



The Data Product Revolution: Unlocking Business Value

Presented by: William McKnight

“#1 Global Thought Leader and Influencer in Cloud and
in Data Center” Thinkers360

President, McKnight Consulting Group

3 X **Inc 5000**

 /in/wmcknight

www.mcknightcg.com
(214) 514-1444



Data Product Strategy

Build and Deploy with Informatica Data Management Cloud

Rob Marx - Data Governance Solution Specialist
March 2026

Where data & AI come to 

Choose Your Architecture. Build Your Data Products Strategy.

DATA MESH · DATA FABRIC · ARCHITECTURAL PATTERNS

DATA MESH

Decentralized Ownership

- › Domain teams own & govern their own data products
- › Products published to a central marketplace for discovery & reuse
- › Federated governance with global standards & data contracts
- › Enables team autonomy + cross-functional interoperability
- › Best for: Large enterprises with strong domain ownership culture

DATA FABRIC

Unified Metadata-Driven Intelligence

- › Abstracts complexity across all data sources, on-prem and cloud
- › Metadata-driven architecture provides unified view of all data
- › AI-powered automation of data discovery & cataloging
- › Centralized or federated governance; self-service marketplace
- › Best for: Organizations modernizing legacy, multi-cloud environments

Choose ONE, or combine BOTH on the Informatica Data Management Cloud (IDMC)

From Data Products to Business Outcomes

Package data products in layers · Align every asset to a measurable KPI · Prove business value end-to-end

HOW TO PACKAGE & PRIORITIZE

CUSTOMIZED

e.g. Equipment Configurator

SPECIFIC

e.g. Fraud Detection Engine

GENERAL

e.g. Inventory Dashboard

FOUNDATIONAL

e.g. Customer 360, Product Catalog

Build up from foundational →

HOW TO MEASURE BUSINESS VALUE

BUSINESS OUTCOME

CPO Goal: Sustainable Sourcing

STRATEGIC KPI

Data Product: Carbon Footprint

PROCESS METRICS

Local Sourcing · Recycled Sourcing · Alternate Sourcing

DATA PRODUCT METRICS

Supplier Profile · Bill of Materials

3x

Faster Time to Market

30%

Lower Cost of Ownership

Informatica = Your Next Move

Start with one stakeholder. One outcome. One data product. Then scale.

01

Identify Your Stakeholder Outcome

Pick one executive goal — revenue growth, cost reduction, compliance. Map the KPI they use to measure it. That's your north star.

02

Build Your First Foundational Product

Identify the core data asset that underpins your KPI — a Customer 360, Supplier Profile, or Product Catalog. Get it governed and into the marketplace.

03

Measure, Iterate & Scale

Track quality, utilization, and business impact. Use feedback to improve. Layer in General and Specific products. Show the value chain to leadership.

The architecture exists. The platform exists. The framework exists.

The only question is: which business outcome are you going to accelerate first?

Let's Build Together with IDMC →

Thank You!

Where data & AI come to 

McKnight Consulting Group Overview

- Turns information into a strategic asset, providing practical, experience-driven advice across Data Architecture, Databases including Vector & Edge, Data Integration & Streaming, Data Maturity, Analytics/Data Warehousing, Big Data, Data Governance/Quality & Data Observability, Master Data Management, & Data DevOps
- We are, by an order of magnitude, the most capable independent third-party cloud performance benchmarking organization with rapid results and comprehensive data analysis
- 100s of publications and 100s of speaking engagements
- Vendor-neutral, so all our recommendations are fully client-focused and aligned with business priorities
- Deliver efficient, timely, and politically sustainable outcomes, anticipating client needs well into the future
- Extensive experience overcoming complex information management challenges across industries, including healthcare, financial services, and large enterprises
- A 2-Time Inc. 5000 Company

What is a Data Product?

- Curated, well-defined datasets
- Packaged product offerings for internal or external consumers
- Delivered in a user-friendly, accessible, and consistent manner
 - Oftentimes as an Application Programming Interface or API
- Enables seamless data integration and consumption

Characteristics of a Data Product

Data-Centric

- Deliver structured, meaningful data (e.g., customer profiles, sales data, inventory levels)

Productized Approach

- Treated like a product with a clear purpose, documentation, versioning, and SLAs

Consumer-Focused

- Designed for ease of use by developers, data scientists, or business users

Security & Governance

- Include authentication, authorization, and data governance to ensure compliance

Self-Service Ready

- Enable users to fetch data on demand without heavy IT involvement, promoting self-service analytics

Database-Pipeline-Product Engineering Dependency Flow Example

Database Engineer

- Designs a schema optimized for sales transactions

Data Pipeline Engineer

- Creates a pipeline that extracts sales data, enriches it with customer information, and transforms it for analytics

Data Product Engineer

- Builds an endpoint to expose aggregated sales insights, enabling business users to create self-service reports

Why Data Products?



Data Sharing



Data Services



**Data
Interoperability**



Data Mesh

Data Sharing

The process of making data accessible across different teams, departments, or external partners in a controlled, secure, and efficient manner

- **Seamless Access:** Data products enable real-time, on-demand data sharing, eliminating manual file transfers
- **Controlled Sharing:** Data products define specific datasets for sharing, ensuring users get only the necessary data with security controls



Example

- An API for customer sales data allows marketing, sales, and external partners to access relevant data securely for campaign planning

Data Services

Functional components that provide data-related capabilities (e.g., querying, transformation, validation) as a service to other systems or users

- **Reusable Components:** Data products expose common data services, like customer lookups or inventory counts, that can be reused across multiple applications
- **Efficiency:** Reduces the need for redundant data processing logic in different systems by centralizing data services



Example

- A product inventory API that integrates with e-commerce platforms, ERP systems, and warehouse management systems, ensuring consistent stock data

Data Interoperability

The ability of different systems, applications, and organizations to exchange and use data in a compatible and standardized manner

- **Standardized Interfaces:** Data products standardize how different systems interact, making data exchange seamless across platforms
- **Cross-System Integration:** Data products facilitate integration between disparate systems by offering well-defined, accessible datasets



Example

- A real estate data product that combines property data from different MLS providers into a standardized format for use across multiple platforms, ensuring consistent property details

Support for the Data Mesh

A decentralized data architecture where data is treated as a product and owned by individual domains, enabling teams to independently manage and share their data

- **Decentralized Ownership:** Each domain (e.g., sales, marketing) can create and manage its data products, exposed via APIs, reducing bottlenecks from central IT
- **Scalability:** API-based data products allow domains to scale independently while maintaining data consistency and accessibility across the organization



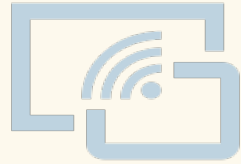
Example

- In a data mesh architecture, the sales team provides a Sales Performance API, while the marketing team offers a Customer Engagement API, enabling cross-functional insights through self-service BI tools

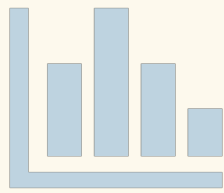
Function: Provide Data



APIs



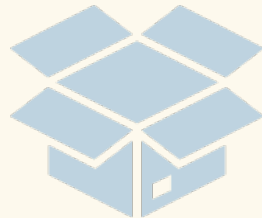
Feeds &
Streaming



BI &
Analytics



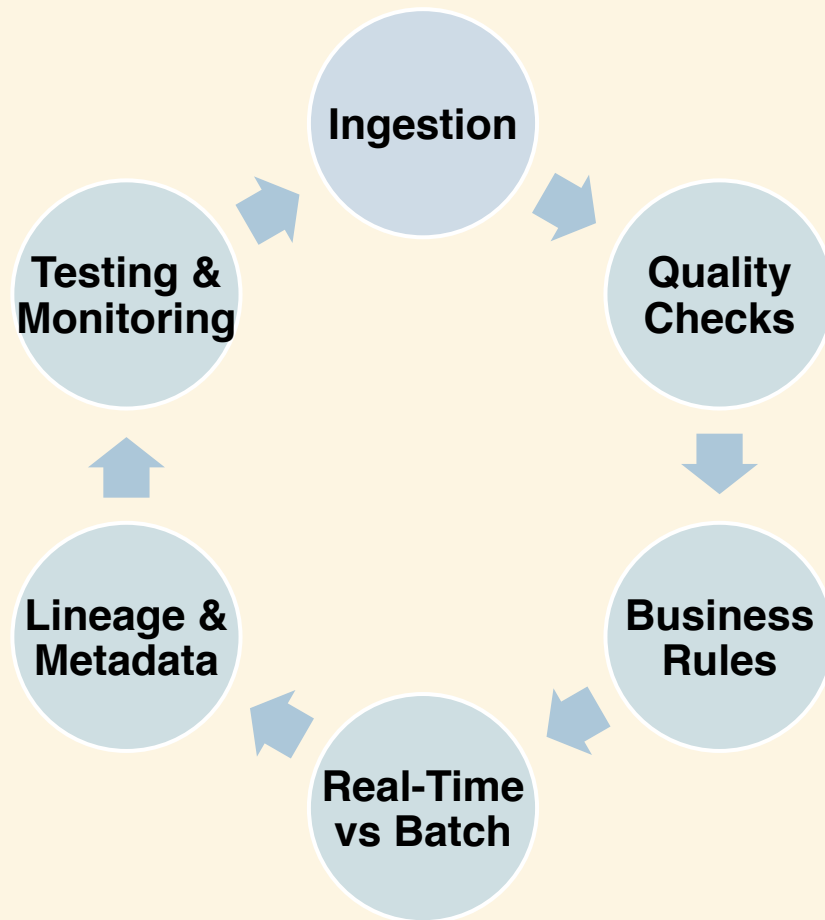
Files &
Exports



Embedded

- **How It Works:** Data products expose their data through APIs, allowing users or applications to access specific datasets programmatically
- **Real-Time Access:** Provides up-to-date data on demand
- **Flexible Integration:** Can be used across various applications (eg, web apps, mobile apps, third-party services)
- **Example:** A Sales Data Product API provides real-time sales figures to Power BI dashboards, mobile apps, and partner applications

Function: Validate & Assure Quality



Function: Transform Data

Modifying and enhancing raw data to make it more meaningful and usable

- **Aggregation:** Summarizing data (eg, total sales by region)
- **Normalization:** Converting data into a consistent format
- **Enrichment:** Adding external data (eg, demographics or market trends)
- **Example:** Enriching customer purchase data with demographic information to enable segmentation analysis

Function: Organize Data

Organization makes data more accessible, understandable, and actionable for both technical and non-technical users by using:

- **Domain-Centric Structure**

- Data is organized around specific business domains (eg, sales, marketing, customer service)
- Makes it easier for teams to access relevant data without navigating unrelated datasets
- **Example:** A Sales Data Product delivers pre-aggregated sales metrics like revenue, transactions, and customer churn, allowing sales teams to quickly build reports

- **Predefined Schema and Metadata**

- Creates a clear schema (data structure) and metadata (data about the data), including field definitions, data types, and relationships
- Ensures consistency and clarity, making data easier to understand and integrate into tools like Power BI or machine learning models
- **Example:** A Customer Data Product with metadata explaining fields like “Customer_ID,” “Purchase_History,” and “Lifetime_Value” helps business users quickly grasp its content

Function: Profile Data

- Data Profiling is the process of examining, analyzing, and summarizing datasets to understand their structure, content, and quality
- Data products, when designed and implemented with APIs, pipelines, and governance, can significantly enhance this process

| Method | Benefit | Example |
|--|---|--|
| Centralized Datasets | Simplifies data profiling by reducing data silos and inconsistencies | Profiling a single Customer Master Data Product instead of multiple customer databases |
| Real-Time Access via APIs | Enables continuous and automated profiling for real-time data insights | Monitoring real-time sales data for anomalies |
| Metadata & Schema Documentation | Provides clarity on data structure, making profiling easier and faster | Validating fields in a Product Catalog API using documented constraints |
| Governance and Quality Rules | Enforces data standards, enabling automated identification of data quality issues | Detecting incomplete records in a Supplier Data Product |
| Self-Service Profiling | Empowers business users to profile data independently, reducing reliance on IT | Marketing team using Power BI to profile customer engagement data |

Data Products by Structure

| Data Type | Definition | Examples | Use Cases |
|-------------------------|--|--|--|
| Structured | Organized in predefined schemas with fixed data types | Relational databases, financial transactions | BI dashboards, financial reporting, CRM systems |
| Unstructured | No predefined format or schema | Emails, videos, social media posts | Sentiment analysis, multimedia search, text analytics |
| Semi-Structured | Partially organized with tags or markers, flexible schema | JSON from APIs, XML files, CSV with inconsistent columns | Web APIs, NoSQL databases, dynamic data models |
| Multi-Structured | Combines structured, semi-structured, and unstructured data from various sources | Customer 360 views, IoT datasets, omnichannel marketing data | Customer insights, IoT analytics, big data integration |

Data Products by Granularity

| Data Type | Definition | Key Characteristics | Example Use Case |
|-------------------------|--|---|--|
| Raw Data | Unprocessed, original data from source systems | Granular, uncleaned, and may contain errors or inconsistencies | Raw IoT sensor data for environmental monitoring |
| Transformed Data | Cleaned and structured data ready for analysis | Consistent, validated, and formatted for specific use cases | Processed customer order data used in operational dashboards |
| Aggregated Data | Summarized data providing high-level insights | Smaller in volume, grouped by key dimensions (e.g., time, location) | Monthly sales totals by region for executive reporting |
| Derived Data | New data generated from existing data through models or algorithms | Enriched with additional insights, often used in advanced analytics | Customer churn prediction based on historical interaction data |

Business Domain Data Product

Example: Sales

| Data Type | Description | Example Use Case in Data Products |
|---------------------------|--|---|
| Transactional Data | Records of sales transactions including customer, product, and revenue details | Sales Performance Data Product for tracking revenue trends |
| Customer Data | Information about customers (name, contact, demographics, history) | Customer Segmentation for personalized marketing |
| Product Data | Details of products sold, including SKUs, categories, and prices | Product Catalog for e-commerce platforms |

Business Domain Data Product

Example: Marketing

| Data Type | Description | Example Use Case in Data Products |
|---------------------------------|---|---|
| Campaign Data | Data on marketing campaigns, channels, performance metrics | Campaign Analytics Data Product to assess ROI by channel |
| Lead Data | Information about potential customers and their engagement levels | Lead Scoring API to prioritize high-potential leads |
| Customer Engagement Data | Interaction data from emails, social media, and websites | Engagement Metrics API for tracking user activity and conversion |

Data Product Key Processes

- **Security and Access Authorization**
- **Error Checking**
- **Composing Response to Request**

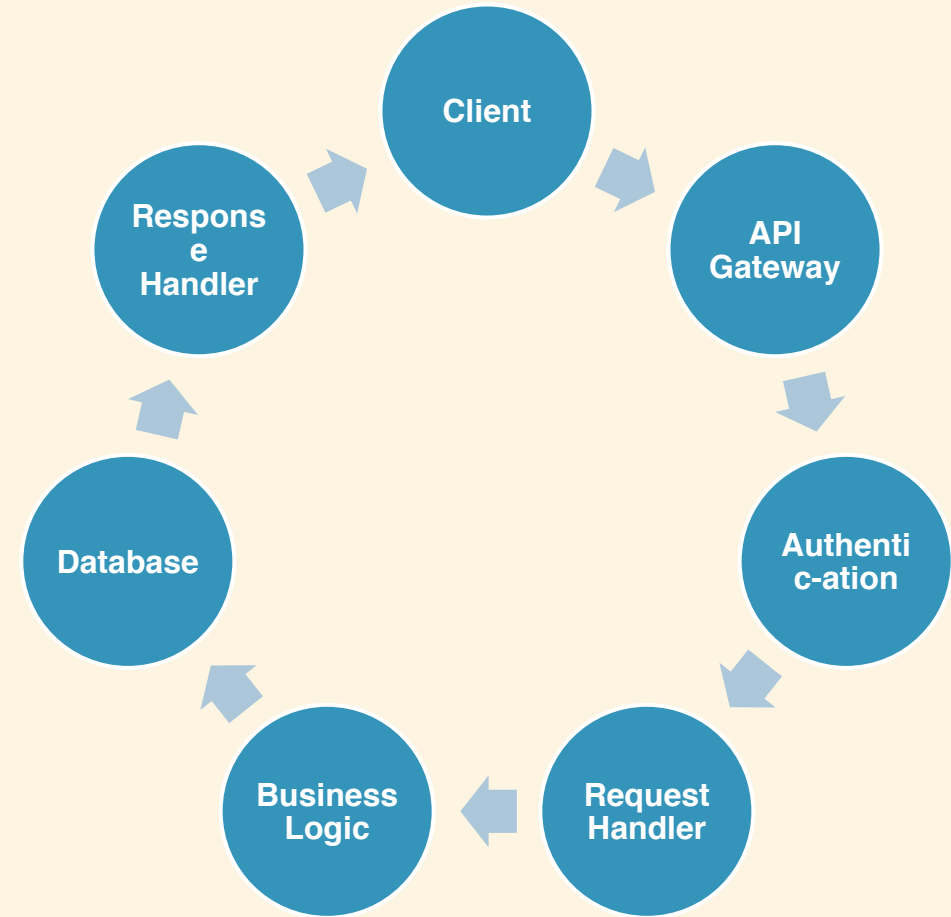


EXAMPLE

A Customer Data API uses OAuth 2.0 for authentication, granting read-only access to analysts while allowing data engineers full edit access. Unauthorized attempts are logged and monitored

API Architecture Overview

- API = Application Programming Interface
- Defines how different components of an application communicate with each other and with external systems
- Includes various layers to handle requests, process data, and return responses



API Architecture Components

| Layer | Description | Example Technologies |
|---|--|---|
| Client | The interface where users or applications interact with the API via HTTP requests | Browsers, Mobile Apps, Postman, cURL |
| API Gateway/Proxy | Manages and routes incoming requests, performs load balancing, rate limiting, and security checks | Kong, NGINX, AWS API Gateway |
| Authentication & Authorization | Validates user identities and determines access rights to ensure secure communication | OAuth 2.0, JWT, API Keys, SAML |
| Request Handling | Processes incoming requests, validates parameters, and routes to the appropriate service or function | Flask, FastAPI, Express.js, Spring Boot |
| Business Logic | Contains the core application logic that processes data and handles transactions | Python, Java, Node.js, .NET |
| Database | Stores and manages data. Could be relational or non-relational databases, or external data sources | PostgreSQL, MongoDB, Redis |
| Response Handling | Formats and returns data to the client in a structured way, typically as JSON, XML, or HTML | JSON, XML, CSV |

API Architectural Styles

★MOST
POPULAR

- **REST** (Representational State Transfer)

- **Key Features:** Stateless, uses standard HTTP methods (GET, POST, PUT, DELETE), and returns responses in JSON or XML
- **Advantages:** Simple, scalable, widely adopted, and compatible with web standards
- **Example:** A RESTful API fetching user data: GET /users/{id} returns user details in JSON

- **GraphQL**

- **Key Features:** Clients can specify exactly what data they need, reducing over-fetching or under-fetching
- **Advantages:** Efficient data retrieval, single endpoint for multiple queries
- **Example:** A query to fetch a user's name and email

- **SOAP** (Simple Object Access Protocol)

- **Key Features:** XML-based protocol that supports more complex operations and stricter security
- **Advantages:** Suitable for enterprise systems with strong security and transaction requirements
- **Example:** A SOAP request to fetch user info involves sending XML payloads over HTTP

- **gRPC** (Google Remote Procedure Call)

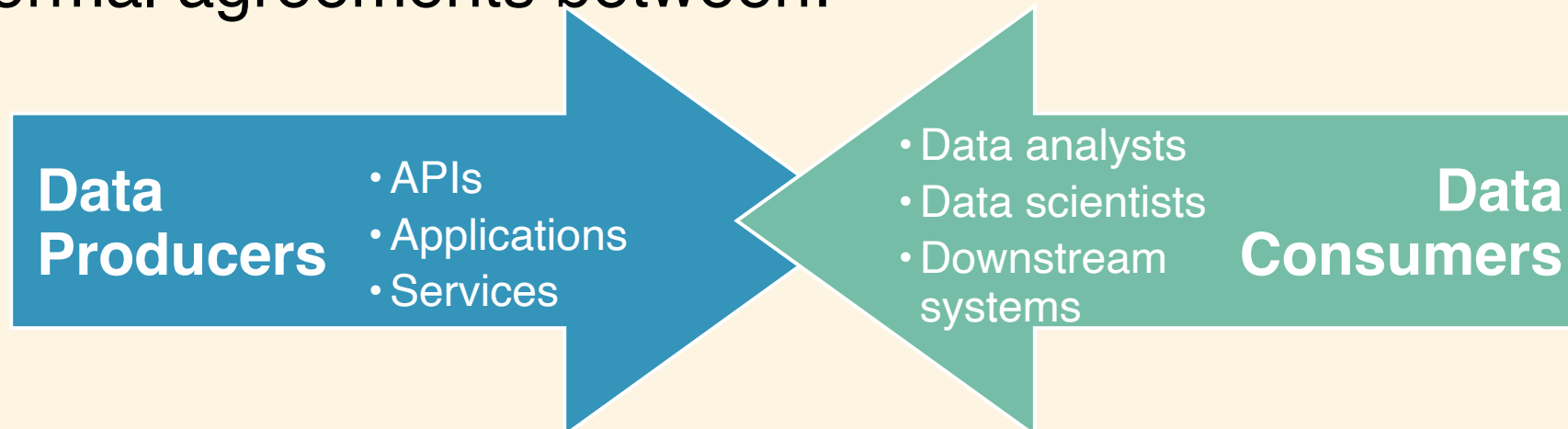
- **Key Features:** Uses Protocol Buffers (binary serialization), supports streaming, and is highly efficient
- **Advantages:** Faster, more efficient, supports bidirectional communication
- **Example:** Defining a user service in a .proto file, which clients use to generate code

Responding to Requests

| Aspect | Description | Example |
|-------------------------|---|---|
| Response Formats | Data products deliver responses in structured formats like JSON, XML, or CSV for easy integration | A Sales API returns a JSON response containing daily sales totals |
| Status Codes | HTTP status codes indicate the success or failure of the request | A GET request for customer data returns a 200 OK if successful |
| - 2xx (Success) | Indicates successful processing | 200 OK, 201 Created |
| - 4xx (Client Error) | Indicates issues with the request | 400 Bad Request, 401 Unauthorized |
| - 5xx (Server Error) | Indicates server-side issues | 500 Internal Server Error, 529 Too Many Requests |
| Response Body | Includes the requested data or an informative message in a structured format, often accompanied by metadata | A Customer API response includes customer data, along with record count and timestamp |

What are Data Contracts?

- Formal agreements between:



- Define the **structure, quality, and rules** governing how data is produced, shared, and consumed
- Ensure data consistency, reliability, and clarity, much like how APIs define communication protocols between systems

Benefits of Data Contracts

✓ Improved Data Quality

Consistent, reliable data by enforcing validation and integrity rules

✓ Enhanced Collaboration

Clear expectations between data producers and consumers, reducing miscommunication

✓ Data Governance

Rules around data security, privacy, and compliance

✓ Scalability & Maintenance

Smooth integration of new data producers and consumers by providing clear versioning and compatibility guidelines

✓ Early Error Detection

By defining strict validation rules, issues can be caught early, preventing downstream data corruption

Tools Supporting Data Contracts



**Great
Expectation**

- For defining and validating data expectations



**Apache Avro
Google Protobuf**

- Schema definition and validation in data streaming



**dbt (Data Build
Tool)**

- For defining and enforcing data models in data pipelines



JSON Schema

- A common standard for defining and validating JSON-based contracts

Example JSON Data Contract

{

"fields": [

{ "name":
"user_id",

{ "name": "email",

{ "name": "signup_date",

"type": "integer",
"constraints": {
 "unique": true,
 "not_null": true

"type": "string",
"constraints": {
 "not_null": true,
 "pattern": "^[\\w.%+-]+@[\\w.-]+\\. [a-zA-Z]{2,}\$"

"type": "timestamp",
"constraints": {
 "not_null": true

}},
] },

}},

"slas": {

"security": {

"latency": "less than 100ms",
"uptime": "99.9%"},

"encryption": "AES-256",
"access_control": ["admin",
"data_consumer"] }

}

API Design Principles

Consumer-Centric Design

- Understand the needs of data consumers and design APIs and data products to meet those requirements
- Provide clear documentation with examples to make it easier for users to onboard and consume the data

Modular and Reusable

- Design APIs and data products as modular components that can be reused across multiple applications and use cases
- Avoid tightly coupling components to enable easier updates and scalability

Schema Design and Contracts

- Define schemas and data contracts to establish clear expectations between producers and consumers
- Use versioning to introduce changes without breaking existing integrations

Summary

- The Data Product shifts data from a byproduct to a curated, high-quality asset designed for internal or external consumers, delivered via consistent interfaces like APIs.
- The Four Pillars of Value: Focuses on Data Sharing for seamless access Data Services for reusability, Interoperability through standardized interfaces, and Efficiency by reducing redundant processing.
- Data Products support a "Domain-Driven" approach where business units (e.g., Sales, Marketing) take end-to-end ownership of their data assets rather than relying on a central IT bottleneck.
- Data Products utilize robust API architectures as the primary mechanism for receiving, interpreting, and responding to data requests reliably.
- Data Products are enforced via Data Contracts.





The Data Product Revolution: Unlocking Business Value

Presented by: William McKnight

“#1 Global Thought Leader and Influencer in Cloud and
in Data Center” Thinkers360

President, McKnight Consulting Group

3 X **Inc 5000**

 /in/wmcknight

www.mcknightcg.com
(214) 514-1444

