DATA+AI SUMMIT
BY databricks

Unity Catalog Lakeguard: Data Governance for Multi-User Apache™ Spark Clusters

Stefania Leone
Director Product Management, Databricks

Martin Grund
Principal Software Engineer, Databricks

Product safe harbor statement

This information is provided to outline Databricks' general product direction and is for informational purposes only. Customers who purchase Databricks services should make their purchase decisions relying solely upon services, features, and functions that are currently available. Unreleased features or functionality described in forward-looking statements are subject to change at Databricks discretion and may not be delivered as planned or at all

OVERVIEW

1. Why Data Governance

2. Data Governance with Apache Spark

3. Unity Catalog Lakeguard

WHY DATA GOVERNANCE?

DANGER
KEEP OUT
AUTHORIZED
PERSONNEL ONLY

DATA GOVERNANCE

Example: PII (Personally identifiable information)

customers

Name	Date of birth	Email	SSN
Jane Data	04-03-1980	jane.data@gmail.com	123-34-5671
John Smith	12-12-1989	john@smith.com	231-45-1231
Alice Bricks	03-08-2000	a.bricks@example.com	999-09-1234
…	…	…	…

DATA GOVERNANCE: EXAMPLE

Data engineers have access to all the data

customers

Name	Date of birth	Email	SSN
Jane Data	04-03-1980	jane.data@gmail.com	123-34-5671
John Smith	12-12-1989	john@smith.com	231-45-1231
Alice Bricks	03-08-2000	a.bricks@example.com	999-09-1234
…	…	…	…

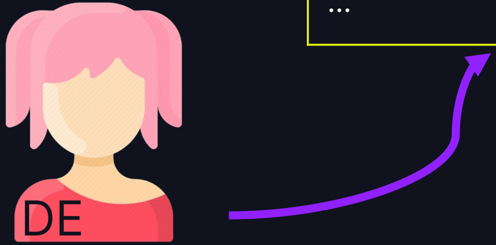
```
GRANT SELECT ON customers TO `data engineers’
```

DATA GOVERNANCE: EXAMPLE

Data engineer with SELECT on customers

customers

Name	Date of birth	Email	SSN
Jane Data	04-03-1980	jane.data@gmail.com	123-34-5671
John Smith	12-12-1989	john@smith.com	231-45-1231
Alice Bricks	03-08-2000	a.bricks@example.com	999-09-1234
...	...	...	...



```
SELECT * FROM customers
```

DATA GOVERNANCE: EXAMPLE

Data scientist don’t have access to all customer data

`customers_view`

Name	Email
Jane Data	jane.data@gmail.com
John Smith	john@smith.com
Alice Bricks	a.bricks@example.com
…	…

`customers`

Name	Date of birth	Email	SSN
Jane Data	04-03-1980	jane.data@gmail.com	123-34-5671
John Smith	12-12-1989	john@smith.com	231-45-1231
Alice Bricks	03-08-2000	a.bricks@example.com	999-09-1234
…	…	…	…

```
GRANT SELECT ON VIEW customer_view TO `data scientists’
```

FINE-GRAINED ACCESS CONTROL

Data scientist with SELECT on customers_view

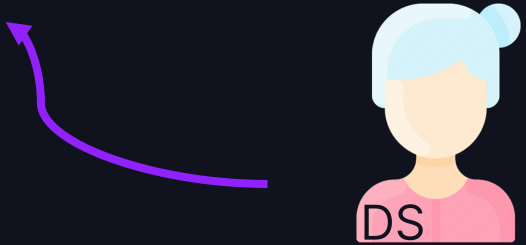
Fine-grained access control (FGAC) includes

- Views
- Row-level & column-level filters
- Attribute-based access control

`customers_view`

Name	Email
Jane Data	jane.data@gmail.com
John Smith	john@smith.com
Alice Bricks	a.bricks@example.com
…	…

```
SELECT * FROM customers_view
```



DATA GOVERNANCE WITH APACHE SPARK

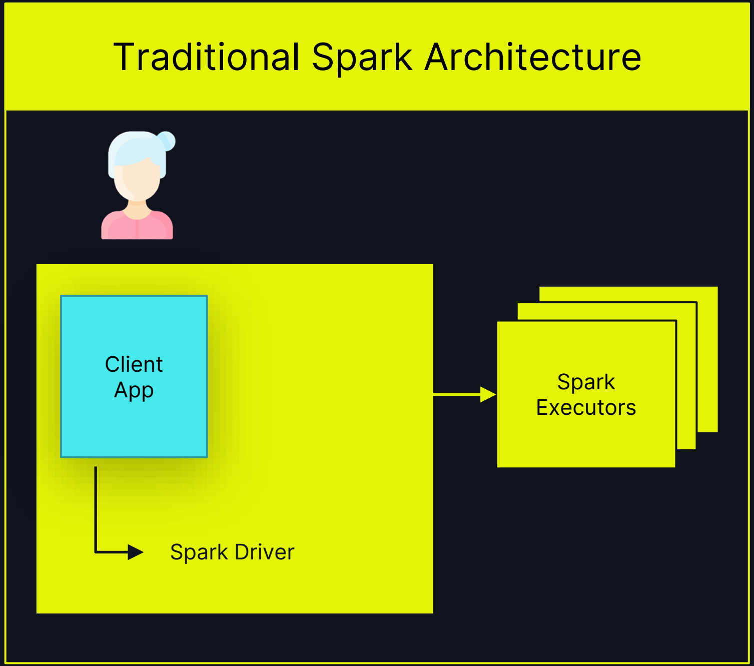
APACHE SPARK AND DATA GOVERNANCE

- Apache Spark de facto big data processing framework
- Wasn't built with data governance in mind:
    - Single JVM, no decoupling of Spark engine and application

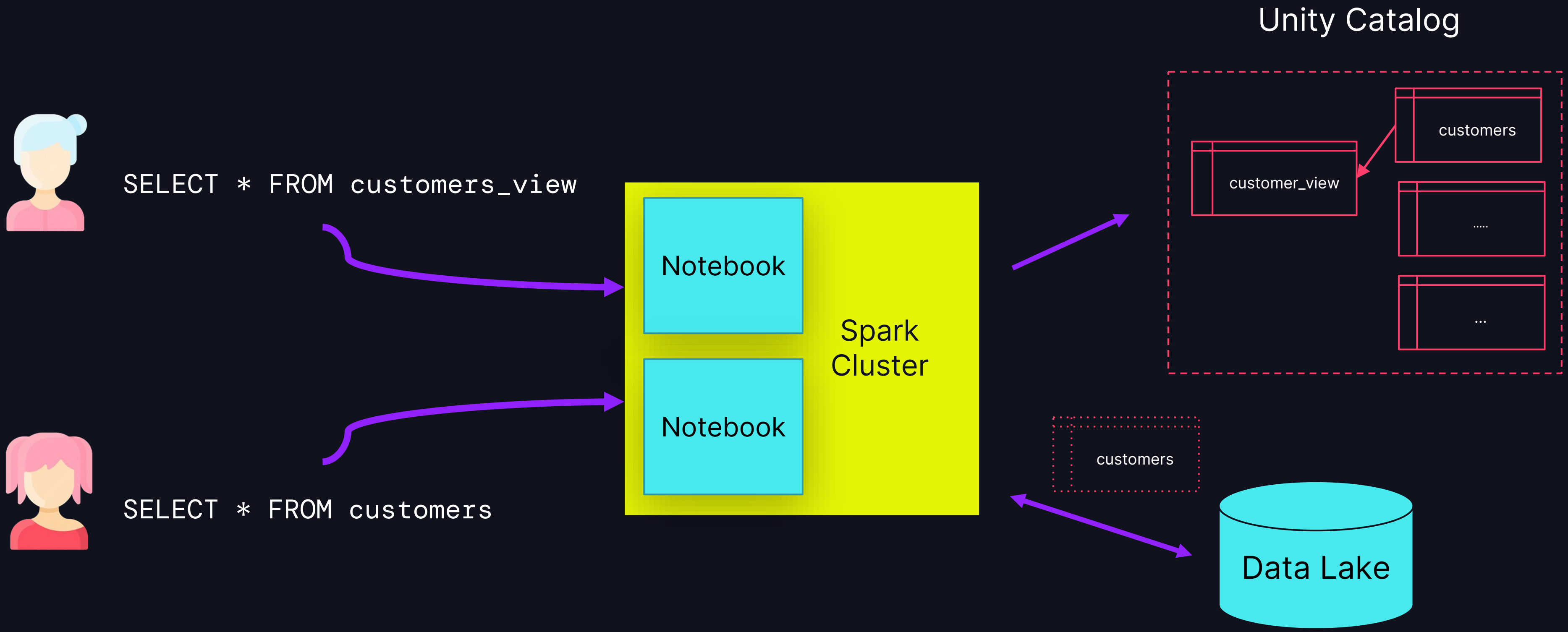
      → Single-application/user

      → Cluster as isolation boundaries

However, users want to share compute to reduce cost and operational burden

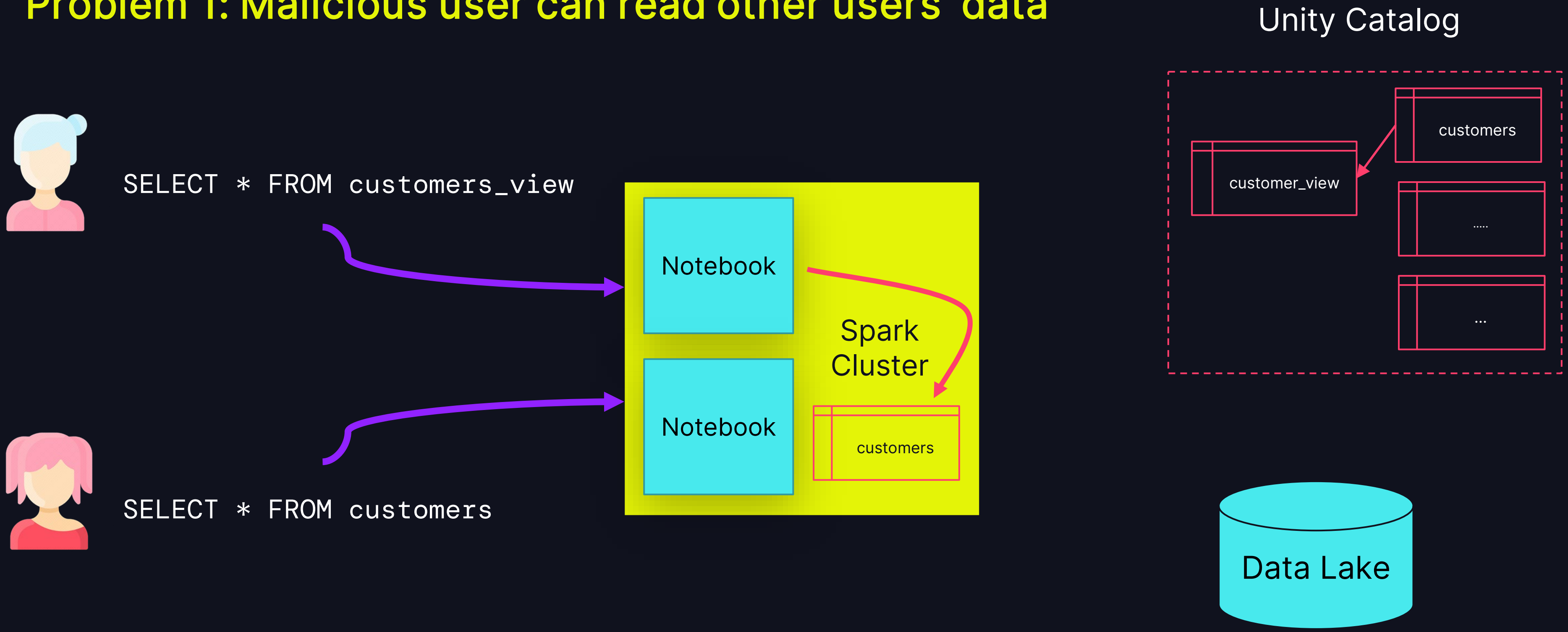


WHEN SHARING COMPUTE



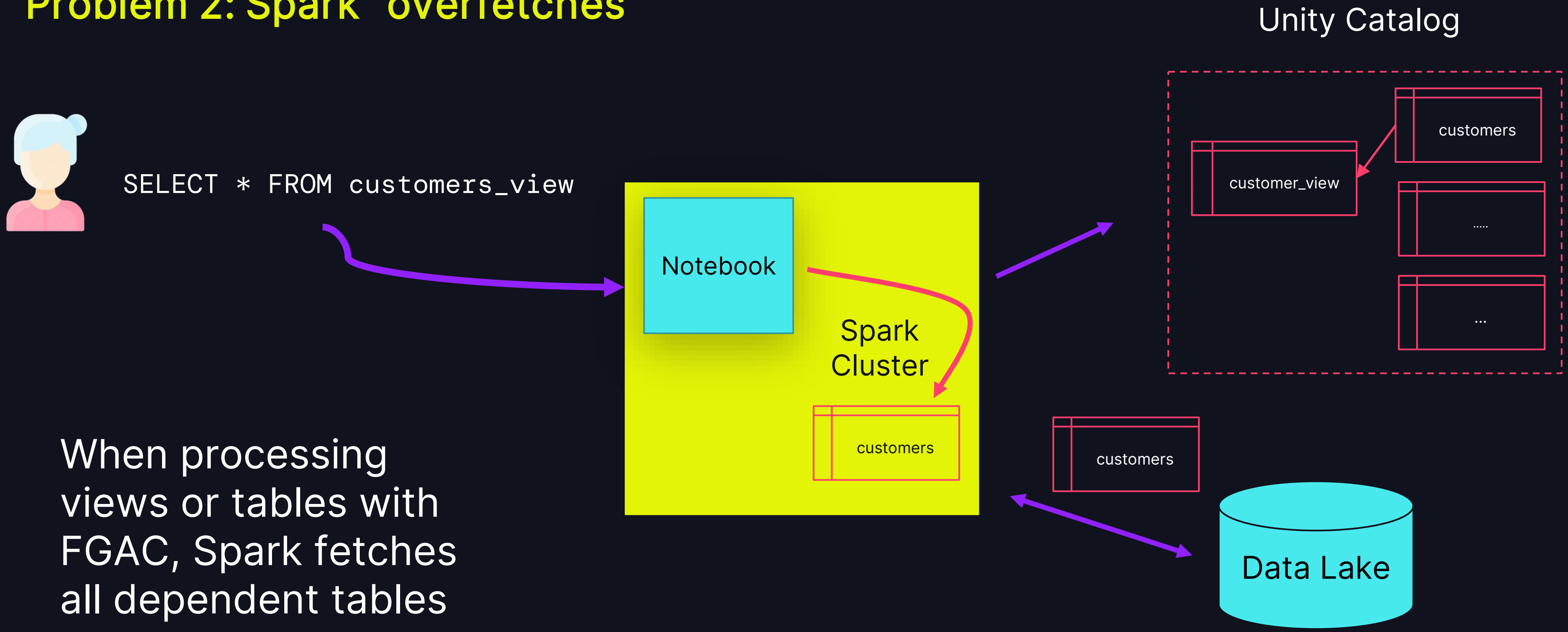
WHEN SHARING COMPUTE

Problem 1: Malicious user can read other users' data



WHEN SHARING COMPUTE

Problem 2: Spark “overfetches”



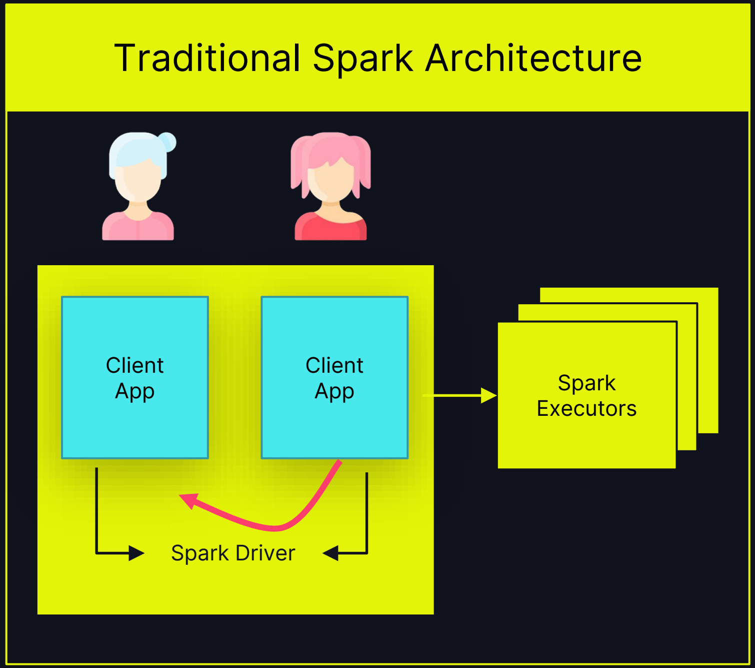
When processing views or tables with FGAC, Spark fetches all dependent tables

APACHE SPARK AND DATA GOVERNANCE

Summary

- Spark enforces governance at cluster boundaries: No isolation between Spark and client applications (Problem 1)
- Spark "overfetches" files when querying view or tables with FGAC (Problem 2)

However, users want data governance and shared compute to reduce cost and operational burden



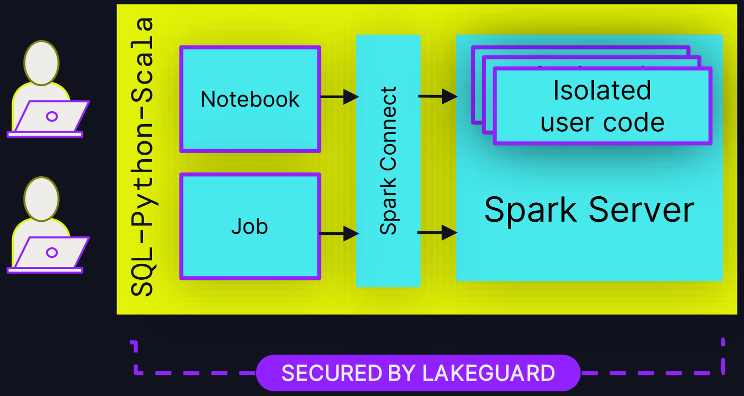
UNITY CATALOG LAKEGUARD

FULL DATA GOVERNANCE IN DATABRICKS

UC COMPUTE: MULTI-USER SPARK CLUSTERS

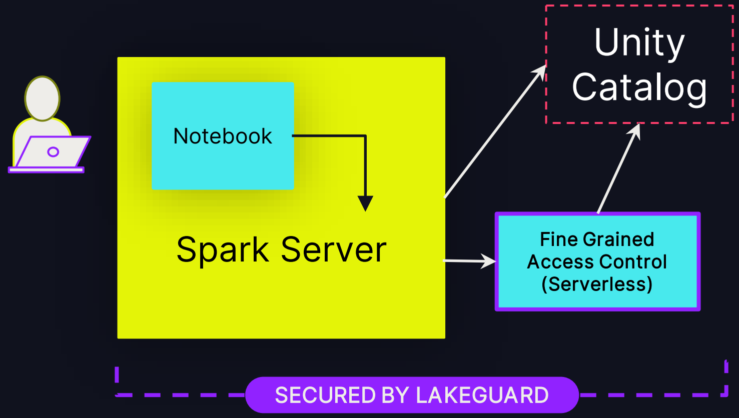
Shared access mode

- Secure multi-user: fully isolates user code
- Full UC governance
- Declarative data access (DataFrame API based on Spark Connect)



Single-user access mode

- Single user w/ privileged access to the underlying machine
- Full, unrestricted Spark API



UC SHARED CLUSTERS

MULTI-USER COMPUTE IN SCALA, PYTHON, AND SQL.

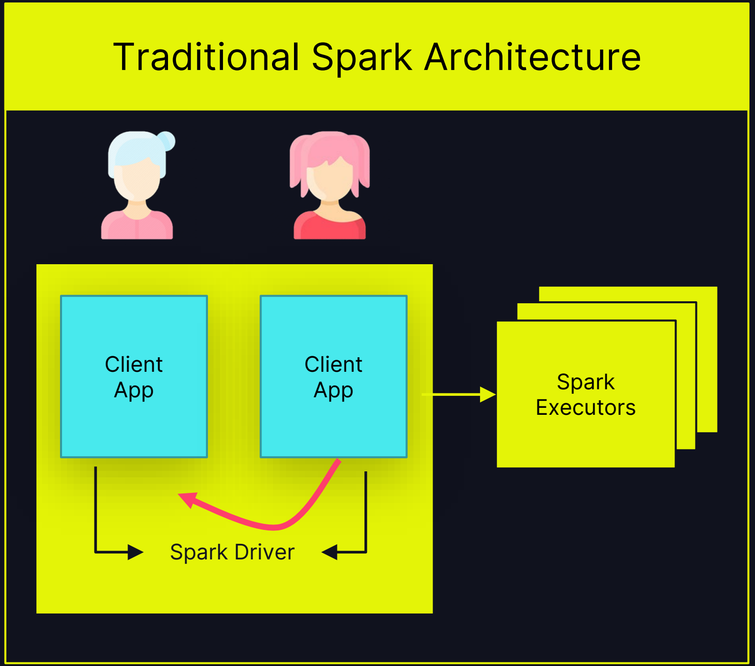
ACHIEVING FULL USER ISOLATION

How we solved problem 1 AND 2 at the same time

Goal: Separate users from each other and from the Spark engine - in SQL, Python and Scala

How:

1. Client Isolation: Isolate Notebooks & Jobs from each other and the engine
2. Executor Isolation: UDFs (SQL, Python, Scala)



CLIENT ISOLATION

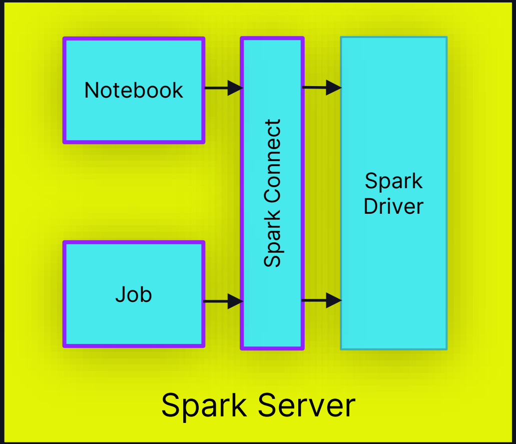
Isolating user code...

... from the Spark engine

- Spark Connect (Apache Spark 3.4)
- Decoupled client-server architecture based on Dataframe API
- Overfetching no longer a problem

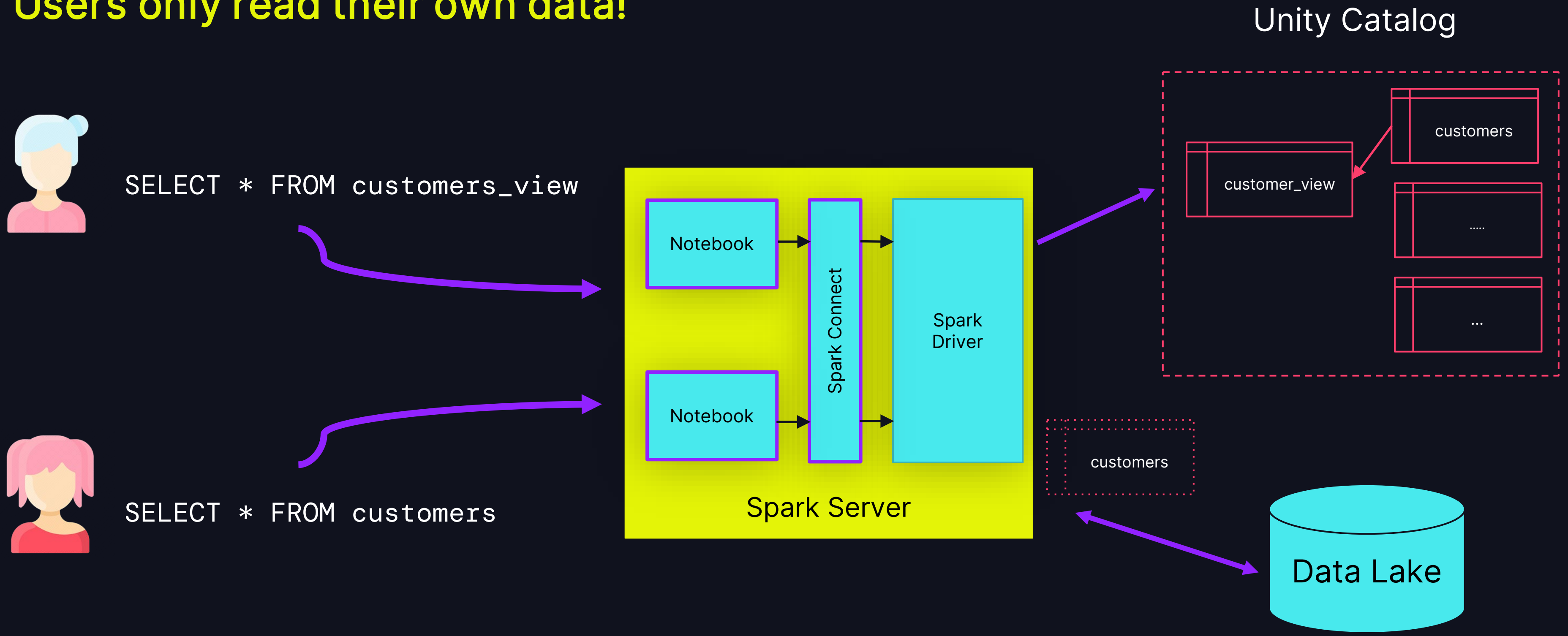
... from other users

- Spark Sessions (Notebook, Job) isolated using sandboxing techniques.



SHARED CLUSTERS: ISOLATED USER CODE

Users only read their own data!

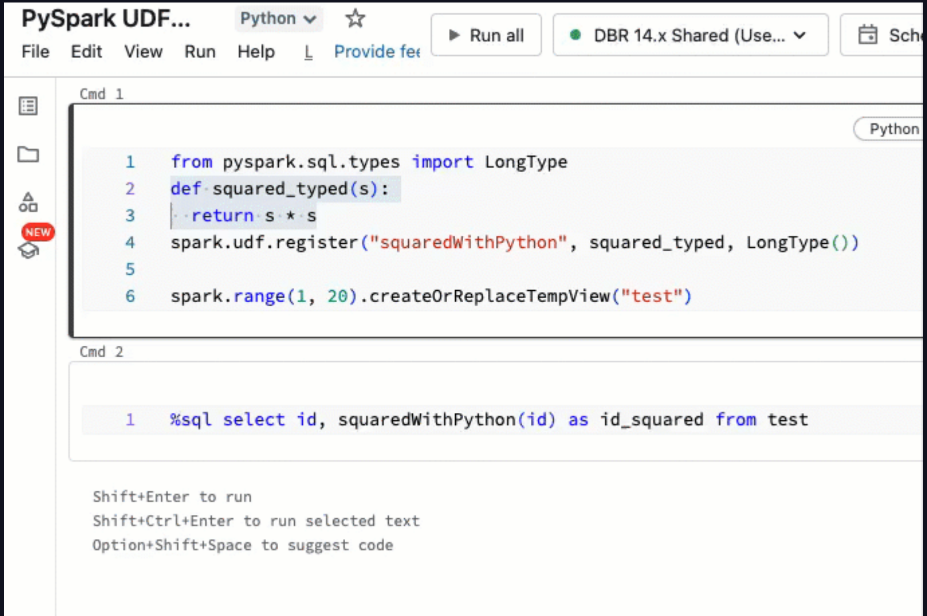


WHAT ABOUT UDFS?

User defined functions

What are UDFs?

- User-defined code in SQL, Python, or Scala
- May define dependencies
- Session-based or registered with UC
- Great for distributed processing: Runs on Spark executors



```
from pyspark.sql.types import LongType
def squared_typed(s):
  return s * s
spark.udf.register("squaredWithPython", squared_typed, LongType())

spark.range(1, 20).createOrReplaceTempView("test")
```

```
%sql select id, squaredWithPython(id) as id_squared from test
```

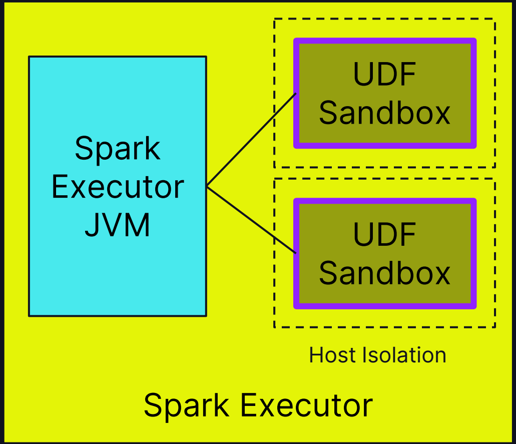
Shift+Enter to run
Shift+Ctrl+Enter to run selected text
Option+Shift+Space to suggest code

EXECUTOR ISOLATION: SANDBOXED UDFS

Isolation of UDF in sandboxed execution environment

- No sharing of the executor JVM
- Isolated network rules and host access
- Dynamically replicating client dependencies into the sandbox

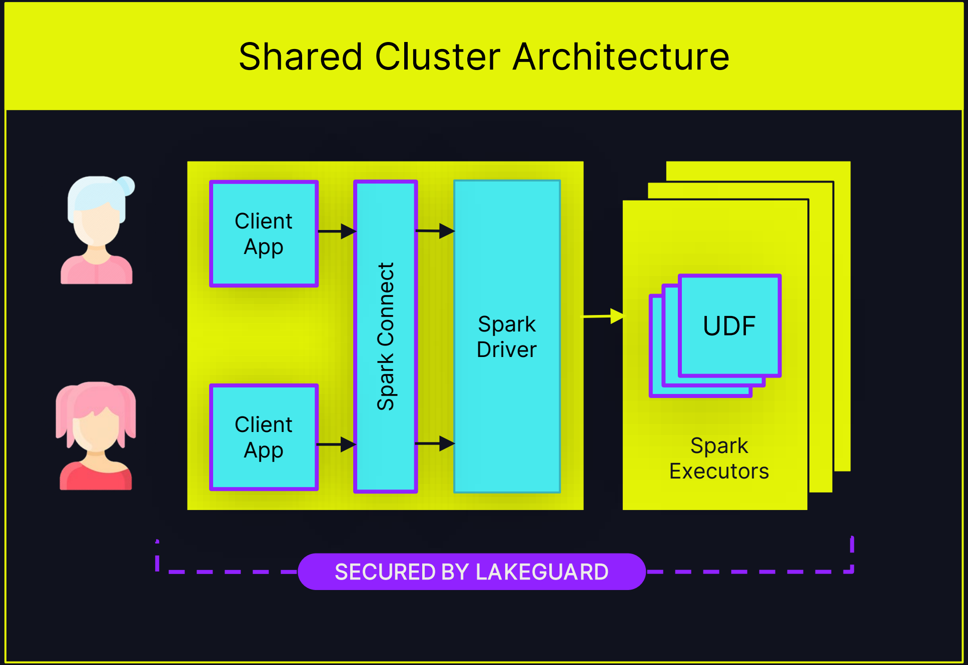
Also available on DBSQL and DLT



UC SHARED CLUSTER IN A NUTSHELL

Lakeguard enforces data governance at compute level

- Cost-efficient multi-user compute in Python, Scala & SQL
- Full data governance incl. fine-grained access control
- Declarative Spark API based on Spark Connect
- For interactive development and automated jobs
- Foundation for serverless



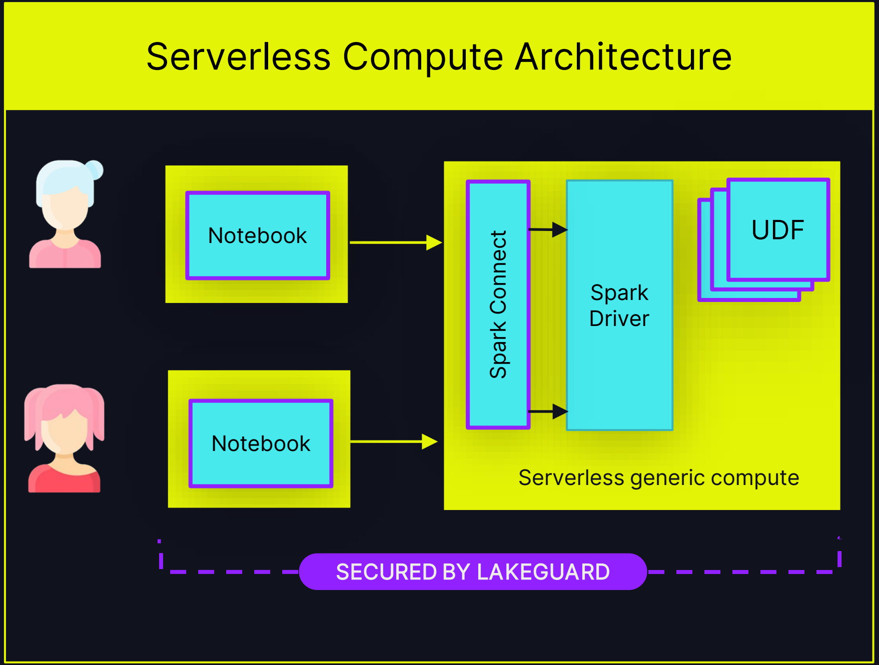
SERVERLESS COMPUTE

Serverless Notebooks and Workflows

Share same architecture & capabilities as Shared Clusters.

=> If your workload runs on Shared Clusters today, simply transition to serverless!

=> If your workload runs on Single-User Clusters today, test using Shared Clusters



Limitations

Not all Spark workloads run on Shared Clusters and Serverless

Machine Learning (MLR)

- Spark Connect does not support RDDs for arbitrary code execution.
  - No support for distributed ML
  - No support for multi GPU clusters
- No flat cluster network to support libraries such as Horovod, PyTorch, Ray

Privileged Machine Access

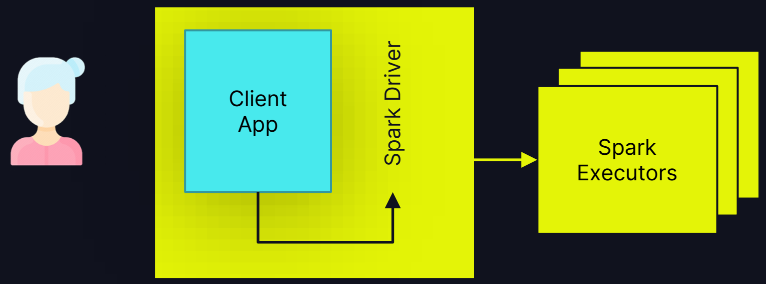
- No configuration of the underlying host VM -> no operating system libraries, compilers etc.

UNITY CATALOG SINGLE-USER CLUSTERS

UNITY CATALOG SINGLE-USER CLUSTERS

Recap

- Single-user with privileged access to the underlying machine
  => No Sharing (Problem 1)



- Full unrestricted Spark API
  => No fine-grained access control. (Problem 2)

How to share compute for ML Workloads?
How to provide Fine-Grained Access Control?

HOW CAN WE SHARE SINGLE-USER CLUSTERS?

Taking a step back:

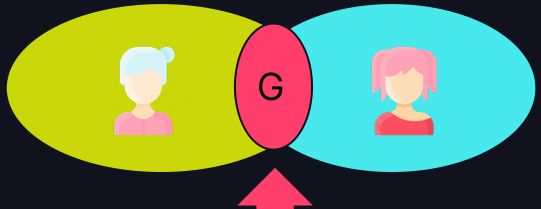
- How do we issue grants?
  - Option 1: GRANT SELECT on `customer` to `John Doe`
  - Option 2: GRANT SELECT on `customer` to `Data Scientists`

HOW CAN WE SHARE SINGLE-USER CLUSTERS?

Taking a step back:

- How do we issue grants?
    - ~~Option 1: GRANT SELECT on `customer` to `John Doe`~~
    - Option 2: GRANT SELECT on `customer` to `Data Scientists`

HOW CAN WE SHARE SINGLE-USER CLUSTERS?



Taking a step back:

- How do we issue grants?
  - ~~Option 1: GRANT SELECT on `customer` to `John Doe`~~
  - Option 2: GRANT SELECT on `customer` to `Data Scientists`

What if we let users share a cluster as a *group*,

where all users have the *same* permissions?

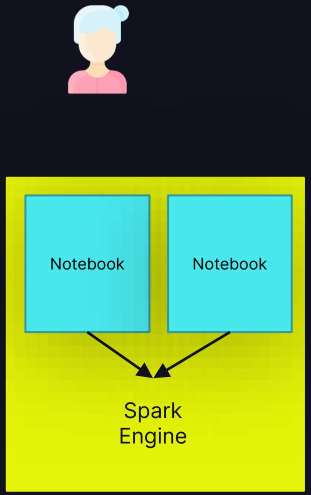
“SINGLE GROUP” CLUSTER

Private Preview

PREVIEW SIGN-UP FORM

Single-User Today:

One cluster, one user.



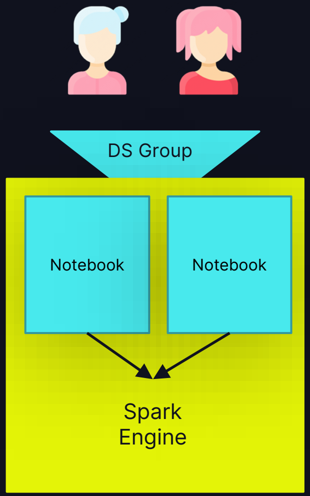
“SINGLE GROUP” CLUSTER

Private Preview

PREVIEW SIGN-UP FORM

Sharing compute by assigning a single group to the cluster.

For teams of Data Scientists and ML engineers.



“SINGLE GROUP” CLUSTER

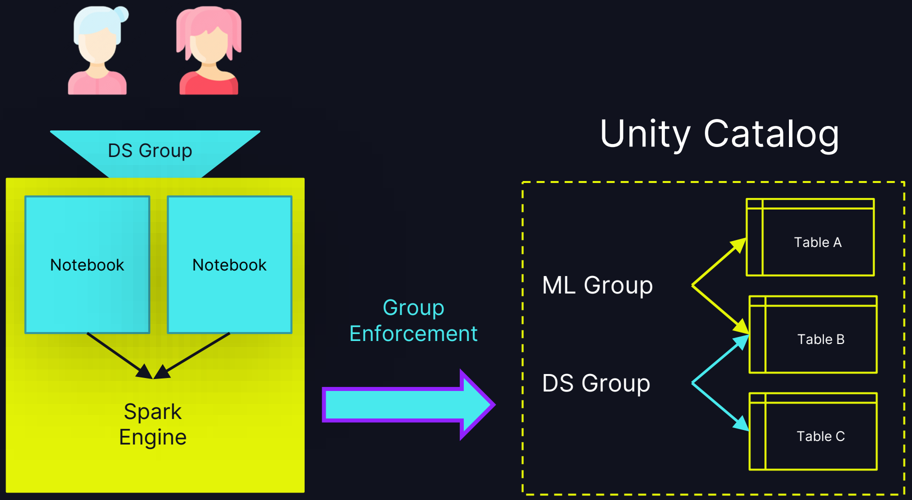
Private Preview

PREVIEW SIGN-UP FORM

Sharing compute by assigning a single group to the cluster.

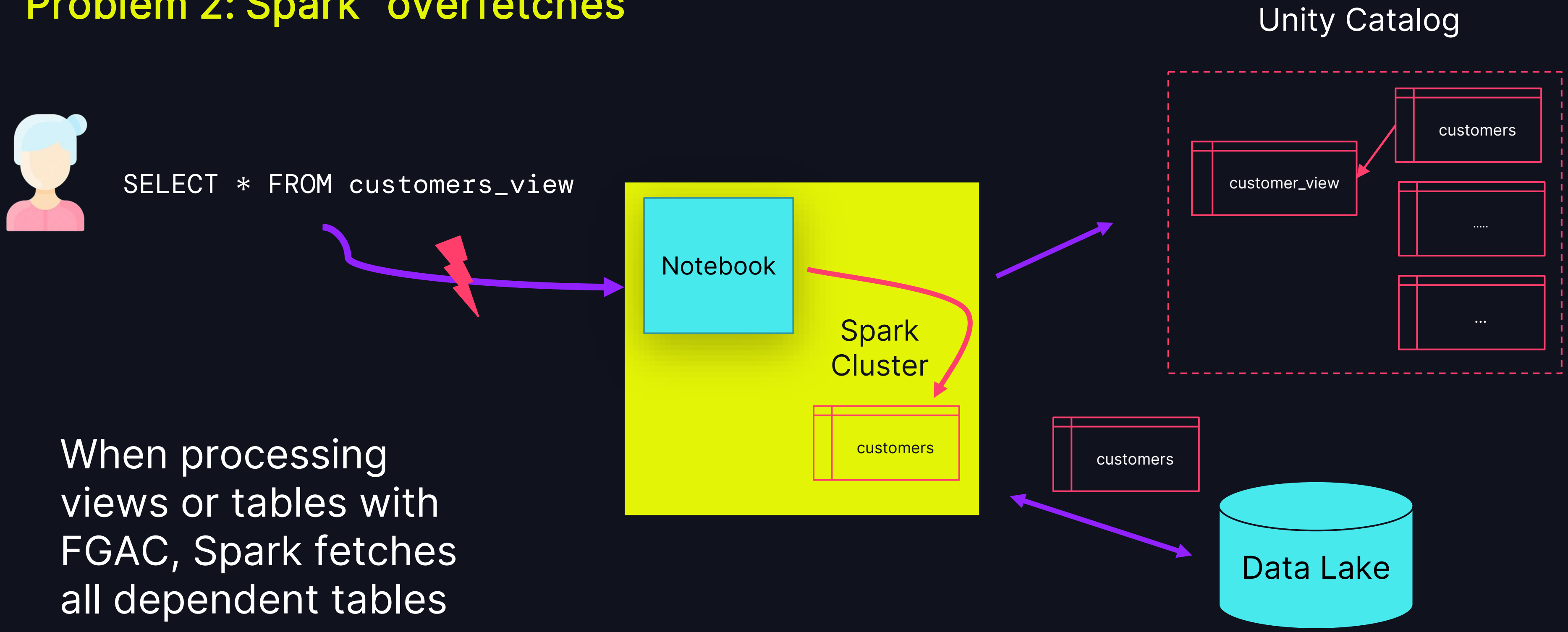
For teams of Data Scientists and ML engineers.

- Single-User and Single-Group are the same access mode!
- With PuPr, we will simplify the naming and UX



RECAP: OVERFETICHING

Problem 2: Spark “overfetches”

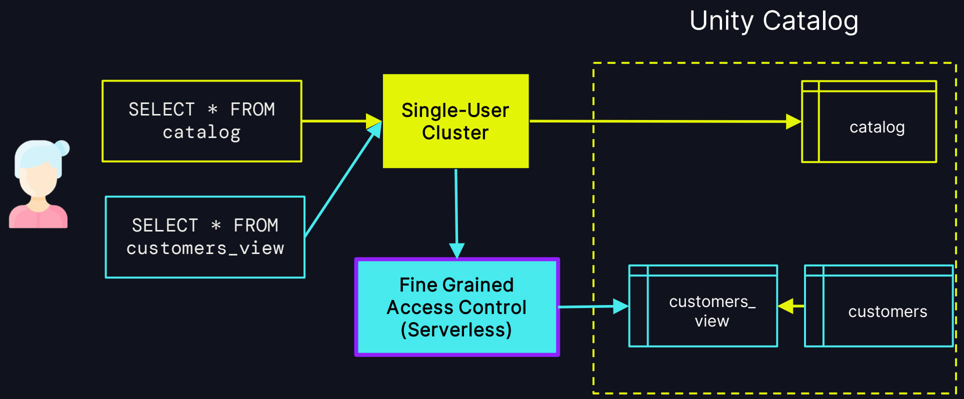


When processing views or tables with FGAC, Spark fetches all dependent tables

FINE-GRAINED ACCESS CONTROL

For Single-User Clusters

Seamlessly query views and tables protected by FGAC securely from Single User clusters!



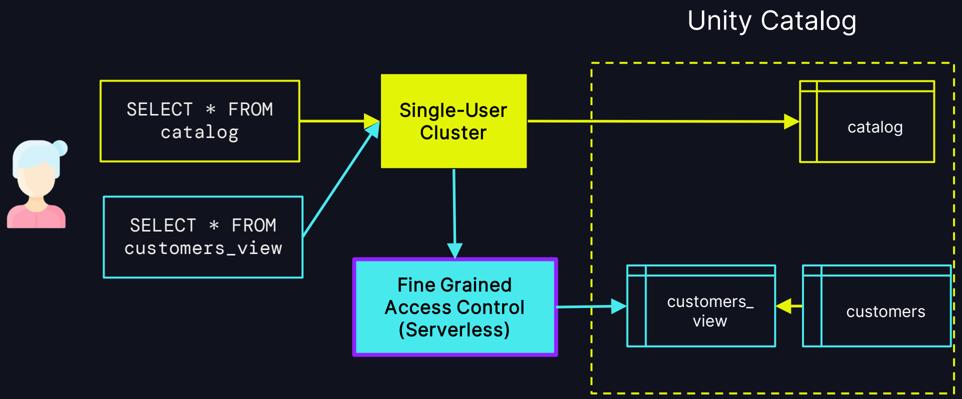
FINE-GRAINED ACCESS CONTROL

For Single-User Clusters

Seamlessly query views and tables protected by FGAC securely from Single User clusters!

View and masked table access:

- Data filtered via secure, serverless filtering service
- Filtered results are sent back to the Single User cluster
- *Priced* at the rate of Serverless Jobs



Public Preview, coming

RECOMMENDATIONS

RECOMMENDATIONS

Working securely with your governed lakehouse

1. Use Shared Clusters as your default compute!

2. If Shared Clusters don’t work, use single-user clusters!

3. Develop and deploy using the same access mode!

Learn more at the summit!

Mobile App

Tells us what you think

- We kindly request your valuable feedback on this session.
- Please take a moment to rate and share your thoughts about it.
- You can conveniently provide your feedback and rating through the Mobile App

What to do next?

- Discover more related sessions in the mobile app!
- Visit the Demo Booth: Experience innovation firsthand!
- More Activities: Engage and connect further at the Databricks Zone!

Get trained and certified

Visit the Learning Hub Experience at Moscone West, 2nd Floor

- Take complimentary certification at the event; come by the Certified Lounge
- Visit our Databricks Learning website for more training, courses and workshops! databricks.com/learn

THANK YOU