Gunosy

Gunosy におけるデータの民主化を促進するデータ基盤

株式会社 Gunosy
Gunosy Tech Lab Data Reliability & MLOps Group
楠 湧夢 <Izumu KUSUNOKI>

2023年5月18日(木)

自己紹介

Gunosy

- 楠 湧夢 <Izumu KUSUNOKI>
- 所属: Gunosy Tech Lab
  - Data Reliability & MLOps Group
- 経歴
  - 大学院修士課程: Wi-Fi 通信の応用技術
  - Gunosy (2021/04~): データ基盤の開発運用

本日話すこと

Gunosy

Gunosy の統一社内データ基盤である Baikal について

① どのようにしてデータレイクを用いてデータ基盤を構成するか

② Gunosy における「データの民主化」とは ～データ基盤の運用事例～

③ データの民主化によって得られたもの・これからのデータ基盤とは

想定する聴講者

- これから AWS 上にデータ基盤を構築したい
- データ基盤を構築する上でのエッセンスを知りたい

① どのようにしてデータレイクを用いて
データ基盤を構成するか

データレイクを用いたデータ基盤構成の方法

Gunosy

本節の概要：データレイクについて技術的なお話をします

- データレイクとは何なのか？
- なぜ・いつデータレイクが採用されるのか？
- どうやったらデータレイクを構築できるのか？

データレイクとは何なのか？

Gunosy

AWS におけるデータレイク：S3 上にすべてのデータを集約する

データレイクはデータウェアハウス（Redshift 等）と異なり、専用のストレージを持ちません。データを格納する際のスキーマ制約もありません。

要するに、S3 にすべてのデータを集めればデータレイクを実現できます。

なぜ・いつデータレイクが採用されるのか？

Gunosy

以下のような利点に価値がある場合に採用されます

- **利点 1：半構造データを取り込みやすく、扱いやすい（寿司原則）**
  とりあえず S3 に置いておけば、あとから帳尻を合わせられる（Schema on Read）

- **利点 2：寿司原則により、データ変換の負荷が最適化される（ETL -> ELT）**
  分析コスト最適化・MLOps への発展

なぜ・いつデータレイクが採用されるのか？

Gunosy

以下のような利点に価値がある場合に採用されます

- 利点 1：半構造データを取り込みやすく、扱いやすい（寿司原則）

  とりあえず S3 に置いておけば、あとから帳尻を合わせられる（Schema on Read）

10.2.5.1　ストレージの多様性

[前略] すなわち、**データを生のままの形で収集し、スキーマの設計については後で考えるようにすれば、データの収集速度を上げられます**（この概念は「**データレイク**」あるいは「エンタープライズデータハブ」と呼ばれます[55]）。

[中略] …

これは、生成側と消費側が別々のチームで、それぞれの優先順位が異なっている場合、メリットになることがあります。**理想的な単一のデータモデルというものはなくても、様々な目的に適したデータの見方がいくつもある場合もあります。**生の形式のままでデータを単純にダンプすれば、そういった変換を何種類も行えます。こういったアプローチは**寿司原則（sushi principle）**、すなわち「**データは生の方が良い**」と表現されてきました[57]。

Ref: Martin Kleppmann, 斉藤 太郎, 玉川 竜司『データ指向アプリケーションデザイン ―信頼性、拡張性、保守性の高い分散システム設計の原理』（O'Reilly Japan, Inc., July 2019）

なぜ・いつデータレイクが採用されるのか？

Gunosy

以下のような利点に価値がある場合に採用されます

- 利点 2：寿司原則により、データ変換の負荷が最適化される（ETL -> ELT）

  分析コスト最適化・MLOps への発展

14.1.2 データレイク

データウェアハウスの複雑さ、コスト、失敗に対する多くの反動によって、設計の振り子は反対の極に振れた。その例の一つに、**データウェアハウスパターンの意図的な逆バージョンであるデータレイクパターンがある**。データレイクパターンは、一元化されたモデルとパイプラインはそのままに、**データウェアハウスの「変換してロードする」モデルを「ロードして変換する」モデルに反転させたもの**だ。データレイクパターンの哲学は、必要のない無駄な変換をデータに行わないことで、通常だと目的のために変換やマスキングが行われてしまう分析データへビジネスユーザーが自然な形でアクセスできるようにするというものだ。 [中略] … **また、多くの機械学習モデルには、変換後のデータより、できるだけ加工されていないデータの方が適していた**。

Ref: Neal Ford, Mark Richards, Pramod Sadalage, Zhamak Dehghani, 島田 浩二『ソフトウェアアーキテクチャ・ハードパーツ ―分散アーキテクチャのためのトレードオフ分析』（O'Reilly Japan, Inc., October 2022）

① どのようにしてデータレイクを用いてデータ基盤を構成するか

なぜ・いつデータレイクが採用されるのか？

Gunosy

以下のような利点に価値がある場合に採用されます

- 利点 1：半構造データを取り込みやすく、扱いやすい（寿司原則）

  どのようなケースか：

  半構造データを柔軟に活用したい、データ取り込みをスケールさせたい

- 利点 2：寿司原則により、データ変換の負荷が最適化される（ETL -> ELT）

  どのようなケースか：

  データウェアハウスのモデリング専門家がいない、データ変換定義の負荷が高い

どうやったらデータレイクを構築できるのか？

Gunosy

AWS におけるデータレイク：アナロジーで理解する

データベースの表現モデルである ANSI/SPARC 3層スキーマを例にします

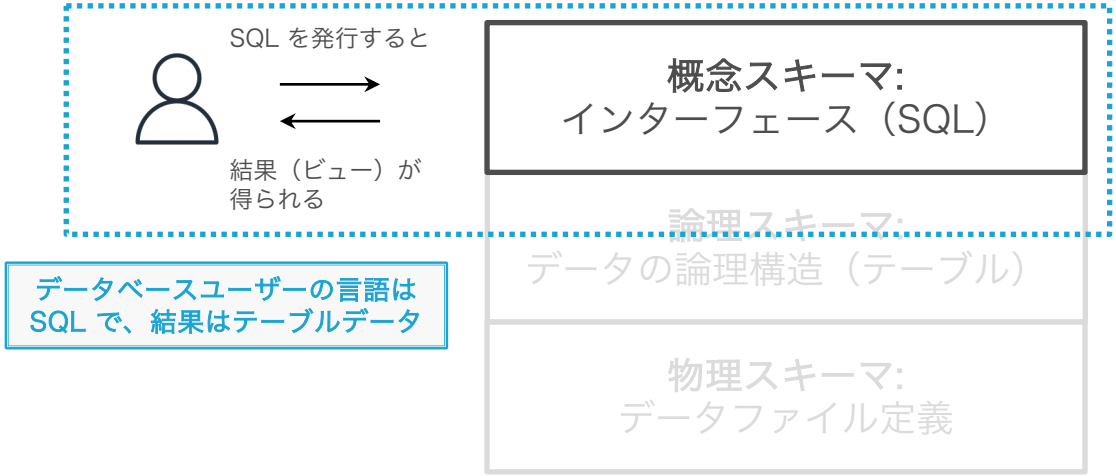
① どのようにしてデータレイクを用いてデータ基盤を構成するか

どうやったらデータレイクを構築できるのか？

Gunosy

AWS におけるデータレイク：アナロジーで理解する

データベースの表現モデルである ANSI/SPARC 3層スキーマを例にします

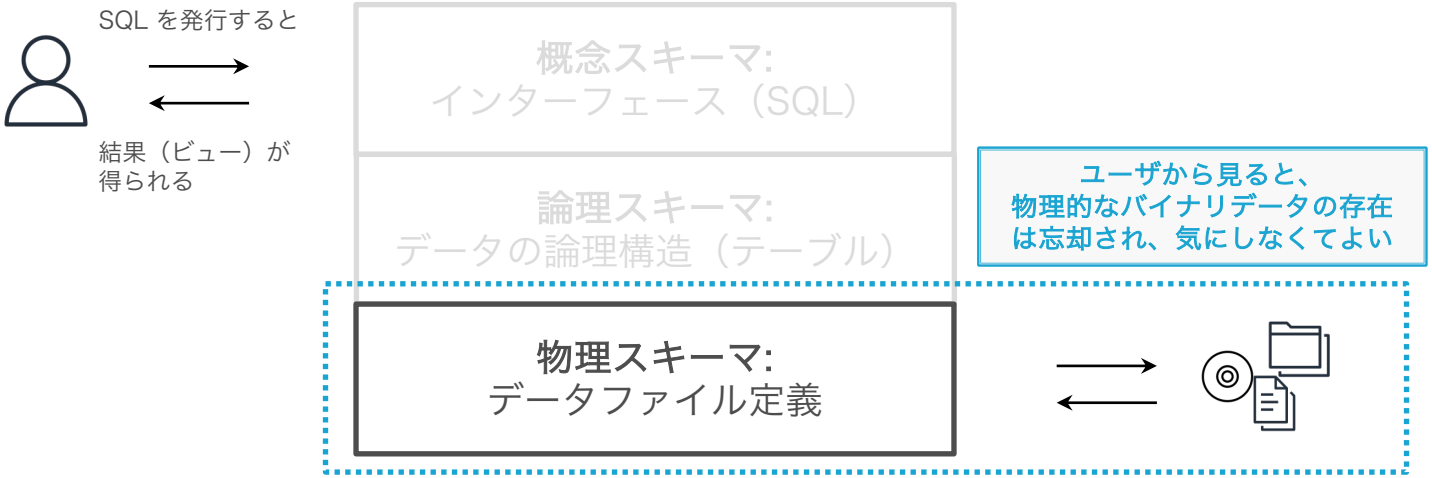


どうやったらデータレイクを構築できるのか？

Gunosy

AWS におけるデータレイク：アナロジーで理解する

データベースの表現モデルである ANSI/SPARC 3層スキーマを例にします



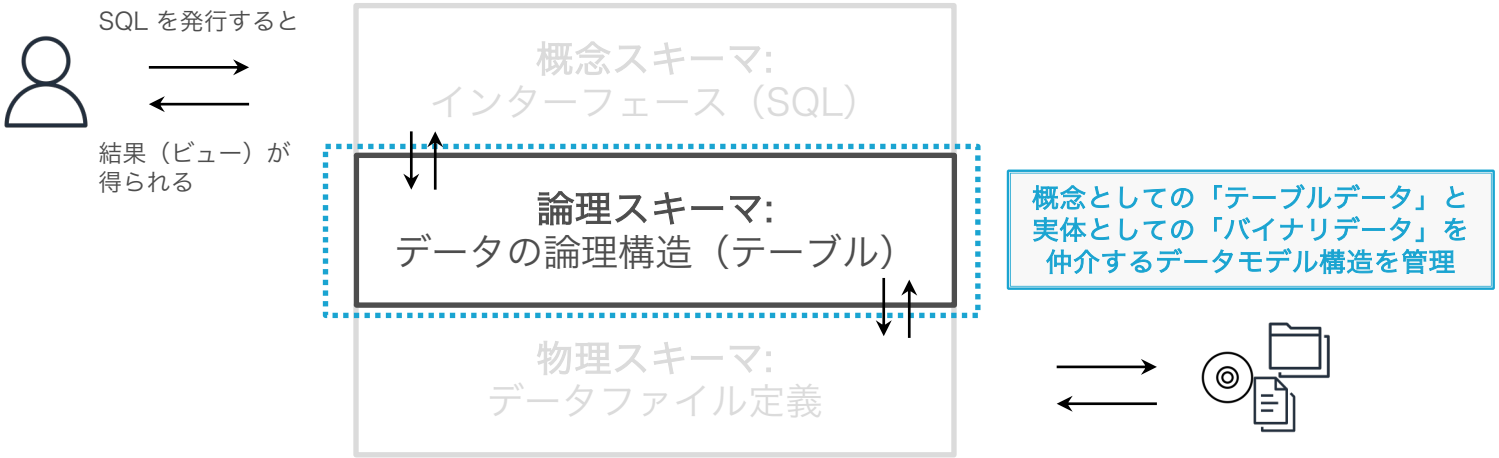
① どのようにしてデータレイクを用いてデータ基盤を構成するか

どうやったらデータレイクを構築できるのか？

Gunosy

AWS におけるデータレイク：アナロジーで理解する

データベースの表現モデルである ANSI/SPARC 3層スキーマを例にします



① どのようにしてデータレイクを用いてデータ基盤を構成するか

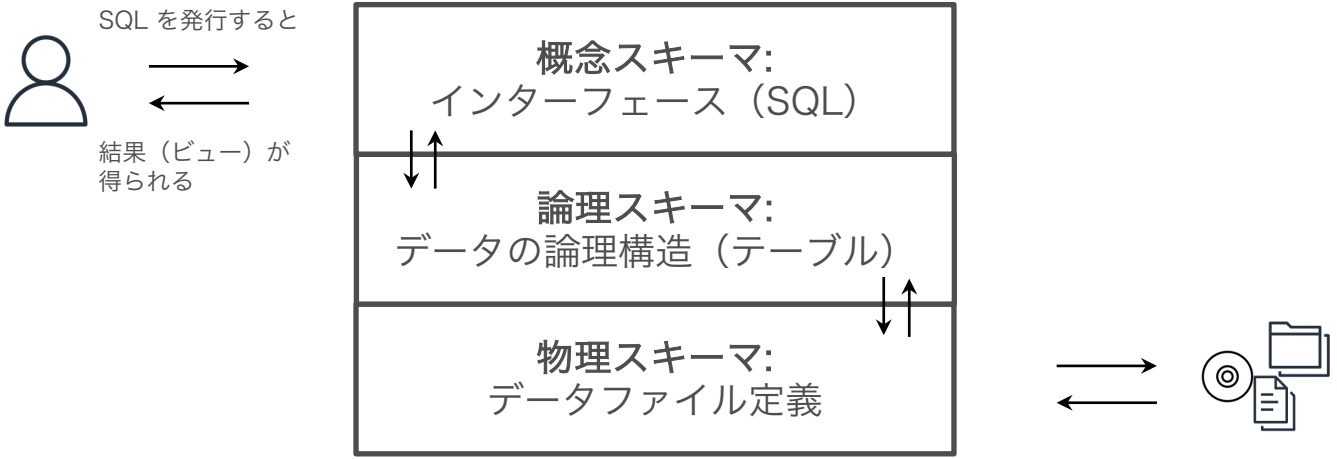
どうやったらデータレイクを構築できるのか？

Gunosy

AWS におけるデータレイク：アナロジーで理解する

データベースの表現モデルである ANSI/SPARC 3層スキーマを例にします

このモデルに従うことで、ユーザーに対する「物理データの隠蔽」ができていると考える



① どのようにしてデータレイクを用いてデータ基盤を構成するか

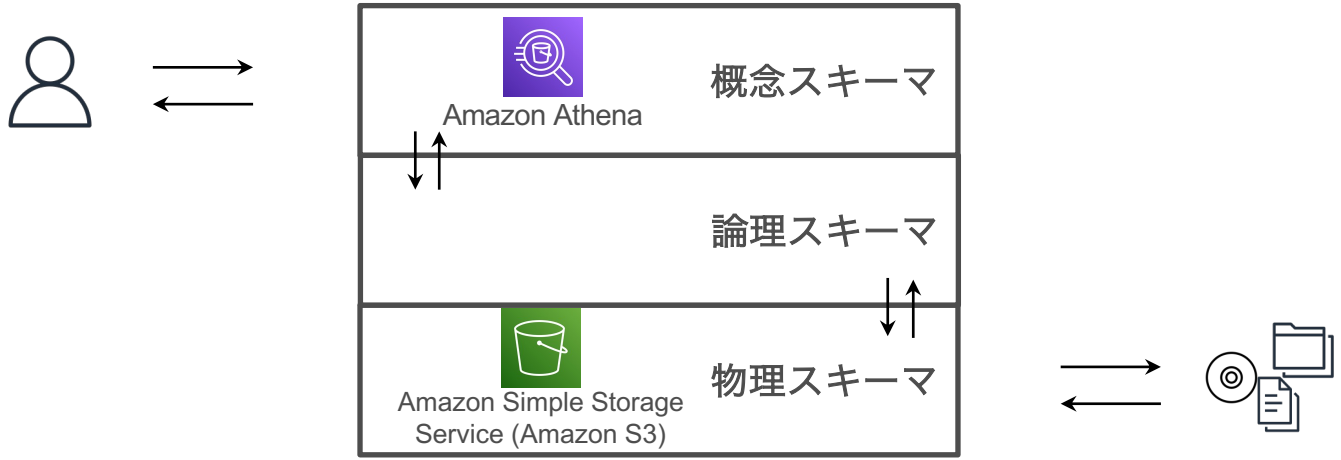
どうやったらデータレイクを構築できるのか？

Gunosy

AWS におけるデータレイク：アナロジーで理解する

AWS のサービスであてはめてみると見えてくるギャップ

AWS サービスでこのモデルに従うことで、ユーザーに対する「物理データ（S3 データ）の隠蔽」がどうできるかを考える



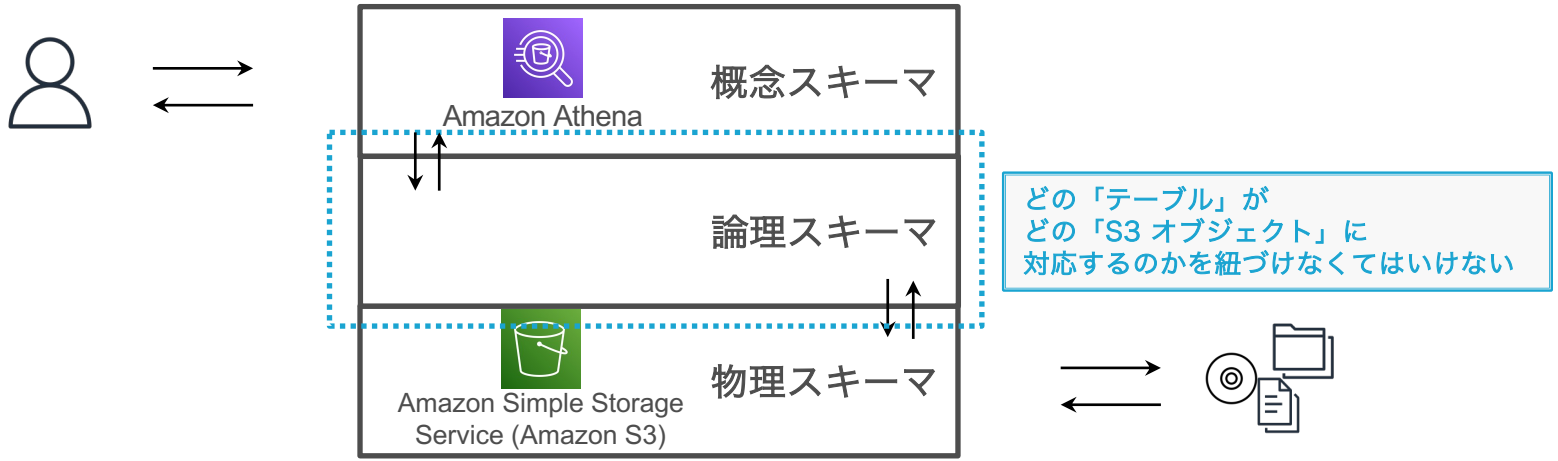
① どのようにしてデータレイクを用いてデータ基盤を構成するか

どうやったらデータレイクを構築できるのか？

Gunosy

AWS におけるデータレイク：アナロジーで理解する

AWS のサービスであてはめてみると見えてくるギャップ



① どのようにしてデータレイクを用いてデータ基盤を構成するか

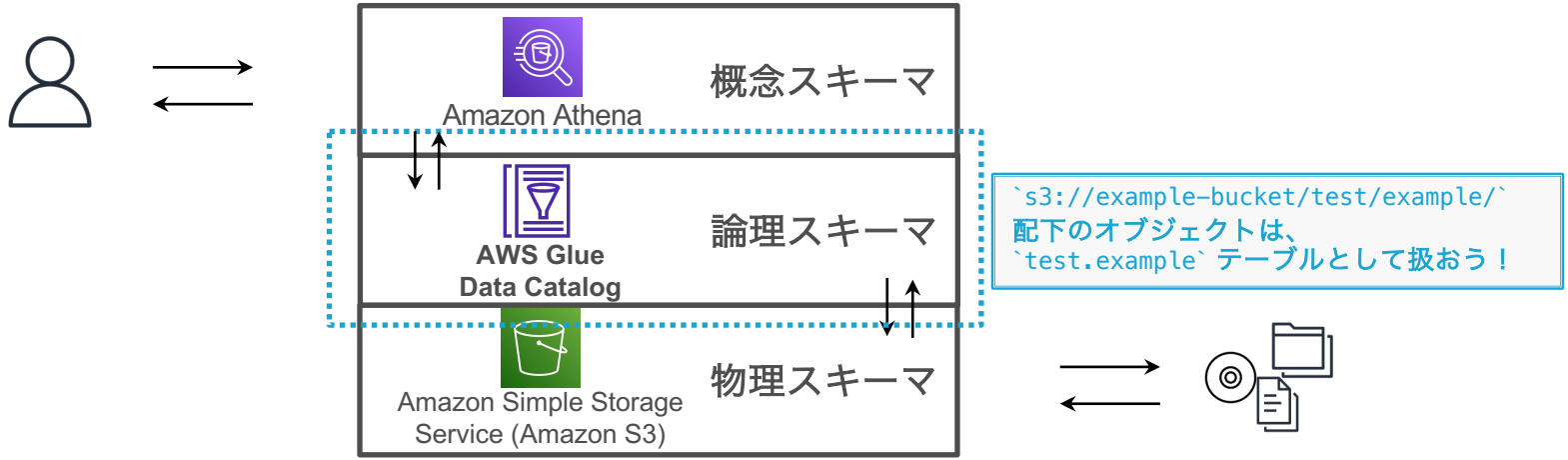
どうやったらデータレイクを構築できるのか？

Gunosy

AWS におけるデータレイク：アナロジーで理解する

AWS のサービスであてはめてみると見えてくるギャップ

ユーザーは `test.example` テーブルを知っていれば
クエリできる（裏が S3 かどうかすら知らなくてよい）



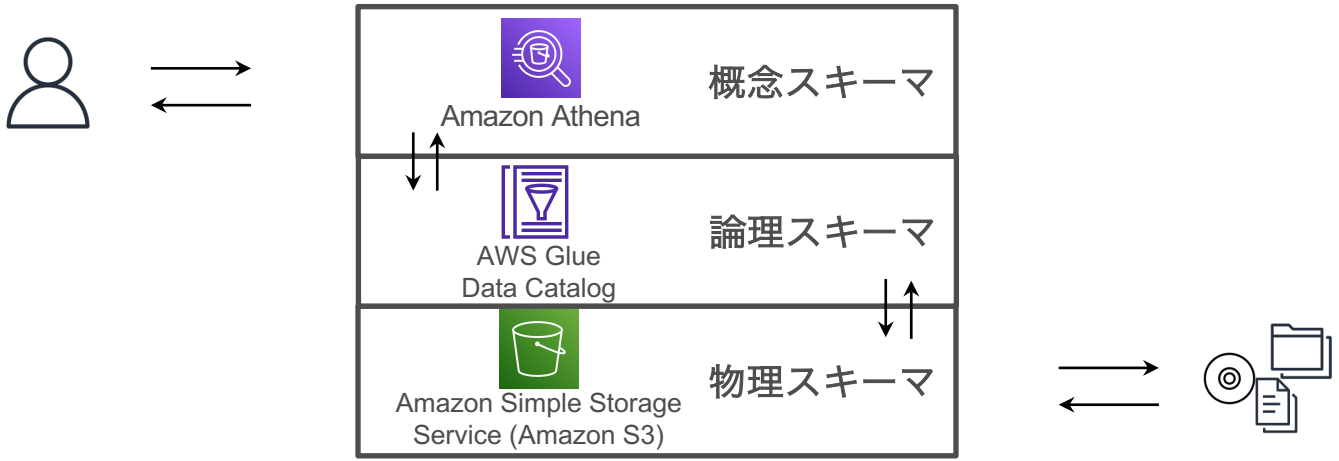
① どのようにしてデータレイクを用いてデータ基盤を構成するか

どうやったらデータレイクを構築できるのか？

Gunosy

AWS におけるデータレイク：アナロジーで理解する

データレイクを使ってデータ基盤を提供できた！



① どのようにしてデータレイクを用いてデータ基盤を構成するか

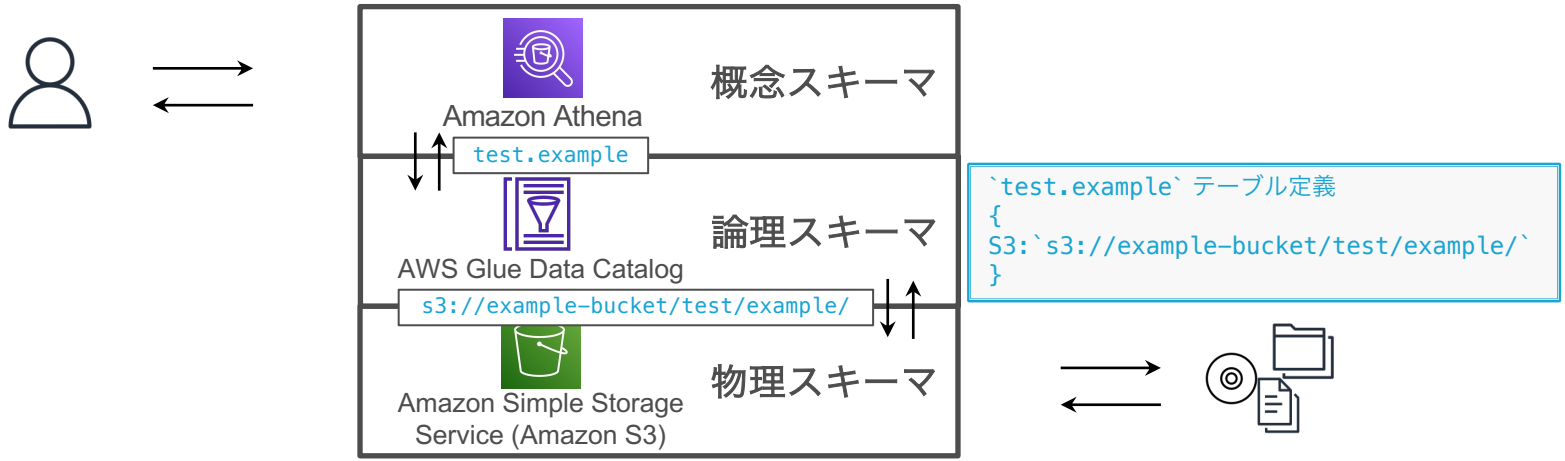
どうやったらデータレイクを構築できるのか？

Gunosy

AWS におけるデータレイク：アナロジーで理解する

さらに深掘り：データレイクのアクセス管理として AWS Lake Formation はなぜ必要なのか？

IAM ロールと S3 バケットポリシーだけでアクセス制限をするのは難しい：関心があるのはオブジェクト（物理スキーマ）ではなく論理スキーマ！



① どのようにしてデータレイクを用いてデータ基盤を構成するか

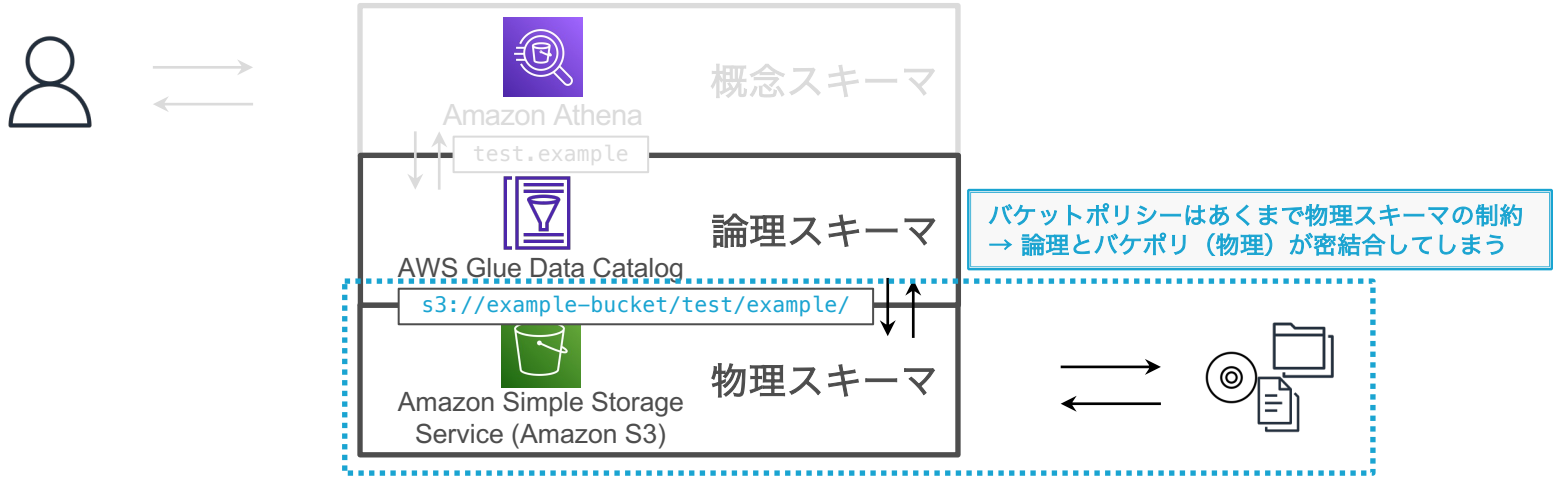
どうやったらデータレイクを構築できるのか？

Gunosy

AWS におけるデータレイク：アナロジーで理解する

さらに深掘り：データレイクのアクセス管理として AWS Lake Formation はなぜ必要なのか？

IAM ロールと S3 バケットポリシーだけでアクセス制限をするのは難しい：関心があるのはオブジェクト（物理スキーマ）ではなく論理スキーマ！



① どのようにしてデータレイクを用いてデータ基盤を構成するか

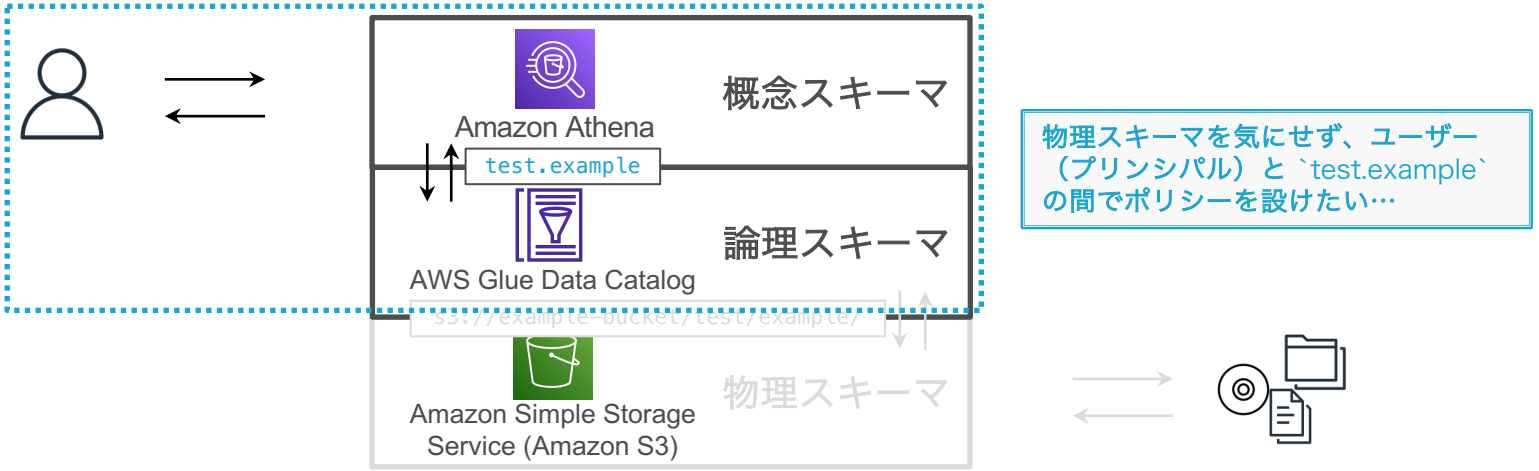
どうやったらデータレイクを構築できるのか？

Gunosy

AWS におけるデータレイク：アナロジーで理解する

さらに深掘り：データレイクのアクセス管理として AWS Lake Formation はなぜ必要なのか？

IAM ロールと S3 バケットポリシーだけでアクセス制限をするのは難しい：関心があるのはオブジェクト（物理スキーマ）ではなく論理スキーマ！



① どのようにしてデータレイクを用いてデータ基盤を構成するか

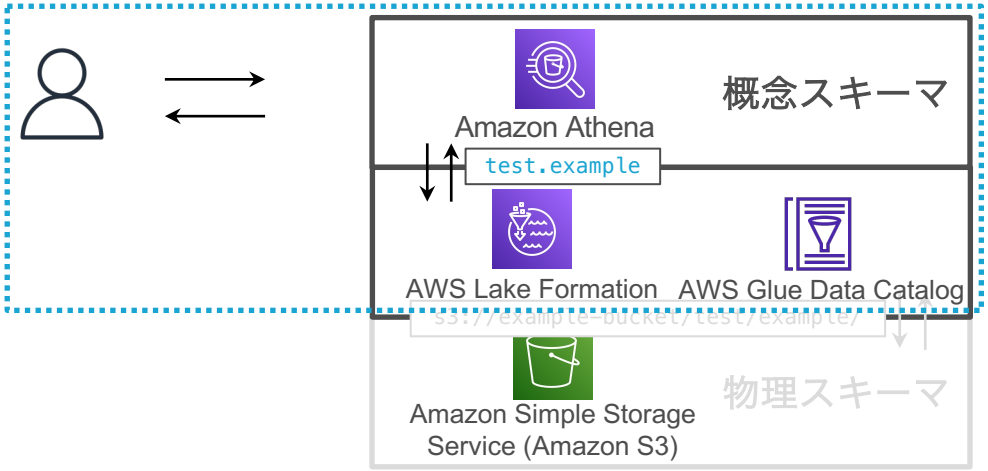
どうやったらデータレイクを構築できるのか？

Gunosy

AWS におけるデータレイク：アナロジーで理解する

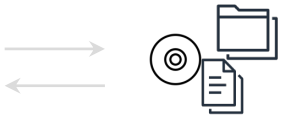
さらに深掘り：データレイクのアクセス管理として AWS Lake Formation はなぜ必要なのか？

IAM ロールと S3 バケットポリシーだけでアクセス制限をするのは難しい：関心があるのはオブジェクト（物理スキーマ）ではなく論理スキーマ！



物理スキーマを気にせず、ユーザー（プリンシパル）と `test.example` の間でポリシーを設けたい…

…それが AWS Lake Formation！

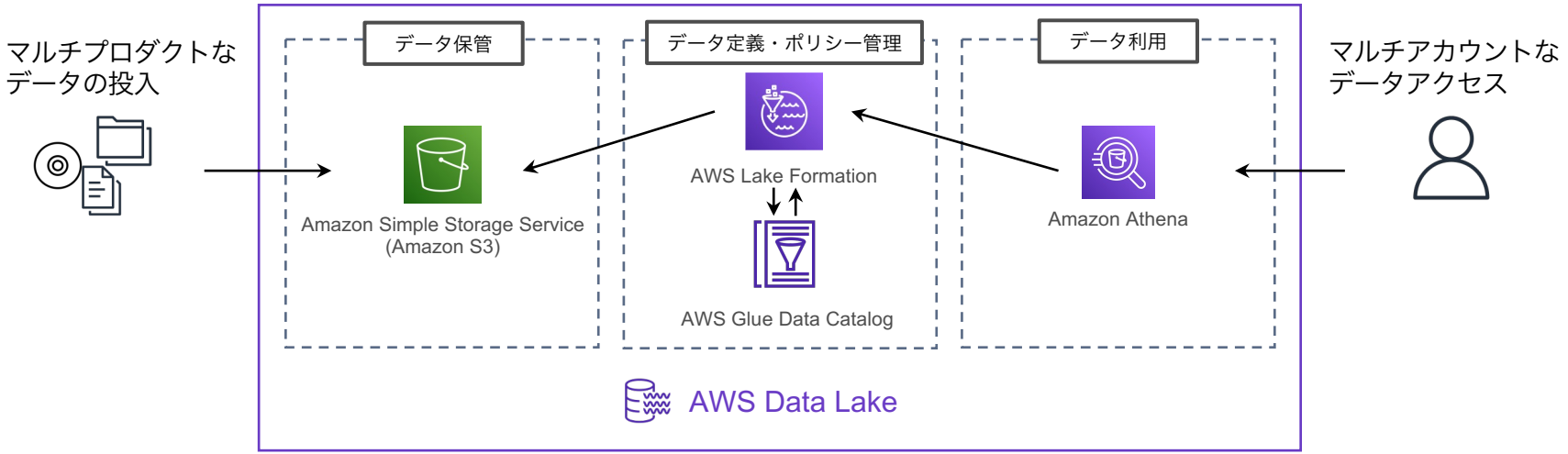


① どのようにしてデータレイクを用いてデータ基盤を構成するか

どうやったらデータレイクを構築できるのか？

Gunosy

AWS におけるデータレイク：まとめ



① どのようにしてデータレイクを用いてデータ基盤を構成するか

② データ基盤 「Baikal」と
データ基盤チームの取り組みの紹介

Gunosy におけるデータの民主化

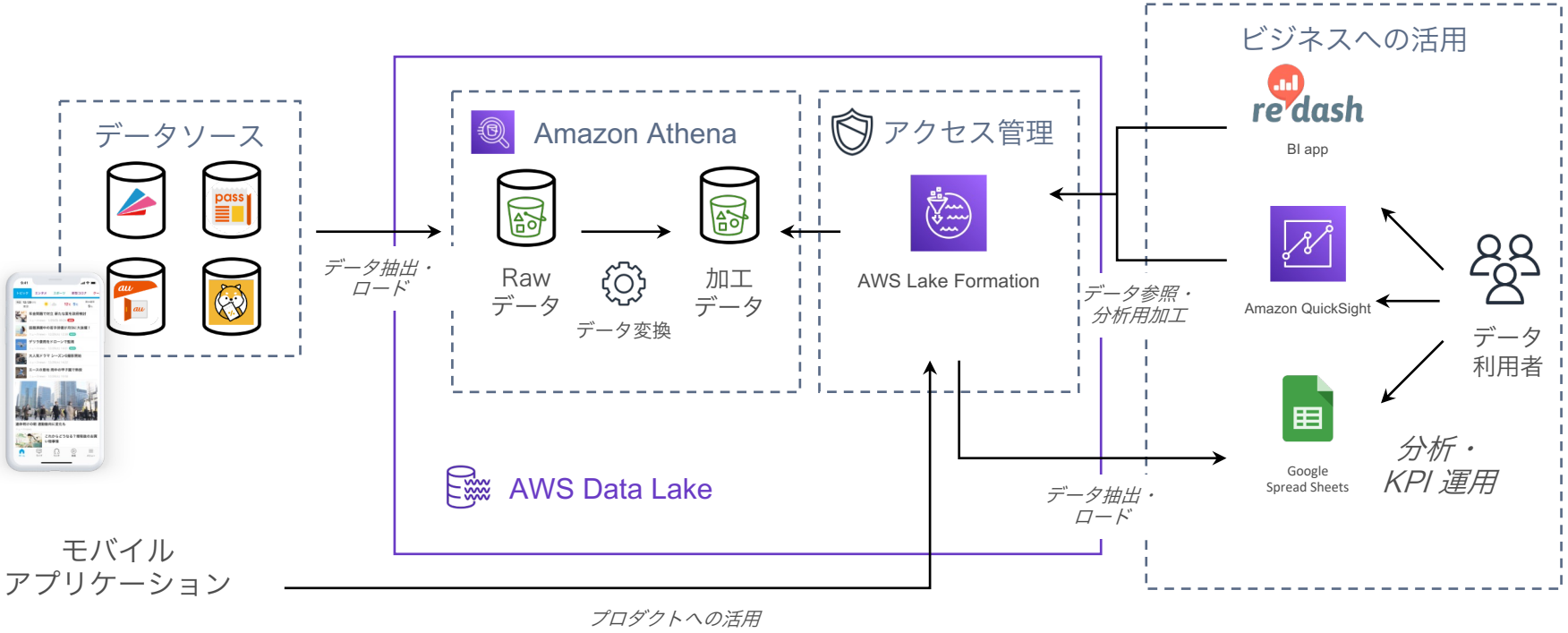
Gunosy

本節の概要：データレイクを用いた組織内の施策的なお話をします

- Gunosy データ基盤の目指す姿とは？
  - 現状のアーキテクチャの紹介
  - ４つの柱と、その施策について

Baikal のアーキテクチャ

Gunosy



データ基盤の目指すすがた

Gunosy

データの一元管理による
車輪の再発明防止

不正や誤操作を防ぐ
データガバナンス

組織横断な
データ基盤開発の民主化

幅広いユーザーに向けた
分析の民主化

データ基盤の目指すすがた

Gunosy

データの一元管理による
車輪の再発明防止

- すべてのデータを一つのデータ基盤に集約し、
  アクセスと利用を容易にする
- 部門間のデータ共有を促進し、**取り組みの重複を防ぐ**

不正や誤操作を防ぐ
データガバナンス

- 粒度の細かい権限管理により、
  誰がどのデータにアクセス・変更できるかを管理
- データ品質を保証し、**誤操作や不正利用を未然に防ぐ**

データ基盤の目指すすがた

Gunosy

組織横断な
データ基盤開発の民主化

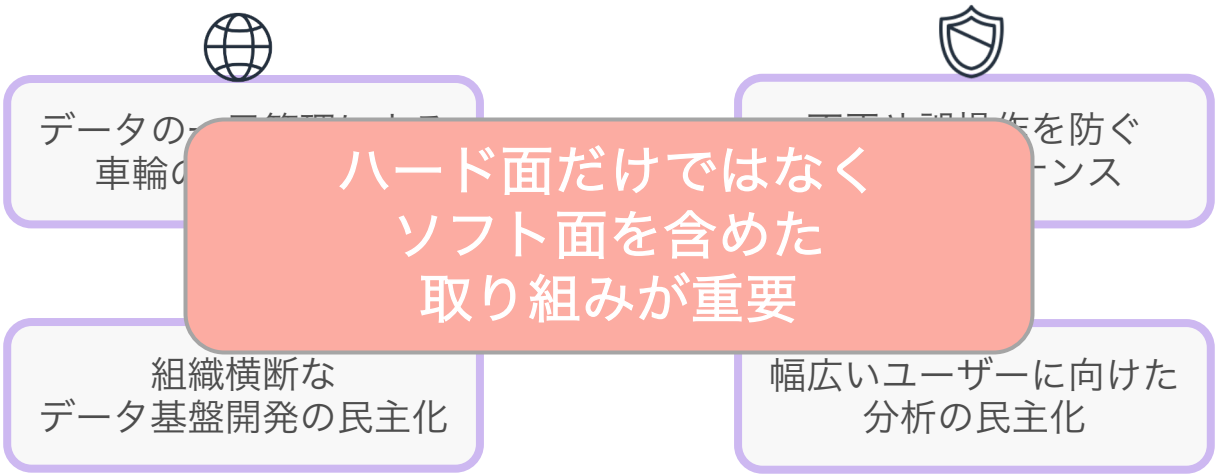
- **各チームがデータ基盤開発に参入しやすい環境**を整備
- データドメインに近いチームが自身で分析データを管理することで、組織全体のアジリティを高める

幅広いユーザーに向けた
分析の民主化

- 分析担当でないユーザーも利用できるツールを提供
- 分析や開発の担当メンバーに対しては**より高度なデータ活用のための環境**を提供

データ基盤の目指すすがた

Gunosy



Baikal における取り組み

Gunosy

1. AWS LakeFormation による横断管理
2. Amazon Athena での SQL によるデータ変換
3. Athena View を IaC で管理
4. 他チームの開発参加支援
5. Amazon QuickSight による開かれたデータ基盤
6. Redash による詳細な分析基盤

Amazon LakeFormation による横断管理

Gunosy

- AWS アカウントをまたいでデータへの権限を付与
  - Amazon Lake Formation の導入で一元管理された
    データカタログを他アカウントに共有できるようになった

- 粒度の細かい権限管理に基づくデータガバナンスの提供
  - テーブルやカラム粒度での権限設定が可能
  - ユーザーが安心して利用できるデータ基盤へ

Amazon Athena での SQL によるデータ変換

Gunosy

- Amazon Athena の CTAS 機能によるデータ変換の提供
  - SQL を記述するだけでデータの変換を実装できる
  - 分散処理などの専門知識は不要で開発に参入できる
  - サーバーレス実行可能なので管理コストが低い

- 実行結果のコスト可視化
  - Amazon Athena 実行ログから実行結果のコストを可視化
  - Slack でコストの高いクエリが実行された場合は通知することで、効率的なクエリの啓蒙

Athena View を IaC で管理

Gunosy

- Athena View を Terraform で記述
  - 頻度の高い分析を共通化し、再利用することができる
  - レビューや CI によって SQL やメタデータの品質を保ちやすい

- Amazon QuickSight の取り込み対象を View に限定
  - Amzaon QuickSight で定常的にみるデータに関しては、特にデータの品質を保つようにした
  - Amazon QuickSight 側でのデータ変換を制限することで、似たようなクエリやデータ変換の乱立を防止する

他チームの開発参加支援

Gunosy

- **データ基盤チーム以外に向けたドキュメントの整備**
  - データのドメインに近いチームが自身で分析に向けたデータを整備できる状態を目指す
- **ペアプロ・モブプロの開催**
  - ペアプロ・モブプロを開催することで、チーム外のメンバーにも開発方法の知見の共有
  - DB からデータ取得など定型的な内容は容易に整備できるようコードを工夫

QuickSight による開かれた分析基盤

Gunosy

- SQL を使わずインタラクティブな視覚化を提供
  - ドリルダウンによってデータの深堀りを可能に
  - 定型的な分析は日々ダッシュボードで確認できるようにし、意思決定を効率化する

Redash による詳細な分析基盤

Gunosy

- SQL を利用した詳細な分析環境の提供
  - より深ぼった分析や可視化のために Redash による分析基盤を提供
  - A/B テスト特有のメトリクスの確認や KPI の異常値調査など、突発的で詳細な分析を可能に

まとめ

Gunosy

- **Gunosy におけるデータ活用の背景**
  - Biz/Dev 問わずデータによる意思決定やプロダクトへの活用が定着している
  - データ基盤として、幅広く社内ユーザー（分析・開発双方の）体験を高める事が重要
- **分析環境の民主化によるデータ分析体験の向上**
  - ユーザーのニーズに沿った複数の分析環境を提供
    - Amazon QuickSight, Redash, Google SpreadSheets など
  - SQL を使わずに品質の高いデータにアクセスできるように整備
    - Athena View の活用
- **開発環境の民主化によるプロダクトへのデータ活用体験の向上**
  - データ基盤の開発を委譲できる体制の整備
    - ペアプロ、ドキュメント整備といった施策

③ データの民主化によって得られたもの・これからのデータ基盤とは

データ基盤によって得られたもの・これから

Gunosy

本節の概要：考察を共有します

- データの民主化によって得られたもの
- これからのデータ基盤

データの民主化によって得られたもの

Gunosy

具体的にどのようなことが起きたのか紹介します

- 開発の民主化：各種チーム内にアクティブなデータ基盤開発者がいる
  - Backend, SRE, Data Science, ML
  - 必要に応じてリポジトリに PR を出してくれる（コントラクトの更新など）
    → 発展して、チーム専用のリポジトリを切り出して管理してもらう例もある

- 分析の民主化：DX の加速
  - セールスチームとの salesforce 連携 PJ により数百万円オーダーでコスト削減
  - 部署間で似た指標を見ていたことが判明し、ダッシュボードを共有できた
  - スプレッドシートをやめられた・手動コピペを廃止して自動化できた

これからのデータ基盤

Gunosy

これからどうなるのか・どうしていきたいのか

- さらなるデータ民主化のための開発者体験の向上: ModernDataStack
  - データ変換はテンプレート SQL が主流に：dbt, Dataform 等の台頭
  - キーワード：データリネージュ・データ観測性・Reverse ETL

- データメッシュ：データレイクの次のトレンド
  - マイクロサービス時代のデータ基盤：サービス開発者側がデータオーナーになる
  - データ生成側（サービス側）がサイドカーで分析データサービスを持つ
  - ドメインはサービス側、インフラは SRE が横串でサポート、の類型に見える

Gunosy

情報を世界中の人に最適に届ける