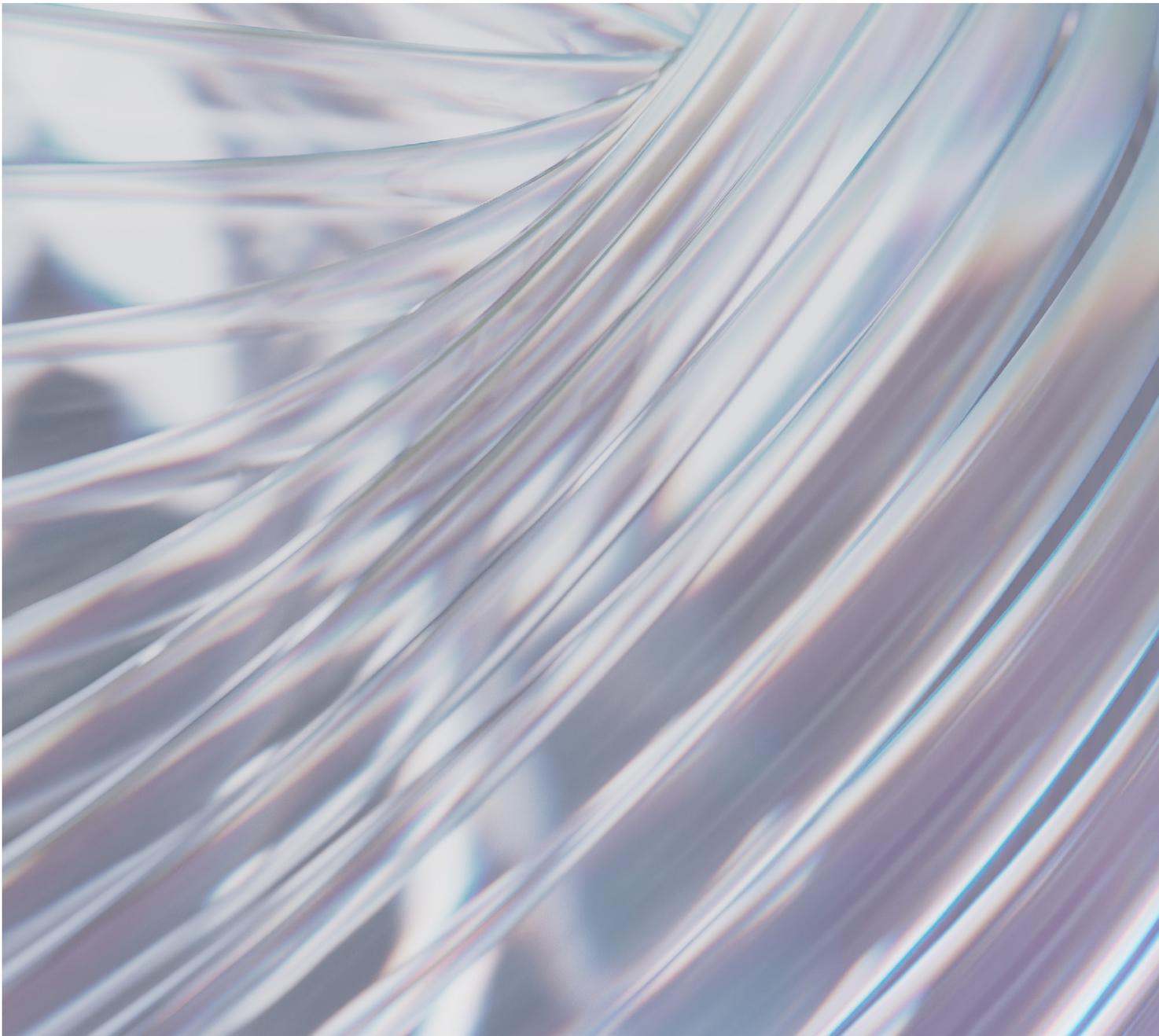


# Why Open Dataspaces:

## 設計思想とアーキテクチャパラダイム

Leveraging distributed data management  
for inter-organization and global interoperability

April, 2026



# *Why Open Dataspaces:*

## 設計思想とアーキテクチャパラダイム

*Design Philosophy and the Architectural Paradigm*

Leveraging distributed data management

*for* inter-organization & global interoperability

### **KEY WORDS:**

distributed data management, data mesh, dataspace, semantic layer, ontology, usage control, Artificial Intelligence, Agentic AI, context engineering, Domain-Driven Design

## 発行者

独立行政法人情報処理推進機構

デジタルアーキテクチャ・デザインセンター

## 著者・編集者

津田 通隆 / Michitaka TSUDA:

Open Data Spaces Chief Architect (最高設計責任者)

## コピーライト

独立行政法人情報処理推進機構

CC BY 4.0 (<https://creativecommons.org/licenses/by/4.0/>)



## 本書の目的

本書は、国や組織ごとの多様性を尊重する、オープンでスケーラブルな分散データマネジメントの技術コンセプトである「Open Data Spaces (ODS)」の設計思想、そしてその中核となるアーキテクチャパラダイムを提示するものである。

はじめに。

## Agentic AI 時代の到来とダークデータの可能性

AI が世界の産業構造そのものの転換を進める中、2026 年は、多くのテクノロジー市場関係者にとって「**データ枯渇元年**」と呼べる転換点である。AI 技術の発展を語るうえで、モデル性能向上の鍵として、(1) **モデル規模 (パラメータ数)**、(2) **学習データ量**、(3) **計算能力**という三つの変数は欠かせない。「**スケーリング則 (Scaling Laws)**」として経験的に観測されるこ

れら三つの変数について、実務的観点では学習データと計算資源の確保が AI 戦略の両輪となっている。近年の市場動向を見ると、計算資源の制約に対しては半導体供給、データセンター建設、電力と通信の契約・調達までが激しい競争として顕在化している一方で、**学習データの供給制約**はまだ社会に十分に認識されていない。

## インターネットデータの有限性

現代の主要な大規模言語モデル (LLM) は、ウェブページ、SNS 投稿、デジタル書籍・論文など、インターネット上のパブリックテキストデータを基礎とする学習データセットを利用する。しかし、インターネット上のデータストックは増え続けているように見えても**有限の成長速度しか持たない**。もし、モデルの学習消費速度がこの増加量を上回れば、学習に必要な質・量のデータは相

対的に不足する。これが「**データ枯渇問題**」である。Epoch AI. (2024) の推計によれば、ベンチマークとなる主要 LLM がこのままの速度で学習を進めた場合、特に**高品質データが 2026 年から 2032 年の間に枯渇する可能性**が示されている (図 1)。この枯渇とは、データ自体の消失ではなく、**学習に耐えうる新規データ供給速度**が追いつかなくなる状況を指す。

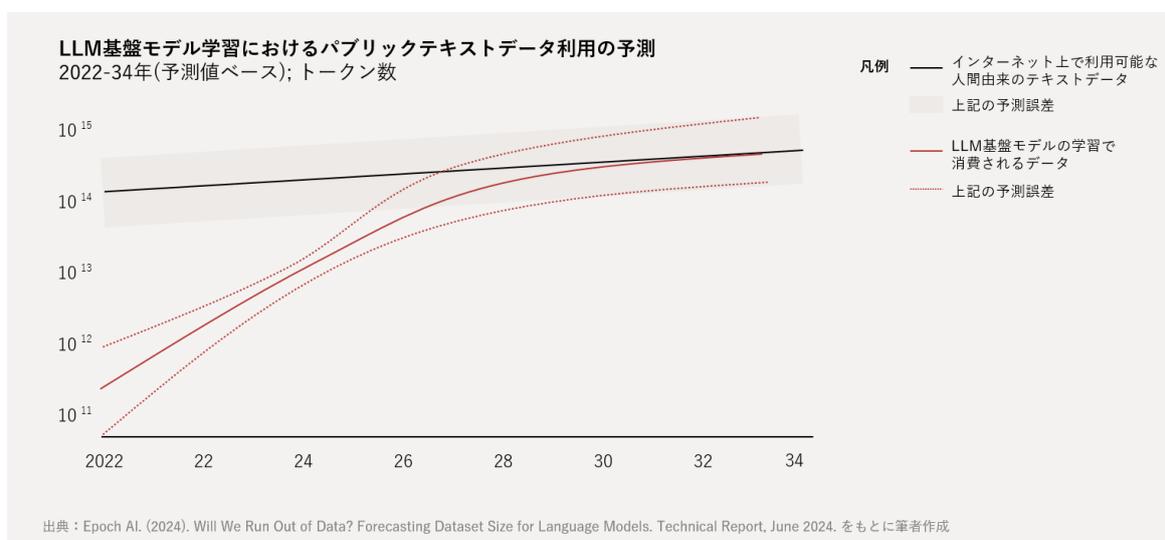


図 1 LLM モデル学習におけるパブリックテキストデータ利用の予測

## データ供給源としてのリアルデータと合成データ

データ枯渇に対して補完可能な供給源としては、以下の2つに大別できる:

- **リアルデータ (Real Data)**: 実世界の観測・取引から得られるデータ (業務データ、設計・図面データ、IoTのセンシングデータ等)
- **合成データ (Synthetic Data)**: モデルや統計的手法により人工生成されたデータ

合成データは供給総量を拡大しやすいものの、再帰的な利用によって**分布の多様性が失われることによる性能劣化リスク (モデル崩壊問題)**があり、リアルデータの供給は必須である。一方で、リアルデータは、プライバシー保護や営業秘密などが含まれることが多く、取得と共有の障壁が高い (図 2)。

	リアルデータ	合成データ
データ品質	低～高 収集・処理工程の精度に依存	低～高 プライバシー保護とトレードオフであり、生成方法に強く依存
バイアス	有 収集過程に依存	有 統計モデル・アルゴリズムに依存
プライバシー保護	低～中 コストと技術的制約	設計次第 データ品質とトレードオフ
経済性 (生成/利用コスト)	中～高 利用可能にするための処理と利用コストが高い	中～高 データ品質とトレードオフ
	▼ プライバシーや営業秘密などの保護がボトルネック	▼ スケーラビリティはあるがモデル崩壊問題がボトルネック

図 2 リアルデータと合成データとの比較

したがって、リアルデータと合成データは**単純な優劣やトレードオフではなく、用途に応じた相互補完の関係**にある。データの最適な供給戦略は、元手となるリアルデータを保持したまま、両者を

適切に組み合わせることであり、再帰汚染の影響を最小化するデータ戦略のデザインが不可欠である。

## リアルデータにおけるダークデータの比率

そもそも、リアルデータはどの程度存在しているのだろうか。国立研究開発法人新エネルギー・産業技術総合開発機構の調査 (NEDO, 2025) では、2025年時点において世界中で創出・取得・複製・消費されるリアルデータの量は、年間約

175 ZB (Zettabyte) と推計されている。また、そのうち、約 71ZB が消費者由来、残り約 104 ZB がエンタープライズ由来であることが分かっている (図 3)。

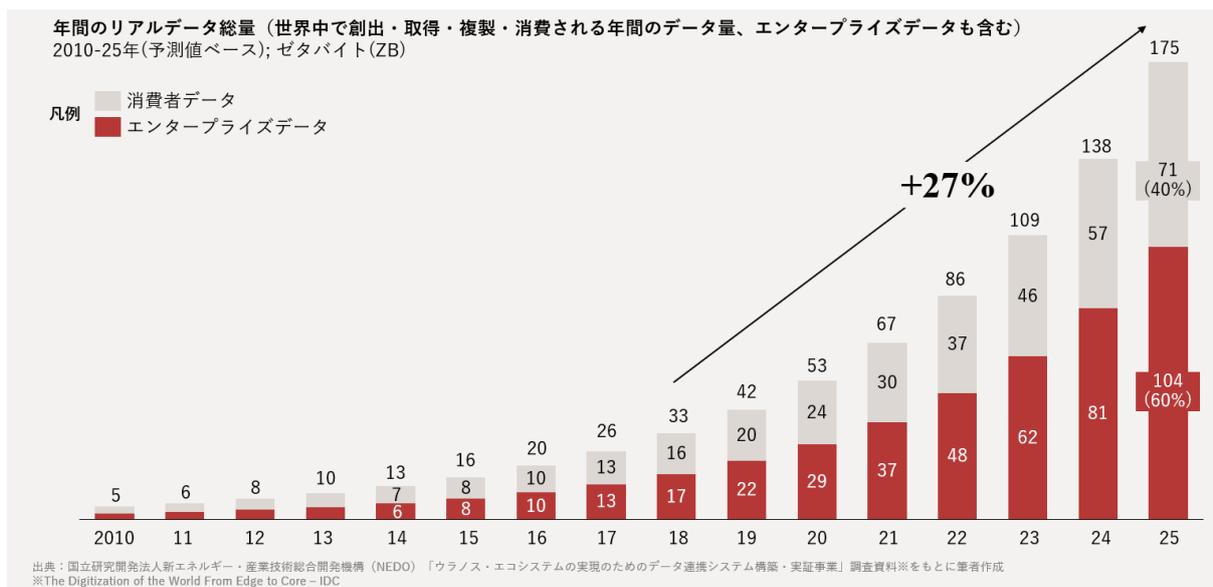


図 3 リアルデータの総量と消費者・エンタープライズ比率

さらに、同調査によれば、複製データを除去したリアルデータの有効量としては約 17.5 ZB が存在し、このうち約 16ZB がインターネット非公開の「**ダークデータ (Dark Data)**」として企業内に留まっていると推定される (図 4)。このダークデータは、**潜在的な学習や推論の資源としての価値**

**を持つものの、そのままでは利用に供せないケースが多い。**ダークデータは偏りや品質を汚染する要素を含むことが多く、また、企業内のデータの多くはビジネス上の目的や文脈に即したフィルタリングやキュレーションが前提となるためである。

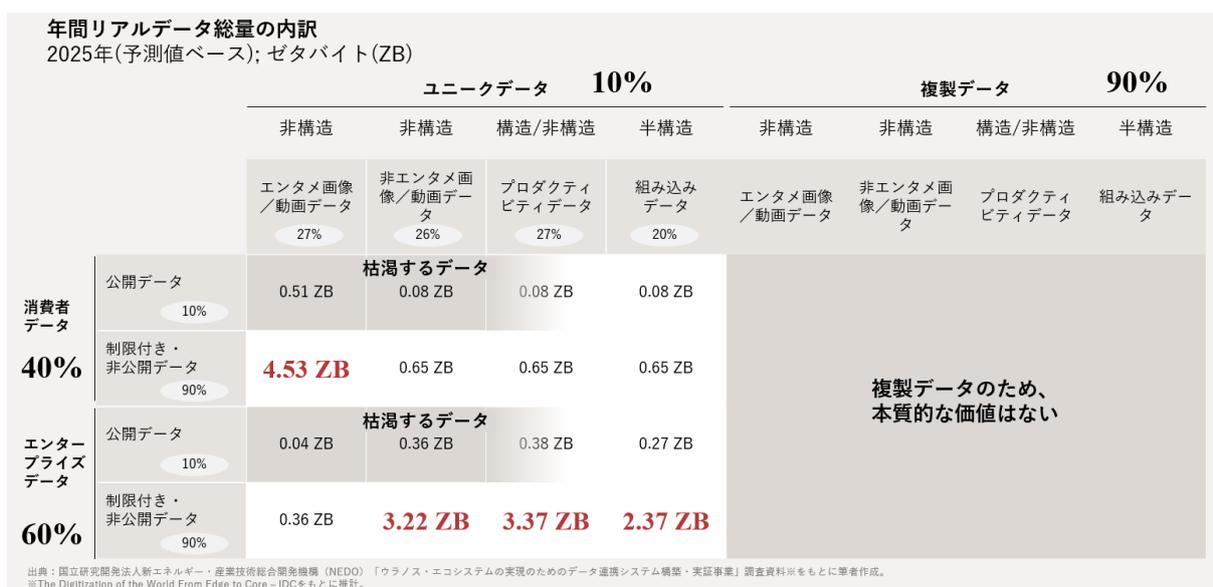


図 4 リアルデータ総量の内訳

## GIGO とドメインコンテキスト、分散するデータ

リアルデータを事前学習（pre-training）段階で学習資源化、あるいは事後学習（post-training）で推論利用する際の最大の障壁は、単純な量ではなく**品質**である。自社の手の内にビッグデータが蓄積されているからといって、それが必ずしも AI、BI（Business Intelligence）、DI（Decision Intelligence）に有用であるとは限らない。利用目的と整合性のないデータを大量に取り込めば、むしろ利用結果の精度や汎用性を損なう可能性がある（Garbage-In, Garbage-Out : GIGO）。そのうえ、本質的に価値を生まないランニングコストが財務を圧迫する。そして、その品質は組織内の自律的な責任単位である「**ドメイン**

**(Domain)**」に依存する。データの品質水準を決定する大きな要素のひとつである**ドメイン固有の目的や運用、慣行などの明示的・暗黙的な文脈の総体である「コンテキスト (Context)」**は、ドメイン単位で分散して存在し、横断的な均質化は困難を極める。

このような背景から、Agentic AI 時代のデータマネジメントについてのアーキテクチャパラダイムの検討では、主に以下のような論点を念頭に実施することが重要である：

- **ダークデータにドメインコンテキストをどのように与えるか**
- **ドメインコンテキストが付与されたデータについて、いかに将来キャッシュフローを生み出す経営資本として機能させるか**
- **そして、本質的に企業や国境を横断して分散して存在するデータとドメインコンテキストをどのようにマネジメントすべきか**

コンテキストは Agentic AI 時代におけるデータの本質であり、それはドメイン固有の財産である。ドメインオーナーがデータにコンテキストを与え、それを「プロダクト（商品）」として提供し、その利用を制御することが企業価値の向上につながる。このようなデータの企業資本としての戦略、そして、そのデータの価値を最大化するソフトウェア戦略の実行こそが、ソフトウェアが企業価値を定義し、そのソフトウェアの競争優位性をデータが規定する「**データに飲み込まれる世界 (Data is Eating the World)**」における、最も本質的な生存戦略となるであろう。

## 免責事項 (Disclaimer)

「Open Data Spaces (ODS)」は、国や組織ごとの多様性を尊重する、オープンでスケーラブルな分散データマネジメントの技術コンセプトの名称である。また、本書で利用する一般名称としての「Open Dataspaces」は、米国のデータスペース原著論文 (Franklin et al., 2005; Halevy et al.,

2006) 及びデータメッシュ (Dehghani, 2019; Dehghani 2022) を中核として、民間企業・団体と連携した研究開発・商用水準での検証を経ながら設計された新世代の分散データマネジメント技術及びそれを構成する概念を指す (表 1)。

表 1 ODS と Open Dataspaces の用語表現について

表記	意味
Open Data Spaces (ODS)	国や組織ごとの多様性を尊重する、オープンでスケーラブルな分散データマネジメントの技術コンセプトの固有名称。本文中では略称の ODS を使用する。
Open Dataspaces (一般技術名称)	本書内で利用するデータスペース原著論文とデータメッシュを源流とする、新世代の分散データマネジメント技術及びそれを構成する概念を指すための一般名称。固有名称の ODS とは区別する。

なお、ODS は、2016 年頃から Fraunhofer 研究所などを中心に欧州で研究開発が進むデータスペースアーキテクチャ及び関連するプロジェクトとは出自が異なる取組である。従って、Open Dataspaces は International Data Spaces や Eclipse Data Spaces をはじめとした技術仕様とは異なる指針に基づいて設計されている。現在 ODS の活動において独立行政法人情報処理推進

機構 (IPA) が ODS の技術的な設計統括を行うとともに、グローバルプロモーションの事務局を務めている。技術コンセプトとしての ODS の活動や最新情報、分散データマネジメント技術としての Open Dataspaces に関連するドキュメントやソースコードについては、公式ホームページを確認いただきたい。

# 目次

はじめに. Agentic AI 時代の到来とダークデータの可能性 .....	5
インターネットデータの有限性.....	5
データ供給源としてのリアルデータと合成データ .....	6
リアルデータにおけるダークデータの比率.....	6
GIGO とドメインコンテキスト、分散するデータ .....	8
1. 集約から分散へ: データマネジメントのパラダイム変遷.....	11
なぜ、分散データマネジメントなのか? .....	11
Push and Ingest パラダイムの功罪とデータメッシュ .....	12
2. 部門間から組織間へ: 米国での「データスペース」誕生の背景 .....	15
なぜ、データスペースなのか? .....	15
Classical Dataspaces とデータメッシュから何を学んだか? .....	16
3. Open Dataspaces の特筆性と設計指針、技術原則 .....	19
なぜ、Open Dataspaces か? .....	19
Open Dataspaces の設計指針 .....	20
4. アーキテクチャパラダイムの最小単位: Architectural Quanta/Quantum.....	21
Open Dataspaces の前提と DPQM の強み .....	22
基本用語と関係性.....	23
5. 柱 1: Ontology and Semantic Interoperability (OSI) .....	25
Ontology as a Product—データモデルから情報モデルを分離する .....	25
Semantic and Ontological Interoperability—推測から知識へ .....	27
6. 柱 2: Data Addressability & Discoverability (DAD) .....	30
Data Addressability—データの「存在」と「同一性」を保証する .....	30
Discoverability—データの関係性を提供する緩やかな検索機構 .....	33
補論: Data Trust と Data Trustworthiness について.....	36
7. 柱 3: Identity and Usage Control (IUC) .....	37
Identity—Trust by Design.....	37
アクセス制御—Graph-to-Graph Control.....	39
利用制御—制度・技術非依存の柔軟な権利関係の選択肢.....	41
補論: 電子契約行為について .....	44
補論: 精算・課金/決済行為について .....	45
8. サービスモデル .....	47
おわりに.....	49
Bibliography .....	50
Acknowledgement.....	51

# 1. 集約から分散へ: データマネジメントのパラダイム変遷

## なぜ、分散データマネジメントなのか？

この新たなパラダイムの必要性を理解するためには、非分散 (non-distributed)、つまり中央集権型 (centralized) のデータマネジメントの歴史とその課題の理解が不可欠である。

21 世紀に入り、ソフトウェアが扱うデータは爆発的に増大し、ビッグデータの時代が幕を開けた。このビッグデータの時代において、データマネジメントの問題関心は、量 (Volume)、種類 (Variety)、速度 (Velocity) と変遷してきた。2001 年、データアナリストの Doug Laney 氏が、「The 3Vs」という基本的なビッグデータの特徴を説明する概念を提唱したことは有名である。2000 年代初頭～2010 年代は、3Vs を対処するため、例えば以下のような技術がデータマネジメント領域を牽引してきた：

- ・ **Volume の問題 (2004-06 年頃)**：代表的な

大規模データ処理エコシステムである Apache Hadoop における分散ファイルシステム HDFS、オブジェクトストレージとしての Amazon S3 など

- ・ **Variety の問題 (-2009 年頃)**：スキーマが揺れることを前提にした NoSQL を代表する Document-oriented DB (MongoDB など) や Key Value Store (Redis など)、Wide Column Store (Apache Cassandra など)をはじめ、半構造のシリアルイゼーションとしての JSON、データカタログや ETL ツールの拡大など
- ・ **Velocity の問題 (-2012 年頃)**：低遅延にデータを処理するため、イベント駆動のメッセージング処理を代表する Apache Kafkaをはじめ、ストリーム処理の Apache Flink、大規模リアルタイム処理の Apache Spark など

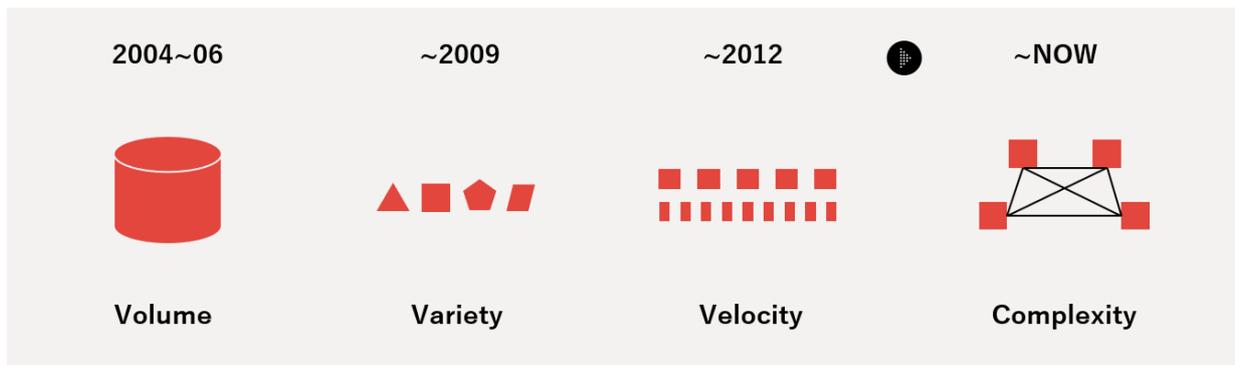


図 5 データマネジメントの問題関心の変遷

このような 3Vs の時代を乗り越え、現在我々が多くの場面で直面するのは、**複雑性**

(Complexity) の問題である (図 5)。この Complexity の問題は、技術的な側面のみならず、産業・ビジネスの側面、社会的・組織的な側

面、法制度・契約的な側面を含む多面的なものである。技術的な側面に着目すると、3Vs、そして Complexity の問題を解決するために、データマネジメントの領域では古典的な Data Base Management System (DBMS) を基盤として、

第一世代の Data Warehouse (DWH)、第二世代の Data Lake (DL)、第三世代の Data Lakehouse (DLH) と発展してきた (図 6)。

世代を経るごとに、アーキテクチャが想定するデータの利用方法も、アプリケーションベースの BI や DI に閉じず、AI への親和性が高まってきて

いる。これは、昨今の学習・推論ニーズを反映したものであるといえよう。このような技術的革新により、ビジネスの観点でもデータ利用者の Time to Value (TtV) が大きく改善しており、データドリブンな意思決定や業務オペレーション、サービスなど、ソフトウェア産業全体の市場規模を押し上げることに寄与している。

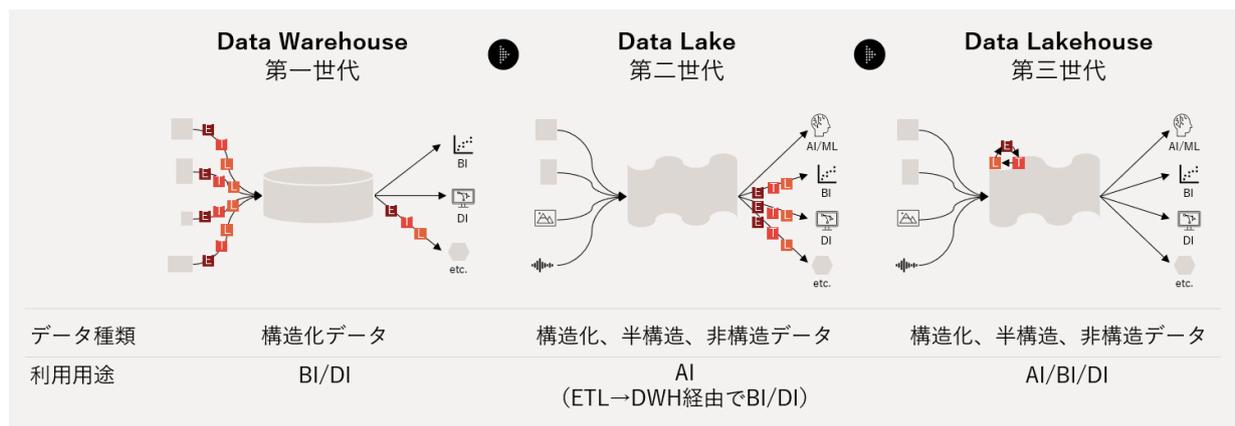


図 6 データマネジメントシステムのアーキテクチャ世代変遷

## Push and Ingest パラダイムの功罪とデータメッシュ

しかし、このような従来のデータマネジメントのアプローチは、組織内外のあらゆるデータを一箇所に集約して保存・処理するという「集約 (Aggregation)」中心のパラダイムからは脱却できていない。

このような伝統的な「Push and Ingest」と呼ばれるパラダイムは、シンプルなドメイン構成で利用目的が限られた組織でこそ非常に機能するものの、ドメインが多岐にわたり、データソースも利用者も多様な組織になった場合、データの増加により技術的な複雑性が増し、利用コストも増大する。そしてそれを扱う組織も複雑性を増大させ、企業がスケーラビリティの限界を迎えることが懸

念される。Complexity の時代には、所在が散逸し、組織の中で分断され、異なる構造や意味で存在するダークデータを適切なオペレーションコストでマネジメントするための解決策が必要である。

このような環境の中で、現代の Push and Ingest パラダイムが抱える問題に対して、「Serving and Pull」という対照的なパラダイムとして [Zhamak Dehghani 氏により考案された分散型の社会技術的アプローチ \(Decentralized Sociotechnical Approach\)](#) が「データメッシュ (Data Mesh)」(Dehghani, 2019; Dehghani 2022) である (図 7)。

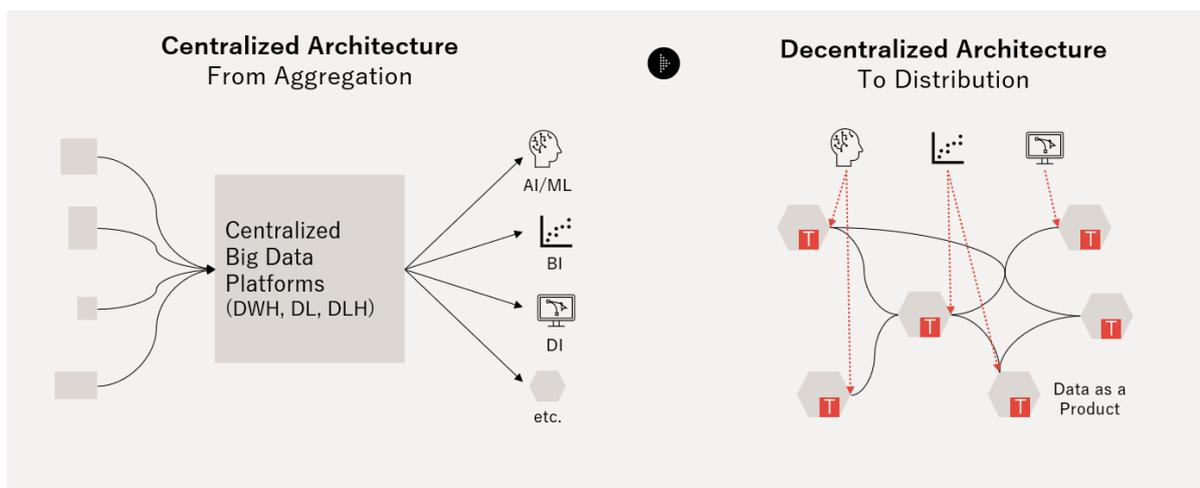


図 7 分散データマネジメントアーキテクチャ「データメッシュ」の概要

データメッシュは、「(1) Domain Ownership」、  
「(2) Data as a Product」、  
「(3) Self-Serve Data Platform」、  
「(4) Federated Computational Governance」の4つの原則のもとに、分散データマネジメントを実現する（図 8）。ここでは、データメッシュのアーキテクチャ

パラダイムや思想について詳述はしないが、データメッシュは、データを部門横断で再利用可能なプロダクトとして捉え直した点で、従来の中央集権型アーキテクチャに対する重要な転換点であった。

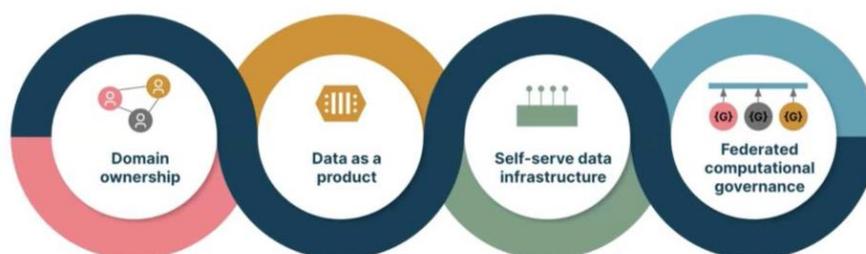


図 8 データメッシュにおける4原則

このパラダイム転換により、もっとも特筆すべきなのは、データメッシュが「データの責任を、中央のデータ基盤から業務ドメインへと戻した」点にある。企業は、技術的な観点から中央集権型のデータ基盤に最適な組織設計を徹底した結果、Biz/Dev/Ops を分業することが伝統的に行われてきた。しかし、これは副作用として組織的な分

断を招来し、データは集約・加工される過程でコンテキストを失うことになる。データ利用者はその意味や制約、品質、利用条件をコードやドキュメントから推測することを求められた。データメッシュは、データを生み出すドメインオーナー自身（例：広報部門、HR 部門、商品開発部門 等）が、データを単なるオペレーションの

副産物ではなく、明示的に設計・提供される「**プロダクト（商品）**」として扱う。この発想の転換により、Biz/Dev/Ops の融合が組織設計の最適解となり、データ活用は中央集権的な調整や事前統合に依存せずとも、組織内でスケールしうるものとなったのである。

この、データの分散という技術的な Complexity と、Biz/Dev/Ops の分断という組織的な Complexity を紐づけて、アーキテクチャパラダイムとして示したということが、まさにデータメッシュが社会技術的アプローチを名乗る資格そのものであり、産業界から高く評価される所以である（図 9）。

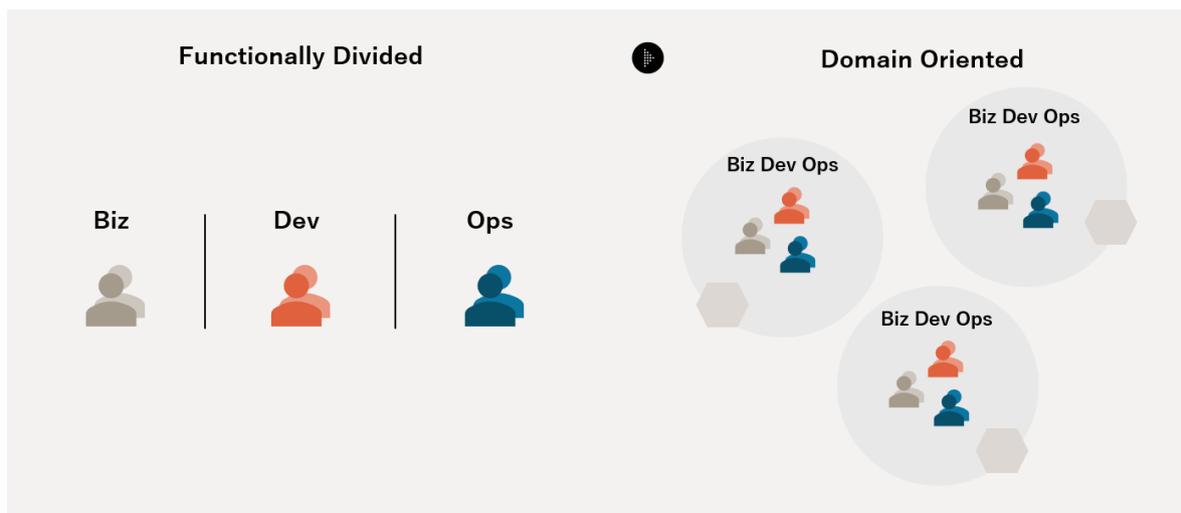


図 9 ドメイン駆動型アプローチに伴う Biz/Dev/Ops の融合

ただし、データメッシュもデータマネジメントにおける銀の弾丸ではない。データメッシュは「組織内」での部門間連携を暗黙の前提として提唱されたが、組織境界を越えた場合に顕在化する **Governance Complexity の問題**には十分に答えきれていない。

Governance Complexity の問題は、(1) データが増え、異質なソース・形式・識別・意味のまま組織内外に散逸する技術的な側面 (2) データの責任をどこが持つかが曖昧になることで生じる組織的な側面、そして (3) 組織・国境を横断する際

に、アクセス制御・利用条件・契約慣行が絡み合う権利的な側面、などが相互に絡み合って顕在化する横断的な性質を持つ。これは、近年広がりを見せつつある分散データマネジメントの世界で取り扱われる最も典型的な課題である。

では、組織境界を越えた場合の Governance Complexity の問題意識に答える技術的な拠り所はないのだろうか。それに示唆を与えるのが、データメッシュが生まれる 14 年前に米国で提唱された「**データスペース (Dataspace)**」という概念である。

## 2. 部門間から組織間へ: 米国での「データスペース」誕生の背景

### なぜ、データスペースなのか？

そもそも、データスペースとは何なのか？前章のとおり、データスペースの概念的な定義は古く、2005年に遡る。「データスペース

(Dataspaces)」は、組織に分断 (Siloed) された異質なデータコレクション (HCoD: Heterogeneous Collection of the Data) を扱うため、当時のデータベースマネジメントシステム (DBMS: Database Management Systems) 技術を補完する概念として、UC Berkeley の M. Franklin、Google の A. Halevy、Portland State University の D. Maier らによって提唱された。

(データスペースを名乗る活動は様々な場所で乱立しており、読者の混乱回避のため、以降区別を行う際には、米国で最初に提唱された概念を「Classical Dataspaces」と呼称する。)

Franklin et al. (2005)、Halevy et al. (2006) の当時のスコープは、DBMS の「(1) アーキテクチャとしての問題」と、それに伴う「(2) 開発者が直面する問題」に集約される：

1. **アーキテクチャとしての問題:** データマネジメントにおいて、すべてのデータを伝統的な DBMS や単一のデータモデル、システムに適合させるのが難しくなっていたこと。
2. **開発者が直面する問題:** それに伴い、開発者が散逸したデータを扱うため、低水準 (low-level) のデータマネジメント (検索・クエリ機能、ルールの適用、整合性制約、命名規則、来歴追跡、アベイラビリティ・リカバリ・アクセス制御の提供、データとメタデータの更新の管理等) を異質なデータコレクションに対して都度対処する必要があること。

この問題を解決するために「データマネジメントの新たな抽象化 (a new abstraction of data management)」として導入された概念が Classical Dataspaces である。原著論文では、データスペースは以下のように定義されている (Franklin et al, 2005) :

- データスペースは、参加者 (participant) と関係 (relationships) で定義される
- データスペースの参加者は個々のデータソース (例: RDB、XML リポジトリ、文書 DB、ウェブサービス、ソフトウェアパッケージ) である。また、参加者は構造化、半構造化、非構造化データのいずれでも問題ない。
- データスペースは、2 以上の参加者のいかなる関係性もモデリングできる。
- データスペースは、ネスト構造にも、重複構造でも存在し得る。
- 異なる性質のデータスペースは、アクセスルールを共有しなければならない。
- データスペースの境界が流動的なこともあるが、おおよそのケースでは明確である。

これらは、明確に中央集権型の Push and Ingest とは異なるアプローチである (図 10)。

Classical Dataspaces は、データメッシュのようにデータソースそのものの提供方法についてのモデルは提供しないが、むしろそれら個々のソースに対し、多層的・重層的に存在する「空間 (spaces)」としてデータマネジメントの概念を定立したことが、最も特徴的である。

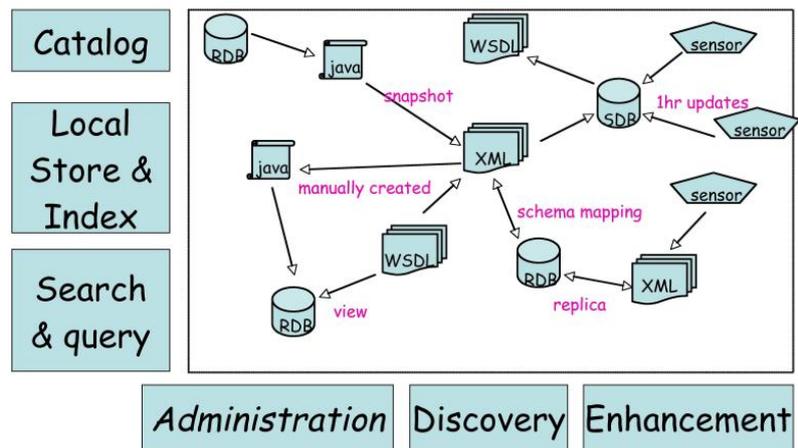


図 10 An example dataspaces and the components of a dataspaces system.  
(Franklin et al. 2005: Figure2.)

なお、データスペースという用語は、欧州の取組でも用いられている。欧州では、2016年頃から Fraunhofer 研究所より Industrial Data Space という名称で Whitepaper (Otto et al., 2016) が公

開され、以降、それを改称した International Data Spaces という技術と制度を織り交ぜた欧州独特のアーキテクチャパラダイムが発展している。

### Classical Dataspaces とデータメッシュから何を学んだか？

Open Dataspaces は、Classical Dataspaces の原著論文である Franklin et al. (2005) に立脚し、HCoD に対しての中央集権的なマネジメントから脱却し、組織と国境を横断したデータマネジメントを行うための分散データマネジメントアーキテクチャ (Distributed Data Management Architecture) として設計した。Open Dataspaces は、HCoD を扱うアプリケーション開発者の煩雑なデータマネジメントコストを低減するとともに、データ提供者・データ利用者それぞれの自律性確保とデータから得られる経済価値の還元を行うための新たなパラダイムである。これは、2019年のスイス・ジュネーブで開催された世界経済フォーラム年次総会 (ダボス会議) で

提唱され、2019年6月のG20大阪サミットにおいて各国首脳からの支持を得て首脳宣言に盛り込まれた「Data Free Flow with Trust (DFFT)」の理念 (Digital Agency, n.d.) を技術的に具体化するものといえよう。

大前提として、Open Dataspaces はデータメッシュのアーキテクチャパラダイムと4つの基本原則を承継している (図 11)。つまり、伝統的な Push and Ingest 型ではなく、Serving and Pull 型へのパラダイムシフトを受け入れ、「ドメイン駆動設計 (DDD: Domain-Driven Design)」を基調としたデータマネジメント方式を採用している。

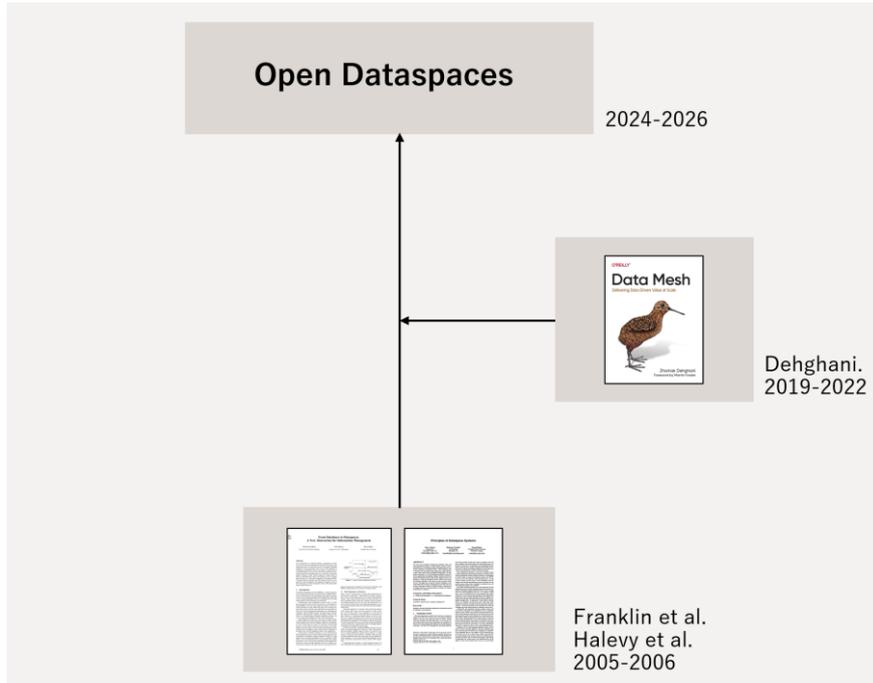


図 11 Open Dataspaces の理論的な源流

ここで、データメッシュとの最大の差分は、Open Dataspaces は、部門間 (Inter-Department) からさらに組織横断 (Inter-

Organization) のデータマネジメントを念頭に置いたパラダイムということである (図 12)。

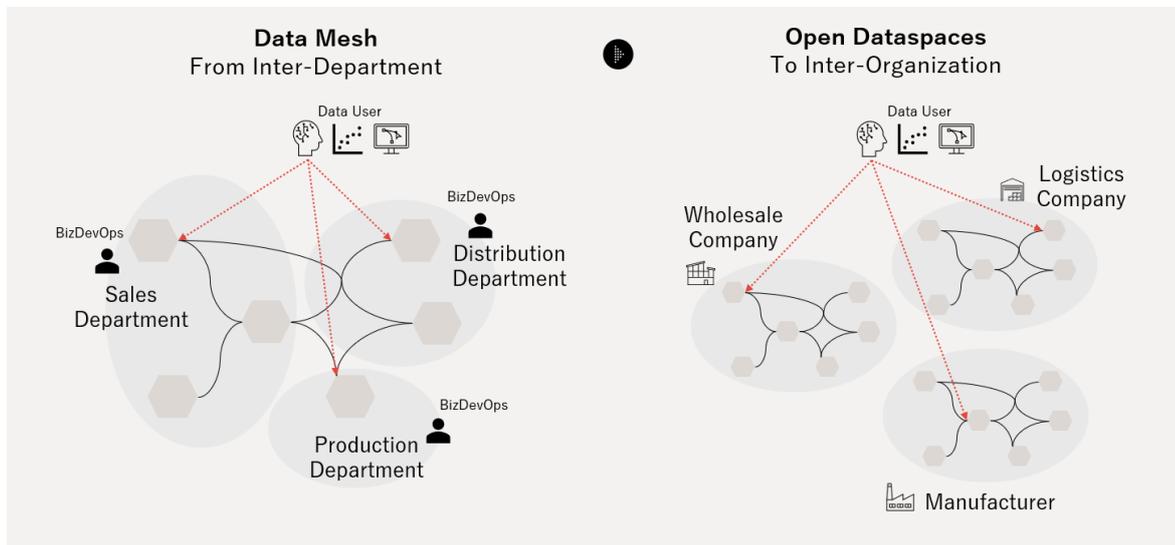


図 12 データメッシュから Open Dataspaces への変遷

ドメインの単位となる部門から、組織を横断することを前提にすることで、なぜパラダイムシフトが生じるのか？

前章で触れたように、データメッシュは商用浸透が進む中で、すでにいくつかの運用上の課題に直面している。例えば、ドメインごとに Data

Product が提供されることによるスキーマやセマンティックの不整合 (Inconsistency) や断片化 (Fragmentation)、アクセス制御上のセキュリティなどの Governance Complexity の問題である。これらを大きくまとめると、「(1) **Where to get**」、「(2) **What to mean**」、「(3) **Who and How to use**」の3つの問題に集約できる:

### 1. **Where to get** の問題:

- **そもそも、どこにデータがあるのか?**  
(例: OEM 契約をした会社の製造部門の実績データはどこで取得できるのか?)
- **そのデータは、このデータと同一のものを指し示しているのか?** (例: Manufacturer の「6AX-10K Industrial Robot」のデータと、Wholesale が命名した「Robot 10kg Standard Model」のデータは同じものか?)

### 2. **What to mean** の問題:

- **そのデータは何を意味するか?** (例: 政府広報チームの Alice は、広報部門に所属しているのか?)
- **そのデータの意味は、このデータの意味と整合しているのか?** (例: このワールドの「高度」は標高を表すのか、それとも対地高を表すのか?)

### 3. **Who and How to use** の問題:

- **データにアクセスしようとしているのは誰か?** (例: この Agentic AI (Client App) は、どの会社が運用しているのか? その運用会社は、存在していて、信用に足るのか?)
- **誰がそのデータにアクセスできるのか?**  
(例: 取引先の航空会社が委託した会社の人事部門は、このデータを閲覧できる

のか?)

- **そのデータはどう使わなければならないか?** (例: このストリーミングデータは、単位時間あたりどの程度の価格か? BI 目的以外の二次利用はできるのか?)

これは、スケーリングに伴う Governance Complexity の問題である。もちろん、データメッシュがこれらを想定していないわけではないが、ドメインを横断し、ドメインの依拠する組織を横断し、さらにそれが国境を横断するとなれば、ドメイン間で調整しておけばよかったガバナンスコストは爆発的に増加する。このスケーリングによる成長痛に対処するためには、最初からアーキテクチャパラダイムに解決策を組み込んでおく必要がある。さらに、現在のテクノロジー環境では、原著論文の 2005 年当時、また、データメッシュが提唱された 2019 年当時にはまだここまでの影響力を持たなかった (Agentic) AI についても、その破壊的な存在を前提とする必要があろう。

注意書きまでに、ここまでの設計思想を簡素にまとめると、以下のとおりである:

- Classical Dataspaces は DBMS をはじめ、データマネジメント基盤である DWH、DL、DLH とは相補的であり、対立する概念ではない。
- Open Dataspaces は、Classical Dataspaces の理念とデータメッシュのパラダイムを引き継ぎ、組織や国境を横断したデータの共有を現実的に駆動していくための補完的なアーキテクチャパラダイム及び技術仕様である。

### 3. Open Dataspaces の特筆性と設計指針、技術原則

#### なぜ、Open Dataspaces か？

Open Dataspaces は、別の言い方をすると、部門や組織、国境を横断した HCoD を扱う際の Governance Complexity の問題を補完するアーキテクチャを提案し、相互運用性を実現するため「秩序と緩やかな規律 (Order and Benign discipline)」をオープンスタンドとして与えるものである。

Open Dataspaces が提示するパラダイムの最大の特筆すべき点は、組織の横断と Agentic AI の存在を前提として、大きな 3 つの柱を持つ分散型

アーキテクチャ (図 13) により、透過的な Single Source of Truth (SSOT) と、データ提供者に相当するドメインオーナーに対する価値還元メカニズムを実現することである：

1. Where to get: 「Data Addressability and Discoverability (DAD)」
2. What to mean: 「Ontology and Semantic Interoperability (OSI)」
3. Who and how to use: 「Identity and Usage Control (IUC)」

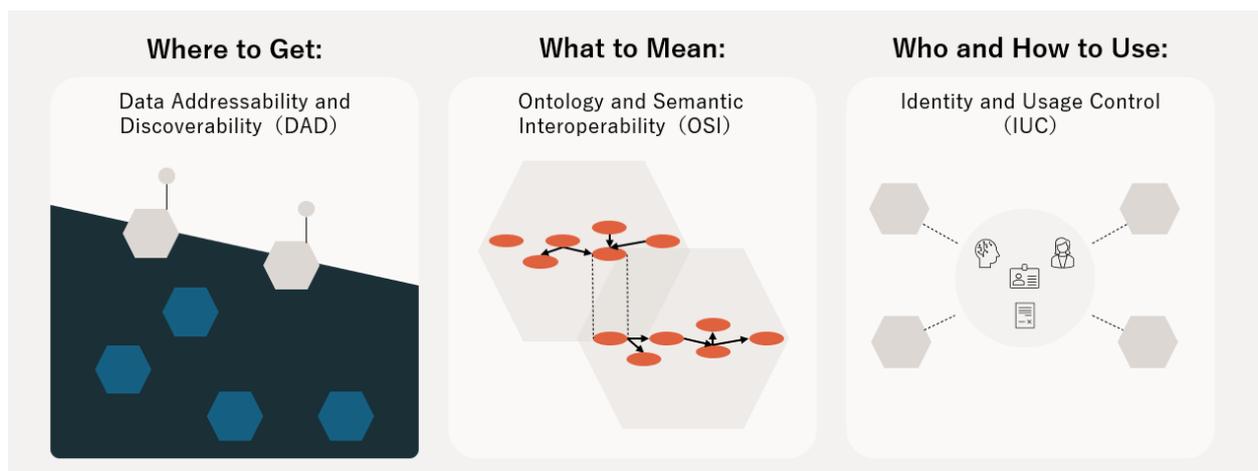


図 13 Open Dataspaces の分散型アーキテクチャ 3 つの柱

DAD、OSI、IUC はそれぞれ、Open Dataspaces の相互運用性を確保するための技術的な規律とデータ提供者の収益機会を確保するための権利的な規律を組み合わせたものである。

ここで重要なのは、Open Dataspaces のアーキテクチャにおいて、**緩やかな規律の実現手段はプロトコルとして提供され、市場の特性や成熟度に合わせて「段階的な導入」を想定すること**であ

る。つまり、Open Dataspaces はフルスイートでのプロトコル適合コストを市場に押し付けるのではなく、**Make Money, Save Money** を基調とする市場の要求に応じて、**オプトイン**できることを許容した、「**必要最小限 (Minimal Yet Viable)**」な設計を重視している。これを実現するため、プロトコルは疎結合に構成され、後方互換性を設計段階でビルトインしている。この「緩やかに規律する」発想は、Classical Dataspaces

の“incremental payoff for incremental investment, and not exist only as monolithic solutions” (Franklin et al., 2005)に依拠したものである。具体的な後方互換についての技術的な整

## Open Dataspaces の設計指針

Open Dataspaces は、分散型アーキテクチャとして掲げる3つの柱を実現するため「(1) ベンダーロックインの回避」、「(2) 制度的ロックインの回避」、「(3) プロダクトライクでサービス志向の設計」を根幹となる設計指針とする：

1. **ベンダーロックインの回避:** マルチクラウド、クラウドレスでの動作を前提とし、特定企業のサービス・商品に依存しないベンダーフリーな設計を採用する。
2. **制度的ロックインの回避:** 特定法域の制度的・規制的要件を技術仕様から明示的に分離し、様々な制度・規制下でのローカライゼーションが可能となるように設計を行う。  
Open Dataspaces はグローバルで適応可能なアーキテクチャパラダイムと技術仕様を提供する。
3. **プロダクトライクでサービス志向の設計:** 解くべき課題、そして、機能要求は常にマーケットにある。本質的なニーズは、法制度や規制、特定のベンダーのプロダクトやシステムにはない。ただし、マーケットを構成するユーザーが答えを教えてくれるわけではない。設計者は、マーケットが潜在的に求める革新から逆算しながら、アジャイルな検証を通じ

理については、Open Dataspaces のリファレンスアーキテクチャモデル (ODS-RAM) やプロトコル (ODS Protocols) に譲ることとする。

て Product Market Fit (PMF) を目指していかなければならない。硬直的な技術仕様は陳腐化し、マーケットから拒絶される。  
Open Dataspaces は、Make Money, Save Money に資するか、という観点を非常に重視する。

ここで注意すべきなのは、技術コンセプトとしての ODS 及び一般技術用語としての Open Dataspaces の「Open」は、データをインターネットに公開するという意味ではない。この Open は、Dataspaces の設計指針であるベンダーロックインの回避、制度的ロックインからの開放とグローバル水準の相互運用性を意味するに過ぎず、誤解されやすい点には留意されたい。むしろこれは、データ提供者が、データの共有先を自身の意思のもとで選択・制御し、同時に、適切な対価を得るためのメカニズムに資するものである。

以上が Open Dataspaces の設計指針である。ここからは、この設計指針と技術原則に依拠した Open Dataspaces におけるアーキテクチャパラダイムの最小構成単位を定義する。

## 4. アーキテクチャパラダイムの最小単位: Architectural

### Quanta/Quantum

Architectural Quanta (AQ) とは、分散データマネジメントにおけるアーキテクチャの最小の構成単位である。Open Dataspaces では、データメッシュの AQ である Data Product の相互運用性の問題を解決するため、**Data Product と表裏一体の関係性である「Ontology Product」という概念を新たに導入し、Data Product と**

**Ontology Product を併せて構成単位の AQ とする。**これが、データメッシュパラダイムとの一番の差分であり、パラダイムシフトを名乗る所以である。Open Dataspaces において導入されたこの AQ のパラダイムを **Double-Product Quanta Model (DPQM)** と呼称する (図 14)。

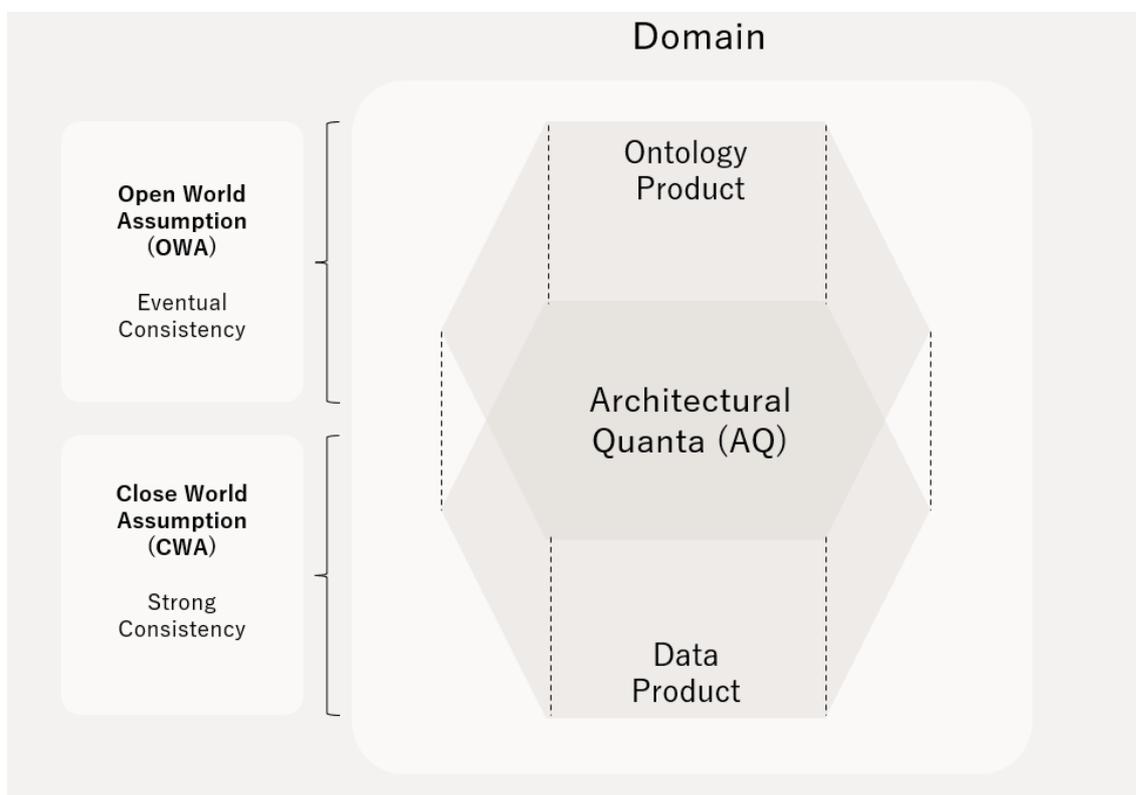


図 14 Double-Product Quanta Model (DPQM)

## Open Dataspaces の前提と DPQM の強み

組織や国境を横断した分散データマネジメントのアプローチとして、Serving and Pull 型のパラダイムを採用し、DDD を志向する Open Dataspaces では、以下の 3 点を「前提 (Assumption)」として置いている:

1. データ提供者は、データを自己宣言的に提供する
2. データ提供者は、不完全又は発展途上の記述を提供するが、その不完全性をもって排除すべきではない
3. データ利用者は、信頼性や完全性、一貫性に関する明示的な保証を求める

この分散データマネジメントにおけるデータ提供者と利用者の期待値に存在するトレードオフを解決するためのパラダイムが、まさに DPQM である。DPQM の真の強さは均質化された完全性を強制することではなく、システム全体の開放性を維持しながら、選択的な厳格性を実現できる点である。

Open Dataspaces は、「(真と) 判断できないことは、偽ではないとみなす」という「開世界仮説 (OWA: Open World Assumption)」を前提としている。これは、データ提供者としてのドメインオーナーが自己宣言的に公開する Data Product の多様性を積極的に包摂するうえで最も親和性の高い発想である。しかし、OWA では、記述されていないことに対する完全性の保証はできない。Push and Ingest 型パラダイムである既存の中央集権型のデータマネジメントアプローチは、「閉世界仮説 (CWA: Closed World Assumption)」を前提としている。これは、「(真と) 判断できないことは、偽であるとみなす」という前提であ

り、この厳格性がデータ利用者のデータに対する信頼性や完全性、一貫性に関する期待値を充足してきた。

従って、Open Dataspaces では、Ontology Product で OWA を前提としながら、Data Product で CWA を選択的に導入するという二層構造を取っている。この設計が、データ提供者と利用者の期待値に存在するトレードオフを埋めるのである (図 14)。

例えば、航空産業における航路情報のような静的なデータ検索は、クエリを実行した時点では整合性が保たれていなくても、最終的に一貫性が保たれていればよいとする「結果整合性 (Eventual Consistency)」が許容され、OWA 的なマネジメントの親和性が高い。一方、運航管理を司るシステムなどでは、検索だけでは不十分である。衝突事故を防ぐためには、動的に変動する飛行計画の整合性を担保し、最新のステータスを担保する同期 (synchronization) が必要になる。これは、「強い一貫性 (Strong Consistency)」と呼ばれ、CWA 的なマネジメントが求められる領域である。

これは、データマネジメント方式に関する根本的な設計思想の転換を意味する。Open Dataspaces が強調しているのは、**集合論の束縛からの解放**にある。RDBMS の設計者である Edgar Codd が定義した関係モデルでは、「現実世界のすべてのデータをあらかじめ定義した集合として扱える」という CWA を前提としている。しかし現実のデータ世界は無限に広がっており、有限の集合として事前に定義することはできない。組織や国境を横断した分散データマネジメントでは、誰がどのよ

うなデータを持っているかを予め定義することが、そもそも不可能である。Open Dataspaces は先にインスタンス（データ）があり、そこに意味を後から付与する「スキーマフレキシブル（Schema Flexible）」という技術指針で解決す

る。これは OWA との整合性によって支えられる設計である。DPQM の強靭さは、本書を読み進めていただくことでさらに理解いただけるだろう。

## 基本用語と関係性

併せて、ここで基本的な用語についても整理しよう。「(An) Open Dataspace」は 2 以上の AQ によって構成される Architectural Quantum (AQM) である。例えば、卸売業企業の物品ドメイン AQ と物流企業の倉庫ドメイン AQ により、Wholesale Distribution Dataspace (AQM) が構成される。同様に、卸売業企業の物品ドメイン AQ と製造企業の生産ドメイン AQ により、Manufacturing Logistics Dataspace (AQM) が構成される。

Open Dataspace は動的・多層的に構成され、例えば、製造企業が卸売業企業と需給調整の最適化の Open Dataspace を構成したい場合は、それぞれの企業の AQ を要素とする形になる。このような多層的・重層的な集合関係を含む Open Dataspace の総体を「Open Dataspaces」または「The Open Dataspace」で呼称する（図 15）。（以降は、便宜的に「Open Dataspaces」と呼称する。）

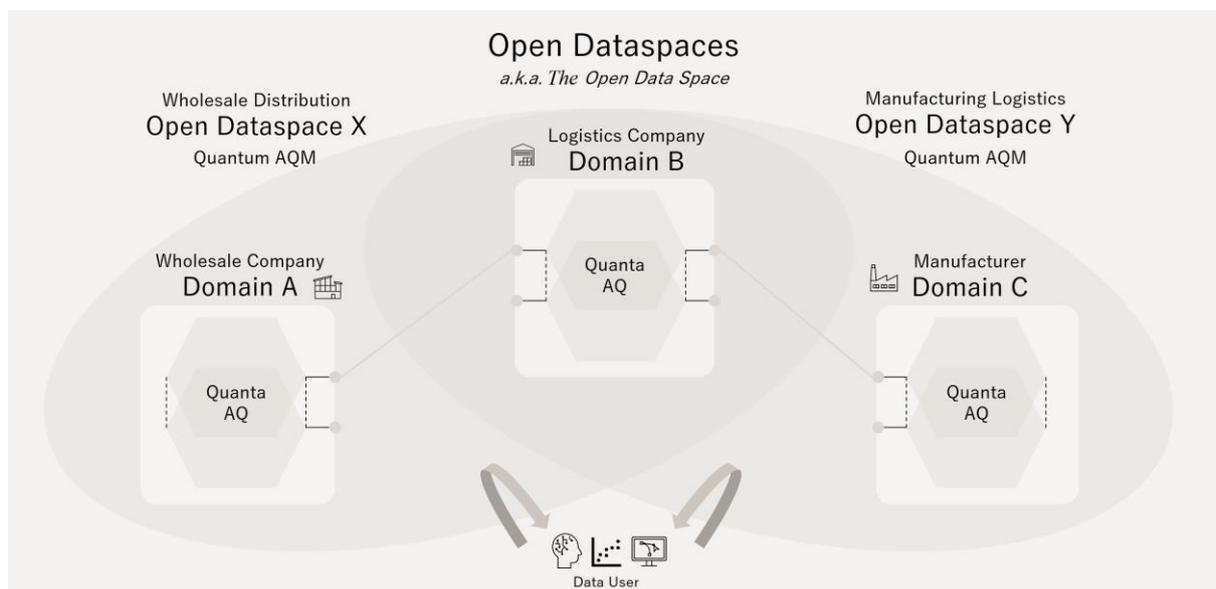


図 15 Open Dataspaces の構成単位 (AQM)

なお、AQ を構成する Ontology Product と Data Product それぞれの間に挟まれた API ゲートウ

エイやマルチモーダルなデータソースの抽象化をはじめとしたトランザクションマネジメント（構

造化、半・非構造化／同期、非同期のデータ通信など)、IUCの一環として実施される認証、認可を司る機能を疎結合に構築するため、Open DataspacesはAQを論理的な4つのレイヤーに分解して機能整理を行っている（

図 16)。ここでは対応関係だけについて触れるため、詳しくはODS-RAMを参照されたい。なぜこのようなレイヤー構造になったのかは、本書を読み進むことで理解いただけると思う。

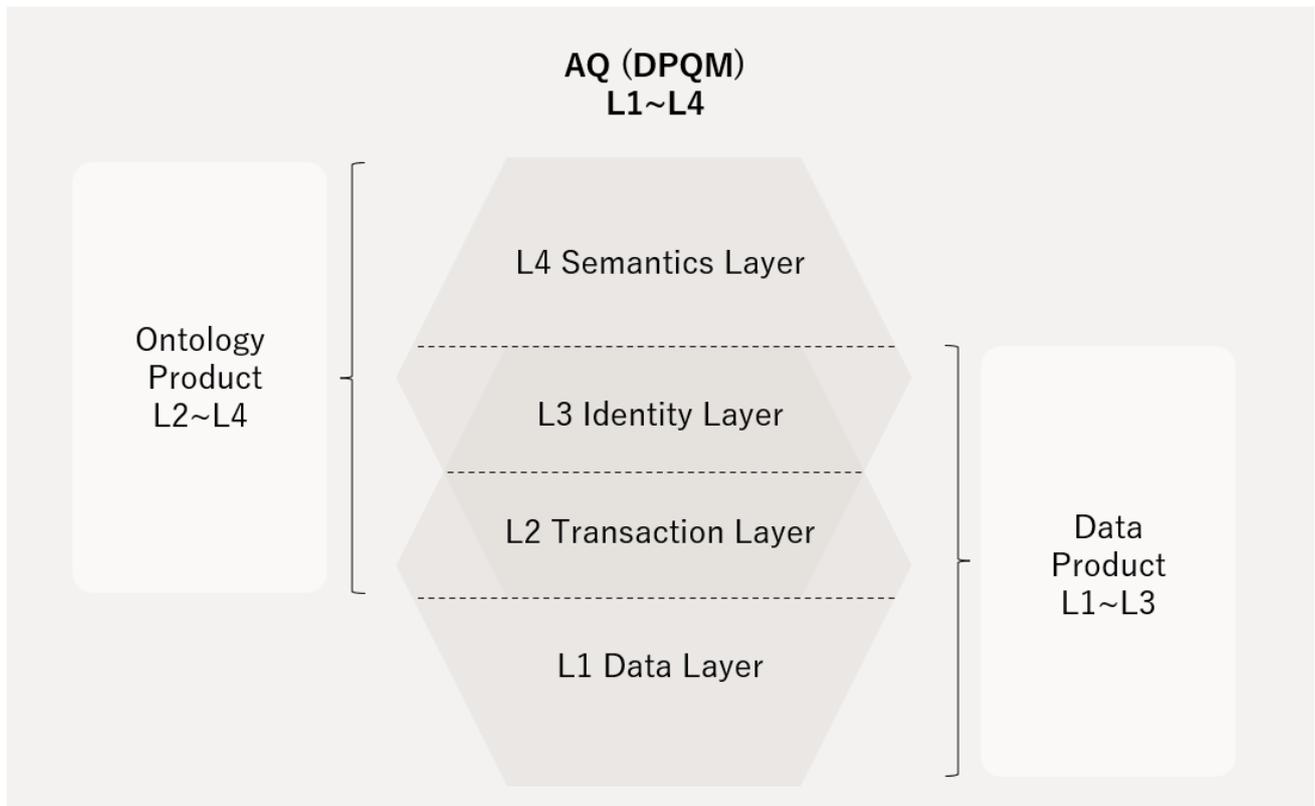


図 16 DPQM と Functional Layer の対応関係

ここから、最大の問題関心である OSI と DAD、IUC について、それぞれ解像度を上げて見ていこう。

## 5. 柱 1: Ontology and Semantic Interoperability (OSI)

### Ontology as a Product—データモデルから情報モデルを分離する

組織や国境を横断した分散データマネジメントで、Data Product が直面する最も原始的で、スケーラビリティの枷になる問題は「意味 (semantics)」である。

この問題を解決するため、Open Dataspaces は、「データモデル (Data Model)」から「情報モデル (Information Model)」を明確に分離する DPQM を採用している。これは、それぞれ Data Product と Ontology Product の分離に対応する。データマネジメントの世界で両者はしばしば混同されるが、Open Dataspaces においては異なる役割と責務を持つ：

- **データモデル:** 業務データ、センサー値、取引記録、ログなどの観測された事実としての要素と、一般に DBMS のスキーマ、テーブルに相当する要素を格納するための構造と関係性を表現するもの。そこでは、効率的な保存、取得、更新が重視され、意味や解釈は最小限に抑えられる。一度記録された事実は、原則として過去に遡って変更されない。
- **情報モデル:** 事実をどのように解釈し、関連付け、制約し、操作するかの意味を表現するもの。(参考：Lee. (1999) 等) 分類、同一性、前提条件、禁止事項といった意味論的な構造は、情報モデル (及

びその抽象化された概念である Ontology) の責務であり、運用の変化に応じて更新されうる。

従来の情報システムでは、これら二つの役割がデータモデル自体に混在している。その結果、意味の変更がスキーマの変更として扱われ、過去データ、アプリケーション、分析ロジックに連鎖的な影響を与えることが多かった (図 17)。ではなぜ、データモデルと情報モデルを分離することが有効なのか？ ある工場の事例で考えてみよう：

- 製造業においては、設備、部品、工程といったデータ構造が長期間にわたり利用される。一方で、それらに付与される「意味」は、品質基準の見直しや安全規制、運用改善によって継続的に変化する。
- 例えば、ある製造設備に対して、当初は「稼働/停止」という状態のみを管理していたとする。その後、品質事故を契機に「異常振動」「過負荷」「人手介入の有無」といった観点での評価が求められるようになった場合、従来の設計ではテーブル定義や API 仕様を変更し、過去データの移行やアプリケーション修正が必要になる。

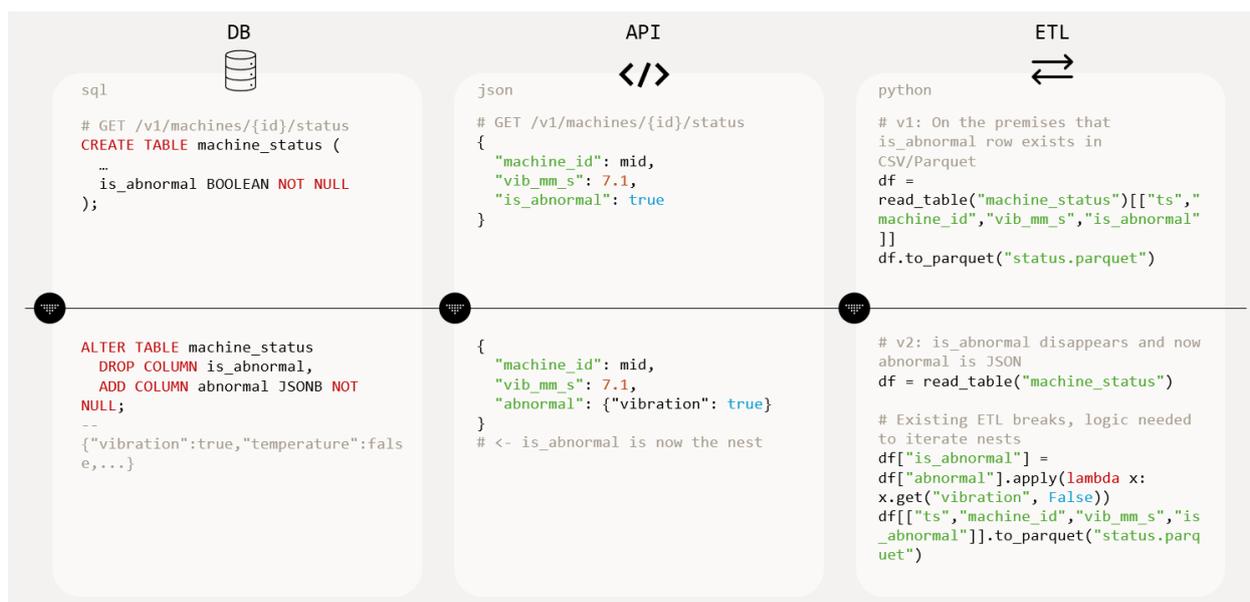


図 17 「意味の変更」が構造の破壊を起こす典型的なケース

これは、データモデルが、観測された事実を記録するため、現実の変化に強く引きずられることを意味する。情報モデルは、その事実をどのように解釈し、関連付け、制約するかを扱うため、時間を遡って再解釈されることがある。両者を同一のスキーマに押し込めると、頻繁に変化するデータの都合で意味が破壊され、結果として莫大なデータベースの再設計コストが発生するのである。そう、**組織における多くのシステム改修は、実際にはデータが変わったのではなく、データの意味が変わったことに起因している。**

一方、情報モデルをデータモデルから分離した設計では、設備データそのものは変更せず、Ontology 側に「異常状態」や「安全リスク」といった新たな概念を追加することで対応できる。既存の設備インスタンスは、情報モデルの更新により条件付きで新しいクラスや属性に分類されるが、元のデータはそのまま保持される。その結果、過去に取得された稼働ログや保全記録を破壊することなく、新しい評価軸を全設備に一貫して適用できる。アプリケーションや分析処理は情報モデルを参照して意味を解釈するため、表示項目

や判定ロジックは拡張されるが、既存のインターフェースを壊す必要はない。つまり、Open Dataspaces が情報モデルを独立した知識単位 (Knowledge Unit) として扱うのは、**意味を保存するためではなく、再解釈可能な状態で維持するため**である。情報モデルを分離することで、実運用で避けられない意味の進化を、データ構造の破壊ではなく、再解釈として扱うことが可能となる。これが、図 14 で示した DPQM の強みである。

この設計思想を反映するため、Open Dataspaces では、Resource Description Framework (RDF) 及び RDF\*により、観測事実・用語・出所を知識単位として保持する。次に RDF Schema (RDFS) により語彙と構造を与え、Web Ontology Language (OWL) により妥当性や排他性といった制約を定義する (表 2)。ここでは、Semantic Web 技術の説明を詳述することはないが、RDF は、Subject-Predicate-Object からなるグラフ構造として定義されるモデルであり、意味が完全に確定していない段階でも、関係性を保持したまま分散的に拡張できるように設計され

た、知識単位である。

表 2 Data Product と Ontology Product のスコープの差異

AQ	Data Product	Ontology Product		
レベル	Data Model	Information Model (狭義)	Semantics	Ontology
表現	任意の Multi-Modal Raw Data	RDF (and RDF*)	RDF Schema	OWL
目的	Reflecting Reality	Knowledge-Unit	Vocabulary and Structure	Validation and Reasoning

## Semantic and Ontological Interoperability—推測から知識へ

さらに、意味の問題をもう一段深掘りしよう。組織や国境を横断した HCoD を扱う Open Dataspaces において、**特定組織の特定ドメインが定義した自己定義 (Self-description) から生じる Semantic Gap** は非常に大きな相互運用性の課題である。

例えば、航空分野における「高度」は、平均海面からの海拔高度である「標高 (MSL: Mean Sea Level)」と直下の地表からの「対地高度 (AGL: Above Ground Level)」で違う意味を持っている。ある航空運航データにおいて、「高度」を標高として定義した値を前提に組織横断で統合を行った結果、別組織が対地高度を同一の語で運用していたことに気づかず、データ利用者は両者を同一概念として誤認してしまう。**表面的には統合が進むが、実際には誤った同一性が蓄積され、後から検証不能な負債が形成される。そしてこの負債は運航管理上のインシデントにつながる。**

Open Dataspaces が採用する DPQM はここでも強みを発揮する。Data Product から Ontology Product を明示的に分離することで、意味の一貫性 (= 矛盾検出) についてはオントロジーが明示

的に制約し、保証することを設計で実現している。実装レベルでは、RDFS は基礎語彙や関係性を提供し、OWL は推論器に対して意味推論や制約、意味的矛盾に対するエラーを可能にする。**意味の起源はドメインにある。Ontology はデータ提供者としてドメインオーナーが定義し、データ利用者はそれを解釈する、という役割分担が OWA との整合性を担保する定礎となる。**

具体例で見ていこう。情報モデルを分離しない通常の統合では高度の参照基準の違いはデータ構造の一部として扱われる。そのため、意味的な齟齬はアプリケーションロジック、つまり実装品質に依存し、整合性保証が不足した場合、バグとして表面化する。(データベース制約では、AGL と MSL は排他的という意味は表現できない。) しかし、情報モデルをオントロジーとして分離すると、「高度は必ずひとつの基準に属する」「AGL と MSL は異なる概念」といった意味制約を論理的に宣言できる。これにより、誤統合はバグではなく、論理的不整合としてエラーが返却される (図 18)。これは、データ設計の違いではなく、意味の所在をコードから論理体系へ移すという設計思想の差異である。

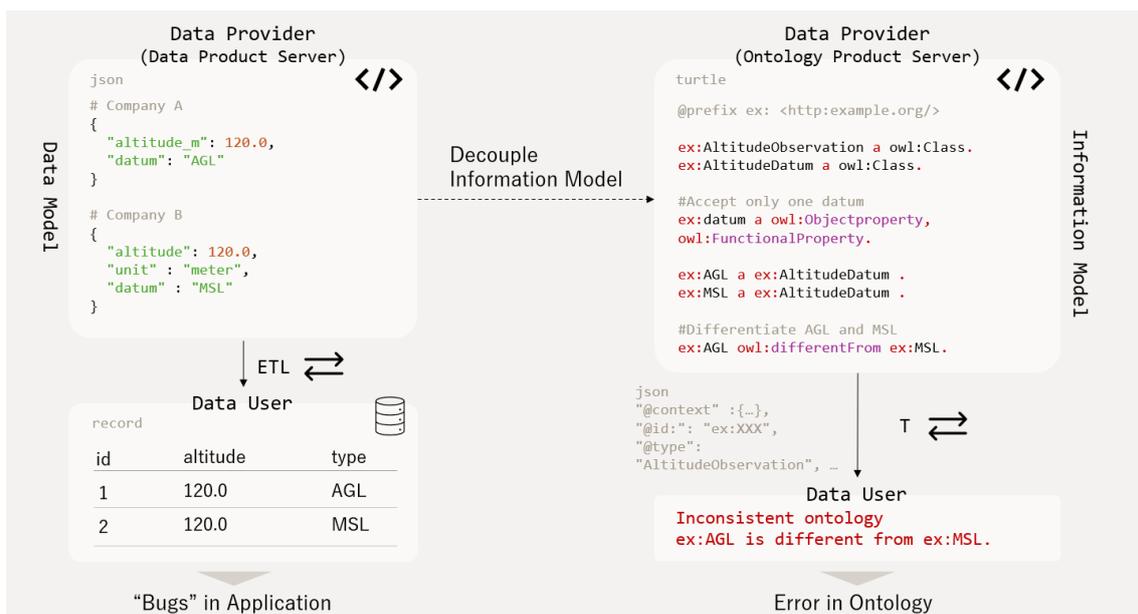


図 18 情報モデルの分離と Ontology の導入による Semantic Gap の解決イメージ

ただし、Ontology が Semantic Gap を解決したとしても、Ontology そのものはデータ提供者が自己宣言的に定義し、Ontology Product として提供するものであり、Ontology 同士のギャップの問題は引き続き存在する。これを、**Ontological Gap の問題**と呼称する。

Ontological Gap は、組織や国境の横断を前提とする Open Dataspace にとっては大きな問題である。特に、Vocabulary と Ontology の分野間の整合性担保、単位変換や同一性判断などのデータ利用者負担の過大化は、スケーラビリティ上のボトルネックとなり得る。データ利用者とデータ提供者の双方の役割を持つ領域などでは、Ontological Gap によるコストは重複投資回避という判断の中で共有化し、上位概念としての標準規格などが一定進められると想定される。

しかし、データ提供者の多様性の高い領域では、そのような協調的な行為は非常に難しい。このような Ontological Gap を乗り越えるための対応付

け (crosswalk) を動的に、かつスケーラブルに実施するために、Open Dataspace では LLM による Ontological Gap の改善を取り込むことにした。歴史的に、Ontological Gap を埋めるためのマッピング作業は人手による労働集約的な解決手段が採用されており、スケーラビリティに課題があった。しかし、その作業を LLM で補完することにより、スケーラビリティを保ちながら、Ontological Gap を埋めることが可能になる。

なぜ、この設計が成立するのか？それは、LLM が異なる Ontology 間に存在しうる対応関係の仮説 (マッピング) を提示する一方で、その仮説はそれぞれの Ontology の構造によって検証・制約されるからである。この相互作用により、意味が確定していない段階でもデータ共有を止めことなく、運用の中で Ontological Gap を再帰的に縮めていくことが可能となる。Open Dataspace では、これを「**Dynamic Ontology**」と呼ぶことにした (図 19)。これは Hypothetical queries などの高度な Query においても有用である。

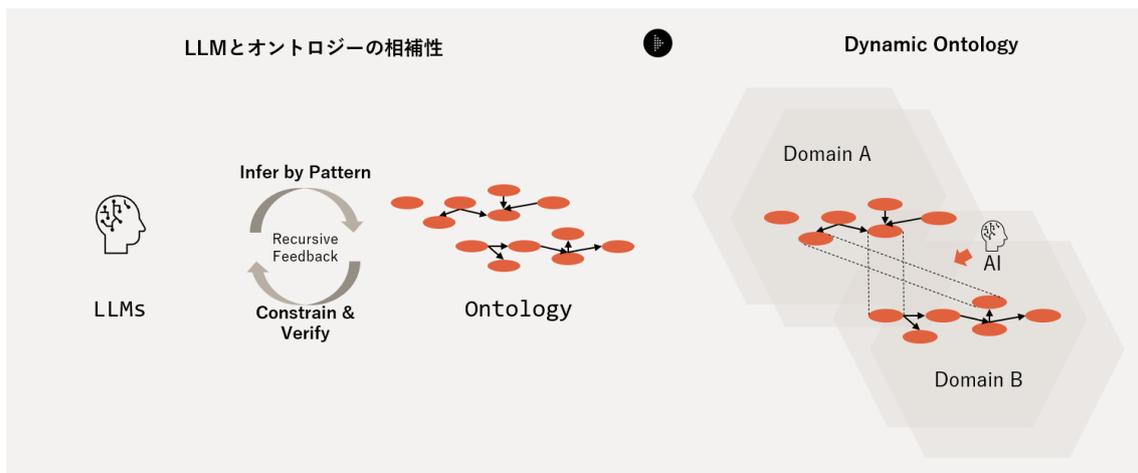


図 19 LLM と Ontology による相補性と Dynamic Ontology

以上のように、AI に与えるべきコンテキストはグラフで表現され (**Graph is Context**)、Open Dataspaces は Agentic AI ネイティブ時代のアーキテクチャとして「**推測 (Guess) から知識 (Knowledge)**」へのパラダイムシフトを促すものである。なお、このような考え方は、第 7 章の IUC でも踏襲されている。

また、Open Dataspaces における Ontology と Semantics は、あらかじめ完成した設計物ではない。それは、**オペレーションの中で更新され続ける対象物**である。Ontology Product は、無頓着に蓄積されたデータは価値が低く GIGO だという問題意識を設計思想に直接反映するものである。データ提供者としてのドメインオーナーが自己宣言的に提供する Ontology そのものが持つ品質の問題は、市場原理によって淘汰されるだろう。

これは、リアルデータを AI-Ready にするという営みと共通する課題であり、Dynamic Ontology のサイクルに暗黙知を形式知化するための

Human-in-the-loop をドメインチームの活動に組み込んでおき、徐々にこの補助輪を外していくことが、スケーラビリティ実現の上で重要である。ただし、Semantic や Ontology を膨大な作業と捉える必要はない。なぜなら、["A Little Semantics Goes a Long Way"](#) だからである。

ただし、OWL は論理推論器であるため意味的矛盾を検出できるが、未宣言の情報を不整合として扱うことはできない。これは OWL 自体が OWA を念頭に置いているからである。不足検出は、検証器の問題であり、アプリケーション側が完全性を宣言するか、SHACL (Shapes Constraint Language) などの閉世界的検証の導入が必要であるが、CWA の強制は、自己宣言とベストエフォートでの緩やかな統合をめざす Open Dataspaces の発想とは相いれない。このトレードオフを解決するために、Open Dataspaces では二段階のクエリ概念を導入しており、その設計を次章で示していくことにする。

## 6. 柱 2: Data Addressability & Discoverability (DAD)

### Data Addressability—データの「存在」と「同一性」を保証する

次に、データ提供者に対する潜在的なデータ利用者の観点で見てみよう。分散データマネジメントにおいて、データ利用者が最初に直面する障壁は、「データにどのように辿り着くか」という Addressability である。Addressability は、「存在」と「同一性」で構成される。

まずは「存在」の問題について見ていこう。前章で触れた通り、多くの組織では、「データは存在しているが、使えない」というダークデータが常

態化している。しかし、分散データマネジメントの観点では、これは正確ではない。

**Addressability を欠いたデータは、存在していないのと同義である。**組織を横断してデータが分散管理された環境では、利用者は事前に「どのドメインに、どのような意味やコンテキストを持つデータが存在するか」を知ることができない。もし仮に、ドメインオーナー側で明確にデータが管理されていても、利用者側からは不可視のまま取り残される (図 20)。

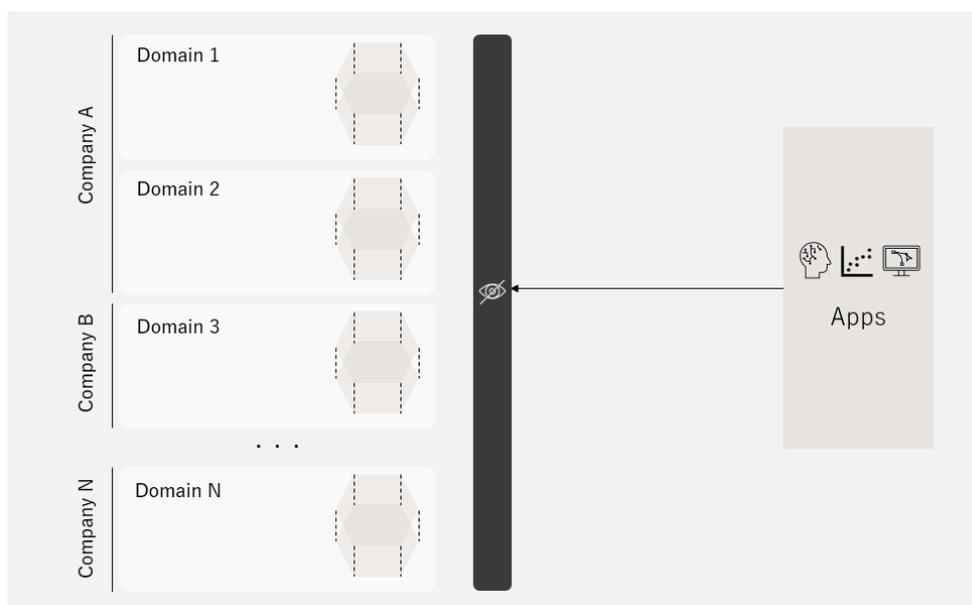


図 20 Addressability の問題

この構造は、Web が直面してきた課題と類似している。Web 上の情報もまた、世界中に分散した主体によって管理されており、URL を知らなければ到達できない。Web はこの問題を、すべてを中央で管理することによってではなく、探索可能性を前提とした設計によって乗り越えた。ここで重要なのは、宛先を完全に列挙することでは

なく、探索を開始できる状態を提供した点にある。Open Dataspaces が Addressability を柱のひとつに位置付ける理由は、この他者から見えないデータを、存在させるという点にある。

DPQM を採用する Open Dataspaces は存在の問題に対して、ドメイン外部に対してそれぞれ

「Ontology Endpoint」と「Data Endpoint」という2つの独立したインターフェースを持つ構造を取る（図 21）。この Web を前提としたエンドポ

イントこそが、AQ が Open Dataspaces に現れるための唯一の存在証明の接点であり、探索を開始するための手がかりである。

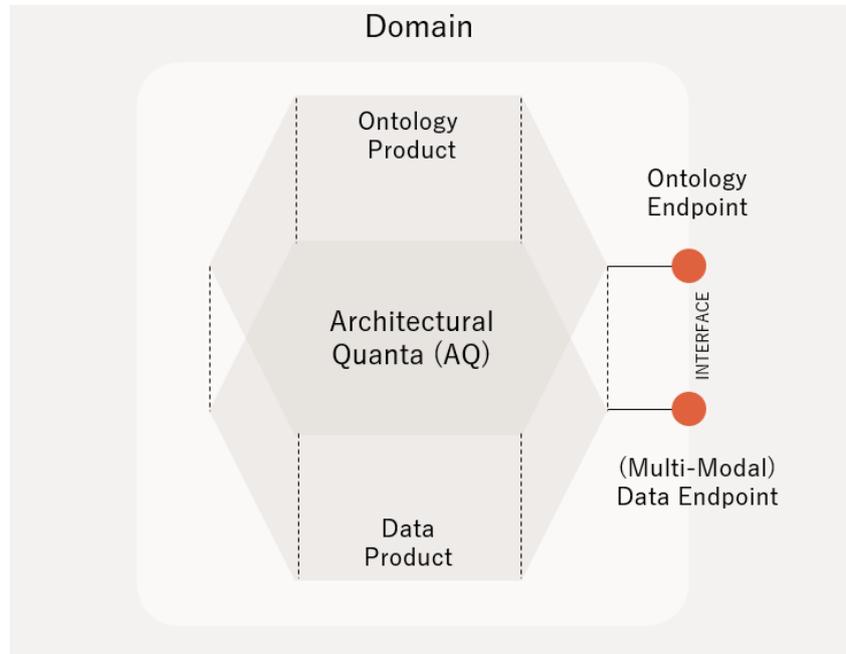


図 21 Ontology Endpoint と Data Endpoint

これで、データ利用者は理論上データの存在を知ることができる。しかしエンドポイントが存在しても、それがドメインごとに異なる識別子（Identifier）で参照されている限り、ドメイン横断、組織横断の Open Dataspaces において同一の実体として認識されることはない。これが、「同一性」の問題である。

例えば、産業用ロボット Manufacturer の事例で考えてみよう。ドメインごとに設計された識別子が異なる場合、組織の中では、以下のような問題が生じる：

- 商品開発部門のドメインで命名した「Prototype Rev.B」（例：設計 BOM）と、生産部門のドメインで定義された「6AX-10K Line3」（例：製造 BOM）は同一のものか？

商品開発部門と生産部門はドメインごとの立場やコンテキストで製品を扱うため、命名と内部識別子がそれぞれで異なるのは当然である。しかし、これらを統合的に扱う場合、実務的には Data Product のマッピングコストが発生する。さらに、企業を横断する AQM 単位で見ると、同一の完成品ロボットが B2B 流通で顧客に届くことを考えた時、以下のような問題がさらに生じる：

- Manufacturer が命名した“6AX-10K Industrial Robot”と、Wholesale が命名した“Robot 10kg Standard Model”、Logistics が命名した“Pallet#12345 /SKU-ROB-10KG-JP”が同一機体だが、製品に紐づいている契約によって命名と内部識別子が異なるため、返品、リコー

ル、保守契約それぞれのコンテキストで参照することに大きなコストが生じる。

- 海外に工場を新設するとき、識別子の多言語対応が発生する。
- 同一型番なのに、商品が導入された現場ごとに異なる識別子が付番されており、同一設備の稼働履歴、不具合の再発傾向、部品交換の因果関係などを生産部門が把握できない。

伝統的な Push and Ingest 型のパラダイムで解決しようとする、データモデルに紐づけるべき部門や組織が増えるたび、Canonical ID の再設計や、エンティティパイプラインの再構築、マッピングテーブルの作成、多言語対応など、永久に終わることのないデータ統合が要求される。しかし、そもそも AQ が動的に生成・接続される Open Dataspaces の前提において、すべてのケースで正解となる識別子を事前定義すること自体が現実的ではない。

ここで重要なのは、Open Dataspaces が解決し

ようとする同一性の問題は、世界に唯一の客観的な同一性が存在し、それをシステムが保証するという発想には立っていないことである。むしろ、ドメインごとに異なる内部識別子とコンテキストで管理される実体が、ビジネス上の必要性に応じて後付けで「同一の実体について話している」と合意できる手段を提供することが目的である。

このような背景から、Open Dataspaces の DPQM では、Ontology Endpoint を明示的にデータソースの取得エンドポイントである Data Endpoint から分離し、IRI (International Resource Identifier) を Ontology Endpoint に採用した (図 22)。ここで IRI の説明は詳述しないが、要するに、URL の概念をリソースに拡張したものである。IRI で自己定義・公開された Ontology Endpoint はグローバルでユニークな識別子として機能する。これは、セマンティックウェブ研究において生み出された知恵であり、インターネットが依拠する名前解決の概念に基づいている。

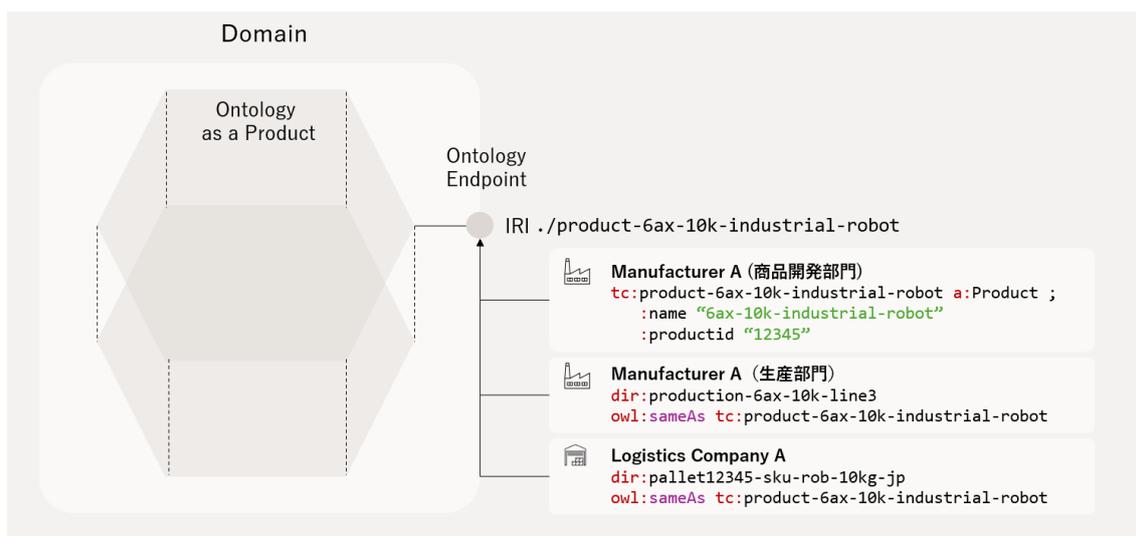


図 22 IRI によるグローバルな存在識別

上述のとおり、IRI は、ドメインごとに異なる内

部識別子を否定するものではない。むしろ、それ

らを温存したまま、**存在についての最小限の合意点**を Open Dataspaces に提供するものである。  
 図 22 に記載のように、Manufacturer A の生産部門や Logistics Company A は内部識別子を後付

けの Ontology 表現で紐づけることができる。これが、拡張可能な IRI 体系の強みである。IRI はドメイン固有の識別子を相互に参照可能な形で紐づける土台を提供するに過ぎないのである。

## Discoverability—データの関係性を提供する緩やかな検索機構

ここまでは、データの所在に対する Addressability の問題にフォーカスを当ててきた。ただし、これは Ontology Endpoint という玄関口をデータ利用者が発見し、そしてそのポイントが指し示す存在の唯一性を外形的に判断できるようになっただけである。この段階では、データ利用者から見て、Ontology で制約された情報モデルの関係性を検索し、クラスタリングする、つまりデータを取得するためのポイントを示す目録を得るための標準的なクエリをどうすればいいかわからないということである。

この問題は、分散型アーキテクチャを採用した時

点で必然的に生じる。単一組織内であれば、データは中央集約され、データモデルの一環でスキーマや命名規則も統制され、そこから独立した包括的なカタログを維持することが可能である。しかし、ドメインごとに自律的な管理を前提とする環境では、唯一の正解となる目録や、全体を網羅する一覧をあらかじめ構築することはできない。

このような問題を解決するために導入するのが、「**ディスカバリー (discovery)**」という概念である。Open Dataspaces では、実体のデータソースに到達するまでに、2段階のクエリプロセスを経ることになる (図 23)。

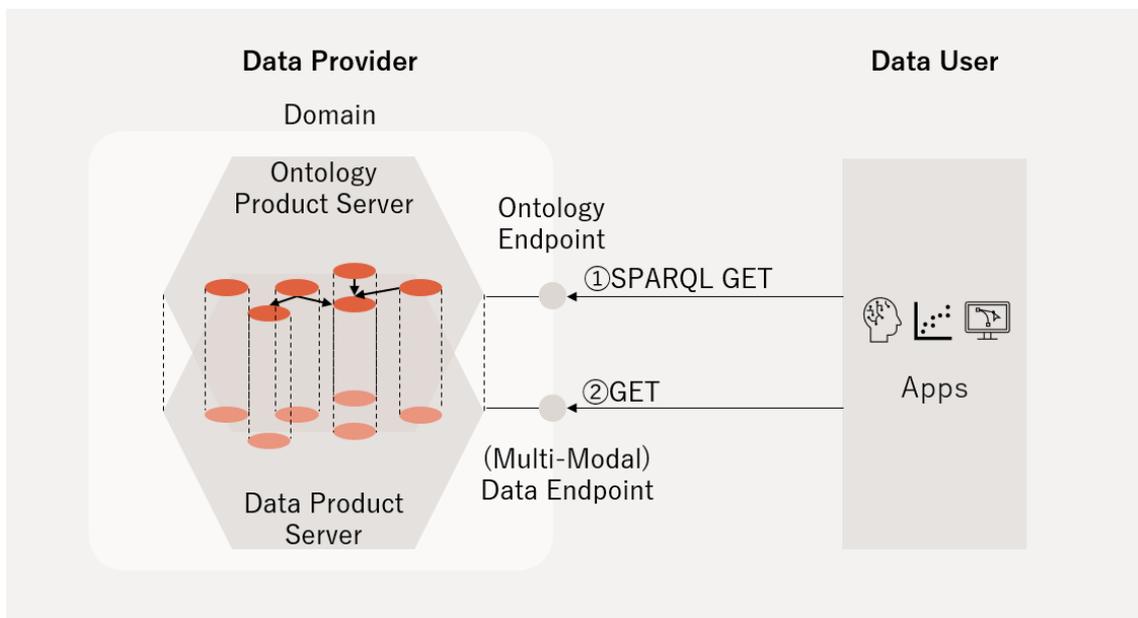


図 23 Discoverability を構成する最小の関係性

1. **Ontology Query (第一段階)** : Everything query は任意のキーワードに対して Best effort result (=データカタログ) を提示する。
2. **Data Query (第二段階)** : Best effort result として提示されたグラフをベースに、それに紐づくデータソースのエンドポイントに対して、アクセスを行う。

Open Dataspaces におけるディスカバリーは、完全な答えを返す検索機構ではない。データ利用

者は、完全性の保たれたデータを知りたい場合、データソース（つまり、Data Product）を直接見に行く必要がある。従って、**Open Dataspaces** でいう「データカタログ (Data Catalog)」は、誰かに硬直的にホストされた静的なリポジトリ（例: **Comprehensive Knowledge Archive Network (CKAN)** を利用したデータポータル、**Clearing House** リポジトリ 等）ではなく、クエリ依存で都度 **Best Effort Result** として生成される動的なビューワー (viewer) に過ぎない (図 24)。

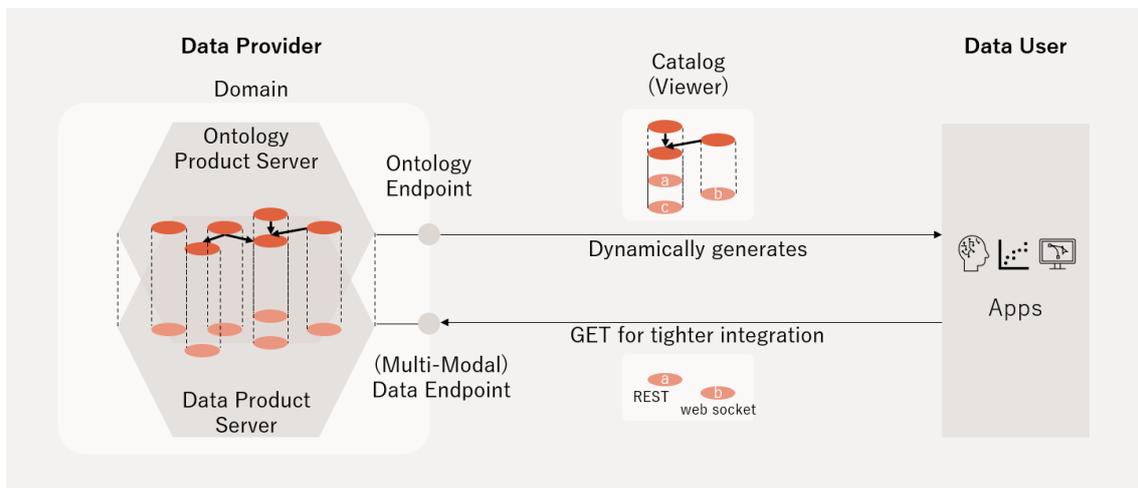


図 24 動的なビューワーとしてのデータカタログ

なお、HCoD を横断した都度の全対全クエリは非効率であり、計算量とランニングコスト最適化のためには、ディスカバリーされた関係性のキャッシュであるローカルストアやインデックス、横断的なクローリングといった仕組みを導入する必要がある。そこで、Open Dataspaces では、Web

の検索エンジンをモデルにした Ontology のインデキシングとクローリングを行う仕組みを導入した。これを Open Dataspaces では「分散カタログ (Distributed Catalog)」と呼称する (図 25)。

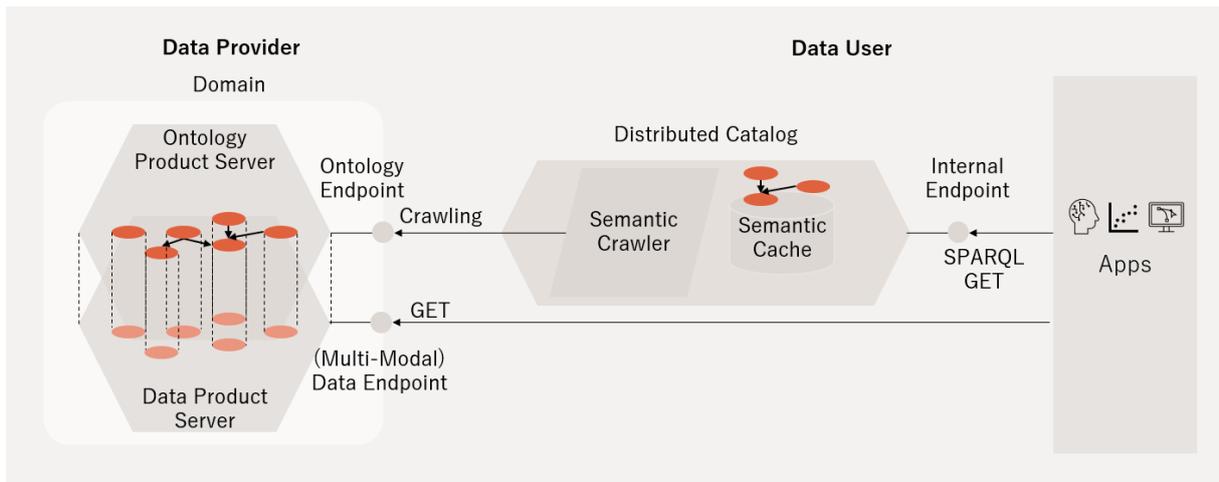


図 25 分散カタログの概要

また、このような分散カタログや、HCoD の検索起点となる最初の Ontology Endpoint を発見する機能を提供するサービスを総称して「**ディスカバリーサービス (Discovery Service)**」と呼称し、このような仲介機能により、データ利用者が Open Dataspaces をより効率的に探索し、より

多くのデータソースに到達できるようになるだろう (図 26)。なお、Ontology 自体が企業秘密に該当する場合も多く存在することから、Data Product 同様、Ontology Product 自体にも IUC の概念が適用される。詳しくは、次章にて後述する。

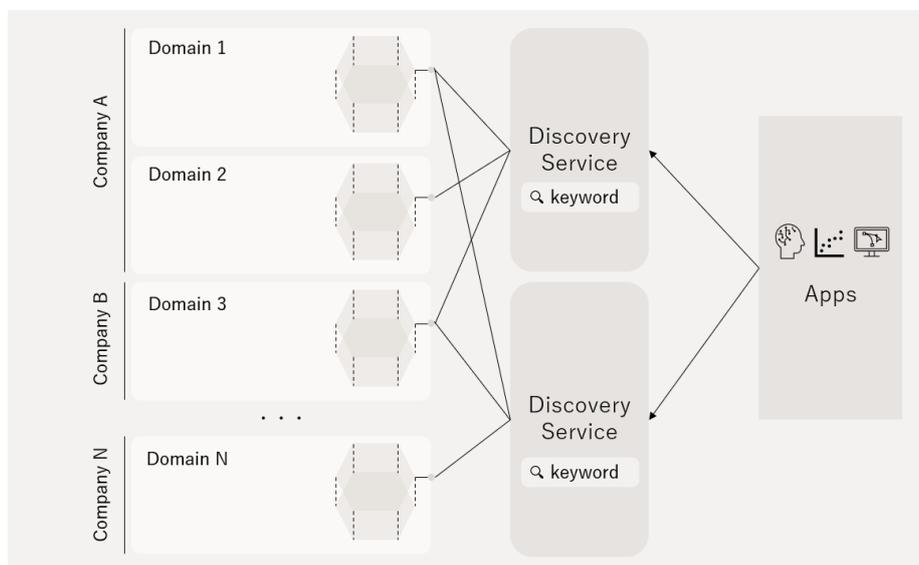


図 26 ディスカバリーサービスの概要

## 補論: Data Trust と Data Trustworthiness について

Open Dataspaces は OWA を前提としているが、データモデル側である Data Product 単位の自己宣言をはじめ、限定された境界における CWA 的な検証を選択肢として導入することは排除しない。これを実現するためにとった手段が、本章で記述した Open Dataspaces における 2 段階クエリである。

そのうえで、部分的な CWA 的な検証を導入するためには Data Trust、すなわちデータプロダクトの完全性 (completeness) についての不確実性に対する保証とリネージュ (lineage) について、第三者が検証可能な技術的インターフェースを用意する必要がある。これをオプションとして提供するの、ODS Protocols を構成する「Data

Trust Assessment Protocol」である。同様に、品質についても、ドメインオーナー自身が Service Level Objective (SLO) として提示した指標を第三者が検証可能なインターフェースのみ「Data Trustworthiness and Quality Assessment Protocol」としてオプションに用意するにとどめている。これは、例えば Great Expectations のような品質アセスメントや、データダウンタイムの低減といった動的な品質管理を含みうるものである。これらをドメイン非依存に統一された基準として規定することは非現実的であり、検証可能性という余白をプロトコルとして残しておくことで、データの監査といった機能を第三者が担うことが考えられる。

## 7. 柱 3: Identity and Usage Control (IUC)

### Identity—Trust by Design

最後に、3本目の柱である IUC についてみていこう。単一組織内の情報システムでは、「信頼 (trust)」はデータ利用者、提供者それぞれの間で暗黙の前提として扱われる。しかし、分散データマネジメントでは、それは成立せず、より大きな問題として顕在化しやすい:

- **組織の境界が固定されているという前提の不成立** (例: 設備メーカーがプロジェクト単位で入れ替わる、保守委託先が年度変更される、合併会社設立に伴い、同一人物が別法人の立場を持つ など)
- **主体の同一性が自明であるという前提の不成立** (例: 同一人物の Alice が、製薬会社の社員、商社の委託先、海外現地法人の一次責任者として振る舞う など)
- **認証できた主体は信頼してよいという前提の不成立** (例: 正規のアイデンティティでログインできるが、NDA が失効している、契約が終了している、対象業務の管轄外になっている など)

このような構造的な問題を扱わないと、本来アクセス不可のログが閲覧できてしまう、契約は有効なのに一時的にアクセス不能になる、「なぜこの人・Agentic AI が見られたのか」を説明できない、といった運用上の問題が発生する。認証と信頼を同一視した設計によって生まれるこの問題は、単なるインシデントではなく、運用と監査の破綻を意味するともいえる。

Open Dataspaces では、こうした組織の流動性に起因する現場の混乱を前提に、**信頼を前提ではなく、設計対象**として扱う (**Trust by Design**)。換言すれば、特定のドメインだけではなく、組織や国境を横断したデータマネジメントでは、信頼は設計によって構築されるべき性質だといえよう。これが、Open Dataspaces が AQ を論理的に分割したレイヤーのひとつである Identity Layer (L3) から、信頼を明示的に分離した横断的パースペクティブとして位置付けている理由である。ここでは対応関係だけについて触れるため、詳しくは ODS-RAM を参照されたい (図 27)。

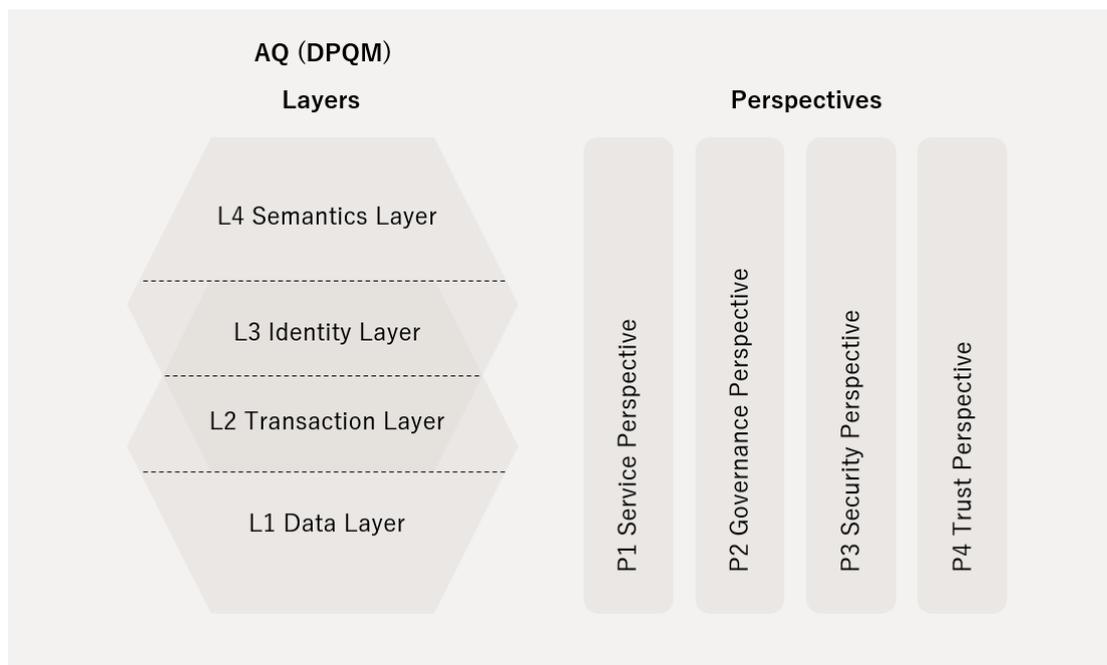


図 27 AQ を論理的に構成する Identity Layer から Trust Perspective の明示的な分離

Open Dataspaces では、Trust by Design という視点で、アイデンティティを3つの要素に分解し、それぞれに異なる責務を与えている:

- **実在性の検証 (Identity Proofing)** : 主体が現実世界に存在することを担保する。つまり、この主体は、現実世界のどの個人・組織に対応するのか? (例: どの法人に雇用・運用され、どの契約に紐づく立場で、それはいつまで有効か など)
- **認証 (Authentication)** : 主体の主張を確認する。つまり、この主体は、主張するアイデンティティの所有者か? (例: アカウント、証明書、API Key など)
- **認可 (Authorization)** : 主体と資源の関係から行為を許可する。つまり、この主体は、この資源に対して、今、この操作をしてよいか? (例: 設備 A の稼働ログは閲覧可能、設備 B の制御 API は不可など)

Trust by Design の視点において最も重要なのは、「**認証された主体**」と「**信頼してよい主体**」を同一視しないことである。

Open Dataspaces では、**その主体が実在することを裏付ける信頼性の根拠を明示的に扱う**。信頼性の根拠には、例えば、現実世界の個人・組織との対応関係を説明でき、時間の経過に伴う失効・更新が可能で、組織間・国境間での相互説明が可能なことが求められる。ここで重要なのは、Open Dataspaces が**単一の信頼性の根拠を強制しない**点にある。これは、国・法域ごとに法人登録や個人識別制度は異なり、業界ごとに異なる契約慣行や責任分界の考え方が存在するという制度的ロックインの回避の設計原則を踏まえた判断である。

Open Dataspaces では、分散データマネジメントのリスクや用途に応じて、必要なトラストレベルを合意し、検証・署名・時刻保証などの手段を組み合わせることで、**用途適合型の信頼構造**を構

築する。これにより、Open Dataspaces は完全な統一ではなく、秩序と緩やかな規律に基づく信頼の相互接続を実現する。これは、Agentic AI 時代の「**非人間アイデンティティ (Non-Human Identity)**」の管理においても重要な考え方である。

また、後述するが、Open Dataspaces における認可の主要方式は、主体と資源の関係をグラフとして扱う。このときアイデンティティは、単なる属性集合ではなく、関係推論の起点として設計されたノードである。Trust by Design の観点では、次が重要となる：

- 主体は、組織やシステムを越えて参照可能であること
- 主体に付随する属性や関係は、時間とともに変化する
- 認可判断は、常に最新の関係状態を前提とすること

## アクセス制御—Graph-to-Graph Control

認証の問題の次に必要となる視点は、資源としてのデータに対するアクセスの許可、つまり「**アクセス制御 (Access Control)**」である。Open Dataspaces では、ゼロトラストアーキテクチャを前提として、下位ネットワークレベルでのアクセス管理に頼らず、認証されたアクセス主体（ユ

アイデンティティの実在性が設計上担保されていなければ、これらの関係推論は正当性を失い、認可結果を説明・監査することができない。Open Dataspaces では認証と認可を中央集権的に統合しない。ドメインを束ねた各組織単位では、主体の最小限の同一性と、信頼・認可管理単位との対応関係を管理し、ドメイン単位の AQ ではコンテキストに即した関係と認可ルールを管理する。この責務分離により、以下のような設計によって担保された信頼構造が成立する：

- 各組織は自らの運用方針を保持できる
- 組織間であっても、個人単位の責任境界を維持できる
- 信頼の失効や更新が、認可判断に即座に反映される

Open Dataspaces におけるアイデンティティ設計とは、単に誰かを識別することではなく、組織・時間軸・責任分界をまたいでも説明可能な信頼を、構造として維持することだといえよう。

ーザ、組織）からアクセス対象（データやサービスなどのリソース）へのアクセス単位での権限認可を強化する。そのため、**PEP/PDP モデル**（Policy Enforcement Endpoint (PEP) /Policy Decision Endpoint (PDP)) を採用している (図 28)。

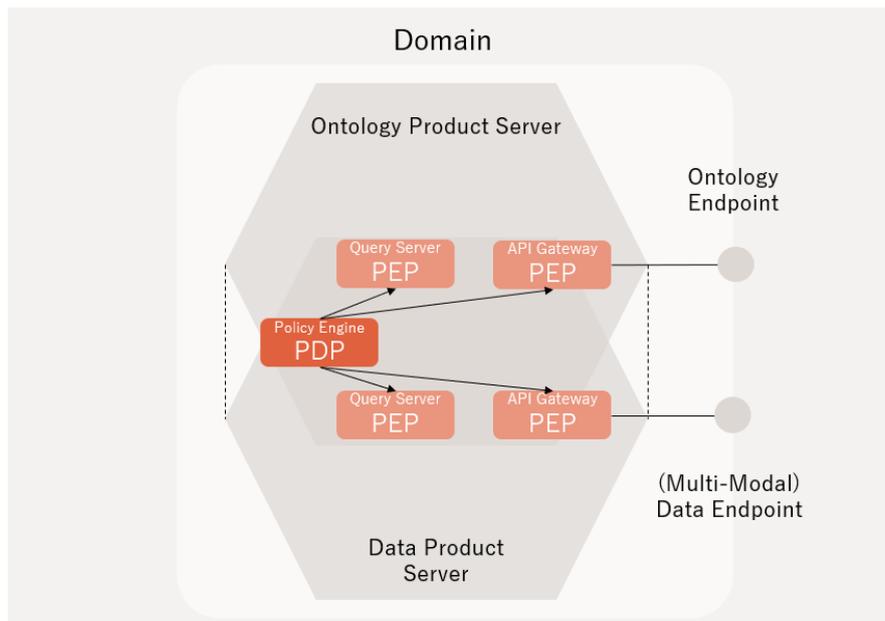


図 28 ゼロトラストアーキテクチャでの PEP/PDP モデルのイメージ

PEP は、Data Product 及び Ontology Product 双方の API ゲートウェイやクエリサーバーなど、実際にリクエストが通過する地点に配置される。一方、PDP はポリシーエンジンとしてドメインごとに配置され、アクセス可否の判断を一元的に行う。この分離による本質的な利点は、単に粗粒度・細粒度の制御を併用できる点に留まらない:

- ポリシー判断がドメイン内の一箇所に集約され、監査や説明責任が容易になる
- ポリシー変更が即座に全 PEP に反映され、運用の俊敏性が確保される
- API レベルとデータクエリレベルで多層的な強制が可能となる

なお、近年は、この設計を標準化しようとする [AuthZEN の仕様化が進んでいる](#)。

アクセス制御の判断モデルとして、Open Dataspaces では精緻なアクセス管理を、簡単に利用できる「**ReBAC (Relationship-Based Access Control)**」を採用し、関係性ベースで認可判断を行うこととしている。ReBAC では、主

体 (アイデンティティ) と資源 (リソース) の間に存在する関係性—所属、契約、委任、属性などをグラフとして表現し、その到達可能性や条件を評価することで許可・不許可を決定する。ここで重要なのは、PDP が行うのは確率的な推測ではなく、**アクセスポリシーグラフによる関係の評価・導出**である点である。なお、Open Dataspaces は RBAC (Role-Based Access Control) や ABAC (Attribution-Based Access Control) 等の他のポリシー言語の採用を否定するものではない点には留意いただきたい。

この ReBAC と Ontology のグラフを組み合わせることで実現する「**Graph-to-Graph Control**」(図 29) の真価は以下である:

- Alice という識別子 (IRI) を与えられると、彼女が人間で、製造業の会社に所属していて、その会社が NDA を締結して、一定のリソースに対してアクセス権限があることを、PDP が推測ではなく「**推論 (infer)**」できる。
- 資源 A というポインタを与えると、それ

に紐づく Ontology を推論して、適切な範囲のグラフに対して、アクセスを与えることができる。

さらに、このアーキテクチャにおいて、Alice は生身の人間でなくても、クローラーでも、Agentic AI でも代替可能である。

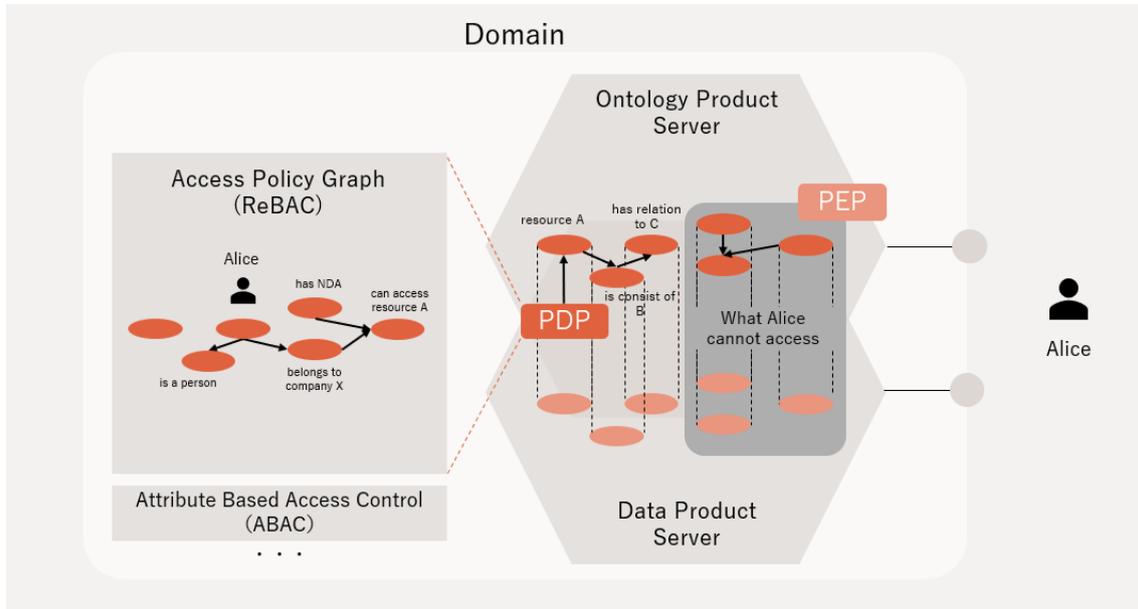


図 29 アクセスポリシーグラフと Ontology の関係性

なお、アクセス制御において Ontology は、認可判断そのものは行わない。**Ontology の役割は、「どこまでが同一の資源・同一のコンテキストに含まれるか」という意味的な境界を定義すること**にある。例えば、ある資源 A が複数のデータ要素や派生データから構成されている場合、その構成関係や同一性は Ontology によって表現される。PDP は、その Ontology に基づいて評価対

象となる資源グラフの範囲を確定し、その範囲に対して ReBAC を適用する。すなわち、Ontology は意味的な境界を与え、アクセスポリシーグラフは許可・不許可を決定するという役割分担が明確に保たれる。この分離により、意味の拡張や再解釈が、意図せずアクセス範囲を拡大することを防ぎつつ、柔軟な制御が可能となる。

## 利用制御—制度・技術非依存の柔軟な権利関係の選択肢

データが将来キャッシュフローを生み出す資源・プロダクトとしての側面を捉えると、「**利用制御 (Usage Control)**」の概念を導入することは、データ提供者に価値還元を行う市場インセンティブとして重要であるといえる。利用制御とは、アクセス制御を拡張したデータを扱う権利・義務的

な側面に着目した概念である。Steinbuss et al. (2021) では、データ処理の Provisions と Obligation を司るものとして利用制御を “the specification and enforcement of restrictions regulating what must(not) happen to data” と定義している (図 30)。

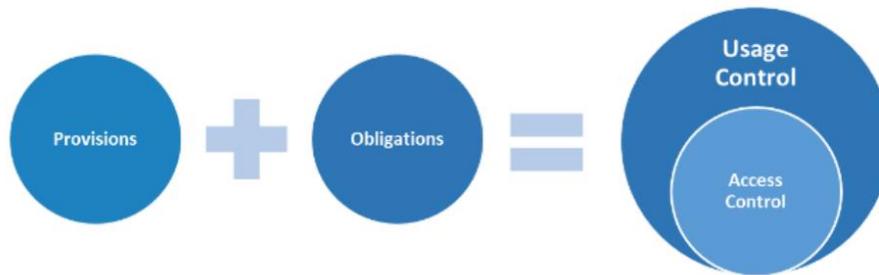


図 30 Usage Control consists of provisions and obligations  
(Steinbuss et al. 2021: Figure 4.)

利用制御は、そもそも Classical Dataspaces には存在しなかった後付けの概念である。なぜ、Open Dataspaces がこの概念を導入するのか？それは、データ提供者と利用者の間にある 2 つの非対称性についての補正を設計段階から取り込んでおくためである：

1. **権利義務関係の非対称性:** まず、一度データソースへのエンドポイントにアクセスを許した相手に対して、データ利用の用途・条件について提供者が制御する術を持たないのであれば、それはデータという資源を無配慮に提供し続けているに他ならない。データ提供者が、市場原理の中で適切なインセンティブを与えられ、自身のデータの利用制御ができることは、Open Dataspaces のスケーラビリティそのものに影響を与える重要な要素である。
2. **価格決定権の非対称性:** また、自己宣言的にデータ提供を行うことを是とする分散データマネジメントに、権利義務関係の非対称性のある状態で市場原理を導入すれば、価格決定権は実質的にデータ利用者が握ることになるだろう。これは、第 4 章の前提に記載したとおりデータ利用者が、信頼性や完全性、一貫性に関する明示的な保証を求めることによ

り、データ利用者の期待値に沿うことができるデータ提供者だけが選別されるためである。

このメカニズムは、前章で記載したように、データ提供者側の Ontology の品質向上などポジティブな要素ももたらす一方で、データ提供者はできるだけ安い価格で、かつ、できるだけ広い用途・条件でデータを利用することを最適戦略とするため、市場の構造として、安価かつ不利な用途・条件でのデータ提供が常態化するであろう。これは、本質的に価値の高いデータを有する提供者の参入を忌避させ、市場の失敗を招く可能性が高い。

従って、利用制御の概念をアーキテクチャレベルで想定しておくことは必須と言える。このような非対称性のある権利義務関係を補完するという意味で、「データ契約 (Data Contract)」は非常に重要な役割を果たすものであるといえよう。しかし、ここで注意しなければならない点がある。権利義務関係と密接な関係性を有する利用制御の手段を単一で硬直的な技術プロトコルで縛ることは、市場導入の大きなボトルネックになる可能性が高いということである。

利用制御という目的を果たすための技術的なアーキテクチャや手段は以下のように多様であり、これらは経済合理性の観点からマーケットにより選択されるべきである:

- BI や DI 領域 (例: サプライヤーを横断したカーボンフットプリント総量の計算) におけるブロックチェーンベースの秘密計算を利用した方法
- AI 領域におけるデータクリーンルーム

(Data Clean Room) での差分プライバシー学習

- 先端研究が進むマシンアンラーニング (Machine Unlearning)

Open Dataspaces ではこの問題意識のもとで、利用制御に関わる技術的な手段を制限することなく、インターフェースを提供するにとどめ、法域・制度的均質性を前提としない様々な方式を許容している (図 31)。

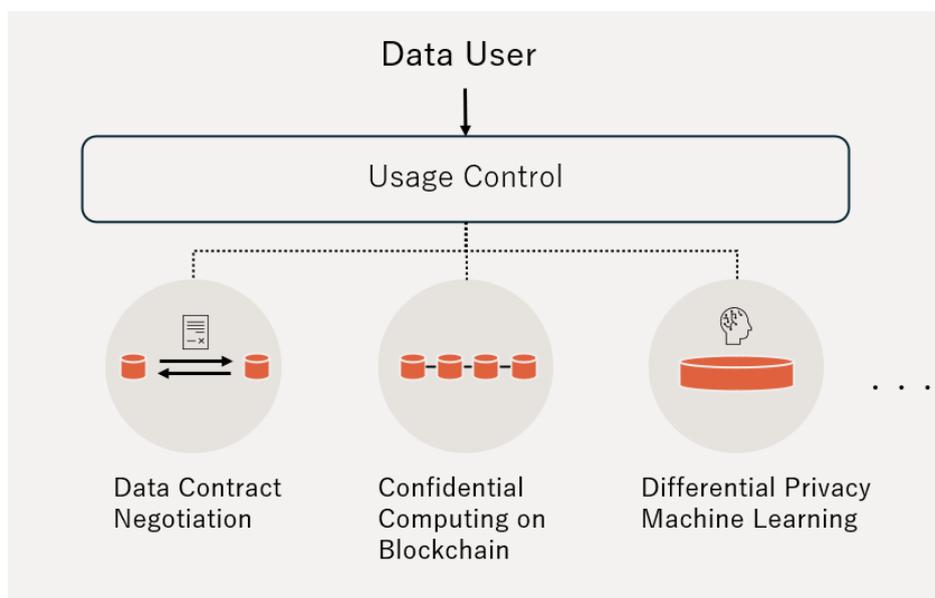


図 31 利用制御の多様性

その意味で、Contract Negotiation に代表されるデータと権利を紐づけて交換・交渉する方式は、数ある利用制御のうちの、ひとつの手段として扱われるべきであろう。ODS では、相互運用性の観点から Contract Negotiation もひとつのパターンとして想定した相互接続検証のための活動を国際的にも展開しているが、これはあくまで、多く

の技術的手段のうちのひとつの選択肢に過ぎない。以上から、市場からの要件由来ではなく、制度要件由来の技術的手段を強制することは、組織にその標準を遵守することにより得られる便益以上の適合コストを強いることになりかねないことは、十分に留意しなければならないだろう。

## 補論: 電子契約行為について

Open Dataspaces では法務的、技術的に実在する様々な利用制御をサポートするため、電子契約行為 (e-Contracting) のための技術的インターフェースを「**Heuristic Contracting Protocol**」としてオプションに用意することにした。Heuristic Contracting Protocol は、電子契約行為

の締結結果を PDP に反映させるところまでが所掌範囲となる (図 32)。従って、このプロトコルは独自の契約交渉手順を含むものではなく、あくまで、すでに大きな市場が存在する電子契約サービスを提供するサードパーティアプリケーションとの接続を想定したものである。

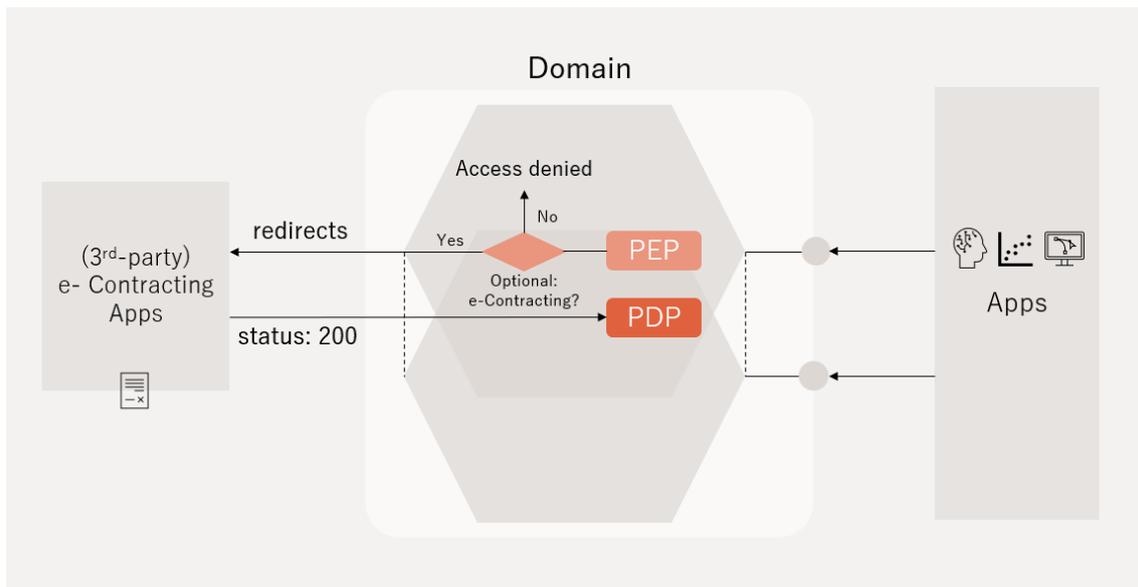


図 32 Heuristic Contracting Protocol (オプション) のイメージ

現状、分散データマネジメントにおけるデータ契約については、事前の規約雛形を前提とした事前契約で運用がなされていることが多い。UN/CEFACT における e-Negotiation 等での先行議論 (図 33) とグローバルでの企業法務慣行、商慣習の実態を踏まえ、データ共有に関する**契約交渉の自動完結を必須とすることは現時点で非現実的と判断した**。なぜなら、**契約は機械可読に還元できない余白を必ず持つため**である。

また、機械完結だけにこだわらず、企業法務部などによる最終判断を行うための Human-in-the-loop のコールバックも許容することも現実のビジネスでは重要である。これは、日常的な契約処理を行う現場と、高度な解釈や判断を支援的に担う法務部の関係性を踏まえた対策である。法域を横断したうえでもなお、Human-in-the-loop を限りなくゼロに近づける未来に向けて、電子自動契約の法務・監査行為に関する技術的な進歩と社会の成熟を期待したい。

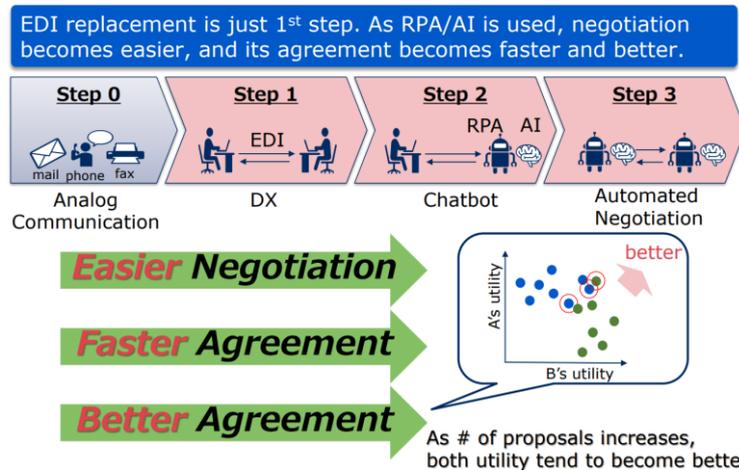


図 33 UN/CEFACT で想定されている e-Negotiation の段階的な進化 (Nakadai et al. 2020: p.7)

### 補論: 精算・課金／決済行為について

同様に、データ利用者による Data Product の利用に際した精算・課金／決済のためのエンドポイントも、契約行為と同様の発想で「Clearing and

Payment Protocol」というオプションなインターフェースを用意している (図 34)。

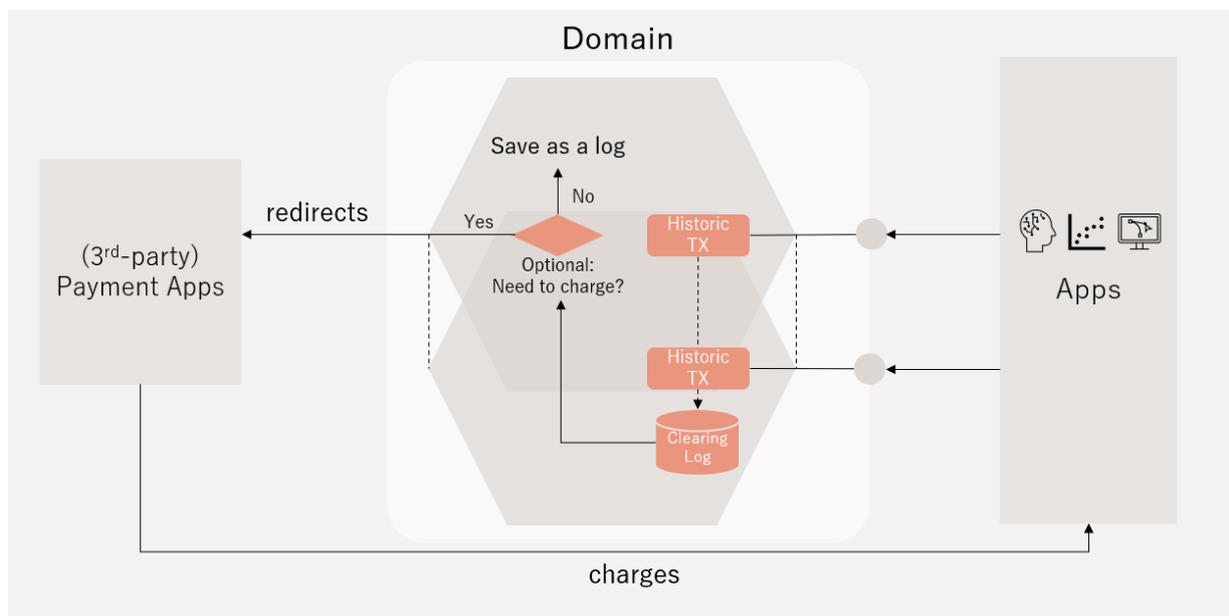


図 34 Clearing and Payment Protocol (オプション) のイメージ

以上のヒューリスティック契約、精算・課金／決済などの利用制御系のインターフェースを組み合わせることにより、分散データを取り扱えるデー

タマーケットプレイスといった高度なサービスも実現することが可能になるであろう。

なお、Open Dataspaces では、このようなオープンなインターフェースと機能を実現するため、トランザクションの横断的なログ・モニタリングを「**Common Functionality**」として位置付け、イベント検知やアラート、通知のような高度

なモニタリング機能を実現できるような余地を残している。ここではすべて触れないので、詳しくは、ODS Protocols における技術仕様等を確認されたい。

## 8. サービスモデル

最後に、このような DPQM を実現し、DDD を基軸とした分散データマネジメントをどのようなサービスモデルで実装するか、市場との対話を通じたアーキテクチャ設計時に、想定していた2つの実装パターンを記載する。

Open Dataspaces のアーキテクチャを実装する最もオーソドックスな方法は、(1) **ドメインオーナー自身が Self-Serve Data Platform を構築し、DPQM に基づいた Data Product 及び Ontology Product を提供する**という方式（「分散型サービスモデル」と呼称）である。しかし、これは大きなデジタル財源を持ち、自前で環境構築と運用を賄える大企業などで成立する話である。例えば、中小企業はドメインオーナーとし

て、規律準拠のための機能を自身で提供する企業体力がないことが多く、導入上の課題を抱えている。

そこで、(2) **DPQM を構成する基本的なソフトウェアスタックをマネージドサービス事業者**に代理で提供してもらいながら、**ドメインオーナーとして、Data Product 及び Ontology Product の提供に責任を持つ**、という方式（「連邦型サービスモデル」と呼称）も想定した。Open Dataspaces では、このコア技術のマネージドサービスを提供する仲介事業を、Classical Dataspaces で利用されていた用語になぞらえ「**Dataspace Service Provider (DSSP)**」と呼称する。

### 引用: Classical Dataspaces における DSSP の定義 (Franklin et al. 2005)

#### Classical Dataspaces において、DSSP は Dataspace Support Platform と呼称

- Must deal with data and applications in a wide variety of formats accessible through many systems with different interfaces. A DSSP is required to support all the data in the dataspace rather than leaving some out, as with DBMS.
- Although a DSSP offers an integrated means of searching, querying, updating, and administrating the dataspace often the same data may also be accessible and modifiable through an interface native to the system hosting the data. Thus, unlike a DBMS, a DSSP is not full control of its data.
- Queries to a DSSP may offer varying levels of service, and in some cases may return best effort or approximate answer. For example, when individual data sources are unavailable, a DSSP may be capable of producing the best results it can, using the data accessible to it at the time of query.
- Must offer the tools to create tighter integration of data in the space as necessary.

DSSP を提供する事業者は便宜上、「**仲介者 (Intermediaries)**」と呼称されることがある。仲介者はドメインオーナーでデータの利用制御を

行う事業者に DSSP の提供を通じてサービス関係を締結（連邦）することで、Open Dataspaces の裾野を拡大する役割を担う重要なプレイヤーで

ある。Open Dataspaces では、ドメインオーナーが自らオンボードすることと、DSSP の仲介を経てオンボードする方式の混成を「**Hybrid Service Model (HSM)**」と呼称することにした(図 35)。実際、車載蓄電池のトレーサビリティ管理を実現するための事例では、OEM や Tier1

に製品を供給する Tier2 以降のプレイヤーは中小企業の構成比率が高いことから、自動車・蓄電池業界が共同で設立した中立的なサービス運営事業体が仲介者として、蓄電池 CFP データの DSSP の提供を開始する事例も出てきている。

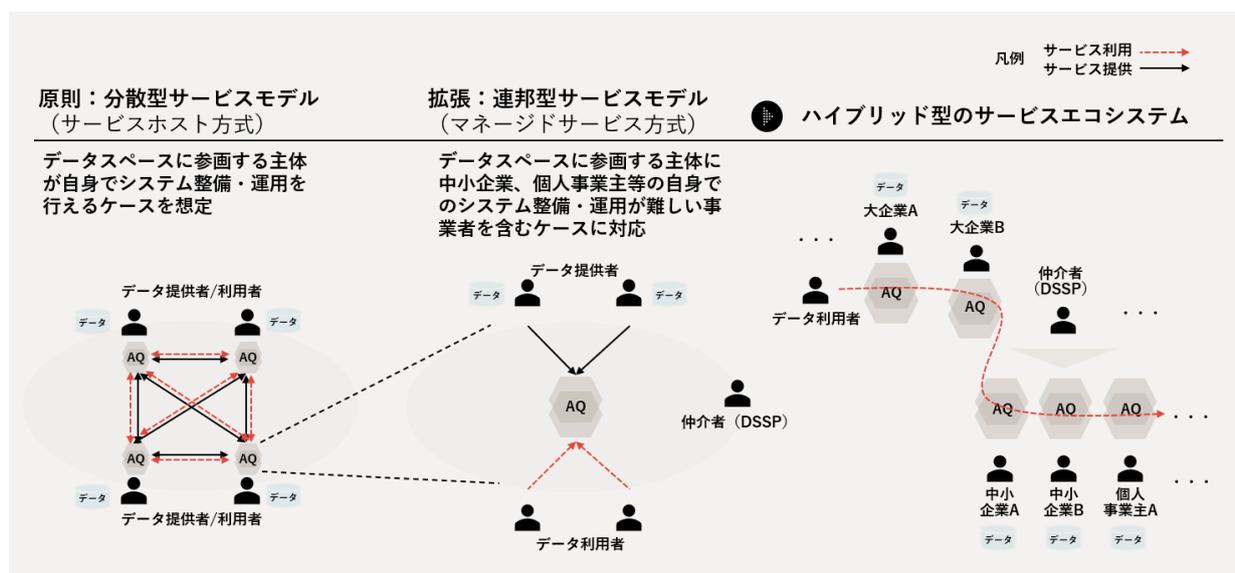


図 35 HSM の概要

HSM における仲介者の役割や、ドメインオーナーの利用制御の担保についての考え方は、ODS-

RAM をはじめとした関連文書を参照いただきたい。

## おわりに。

以上が Open Dataspaces の設計思想、アーキテクチャパラダイムとそれを構成する特徴である。より具体的な説明は設計物であるリファレンスアーキテクチャモデル (ODS-RAM)、プロトコル (ODS Protocols)、それを実現するための機能的な手段については、参照実装 OSS (ODS Middleware) のソースコードなどに譲りたい。

Open Dataspaces は市場での実証や企業商用レベルでの検証も終え、中核的なコンポーネントはメジャーリリースに近い状態であるが、いかにこれをスケーラブル、かつ持続的な形で発展させていけるかが、分散データマネジメントを社会に実装していくうえで非常に重要である。私たちは、この Open Dataspaces という革新を、伽藍とバザールの理念を胸に、全世界のオープンソースコ

ミュニティの力を信じて、前進させていきたいと考えている。Open Dataspaces の技術仕様やコンポーネントは、上述のとおりまだ産声をあげたばかりのリビングドキュメント (リビングソース) であり、改善の余地を多分に含んでいる。このアーティファクトは特定の強力なベンダーや、規制当局の論理からは切り離された、グローバルで民主的なプロセスで管理されることが望ましい。

IPA は、このような開かれたスキームの在り方を検討しており、グローバルでのさまざまな組織、開発者の協力が必要である。Open Dataspaces の理念や取組に共感をしてくれた読者がいれば、是非協力していただきたい。

**津田通隆**

**Open Data Spaces Chief Architect**

連絡先:

独立行政法人情報処理推進機構 デジタルアーキテクチャ・デザインセンター  
dadc-ods-ml@ipa.go.jp

## Bibliography

- **Dehghani Z.** (2019). How to Move Beyond a Monolithic Data Lake to a Distributed Data Mesh.
- **Dehghani Z.** (2022). Data Mesh: Delivering Data-Driven Value at Scale. O'Reilly Media.
- **Digital Agency.** (n.d.). Overview of DFFT. <https://www.digital.go.jp/en/policies/dfft/dfft-overview>
- **Epoch AI.** (2024). Will We Run Out of Data? Forecasting Dataset Size for Language Models. Technical Report, June 2024.
- **Franklin M, Halevy A, Maier D.** (2005). From databases to dataspace: a new abstraction for information management.
- **Halevy A, Franklin M, Maier D.** (2006). Principles of Dataspace Systems.
- **Nakadai S, Sugamata H.** (2020). 35<sup>th</sup> UN/CEFACT Forum Webinar Supply Chain Management and Procurement Domain meeting: eNegotiation Project.
- **New Energy and Industrial Technology Development Organization (NEDO).** (2025). Market Size Study of Data Spaces and Impact Modeling & Scenario Analysis Report. <https://www.nedo.go.jp/content/800039315.pdf>
- **Ministry of Economy Trade and Industry, Information-technology Promotion Agency.** (2025). Whitepaper: Ouranos Ecosystem Dataspace Reference Architecture Model.
- **Otto B. et al.** (2016). WHITEPAPER: Industrial Data Space. Fraunhofer.
- **Steinbuss S. et al.** (2021). Usage Control in the International Data Spaces. International Data Spaces Association.
- **Y. Tina Lee.** (1999). Information modeling from design to implementation. National Institute of Standards and Technology.

# Acknowledgement

## 校閲協力者

田村 公孝 / Kimitaka TAMURA アーキテクチャ戦略企画部 シニアエキスパート

青木 宣明 / Nobuaki AOKI 共通技術グループ 研究員

齊藤 元伸 / Motonobu SAITO 共通技術グループ 研究員

花谷 嘉一 / Yoshikazu HANATANI 共通技術グループ 研究員

今井俊博、禹ナリ、後川彰久、甲斐隆嗣、片山  
玲文、金子崇之、小島雄一郎、小林真輔、澤田  
悠希、清家大嗣、関堅吾、関享太、田嶋聡司、  
田中良夫、森拓郎、中島一雄、中台慎二、中村

龍太、松崎大樹、的野晃整、坂内健人、廣野正  
純、深津航、福田昌昭、藤原輝嘉  
(50音順、敬称略)

表紙／資料デザイン

津田通隆