



# Data + AI SUMMIT 2021

DAIS 2021で発表された新機能のご紹介

# Data + AI Summit 2021で発表された新機能

## 1. Delta Sharing 2021/10からPublic Preview

- セキュアなデータ共有のためのオープンプロトコル

## 2. Unity Catalog 2021/10からPrivate Preview

- データ、モデルに対するシンプルなガバナンス

## 3. Databricks SQL Gated Public Preview

- データレイクにおけるBIの実現

## 4. Delta Live Tables Public Preview

- 簡単にDelta Lakeの高信頼ETLを実現

## 5. Feature Store Public Preview

- データ、MLOpsと協調設計された特徴量ストア

## 6. AutoML Public Preview

- 機械学習モデル開発の自動化に対するガラスボックスアプローチ

# Data + AI Summit 2021で発表された新機能



<b>Delta Sharing</b>	○	○	○	○
<b>Unity Catalog</b>	○	○	○	○
<b>Databricks SQL</b>	○	○		○
<b>Delta Live Tables</b>	○			○
<b>Feature Store</b>	○		○	○
<b>AutoML</b>			○	

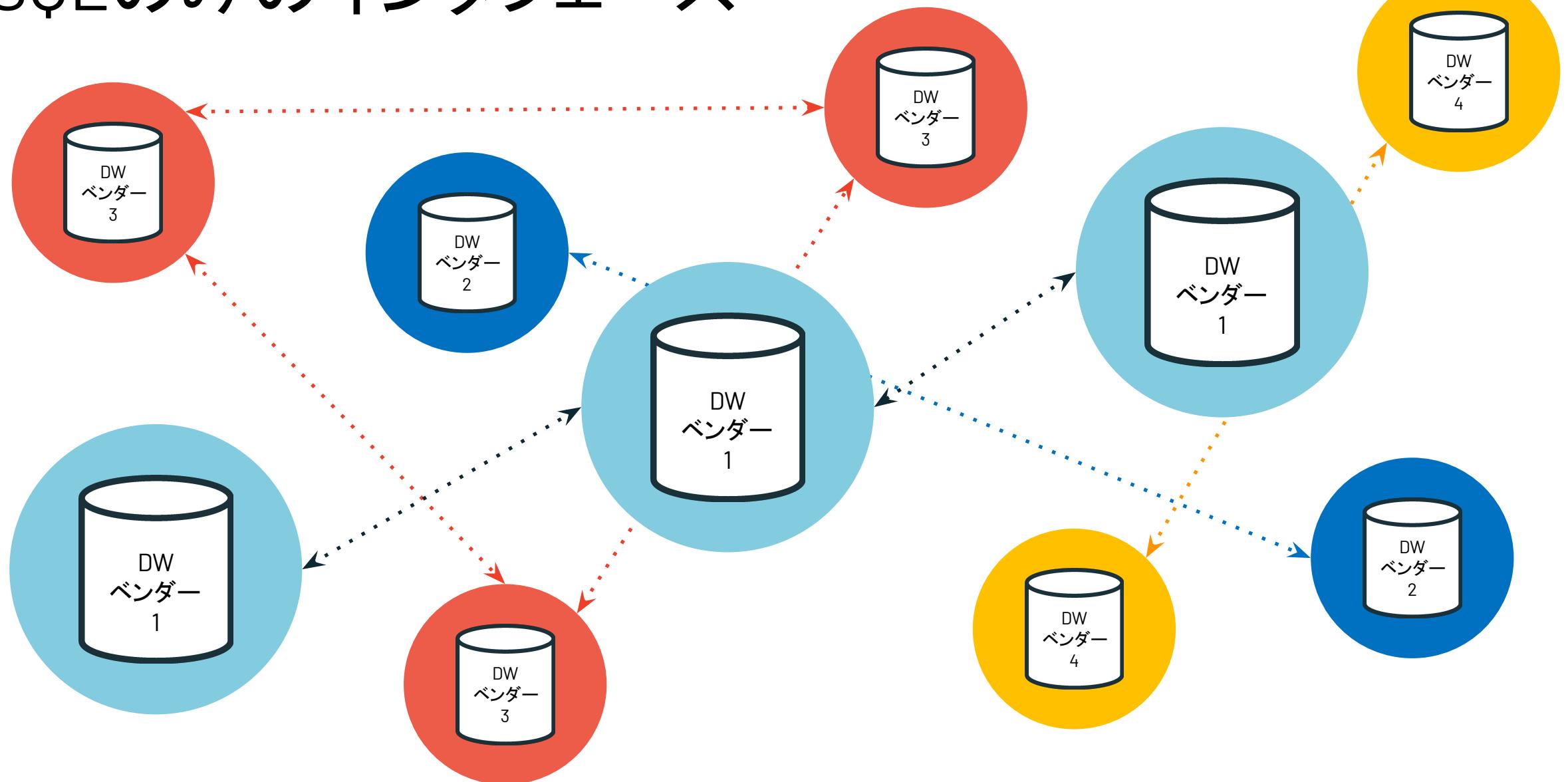


# Delta Sharing

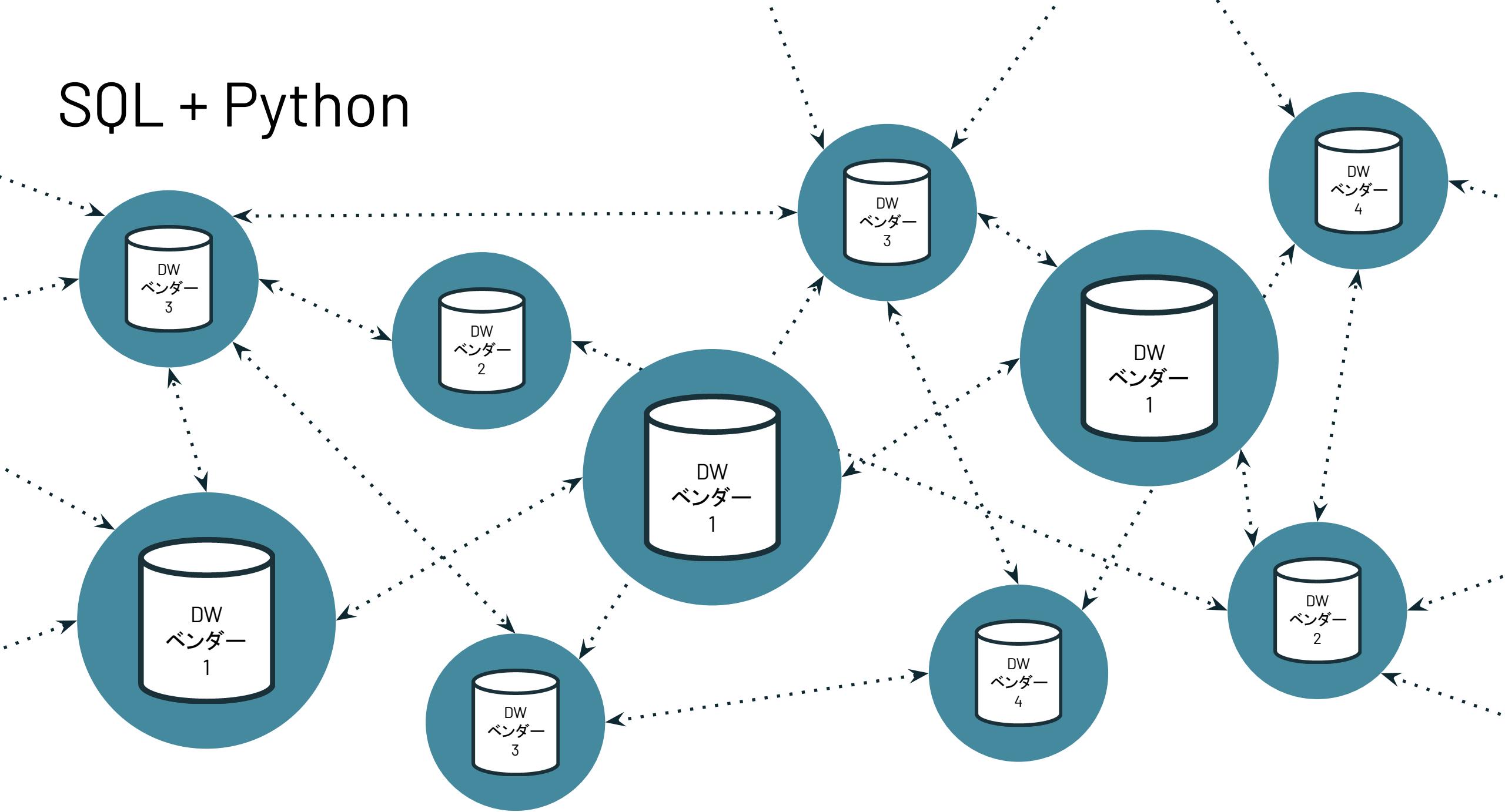
セキュアなデータ共有のためのオープンプロトコル

データは企業の垣根を越えて流れ  
いくべきです

# SQLのみのインターフェース



# SQL + Python

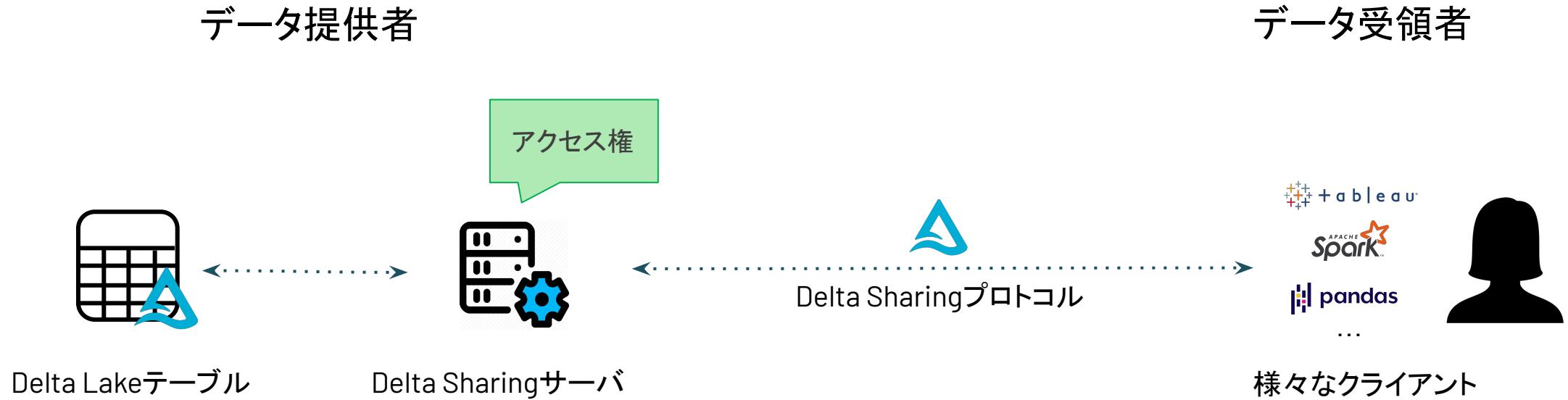


オープンな未来はデータ共有に対する  
オープンなアプローチを必要とします

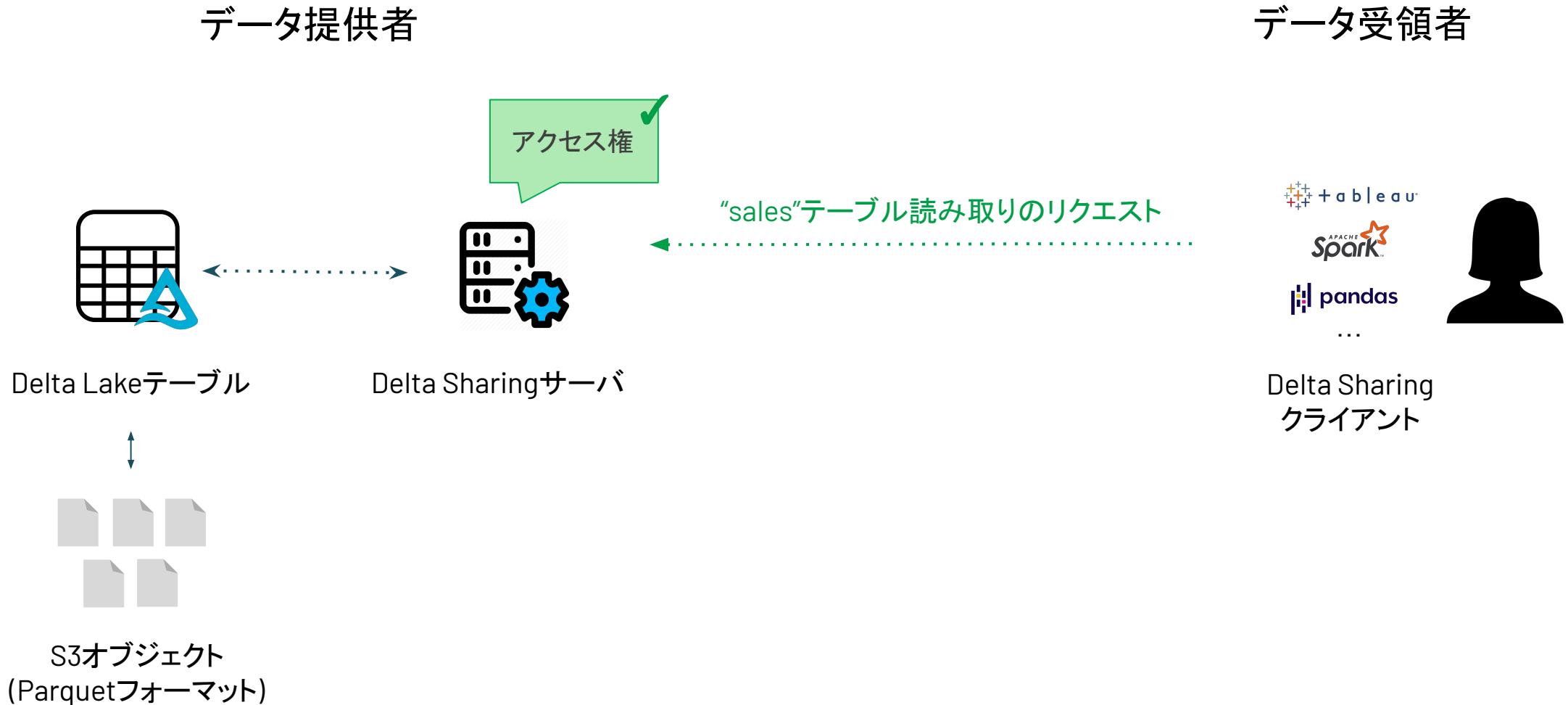
# Delta Sharingの目指すゴール

1. データレイク/レイクハウスにある既存のライブデータを(コピーせずに)共有
2. 既存のオープンデータフォーマットによる幅広いクライアントのサポート
3. 強力なセキュリティ、監査、ガバナンス
4. 大規模データセットに対する効率的なスケーラビリティ

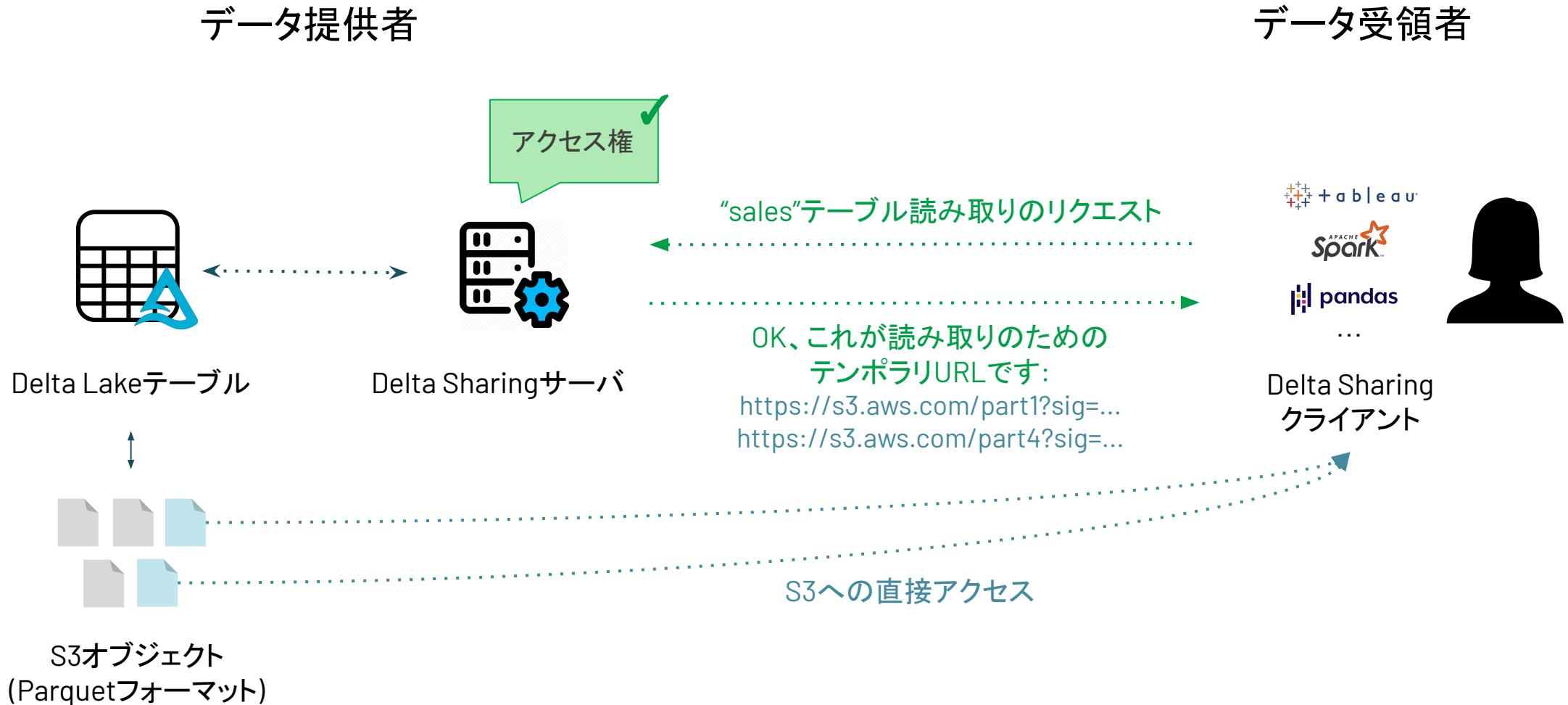
# 動作原理



# 内部処理



# 内部処理



# デザインのメリット

- データ提供者は容易に特定のバージョン、特定パーティションのみを共有できます
- テーブルはライブで共有されACIDトランザクションによって更新されます
- Parquetを読み込むあらゆるクライアントがDelta Sharingをサポートします
- S3/ADLS/GCSを用いることで、転送が高速、安価、高信頼性、高並列性なものとなります

# Delta Sharingのエコシステム

## オープンソースクライアント



## 商用クライアント



## データ提供者



# Delta Sharingを始めましょう

Delta SharingはDelta Lake 1.0の一部です！

リファレンスサーバー、Pandas、Spark、Rustのコネクターをオープンソース化しました。まもなく、多くのベンダーがコネクターをリリースします。

Get started: [delta.io/sharing](https://delta.io/sharing)





# Unity Catalog

データ、モデルに対するシンプルなガバナンス

# データレイクのガバナンス管理は大変です

ユーザー



ファイルベースの権限管理

- user1 は /pages/ を読める
- user2 は /users/ を読める
- user3 は /users/us/ ...

データ (S3/ADLS/GCS上のファイル)

/dataset/pages/part-001  
/dataset/pages/part-002  
/dataset/users/uk/part-001  
/dataset/users/uk/part-002  
/dataset/users/us/part-001



メタデータ  
(e.g. Hive メタストア)



テーブル & ビュー

SQLデータベース



amazon  
REDSHIFT



BigQuery

機械学習モデル



ユーザーにテーブルの特定の列、行だけを見せたい場合にはどうしたら？

データレイアウトを変更したらどうなる？

ガバナンスのルールが変わったら？

データと同期が取れないかも？

異なるガバナンスモデル

異なるガバナンスモデル

# Databricks Unity Catalog

ユーザー

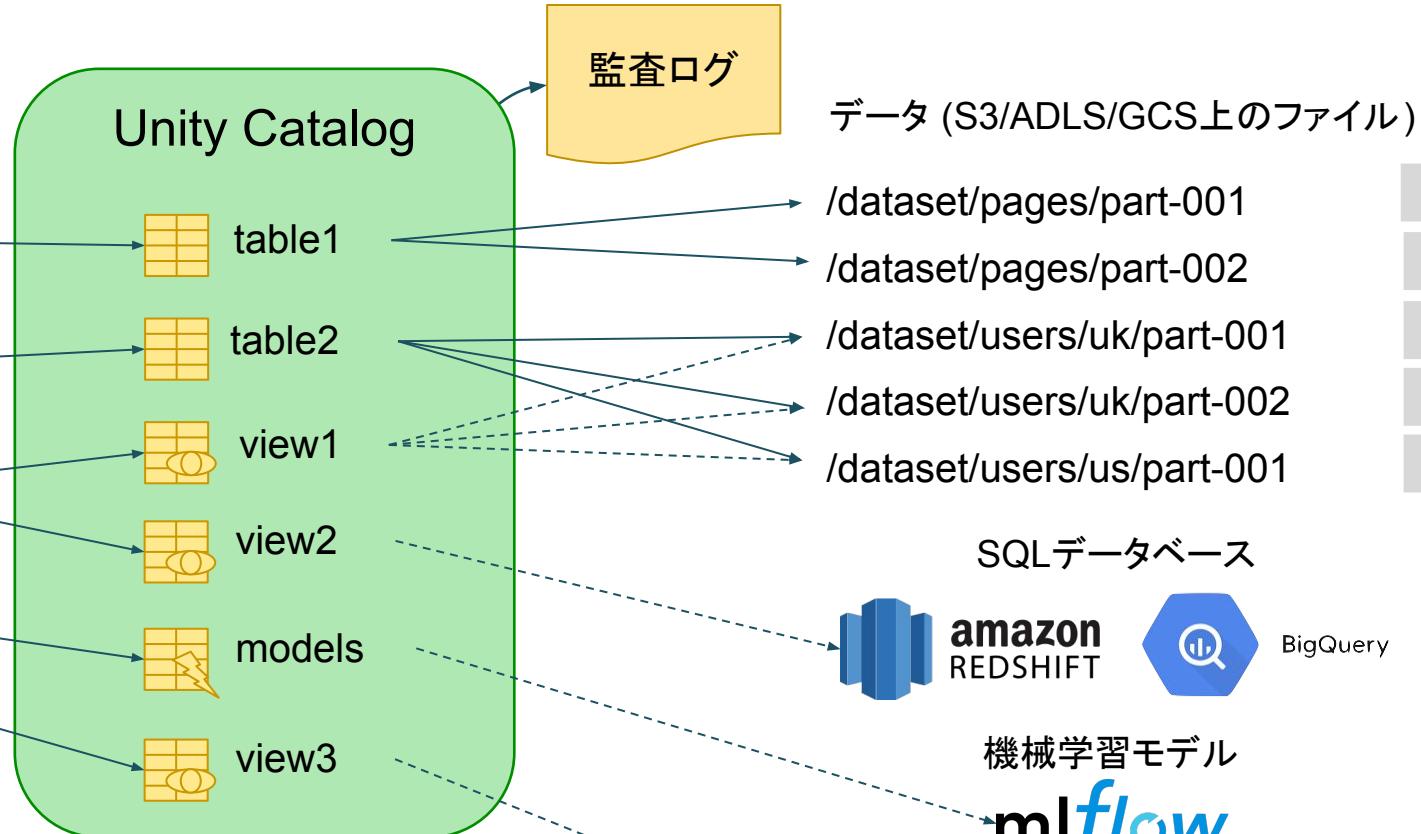


テーブル、行・列、ビューレベルでの  
アクセス権設定

SQLのgrantでアクセス権を設定

すべてのアセットに対して統一されたアクセス権管理

集中管理による監査の実現



# Unity Catalogの使用

```
CREATE TABLE iot_events
```

(新規テーブル)

**OR**

```
CREATE EXTERNAL TABLE iot_events LOCATION s3:/... (既存データ)  
WITH CREDENTIAL iot_iam_role
```

```
GRANT SELECT ON iot_events TO engineers
```

```
GRANT SELECT(date, country) ON iot_events TO marketing
```

# ビューベースのアクセスコントロール

```
CREATE VIEW aggregate_data AS  
SELECT date, country, COUNT(*) AS num_events  
FROM iot_events
```

```
GRANT SELECT ON aggregate_data TO business_analysts
```

任意のユーザー指定のビューに対するアクセス権の許可

# 属性ベースのアクセスコントロール

```
CREATE ATTRIBUTE pii
```

```
ALTER TABLE iot_events ADD ATTRIBUTE pii ON email  
ALTER TABLE users ADD ATTRIBUTE pii ON phone
```

...

```
GRANT SELECT ON DATABASE iot_data  
HAVING ATTRIBUTE NOT IN (pii)  
TO product_managers
```

piiとタグ付けされた全てのカラムにアクセス権を設定

# Catalog UI

unity\_catalog / city\_data /

## Firehose

[GDPR](#) [Edit](#)

**Description** [Edit](#)  
This table contains raw events from across the LOC platform. Use this table to find all reported game-play events.

**Recommendations** [Edit](#)  
Optimizing metric may improve performance. [Learn more](#). Table\_name has not been vacuumed in 90 days. [Learn more](#).

**Owners** SO AA PA JS      **Frequent users** JD BT SB AD

**ABAC Policies** [Edit](#)  
 PII  
 Cost  
 Inventory

**Top Queries**  
Champions stats 1H 2021  
Gameplay analysis Q2 2021  
Gameplay hours LOC semi-finals  
Purchase prediction model 2021

**Statistics**  
Format Delta   
Updated 5 hours ago BT  
Created 1 year ago SO  
Size 500 GB

**Schema**

ProductID	integer
Add description	Add Tag

Status	string
Add description	Add Tag
Inventory	
Cost	

PricingTier	float
Add description	Add Tag

DistributionTier	string
Add description	Add Tag

AccountInfo	string
Add description	Add Tag
PII	

**Lineage**

```
graph LR; inbndcall --> Firehose[Firehose]; Firehose --> disp_rec[disp_rec]; Firehose --> inventory[inventory]
```

**Privileges** [Edit](#)

User / Group	Permissions	Modified
bi_team	Select, Modify, Manage	2021-05-20 15:56
dev	Select	2021-05-12 10:25

**Data Sample**

ProductID	Status	PricingTier	DistributionTier	AccountInfo
135018	active	Enterprise	P1	15

[Open in SQL Editor](#) [Grant Privileges](#)

## データ整理

必要なデータの定義、加工、アクセス権設定が可能

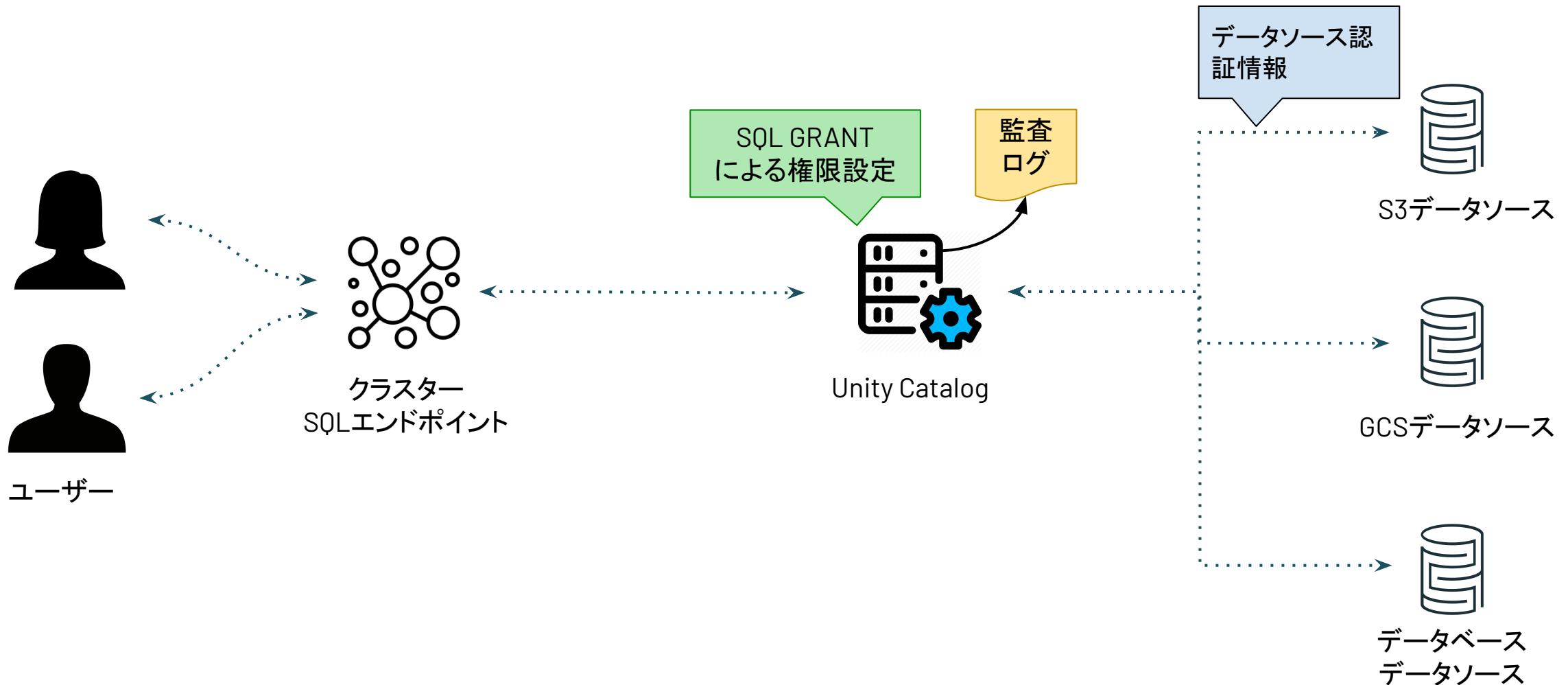
## データ探索

欲しいデータアセットを探して、データにアクセス

## データリネージュ

データアセットのリネージュ(データの繋がり)を可視化

# Unity Catalogの動作原理



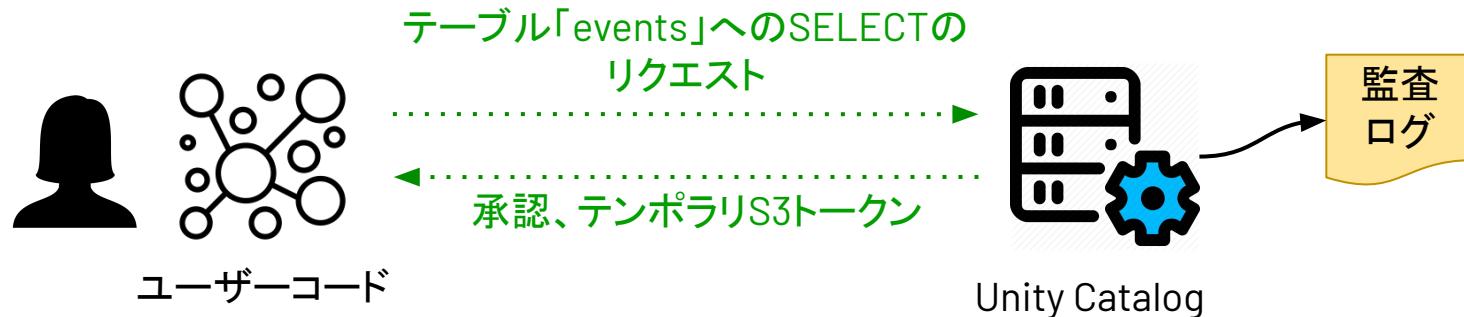
# アクセス強制

あらゆるタイプのユーザーコードに統一されたセキュアなアクセスを強制

- (ライブラリに非依存)SQL、Python、Scala、Java、R

ユーザーコードは生データ、認証情報(IAMロールなど)を取得しません

- Databricksはテーブルをフィルタするか、ユーザーコードに対して特定のデータファイルに対する直接アクセスを許可する一時的に有効なトークンを提供します。



# テーブル以外のアセット管理

```
GRANT EXECUTE ON MODEL fraud_ranking TO engineers
```

```
GRANT EXECUTE ON MODELS HAVING ATTRIBUTE (eu_data)  
TO eu_product_managers
```

同様の属性ベースのシステムを用いてMLモデル、ファイルなどのガバナンスを実現

# オープンなエコシステム

Unity Catalogは既存のカタログ、ストレージシステムと動作します

- Hiveメタストア、S3、ADLS、GCSなどにある既存データのマウント
- ImmutaやPrivaceraによって製品横断でのポリシー管理

Databricks外からのオープンなアクセス: JDBC/ODBC、Delta Sharing

アクセス権設定のためのオープンかつ標準化されたインターフェース: ANSI SQL DCL

# まとめ

Unity Catalogは、シンプルかつ標準化されたインターフェース(ANSI SQL)を通じて、レイクハウス、ML、データにきめ細かい、集中管理されたガバナンスをもたらします。

クラウド、ストレージシステム横断で既存データと連携します。

アップデートを確認するためにはウェイティングリストにご登録ください:

[databricks.com/unity](https://databricks.com/unity)



# Databricks SQL

データレイクにおけるBIの実現



データ  
エンジニアリング

BI & SQL

リアルタイム

データサイエンス &  
機械学習

データ管理 & ガバナンス

オープンなデータレイク



**databricks**® レイクハウスプラットフォーム

シンプル・オープン・コラボレーティブ



構造化データ



準構造化データ



非構造化データ



ストリーミング



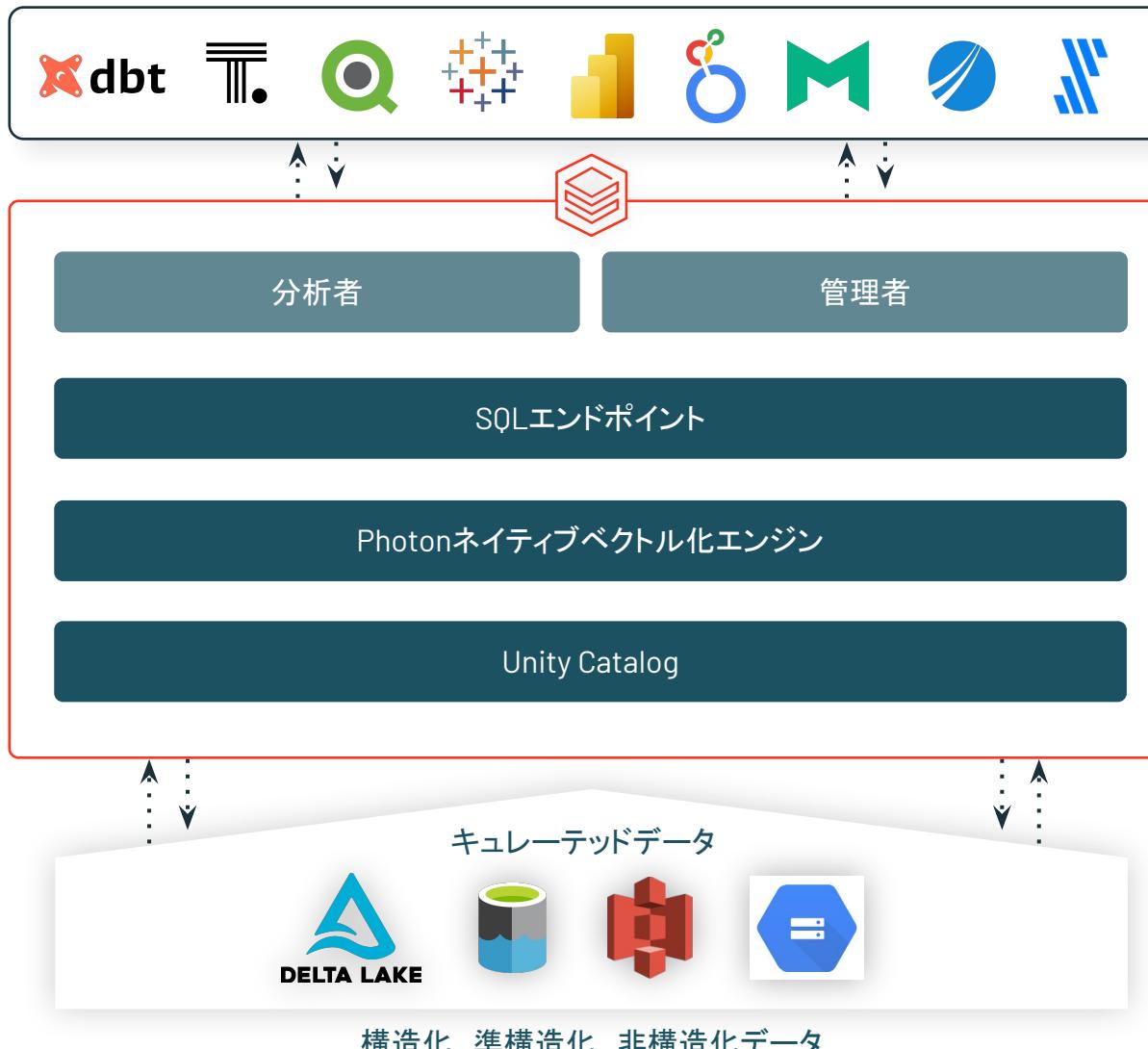
# Databricks SQL: データレイクにおけるBI



 **databricks**® レイクハウスプラットフォーム

シンプル・オープン・コラボレーティブ

# Databricks SQLの内部



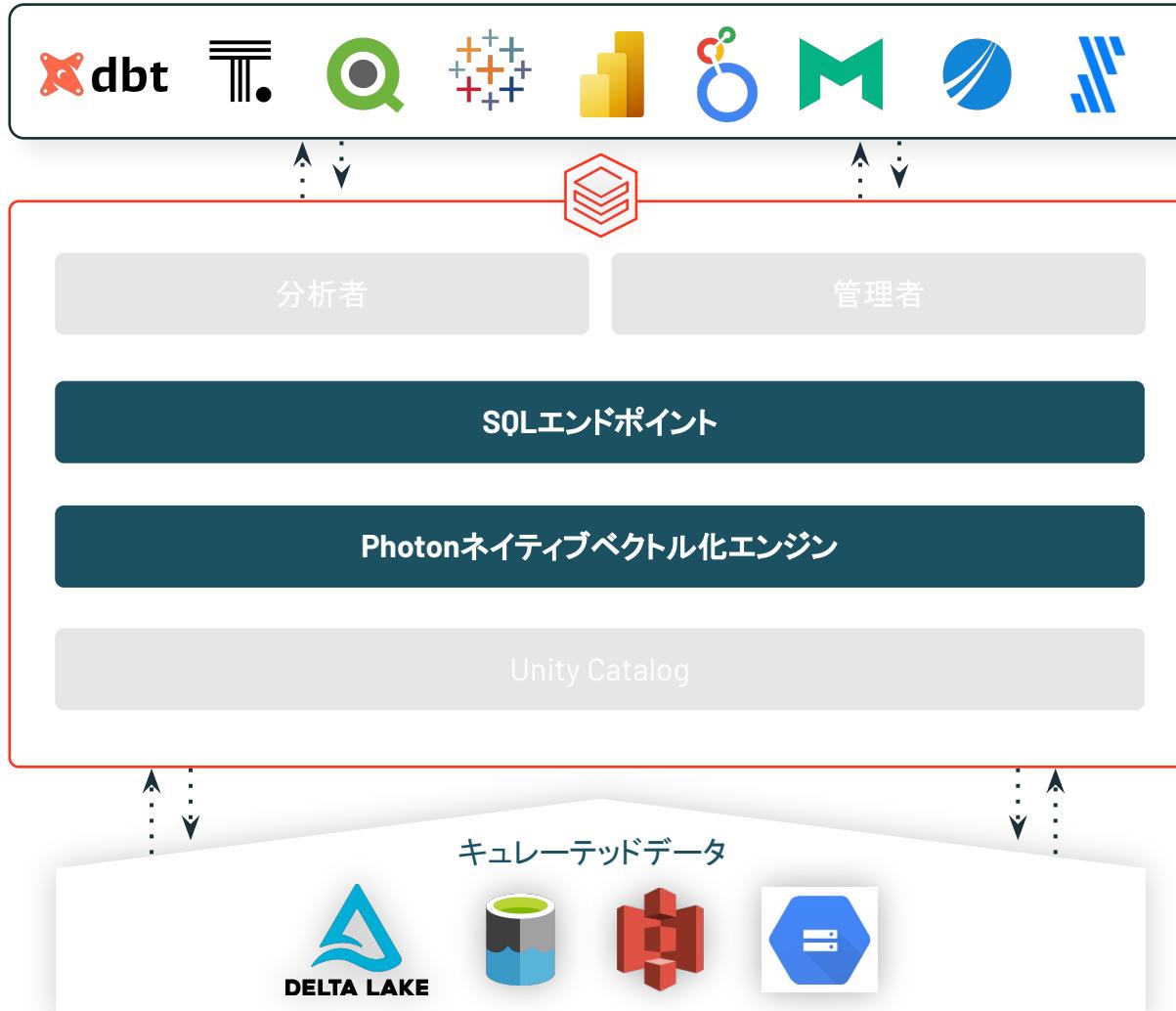
使いやすさ

リアルワールドの性能

集中管理のガバナンス

レイクハウス基盤上の  
オープンかつ高信頼のデータレイク

# リアルワールドの性能

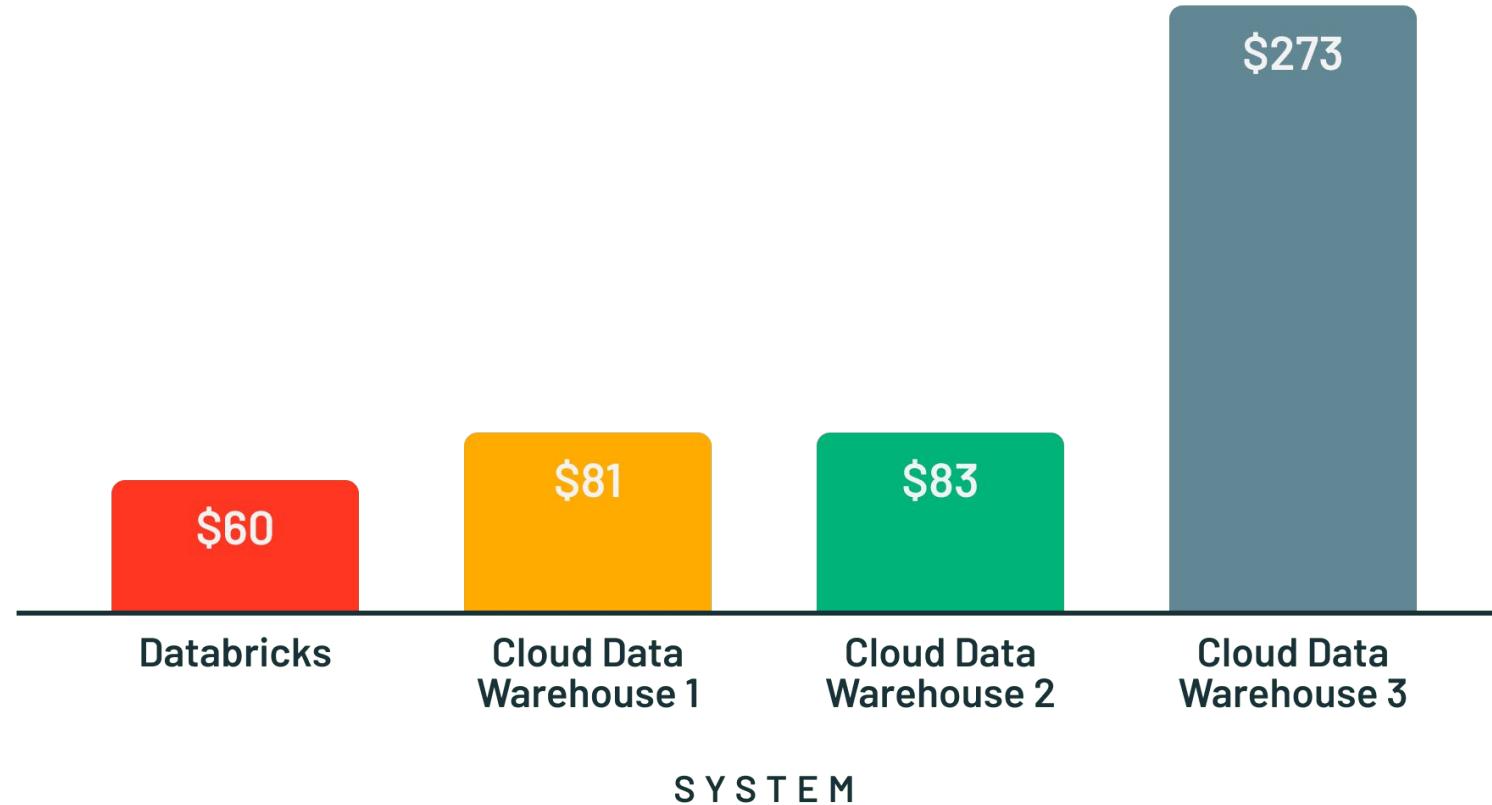


全てのクエリーに対して  
高速、予測可能な性能を提供

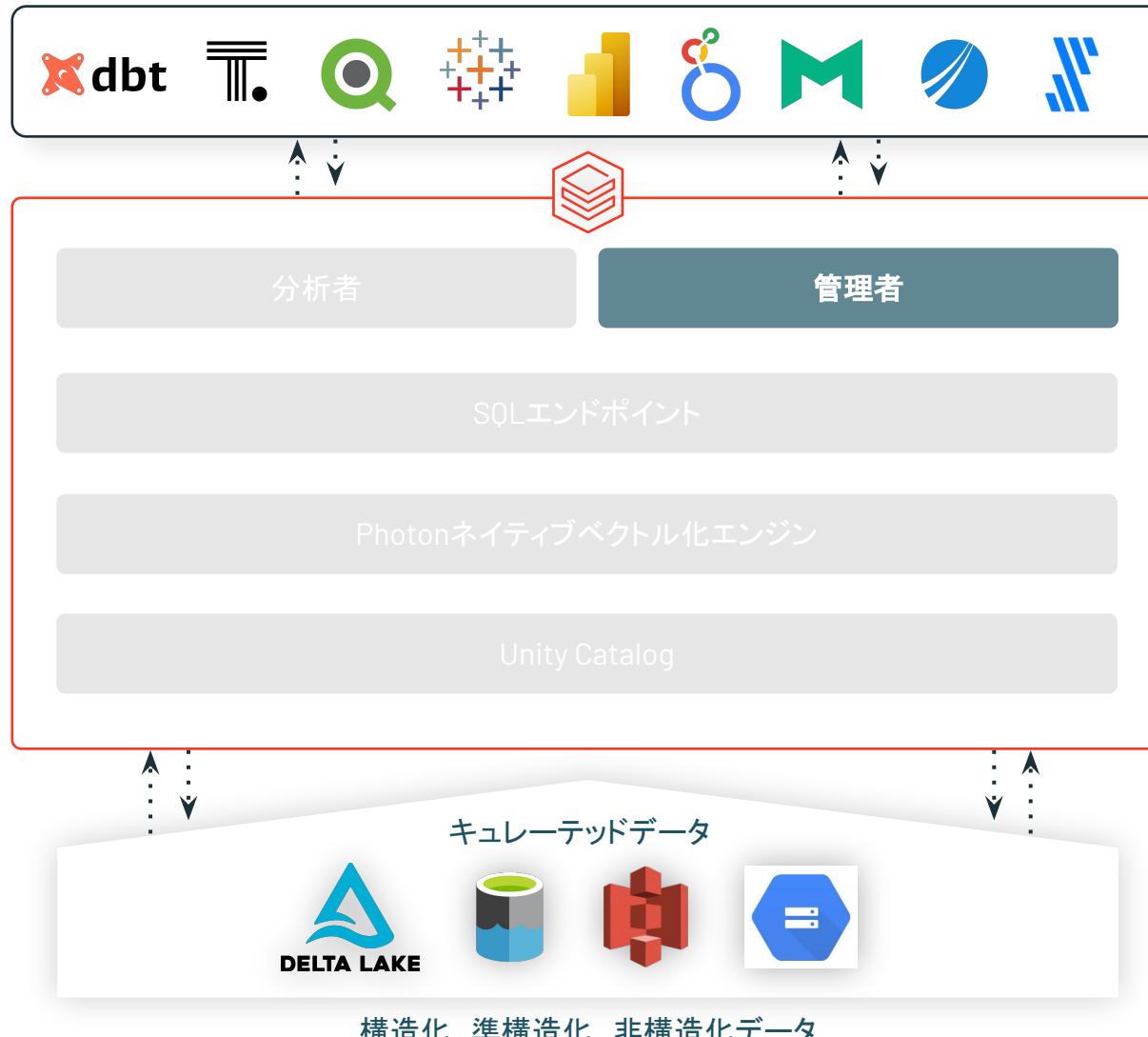
# 大規模クエリーの性能

**30TB TPC-DS** における性能対価格

低いほど望ましいものとなります



# 管理の簡素化

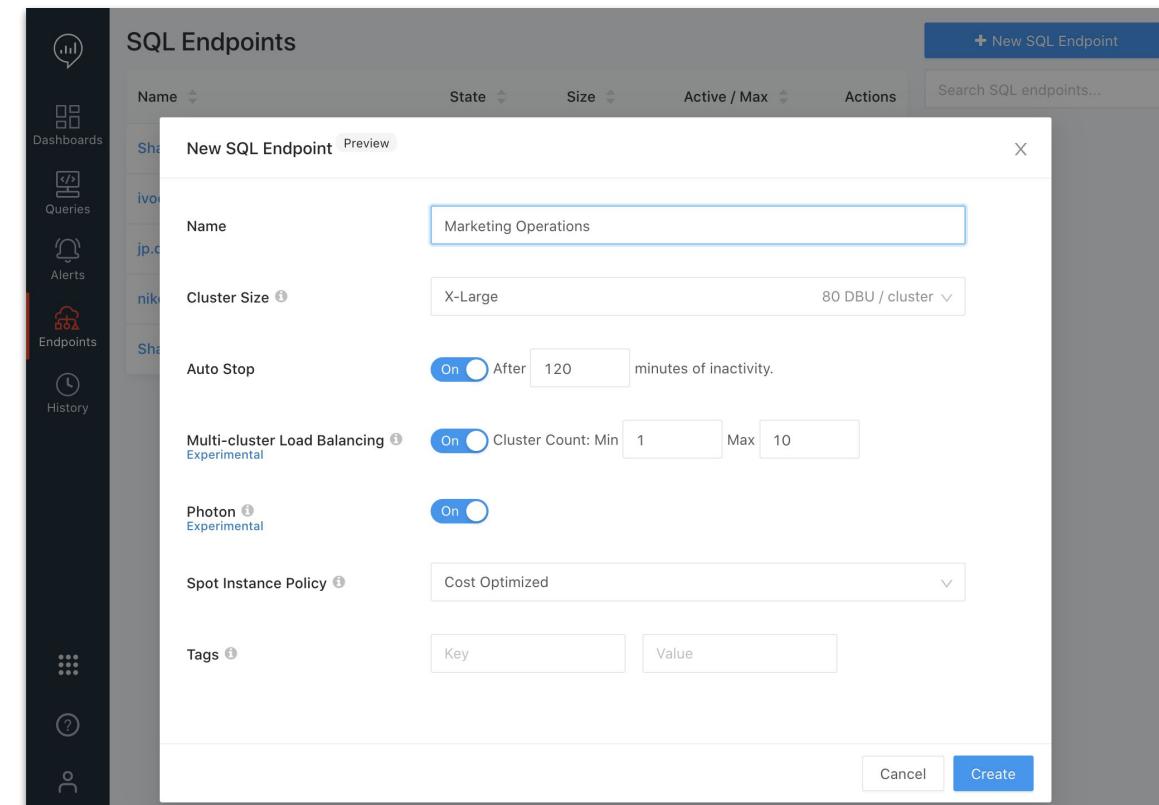


SQL、BIワークフローに対する  
大規模リソースの容易なセットアップ  
およびモニタリング

# シンプルな管理

すぐにスタートできます

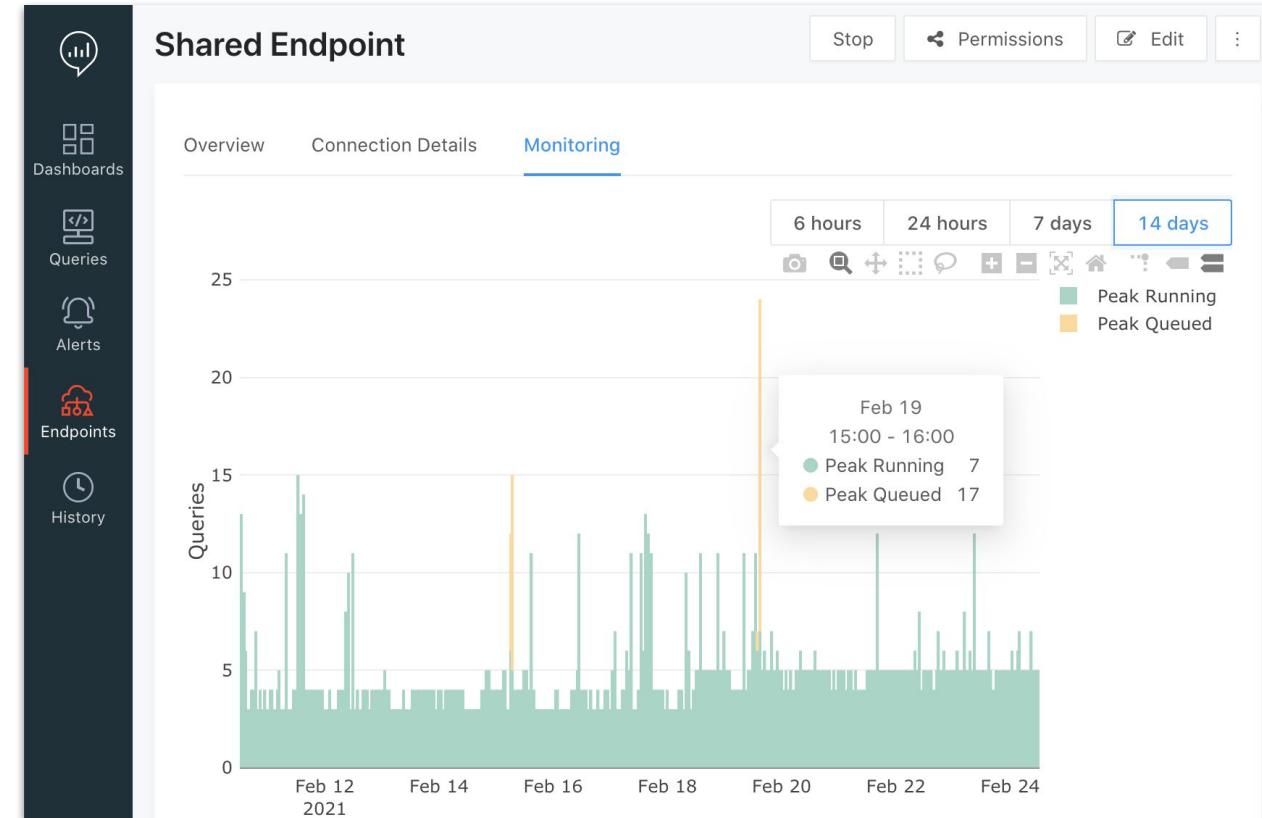
- ・ シンプルなTシャツサイズのクラスター
- ・ 同時実行数に合わせたオートスケーリング
- ・ SQLワークロードにチューニングされたVM、設定



# シンプルな管理

## ワークフロー、使用量の理解

- ・ ビルトインのエンドポイントモニタリング
- ・ 特定期間におけるドリルダウン



# シンプルな管理

## 迅速なトラブルシューティング

- ・ 集中管理されるクエリー履歴
- ・ クエリー実行時間のブレークダウン
- ・ クエリー実行の詳細

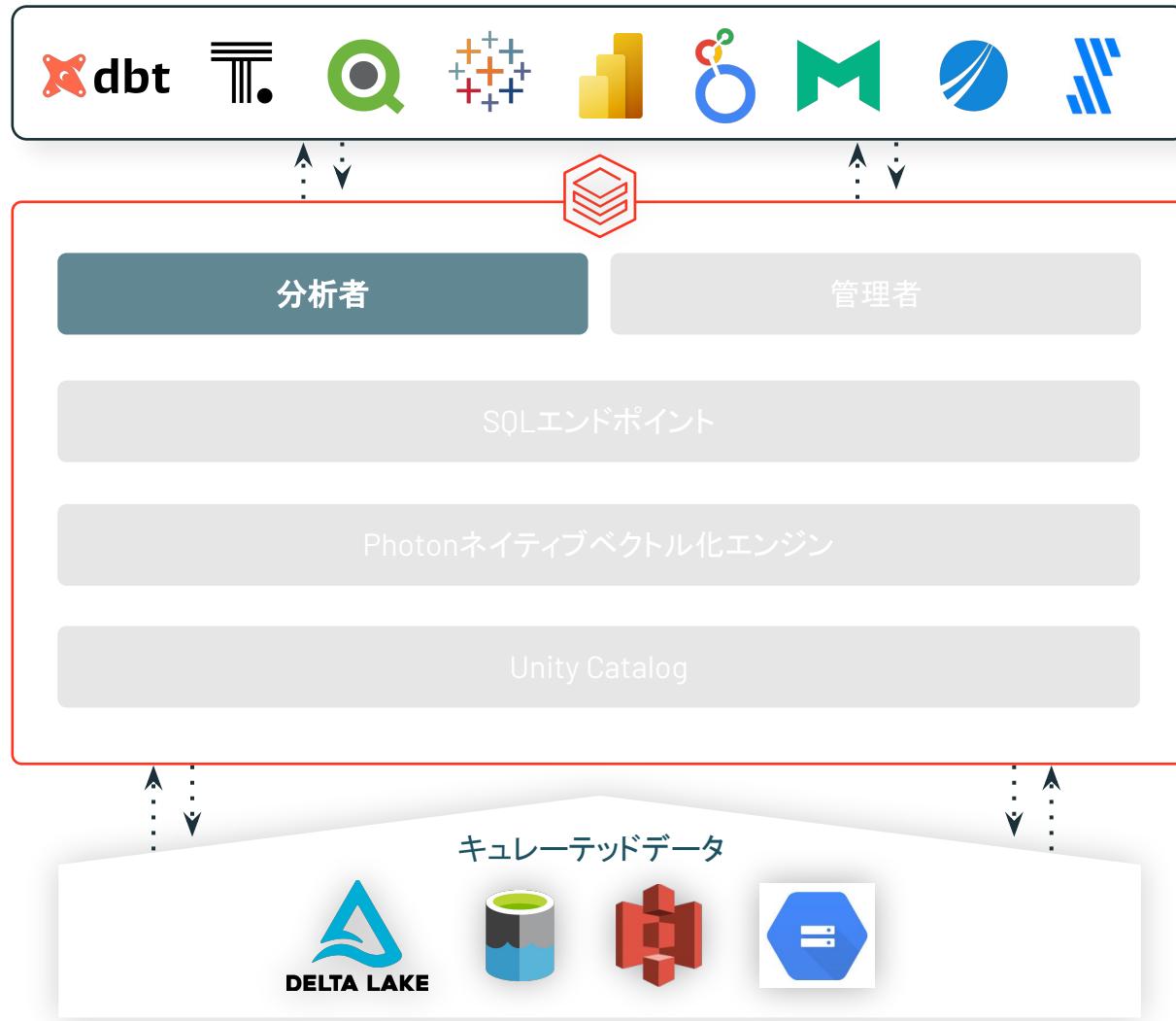
The screenshot shows the Databricks web interface. On the left is a dark sidebar with navigation icons: Create, Queries, Dashboards, Alerts, Endpoints, and History (which is highlighted with a red bar). The main area has two tabs: 'Query History' and 'Query Details'. The 'Query History' tab is active, showing a table of recent queries. The table columns are 'Query', 'SQL Endpoint', and 'Started At'. The queries listed are:

Query	SQL Endpoint	Started At
WITH top_exposure AS ( SELECT concept_name AS drug_name , drug_conce...	Shared Endpoint	2021-04-2...
SELECT count(1) as exposure_occurrence_count , d.drug_type_concept_i...	Shared Endpoint	2021-04-2...
USE omop600 -- user_id: 5724191577397205	Shared Endpoint	2021-04-2...
USE omop600 -- user_id: 5724191577397205	Shared Endpoint	2021-04-2...
SHOW DATABASES -- user_id: {}	Shared Endpoint	2021-04-2...
select * from chocolate.reviews	Shared Endpoint	2021-04-2...
select avg(Rating), Company_Location from chocolate.reviews where Sp...	Shared Endpoint	2021-04-2...
select * from chocolate.reviews	Shared Endpoint	2021-04-2...

Below the table are dropdown filters for 'Me (fuatcan.efeoglu@databricks.com)' and 'Last 14 days'. The 'Query Details' tab is selected, showing a breakdown of the execution time for the first query. The details are as follows:

Duration	4.03 s	100%
Compilation	1.14 s	28%
Execution	2.89 s	72%
Result fetching	3 ms	0%
Rows returned	20	
IO		
Rows read	35,850,380	
Bytes read	195.14 MB	
Bytes read from cache	100 %	
Bytes written	0 bytes	
Files & Partitions		
Files read	2	
Partitions read	0	

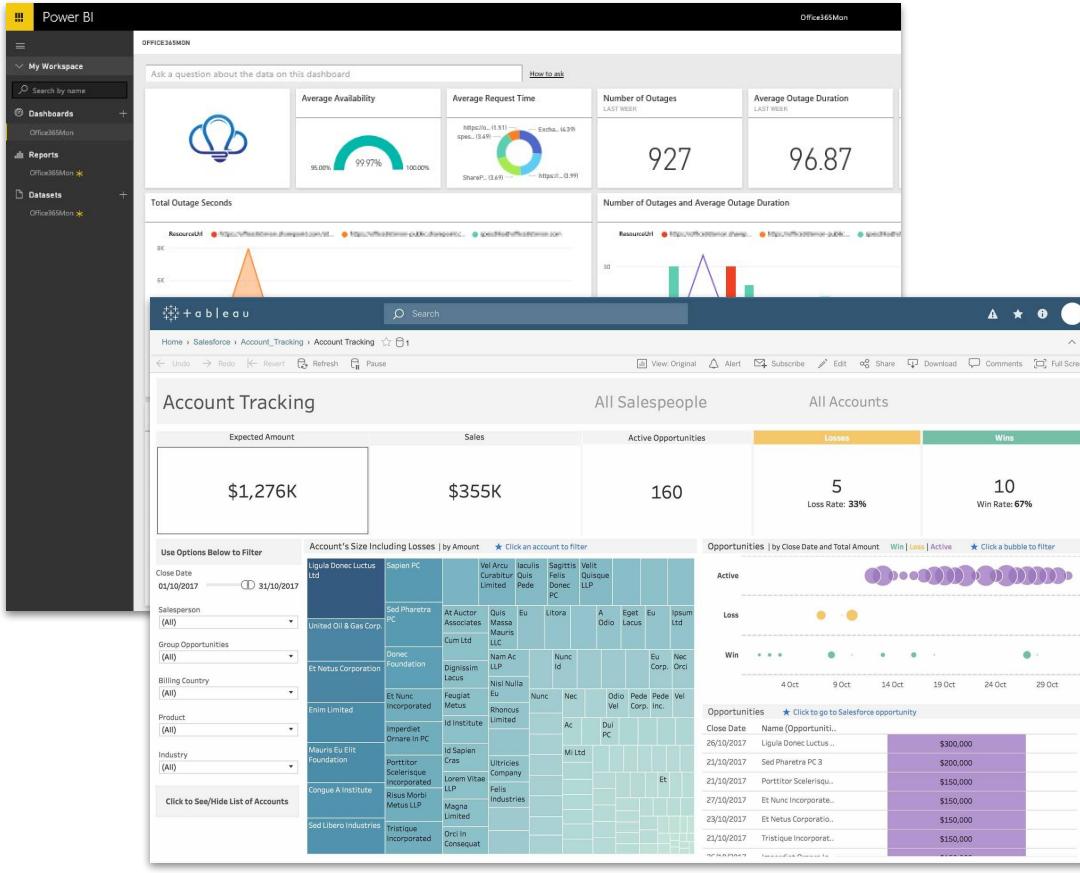
# 素晴らしい分析者体験



お好きなツールを用いたデータレイク上のSQL  
分析、BI

# 拡大するエコシステム

## 分析者の生産性を改善するために



& more

# ファーストクラスのSQL開発エクスペリエンス

## SQLによるデータレイクでのシンプル、迅速かつアドホックな探索的分析の実現

### 開発

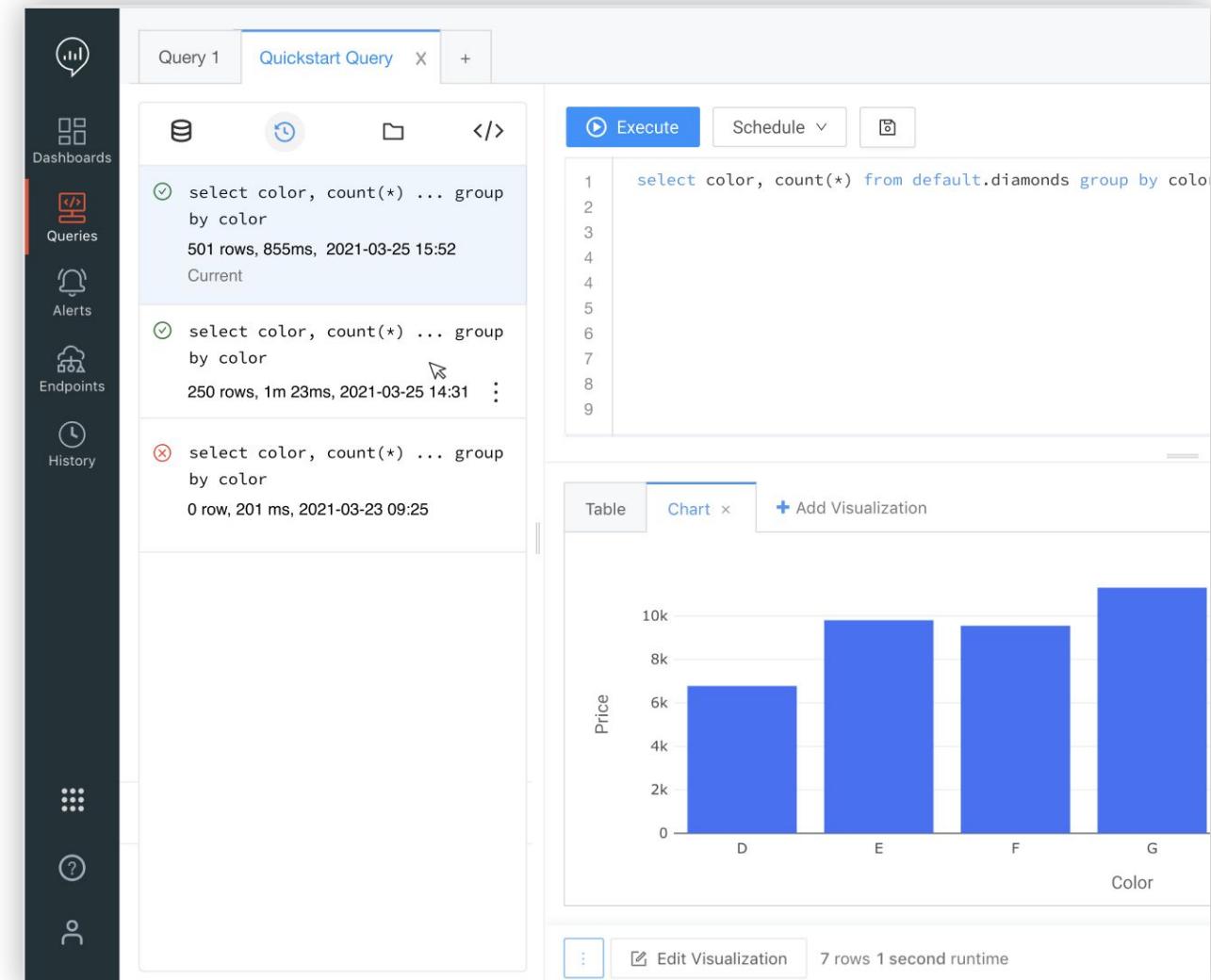
- クエリータブ
- ドラフト & “前の時点から再開”
- コマンド履歴
- コンテキストベースのオートコンプリート

### トラブルシュート

- クエリーの進捗状況
- エラーのハイライト
- 実行時間のブレークダウン

### コラボレーション

- メール送信のスケジュール処理
- アクセス権設定





Databricks SQLを使い始めるには  
2021/6からパブリックプレビュー:  
[databricks.com/try](https://databricks.com/try)



# Delta Live Tables

簡単にDelta Lakeの高信頼ETLを実現



# databricks

## レイクハウスプラットフォーム

シンプル オープン コラボレーティブ

データエンジニアリング

BI & SQL  
アナリティクス

リアルタイムデータ  
アプリケーション

データサイエンス &  
機械学習

データマネジメント & ガバナンス



DELTA LAKE

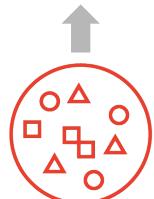
オープンなデータレイク



構造化データ



準構造化データ



非構造化データ

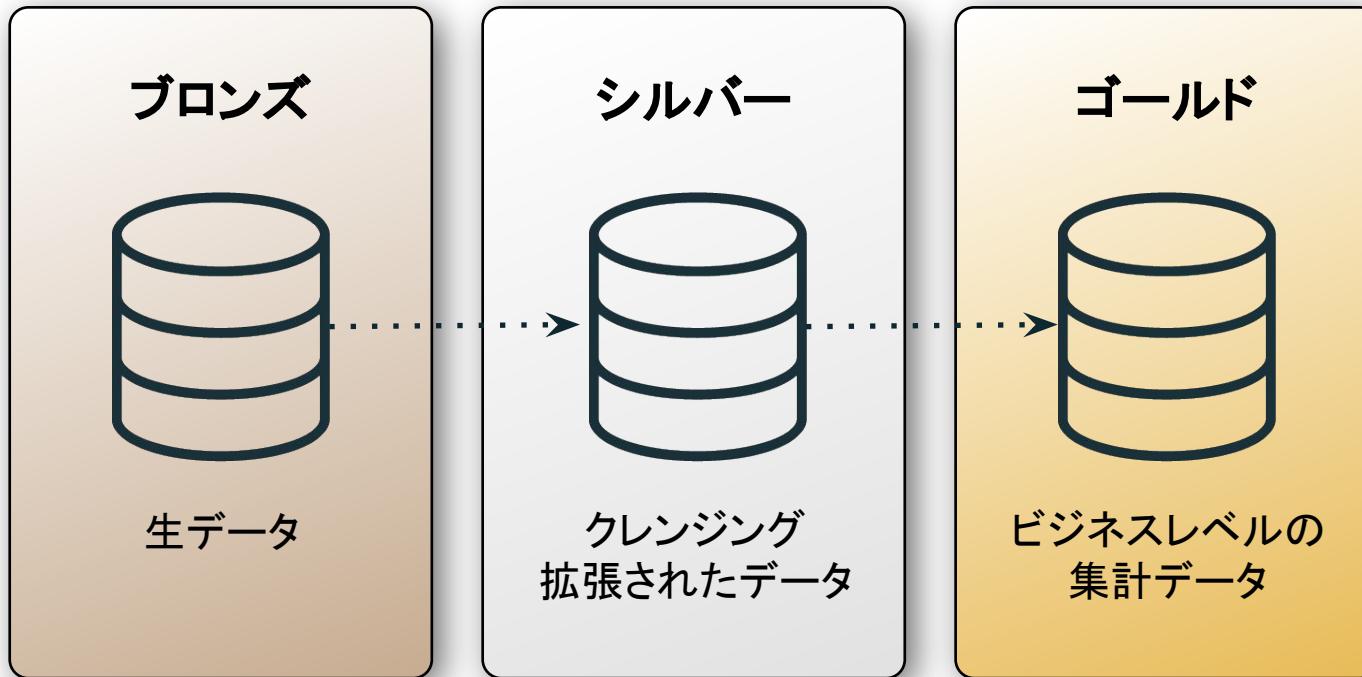


ストリーミング



Google Cloud

# Delta Lakeはレイクハウスの基盤と言えます

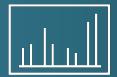
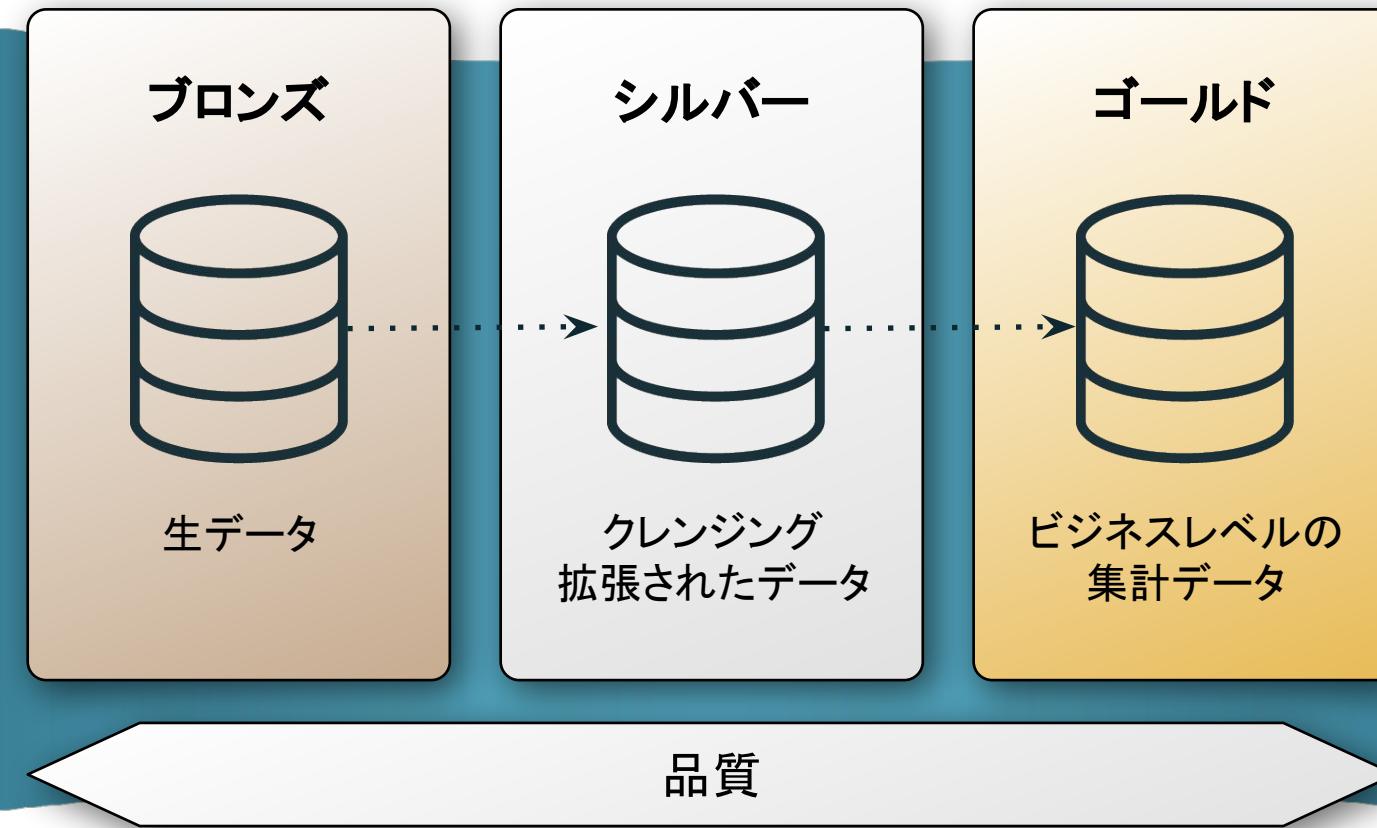


# ETLを通じてレイクハウスの基盤を築きます



CSV,  
JSON, TXT...

Data Lake



Streaming  
Analytics



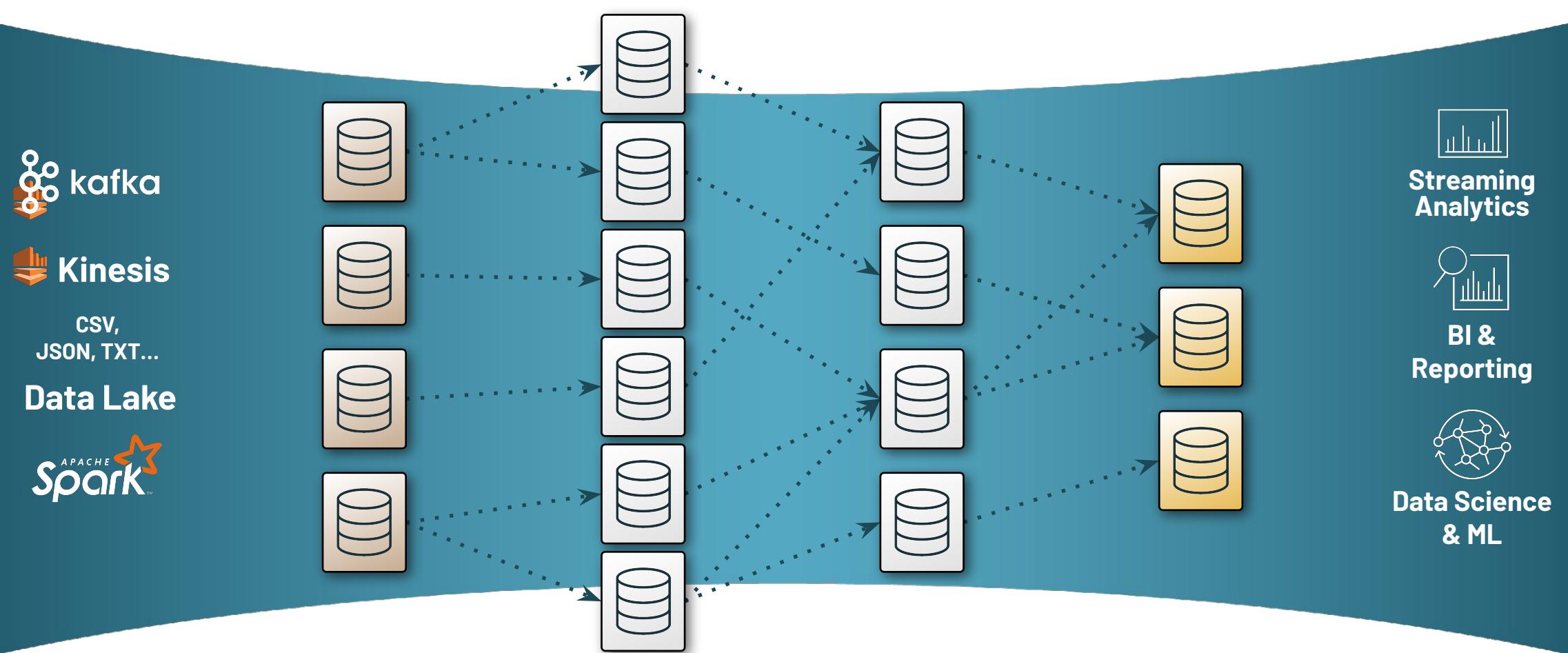
BI &  
Reporting



Data Science  
& ML

# でも、現実はそんなにシンプルなものではありません

大規模データに対してデータの品質と信頼性を維持するのは複雑で不安定なものになります



# 大規模ETLは複雑かつ不安定なものです

## 複雑な パイプライン の開発

依存性を構築、維持  
することが困難

バッチとストリーム  
処理を切り替えるのが  
困難

## 貧弱な データ品質

データ品質の監視、  
強制が困難

データのリネージュを追  
跡できない

## パイプライン オペレーションが困 難

詳細なデータレベルでの  
貧弱な観察可能性

エラーハンドリングとリカ  
バリが面倒



# Delta Live Tablesのご紹介

新鮮かつ高品質データを構築、管理するシンプルな方法

## パイプラインの容易な開発、維持

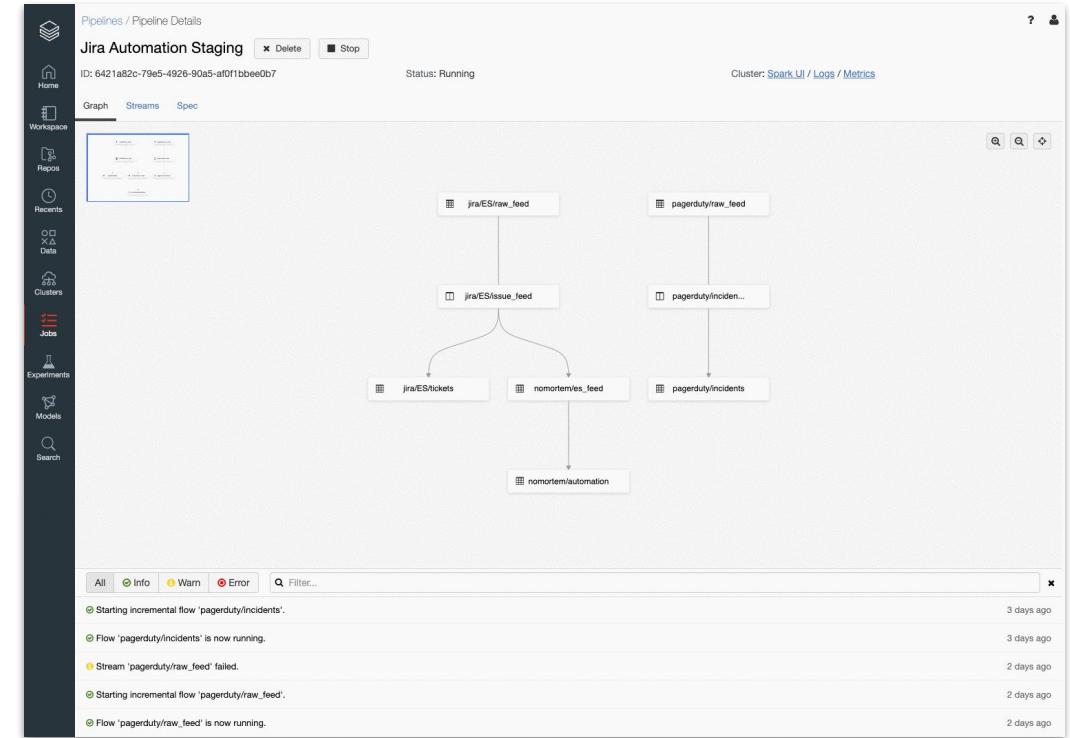
データパイプライン(バッチ、ストリーミング)を構築、管理するための記述ツール

## 自動テスト

ビルトインの品質管理、データ品質モニタリング

## 簡素化されたオペレーション

パイプラインオペレーションに対するディープな可視化を通じた自動エラーハンドリング

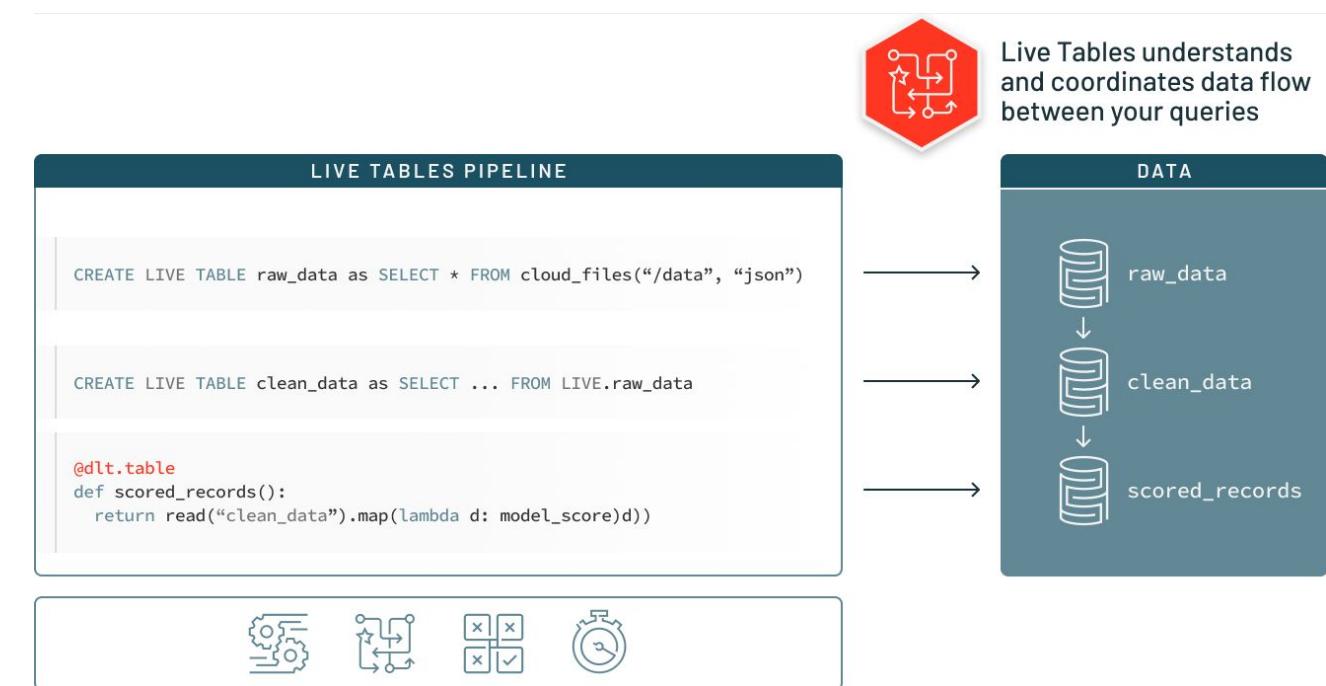


# データパイプラインの容易な構築、維持

ビジネスロジックとテーブルの依存関係を含めて、記述的にデータパイプラインを構築します

構造化/非構造化データをバッチ、ストリーミングで実行します

環境に渡ってETLパイプラインを再利用できます



# データに対する信頼性

Deltaのエクスペクテーションにより不正なデータがテーブルに流れ込むことを防ぎます

事前定義されたエラーポリシー(失敗、欠損、警告、データ検疫)によってデータ品質エラーを回避、対策します

長期にわたるデータ品質のトレンドをモニタリングします

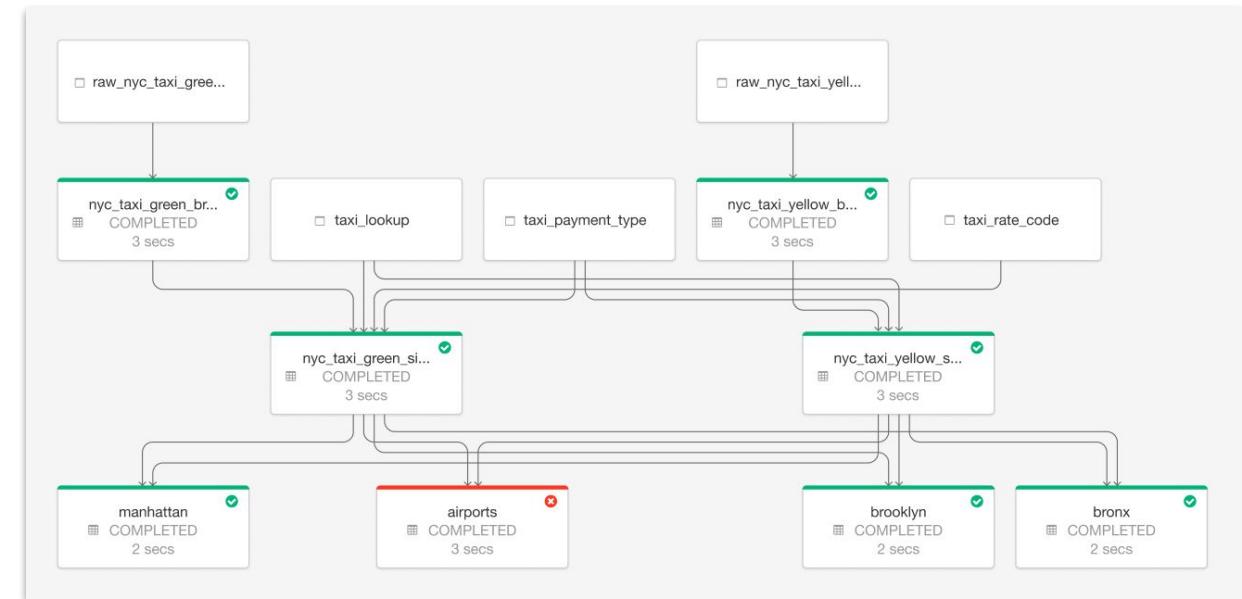


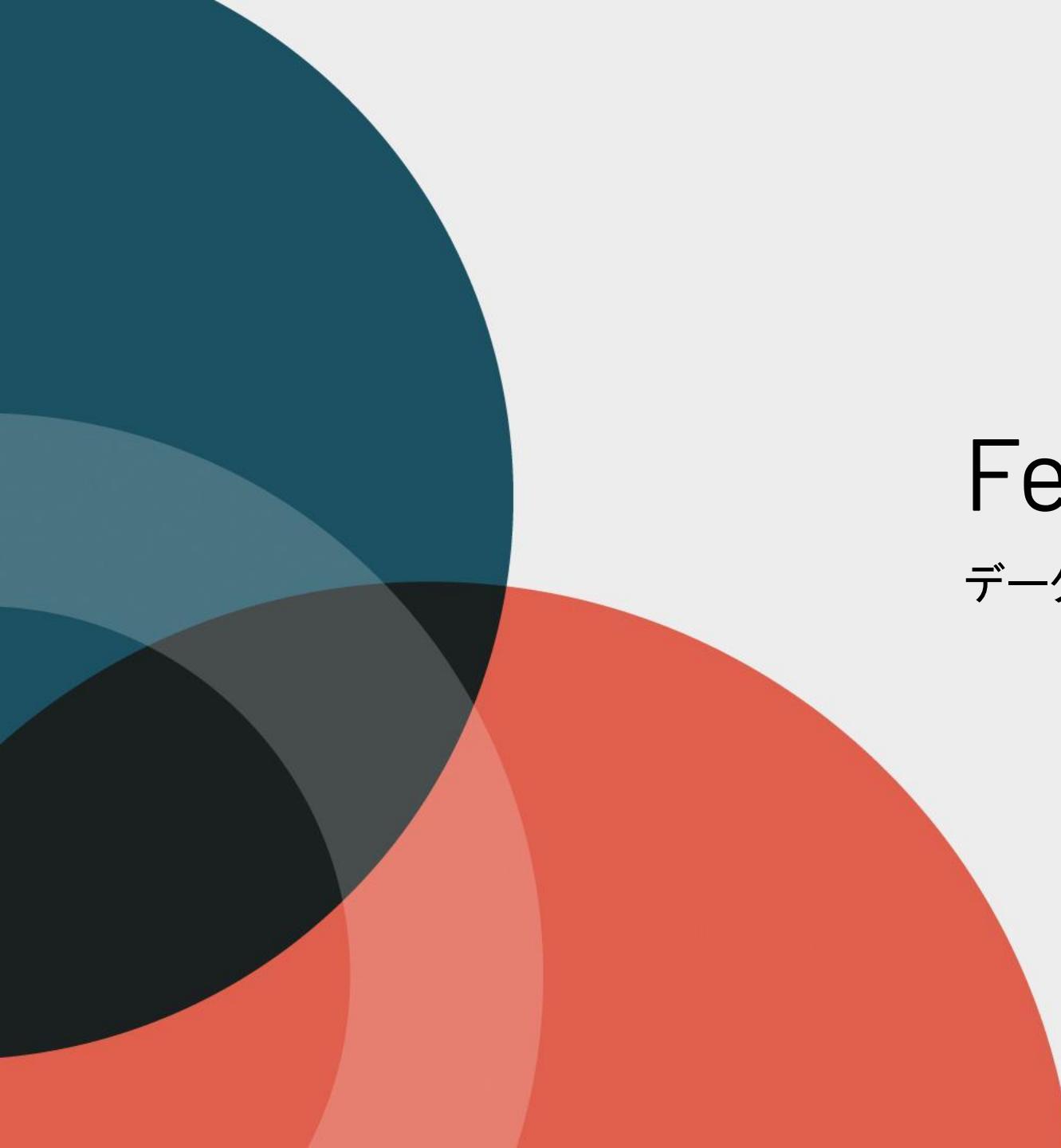
# 信頼性とオペレーションのシンプルさを保ちながらスケールします

オペレーションの状態とデータリネージュをビジュアルで追跡できるツールを用いたパイプラインオペレーションのディープな可視性の獲得

自動エラーハンドリングおよび容易なリトライによるダウンタイムの削減

シングルクリックによるデプロイメント、アップグレードによるメンテナンスのスピードアップ





# Feature Store

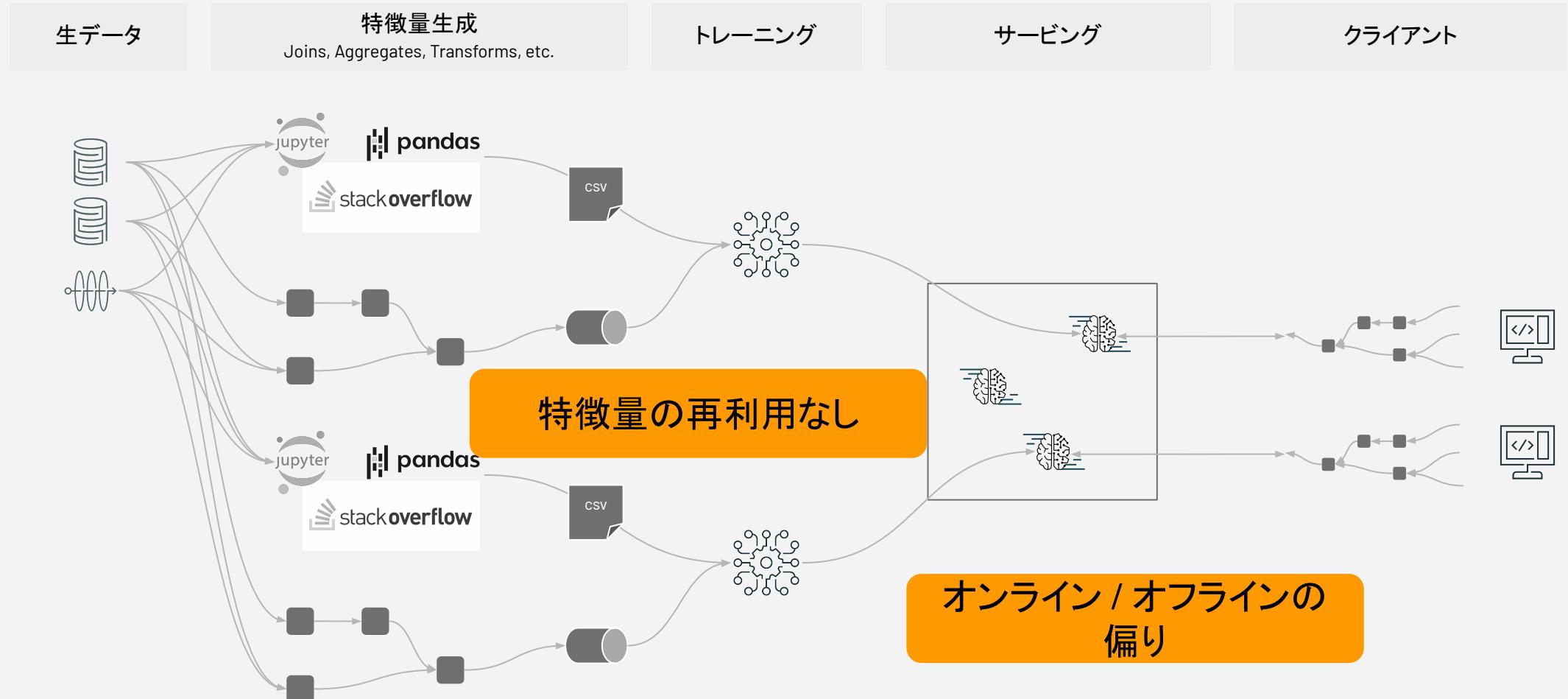
データ、MLOpsと協調設計された特徴量ストア

# Feature Store

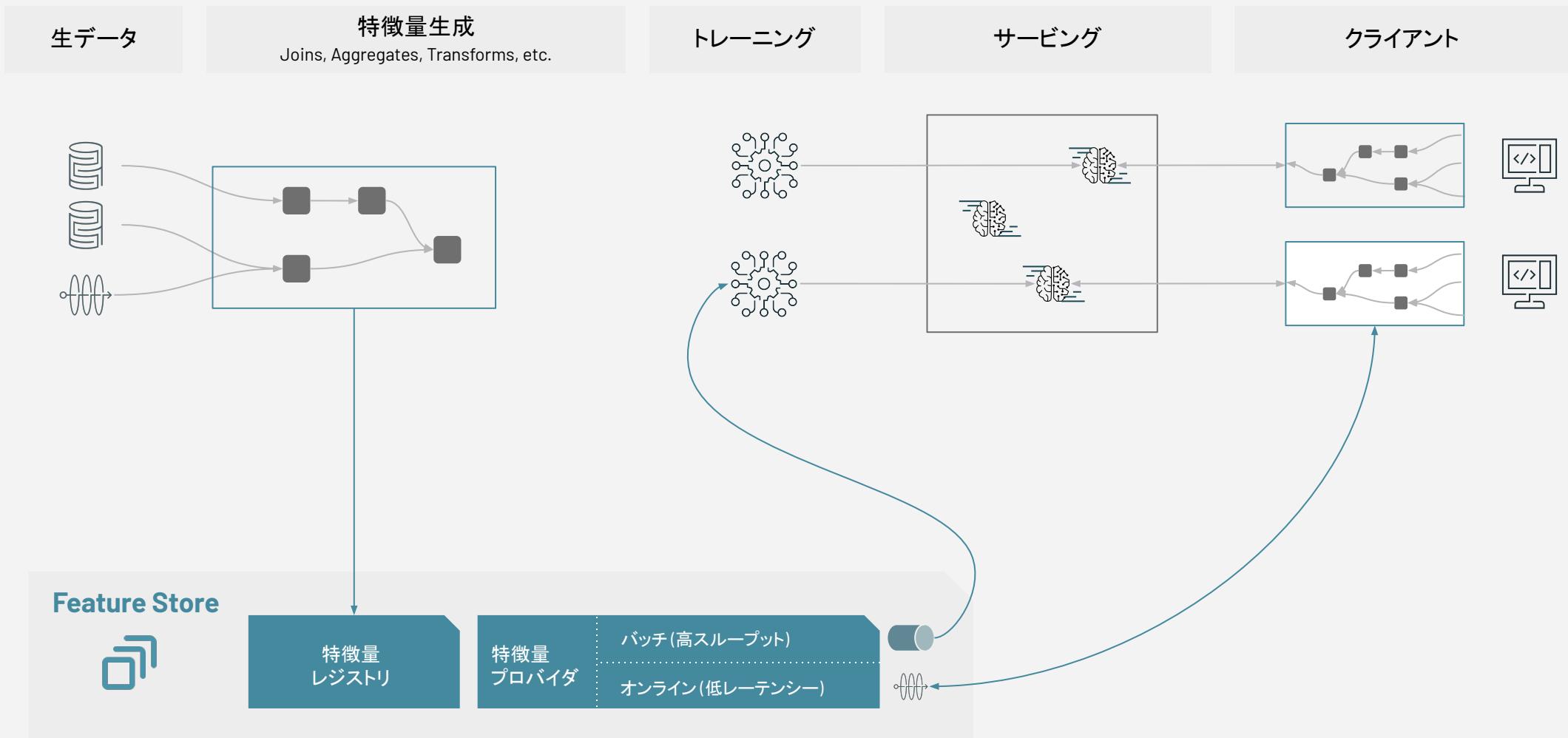
データ、MLOpsと協調設計された史上初の特徴量ストア



# あるMLモデルの1日(あるいは6ヶ月)の生活



# 特徴量ストア問題を解決



# 特徴量ストア問題を解決



## 特徴量レジストリ

- 発見容易性、再利用可能性
- バージョン管理
- アップストリーム、ダウンストリームのリネージュ

## 特徴量プロバイダ

- 特徴量に対するバッチ、オンラインアクセス
- モデルにパッケージされた特徴量検索
- 簡素化されたデプロイメントプロセス

 **DELTA LAKE** との協調設計

- オープンフォーマット
- ビルトインのデータバージョン管理、ガバナンス
- PySpark、SQLなどによるネイティブアクセス

 **mlflow**との協調設計

- 全てのMLフレームワークをサポートするオープンなモデルフォーマット
- 特徴量のバージョン管理、モデルに組み込まれる特徴量検索ロジック



# AutoML

機械学習モデル開発の自動化に対するガラスボックスアプローチ

# Databricks AutoML

管理権限を損なわずにデータチームを強化するガラスボックスソリューション



# AutoMLとは?

**Automated machine learning (AutoML)**は、機械学習を「民主化」するために完全に自動化されたモデル開発ソリューションです。自動化のスコープに違いはありますが、通常AutoML技術はデータからモデル選択までのMLプロセスを自動化します。



# AutoMLはデータサイエンティストの 2つのペインポイントを解決します

データセットのもたらす予測能力を  
クイックに検証する



“このデータセットは顧客解約予測に  
使えるのか?”

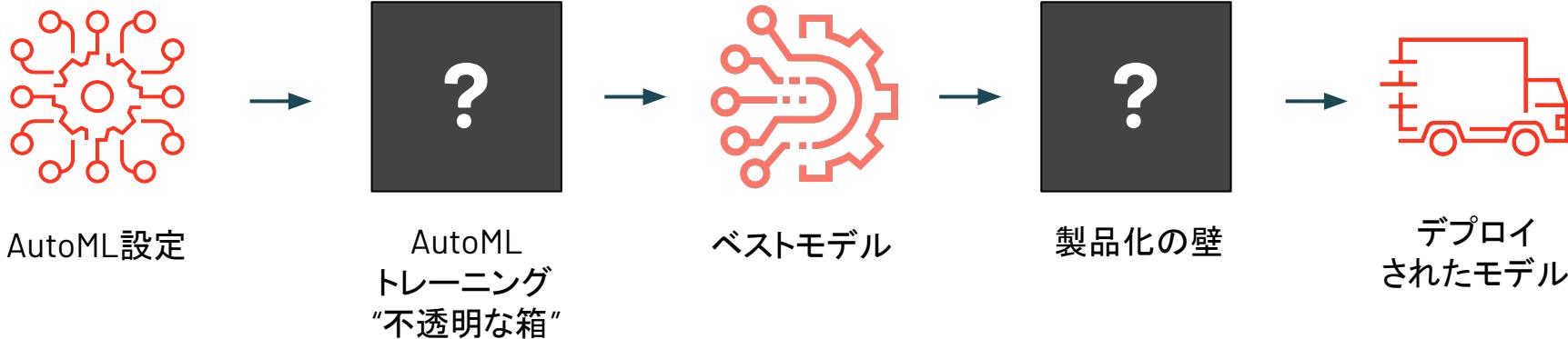
プロジェクトの方向性をガイドするためのベース  
ラインモデルを取得する



“このMLプロジェクトはどの方向に  
進むべきか、目指すべき  
ベンチマークは何か?”

# 既存のAutoMLソリューションの問題

## AutoMLにおける不透明なボックス・製品化の壁問題



問題	結果 / ペインポイント
<ol style="list-style-type: none"><li>生成された"ベスト"モデルをデプロイの前にドメイン知識に基づいて変更する"製品化の壁"が存在します。</li><li>規制対応(FDA、GDPRなど)のためにデータサイエンティストはトレーニングされたモデルを説明できる必要がありますが、多くのAutoMLソリューションは"不透明な箱"モデルとなっています。</li></ol>	<ul style="list-style-type: none"><li>生成される"ベスト"モデルがデプロイには不十分なケースが多くあります。</li><li>モデルを変更、説明できるようにするために、生成された"不透明な箱"をリバースエンジニアリングする労力を費やさなくなりません。</li></ul>

# Databricks AutoMLがサポートする機能

問題の種類	モデル / チューニング	特徴量	トラッキング	評価	デプロイメント
 分類  回帰	 <b>XGBoost</b> 	 数値  カテゴリ変数  タイムスタンプ	 メトリクス  パラメータ  アーティファクト  モデル	 <b>mlflow</b>	 バッチスコアリング  モデルサービング
 時系列予測	 <b>TensorFlow</b>	 テキスト			

# Databricks AutoML

管理権限を損なわずにデータチームを強化するガラスボックスソリューション

AutoMLトレーニングを  
スタートするためのUIと  
API

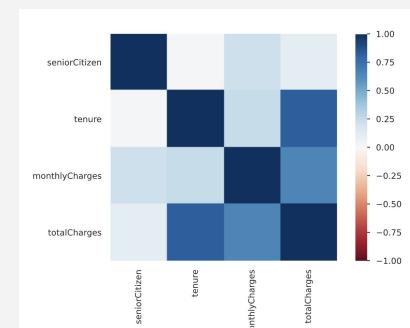
The screenshot shows the 'Configure AutoML experiment' interface. It includes tabs for 'Preview' and 'Prov...'. Step 1 is 'Configure' (highlighted), Step 2 is 'Augment', and Step 3 is 'Train'. Under 'Configure', there's a section for 'AutoML Experiment Configuration' with a 'Compute' dropdown set to 'dais\_mlr\_8\_new'. A note says 'Select an existing cluster with a Databricks Runtime for ML 8.0+ or later'.

A table showing a list of MLflow experiments. The columns are 'Start Time', 'Run Name', 'User', and 'Source'. All runs were started on '2021-05-05 1' and are associated with 'Notebook'. The runs include 'logistic\_r...', 'decision...', and 'random\_f...'.

Start Time	Run Name	User	Source
2021-05-05 1	logistic_r...	kase...	Notebook
2021-05-05 1	logistic_r...	alkis...	21-05
2021-05-05 1	logistic_r...	alkis...	21-05
2021-05-05 1	logistic_r...	kase...	Notebook
2021-05-05 1	logistic_r...	kase...	Notebook
2021-05-05 1	logistic_r...	kase...	Notebook
2021-05-05 1	logistic_r...	kase...	Notebook
2021-05-05 1	decision...	kase...	Notebook
2021-05-05 1	random_f...	kase...	Notebook

## MLflow エクスペリメント

モデルとメトリクスを追跡するために  
自動生成されるMLflowエクスペリメン  
ト



## データ探索ノートブック

特徴量のサマリー統計情報、分布を  
示すノートブックを生成

A screenshot of a 'Generated Trial Notebook (Python)' titled 'dais\_mlr\_8\_new'. It shows a snippet of code for 'Random Forest training' which includes 'Load Data', 'Preprocessors' (with 'Numerical columns' like 'One-hot encoding' and 'Feature standardiz...'), and 'Train classification mo...'. The code uses libraries like pandas, numpy, and shap.

```
# Choose results.
# example
# Use predictors
# Explainer
# Shap summary
```

## 再現可能なトライアルノートブック

全てのモデルに対応するソースコードを含  
むノートブックを生成

モデルレジストリへの  
デプロイが容易

データ品質、前処理  
の理解、デバッグ

AutoMLのモデルに  
専門知識を埋め込み  
精度を改善

# 設定

## Configure AutoML experiment

Experiments > Configure AutoML experiment

1 Configure —— 2 Augment —— 3 Train —— 4 Evaluate

## トレーニング・評価

### AutoML Experiment Configuration

\* Compute

dais\_mlr\_8\_new

Select an existing cluster with a Databricks Runtime for ML 8.0.

\* ML problem type

Classification

\* Training data

default.usage\_logs

\* Target column

isMining

\* Experiment name

isMining\_usage\_logs-2021\_05\_05-22\_33

\* Data directory



AutoML  
Configure

Train

AutoML Evaluation complete.

All runs have completed, and have been added to the table below. Click a specific run to view details or review the data exploration notebook.

Model with best val\_f1\_score

The model is ready to be registered and deployed. Or, access the source code in the generated trial notebook.

Register and deploy model

Edit model

Showing 16 matching runs

Refresh Compare Delete Download CSV

Columns

metrics.rmse < 1 and params.n\_estimators >= 10

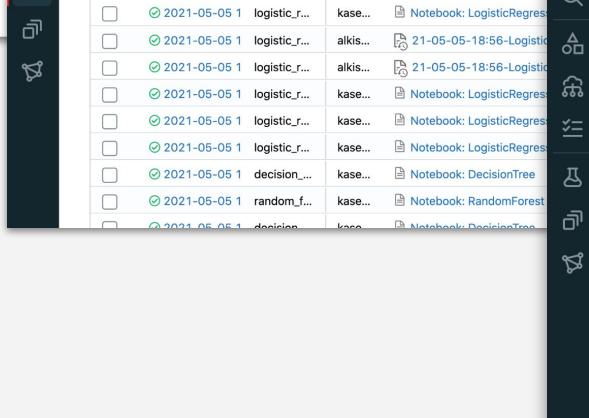
Start Time Run Name User Source

2021-05-05 1 logistic\_r...

2021-05-05 1 decision...

2021-05-05 1 random\_f...

2021-05-05 1 decision...



AutoML  
Configure  
Train

# UIによる“ガラスボックス”AutoML

## カスタマイズ

### Generated Trial Notebook (Python)

```
# Choose any prediction to explain, or sample multiple examples for more thorough
# results.
example = X_val.sample(n=25)

# Use Kernel SHAP to explain feature importance on example
predict = lambda x: model.predict(x)
explainer = KernelExplainer(predict, X_train)
shap_values = explainer.shap_values[example]
summary_plot(shap_values, X_train, example)

except Exception as e:
    print(f"An unexpected error occurred: {e}")
```

Load Data  
Preprocessors  
Numerical columns  
One-hot encoding  
Feature standardiz...  
Train classification mo...  
Feature importance  
Inference

Command took 3.08 minutes - Cnd 25

**デプロイ**

Status: Ready - Stop Cluster: mlflow-model-dais\_demo

Model Versions Model Events Cluster Settings

Model Versions

Version 2 Ready

Model URL: https://dbc-60ef17e8-d99b.dev.databricks.com/model/dais\_demo/2/invocations https://dbc-60ef17e8-d99b.dev.databricks.com/model/dais\_demo/Production/invocations

Call The Model

Browser Curl Python

Request

```
[{"notebookLanguage": "python", "cpu": "3", "ip_address": 165225104128, "account_id": 1587714725935792, "memberStartTime": 1614074788455}
```

Send Request Show Example

Logs Version Events

\$(GUNICORN\_CMD\_ARGS) -- mlflow.pyfunc.scoring\_server.wsgi:app [2021-05-06 03:18:46 +0000] [8143] [INFO] Starting gunicorn 20.1.0

# APIによる“ガラスボックス”AutoML

```
databricks.automl.classify(df, target_col='label', timeout_minutes=60)
```

↓

AutoML

Configure Train

AutoML Evaluation complete.

All runs have completed, and have been added to the table below. Click a specific run to view details or review the [data exploration notebook](#).

Model with best val\_f1\_score

The model is ready to be registered and deployed. Or, access the source code for the model training to make modifications by clicking a notebook under the Source code section.

Register and deploy model Edit model

Showing 16 matching runs

Refresh Compare Delete Download CSV

Columns Search Filter Clear

Start Time	Run Name	User	Source	Version	Models	Parameters >	Metrics >
2021-05-05 1	logistic_r...	kase...	Notebook: LogisticRegressi	-	pyfunc	Logi...	0.0... - 1 1 0.00
2021-05-05 1	logistic_r...	alkis...	21-05-05-18:56-Logisti...	-	pyfunc	Logi...	0.0... - 1 1 9.9..
2021-05-05 1	logistic_r...	alkis...	21-05-05-18:56-Logisti...	-	pyfunc	Logi...	0.0... - 1 1 9.9..
2021-05-05 1	logistic_r...	kase...	Notebook: LogisticRegressi	-	pyfunc	Logi...	0.0... - 1 1 0.00
2021-05-05 1	logistic_r...	kase...	Notebook: LogisticRegressi	-	pyfunc	Logi...	0.0... - 1 1 0.01
2021-05-05 1	logistic_r...	kase...	Notebook: LogisticRegressi	-	pyfunc	Logi...	0.0... - 1 1 0.00
2021-05-05 1	decision_...	kase...	Notebook: DecisionTree	-	pyfunc	Deci...	- - 0.999 0.999 0.00
2021-05-05 1	random_f...	kase...	Notebook: RandomForest	-	pyfunc	Ran...	- False 0.999 0.999 0.04
2021-05-05 1	decision_...	kase...	Notebook: DecisionTree	-	pyfunc	Deci...	- - 0.999 0.999 0.00

Generated Trial Notebook (Python)

Random Forest training

Load Data

Preprocessors

Numerical columns

One-hot encoding

Feature standardiza...

Train classification mo...

Feature importance

Inference

```
# Choose any prediction to explain, or sample multiple examples for more thorough results.
example = X_val.sample(n=25)

# Use Kernel SHAP to explain feature importance on example from validation set
predict = lambda x: model.predict(pd.DataFrame(x, columns=X_train.columns))
explainer = KernelExplainer(predict, train_sample, link="identity")
shap_values = explainer.shap_values(example, l1_reg=False)
summary_plot(shap_values, example)

except Exception as e:
    print(f>An unexpected error occurred while plotting feature importance using SHAP: {e})
```

100% | 25/25 [03:03<00:00, 7.36s/it]

Command took 3.08 minutes -- by kasey.uhlenhuth@databricks.com at 5/5/2021, 10:14:14 PM on dais\_mlr\_8\_new

Cmd 25

# Thank you