DATA+AI SUMMIT 2022

Interactive Querying with Spark SQL

ORGANIZED BY databricks

Ashish Singh

Tech Lead, Big Data Query Platform,

Pinterest

Querying

Extracting information from data using SQL.

Querying

What about querying metadata of data?

Querying

- Scheduled Query: Query NOT executed by a user.

Querying

- Scheduled Query: Query NOT executed by a user.
- Interactive Query: Query executed by a user.

Scheduled vs Interactive Querying

	Scheduled	Interactive
Predictability	High	None
Results	Based on SLAs	Immediate
Syntax/Analysis errors	Low	High
ROI on Tuning	High	Low
Resource Efficiency Importance	High	Low

Data Scale at Pinterest

- 700+ PB in AWS S3
- 150K+ Data Compute Jobs per day (not including metadata only operations)
- ~20K Hadoop Nodes
- 1000+ Presto Workers
- 300K+ Hive Tables
- Everything in Cloud (AWS)

Query Engines at Pinterest

- Spark SQL
- Presto
- Flink SQL

Learn more about our Hive to Spark SQL migration

Advanced Migrations: From Hive to Spark SQL

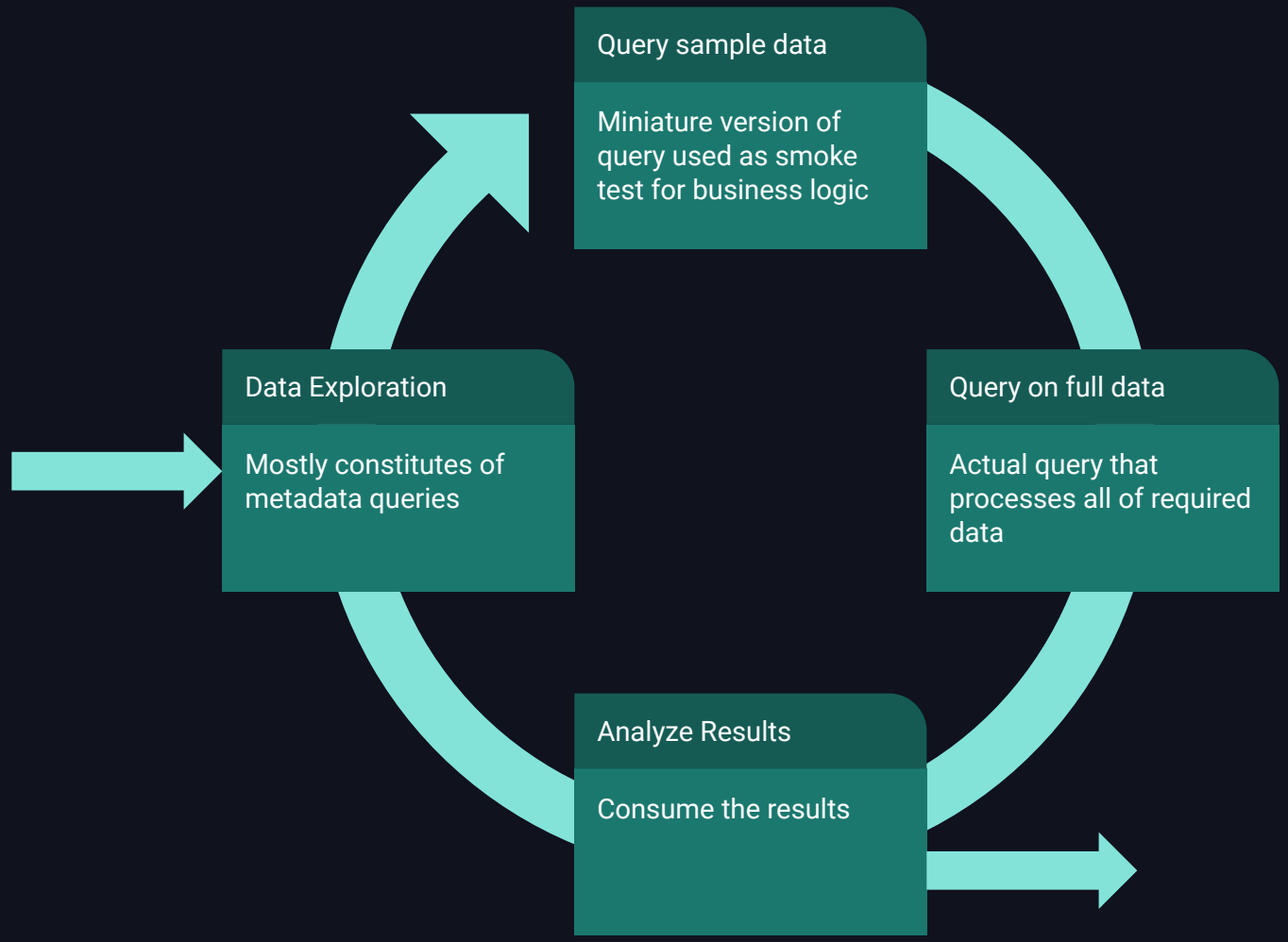
Wednesday, June 29 @4:45 PM

MOSCONE SOUTH | UPPER MEZZANINE | 211

By Zaheen Aziz

Interactive Querying

Usage pattern

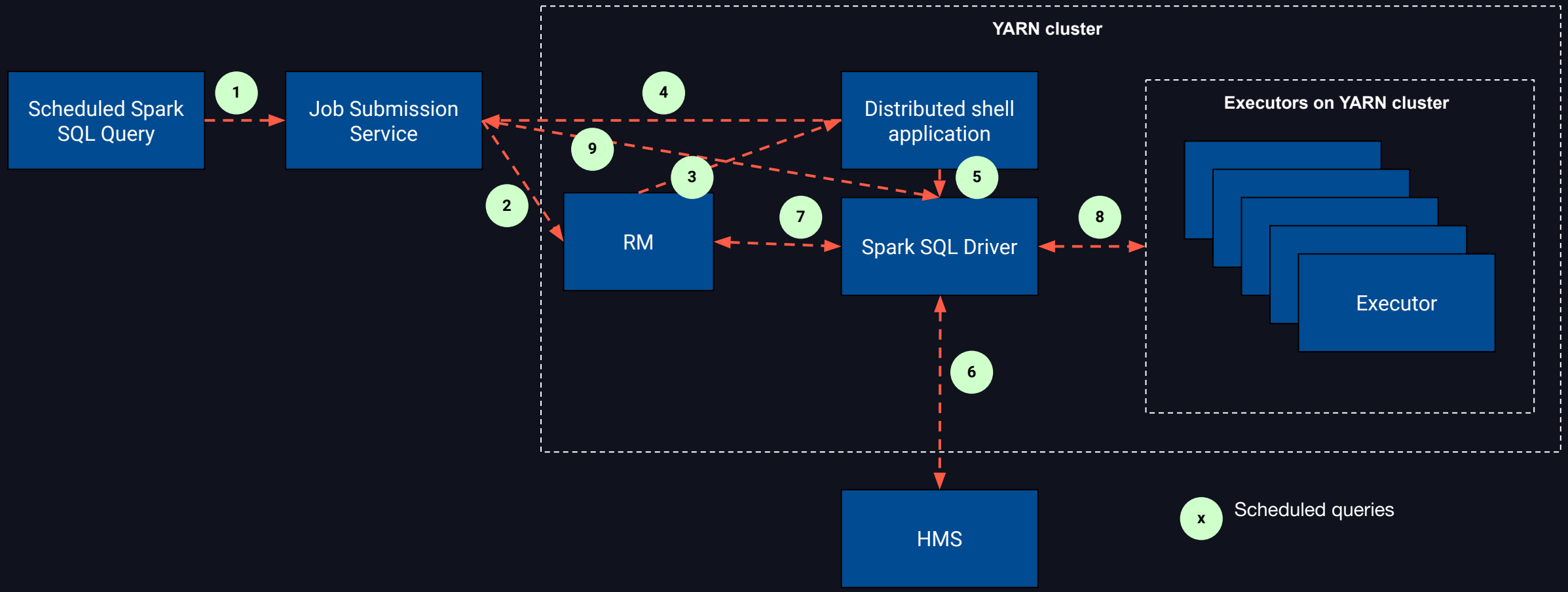


Interactive Querying

Requirements

- Seamless query submission
- Fast metadata queries
- Fail fast
- Easy access to results
- Error handling and tuning suggestions
- Auto error handling and retries

Scheduled Querying With Spark SQL



Interactive Querying

Requirements Check

- Seamless query submission ✖
- Fast metadata queries ✖
- Fail fast ✖
- Easy access to results ✖
- Error handling and tuning suggestions ✖
- Auto error handling and retries ✖

Architectural Choices

Distributed Shell Application

- Spark-sql CLI as dist shell application on YARN
- Waiting for container allocations
- Retrieving results is tricky
- Tracking statement-level progress is hard

Apache Spark Thrift Server

- Similar to HiveServer2
- JDBC/ ODBC protocols
- No isolation between queries

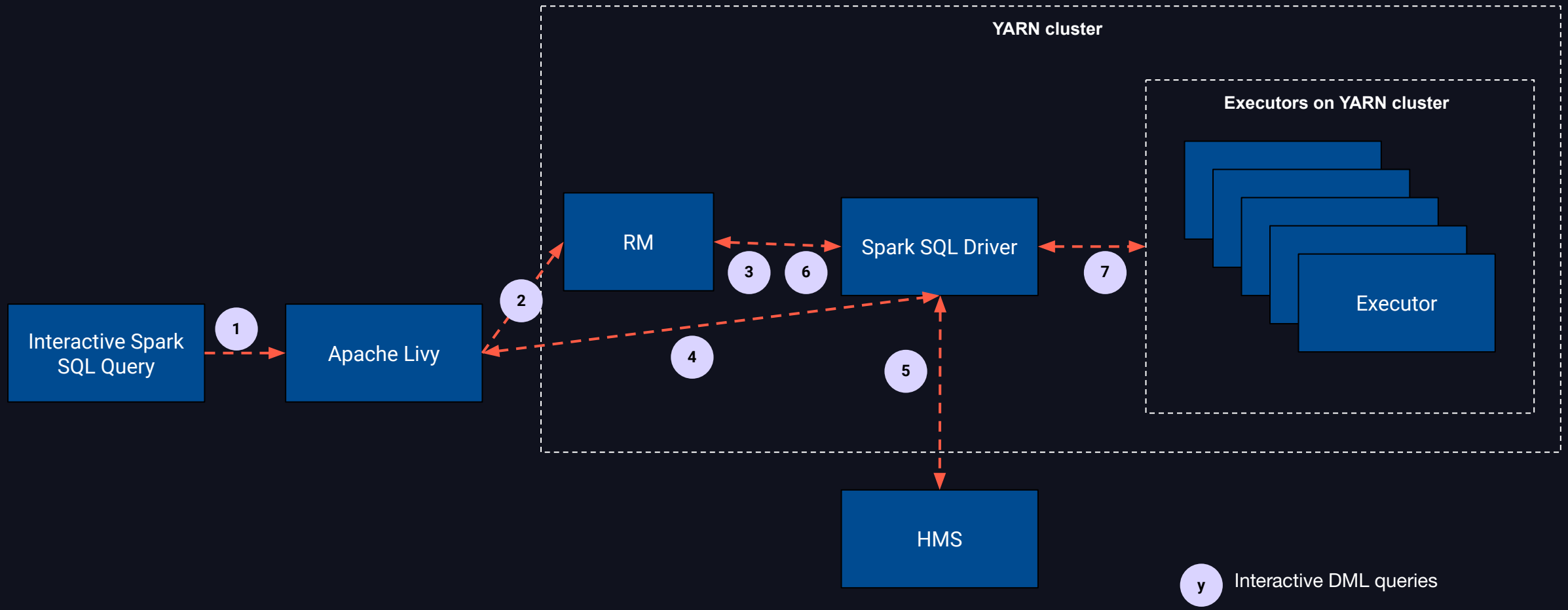
Apache Livy - Batch Sessions

- REST interface to Spark cluster
- Similar to spark-submit
- Multi-tenancy, high availability, and failure isolation

Apache Livy - Interactive Sessions

- Enables reusing sessions
- Multi-tenancy, high availability, and failure isolation

Querying With Spark SQL



Interactive Querying

Requirements Check

- Seamless query submission ✖
- Fast metadata queries ✖
- Fail fast ✖
- Easy access to results ✖
- Error handling and tuning suggestions ✖
- Auto error handling and retries ✖

Seamless Query Submission

From existing query clients like querybook, jupyter, etc

- Generic DB-API 2.0 compliant Python client on top of Livy's REST API

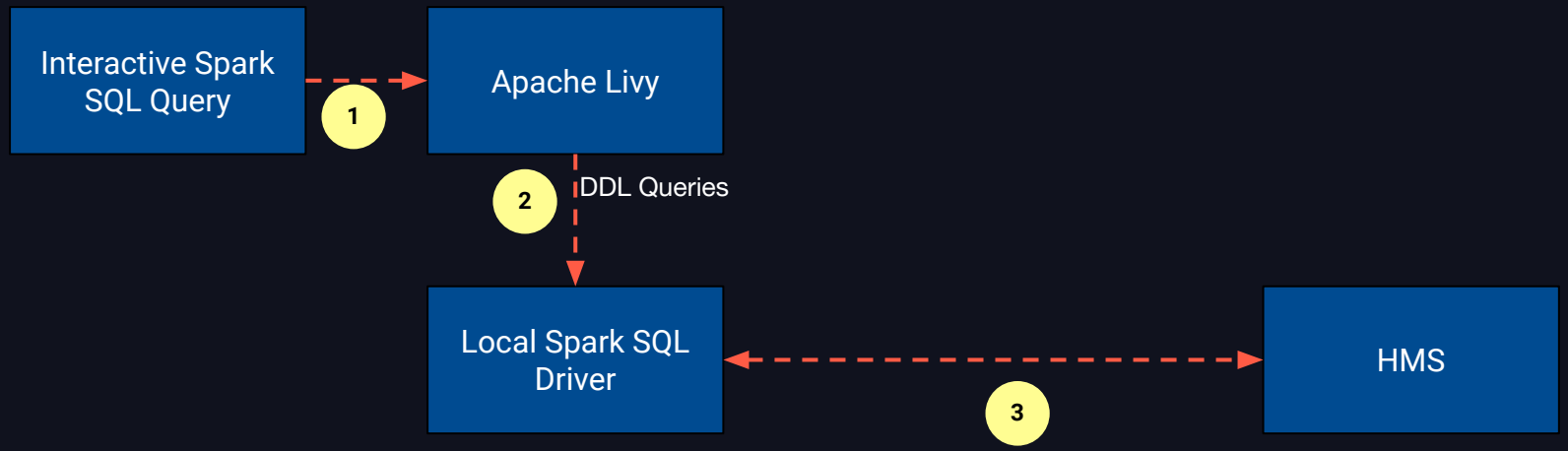
```
from bigpy.hive.livy.livy_client import Connection
from bigpy.hive.livy.livy_client import Cursor

cursor = Connection(livy_connection_urls='host1:port1').cursor()
cursor.execute('SELECT * FROM my_awesome_data LIMIT 10', async=True)

status = cursor.poll().operationState
while status == Cursor._STATE_RUNNING):
    logs = cursor.fetch_logs()
    for message in logs:
        print message
    status = cursor.poll().operationState

print cursor.fetchall()
```

Fast Metadata Queries

The problem..

- Waiting for container allocations
- Waiting for Spark Context to start up

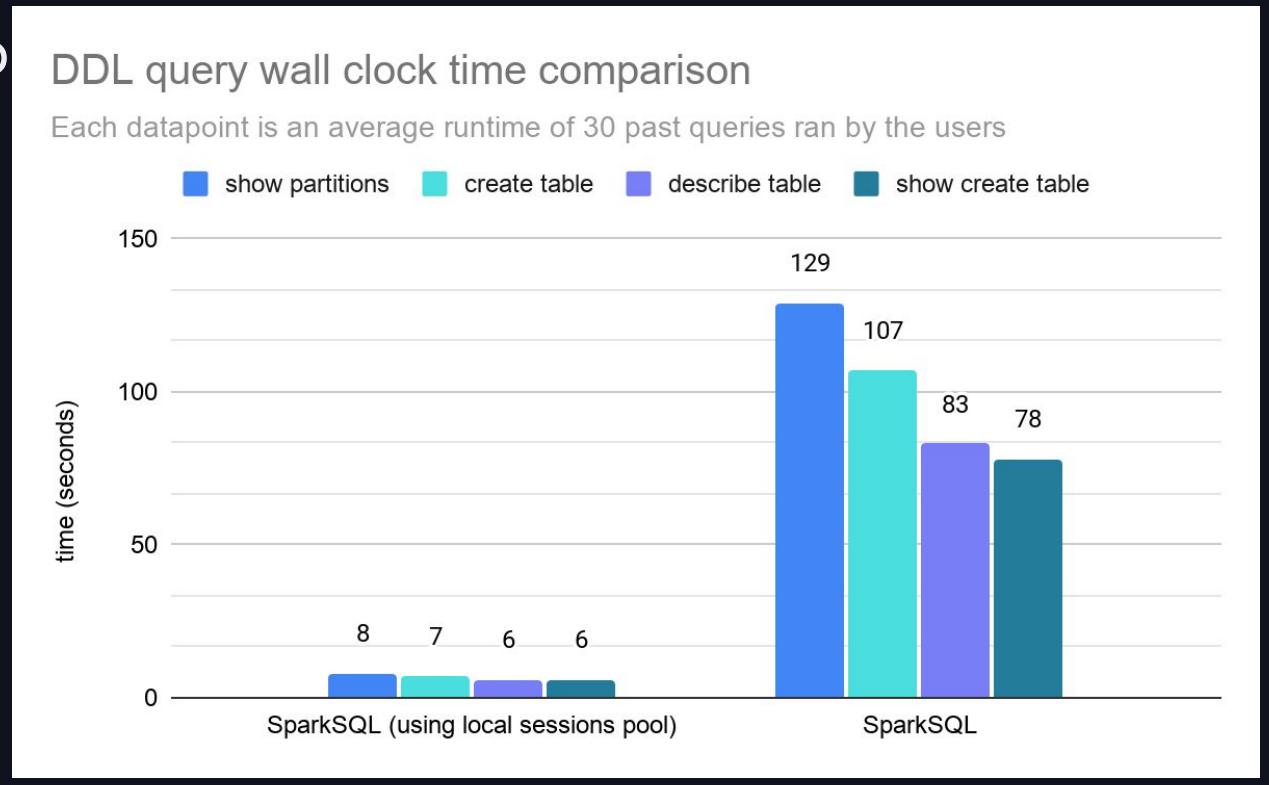
Fast Metadata Queries



z Interactive DDL queries

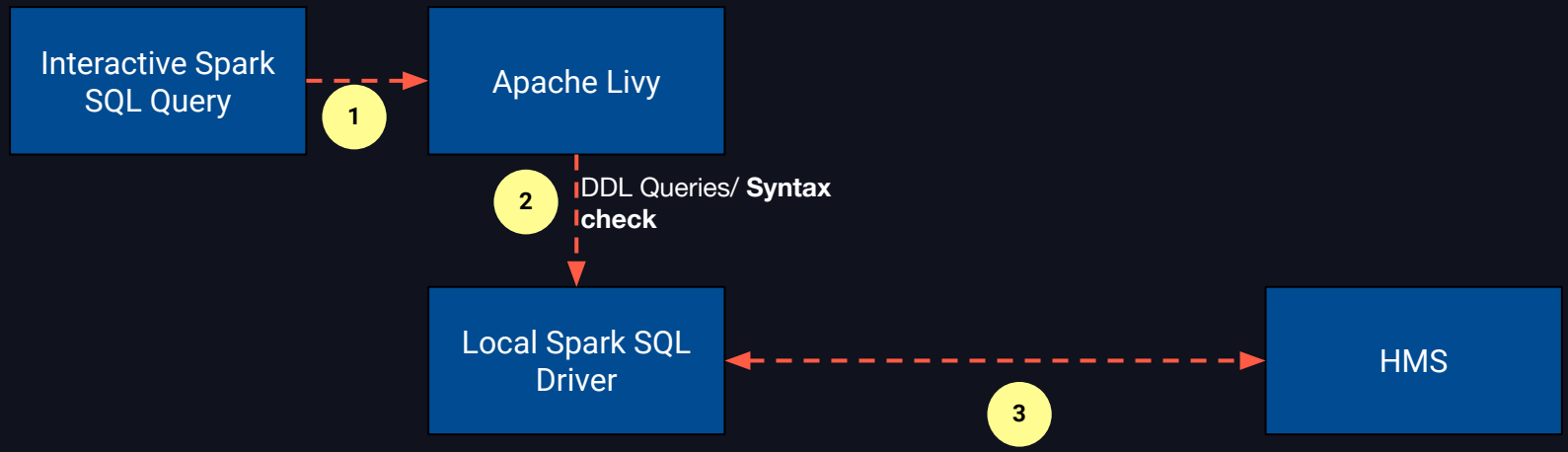
Fast Metadata Queries

- Pool is initialized on server startup
- Self-reliant, some of the features include:
  - Automatic garbage collection
  - Health monitoring
  - Asynchronous loading



With this design, we reduced query latency from 70 seconds to an average of 10 seconds (~6.3x improvement)

Fail Fast



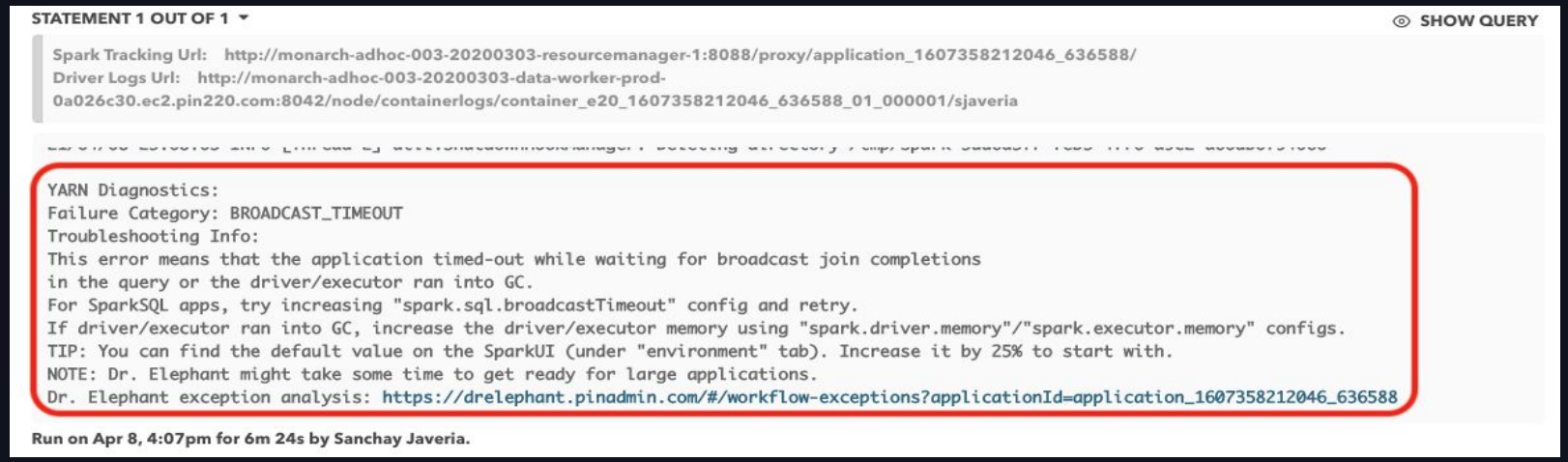
z Interactive DDL queries

Easy Access to Results

- Livy limits the number of rows a query can return
- Needed to avoid Livy from choking on memory
- For large results, we redirect results to AWS S3
- Clients read results from S3 on query completion

Error Handling Recommendations

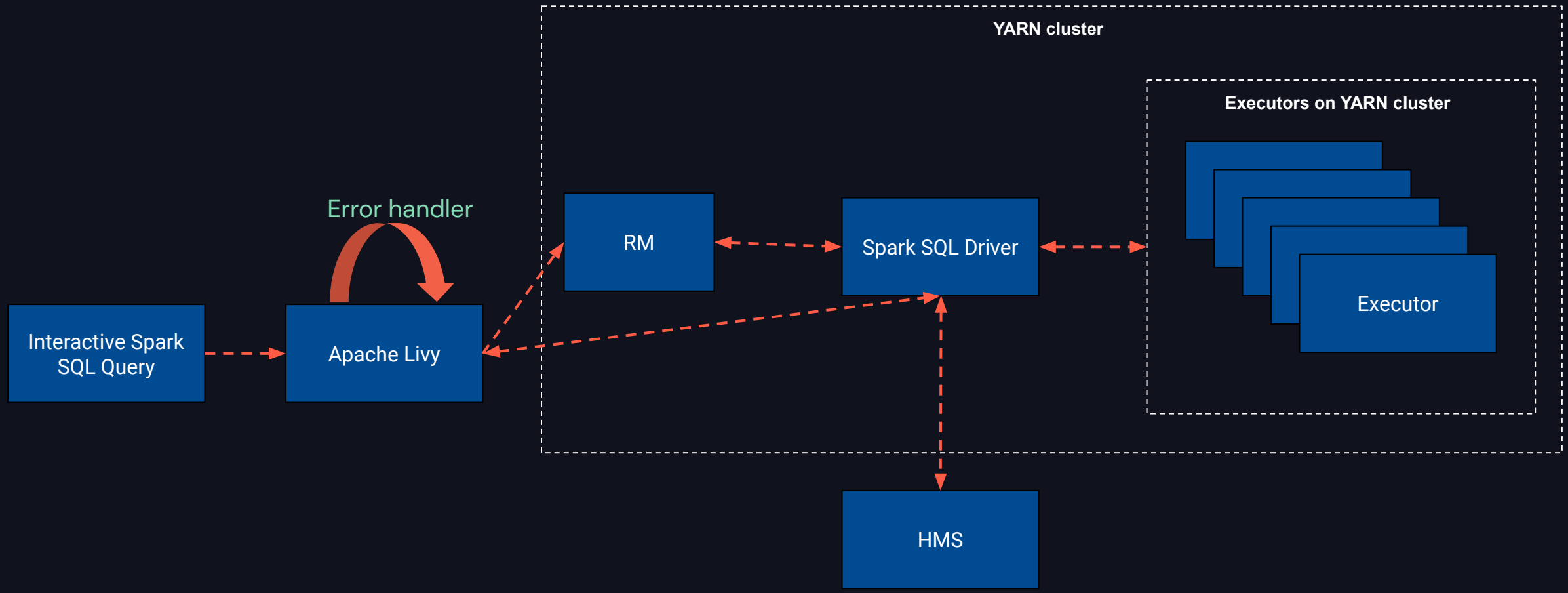
- Utilize YARN diagnostic and Spark error messages
- Fail YARN application based on last query's execution status
- Added an error classifier/ troubleshooting information to Dr. Elephant
- Clients retrieve troubleshooting information from Dr. Elephant APIs



Resource Tuning Suggestions

- User applications are often too generous with memory
- Provide visibility into over/ under memory consumption

Auto Error Handling and Retries



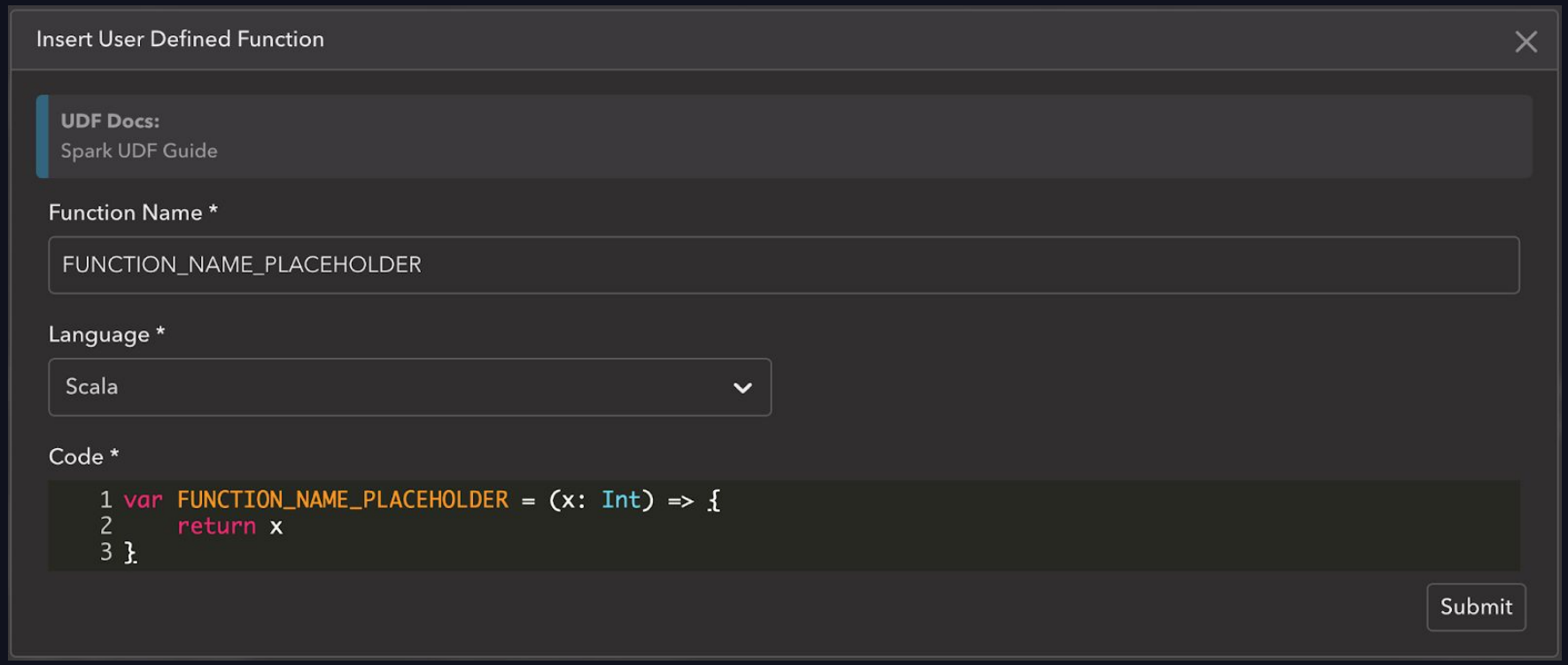
Interactive Querying

Requirements Check

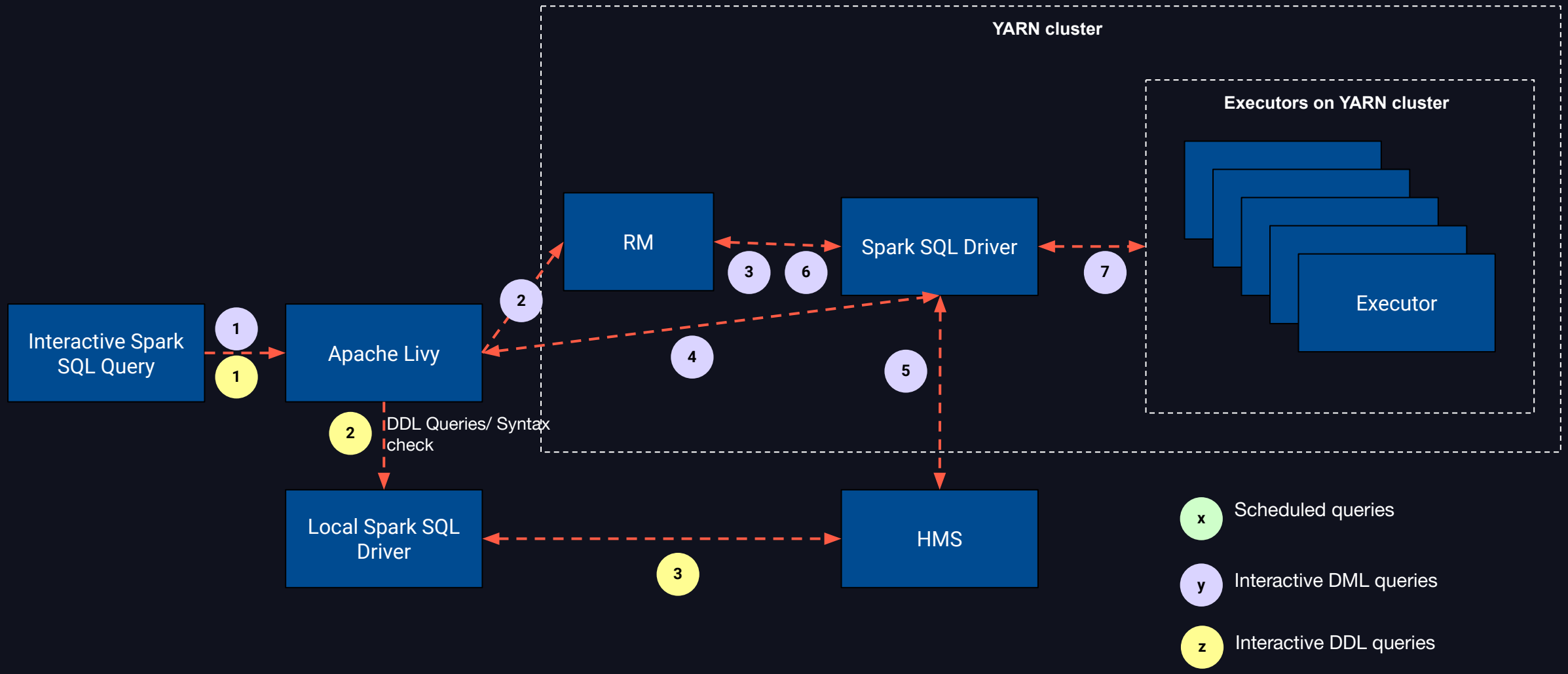
- Seamless query submission
- Fast metadata queries
- Fail fast
- Easy access to results
- Error handling and tuning suggestions
- Auto error handling and retries

Interactive UDFs

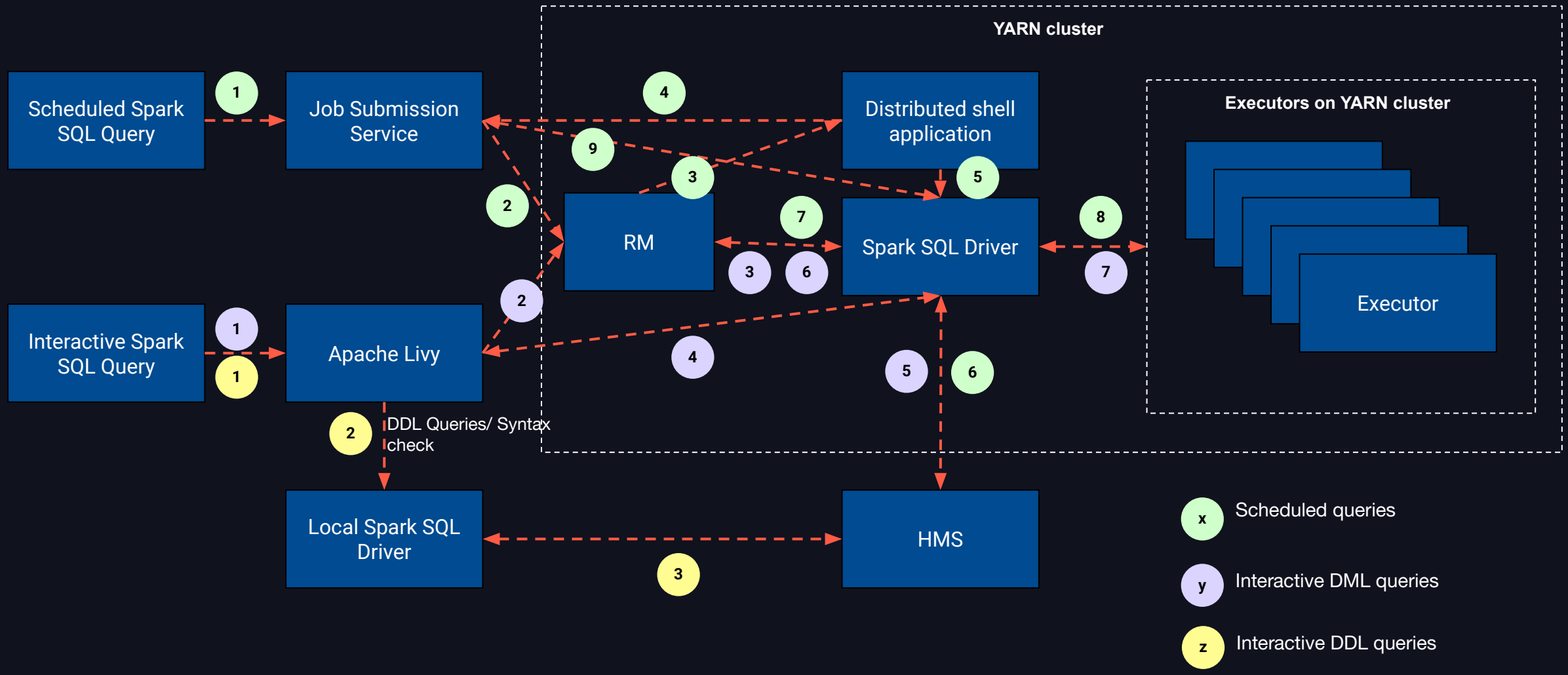
- Allow users to author UDFs along with queries.



Interactive Querying With Spark SQL



Querying With Spark SQL



DATA+AI SUMMIT 2022

Thank you

Ashish Singh

Tech Lead, Big Data Query Platform,

Pinterest