B-012

# データ利活用の加速に向けたアーキテクチャの検討
# Research on System Architecture Design for Accelerating Data Utilization

礒田 有哉†
Yuya Isoda†

## 1. はじめに

近年、オープンソースやオープンデータの普及や多様化に伴い、様々なデータ分析による企業活動の変革が求められており、デジタルトランスフォーメーションによる事業継続力の強化が重要な課題となっている。

また、データ利活用の普及に伴い、同業種／異業種のデータ連携が加速している。同業種連携では、銀行間のデータ共有による不正取引対策で、誤検出を１２％抑制した事例がある。異業種連携では、銀行が持つ個人属性とレンタカー事業者が持つ違反履歴から、交通違反保険をダイナミックプライシングで提供することにより、収益を１．５倍に向上させた事例がある。

これらの背景から、我々は、データ利活用の進展に伴い、データを様々なアプリケーションやサービスに繋げることで、サービスの提供範囲拡大や付加価値の向上が可能と考え、データ利活用の加速に向けた研究開発を推進している。

本稿では、アプリケーションが利用可能なデータを管理し、アプリケーションとデータを利用するための環境構築を自動化するアーキテクチャの検討について報告する。

## 2. 課題

先行調査によれば、データ利活用により生産性や業務効率の改善が期待できる一方、５段階の満足度調査では、「非常に効果があった」と回答した割合は１３．３％であり、８０％以上の企業がデータ利活用の効果に満足できていない[1]。同調査によればデータ利活用の課題として、利用可能なデータが不明、規範的なデータの収集や管理が困難、費用対効果の判断が困難、データや利活用方法の知識不足などが示されている。

これらの調査結果から、データ利活用の課題に起因した費用の増加が費用対効果を圧迫し、満足度を低下させていると考え、データやアプリケーションの知識不足を解消し、データ利活用の検討工数の削減に伴う費用削減により、費用対効果や満足度を改善できると考えた。

### 2.1 従来技術

一般に、データを管理・共有するための仕組みとして、データガバナンス／メタデータ管理／データカタログなどのツールがあり、オープンソースの開発や標準化活動が盛んである。オープンソースの Apache Atlas や OpenMetadata などは、主にデータベースに関連したデータ処理を監視・管理する機能を提供しており、データとデータ処理の関係をデータリネージで表現する。一方、セマンティックウェブに基づくデータカタログでは、オープンソースの CKAN が普及しており、W3C DCAT で定義されたデータモデルに従ってデータを管理・共有する機能を提供している。

これらのツールにより、アプリケーションやデータの把握や理解を促せるが、これらの情報だけでは、データ利活用環境を構築することは困難である。例えば、どこに・どうやって環境を構築すればよいか分からない。

データを監視・管理・共有する仕組みについては、様々な取り組みがある一方で、データ利活用を支援する仕組みは少ない。例えば、仮想化技術の Kubernetes や Helm を用いることで、データベースやアプリケーションを構築できるが、利用対象のデータをデータベースに格納する必要がある。このとき、データとデータベースのスキーマ定義が完全一致する場合は、データをデータベースにインポートするだけでデータ利活用を開始できるが、スキーマ定義が完全一致することは極めて少ない。このため、利用者は、データベースのスキーマ定義に準拠したデータを準備する必要があり、適切なデータをインポートするためのデータ・アプリケーション・データ処理に関する知識が求められる。

このように、データ利活用者には、データ／アプリケーション／環境構築／データ処理などの幅広い知識が求められ、このような知識を有する担当者は極めて少なく、データ利活用の課題となっている。

これらの課題から、本稿では，データ／アプリケーション／環境構築／データ処理などの知識不足に伴いデータ利活用を開始できない課題を解決するために、データやアプリケーションの探索自動化やデータ利活用環境の構築自動化を実現するためのアーキテクチャについて検討した。

## 3. 提案

本章では、データ利活用を支援するアーキテクチャとしてスキーマハブを提案する。スキーマハブでは、データとアプリケーションの入出力形式（スキーマ）の統合管理により、利用可能なデータとアプリケーションの組合せを特定し、データとアプリケーションの環境構築を支援するテンプレートを運用する。これにより、データ利活用者は、利用したいデータもしくはアプリケーションを指定するだけで、利用可能なデータやアプリケーションの候補を把握でき、選択したデータとアプリケーションを活用したデータ利活用環境を、テンプレートを利用して簡単に構築できる。本章では、スキーマハブの検討背景やパブリッククラウドを用いた設計概要について提案する。

### 3.1 検討背景

スキーマハブは、データウェアハウス／データレイク／データレイクハウス／3 factor app などのアーキテクチャに基づいて設計しており、本節では、これまでのアーキテクチャの進化と、スキーマハブの検討背景について述べる。

これまでのアーキテクチャの進化を図１に示す。データウェアハウス（図１．A）の時代では、定期的にファイル（CSV など）をデータベースにインポートすることで、データを利用していた。その後、IoT デバイスの普及による多種多様なデータ活用の要求から、システムの複雑化やリアルタイム性が課題となり、データレイク（図１．B）が

†株式会社 日立製作所，Hitachi, Ltd.

考案され、構造／半構造／非構造のデータ形式をオブジェクトストレージで統合管理し、リアルタイム処理とバッチ処理の共存による適時適切なデータ活用を可能にした。また、データベースをオブジェクトストレージとスキーマに分離するサーバーレスデータベースの技術確立により、データ活用の即時性が向上した。更に、データレイクハウス（図１．C）では、サーバーレスデータベースでトランザクション制御を実現するファイルフォーマット（Apache Iceberg）の確立により、あらゆるデータを一元管理できる見通しが立った。これらの技術により、多種多様なビッグデータの統合管理が実現可能になった。

また、多種多様なビッグデータを活用するためのインターフェース技術も進化した。従来の REST API では、多種多様なビッグデータを通信量少なく利用するためには、アプリケーションの用途ごとに API を開発する必要があり、開発工数の増加とシステムの複雑化を招く課題があった。この課題を解決したの GraphQL であり、API 向けのスキーマ定義により、API を追加開発することなく、アプリケーションは必要十分なデータを取得可能になった。この GraphQL とデータベースを統合したアーキテクチャが 3 factor app（図１．D）であり、API のスキーマとデータベースのスキーマの統合管理により、データの整合性を担保する。また、データ処理においてもスキーマを利用することで、データの整合性を管理でき、データ品質の向上が可能になった。

このように、データベースやインターフェースの技術革新により、データやアプリケーションなどのコンポーネントをスキーマで分離し、システム設計や開発の独立性を向上させてきた。これらの潮流に沿って、スキーマを中心とした疎結合なデータとアプリケーションの管理方式として、スキーマハブを提案する。

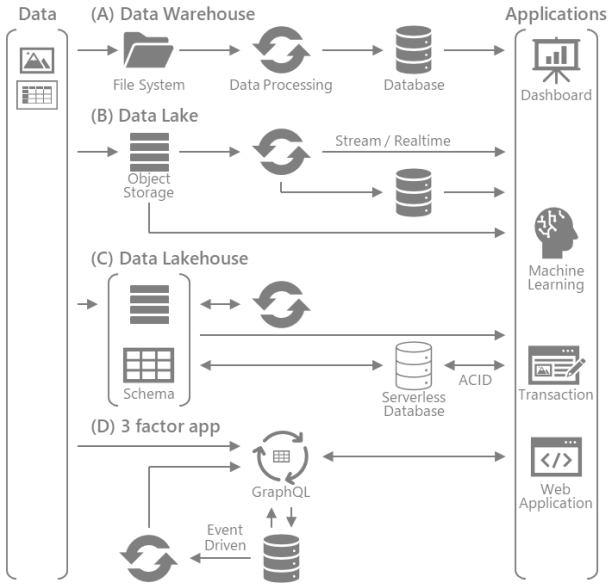


図 1　アーキテクチャの進化

### 3.2 設計概要

スキーマハブの設計思想を図２に示す。従来、アプリケーションやデータ処理の入出力形式は、（API を除き）未定義／未管理なことが多く、再利用が困難であった。本提案では、アプリケーションやデータ処理の入出力形式をスキーマに関連付けて管理することで、データやデータベースとの関係性を明らかにし、データカタログなどの既存サービスと連携可能なアーキテクチャを考案した。

パブリッククラウドを用いた実装例を図３～４に示す。図３は、データ利活用環境を自動構築するためのテンプレート管理について記載しており、テンプレートの関係性を有効／無向グラフで管理する。例えば、データ処理では、入力と出力でデータ形式が異なることがあるため、有向グラフでデータ処理とスキーマの関係を管理する。図４は、データ利活用環境の構築順序を示しており、コンポーネントの依存関係に基づきデータから順にテンプレートをデプロイする。

このように、スキーマをテンプレート化し、データやアプリケーションのテンプレートと連携した疎結合なコンポーネントを自動構築することがスキーマハブの特徴である。

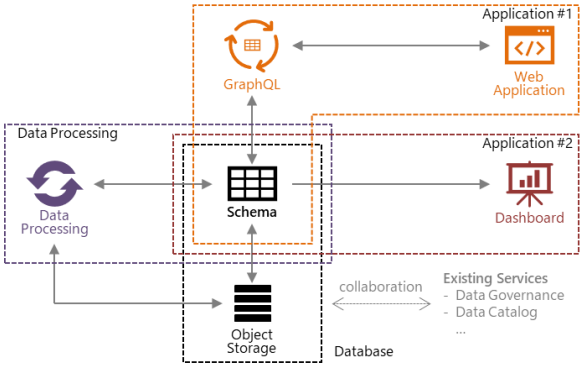


図 2　スキーマハブの設計思想

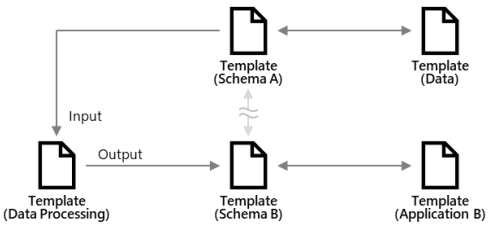


図 3　スキーマハブのテンプレート管理

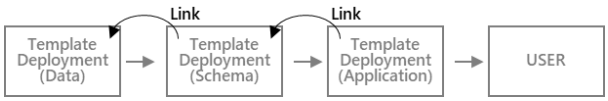


図 4　テンプレートを用いた構築順序

## 4. おわりに

本稿では、データ利活用の満足度向上を実現するために、データ／アプリケーション／環境構築／データ処理などの幅広い知識が不要なデータ利活用基盤として、スキーマハブを提案した。スキーマハブでは、データやアプリケーションの入出力形式（スキーマ）をテンプレート化し、データやアプリケーションをスキーマのテンプレートに関連付けて管理することで、IT 知識不要なシステム構築を実現した。今後は、テンプレート間の関係性の自動推論や自動補完する技術を確立し、全体全のコンポーネント接続によるユーザビリティ向上をめざす。

### 参考文献

[1] 総務省、“情報通信白書令和 2 年版、”2020/08.