Data Mesh

Introduction

JOCHEN CHRIST
@JOCHEN_CHRIST

Hi,
I am Jochen

Jochen Christ

Data Mesh Consultant
Product Manager Data Mesh Manager

💪 *Java*
👍 *Data Mesh*
🤩 *Data-driven Product Development*

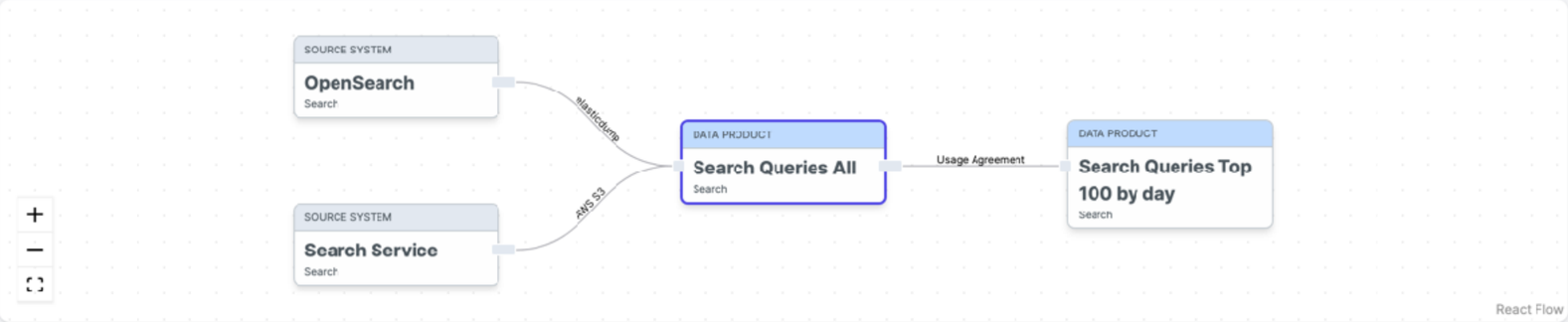
Data Mesh Manager | Data Products | Data Contracts | Global Policies | Teams | ACME

Data Products > Search Queries All

Search Queries All

Search | source-aligned | active | managed | demo

Show specification | Edit | Request Access



Info

Information about the data product

Data Catalog | Repository | Documentation

Name
Search Queries All

ID
urn:dataproduct:search:search-queries-all

Description
All search queries and results since 2020.

Metrics

Monitor business value, costs, and compliance

Consumers
1

Costs
$6,560.00

Compliance
0 / 1 policies

Input Ports

The source of the data, source systems or other data products.

OpenSearch
acme.search.clicks

Output Ports

Technology, dataset, and version of provided data.

search_queries_all_v1
SEARCH_DB.SEARCH_QUERIES_ALL_NPII_V1
Data Contract | 1 usage

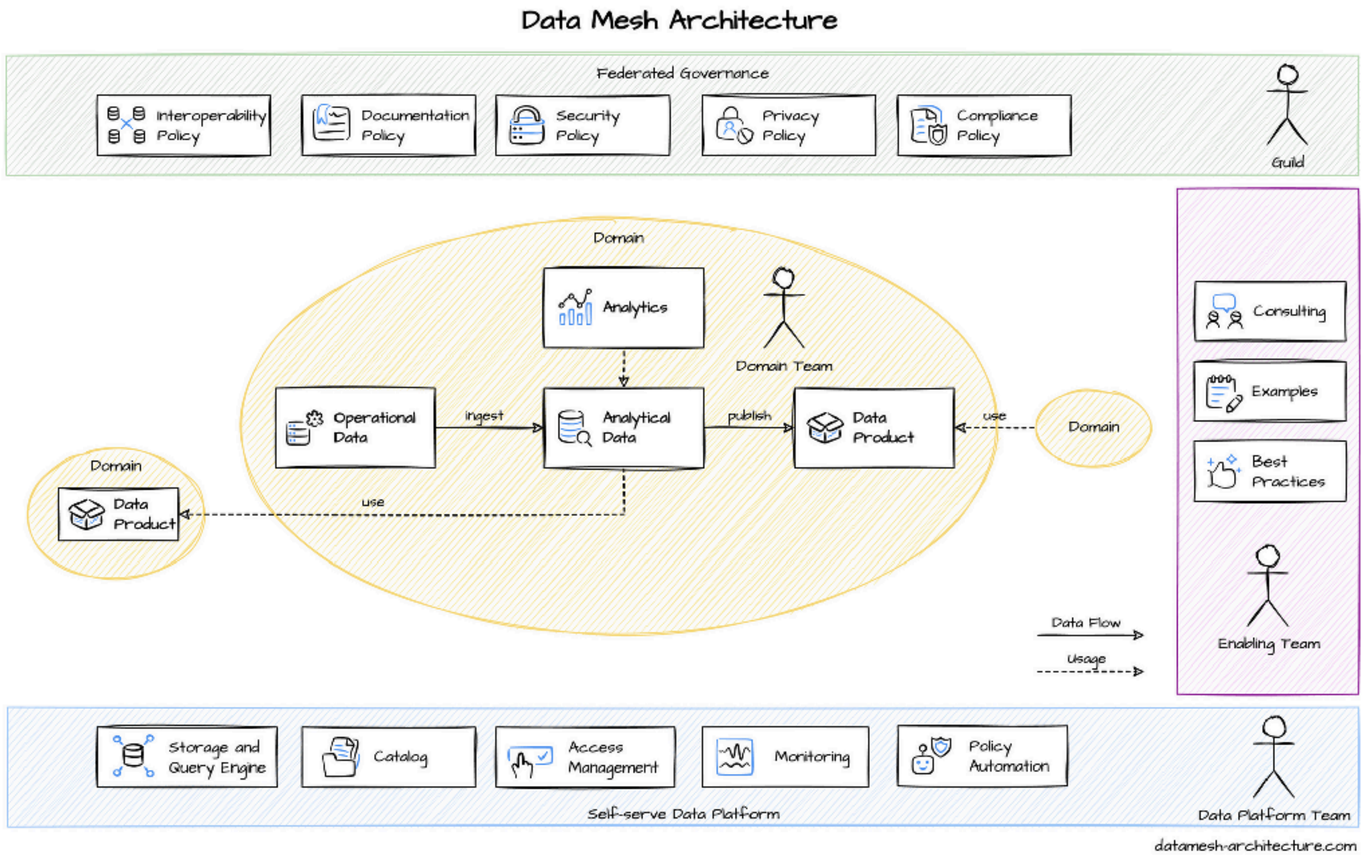
datamesh-manager.com

Data Mesh Architecture

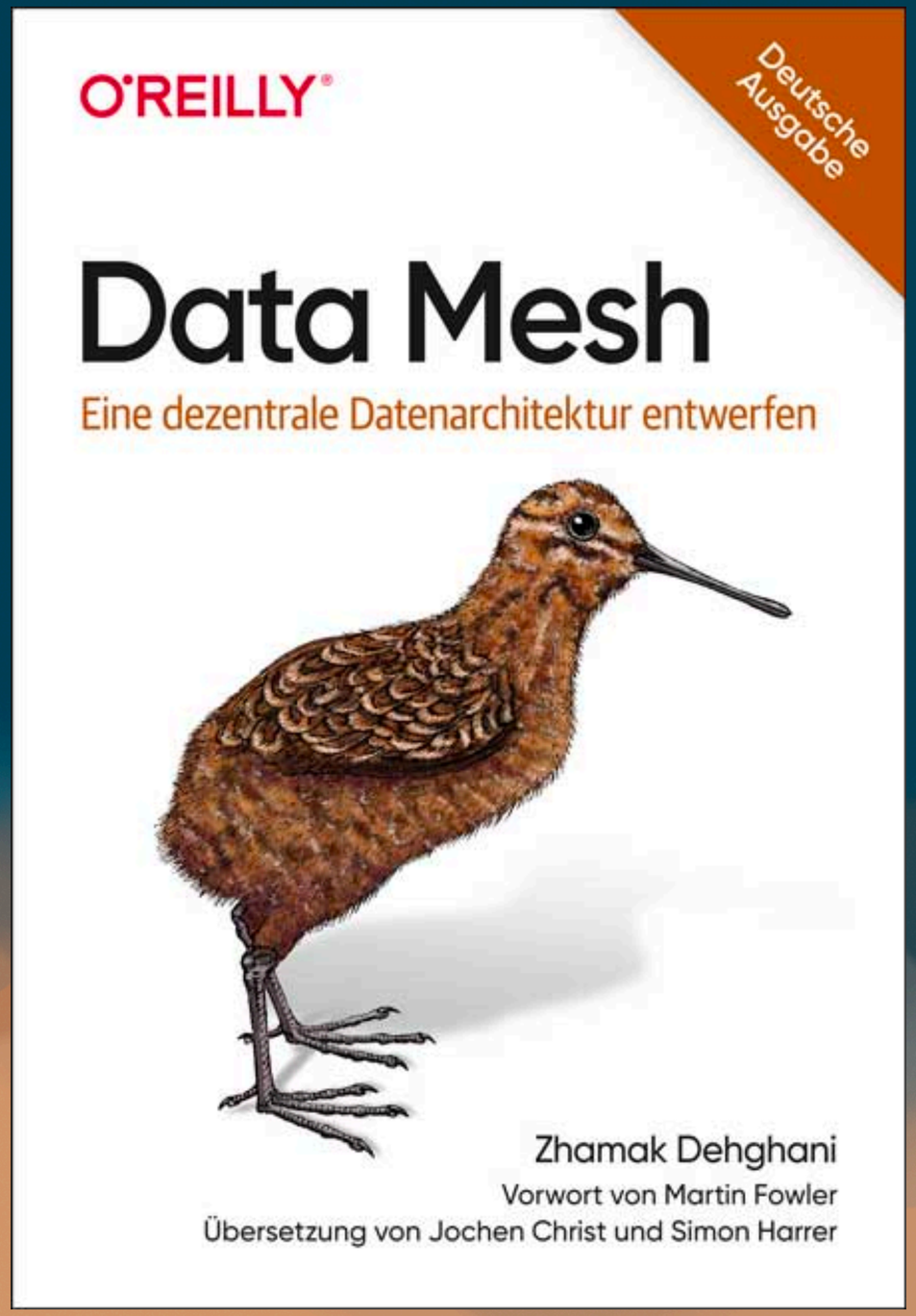
https://www.datamesh-architecture.com

How To Design a Data Mesh?

A data mesh architecture is a decentralized approach that enables domain teams to perform cross-domain data analysis on their own. At its core is the domain with its responsible team and its operational and analytical data. The domain team ingests operational data and builds analytical data models to perform their own analysis. It uses analytical data to build data products based on other domains' needs.

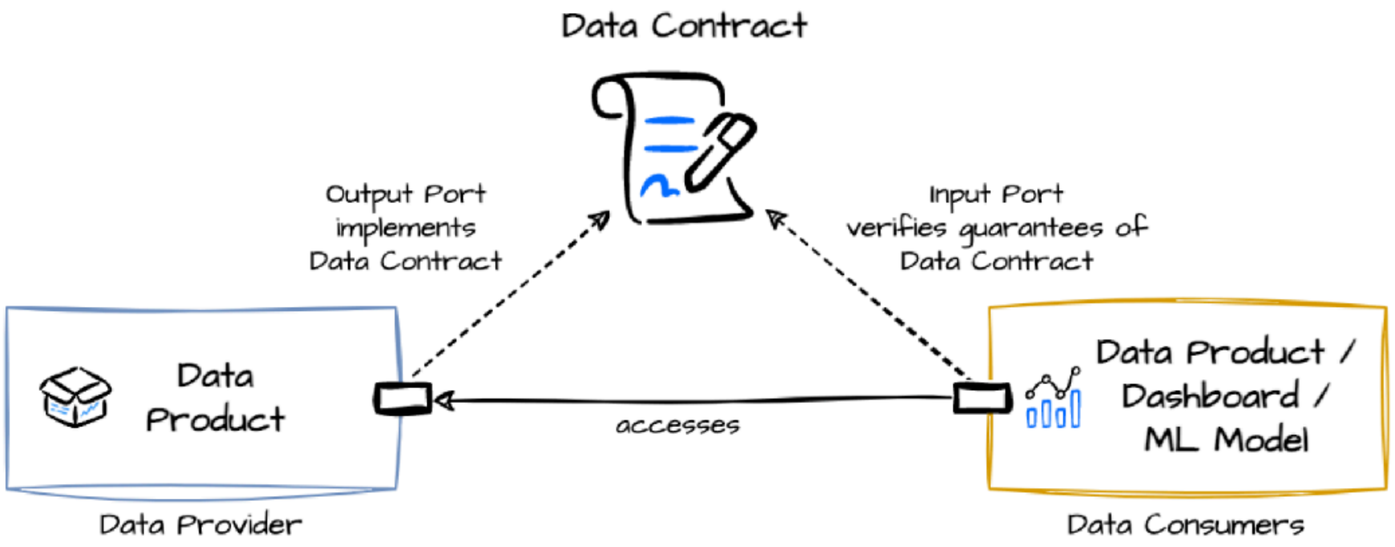


datamesh-architecture.com


O'REILLY
Deutsche Ausgabe
Data Mesh
Eine dezentrale Datenarchitektur entwerfen
Zhamak Dehghani
Vorwort von Martin Fowler
Übersetzung von Jochen Christ und Simon Harrer

datacontract.com

Data Contract Specification



Data contracts bring data providers and data consumers together.

A *data contract* is a document that defines the structure, format, semantics, quality, and terms of use for exchanging data between a data provider and their consumers. A data contract is implemented by a data product's output port or other data technologies. Data contracts can also be used for the input port to specify the expectations of data dependencies and verify given guarantees.

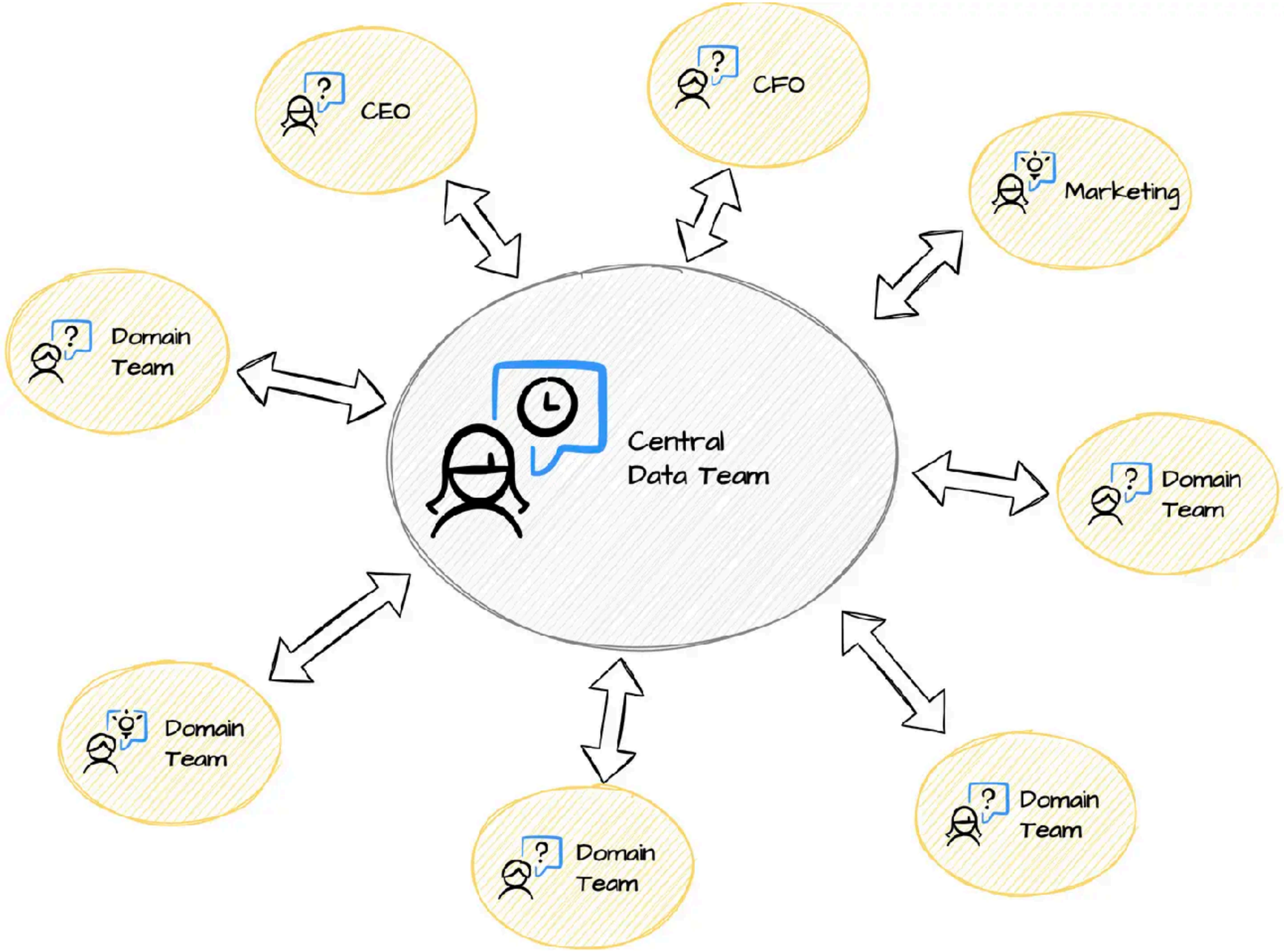
The *data contract specification* defines a YAML format to describe attributes of provided data sets. It is data platform neutral, yet supports well-known formats to express schemas (e.g., dbt models, JSON Schema, Protobuf, SQL DDL) and quality tests (e.g., SodaCL, SQL queries) to avoid unnecessary abstractions. The data contract specification is an open initiative to define a common data contract format. Think of an OpenAPI specification, but for data sets.

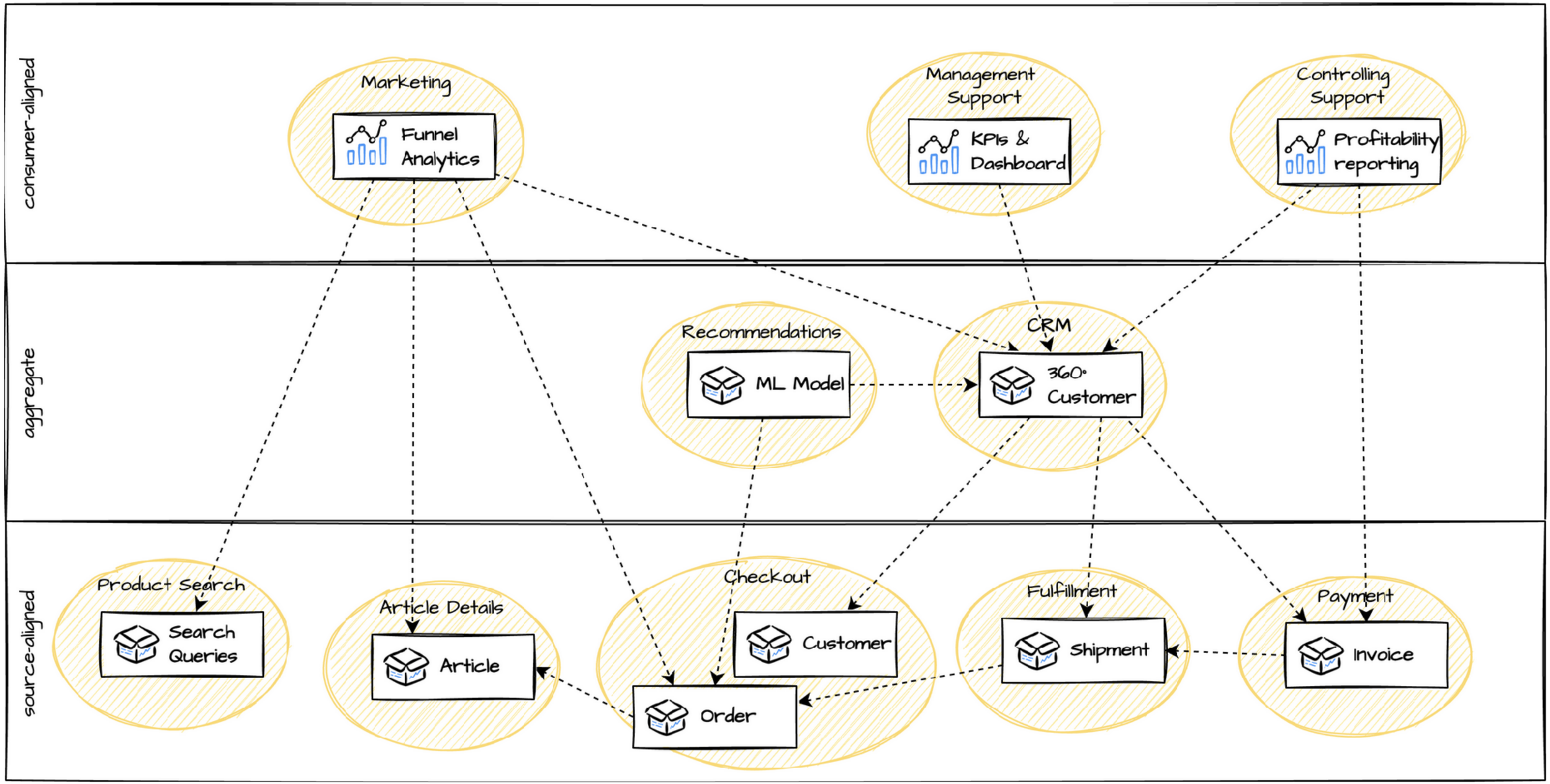
datacontract.com

What?

A decentralized data architecture gives
ownership and competence
for (analytical) data
to the teams that
understand the business context.

-- Jochen


CEO
CFO
Marketing
Domain Team
Central Data Team
Domain Team
Domain Team
Domain Team
Domain Team


consumer-aligned
aggregate
source-aligned
Marketing
Funnel Analytics
Management Support
KPIs & Dashboard
Controlling Support
Profitability reporting
Recommendations
ML Model
CRM
360° Customer
Product Search
Search Queries
Article Details
Article
Checkout
Customer
Order
Fulfillment
Shipment
Payment
Invoice

Decentralized Data Architecture

Why?

Make qualified data-driven decisions in your domain

Use data to better understand your users and system behavior. Derive features from insights, qualify value, and fast iterations. Also qualified rejection of unnecessary tasks.

Do the right things, purpose, motivation

Build innovative services in your domain

Enhance your customer experience with data technologies, such as LLMs, visualizations, classifications, and ML models for predictions and recommendations.

Customer value through innovation

Provide data as business value for other domains

Domain data is valuable for other business units as reference data and to aggregate. Needs managed, explained, high-quality and easy accessible data as products.

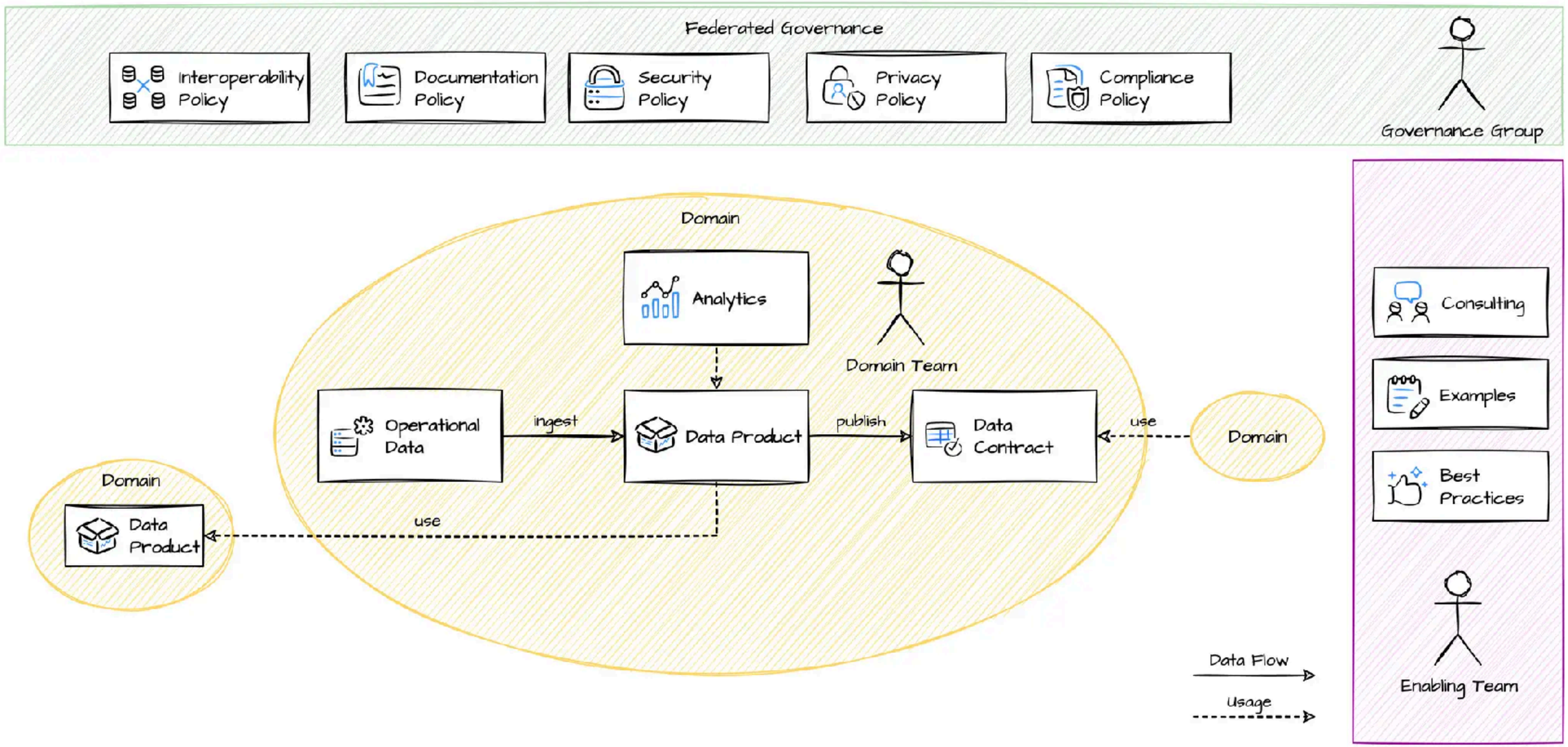
Company success

What Is Data Mesh?

	Domain Ownership	Data as a Product	Self-serve Data Platform	Federated Governance
Strategic Domain-driven Design	Domain Bounded Context	Product Thinking	Domain-agnostic	Context Mapping
Socio-technical Perspective	Domain Teams	Data Product by Domain Team	Data Platform Team	Guild
Technology	Operational & Analytical Data	Interoperability Interfaces	Self-serve Data Platform	Data Governance & Automation

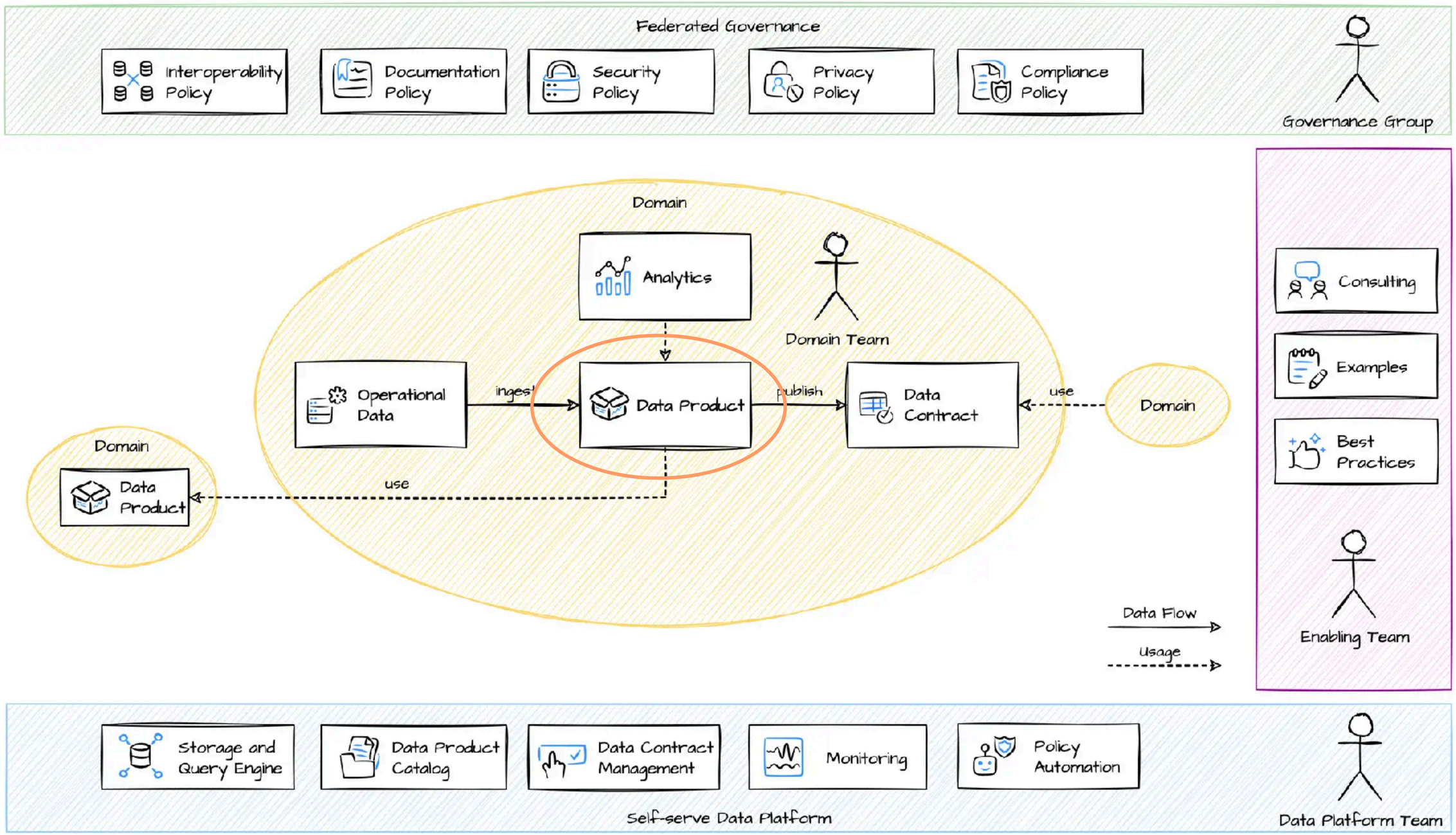
datamesh-architecture.com

Data Mesh Architecture



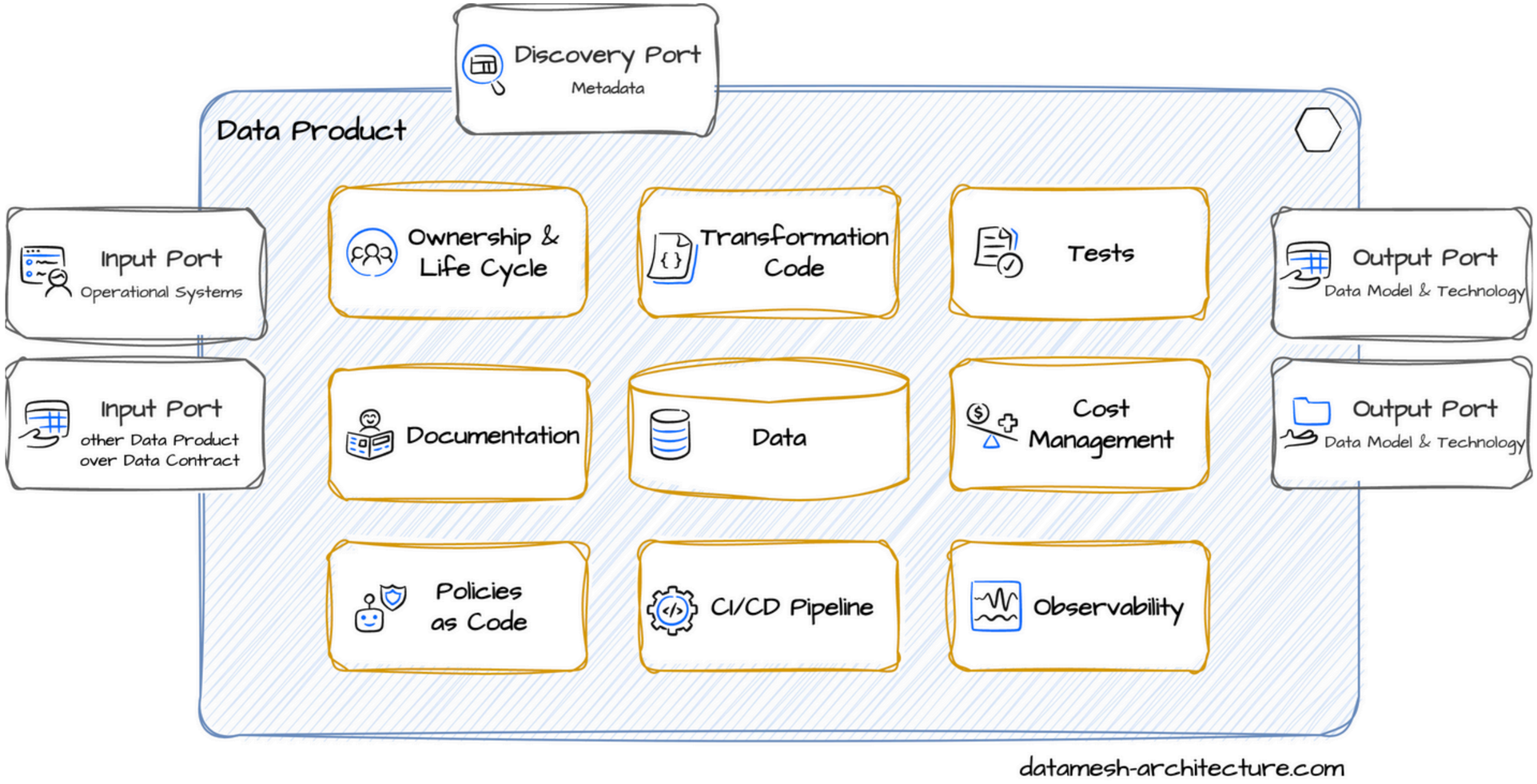
datamesh-architecture.com

Data Mesh Architecture



datamesh-architecture.com

Data Products are Modules



Output Port Example

Row	sku	location	available	updated_at
1	9520010951145	20	0	2021-02-28 12:29:21 UTC
2	9520010951145	20	1	2021-03-02 09:07:21 UTC
3	9520010951145	20	0	2021-03-03 16:36:21 UTC
4	9520010951145	20	1	2021-03-04 13:03:21 UTC
5	9520010951145	20	2	2021-03-05 17:26:21 UTC
6	9520010951145	20	3	2021-03-06 03:35:21 UTC
7	9520010951145	20	2	2021-03-06 17:25:21 UTC
8	9520010951145	20	1	2021-03-07 18:10:21 UTC

- Technical Endpoint
- Hides implementation details
- Large data set
- Read-only
- Technology
  - Tables
  - Files in Bucket
  - Topic
- Data Model
  - With PII
  - Without PII
- Version

Implementation depends on Tech Stack

Stack	Storage	Query Engine	Framework
AWS	S3	Athena (SQL)	Lambda / Step Functions
Google Cloud	BigQuery	BigQuery (SQL)	dbt
Azure (MS Fabric)	OneLake	Spark	Fabric notebook
Databricks	Deltalake	Spark	Databricks Asset Bundles
Snowflake	Snowflake	Snowflake (SQL)	dbt
Trino	S3 compliant	Trino (SQL)	dbt
Dremio	S3 compliant	Dremio Sonar	dbt
Java	S3 compliant	Java	Spring Cloud Data Flow
DuckDB	S3 compliant	DuckDB (SQL)	dbt

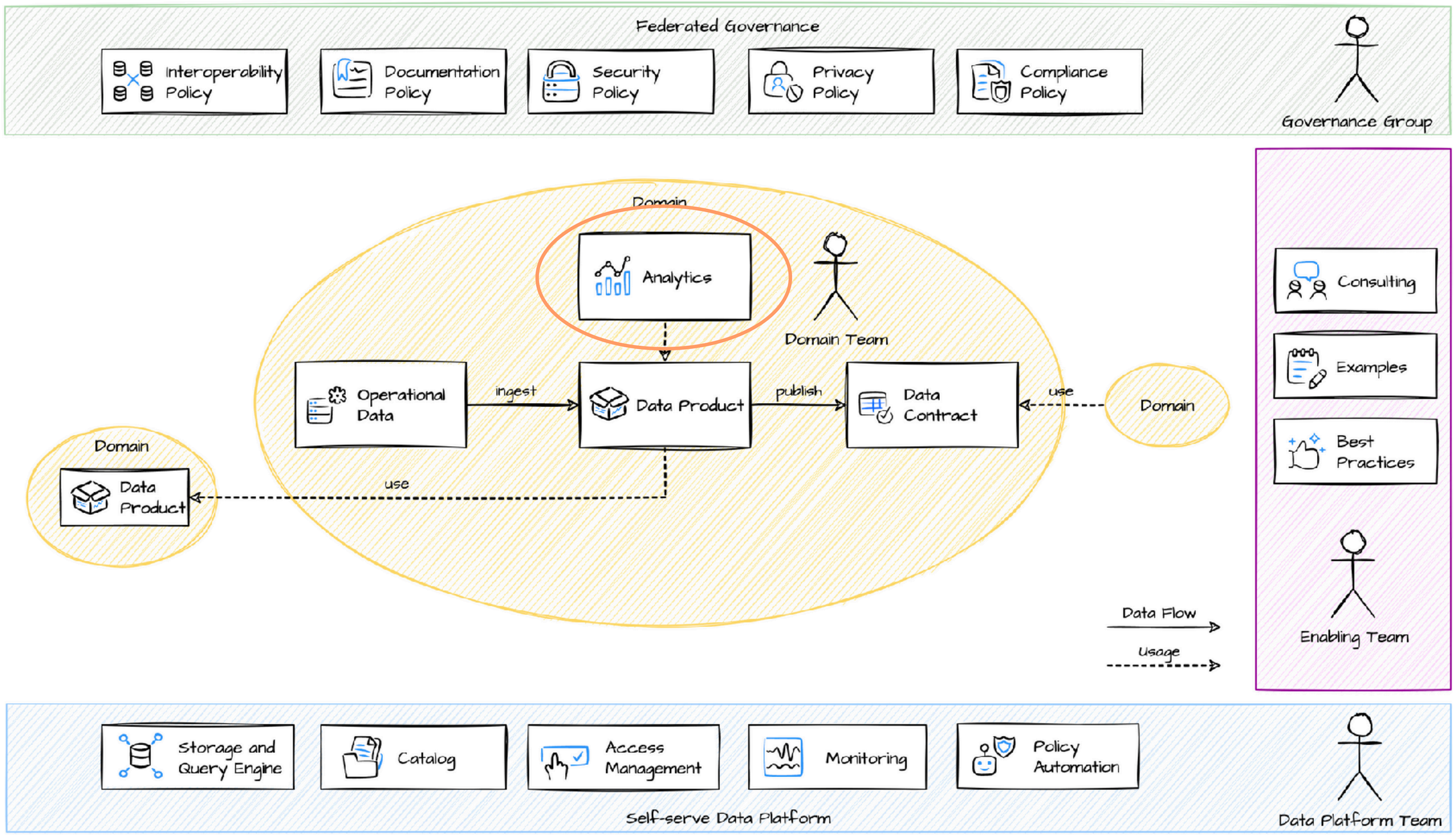
Transformation: SQL

entities__inventory_history.sql

Raw

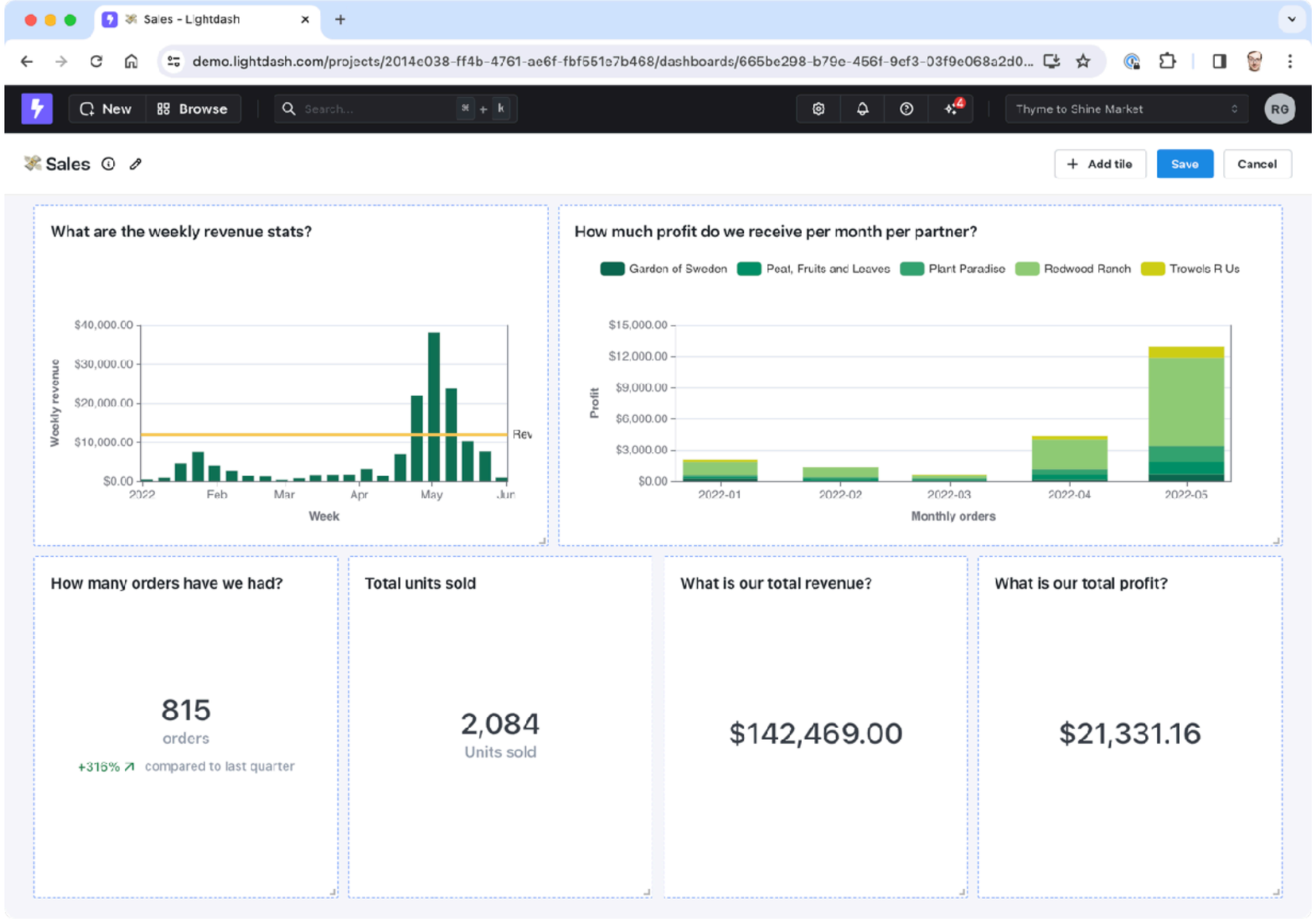
```sql
-- Step 1: Deduplicate
WITH inventory_deduplicated AS (
    SELECT *
    EXCEPT (row_number)
    FROM (
       SELECT *,
              ROW_NUMBER() OVER (PARTITION BY id ORDER BY time DESC) row_number
       FROM `datameshexample-fulfillment.raw.inventory`)
    WHERE row_number = 1
),
-- Step 2: Parse JSON to columns
inventory_parsed AS (
    SELECT
        json_value(data, "$.sku")                          AS sku,
        json_value(data, "$.location")                     AS location,
        CAST(json_value(data, "$.available") AS int64)     AS available,
        CAST(json_value(data, "$.updated_at") AS timestamp) AS updated_at,
    FROM inventory_deduplicated
)
-- Step 3: Actual Query
SELECT sku, location, available, updated_at
FROM inventory_parsed
ORDER BY sku, location, updated_at
```

Data Mesh Architecture

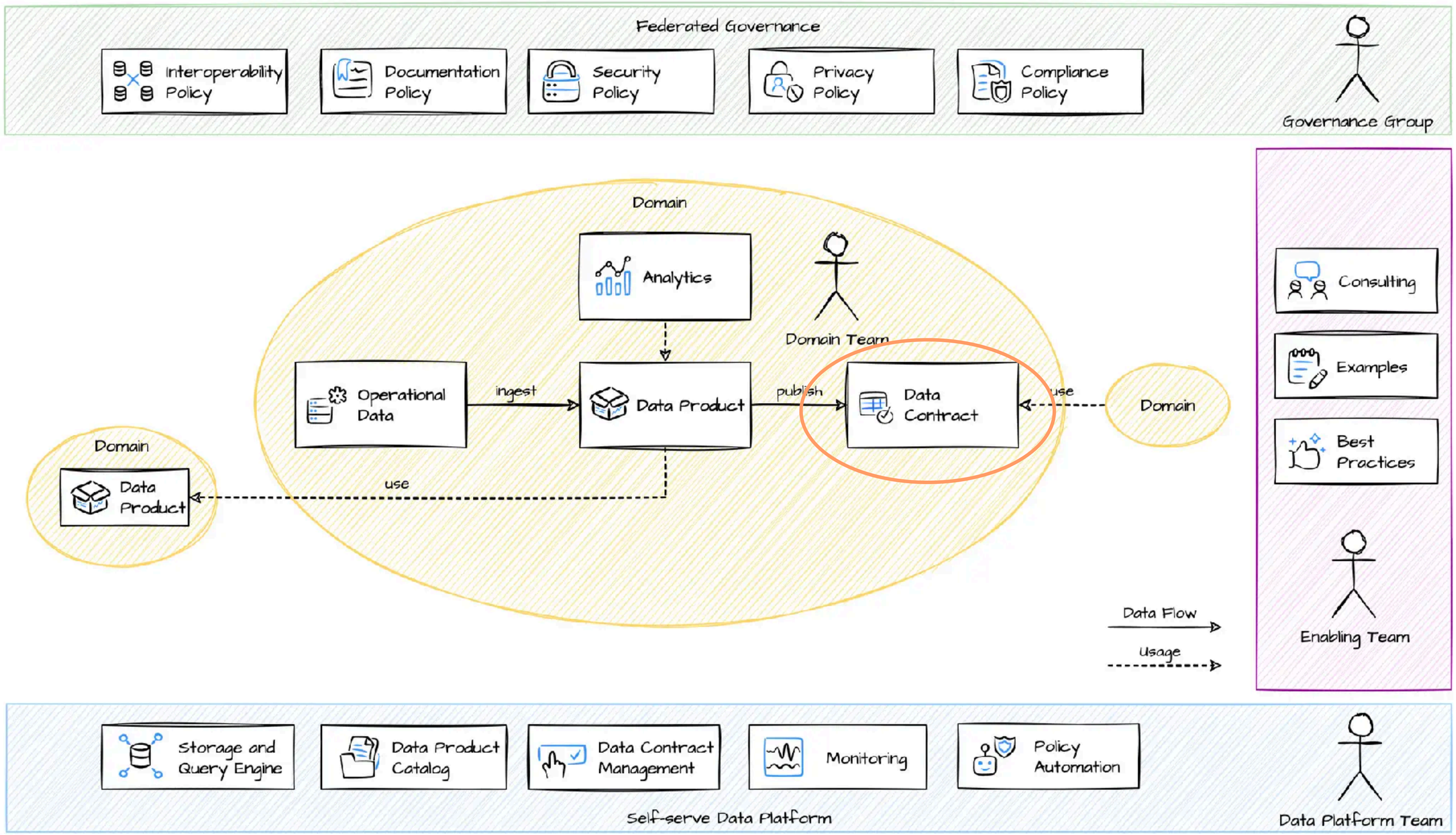


datamesh-architecture.com

Analytics: Enable Data Culture

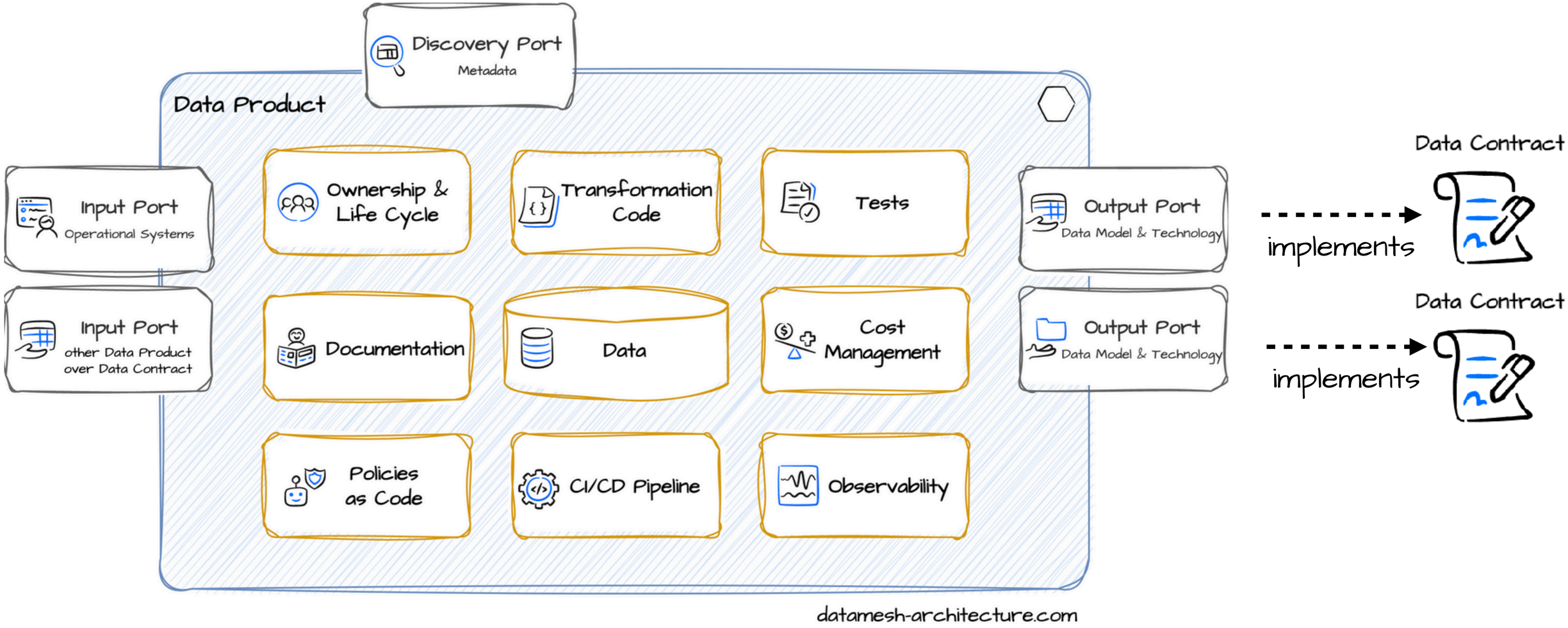


Data Mesh Architecture



datamesh-architecture.com

Data Products Data Contracts



APIs

OPENAPI INITIATIVE

Messages

AsyncAPI

Data

Data Contract Specification

Data Contract

```yaml
dataContractSpecification: 0.9.1
id: web-orders-with-consent-v1
info:
  title: Web Orders With Consent V1
  version: 1.0.0
  description: "All orders made through the web channel.\r\nFiltered for orders where customers have expressed consent for analytical use."
  owner: checkout
  contact:
    url: https://teams.example.com/datacontracts/web-orders-with-consent-v1
terms:
  usage: "The data can be used for analytical and data science use cases, as the customer has expressed their consent."
  limitations: "As the dataset is filtered, these data set cannot be used to aggregate financial KPIs.\r\nNot suited for real-time use cases."
  billing: $1000 per month
  noticePeriod: P3M
models:
  orders:
    type: table
    description: A successful sale in the web shop
    fields:
      order_id:
        type: string
        description: Primary key of the order
      billing_customer_id:
        type: string
        description: Customer ID of the billing customer
      shipment_customer_id:
        type: string
        description: Customer ID of customer to ship the order to
      sold_timestamp:
        type: timestamp_tz
        description: The timestamp of the final confirmation step in the web form.
      total_amount:
        type: bigint
        description: The total order amount in the smallest unit of the currency (such as Eurocents)
```

datacontract.com
cli.datacontract.com

- Interface Specification (like OpenAPI, but for data)
- YAML
- Define Requirements
- Make expectations explicit
- Make domain knowledge explicit
- Common language for data providers and consumers
- Owned by a team
- Contract-first
- Enforce Contract in CI/CD

```yaml
dataContractSpecification: 0.9.1
id: web-orders-with-consent-v1
info:
  title: Web Orders With Consent V1
  version: 1.0.0
  description: "All orders made through the web channel.\r\nFiltered for orders where customers have expressed consent for analytical use."
  owner: checkout
  contact:
    url: https://teams.example.com/datacontracts/web-orders-with-consent-v1
terms:
  usage: "The data can be used for analytical and data science use cases, as the customer has expressed their consent."
  limitations: "As the dataset is filtered, these data set cannot be used to aggregate financial KPIs.\r\nNot suited for real-time use cases."
  billing: $1000 per month
  noticePeriod: P3M
models:
  orders:
    type: table
    description: A successful sale in the web shop
    fields:
      order_id:
        type: string
        description: Primary key of the order
      billing_customer_id:
        type: string
        description: Customer ID of the billing customer
      shipment_customer_id:
        type: string
        description: Customer ID of customer to ship the order to
      sold_timestamp:
        type: timestamp_tz
        description: The timestamp of the final confirmation step in the web form.
      total_amount:
```

Contract Enforcement

```yaml
order_total:
  description: Total amount the smallest monetary unit (e.g., cents)
  type: long
  required: true
customer_id:
  description: Unique identifier for the customer.
  type: text
  minLength: 10
  maxLength: 20
customer_email_address:
  description: The email address, as entered by the customer. The ema
  type: text
  format: email
  required: true
```

```yaml
quality:
  type: SodaCL
  specification:
    checks for orders:
      - freshness(order_timestamp) < 24h
      - row_count > 500000
      - duplicate_count(order_id) = 0
    checks for line_items:
      - row_count > 500000
```

Data Contract CLI

```yaml
dataContractSpecification: 0.9.3
id: urn:datacontract:orders-latest
info:
  title: Orders Latest
  version: 1.0.0
models:
  orders:
    type: table
    fields:
      order_id:
        type: text
        format: uuid
```

datacontract.yaml

test

BigQuery | Postgres | databricks | snowflake | Kafka | AWS S3

github.com/datacontract/cli

datacontract test

```
$ datacontract test datacontract.yaml
```

datacontract test

jochen — -zsh

```
jochen@Jochens-MacBook-Pro-2 ~ % datacontract test https://datacontract.com/examples/orders-latest/datacontract.yaml
Testing https://datacontract.com/examples/orders-latest/datacontract.yaml
```

Result	Check	Field	Details
passed	Check that JSON has valid schema	orders	All JSON entries are valid.
passed	Check that JSON has valid schema	line_items	All JSON entries are valid.
passed	Check that field order_id is present	orders	
passed	Check that field order_timestamp is present	orders	
passed	Check that field order_total is present	orders	
passed	Check that field customer_id is present	orders	
passed	Check that field customer_email_address is present	orders	
passed	Check that field processed_timestamp is present	orders	
passed	row_count >= 5	orders	
passed	Check that required field order_id has no null values	orders.order_id	
passed	Check that unique field order_id has no duplicate values	orders.order_id	
passed	duplicate_count(order_id) = 0	orders.order_id	
passed	Check that required field order_timestamp has no null values	orders.order_timestamp	
passed	Check that required field order_total has no null values	orders.order_total	
passed	Check that required field customer_email_address has no null values	orders.customer_email_address	
passed	Check that required field processed_timestamp has no null values	orders.processed_timestamp	
passed	Check that field lines_item_id is present	line_items	
passed	Check that field order_id is present	line_items	
passed	Check that field sku is present	line_items	
passed	values in (order_id) must exist in orders (order_id)	line_items.order_id	
passed	row_count >= 5	line_items	
passed	Check that required field lines_item_id has no null values	line_items.lines_item_id	
passed	Check that unique field lines_item_id has no duplicate values	line_items.lines_item_id	

```
🟢 data contract is valid. Run 23 checks. Took 6.776398 seconds.
jochen@Jochens-MacBook-Pro-2 ~ %
```

Discover: Real User Interactio | Change column name · datacc

github.com/datacontract/cli-examples/actions/runs/6423146219/job/17441032658?pr=2

Code | Issues | Pull requests 1 | Actions | Projects | Wiki | Security | Insights | Settings

← Back to pull request #2

Change column name #11

Re-run jobs

Summary

Jobs

checkBreakingChanges

Run details

Usage

Workflow file

checkBreakingChanges

failed 5 days ago in 7s

Search logs

Set up job 1s

Run actions/checkout@v4 1s

Get CLI 0s

Check backwards compatibility 0s

```
1  ▶ Run ./datacontract breaking --with https://raw.githubusercontent.com/datacontract/cli-examples/main/datacontract.yaml
4  Found 1 differences between the data contracts!
5
6  🔴 Difference 1:
7  Description:  field 'my_table.my_column' was removed
8  Type:         field-removed
9  Severity:     breaking
10 Level:        field
11 Model:        my_table
12 Field:        my_column
13 Exiting application with error: found breaking differences between the data contracts
14 Error: Process completed with exit code 1.
```

Data Contract CLI

import

```
dataContractSpecification: 0.9.3
id: urn:datacontract:orders-latest
info:
  title: Orders Latest
  version: 1.0.0
models:
  orders:
    type: table
    fields:
      order_id:
        type: text
        format: uuid
```

datacontract.yaml

diff

export

test

github.com/datacontract/cli

Data Marketplace

Data Mesh Manager | Data Products | Data Contracts | Policies | ACME

Data Contracts

Data Contracts

Definitions | + Add Data Contract

Search | Owner | Data Product | Sort

Customer Cohorts
Marketing | Customer Cohorts 1
A table with customer cohorts and their properties

s3_customers_history_npii_v1
Checkout | Customers
All customer states, updated on every modifying event. PII removed.

s3_customers_history_pii_v1
Checkout | Customers
All customer states, updated on every modifying event. PII included.

search_queries_all_v1
Search | Search Queries All 1
All search queries and result sets with PII removed.

snowflake_articles_history
Products | Articles history 2
All article snapshots since 2020

snowflake_articles_latest
Products | Articles latest 1
Current state of all articles

snowflake_customers_latest_npii_v1
Checkout | Customers
All customers in their latest state, PII removed.

snowflake_customers_latest_pii_v1
Checkout | Customers 1
All customers in their latest state, PII included.

snowflake_fulfillment_shelf_warmers
Fulfillment | Shelf Warmers
A list of articles with no sales in last 6 months

snowflake_fulfillment_stock_update_events
Fulfillment | Stock Update Events 1
All stock updates since 2020

snowflake_orders_npii_v2
Checkout | Orders 1
All order-created events, PII removed.

snowflake_orders_pii_v2
Checkout | Orders 1
All order-created events, with PII.

Request Access

You are requesting access to the data product Articles history on output port snowflake_articles_history .
The system will create a data usage agreement for the team Products to approve.

Consumer Required

Article Profitability Analysis (Team Controlling)

Select your data product that wants to access and use the provided data.

Purpose Required

Profitability analysis.

Why do you want access and what do you want to do with the data?

Cancel | Customize | Request access

Data Mesh Manager

Data Products | Data Contracts | Policies

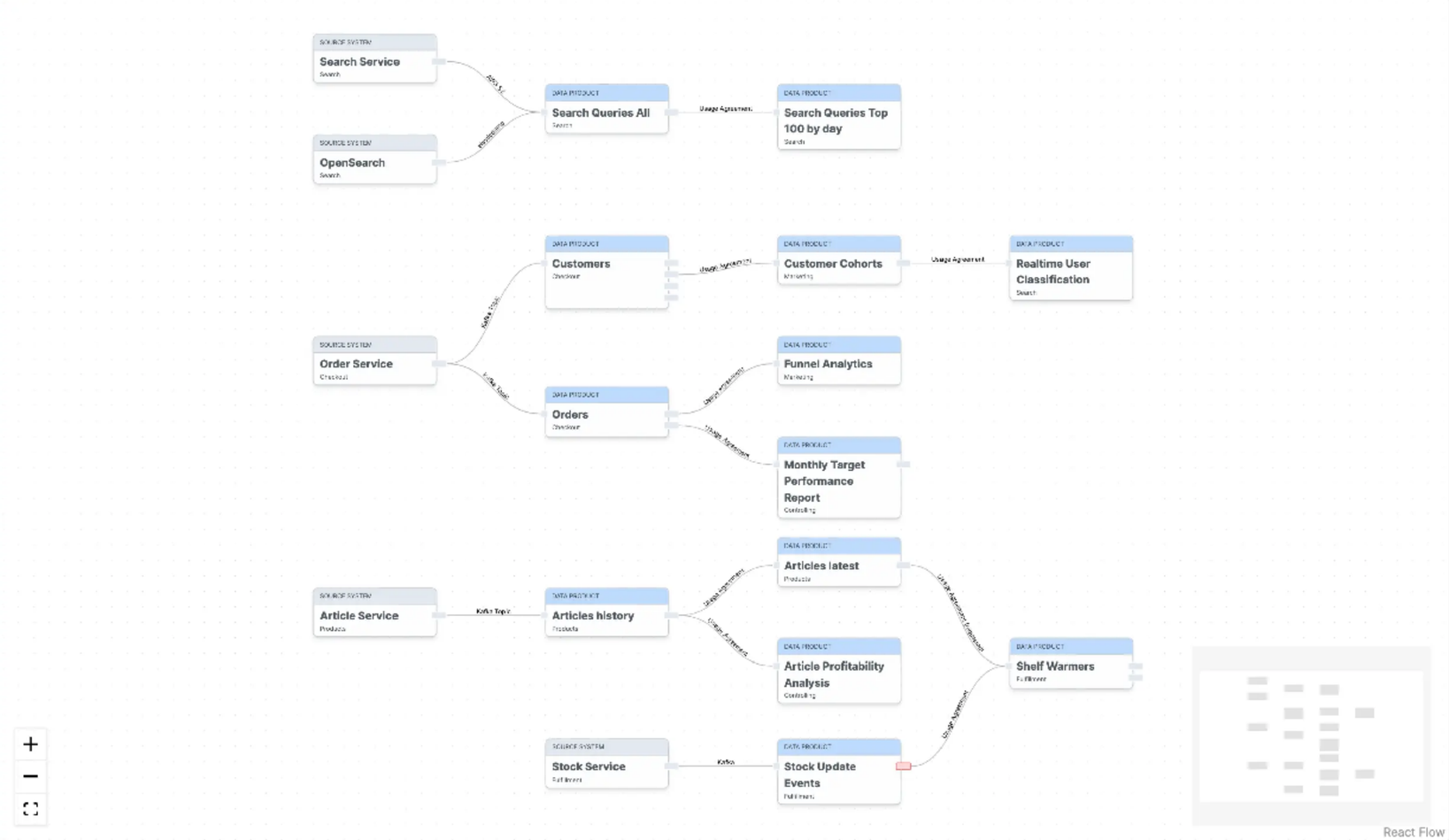
ACME

Data Map

Data Map

Data Products | Source Systems | Domain Teams | + Add Data Product

Search | Owner | Status



Decentralized Data Architecture

Start using your own data

Make qualified data-driven decisions in your domain

Use data to better understand your users and system behavior. Derive features from insights, qualify value, and fast iterations. Also qualified rejection of unnecessary tasks.

Do the right things, purpose, motivation

Build innovative services in your domain

Enhance your customer experience with data technologies, such as LLMs, visualizations, classifications, and ML models for predictions and recommendations.

Customer value through innovation

Decentralized Data Architecture

And share when there is demand

Make qualified data-driven decisions in your domain

Use data to better understand your users and system behavior. Derive features from insights, qualify value, and fast iterations. Also qualified rejection of unnecessary tasks.

Do the right things, purpose, motivation

Build innovative services in your domain

Enhance your customer experience with data technologies, such as LLMs, visualizations, classifications, and ML models for predictions and recommendations.

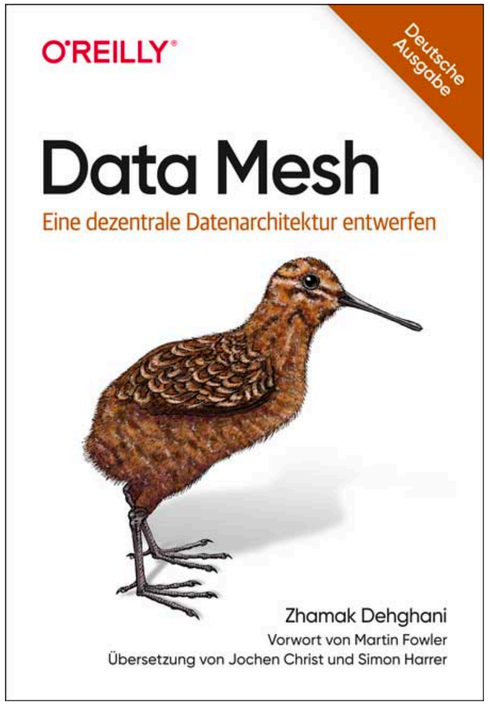
Customer value through innovation

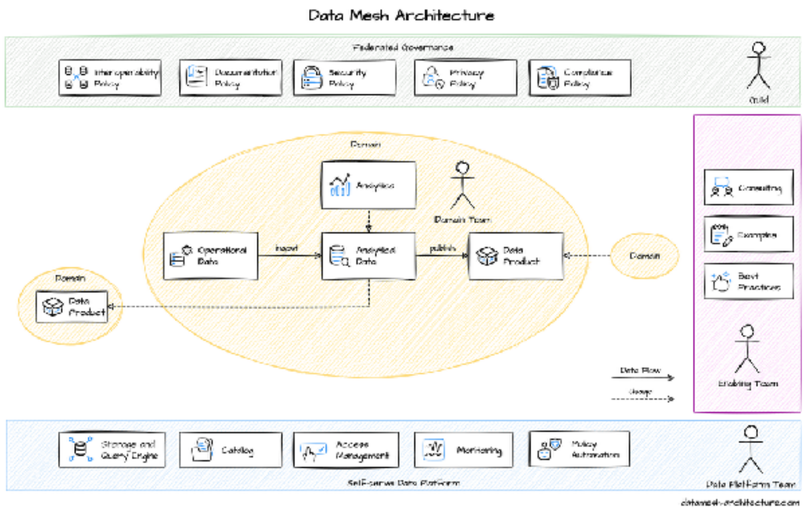
Provide data as business value for other domains

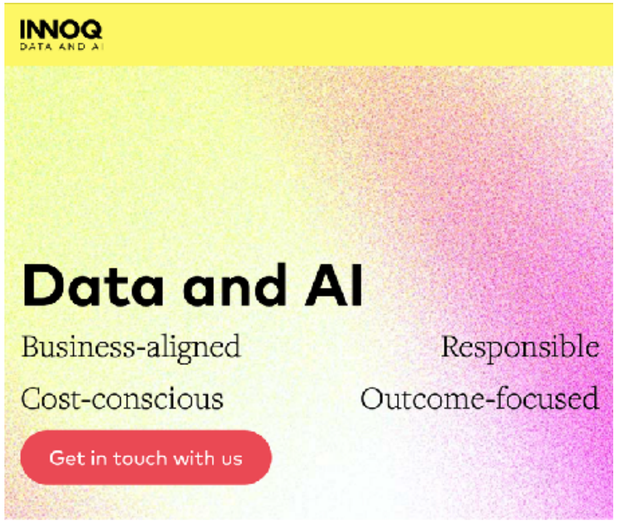
Domain data is valuable for other business units as reference data and to aggregate. Needs managed, explained, high-quality and easy accessible data as products.

Company success

Learn more







oreilly.de/produkt/data-mesh

datamesh-architecture.com

INNOQ.ai
Data Mesh Consulting, Trainings,
Data Product Engineering

Data Mesh

Introduction

JOCHEN CHRIST
@JOCHEN_CHRIST