JAPAN | JUNE 20, 2024

aws SUMMIT

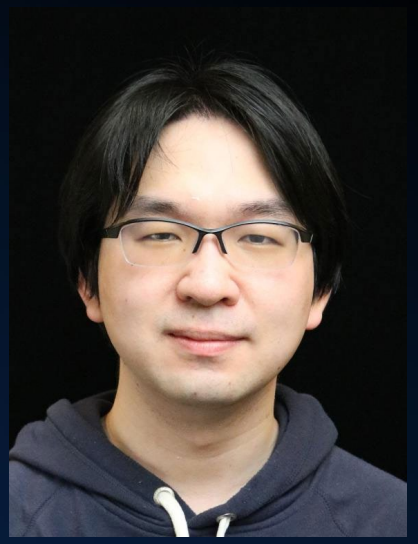
DOL-10

実践 Apache Iceberg! Transactional Data Lake 構築のキーポイント

田中 智大 (Tomohiro Tanaka)
技術支援本部
シニアクラウドサポートエンジニア
アマゾンウェブサービスジャパン合同会社

疋田 宗太郎 (Sotaro Hikita)
Global Financial Services
ソリューションアーキテクト
アマゾンウェブサービスジャパン合同会社

自己紹介



疋田 宗太郎 / Sotaro Hikita
Global Financial Services ソリューションアーキテクト

金融機関のクラウド活用を総合的に支援
多様な業界のデータ分析基盤の構築や改善を支援

- 好きなAWSサービス：AWS Glue
- 趣味：Apache Iceberg

田中 智大 / Tomohiro Tanaka
技術支援本部 シニアクラウドサポートエンジニア

ビッグデータ技術の専門家としてお客様を支援
著書『Serverless ETL and Analytics with AWS Glue』

- 好きなAWSサービス：AWS Glue
- 趣味：Apache Iceberg

想定する対象

データレイクの開発、運用に携わっている方

Apache Spark や Trino, Hive などの分散クエリエンジンを扱う方

Data Lake のコンセプト

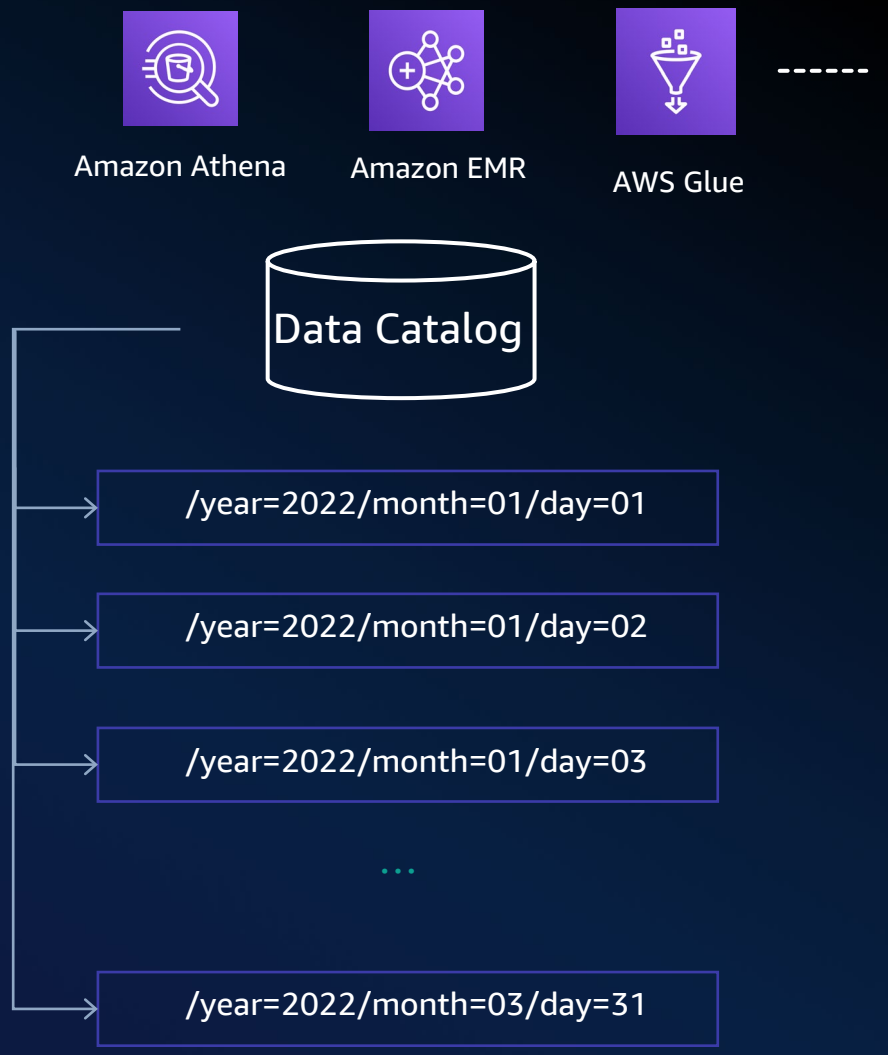
データサイロを排し、大規模な分析を可能にする

- 構造 / 非構造データをスケーラブルなストレージに**一元的に保存**
- **蓄積と分析を分離**して、拡張性とスケーラビリティを確保
- オープンソースなファイルフォーマットにより**多様なツールで分析可能**
- 組織を越えて**すべてのデータ**の分析、活用を可能にする

従来の Data Lake の課題

データレイクはRDBMSとは異なるパラダイムで出来ている

- **同時書き込み時の整合性**を担保できない
- レコードレベルの**操作が非効率**
- データの**過去の状態**が復元が難しい
- **スキーマ変更やデータ構造の変化**への追従負荷が大きい
- クエリを実行する**ユーザが パーティション構造を意識**する必要がある



Transactional な Data Lake が必要とされる代表例

データ分析基盤の成熟に伴って登場するニーズ

- **ストリームパイプライン, Change Data Capture (CDC)**
  - 書込みと読込みのパフォーマンス最適化
  - ソースデータのスキーマ変更への追従
- **個人情報を扱う要件**
  - 特定の個人情報(主にレコードベース)の効率的な削除、更新
  - データ削除中のトランザクションの一貫性
- **データサイエンス**
  - 実験を再現するための、過去のスナップショットの参照
  - 継続的に更新されるデータに対する一貫性を確保した参照

Open Table Format (OTF) によって高度なニーズに対応する

クエリエンジン

trino

Flink

OTF

Apache Iceberg

Apache Hudi

Delta Lake

分散ストレージ

Amazon S3

Open Table Format (OTF) によって高度なニーズに対応する

クエリエンジン

trino

Flink

Amazon Redshift

OTF

Apache Iceberg

Apache Hudi

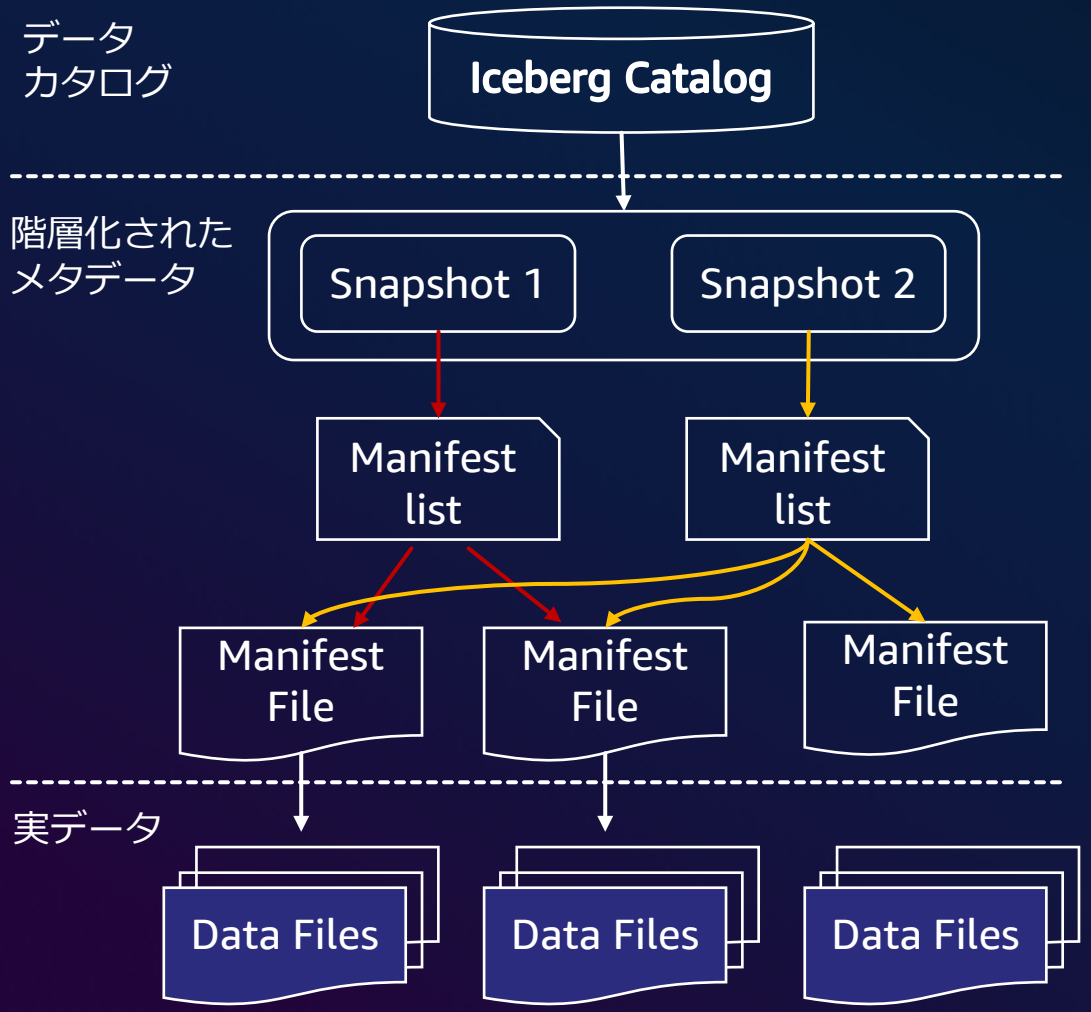
Delta Lake

分散ストレージ

Amazon S3

Apache Iceberg

実データとメタデータを分離して高度な機能を実現する



- **ACID transaction** – データレイクで Serializable isolation を実現
- **Speed** – メタデータ構造の工夫、統計情報の活用や Small File の統合により性能を最適化
- **Storage separation** – ユーザにテーブルの物理構造を意識させない

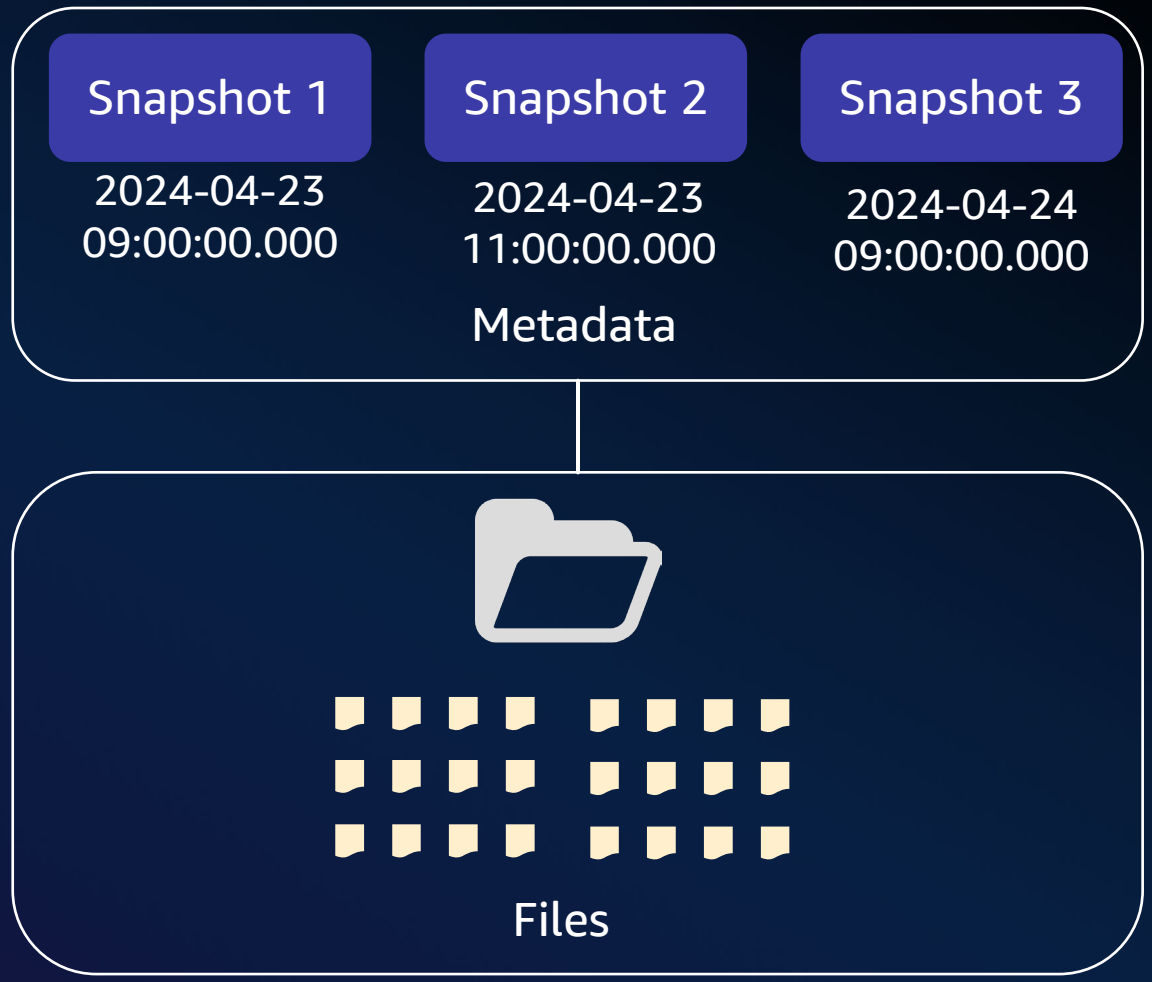
Time Travel, Rollback

テーブルの過去の状態を柔軟に参照

- Iceberg Table は、テーブルへの **各変更をSnapshot として追跡**する
- **Time Travel -** 過去のある時点 / 任意の Snapshot を指定して参照
- **Rollback –** テーブルの状態を過去の Snapshot へ差し戻す

```
SELECT * FROM table_name FOR TIMESTAMP
AS OF timestamp

SELECT * FROM table_name FOR VERSION
AS OF snapshot_id
```



Evolution

テーブル構造の変化に柔軟に対応

テーブル構造の変化に**メタデータの変更**のみで対応
新旧のテーブル構造を**シームレスに操作**可能

- **Schema Evolution**
  - カラムの追加、削除、更新、並び替えが可能
- **Partition Evolution**
  - パーティション構造を変更が可能
  - Hidden Partitioning によりユーザは構造を意識しない
- **Sort order Evolution**
  - ソートに使用するカラム、順序を変更が可能

Write-Audit-Publish (WAP) によるデータ品質管理

データを効率的に処理・公開するためのデータパイプラインの設計手法

- **Write:**
  新しいデータを、Staging 環境に書き込み
- **Audit:**
  書き込まれたデータに対して、品質チェックや整合性の検証を行う
- **Publish:**
  Audit 済のデータを、他のシステムやエンドユーザーが利用できるように公開

Branching

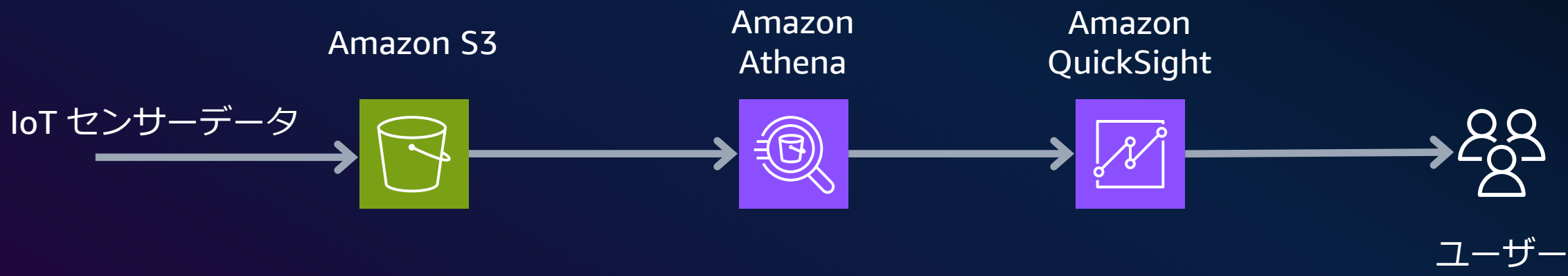
テーブルバージョンを独立ライフサイクルのブランチで管理

- Iceberg のテーブルには Branch の概念があり、デフォルトは Main
- ある時点の Snapshot を元に、Main から独立したライフサイクルの branch が作成可能
- Fast Forward によって、Main への書き戻しも可能

DEMO:
Iceberg の Branch 機能を利用し、Write-Audit-Publish パターンを実現する

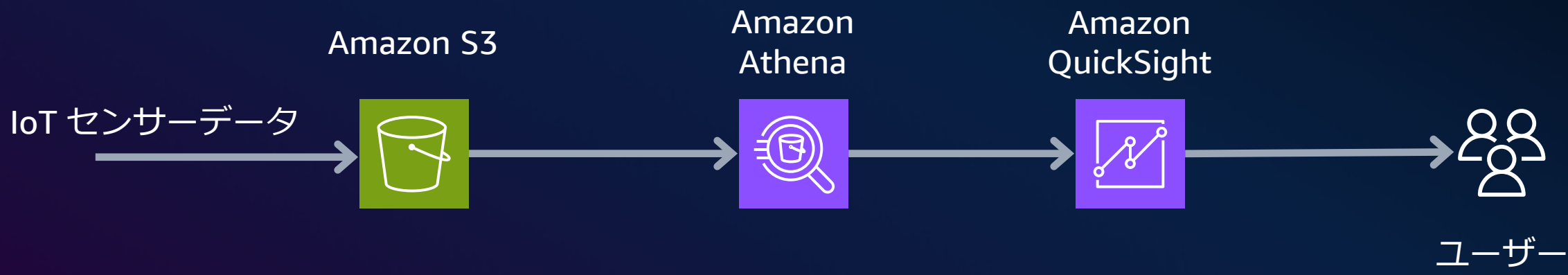
Demo: ユースケース

様々な部屋における IoT センサーデータを 、
Amazon S3 上のデータレイクに保存する。
これを Amazon QuickSight により可視化し、分析する



Demo: 想定する問題

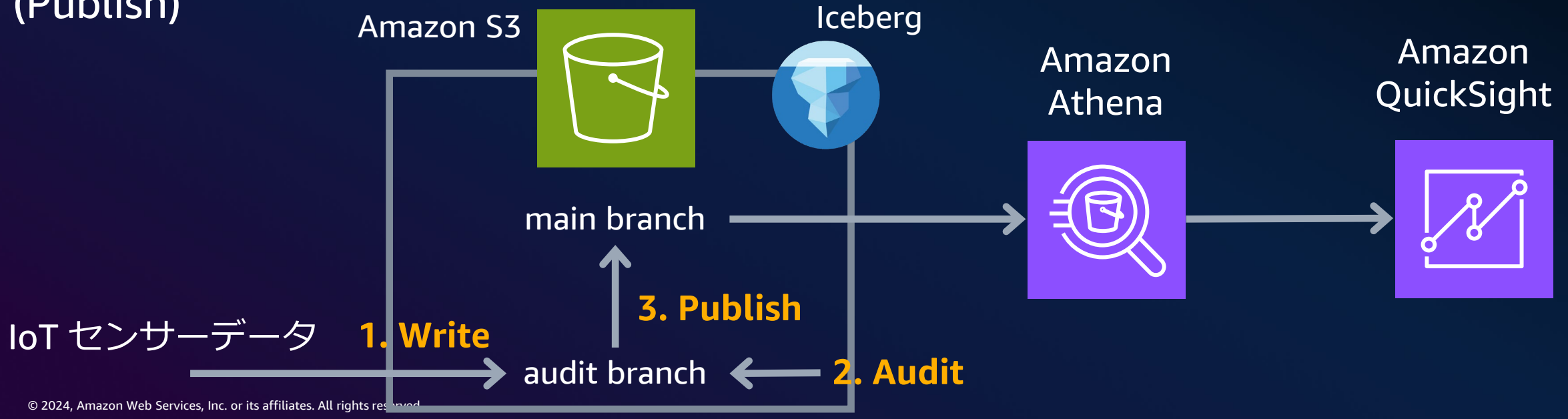
センサー故障により不正データが S3 に書き込まれ、
QuickSight での可視化、および分析結果に問題が発生する



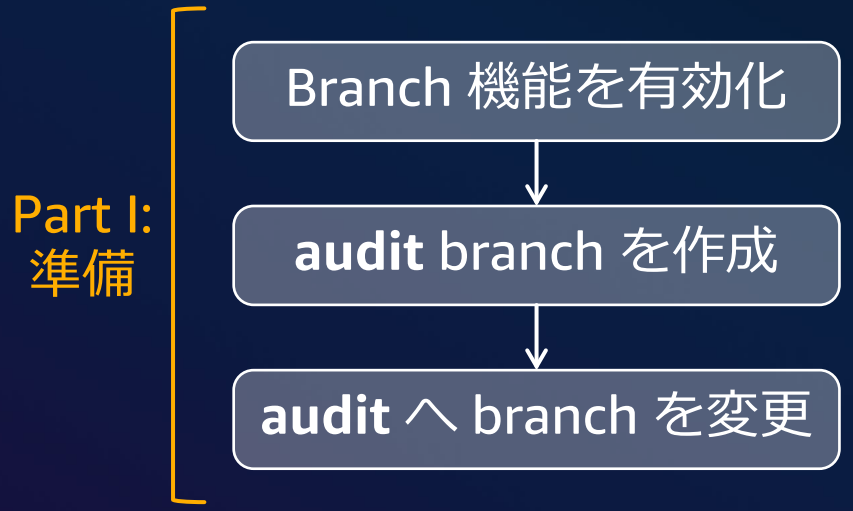
Demo: Iceberg Branch 機能で解決できること

Iceberg Branch 機能により、以下の "WAP" パターンで正常なデータをユーザーにパブリッシュする

1. IoT センサーデータを一時的に別の "Branch" へ書き込む (Write)
2. 別 Branch へ書き込んだデータが正常か確認する (Audit)
3. データが正常な場合に、"main" branch へ書き込み、パブリッシュする (Publish)



Part I. Iceberg Branch 機能の準備

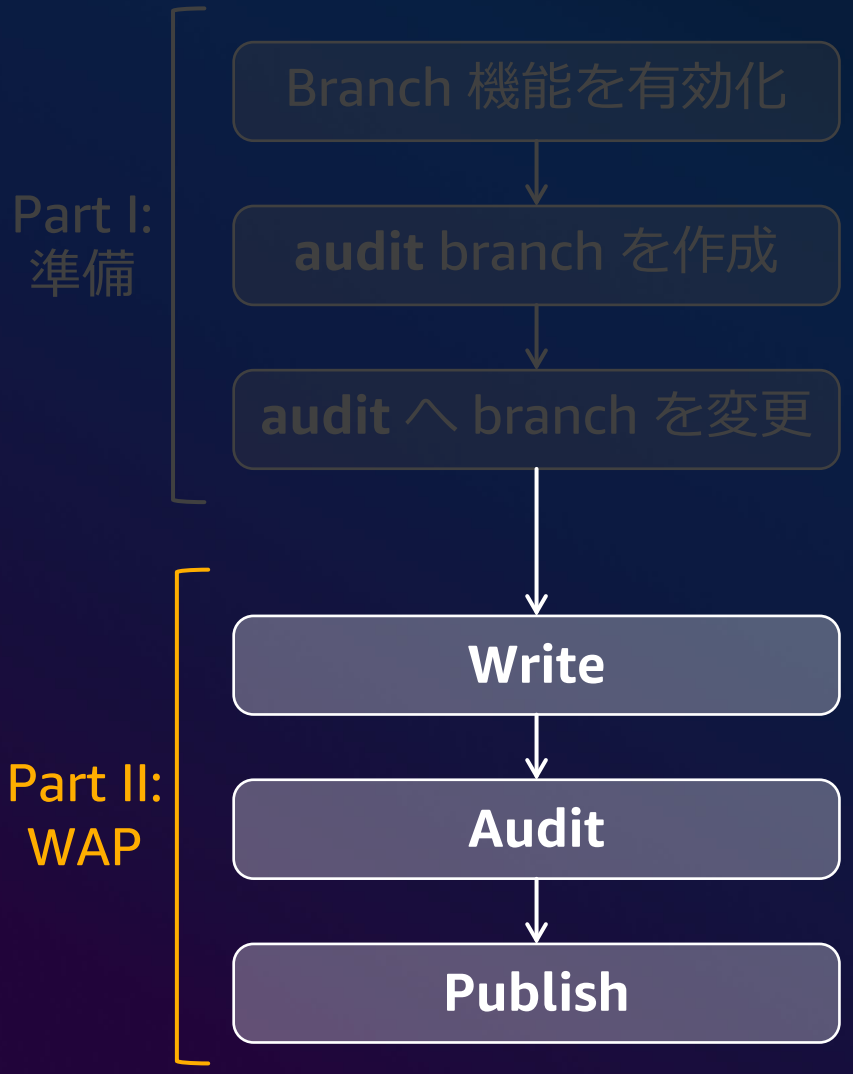



```
ALTER TABLE db.tbl
SET TBLPROPERTIES ('write.wap.enabled'='true')

ALTER TABLE db.tbl CREATE BRANCH audit

SET spark.wap.branch = audit
```

Part II. WAP を Iceberg Branch 機能で行う




```
ALTER TABLE db.tbl
SET TBLPROPERTIES ('write.wap.enabled'='true')
```

```
ALTER TABLE db.tbl CREATE BRANCH audit
```

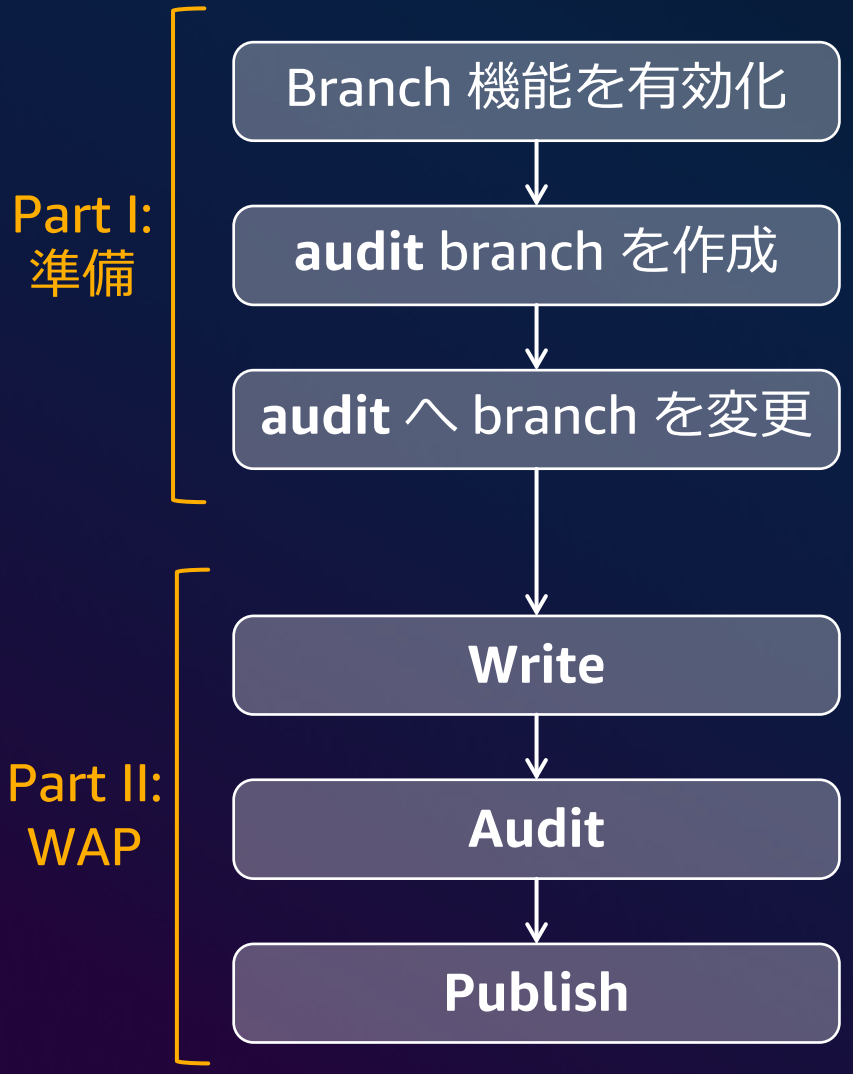
```
SET spark.wap.branch = audit
```

壊れたデータが audit branch に対して書き込まれる

Glue Data Quality を利用し、センサー対応温度範囲かどうかデータ品質を確認する

データ品質チェックにパスしたデータを **main** branch に *fast forward* し、データをパブリッシュする

Demo: Iceberg Branch 機能で WAP を実現する




```
ALTER TABLE db.tbl
SET TBLPROPERTIES ('write.wap.enabled'='true')
```

```
ALTER TABLE db.tbl CREATE BRANCH audit
```

```
SET spark.wap.branch = audit
```

壊れたデータが audit branch に対して書き込まれる

Glue Data Quality を利用し、センサー対応温度範囲かどうかデータ品質を確認する

データ品質チェックにパスしたデータを **main** branch に *fast forward* し、データをパブリッシュする

まとめ

- Transactional data lake は従来の data lake で実現が難しかった、ACID 特性や、タイムトラベルクエリ、UPSERT/UPDATE/DELETE といったオペレーションを実行できる
- OTF はコンピュートとストレージレイヤーを分離し、分散クエリエンジンとストレージを自由に組み合わせられるようにする
- OTF の 1 つである Apache Iceberg は、ACID 特性を持ち、タイムトラベル/ロールバック機能や、コンパクションといったファイルマネジメント機能を持つ
- Iceberg には Branch 機能があり、Write-Audit-Publish パターンを実現し、高品質なデータをコンシューマーにデリバリーすることができる

Thank you!

田中 智大 (Tomohiro Tanaka)

tomtongue

in/ttomtan

疋田 宗太郎 (Sotaro Hikita)

lawofcycles

in/hsotaro