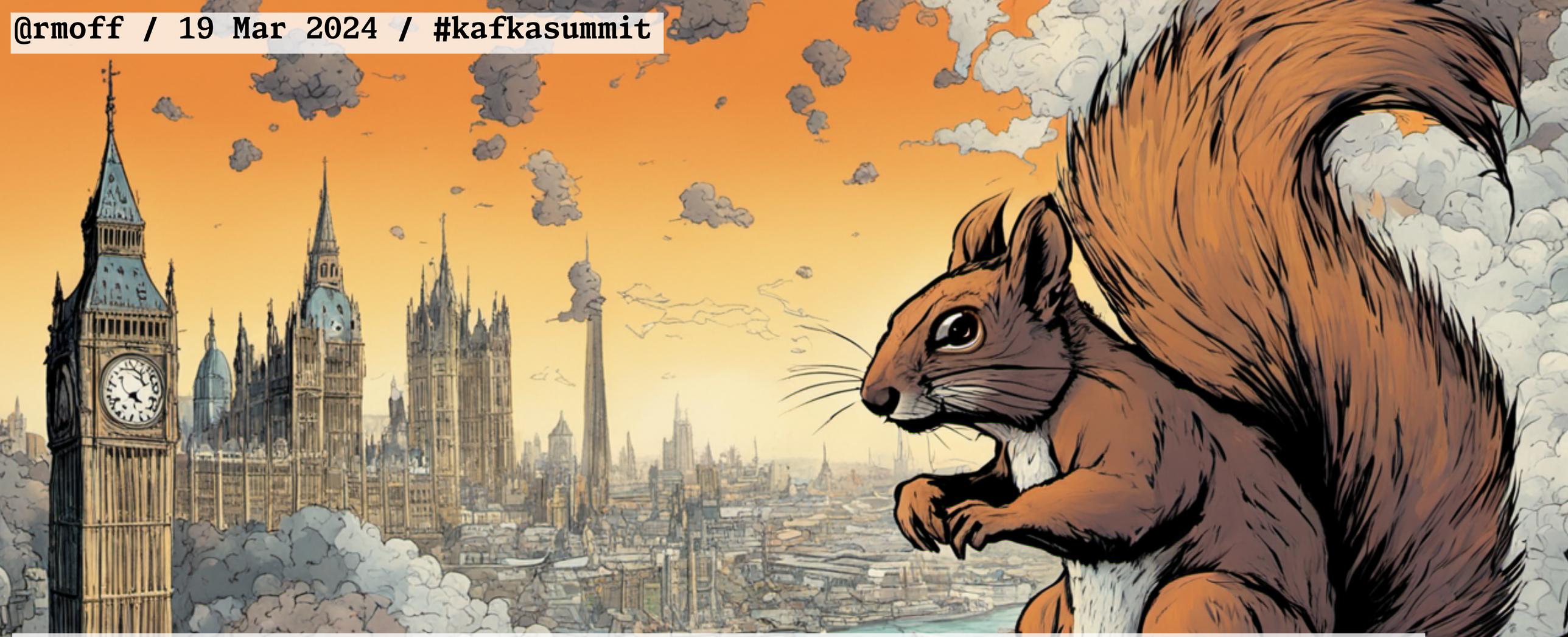


Here be ~~Dragons~~ Stacktraces

Flink SQL for Non-Java Developers

Robin Moffatt, Principal DevEx Engineer @ Decodable

Actual footage of a SQL Developer looking at Apache Flink for the first time

```
$ mvn archetype:generate                               \
  -DarchetypeGroupId=org.apache.flink                  \
  -DarchetypeArtifactId=flink-quickstart-java          \
  -DarchetypeVersion=1.20-SNAPSHOT
```

```
<modelVersion>4.0.0</modelVersion>
<groupId>org.example</groupId>
<artifactId>myProject</artifactId>
<version>1.0-SNAPSHOT</version>

<dependencies>
  <!-- other project dependencies ...-->
  <dependency>
    <groupId>org.apache.flink</groupId>
    <artifactId>flink-sql-connector-hive-3.1.3_2.12</artifactI
    <version>1.20-SNAPSHOT</version>
  </dependency>

  <dependency>
    <groupId>org.apache.flink</groupId>
    <artifactId>flink-parquet_2.12</artifactId>
    <version>1.20-SNAPSHOT</version>
  </dependency>

</dependencies>
```

```
EnvironmentSettings settings = EnvironmentSettings.inStreamingMode()
TableEnvironment tableEnv = TableEnvironment.create(settings);

String name            = "myhive";
String defaultDatabase = "mydatabase";
String hiveConfDir     = "/opt/hive-conf";

Hi
ta

//
ta
```

> Since Flink v1.16, TableEnvironment introduces a user class loader to have a consistent class loading be
> programs, SQL Client and SQL Gateway. The user classloader manages all user jars such as jar added by `ADD JAR`
> `FUNCTION .. USING JAR ..` statements. User-defined catalogs should replace
> `Thread.currentThread().getContextClassLoader()` with the user class loader to load classes. Otherwise,
> `ClassNotFoundException` maybe thrown. The user class loader can be accessed via
> `CatalogFactory.Context#getClassLoader`.

What Is Apache Flink?

A Brief History of Flink

or


APACHE FLINK
Why are you the way that you are?
imgflip.com

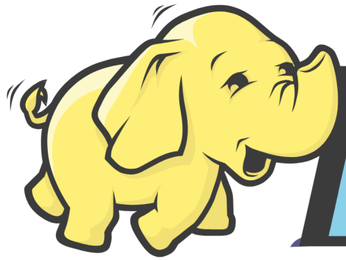
Back in the time of ~~dinosaurs~~ Hadoop

APACHE ZooKeeper™

Apache Pig

- Started life as a research project in 2011, called **Stratosphere**.
- This was the time of MapReduce. Java and Scala were the only way to do this.

OOZIE



APACHE hadoop™

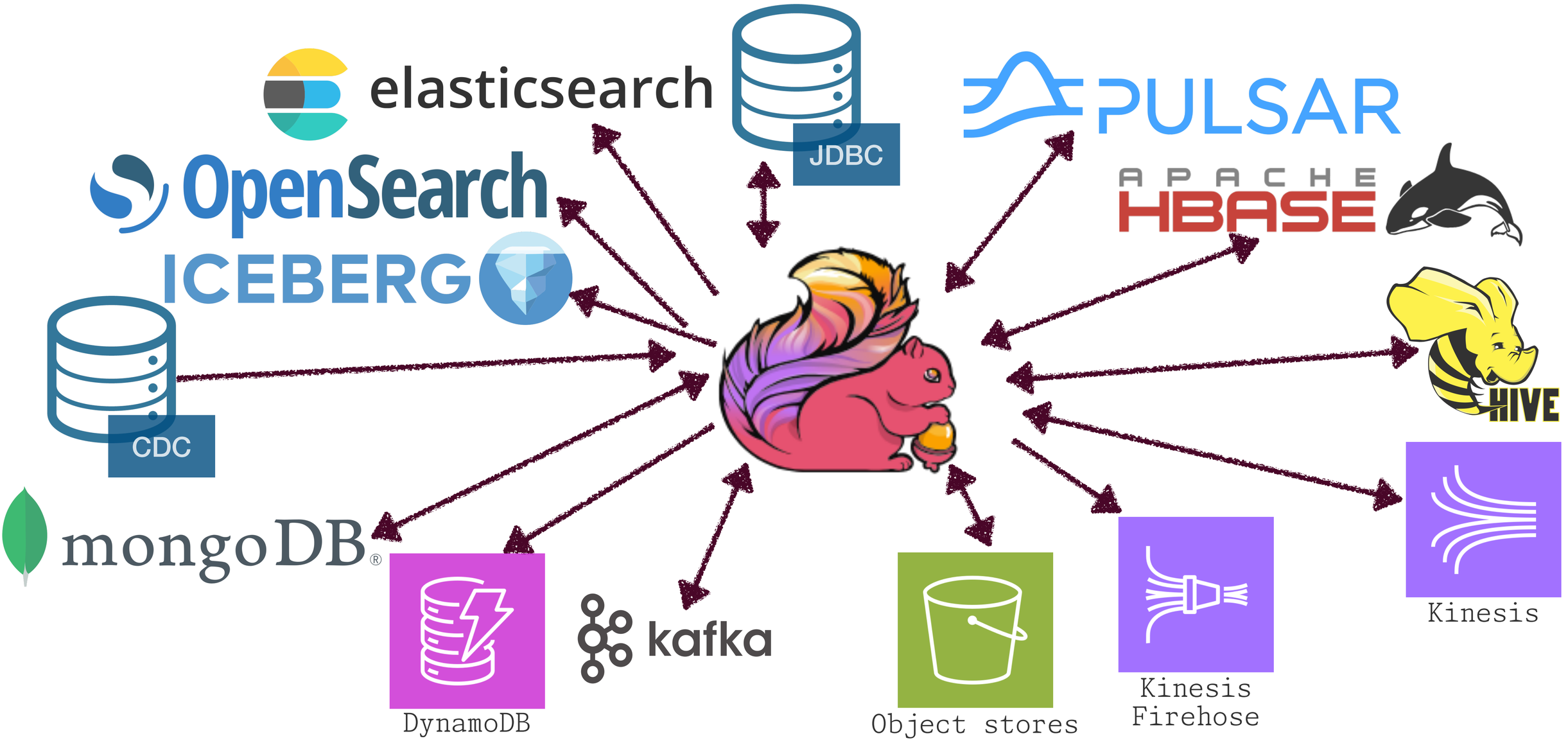
APACHE HBASE

Flink is a big project

- Flink
- Stateful Functions
- ML
- Kubernetes Operator
- CDC Connector
- Paimon (incubating)

Capabilities

Connect to Lots of Source and Target Systems



Stateful and Stateless computations

- Filtering

```
SELECT * FROM myStream WHERE foo=42
```

- Joining

```
SELECT a.*, b.* FROM myStream a
       INNER JOIN myLookup b ON a.id=b.foo_id
```

- Transforming

```
SELECT cost * tax_rate AS total_cost
  FROM myStream
```

- Aggregations

```
SELECT SUM(order_value) AS total_order_values
  FROM orders
```

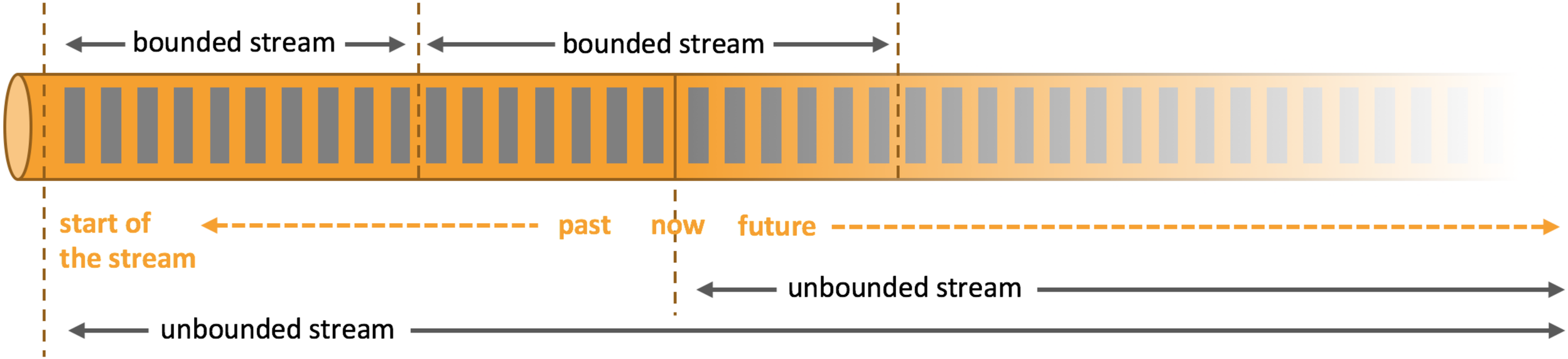
- Pattern matching

```
SELECT *
  FROM myStream
    MATCH_RECOGNIZE (
         PARTITION BY id
         ORDER BY user_action_time
         […]
```

- …and a whole lot more

Batch *and* Streaming

Bounded *and* Unbounded



How Does Flink Work?

<< magic   >>

Flink Client

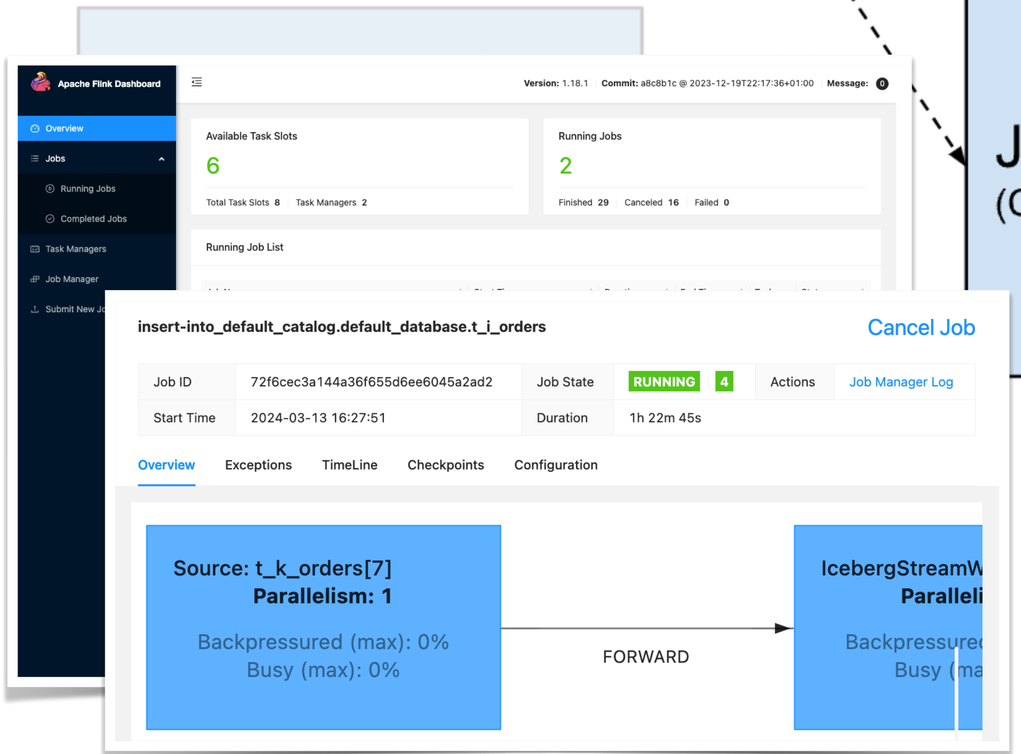
Job Submission

JobManager
(Central Work Coordination)

TaskManager
(WorkerNodes)

TaskManager
(WorkerNodes)

TaskManager
(WorkerNodes)



https://nightlies.apache.org/flink/flink-docs-master/docs/deployment/overview/

Flink Client

Jo

Flink's SQL Engine: Let's Open the Engine Room!

🗣️ Timo Walther

📆 Tuesday

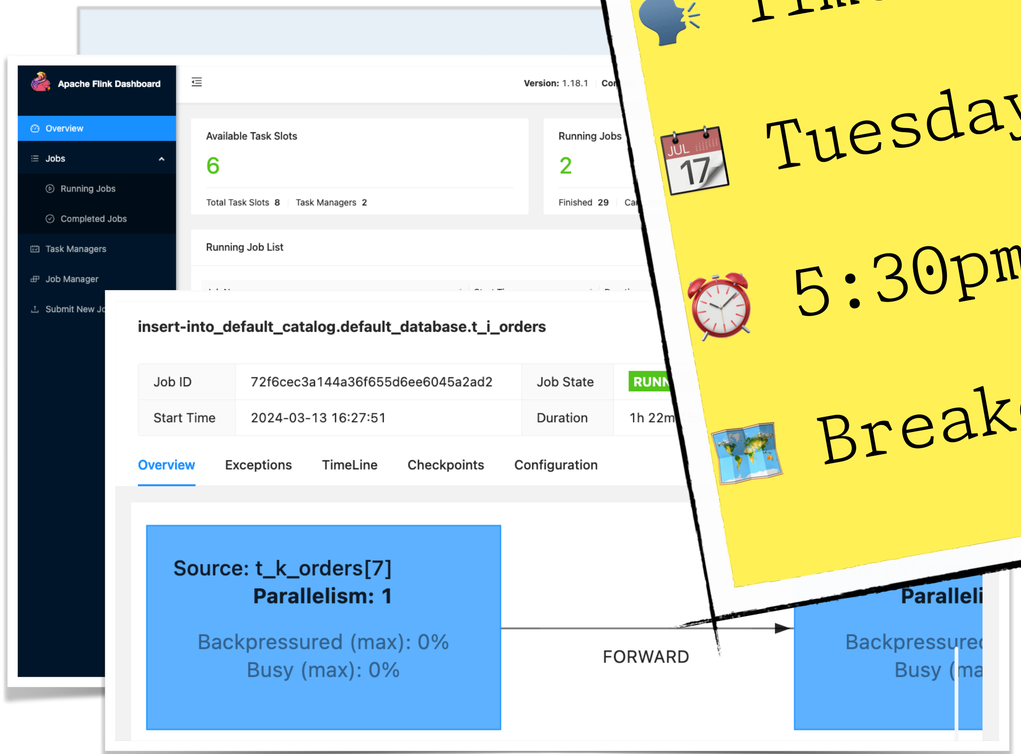
⏰ 5:30pm

🗺️ Breakout Room 2

TaskManager
(WorkerNodes)

askManager
orkerNodes)

TaskManager
(WorkerNodes)



https://nightlies.apache.org/flink/flink-docs-master/docs/deployment/overview/

Running Flink

Works on my machine…

```
$ ./bin/start-cluster.sh
Starting cluster.
Starting standalonesession daemon on host asgard08.
Starting taskexecutor daemon on host asgard08.
```

```
$ jps -l
14656 org.apache.flink.runtime.taskexecutor.TaskManagerRunner
14379 org.apache.flink.runtime.entrypoint.StandaloneSessionClusterEntrypoint
```

Using Flink

It's not just Java

- **PyFlink**
  - added in 1.9.0 in 2019
- **Flink SQL**
  - Added in 1.5.0 in 2018

Flink SQL

SQL Language Support

- Built on **Apache Calcite**
- **Common Table Expression** (CTE) (WITH)
- **Set-based** operations
- **Joins**
- **Aggregations**
- And lots more…

APACHE calcite™

Running Flink SQL

- SQL Client
- SQL Gateway
  - REST API
  - Hive
  - JDBC Driver
- From Java or Python

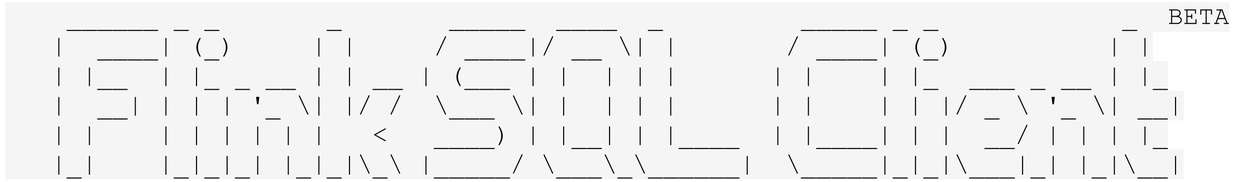
Flink SQL Client

```
$ ./bin/sql-client.sh
```

```
        Welcome! Enter 'HELP;' to list
all available commands. 'QUIT;' to
exit.

Command history file path: /opt/
flink/.flink-sql-history

Flink SQL>
```



DEMO

https://github.com/decodableco/examples/kafka-iceberg

A Few Useful Settings

Runtime Mode

```
SET 'execution.runtime-mode' = 'streaming';
```

- streaming [default]
- batch

Result Mode

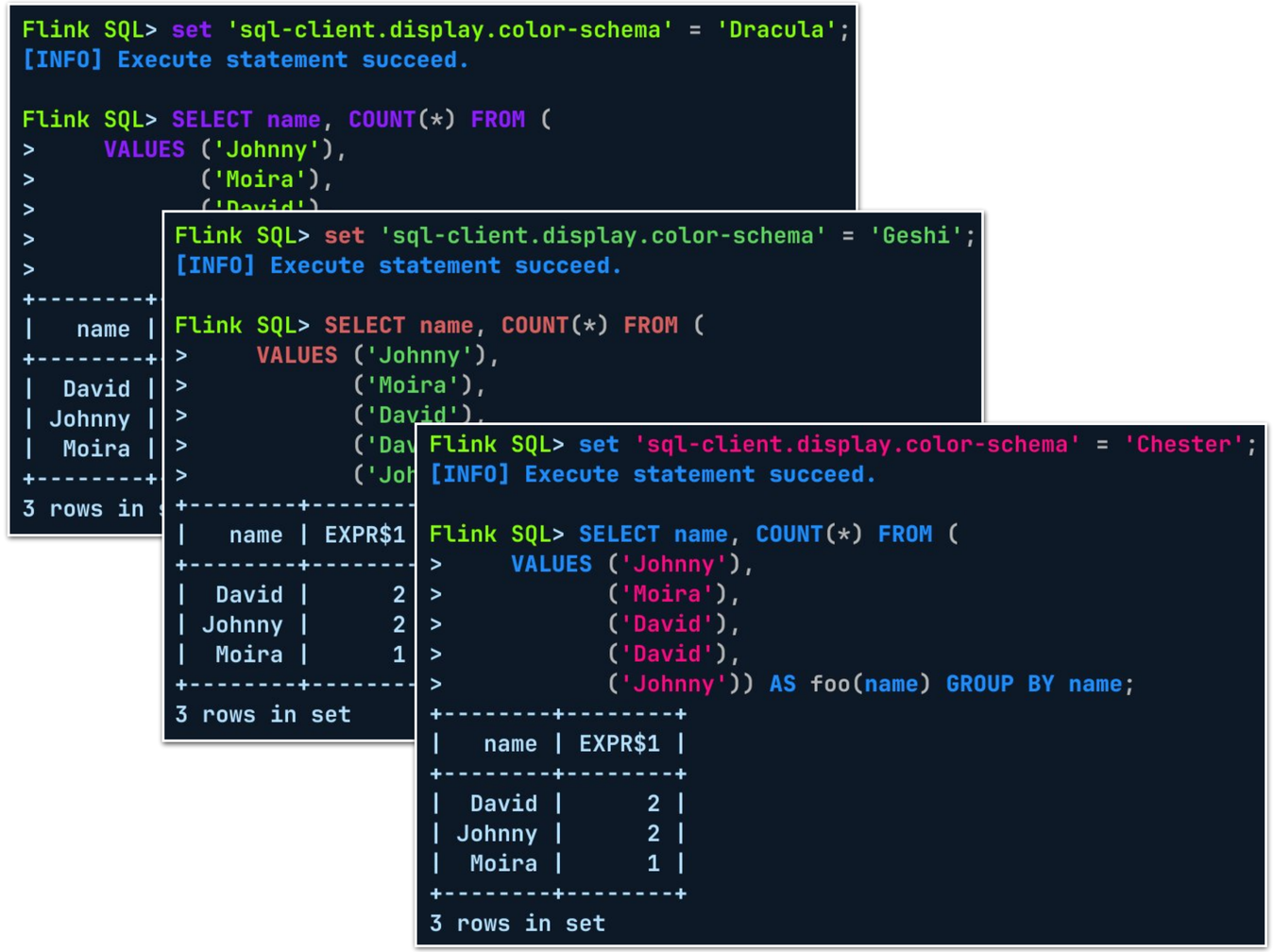
```
SET 'sql-client.execution.result-mode' = 'table';
```

- table [default]
- changelog
- tableau

Colour Scheme

SET 'sql-client.display.color-schema' = 'Chester';

- Because why not?!



Changing the defaults

Setting up a SQL Client initialisation file

- Create a SQL file:

```
$ cat init.sql
SET 'execution.runtime-mode' = 'batch';
SET 'sql-client.execution.result-mode' = 'tableau';
```

- Launch SQL Client with the `-i` flag and pass the file as a parameter:

```
./bin/sql-client.sh -i init.sql
```

Submitting SQL as a job

- SQL Client

```
$ ./bin/sql-client.sh --file ~/my_query.sql
```

- SQL Gateway

```
curl --location 'localhost:8083/sessions/42/statements' \
--header 'Content-Type: application/json' \
--header 'Accept: application/json' \
--data '{
  "statement": "SELECT * FROM foo;"
}'
```

- Application mode support: FLIP-316: Support application mode for SQL Gateway

Some of the Gnarly Stuff

The Joy of JARs

- For each connector, format, and catalog you need to install dependencies.
- All of these are available as JARs (Java ARchive)

This might jar a bit…

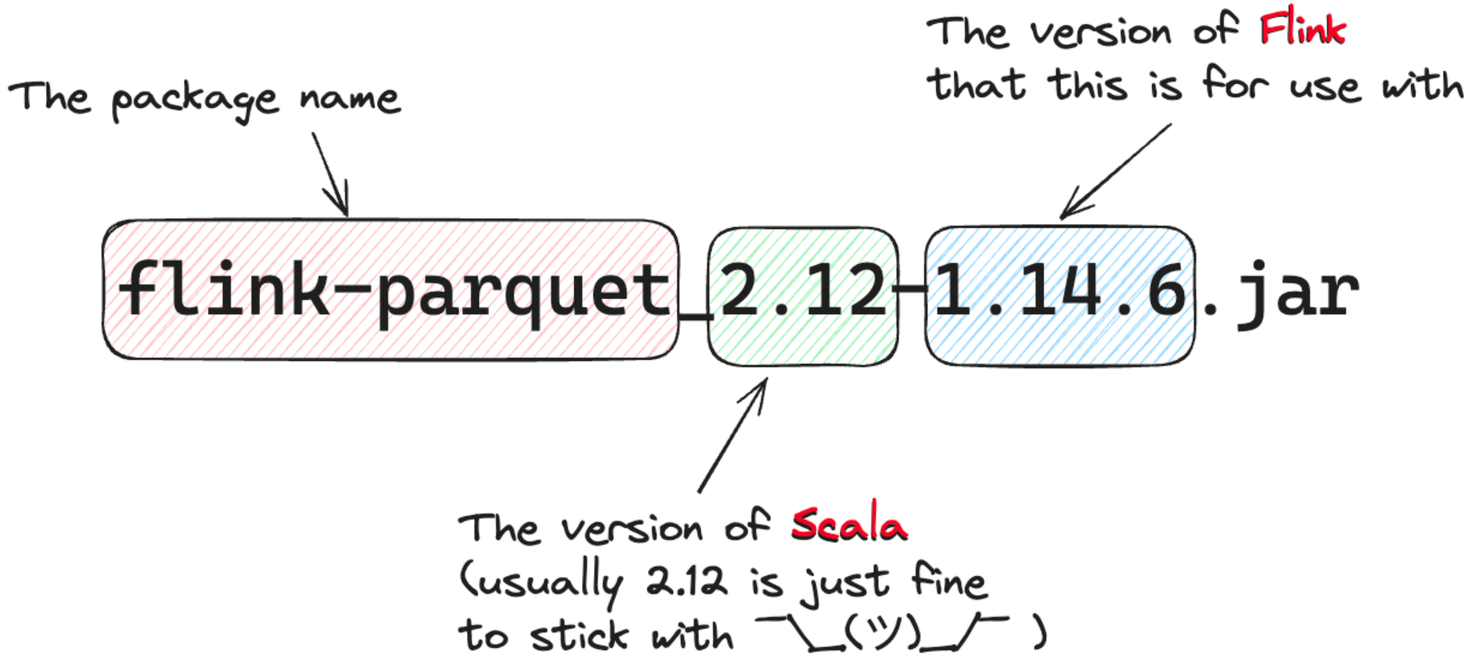
```
Could not execute SQL statement.
Reason: java.lang.ClassNotFoundException
```

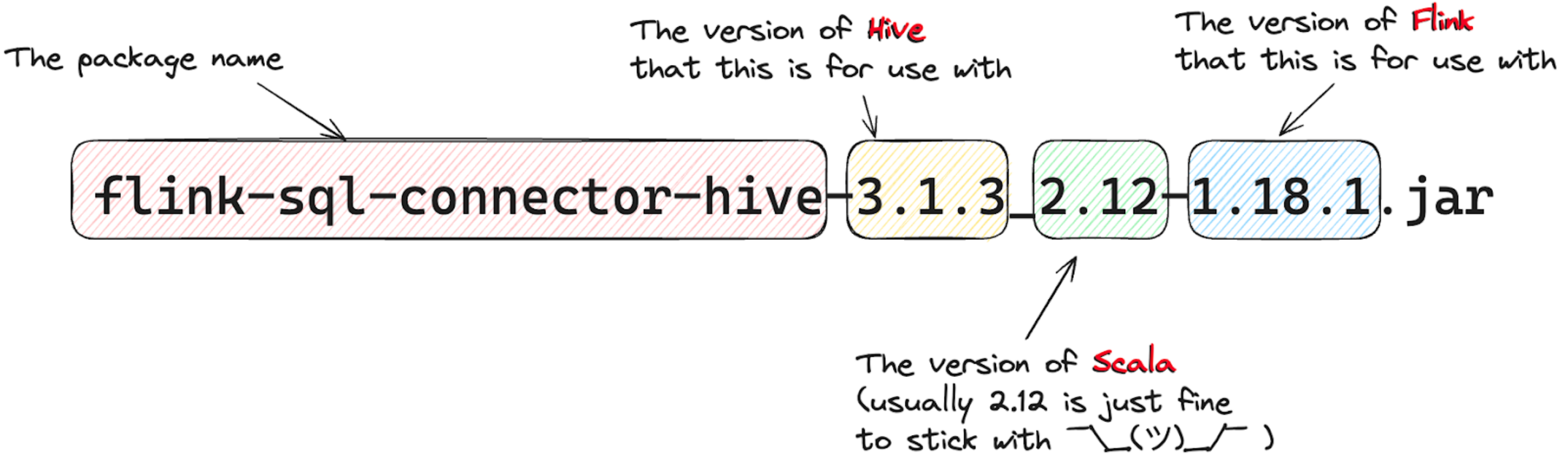
```
org.apache.flink.core.fs.UnsupportedFileSystemSchemeException: Could not find a file system implementation for scheme 's3'
```

```
Could not find any factory for identifier 'hive' that implements 'org.apache.flink.table.factories.CatalogFactory' in the classpath.
```

Finding JARs

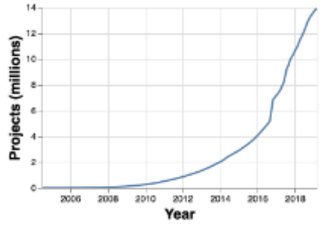
- Usually the docs will tell you which JAR you need.
- JARs are very specific to the versions of the tools that you're using.




The package name
The version of Hive that this is for use with
The version of Flink that this is for use with
flink-sql-connector-hive-3.1.3_2.12-1.18.1.jar
The version of Scala (usually 2.12 is just fine to stick with ¯_(ツ)_/¯)

MVN REPOSITORY Search for groups, artifacts, categ Search Categories | Popular | Contact Us

Indexed Artifacts (38.1M)



Popular Categories

- Testing Frameworks & Tools
- Android Packages
- Logging Frameworks
- Java Specifications
- JSON Libraries
- JVM Languages
- Language Runtime
- Core Utilities
- Mocking
- Web Assets
- Annotation Libraries
- HTTP Clients
- Logging Bridges
- Dependency Injection
- XML Processing
- Web Frameworks

Home » org.apache.flink » flink-parquet » 1.18.1

Flink : Formats : Parquet » 1.18.1

Flink : Formats : Parquet

License	Apache 2.0
Tags	parquet flink serialization apache column
Date	Jan 16, 2024
Files	pom (17 KB) jar (174 KB) View All
Repositories	Central
Ranking	**#9617 in MvnRepository (See Top Artifacts)**
Used By	**42 artifacts**
Vulnerabilities	**Vulnerabilities from dependencies:** CVE-2023-2976 CVE-2022-26612 CVE-2020-8908

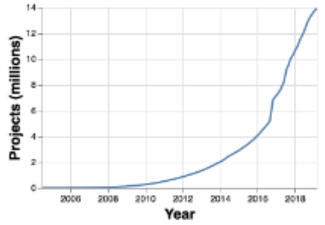
Maven | Gradle | Gradle (Short) | Gradle (Kotlin) | SBT | Ivy | Grape | Leiningen | Buildr

```
<!-- https://mvnrepository.com/artifact/org.apache.flink/flink-parquet
-->
<dependency>
    <groupId>org.apache.flink</groupId>
    <artifactId>flink-parquet</artifactId>
    <version>1.18.1</version>
    <scope>provided</scope>
```

☑ Include comment with link to declaration

MVN REPOSITORY | Search for groups, artifacts, categ | Search | Categories | Popular | Contact Us

Indexed Artifacts (38.1M)



Popular Categories

- Testing Frameworks & Tools
- Android Packages
- Logging Frameworks
- Java Specifications
- JSON Libraries
- JVM Languages
- Language Runtime
- Core Utilities
- Mocking
- Web Assets
- Annotation Libraries
- HTTP Clients
- Logging Bridges
- Dependency Injection
- XML Processing
- Web Frameworks

Home » org.apache.flink » flink-parquet » 1.18.1

Flink : Formats : Parquet » 1.18.1

Flink : Formats : Parquet

License	Apache 2.0
Tags	parquet flink serializatio apache column
Date	Jan 16, 2024
Files	pom (17 KB) jar 194 KB) View All
Repositories	Central
Ranking	#9617 in MvnRepository (See Top Artifacts)
Used By	42 artifacts
Vulnerabilities	Vulnerabilities from dependencies: CVE-2023-2976 CVE-2022-26612 CVE-2020-8908

Maven | Gradle | Gradle (Short) | Gradle (Kotlin) | SBT | Ivy | Grape | Leiningen | Buildr

```
<!-- https://mvnrepository.com/artifact/org.apache.flink/flink-parquet
-->
<dependency>
    <groupId>org.apache.flink</groupId>
    <artifactId>flink-parquet</artifactId>
    <version>1.18.1</version>
    <scope>provided</scope>
```

☑ Include comment with link to declaration

```
$ tree /opt/flink/lib
├── aws
│   ├── aws-java-sdk-bundle-1.12.648.jar
│   └── hadoop-aws-3.3.4.jar
├── flink-cep-1.18.1.jar
├── flink-parquet_2.12-1.18.1.jar
├── flink-table-runtime-1.18.1.jar
├── hadoop
│   ├── commons-configuration2-2.1.1.jar
│   ├── commons-logging-1.1.3.jar
│   ├── hadoop-auth-3.3.4.jar
├── hive
│   └── flink-sql-connector-hive-3.1.3_2.12-1.18.1.jar
├── iceberg
│   └── iceberg-flink-runtime-1.18-1.5.0.jar
├── kafka
│   └── flink-sql-connector-kafka-3.1.0-1.18.jar
```

Don't forget to restart!

Tables, Connectors, and Catalogs

Tables

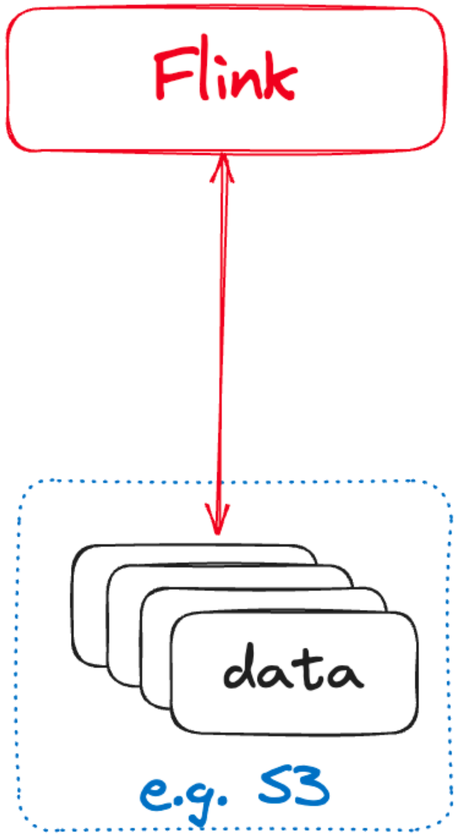
```
CREATE TABLE t_k_orders
  (
    orderid             STRING,
    customerid          STRING,
    ordernumber         INT,
    product             STRING,
    discountpercent     INT
  ) WITH (
    'connector'                         = 'kafka',
    'topic'                             = 'orders',
    'properties.bootstrap.servers'      = 'broker:29092',
    'scan.startup.mode'                 = 'earliest-offset',
    'format'                            = 'json'
  );
```

This used to be simple

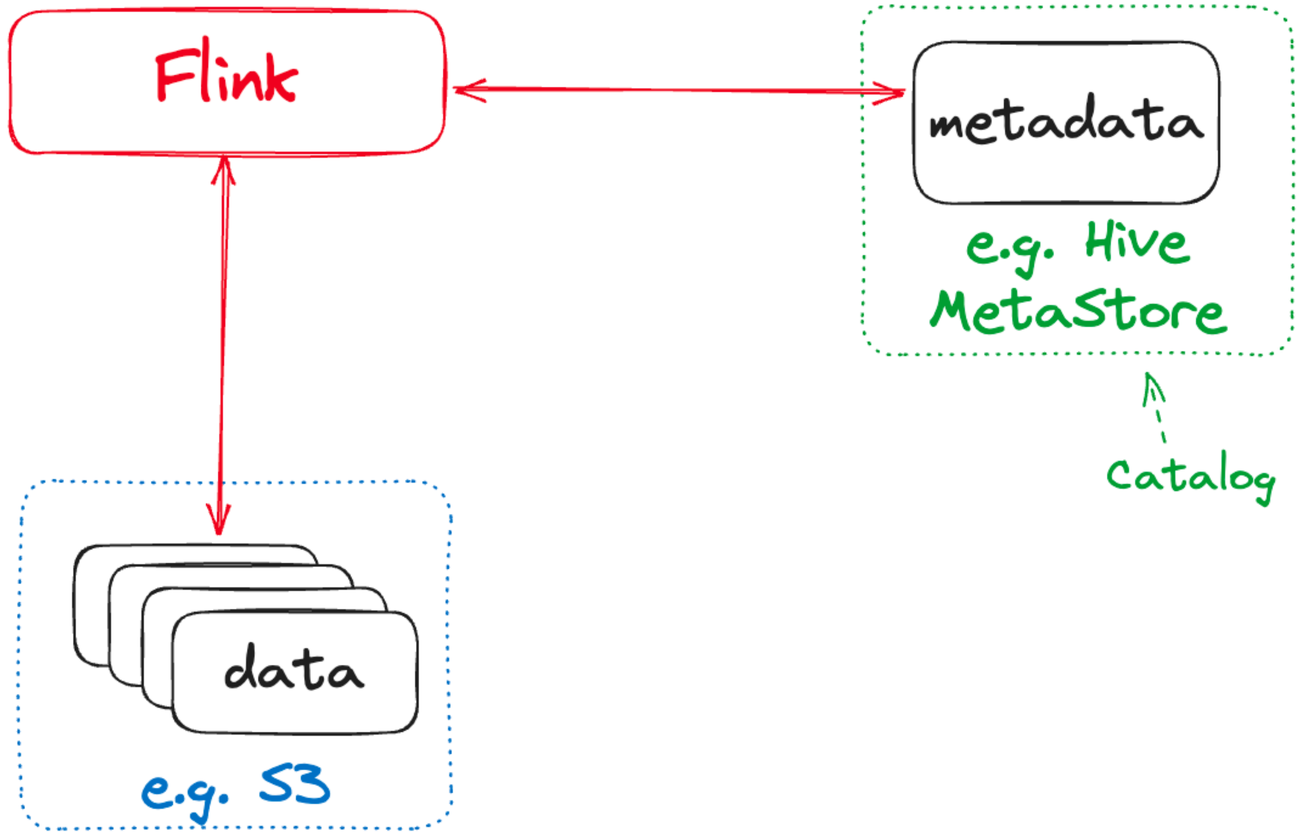
- The data and information about the data was all stored in the database
  - Information Schema
  - System Catalog
  - Data Dictionary Views

Now it's not so simple

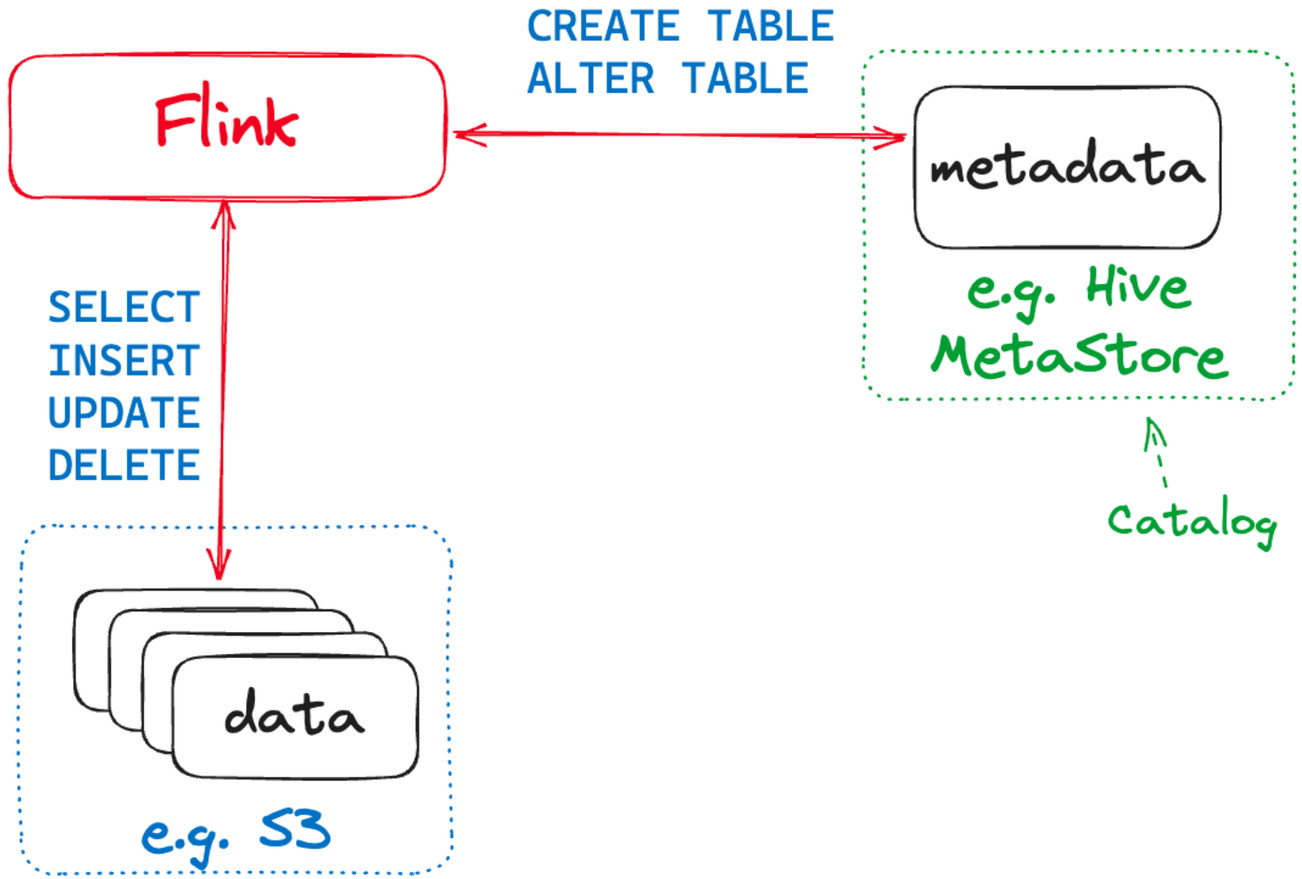
Flink Catalogs



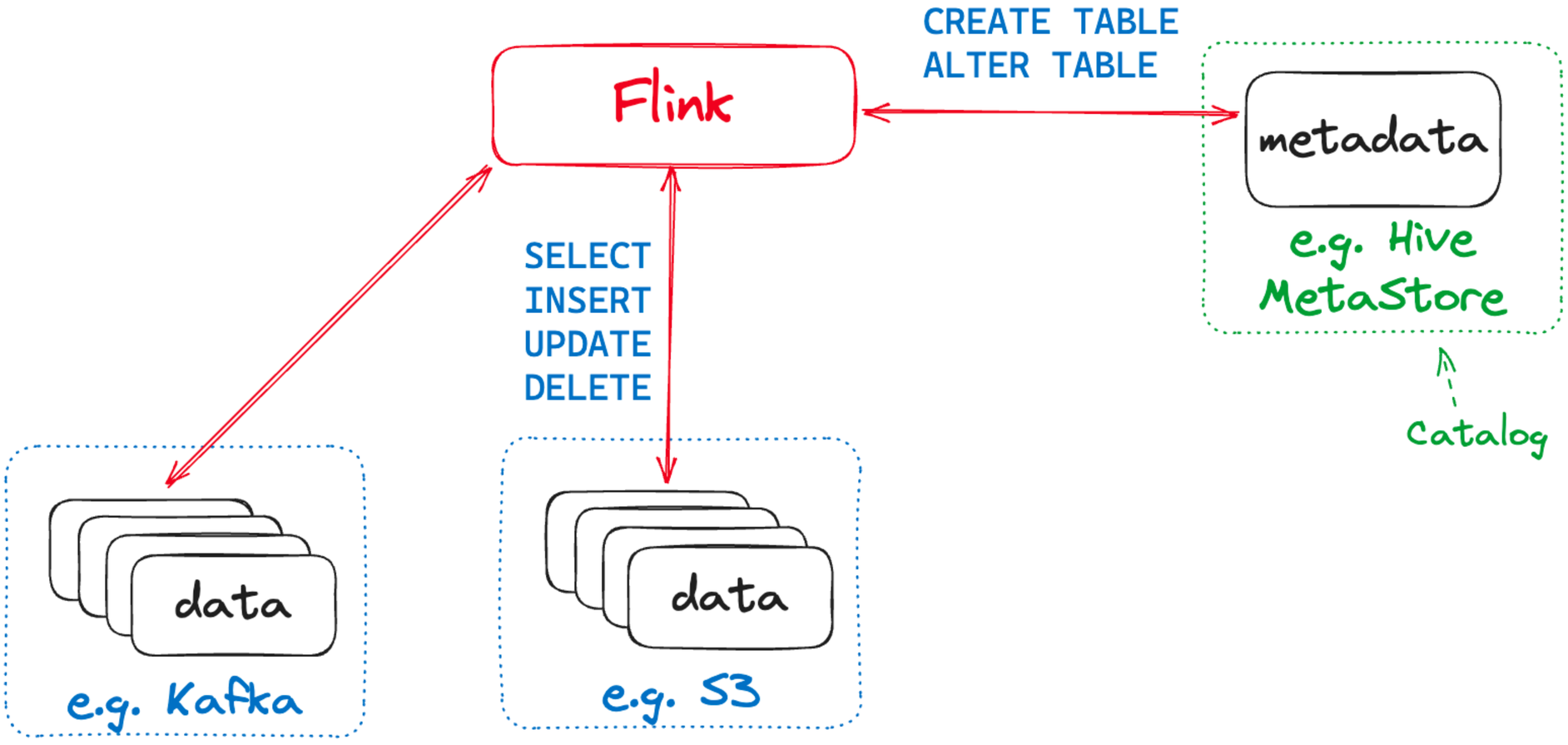
Flink Catalogs



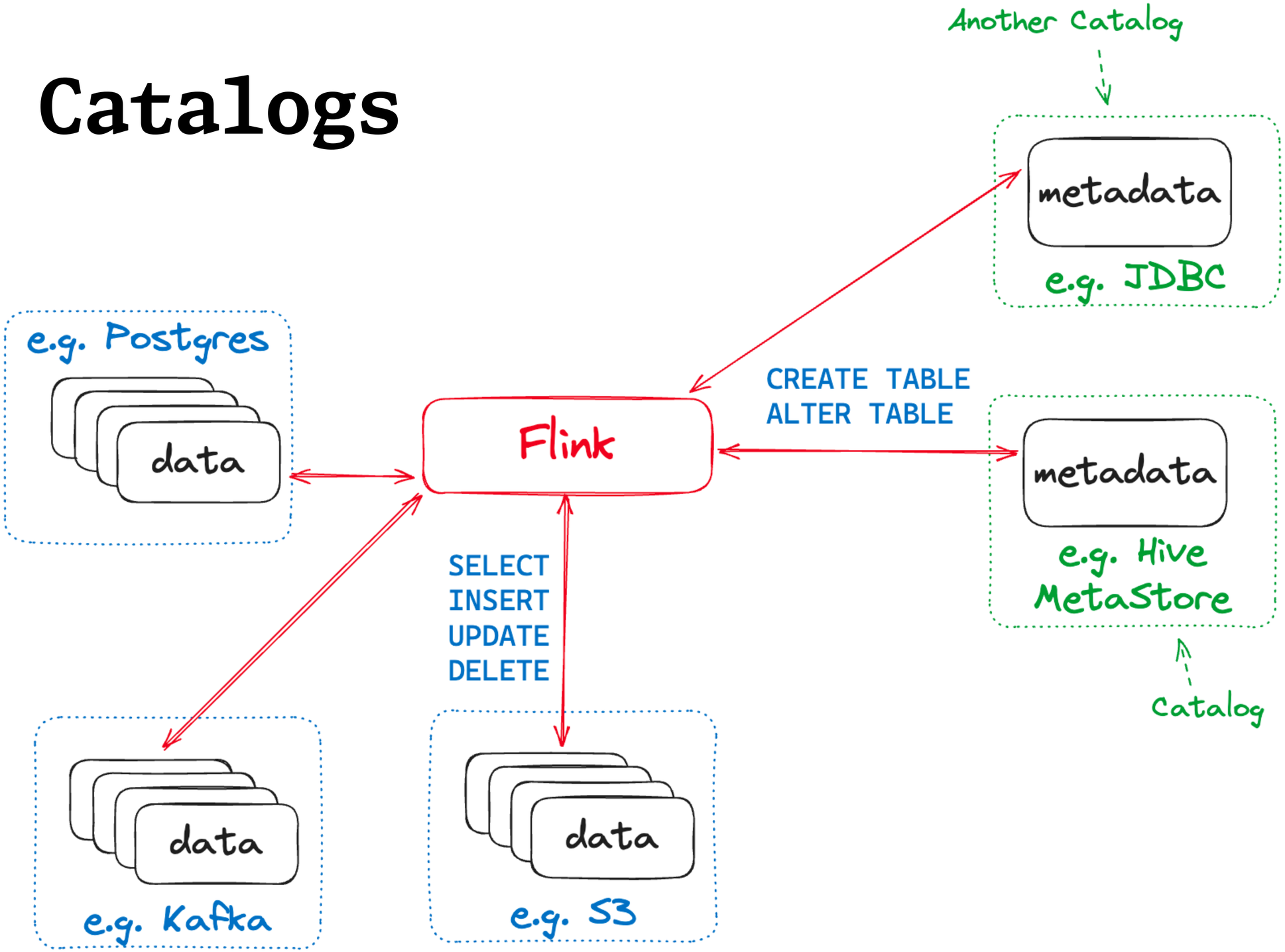
Flink Catalogs



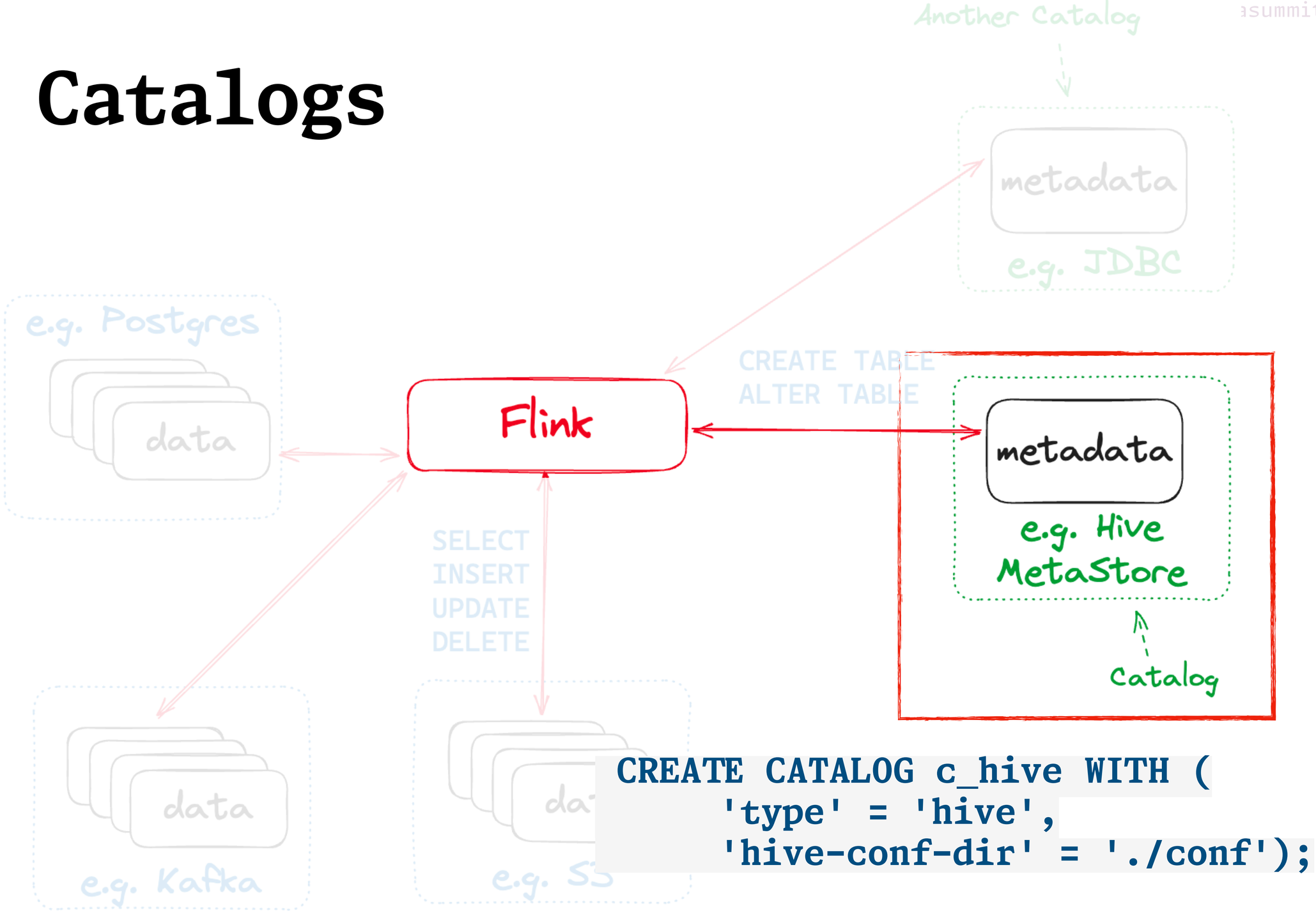
Flink Catalogs



Flink Catalogs

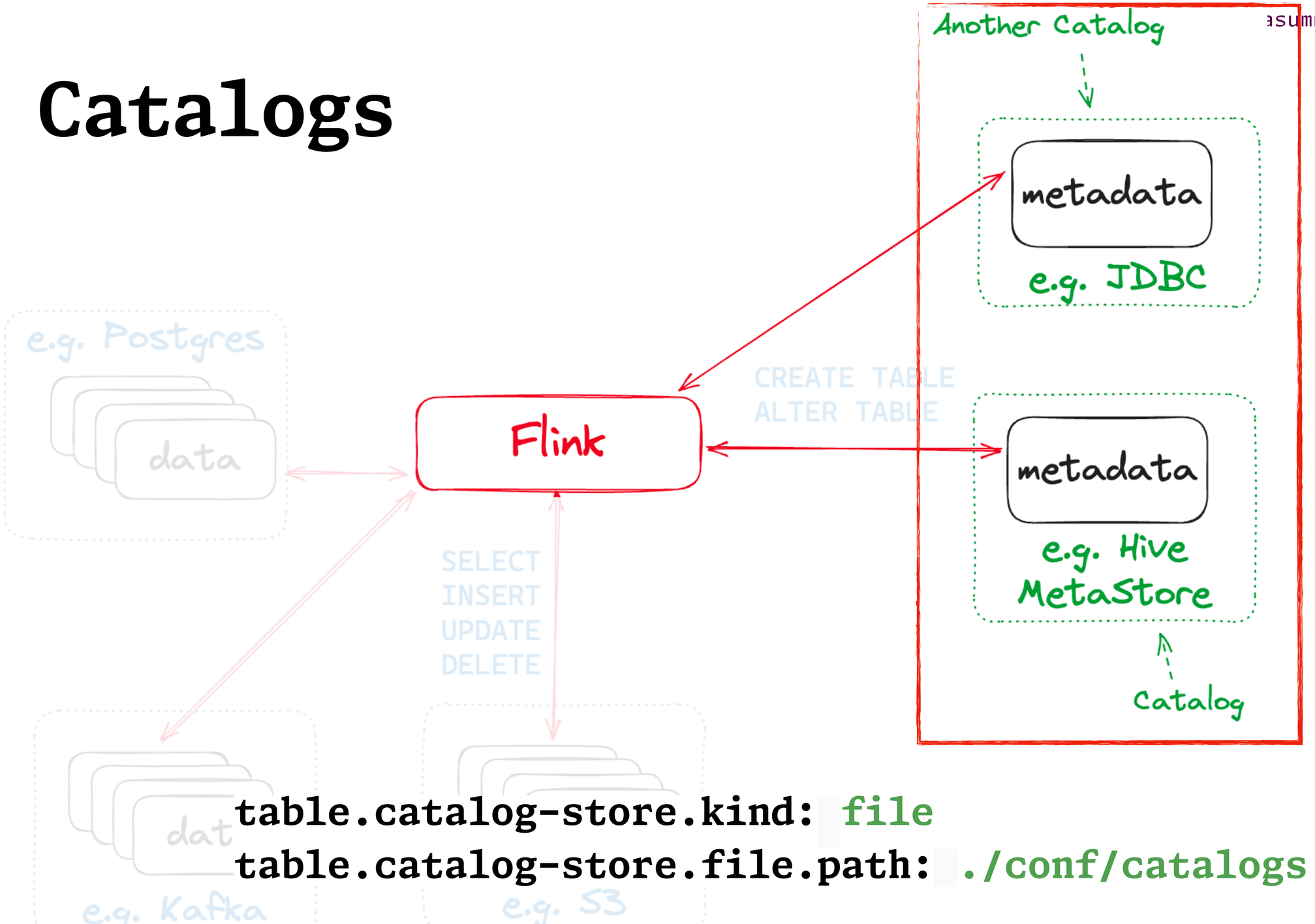


Flink Catalogs




```
CREATE CATALOG c_hive WITH (
      'type' = 'hive',
      'hive-conf-dir' = './conf');
```

Flink Catalogs




```
table.catalog-store.kind: file
table.catalog-store.file.path: ./conf/catalogs
```


kafka
DEMO
ICEBERG
https://github.com/decodableco/examples/kafka-iceberg

In Conclusion…

Flink SQL is Fun!

But there's a bit of a learning curve

- Run ad-hoc queries with the SQL Client
- Understand JAR dependencies for connectors, catalogs, formats, etc
- Don't be put off by the docs - there ***is*** SQL content there if you look hard enough

decodable.co/blog

February 19, 2024 10 min read

Catalogs in Flink SQL—Hands On

A hands-on guide to using catalogs with Flink SQL, including Apache Hive, JDBC, and Apache Iceberg with different metastores. Covers installation, setup, and usage.

February 27, 2024 4 min read

Flink SQL and the Joy of JARs

A handy guide to some of the most common issues with Flink SQL and JAR dependencies, how to troubleshoot them—and how to fix them! 🎉

March 12, 2024 5 min read

Exploring the Flink SQL Gateway REST API

Learn how to use the Flink SQL gateway's REST API to send SQL statements to your Flink cluster programmatically.

February 16, 2024 8 min read

Catalogs in Flink SQL—A Primer

Explore the essentials of catalogs in Flink SQL. Catalogs store object definitions like tables and views for the Flink query engine. This primer covers the role of catalogs in managing metadata in Flink, the different catalogs available in Flink, and how to use the CatalogStore.


#EOF
@rmoff / 19 Mar 2024 / #kafkasummit