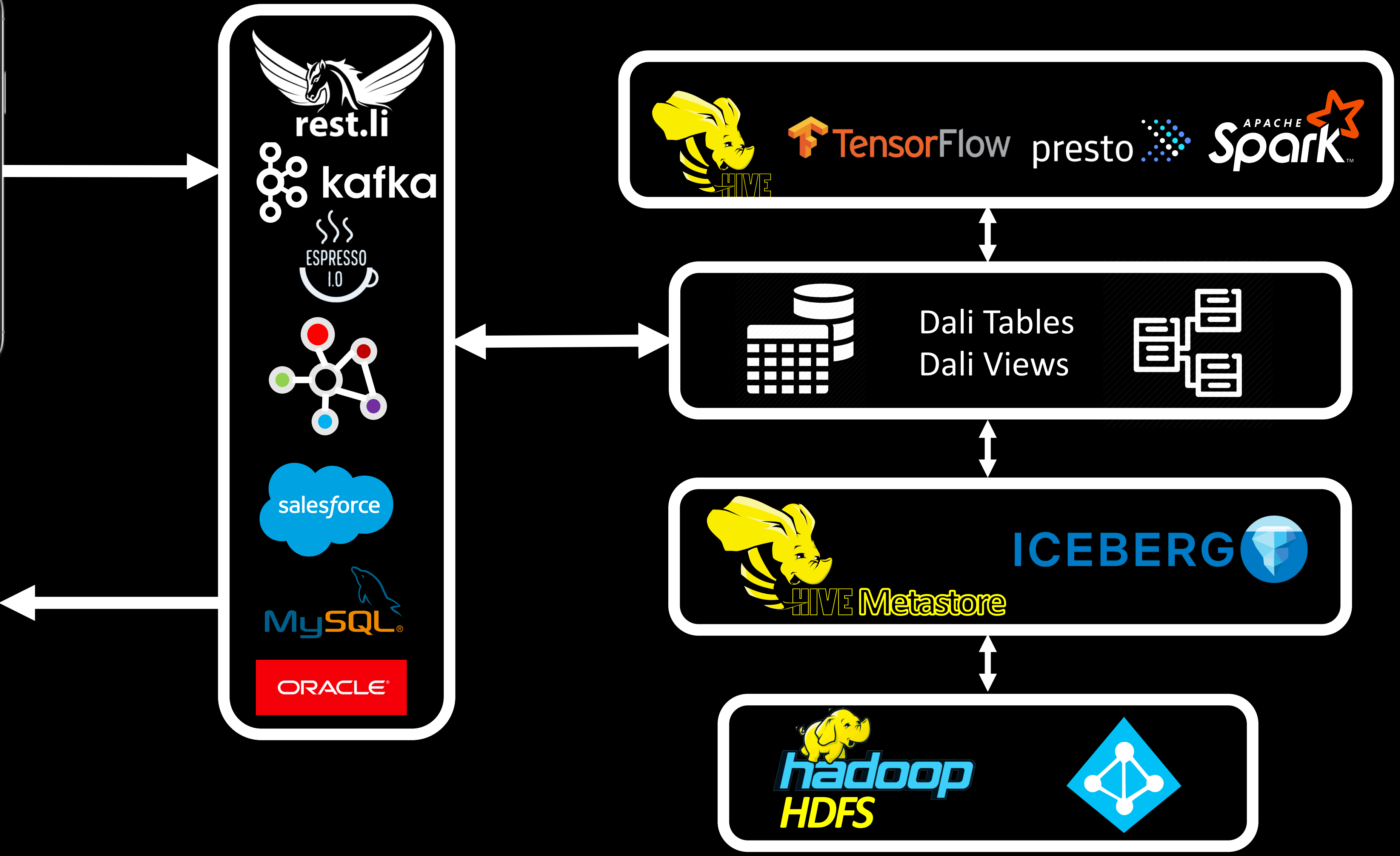
Manager II, Engineering: Core Data
Uber
Palo Alto, CA, US
12 connections
Top 10% of 44 applicants
9 days ago

Software Engineer, Search Ranking
Facebook
Menlo Park, CA, US
Be an early applicant
Top 10% of 17 applicants
25 days ago

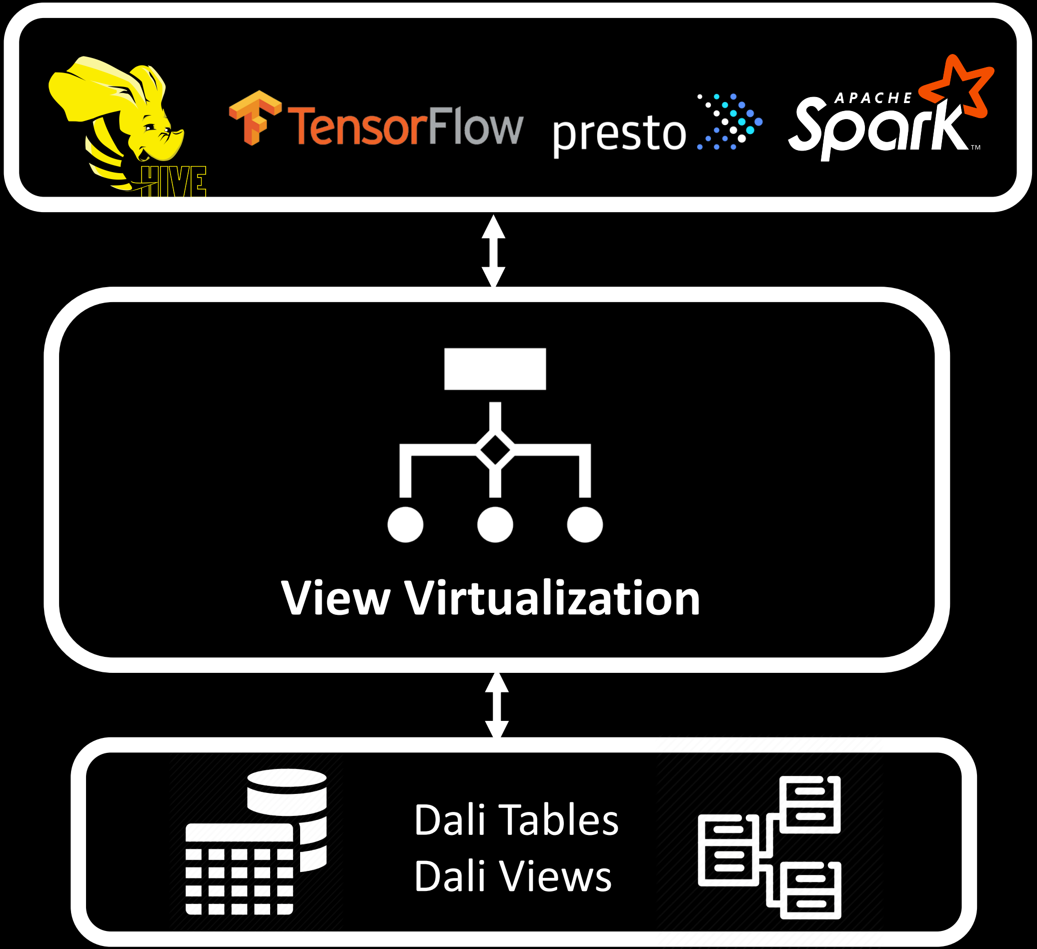
Software Engineer, Big Data Workflow Platform
Pinterest
Palo Alto, CA, US
2 connections
Top 10% of 106 applicants
20 days ago

Software Engineer, Systems - Discovery
Twitter
San Francisco, CA, US
4 connections
7 days ago

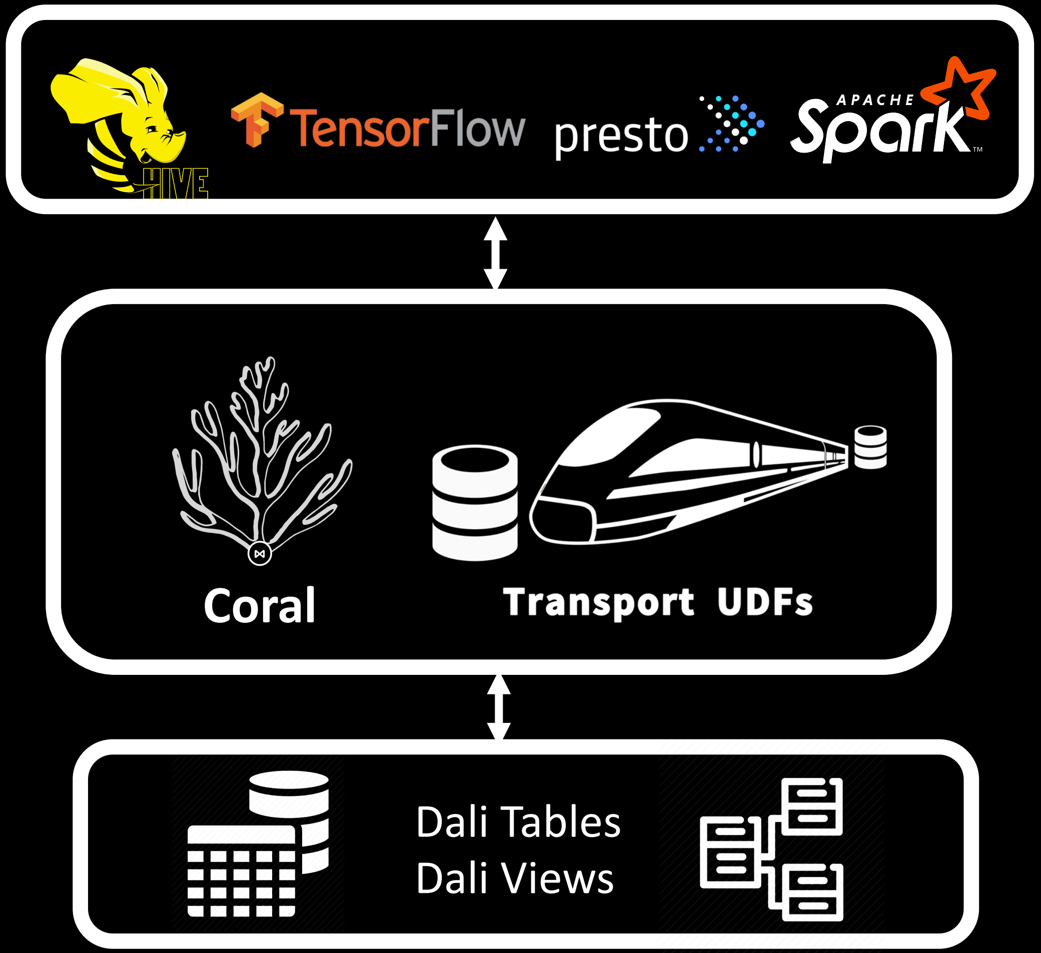
Home | My Network | Post | Notifications | Jobs



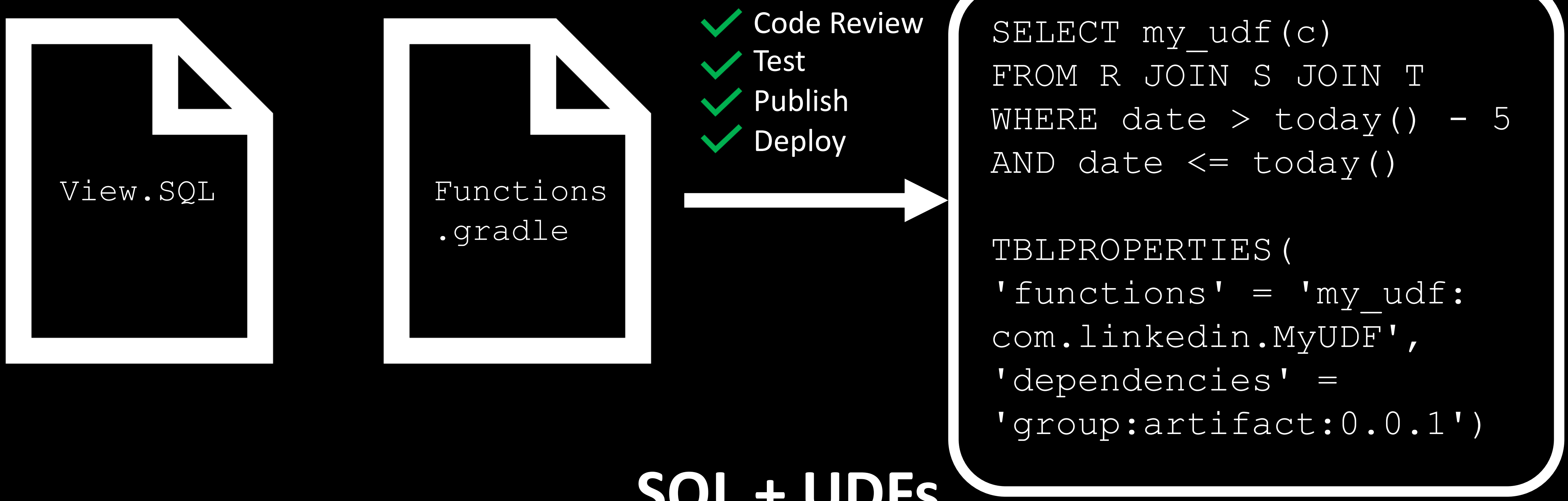
View Virtualization



View Virtualization



Dali Views

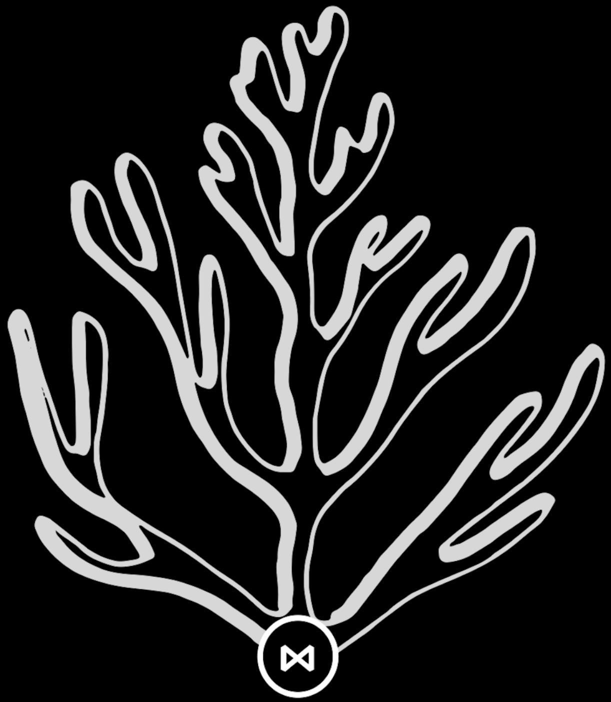


```
SELECT my_udf(c)
FROM R JOIN S JOIN T
WHERE date > today() - 5
AND date <= today()

TBLPROPERTIES(
'functions' = 'my_udf:
com.linkedin.MyUDF',
'dependencies' =
'group:artifact:0.0.1')
```

SQL + UDFs

Coral

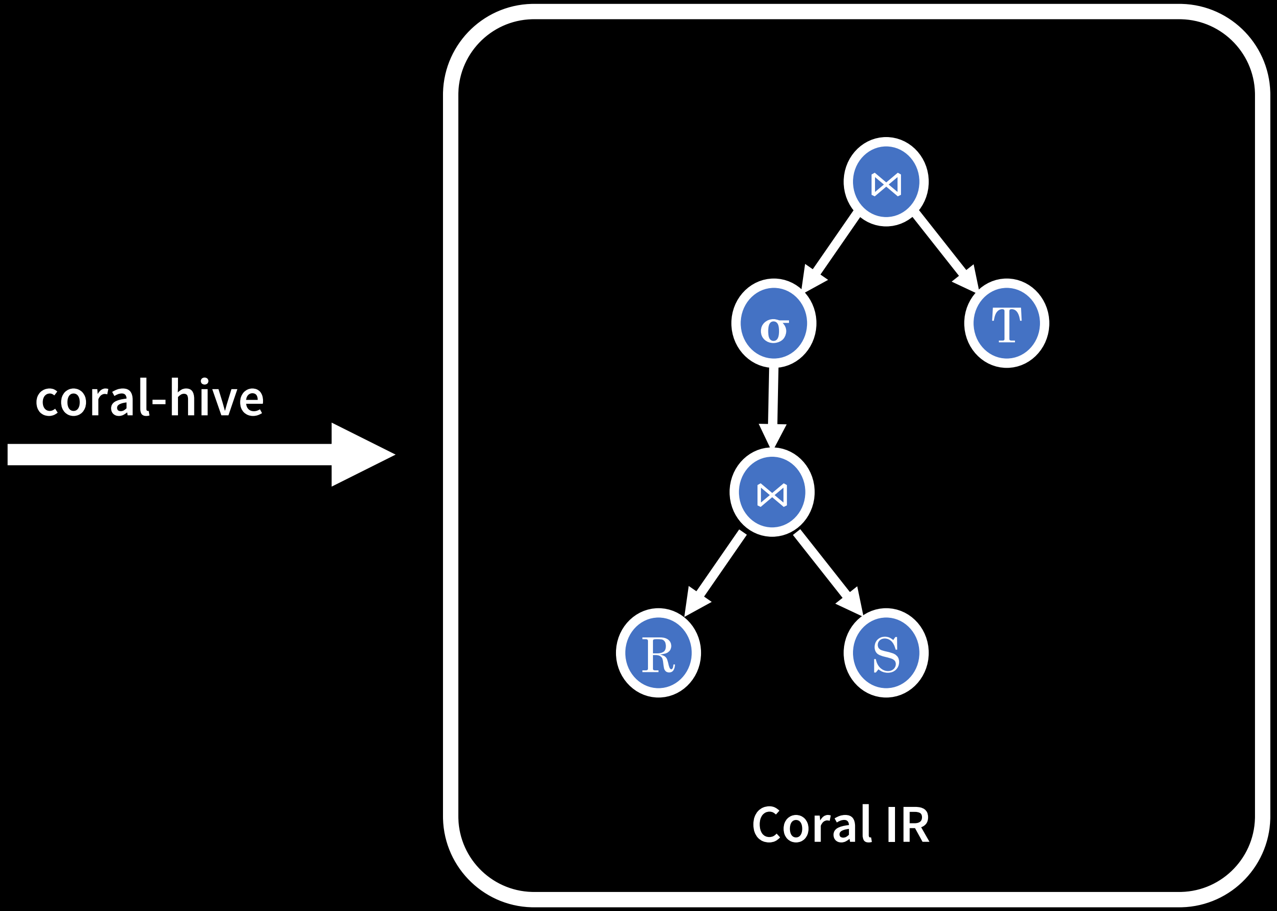


github.com/linkedin/coral

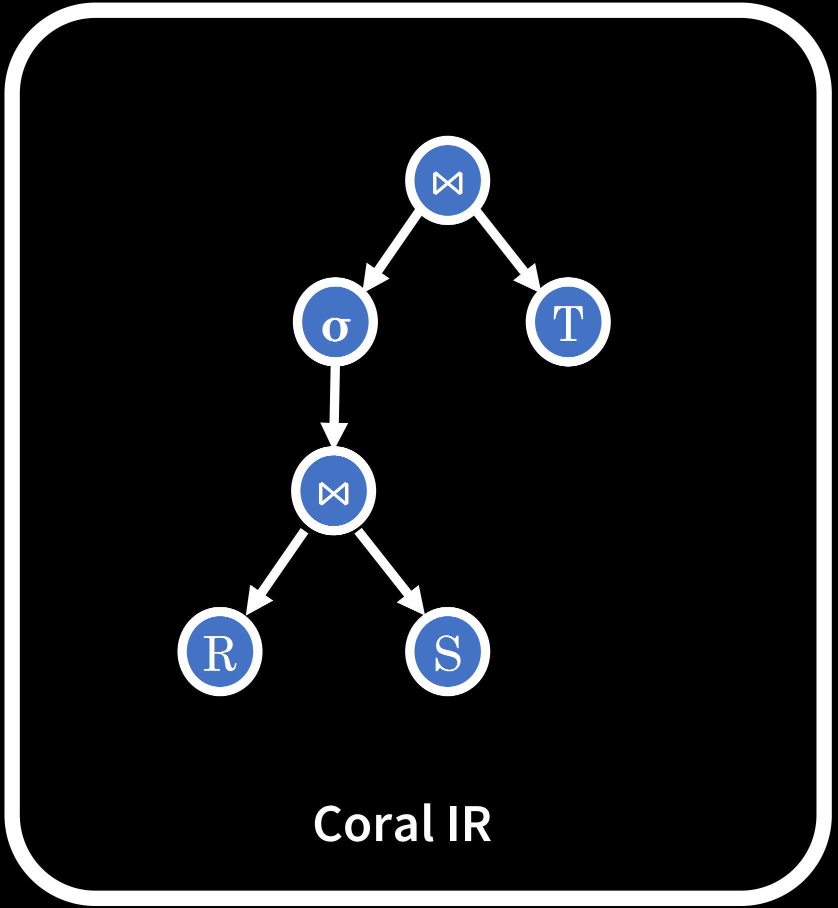
Coral

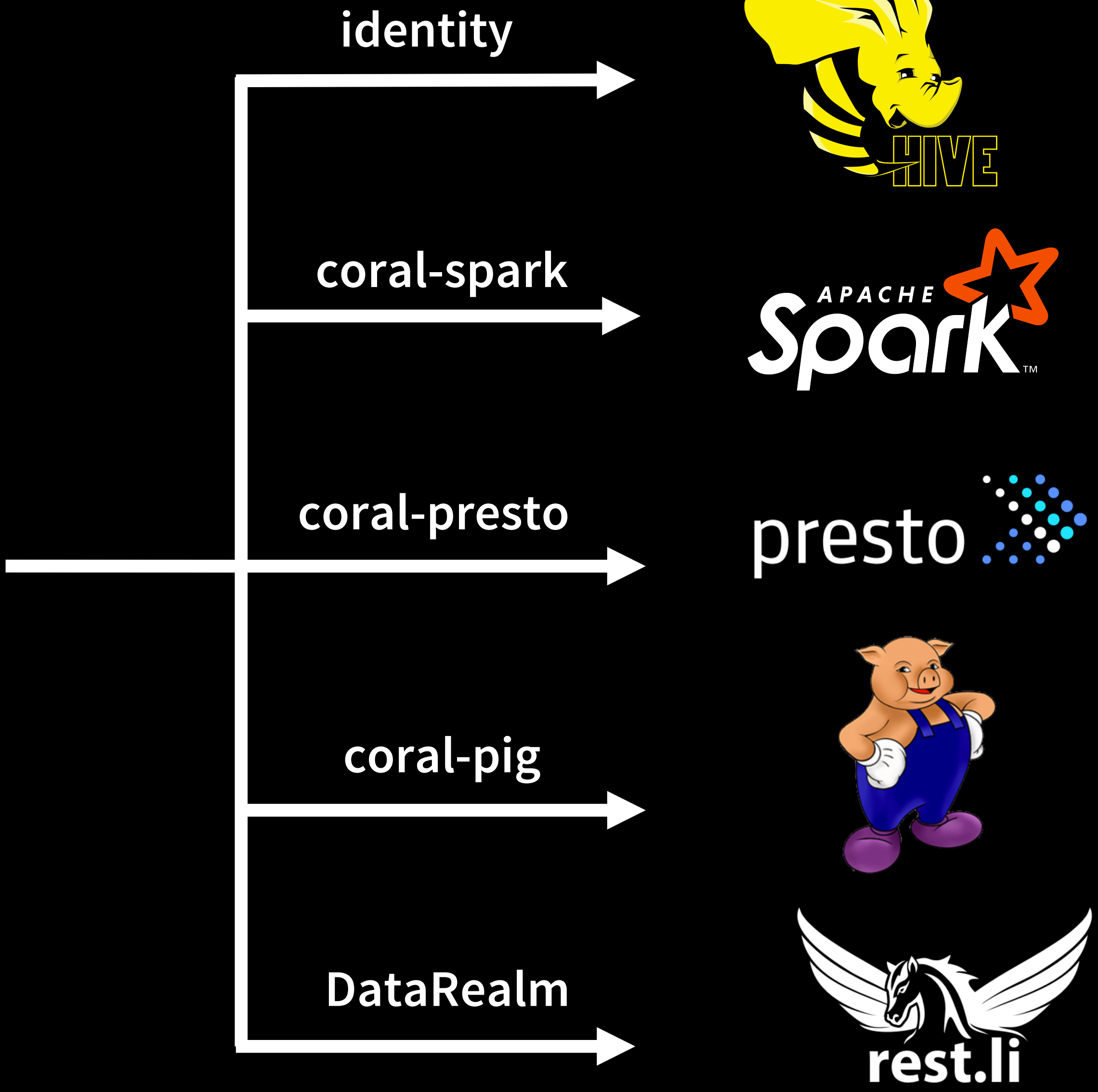
```
SELECT my_udf(c)
FROM R JOIN S JOIN T
WHERE date > today() - 5
AND date <= today()

TBLPROPERTIES(
'functions' = 'my_udf',
'dependencies' =
'group:artifact:0.0.1')
```

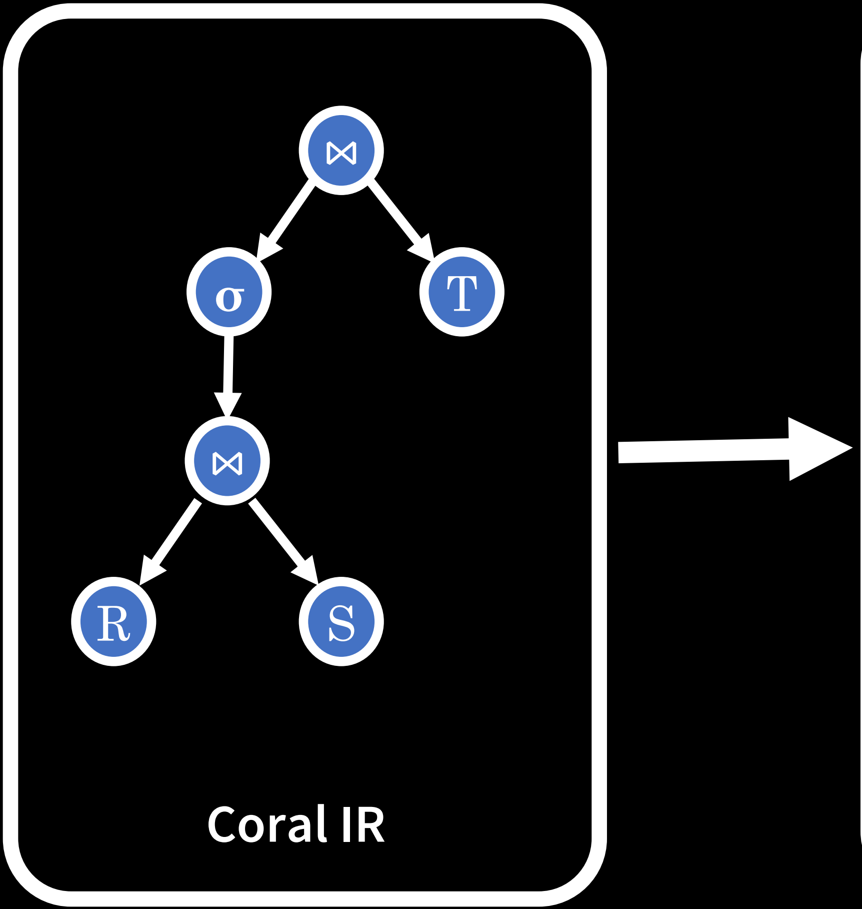


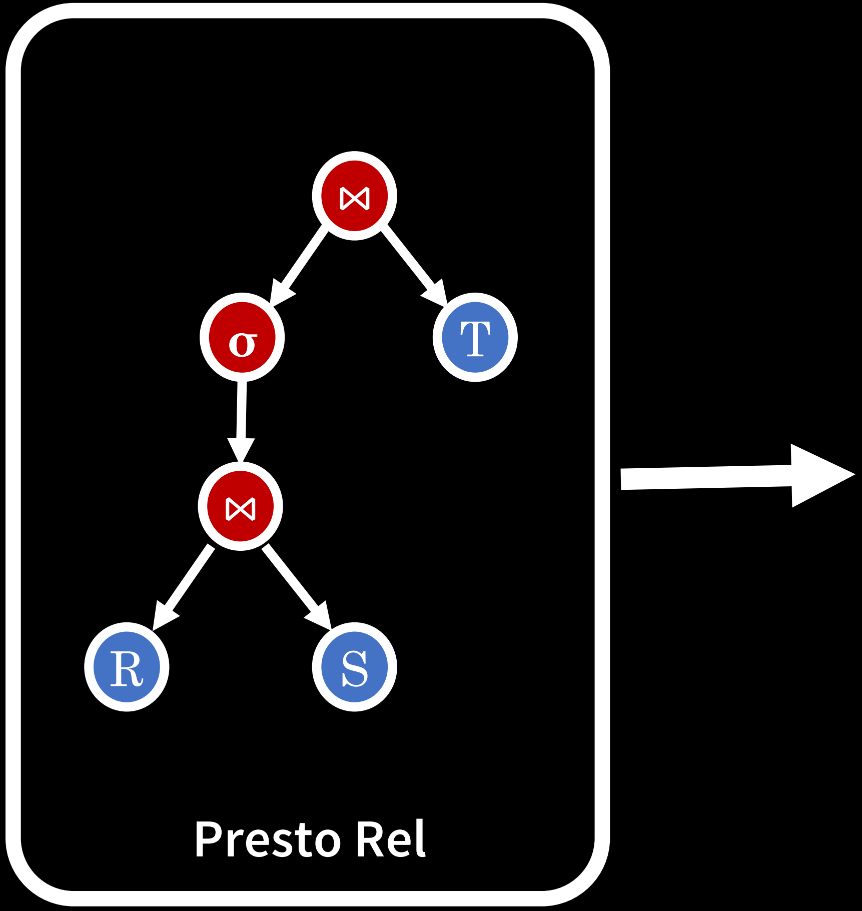
Coral


⋈
σ
T
⋈
R
S
Coral IR


identity
coral-spark
coral-presto
coral-pig
DataRealm

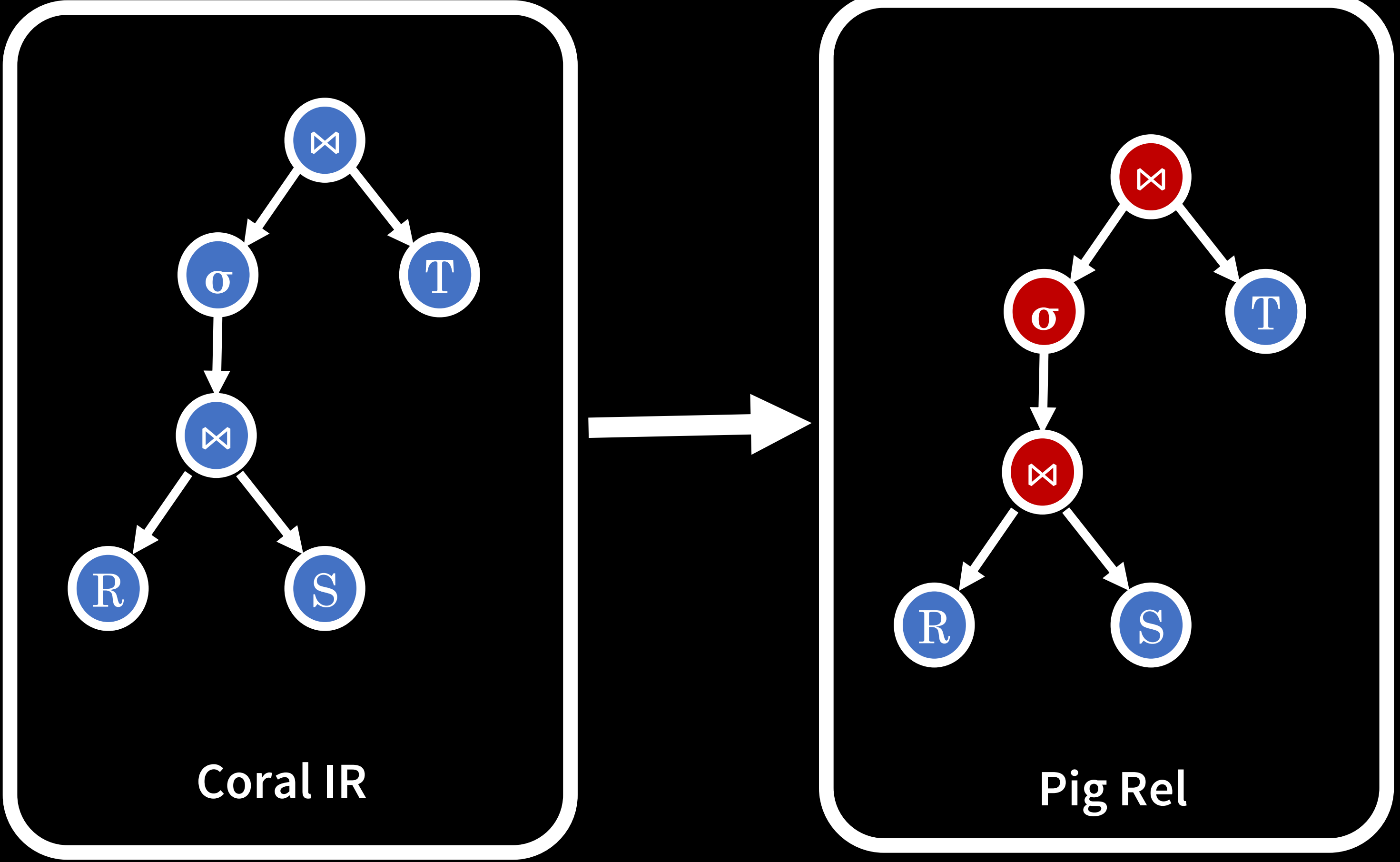
Coral-Presto





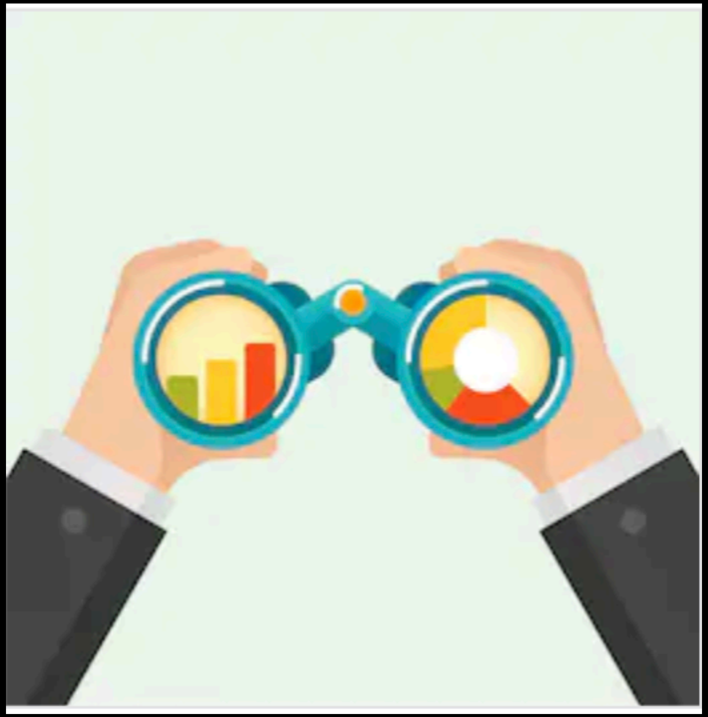
```
SELECT presto_udf(c + 1)
FROM R JOIN S JOIN T
WHERE date > today() - 5
AND date <= today()
```

Coral-Pig



```
R = LOAD "R"
S = LOAD "S"
T = LOAD "T"
RS = R JOIN S
RSF = FILTER RS BY ...
RSFT = RSF JOIN T
```

What does the future look like?



Flexible Language Interface

```
user {
  name
address{}
}
```

GraphQL

```
A = LOAD
B = LOAD
C = A
JOIN B
```

Pig Latin

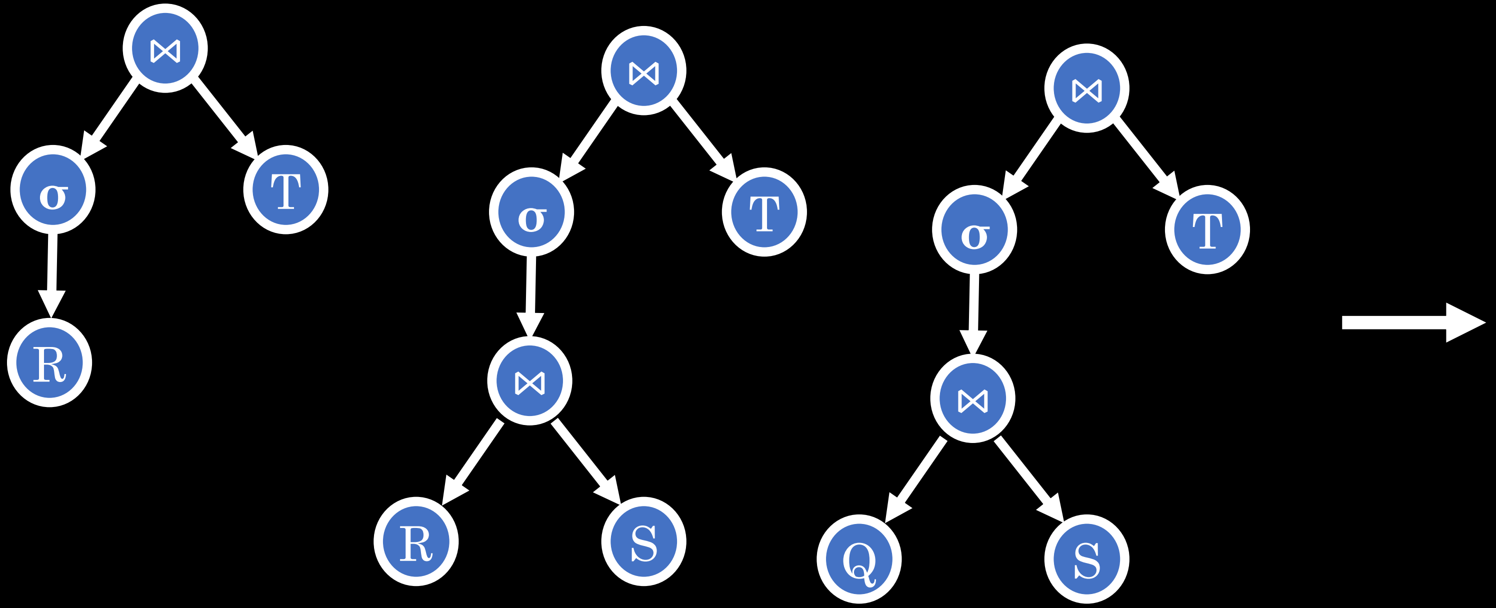
```
d = load()
.filter()
.map()
.count()
```

Dataflow

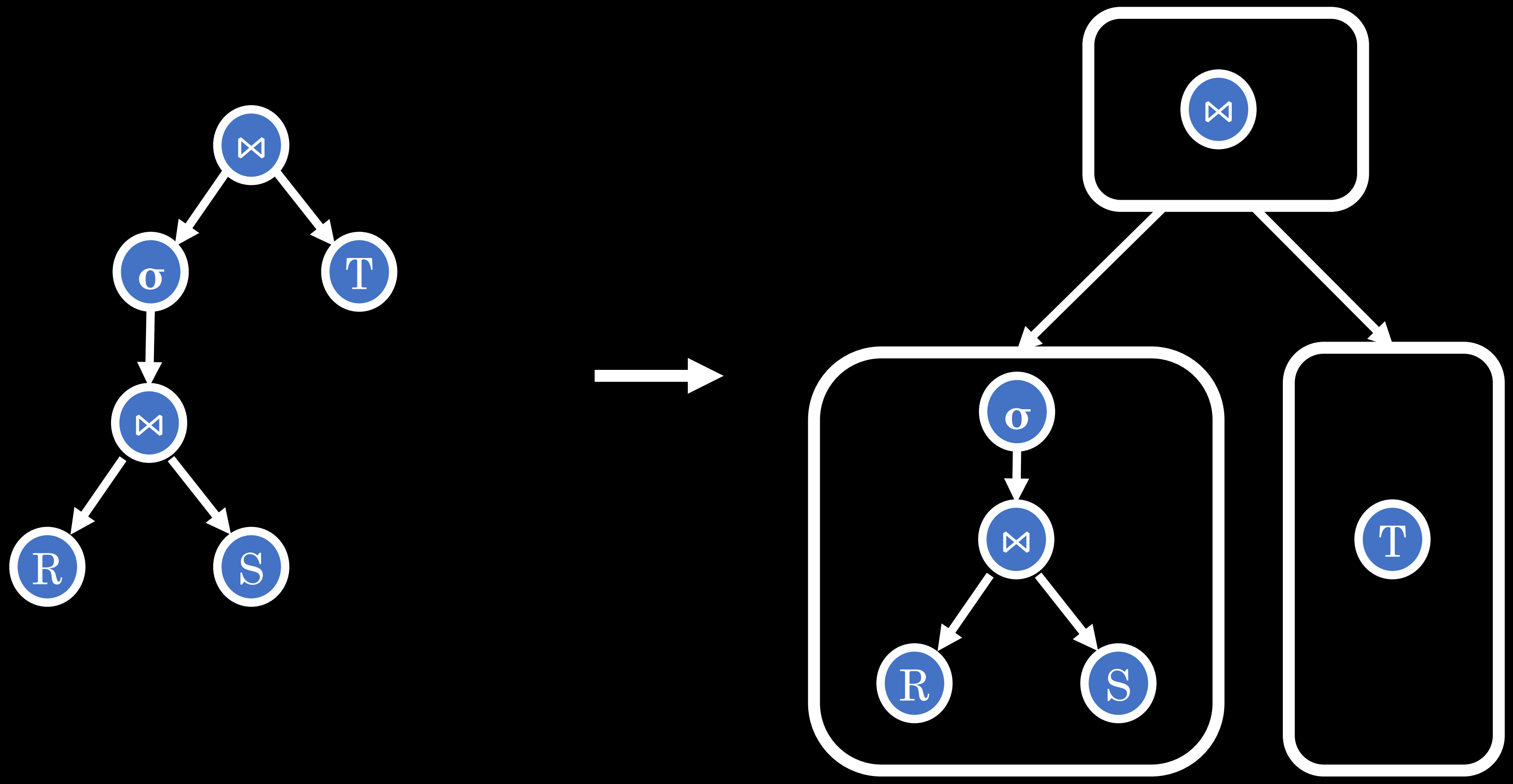
```
TwoHop
(X,Y) :-
E(X,Z),
E(Z,Y).
```

Datalog

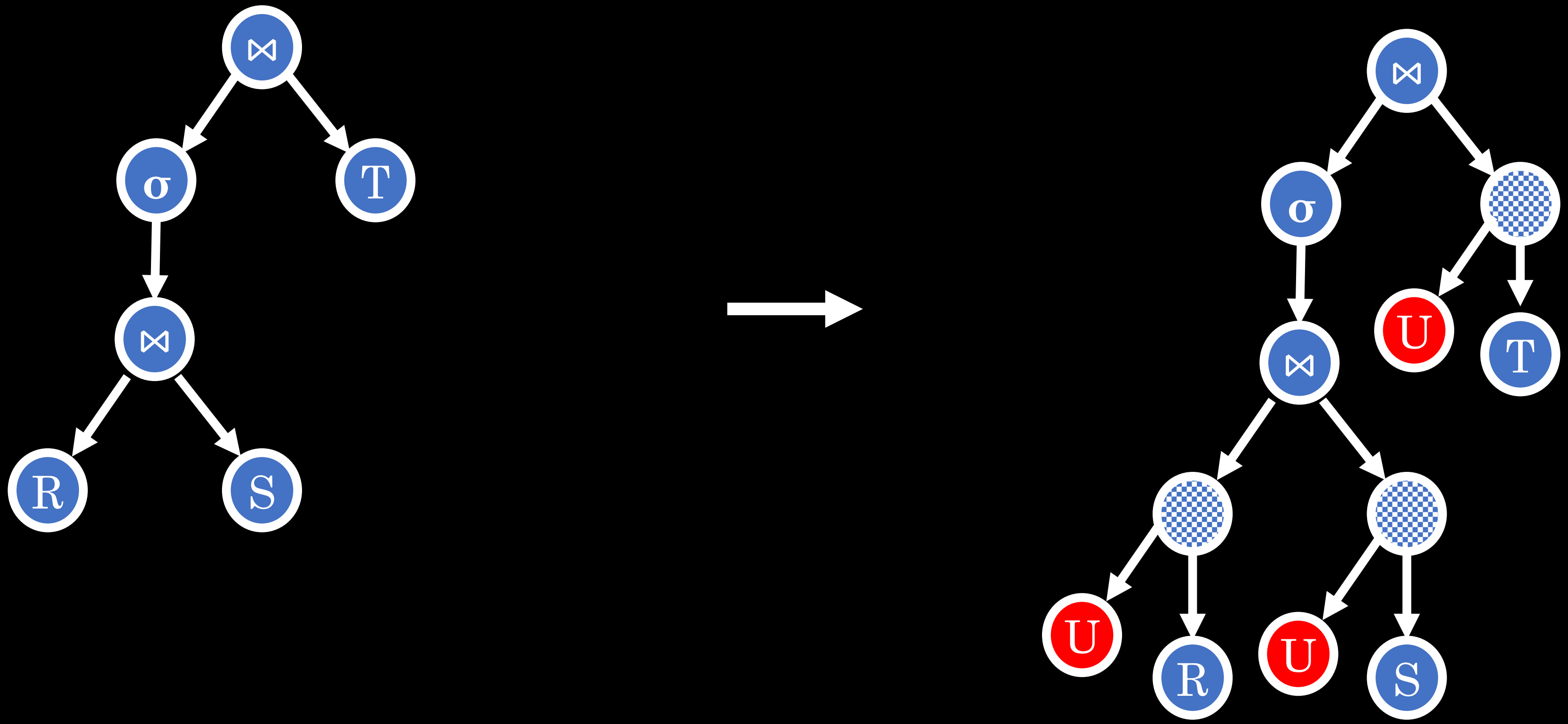
Workload Analytics



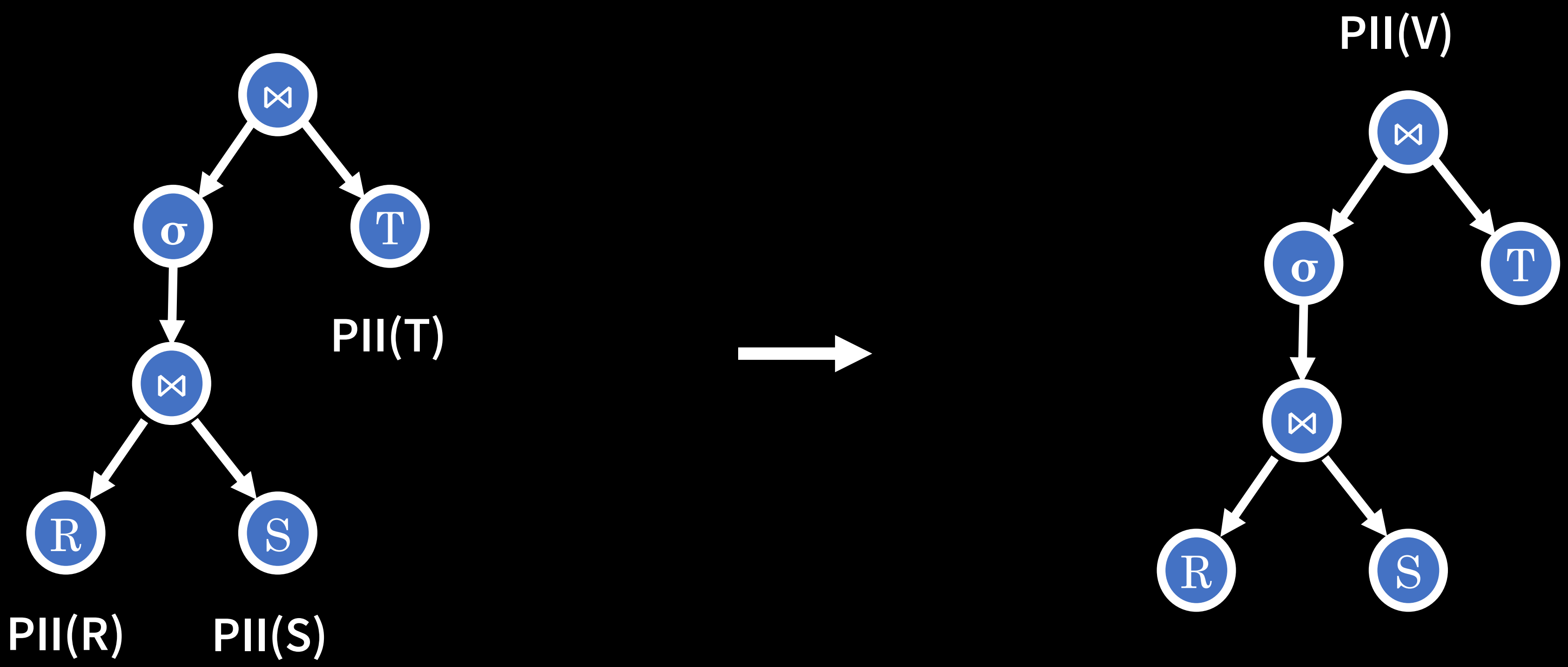
Materialized View Selection



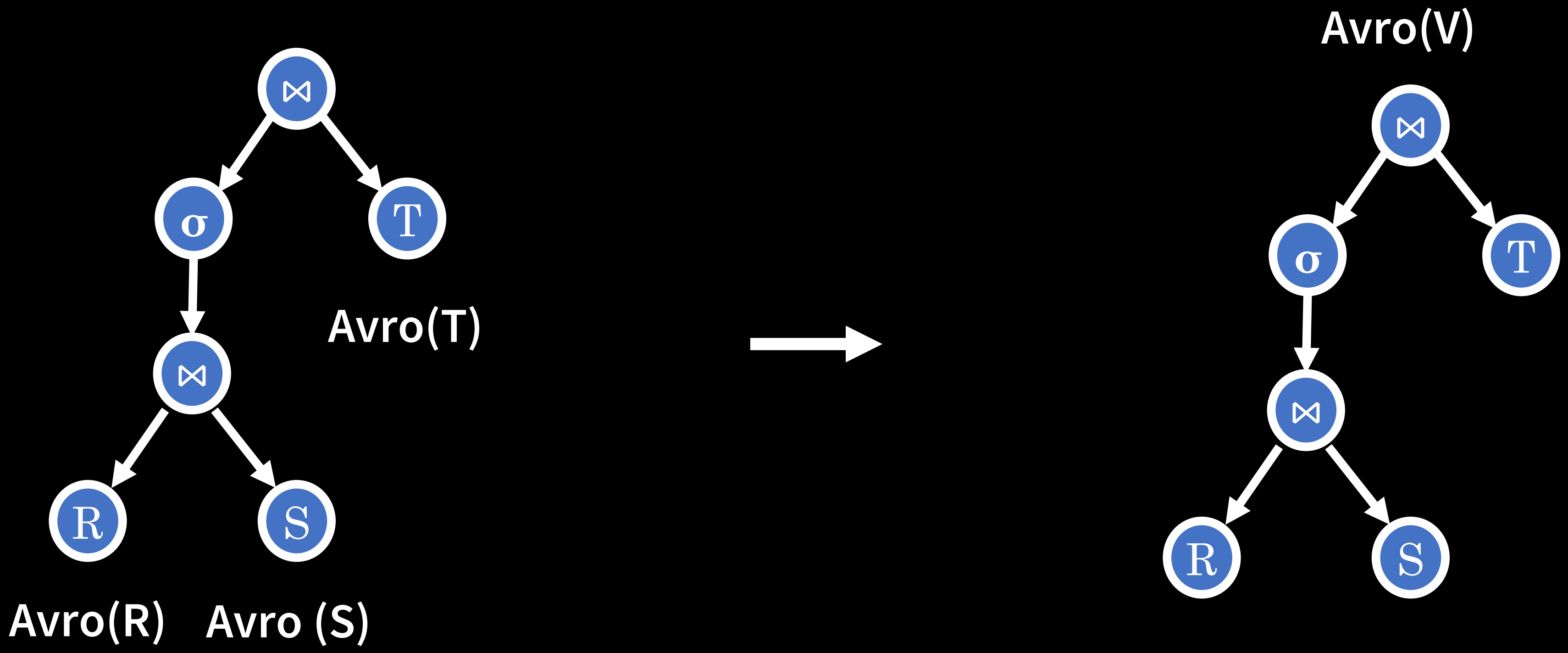
Data Governance



PII Lineage and Derivation



Format-specific View Schema Derivation



Transport UDFs: *Translatable Portable* UDFs

github.com/linkedin/transport

IR

- SQL has pretty well-understood IR: Relational Algebra
- Standard Operators
  - Scan, Filter, Project, Join, Group By, etc
- UDFs
  - Opaque
  - Use imperative language
  - Not portable or translatable

UDF Denormalization

Duplication

Multiple versions of the same UDF. Not clear which is the source of truth.

Inconsistency

Duplicate implementations can diverge causing data inconsistency

Low Productivity

Developers need to learn multiple APIs, implement same logic multiple times.

Low Performance

In some cases, use tuple conversion adapters to enable portability.

UDF APIs

- API Complexity
  - APIs expose low-level details of engines
  - Data types may not intuitvely map to SQL type-system
- API Disparity
  - APIs differ in what to expect from developer
  - APIs differ in features they can provide

Transport UDFs

```java
public class MapFromTwoArrays
    extends StdUDF2<StdArray, StdArray, StdMap> {

  @Override
  public List<String> getInputParameterSignatures() {
    return ImmutableList.of(
        "array(K)",
        "array(V)"
    );
  }

  @Override
  public String getOutputParameterSignature() {
    return "map(K,V)";
  }

  @Override
  public StdMap eval(StdArray a1, StdArray a2) {
    StdMap map = getStdFactory().createMap(
        getOutputParameterSignature());
    for (int i = 0; i < a1.size(); i++) {
      map.put(a1.get(i), a2.get(i));
    }
    return map;
  }
}
```

Transport UDFs

```java
public class MapFromTwoArrays
    extends StdUDF2<StdArray, StdArray, StdMap> {

  @Override
  public List<String> getInputParameterSignatures() {
    return ImmutableList.of(
        "array(K)",
        "array(V)"
    );
  }

  @Override
  public String getOutputParameterSignature() {
    return "map(K,V)";
  }

  @Override
  public StdMap eval(StdArray a1, StdArray a2) {
    StdMap map = getStdFactory().createMap(
        getOutputParameterSignature());
    for (int i = 0; i < a1.size(); i++) {
      map.put(a1.get(i), a2.get(i));
    }
    return map;
  }
}
```

Transport UDFs

```java
public class MapFromTwoArrays
    extends StdUDF2<StdArray, StdArray, StdMap> {

  @Override
  public List<String> getInputParameterSignatures() {
    return ImmutableList.of(
        "array(K)",
        "array(V)"
    );
  }

  @Override
  public String getOutputParameterSignature() {
    return "map(K,V)";
  }

  @Override
  public StdMap eval(StdArray a1, StdArray a2) {
    StdMap map = getStdFactory().createMap(
        getOutputParameterSignature());
    for (int i = 0; i < a1.size(); i++) {
      map.put(a1.get(i), a2.get(i));
    }
    return map;
  }
}
```

Transport UDFs

```
public class MapFromTwoArrays
    extends StdUDF2<StdArray, StdArray, StdMap> {

  @Override
  public List<String> getInputParameterSignatures() {
    return ImmutableList.of(
        "array(K)",
        "array(V)"
    );
  }

  @Override
  public String getOutputParameterSignature() {
    return "map(K,V)";
  }

  @Override
  public StdMap eval(StdArray a1, StdArray a2) {
    StdMap map = getStdFactory().createMap(
        getOutputParameterSignature());
    for (int i = 0; i < a1.size(); i++) {
      map.put(a1.get(i), a2.get(i));
    }
    return map;
  }
}
```

Transport UDFs

```java
public class MapFromTwoArrays
    extends StdUDF2<StdArray, StdArray, StdMap> {

  @Override
  public List<String> getInputParameterSignatures() {
    return ImmutableList.of(
        "array(K)",
        "array(V)"
    );
  }

  @Override
  public String getOutputParameterSignature() {
    return "map(K,V)";
  }

  @Override
  public StdMap eval(StdArray a1, StdArray a2) {
    StdMap map = getStdFactory().createMap(
        getOutputParameterSignature());
    for (int i = 0; i < a1.size(); i++) {
      map.put(a1.get(i), a2.get(i));
    }
    return map;
  }
}
```

Transport UDFs

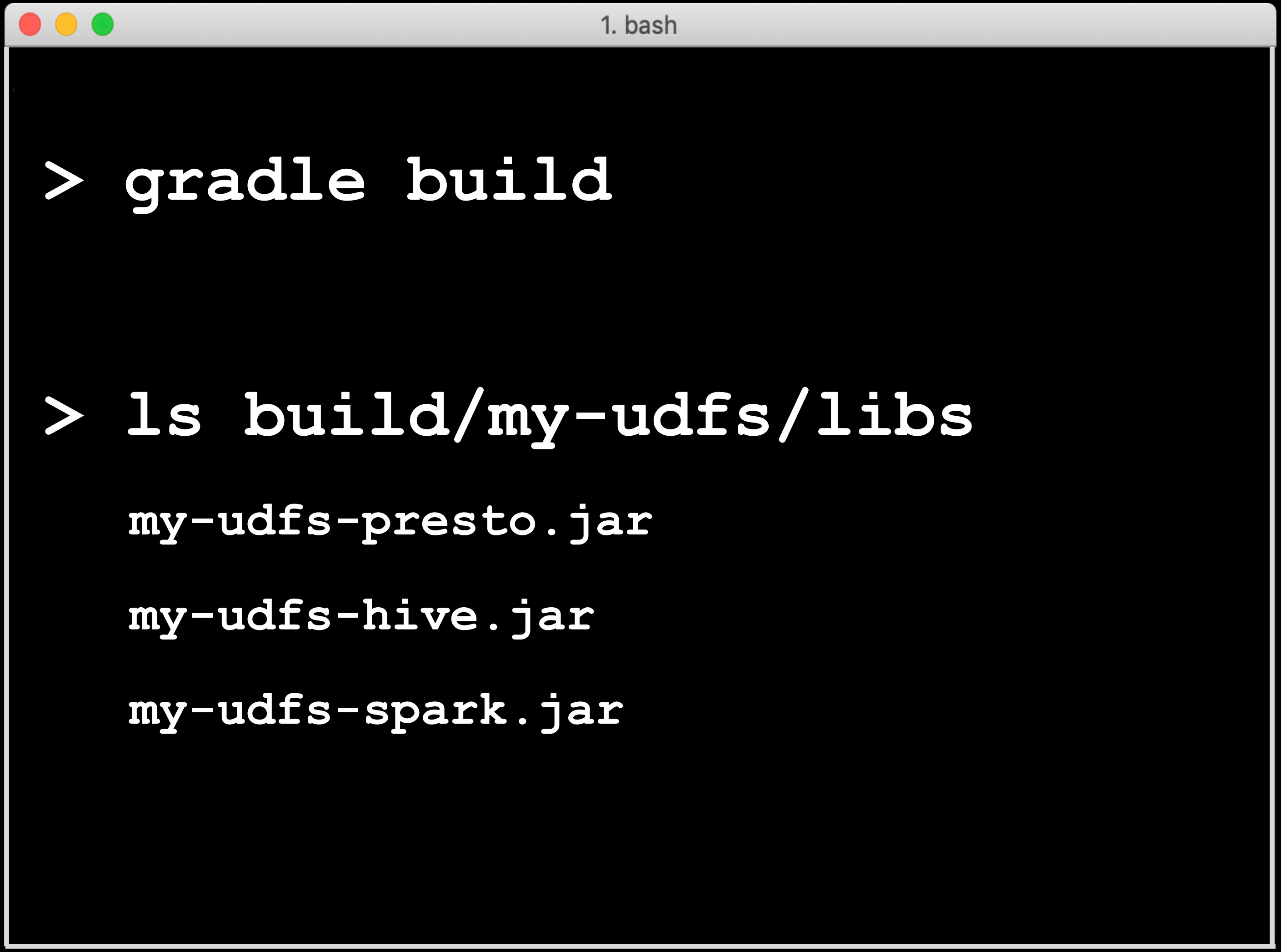
```
public class MapFromTwoArrays
    extends StdUDF2<StdArray, StdArray, StdMap> {

  @Override
  public List<String> getInputParameterSignatures() {
    return ImmutableList.of(
        "array(K)",
        "array(V)"
    );
  }

  @Override
  public String getOutputParameterSignature() {
    return "map(K,V)";
  }

  @Override
  public StdMap eval(StdArray a1, StdArray a2) {
    StdMap map = getStdFactory().createMap(
        getOutputParameterSignature());
    for (int i = 0; i < a1.size(); i++) {
      map.put(a1.get(i), a2.get(i));
    }
    return map;
  }
}
```

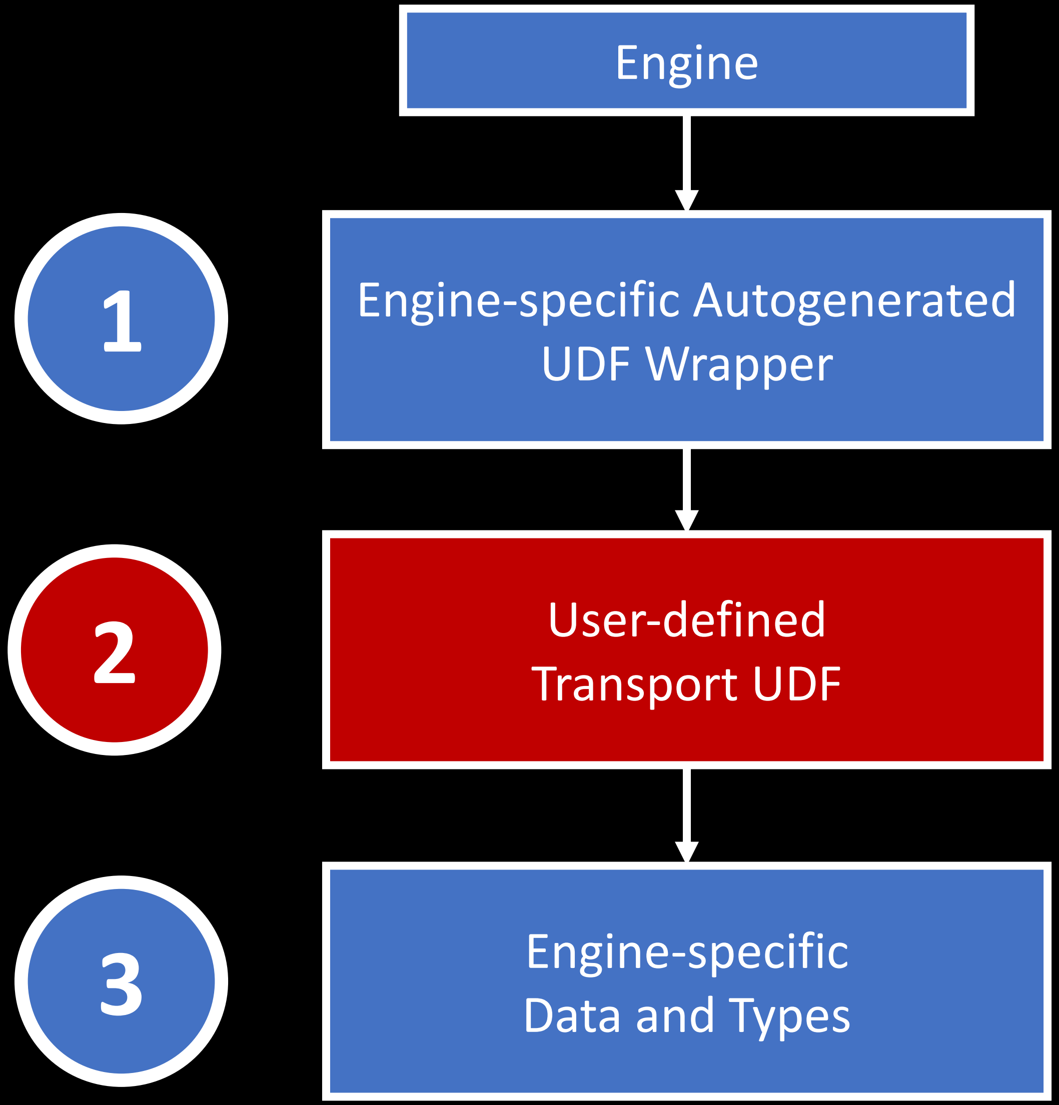
Then What?



```
> gradle build


> ls build/my-udfs/libs
  my-udfs-presto.jar
  my-udfs-hive.jar
  my-udfs-spark.jar
```

Architecture


Engine
1
Engine-specific Autogenerated UDF Wrapper
2
User-defined Transport UDF
3
Engine-specific Data and Types

Contributors