

Data Mesh

(für Entwickler:innen)



JOCHEN CHRIST
@JOCHEN_CHRIST

Hi,
I am Jochen

Jochen Christ

Tech Lead at INNOQ



Java

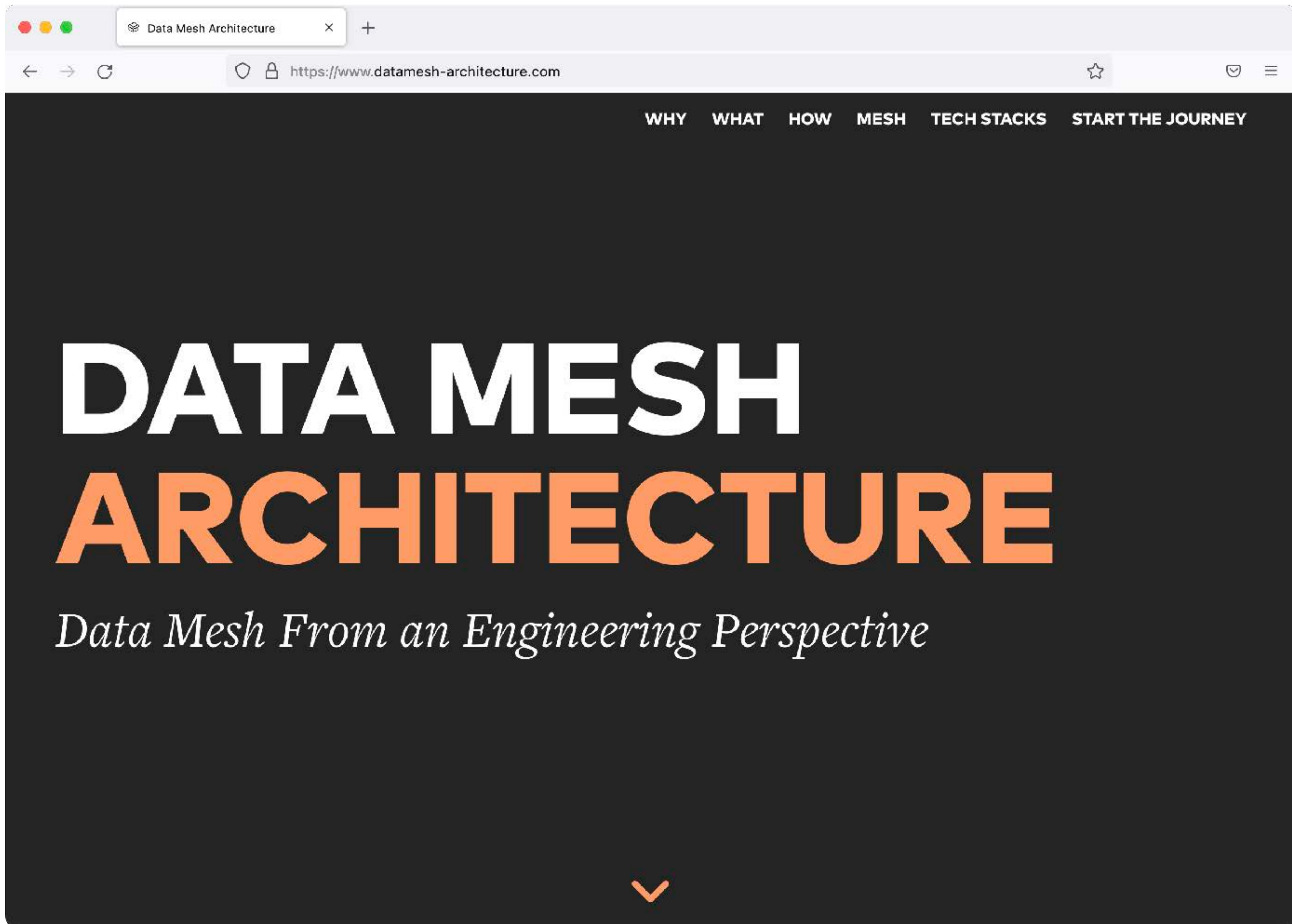


Self-contained Systems



Remote Mob Programming

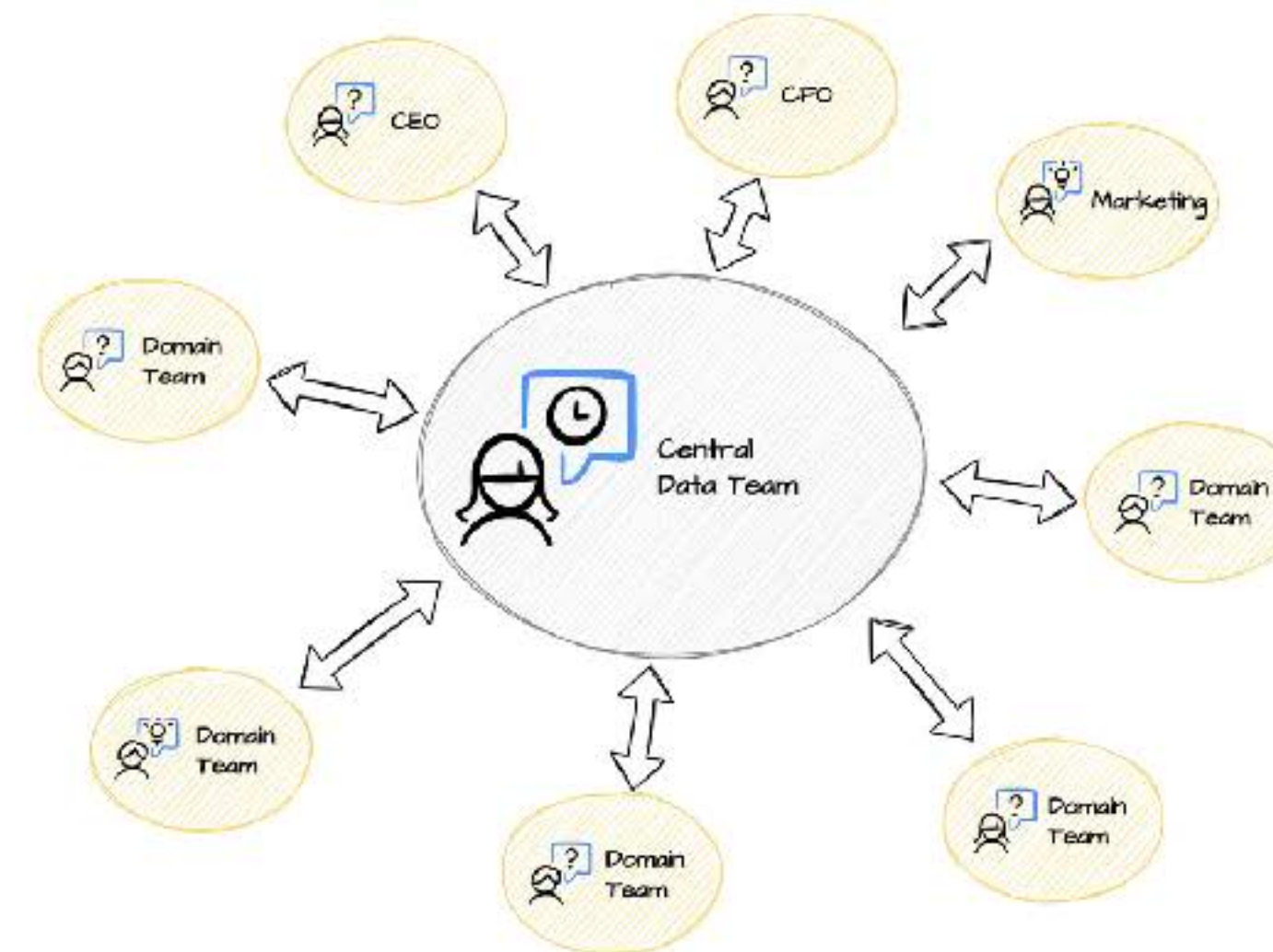




Why You May Need a Data Mesh

Many organizations have invested in a central data lake and a data team with the expectation to drive their business based on data. However, after a few initial quick wins, they notice that **the central data team often becomes a bottleneck**. The team cannot handle all the analytical questions of management and product owners quickly enough. This is a massive problem because making timely data-driven decisions is crucial to stay competitive. For example: Is it a good idea to offer free shipping during Black Week? Do customers accept longer but more reliable shipping times? How does a product page change influence the checkout and returns rate?

The data team wants to answer all those questions quickly. In practice, however, they struggle because they need to spend too much time fixing broken data pipelines after operational database changes. In their little time remaining, **the data team has to discover and understand the necessary domain data**. For every question, they need to learn domain knowledge to give meaningful insights. Getting the required domain expertise is a daunting task.

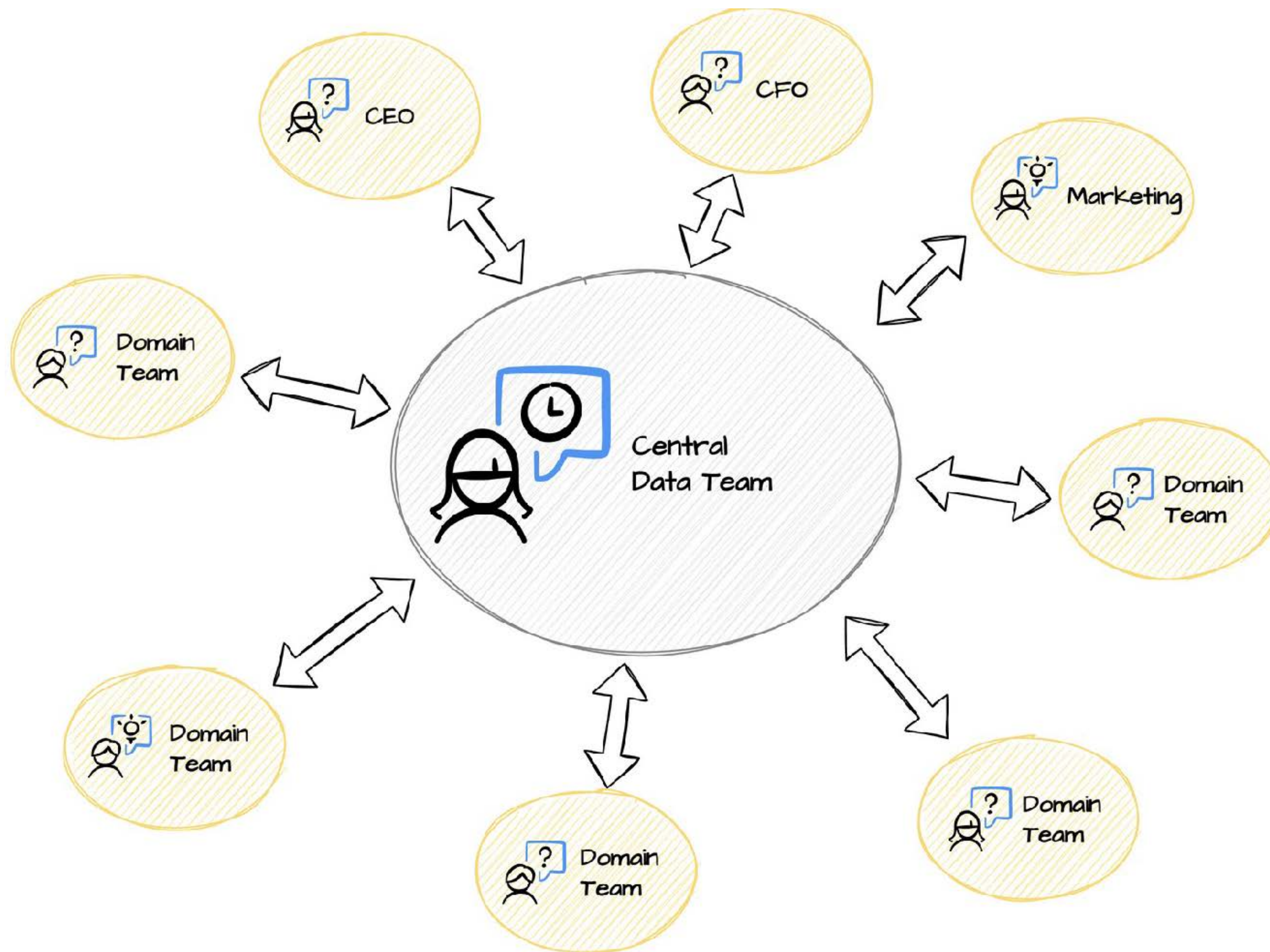


On the other hand, organizations have also invested in domain-driven design, autonomous domain teams (also known as stream-aligned teams or product teams) and a decentralized microservice architecture. These **domain teams own and know their domain**, including the information needs of the business. They design, build, and run their web applications and APIs on their own. Despite knowing the domain and the relevant information needs, the domain teams have to reach out to the overloaded central data team to get the necessary data-driven insights.

- ☒ Decentralize business into domains
 - ☒ Decentralize engineering into autonomous teams
 - ☒ Decentralize models
- Scale up software development

PROMISE

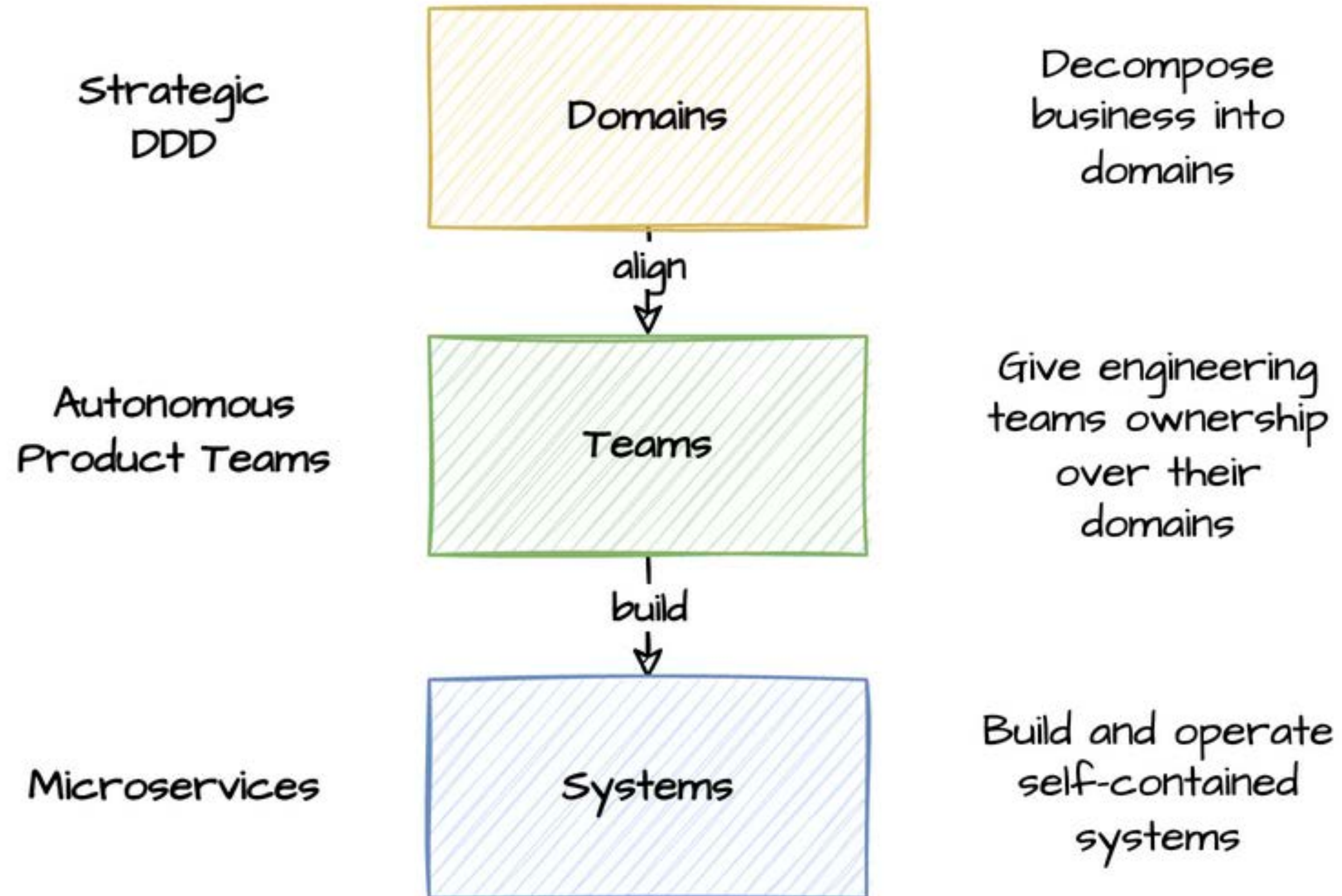
**DRIVE BUSINESS
BASED ON DATA**



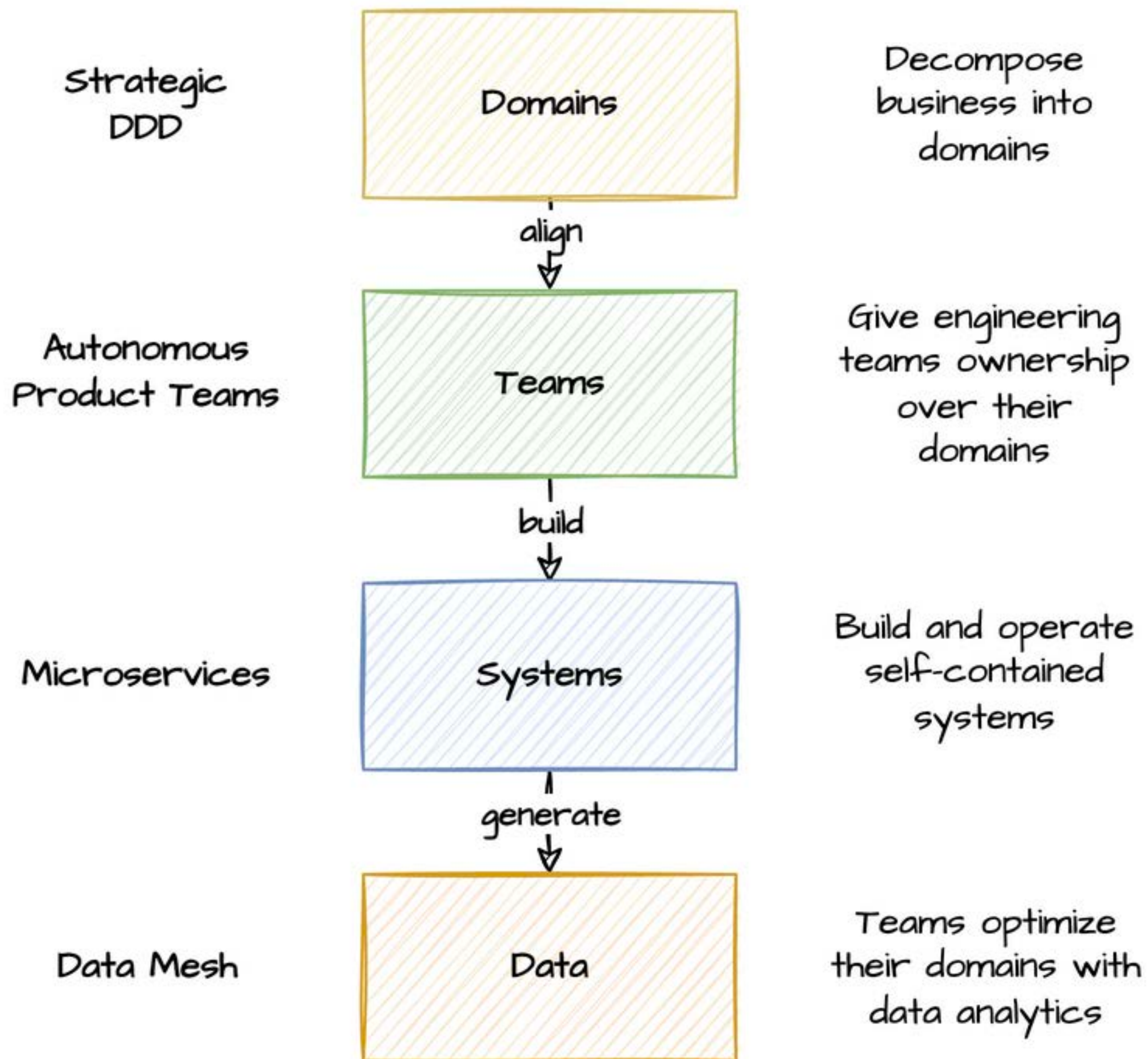
BUSINESS INTELLIGENCE FAILED TO DELIVER

...because BI failed to scale.

Decentralize to scale



Decentralize to scale



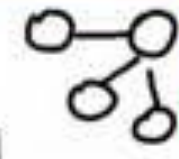
**DATA MESH
ENABLES YOU
TO PERFORM CROSS-DOMAIN
DATA ANALYSIS
ON YOUR OWN**

What Is Data Mesh?

Strategic
Domain-driven
Design

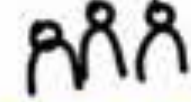
Domain
Ownership

Domain
Bounded Context



Socio-technical
Perspective

Domain Teams



Technology

Operational &
Analytical Data

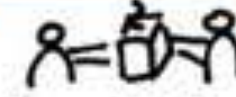


Data as a
Product

Product Thinking



Data Product by
Domain Team



Interoperability
Interfaces



Self-serve
Data Platform

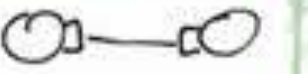
Domain-agnostic

Data Platform
Team

Self-serve
Data Platform

Federated
Governance

Context Mapping



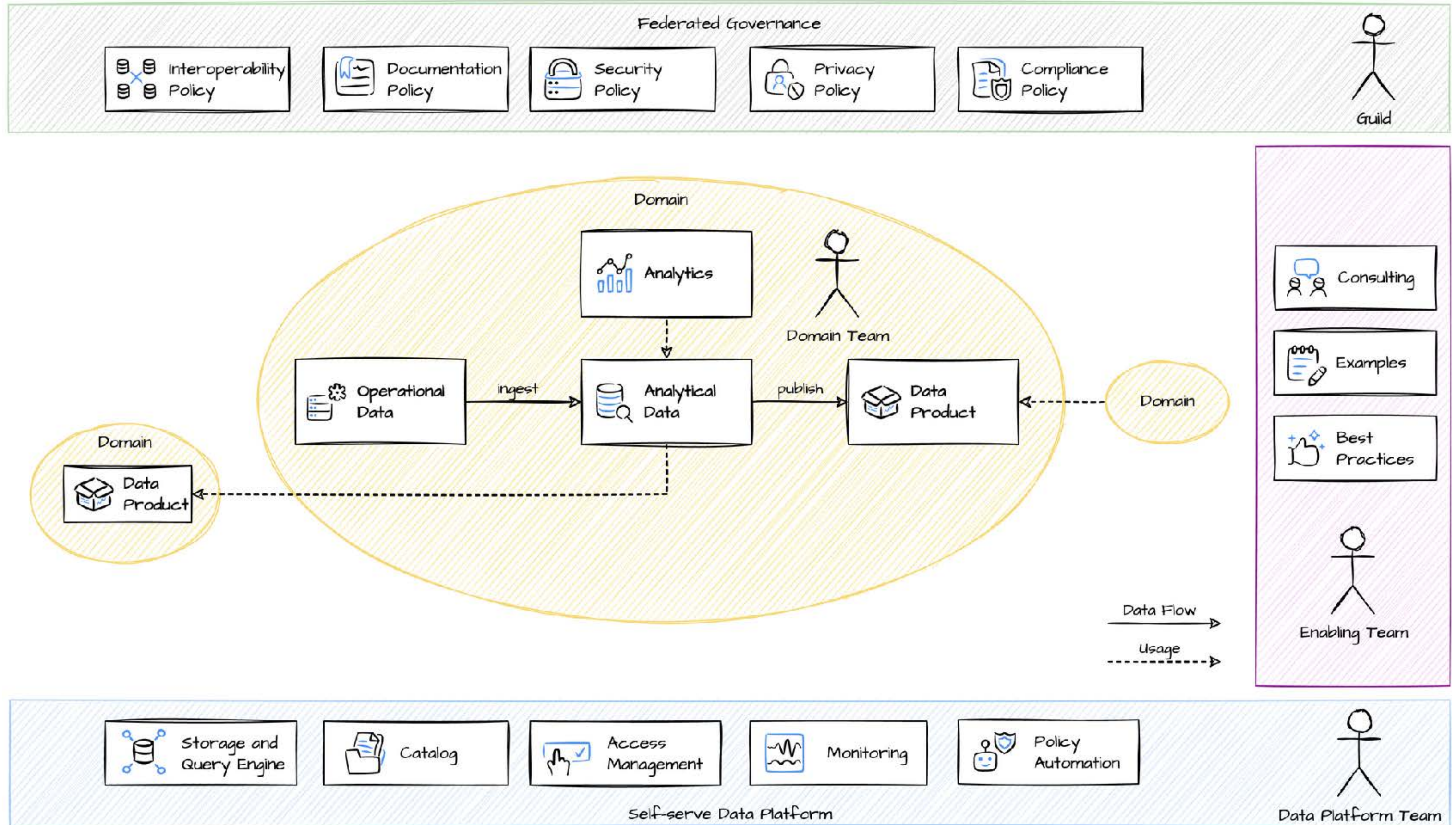
Guild

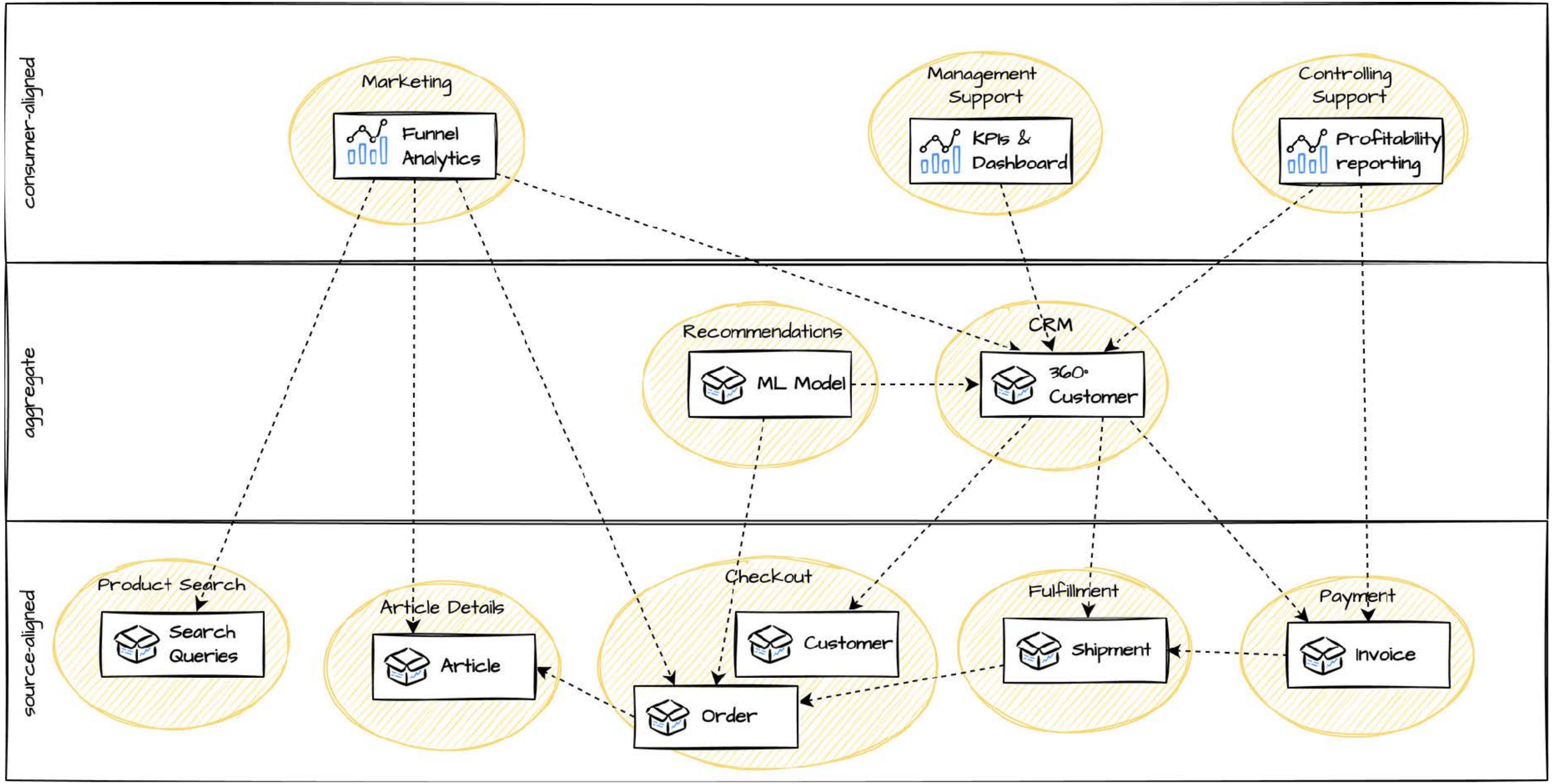


Data Governance



Data Mesh Architecture





Data Product

Data

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

Metadata

Tags (1)

Data Product	
Display name	Value
Domain Team	Fulfillment
Update Frequency	Hourly
Environment	PRODUCTION

Schema

 Filter

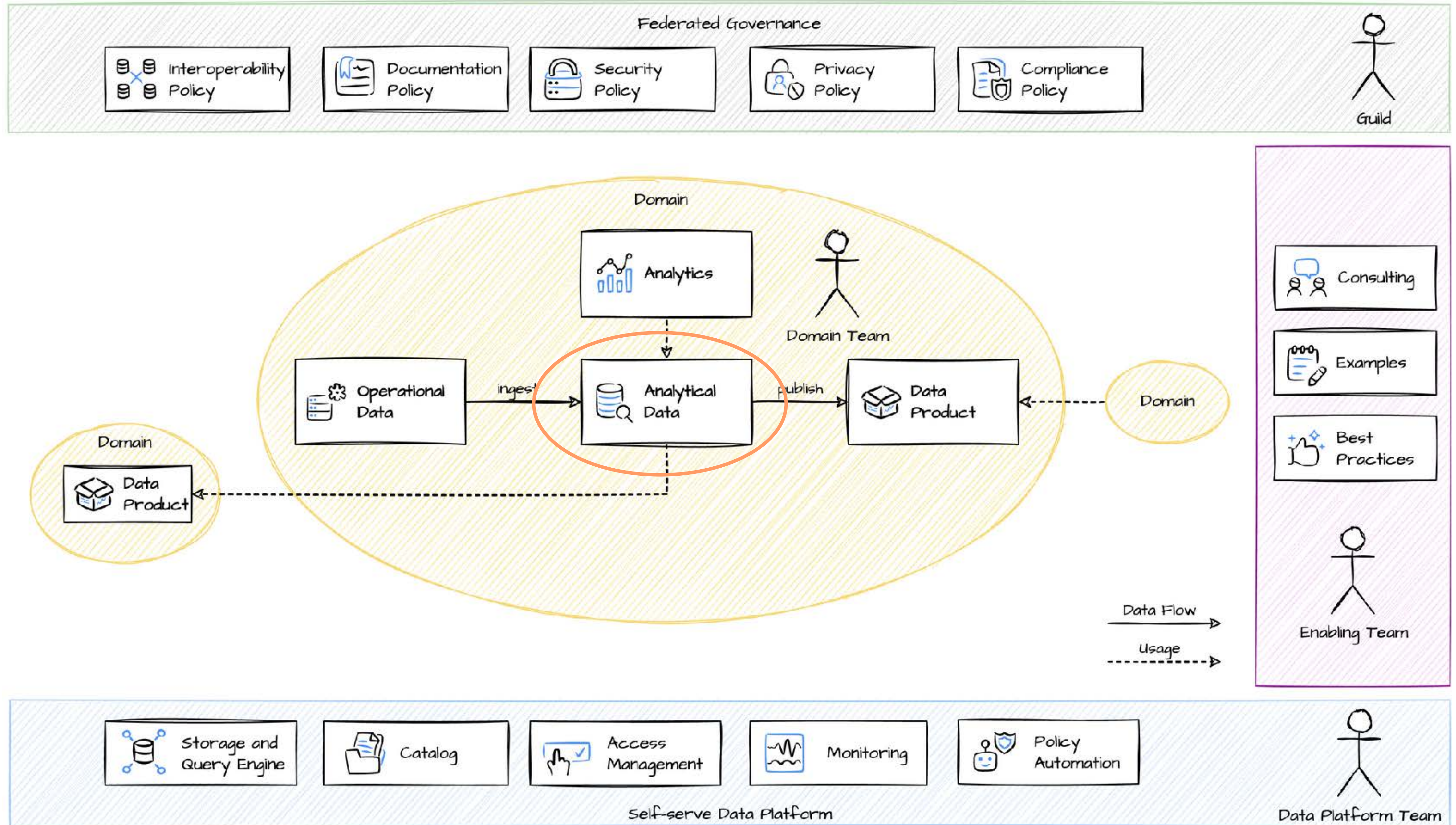
Enter property name or value

Field name	Type	Mode
sku	STRING	NULLABLE
location	STRING	NULLABLE
available	INT64	NULLABLE
updated_at	TIMESTAMP	NULLABLE

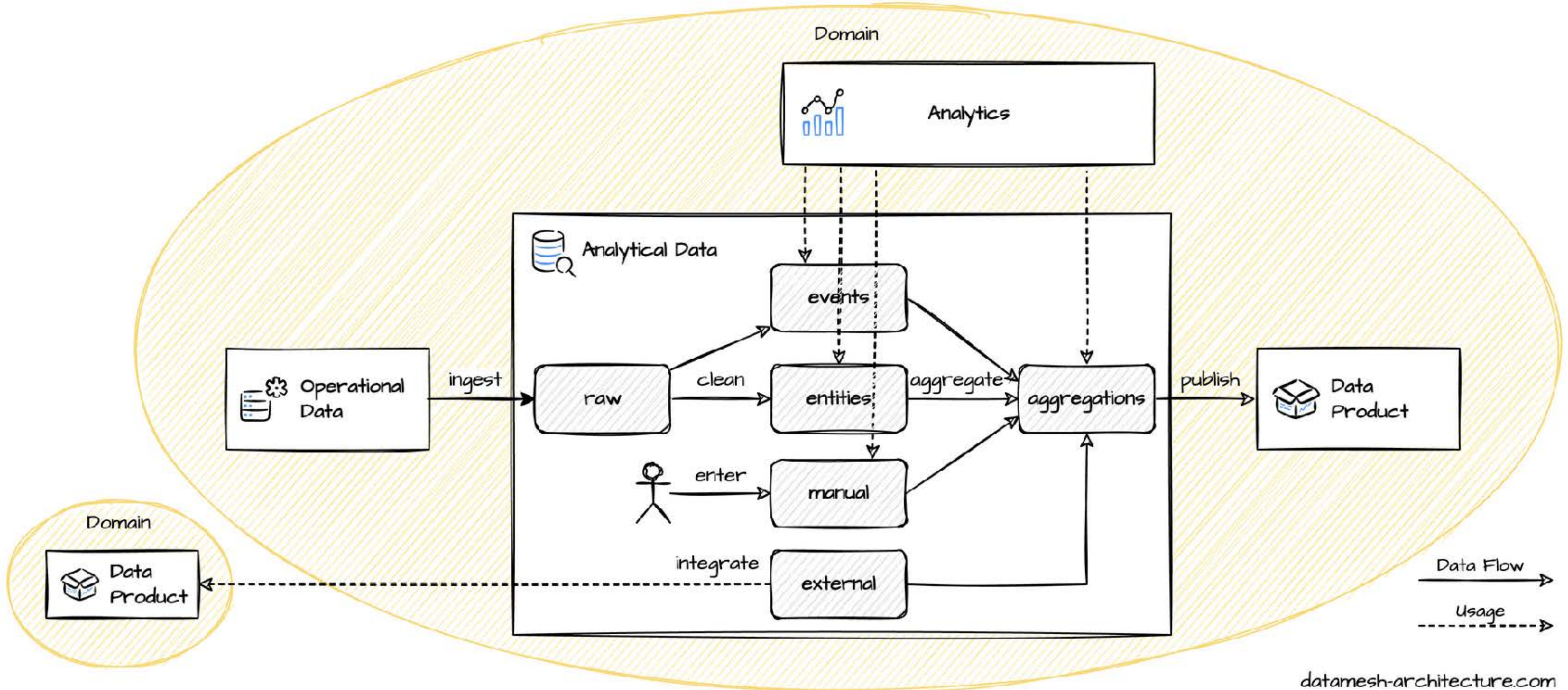
Operations

132M	0	5 min ago	42	\$1.4k
Rows	Duplicates	Freshness	24h Usages	Monthly Costs

Data Mesh Architecture



Analytical Data



```
1  -- Step 1: Deduplicate
2  WITH inventory_deduplicated AS (
3      SELECT *
4      EXCEPT (row_number)
5      FROM (
6          SELECT *,
7              ROW_NUMBER() OVER (PARTITION BY id ORDER BY time DESC) row_number
8          FROM `datameshexample-fulfillment.raw.inventory`)
9      WHERE row_number = 1
10 ),
11 -- Step 2: Parse JSON to columns
12 inventory_parsed AS (
13     SELECT
14         json_value(data, "$.sku") AS sku,
15         json_value(data, "$.location") AS location,
16         CAST(json_value(data, "$.available") AS int64) AS available,
17         CAST(json_value(data, "$.updated_at") AS timestamp) AS updated_at,
18     FROM inventory_deduplicated
19 )
20 -- Step 3: Actual Query
21 SELECT sku, location, available, updated_at
22 FROM inventory_parsed
23 ORDER BY sku, location, updated_at
```



dbtTM

5

EXPLORER

...

FULFILLMENT_DBT

> analyses

> dbt_packages

> logs

> macros

> models

> analytics

> data_products

> staging / inventory

! src_inventory.yml

stg_inventory__inven...

! stg_inventory.yml

> seeds

> snapshots

> target

> tests

inventory_history_test....

.gitignore

! dbt_project.yml

i README.md

> OUTLINE

> TIMELINE

stg_inventory__inventory_history.sql

inventory_history.sql

inventory_latest.sql

inventory__avai

...

models > staging > inventory > stg_inventory__inventory_history.sql

1 with source as (
2 | select * from {{ source('fulfillment_sources', 'inventory') }}
3 |),
4
5 parsed as (
6 | select
7 | record_metadata:"CreateTime"::number as metadata_createtime,
8 | record_metadata:"headers"."ce-id"::string as event_id,
9 | record_metadata:"headers"."ce-source"::string as event_source,
10 | record_metadata:"headers"."ce-subject"::string as event_subject,
11 | record_metadata:"headers"."ce-time"::timestamp as event_time,
12 | record_metadata:"headers"."ce-type"::string as event_type,
13 | record_metadata:"key"::string as kafka_key,
14 | record_metadata:"offset"::number as kafka_offset,
15 | record_metadata:"partition"::number as kafka_partition,
16 | record_metadata:"topic"::string as kafka_topic,
17 | record_content:available::number as available,
18 | record_content:location::string as location,

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

> zsh + ▾ ▢ 🗑 ⬆ ✕

jochen@Jochens-MacBook-Pro-2 fulfillment_dbt % dbt run

0 0 AWS

Ln 31, Col 3 Spaces: 4 UTF-8 LF SQL

🗨 🔔

5

EXPLORER

...

FULFILLMENT_DBT

> analyses

> dbt_packages

> logs

> macros

> models

> analytics

> data_products

> staging/inventory

! src_inventory.yml

stg_inventory__inven...

! stg_inventory.yml

> seeds

> snapshots

> target

> tests

inventory_history_test....

.gitignore

! dbt_project.yml

i README.md

> OUTLINE

> TIMELINE

stg_inventory__inventory_history.sql

inventory_history.sql

inventory_latest.sql

inventory__avai

models > staging > inventory > stg_inventory__inventory_history.sql

23

24 deduplicated as (
25 SELECT *
26 FROM (
27 SELECT *,
28 ROW_NUMBER() OVER (PARTITION BY event_id ORDER BY event_time DESC) row_number
29 FROM parsed)
30 WHERE row_number = 1
31),
32 final as (
33 select
34 available,
35 location,
36 sku,
37 updated_at
38 from deduplicated
39)
40 select * from final

Terminal (^`)

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

> zsh

+

^

✕

jochen@Jochens-MacBook-Pro-2 fulfillment_dbt % dbt run

0 0

AWS

Ln 31, Col 3

Spaces: 4

UTF-8

LF

SQL



EXPLORER



! stg_inventory.yml X



FULFILLMENT_DBT

> analyses

> dbt_packages

> logs

> macros

▼ models

> analytics

> data_products

▼ staging / inventory

! src_inventory.yml

stg_inventory__inven...

! stg_inventory.yml

> seeds

> snapshots

> target

▼ tests

inventory_history_test....

.gitignore

! dbt_project.yml

i README.md



> OUTLINE

> TIMELINE

models > staging > inventory > ! stg_inventory.yml

```
1  version: 2
2
3  models:
4    - name: stg_inventory__inventory_history
5      database: FULFILLMENT_ANALYTICS
6      schema: PUBLIC
7      columns:
8        - name: location
9          tests:
10            - not_null
11        - name: sku
12          tests:
13            - not_null
14        - name: available
15          tests:
16            - not_null
17        - name: updated_at
18          tests:
19            - not_null
20
```

5

EXPLOLER

...

FULFILLME...

> analyses

> dbt_packages

> logs

> macros

> models

> seeds

> snapshots

> target

> tests

inventory_history_test....

.gitignore

dbt_project.yml

README.md

OUTLINE

TIMELINE

Untitled-7

data_products.yml

inventory.yml

inventory_history_test.sql

.gitignore

tests > inventory_history_test.sql

1 with inventory_history_entry as (
2 | select count(*) c
3 | from {{ref('inventory_history')}}
4 |)
5 select * from inventory_history_entry where c < 2000000

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

zsh

+

^

×

14:41:14 8 of 11 PASS not_null_stg_inventory__inventory_history_sku..... [PASS in 0.45s]

14:41:14 9 of 11 START test not_null_stg_inventory__inventory_history_updated_at..... [RUN]

14:41:20 9 of 11 PASS not_null_stg_inventory__inventory_history_updated_at..... [PASS in 6.23s]

14:41:20 10 of 11 START test source_not_null_fulfillment_sources_inventory_RECORD_CONTENT [RUN]

14:41:21 10 of 11 PASS source_not_null_fulfillment_sources_inventory_RECORD_CONTENT..... [PASS in 0.92s]

14:41:21 11 of 11 START test source_not_null_fulfillment_sources_inventory_RECORD_METADATA [RUN]

14:41:23 11 of 11 PASS source_not_null_fulfillment_sources_inventory_RECORD_METADATA..... [PASS in 1.70s]

14:41:23

14:41:23 Finished running 11 tests in 15.67s.

14:41:23 Completed successfully

14:41:23

14:41:23 Done. PASS=11 WARN=0 ERROR=0 SKIP=0 TOTAL=11

jochen@Jochens-MacBook-Pro-2 fulfillment_dbt %

0

0

AWS

Ln 5, Col 11

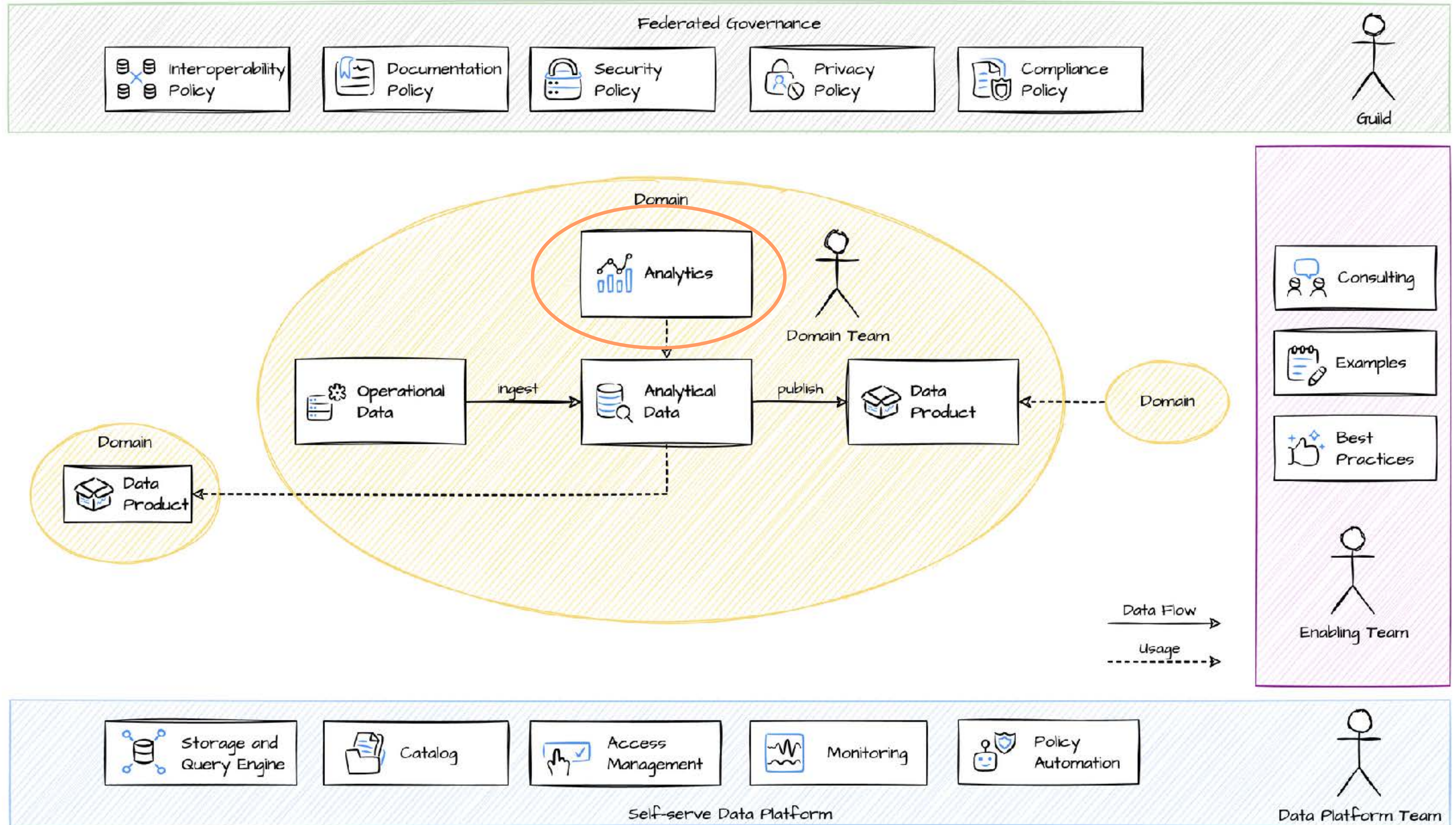
Spaces: 4

UTF-8

LF

SQL

Data Mesh Architecture



Visualize

Filter

Parameters

Actions

Themes

Settings

Community

Dataset

SPICE Business Review

100%

Fields list

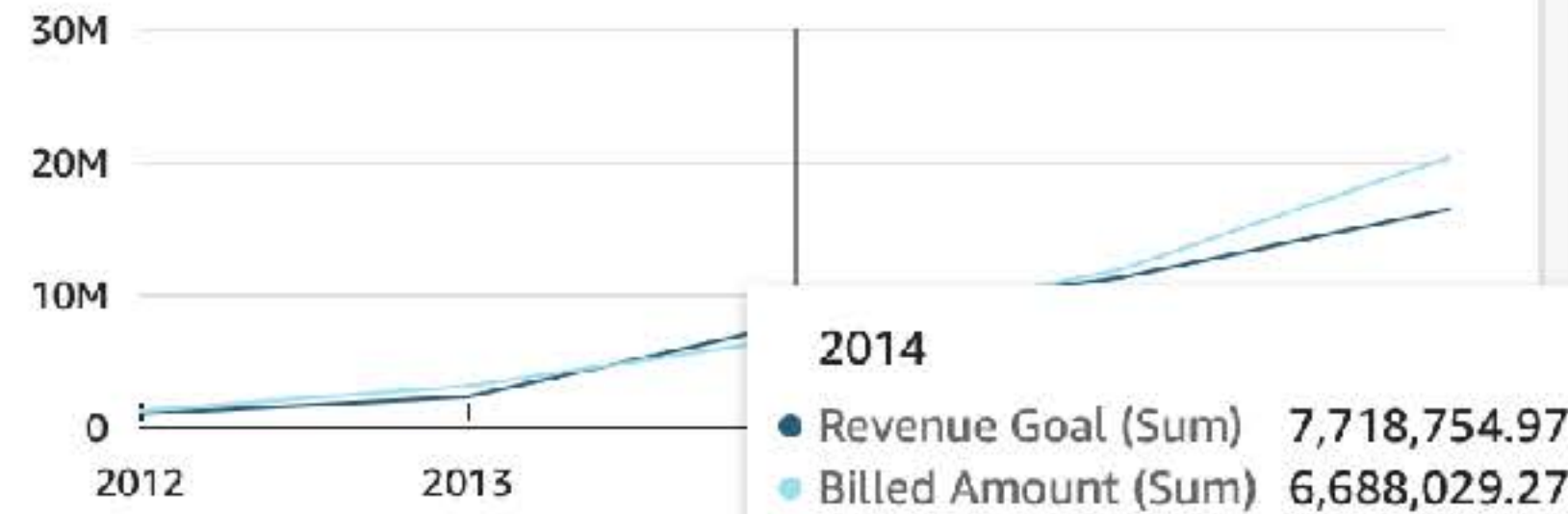
Search fields

- # Billed Amount
- Channel
- # Cost
- Customer ID
- Customer Name
- Customer Region
- Date
- # Distinct ID
- # Revenue Goal
- Segment
- Service Line

Field wells

Sheet 1

Revenues vs Goals

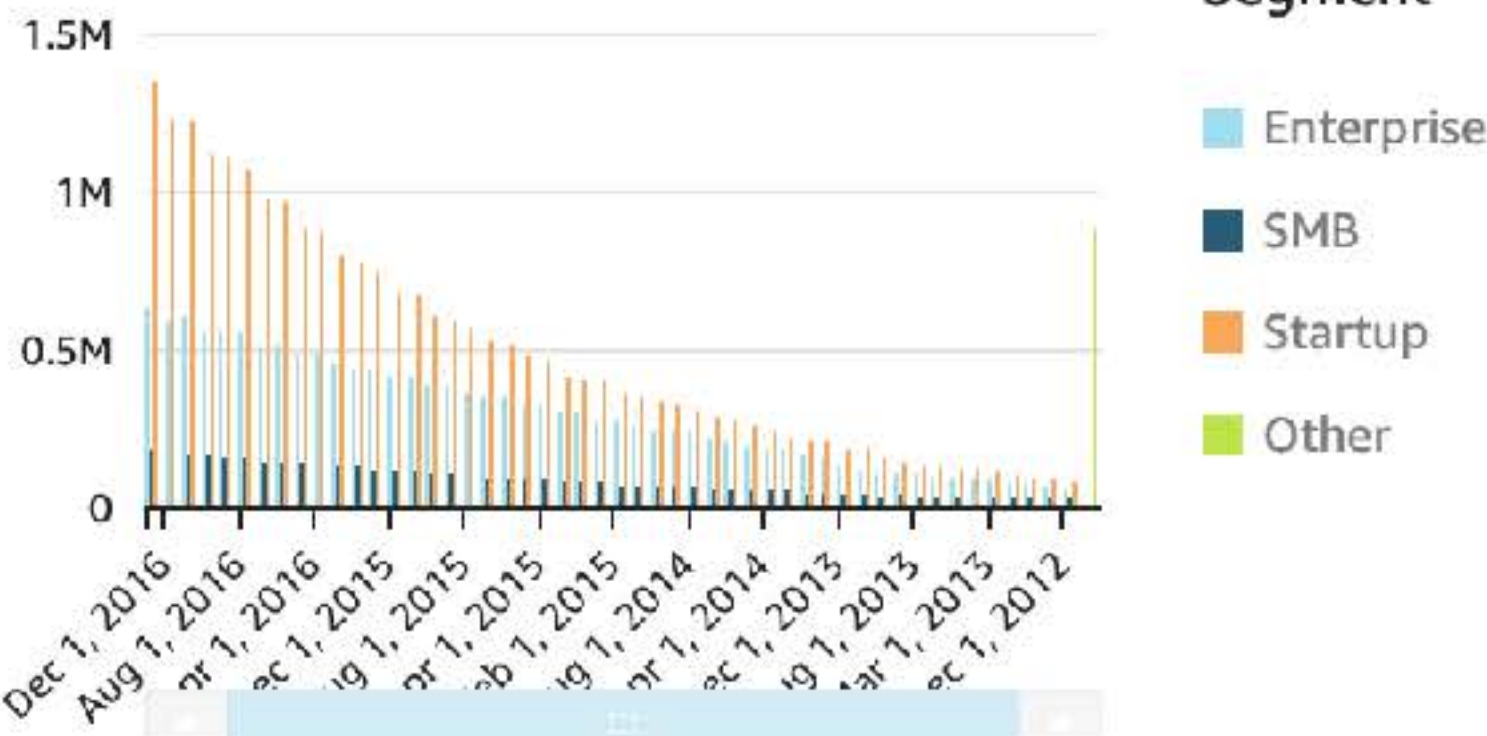


YTD Revenues by Service Line and Segment

Service Line	Billed ...	Customer ID
Billing	17,347,198.35	12,023
HR	17,221,974.97	12,076
Marketing	8,406,810.21	8,950

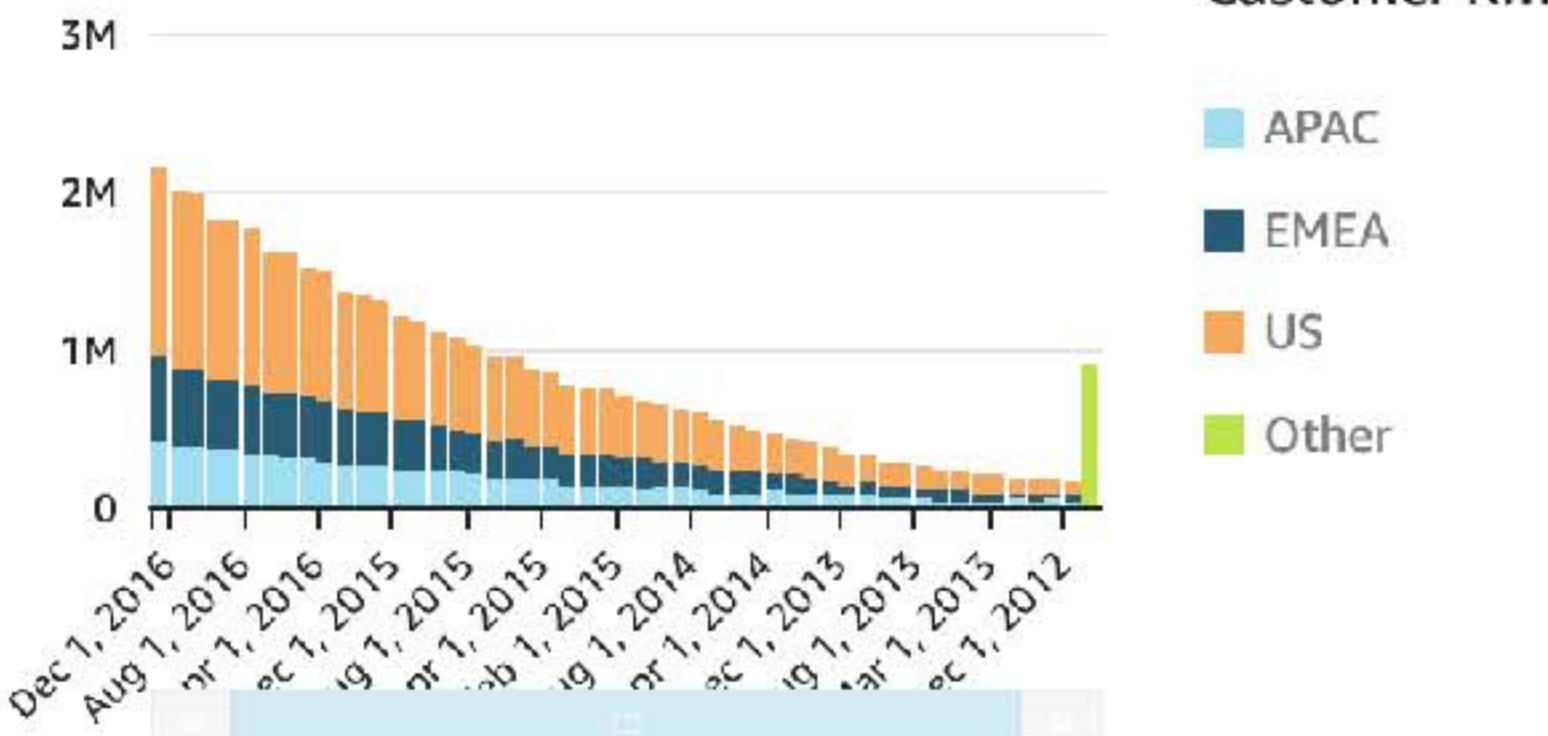
Revenues by Customer Segment

SHOWING TOP 50 IN DATE AND TOP 4 IN SEGMENT



Revenues by Region

SHOWING TOP 50 IN DATE AND TOP 4 IN CUSTOMER REGION



Channel adoption across Time

Average Revenues vs Costs by Segment



+ Code + Text

Connect

Editing



```
[ ] from google.colab import auth
    auth.authenticate_user()
    print('Authenticated')
```

Authenticated

```
[ ] %load_ext google.colab.data_table
```

How is current inventory distributed over locations?

```
%%bigquery --project datameshexample-fulfillment
SELECT location, sum(available) as available_total
FROM `aggregates.inventory_latest`
group by location
order by location
```



1 to 10 of 11 entries Filter ?

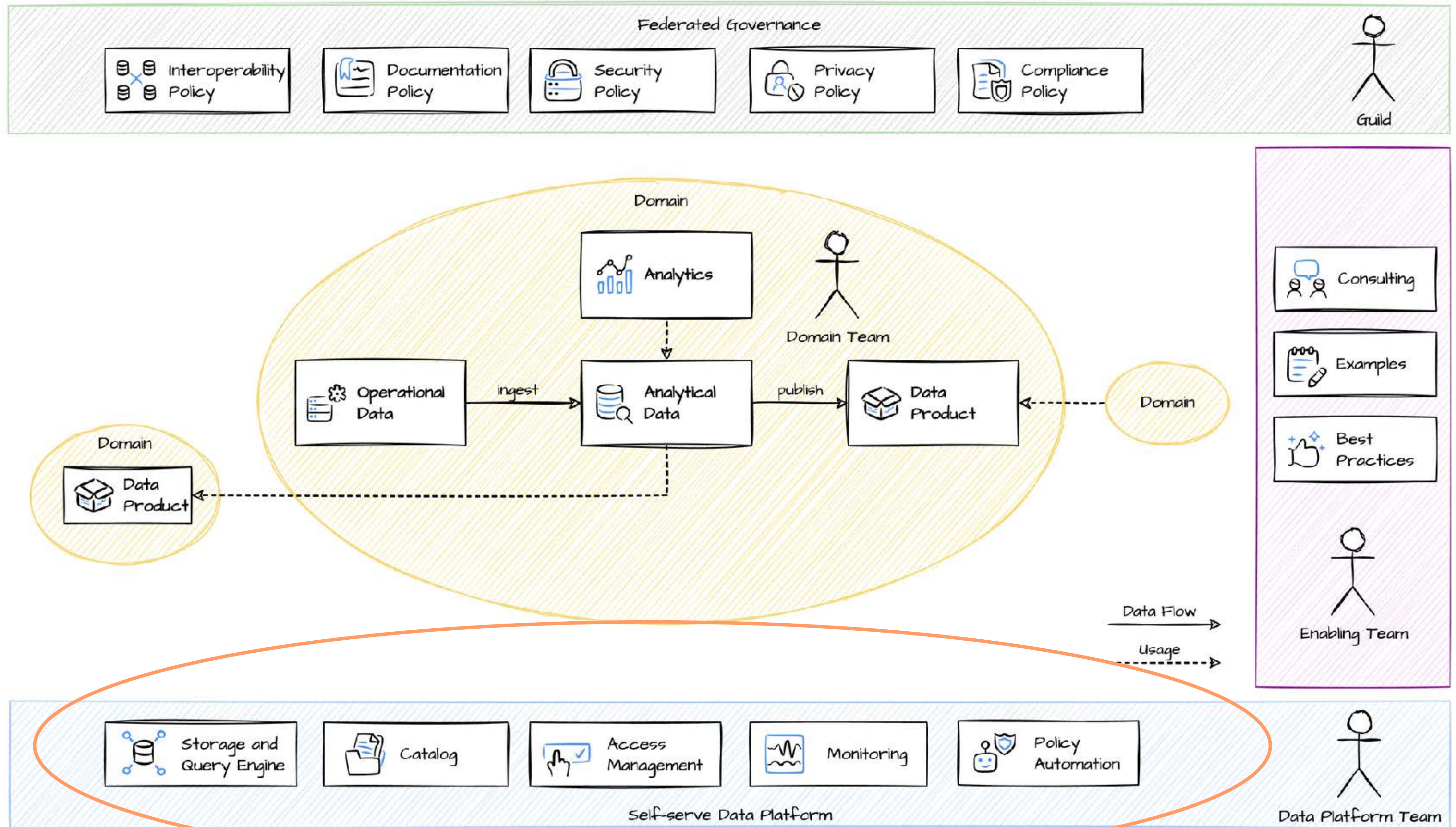
index	location	available_total
0	10	84
1	11	99
2	12	77
3	13	79
4	14	84
5	15	99
6	16	100
7	17	80
8	18	63
9	19	65

Show 10 per page

1

2

Data Mesh Architecture





AWS Glue Data Catalog

[*] SELECT STAR



Data Platforms



Characteristics

Cloud Warehouse

- SQL-like experience
- Integrated Storage and Query
- Support lake and streaming

Distributed Query Engines

- SQL-like experience (on files!)
- Query and storage separated
- Direct query to external systems

Spark

- Python
- Fast data processing
- Optimized for data science

Vendors

Google BigQuery

Snowflake

AWS Redshift

Azure Synapse Analytics Dedicated SQL Pool

AWS Athena

Starburst/Trino/Presto

Azure Synapse Analytics Serverless SQL Pool

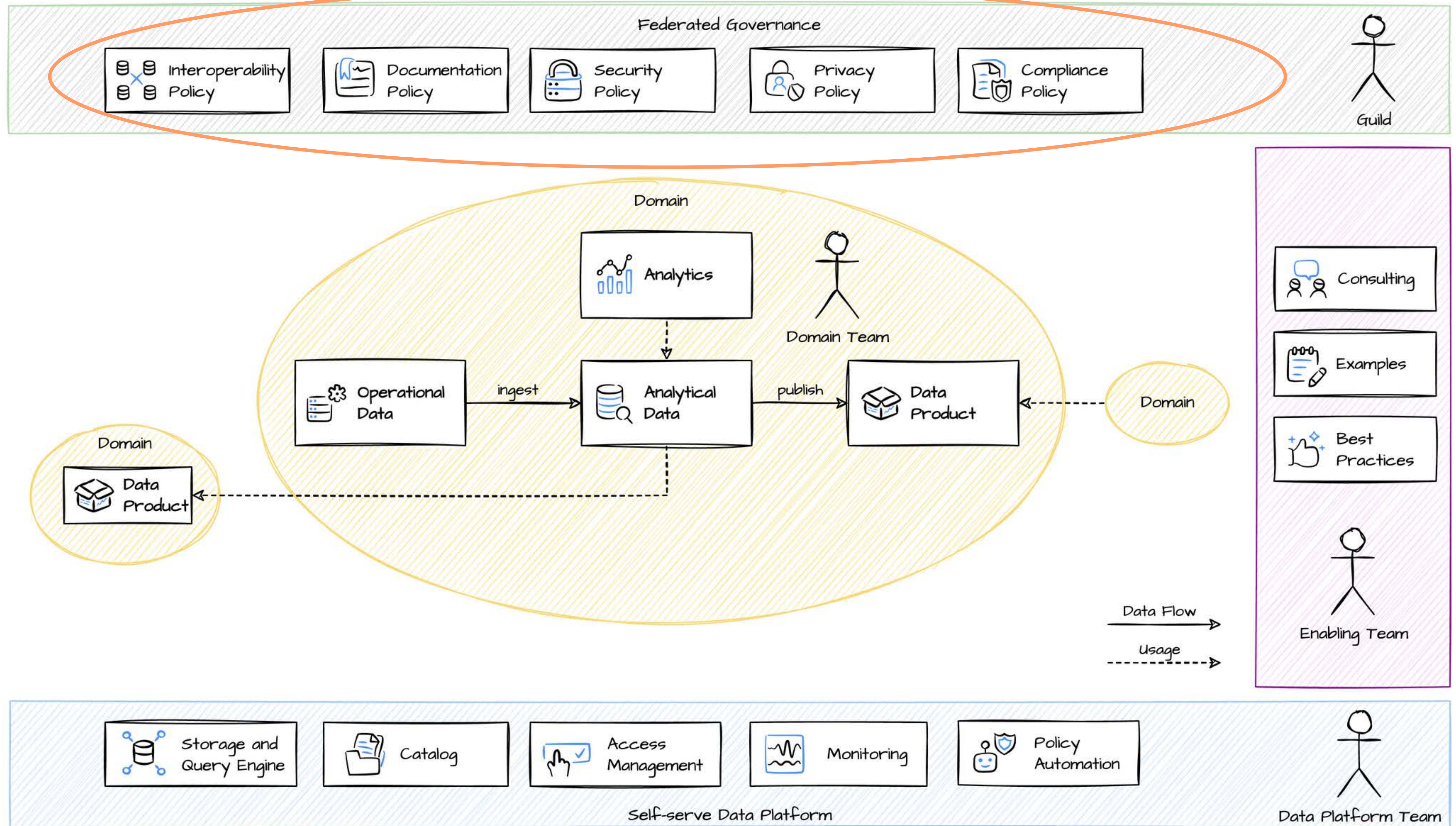
Databricks

Azure Databricks

AWS SageMaker

GCP Serverless Spark

Data Mesh Architecture



Global Policy 1: Data Format

CSV-Files

Standard: RFC 4180

Delimiter: COMMA

Header-Record: Required

Global Policy 2: Location

S3

Bucket is managed by domain team

Global Policy 3: Data Discovery

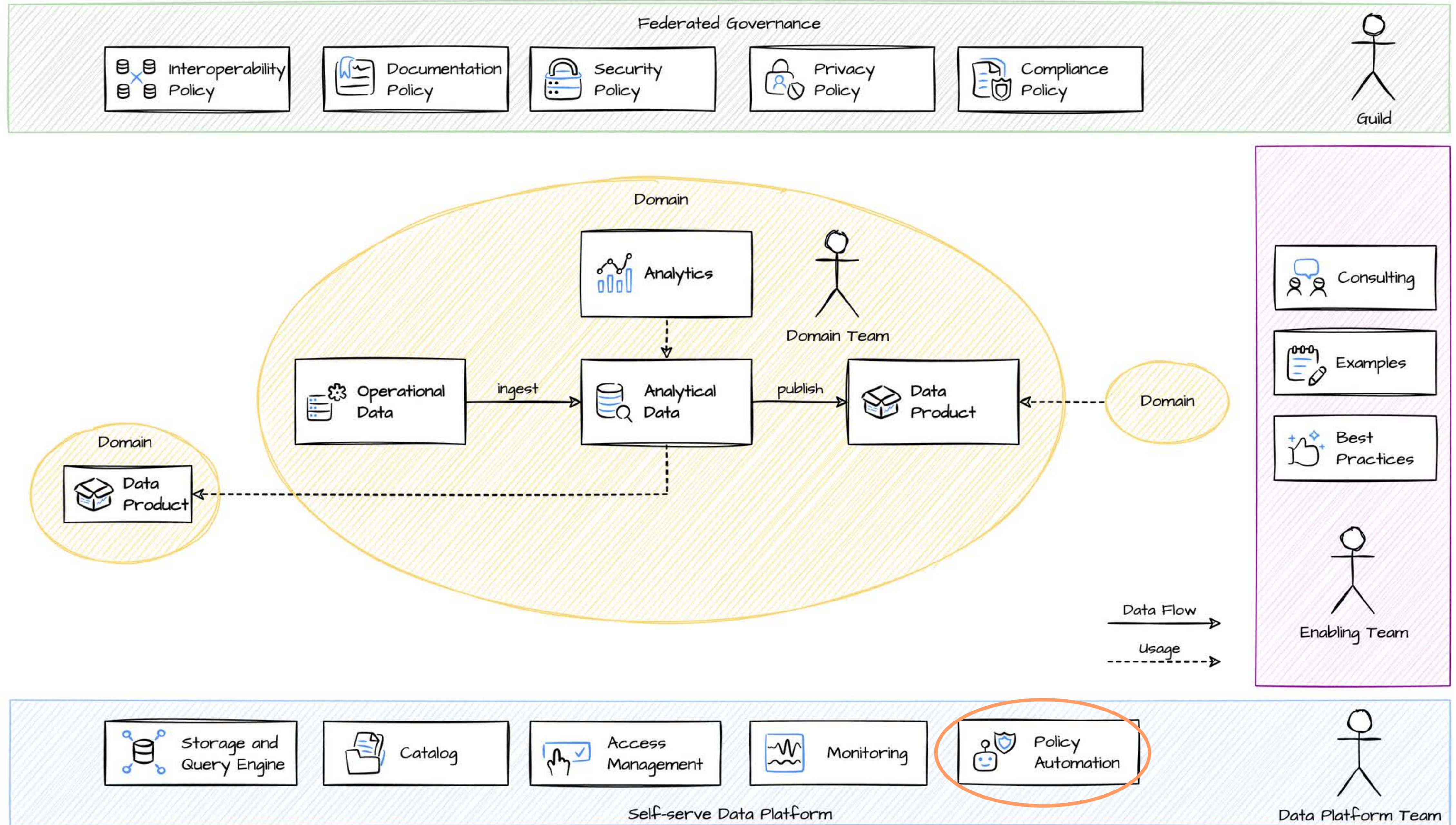
A common table in Wiki that lists:

- Domain Team
- Data Product Name
- Responsible Product Owner
- S3-Location
- Data Freshness / Update frequency
- Tags
- Data Model

Global Policy 4: Access Control

AWS IAM Role based ACL

Data Mesh Architecture





The Problem with Data Mesh



Dev Teams' Cognitive Load

Dev

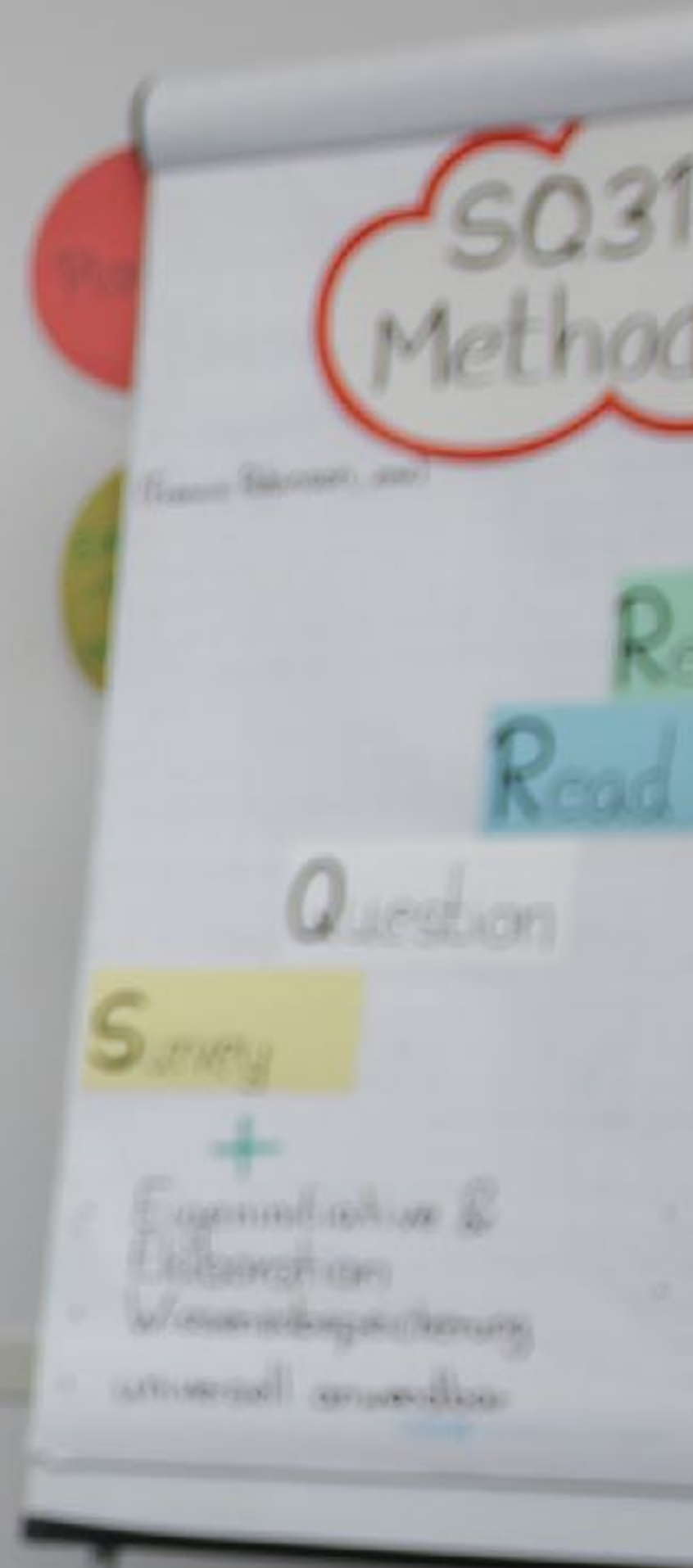
**Fullstack
Dev**

Fullstack DevOps

**Fullstack
DevSecOps**

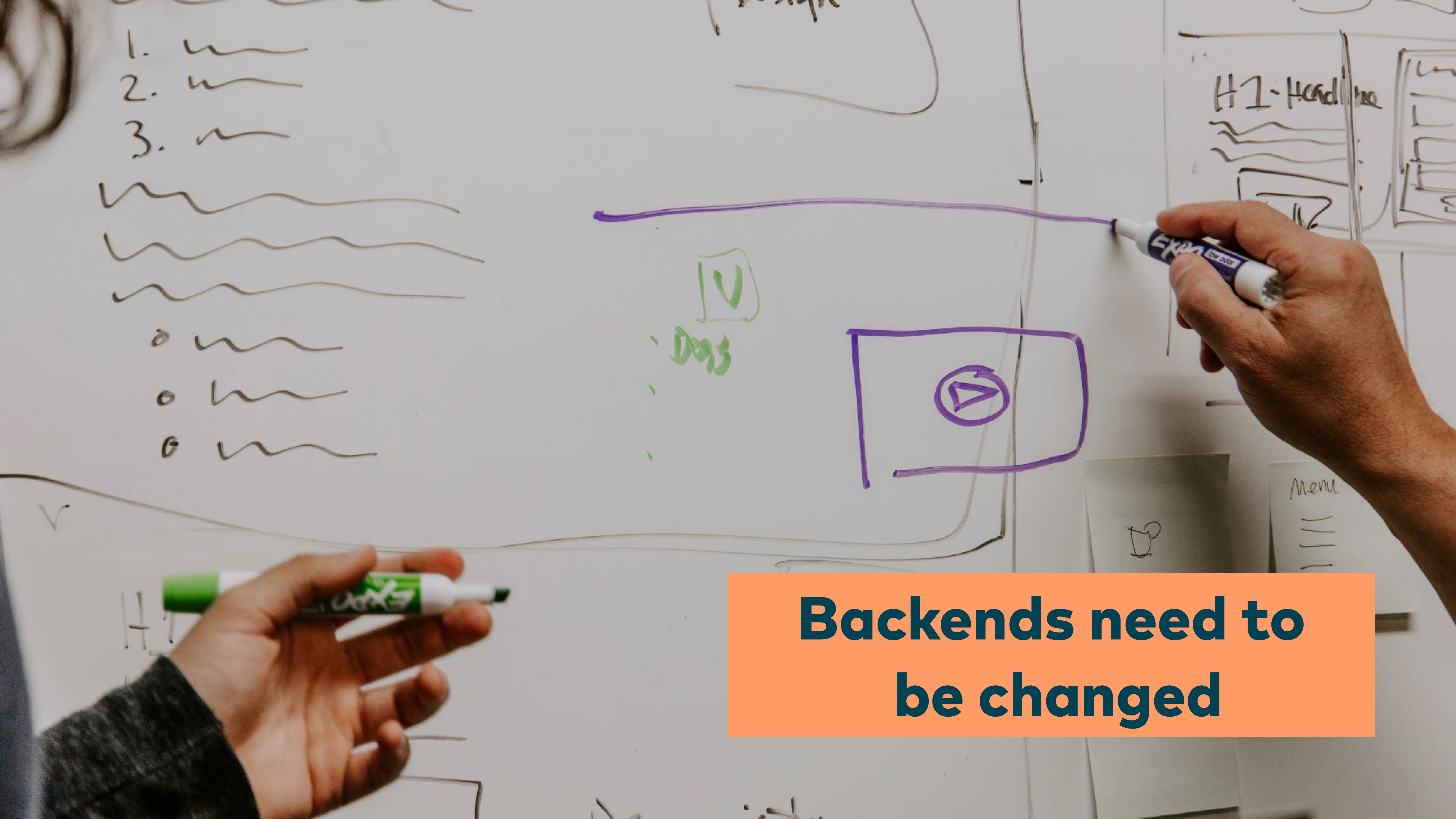
Fullstack
BizDevSecOps

Fullstack
BizDevSecDataOps





Return on Investment



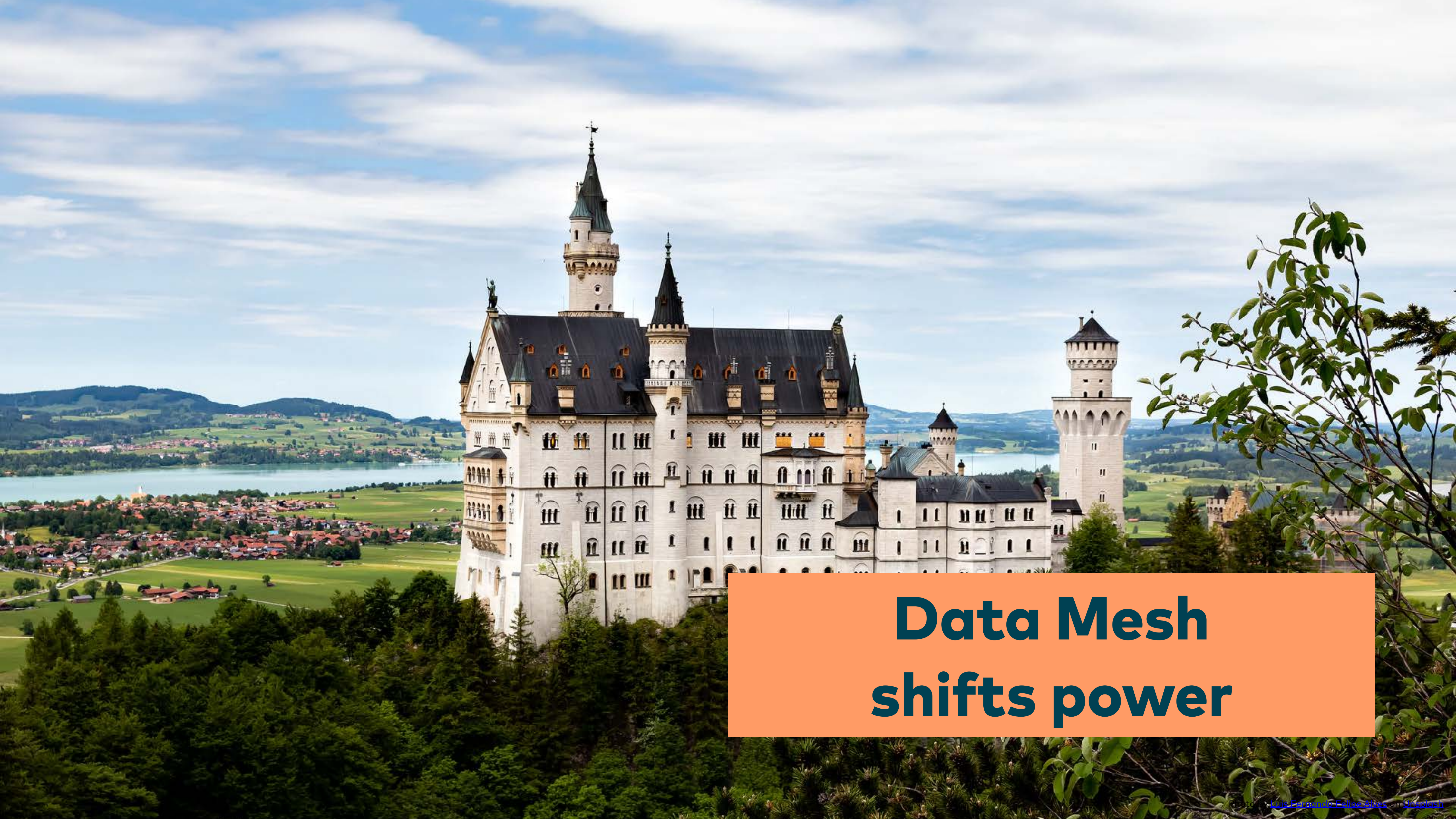
**Backends need to
be changed**



Teams are different



Data Governance Know How

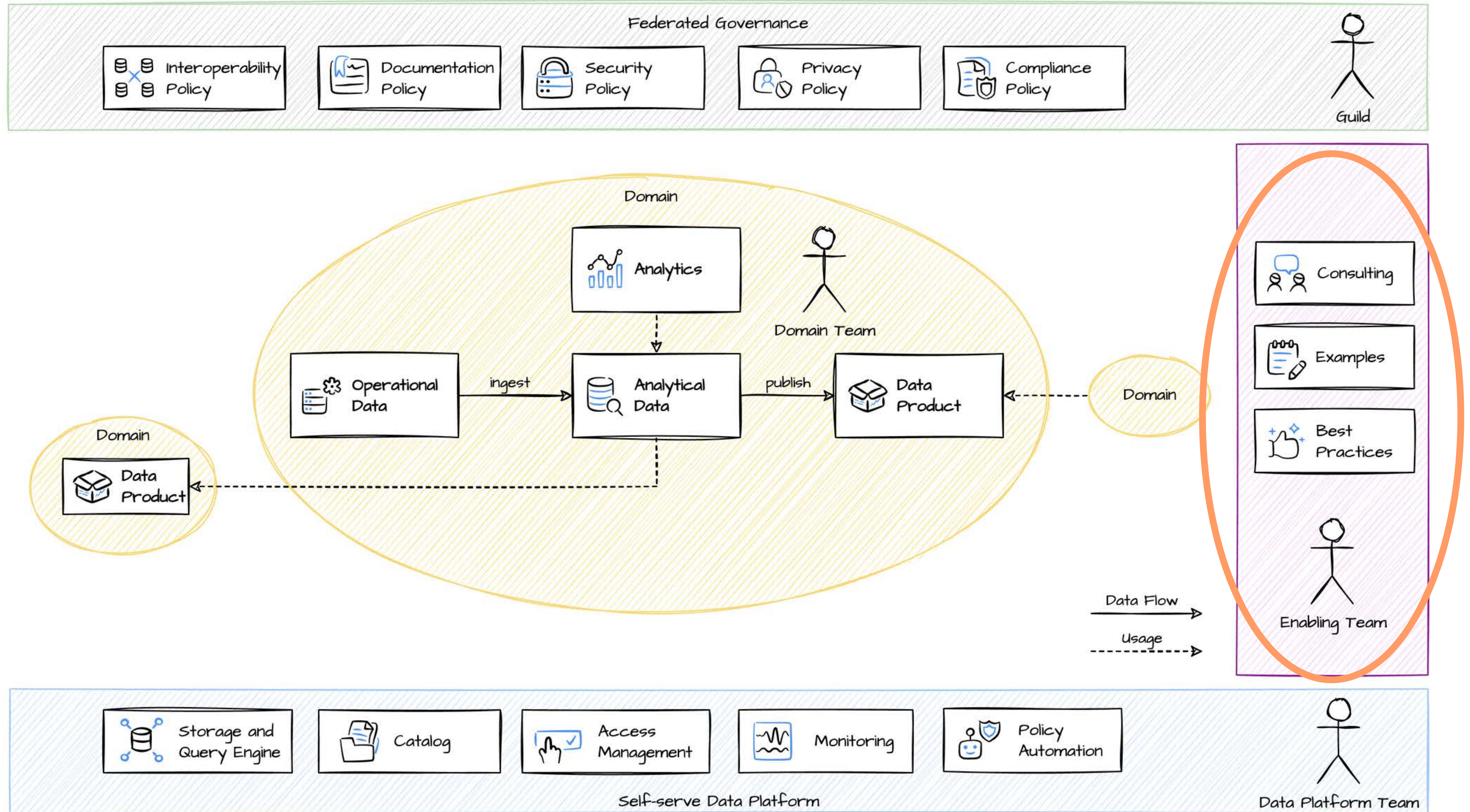


Data Mesh shifts power

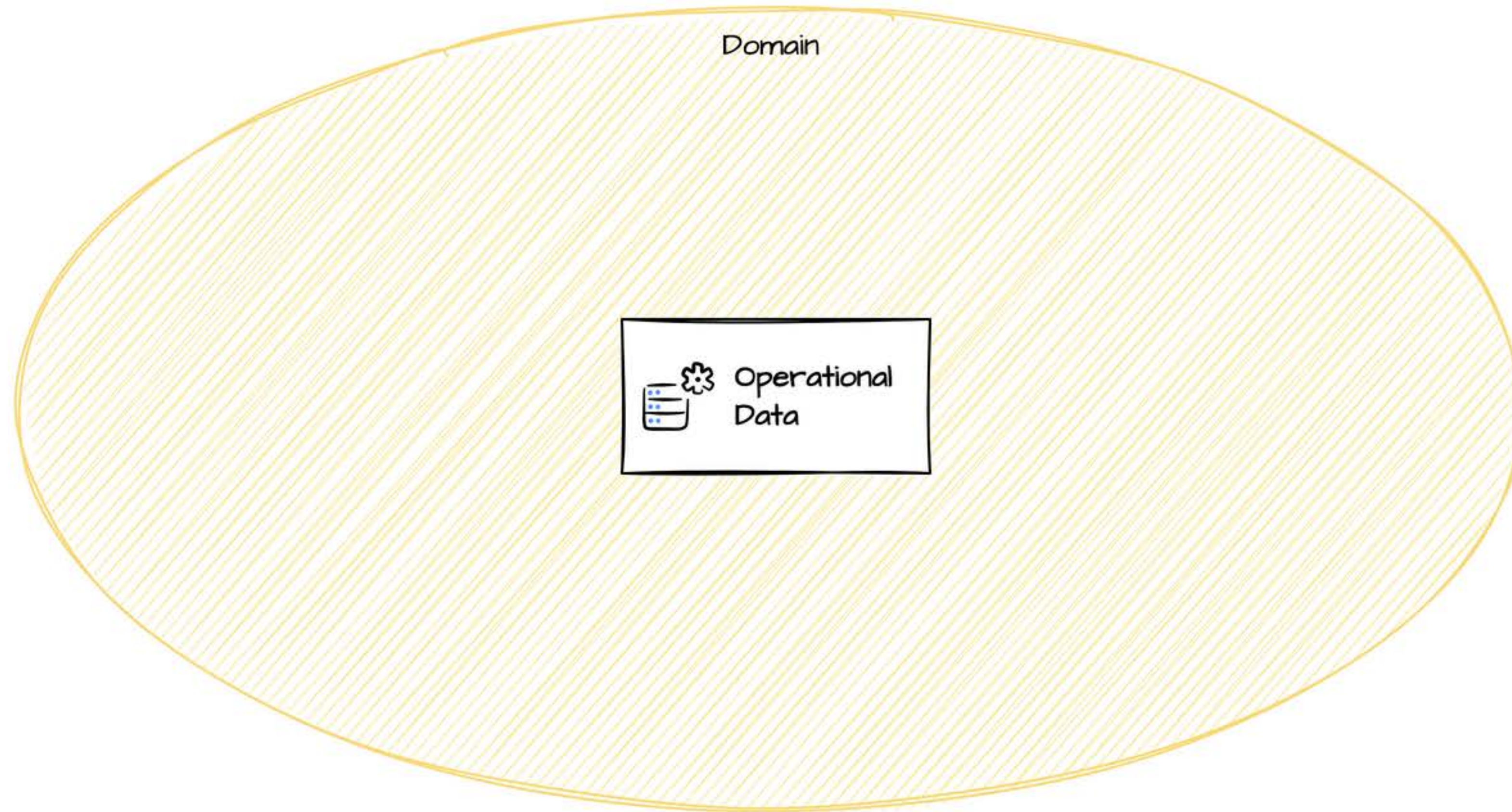


How to start?

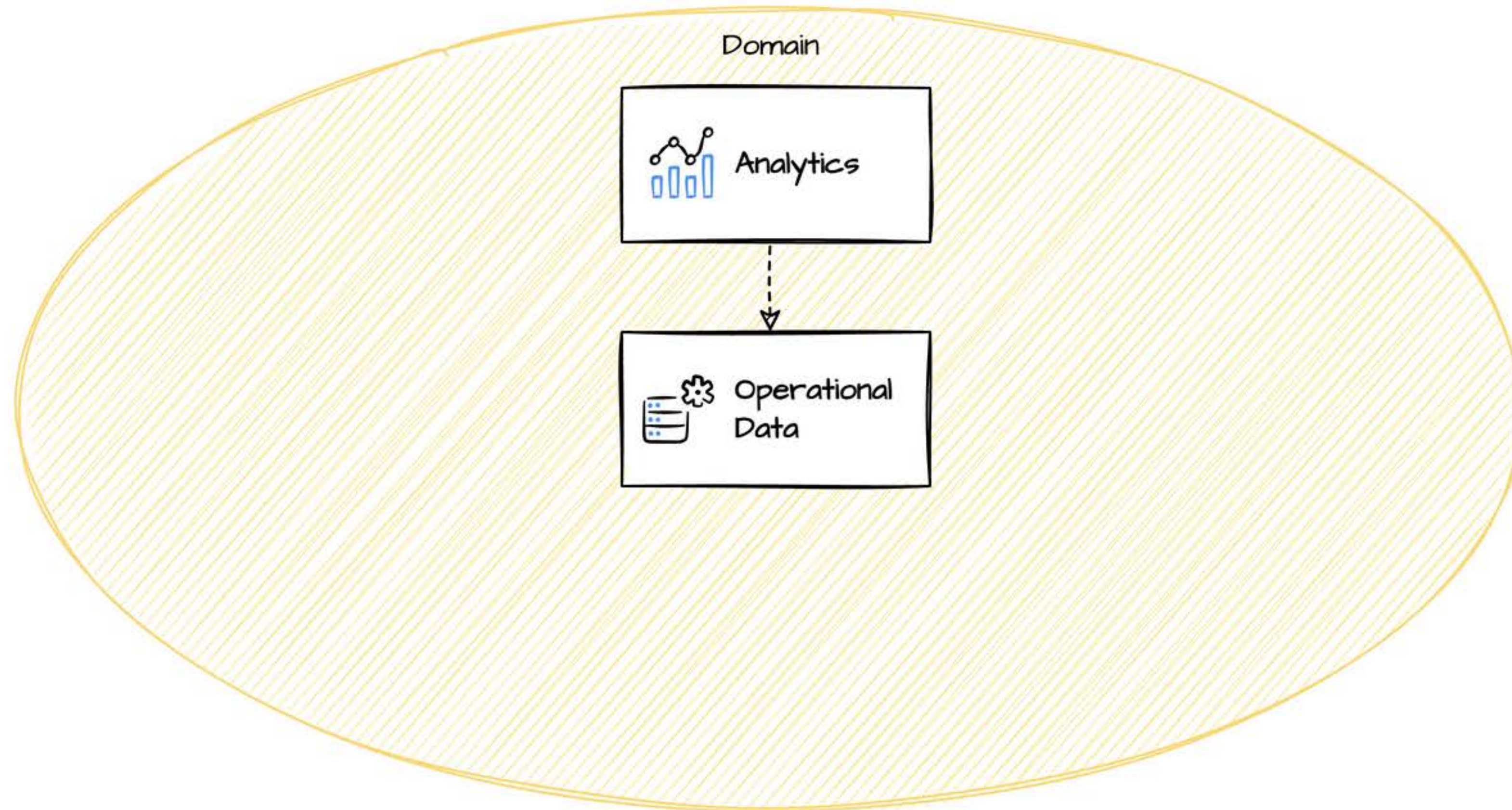
Data Mesh Architecture



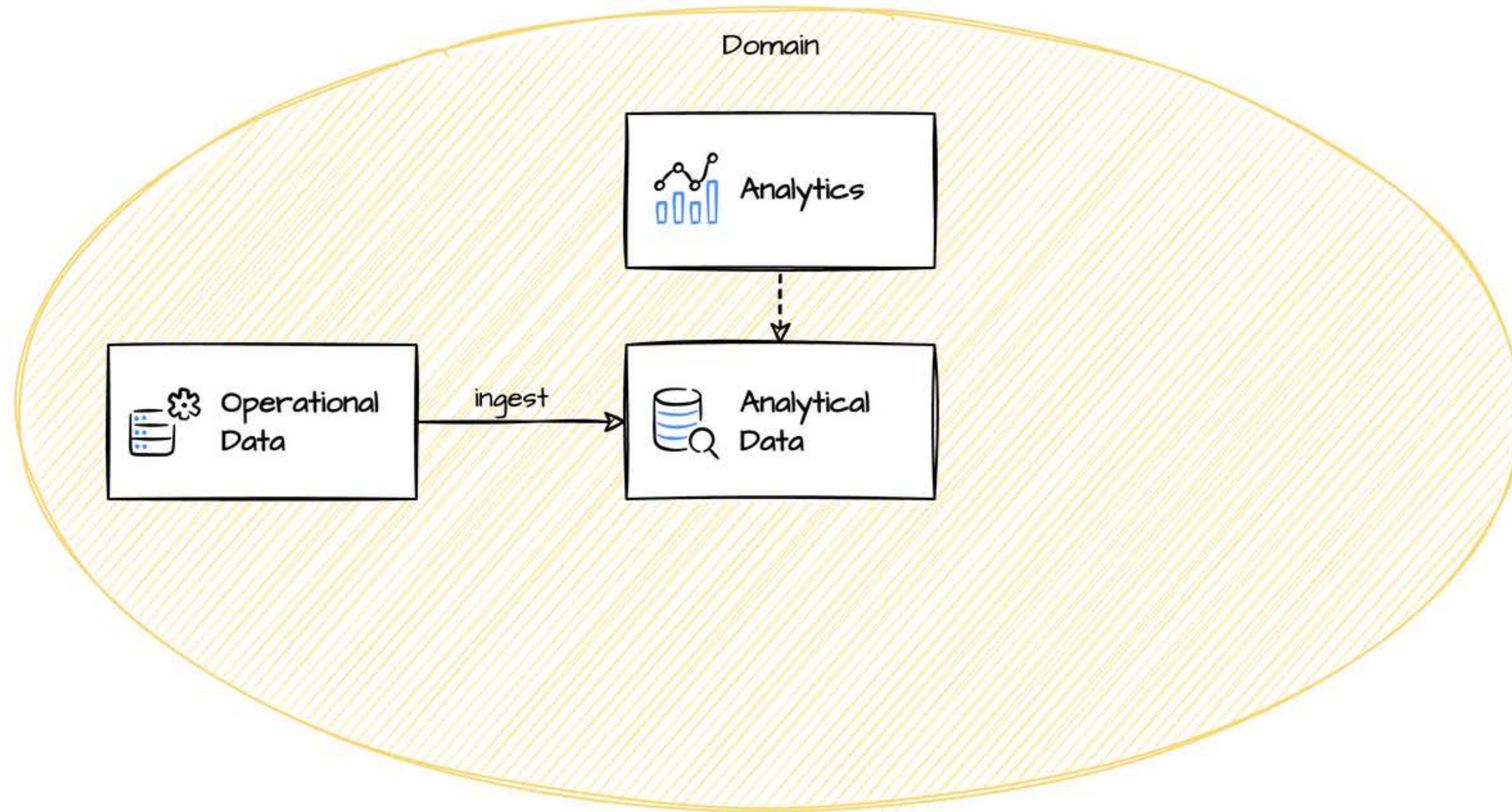
Level 0



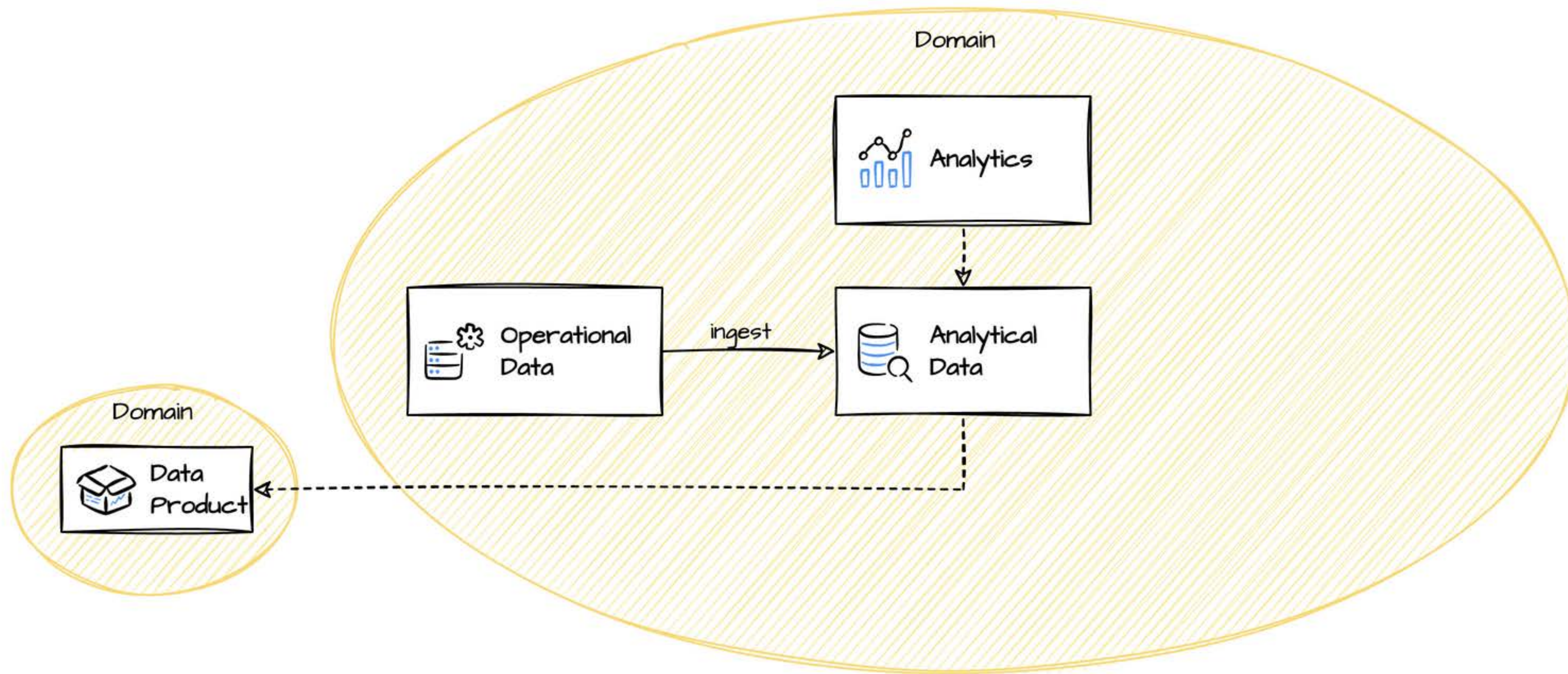
Level 1



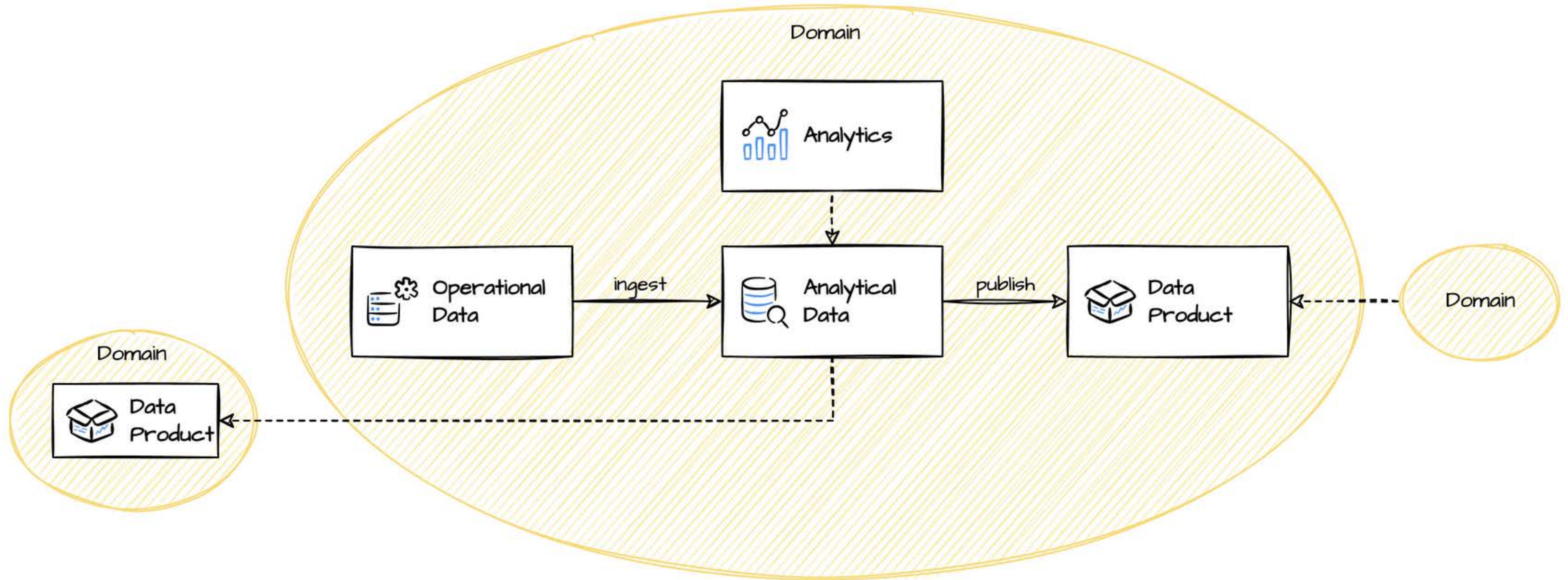
Level 2



Level 3



Level 4



DATA MESH IS A SOCIO-TECHNICAL APPROACH

Building a good data platform is **important**

Changing organisation structures is **hard**

Establishing a data culture is **key**

<ad>

Entscheidung



Data Mesh Workshop

INNOQ bietet einen 1-Tages-Workshop zu Data Mesh an, der für dein Unternehmen folgende Fragen klärt: Was ist Data Mesh wirklich? Ist Data Mesh sinnvoll für uns? Sind wir bereit? Und wie führe ich Data Mesh ein?

Wie läuft der Tag ab?

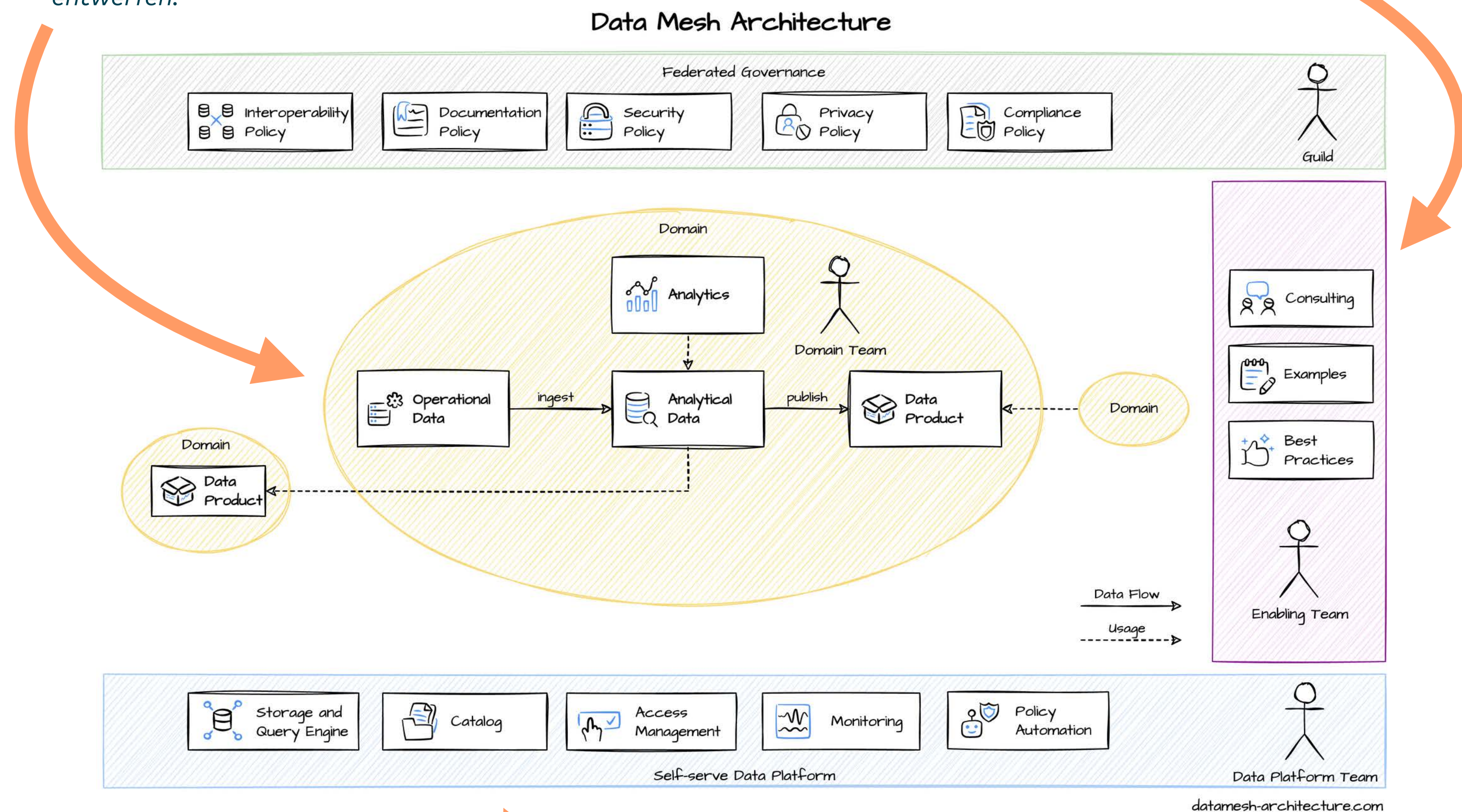
- 10:00 Vorstellung und Erwartungen (30min)
- 10:30 Was ist Data Mesh wirklich? (90min)
- 12:00 Mittagspause (60 min)
- 13:00 Ist Data Mesh sinnvoll für uns? Sind wir bereit? (90min)
- 14:30 Kaffeepause (30min)
- 15:00 Wie führe ich Data Mesh ein? (90min)
- 16:30 Offene Fragen und Feedback (30min)

<https://www.innoq.com/de/topics/data-mesh-workshop/>

Umsetzung

"Wir bieten **Trainings** für Entwickler:innen an, um mit entwicklungsnahe Werkzeugen wie dbt eigene Analysen durchzuführen und Datenprodukte zu entwerfen."

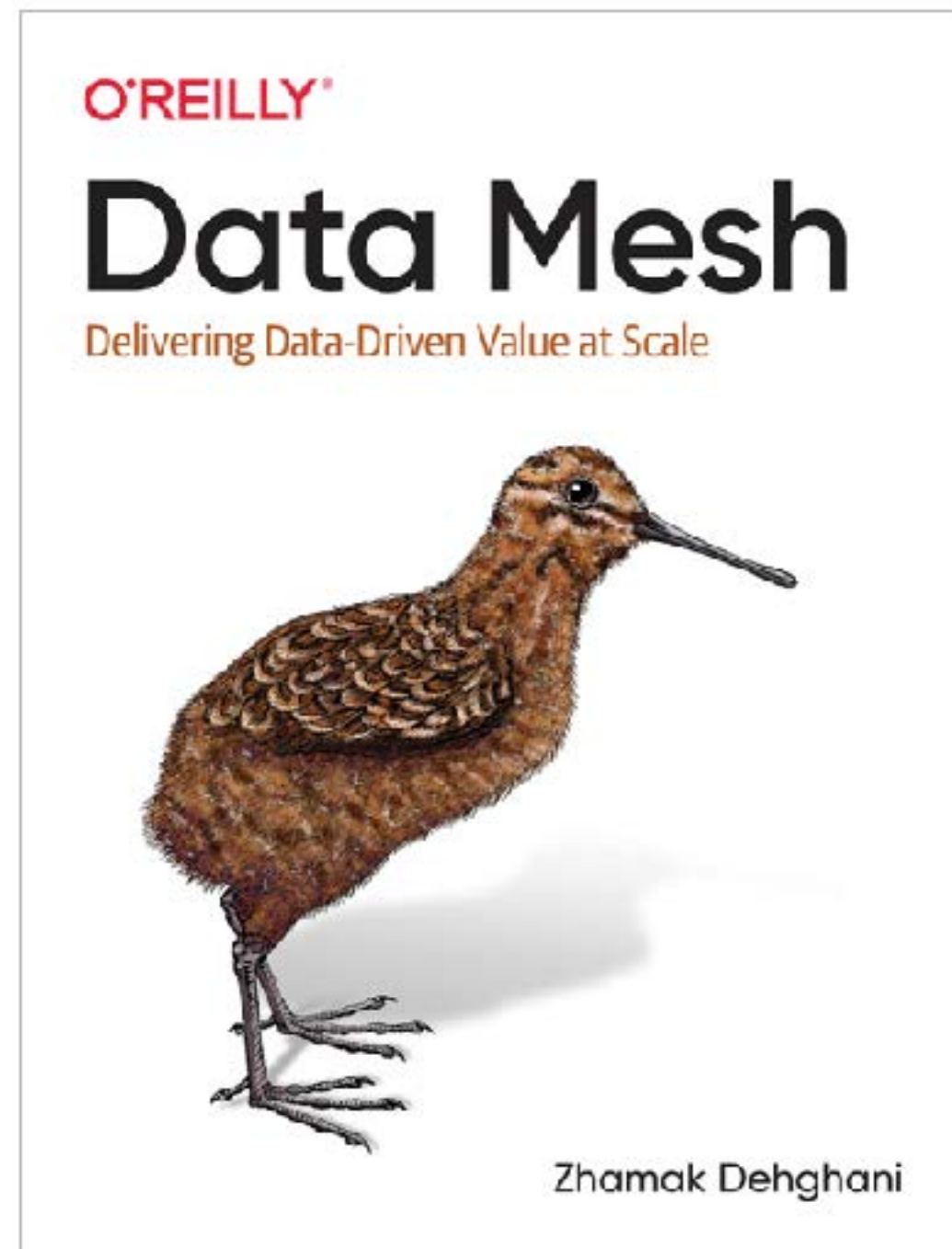
"Als Teil eines **Enabling Teams** helfen wir Data Mesh innerhalb eines Unternehmens aufzusetzen, zu promoten und es Engineers leicht zu machen, die Data Platform für ihre analytischen Daten zu nutzen."



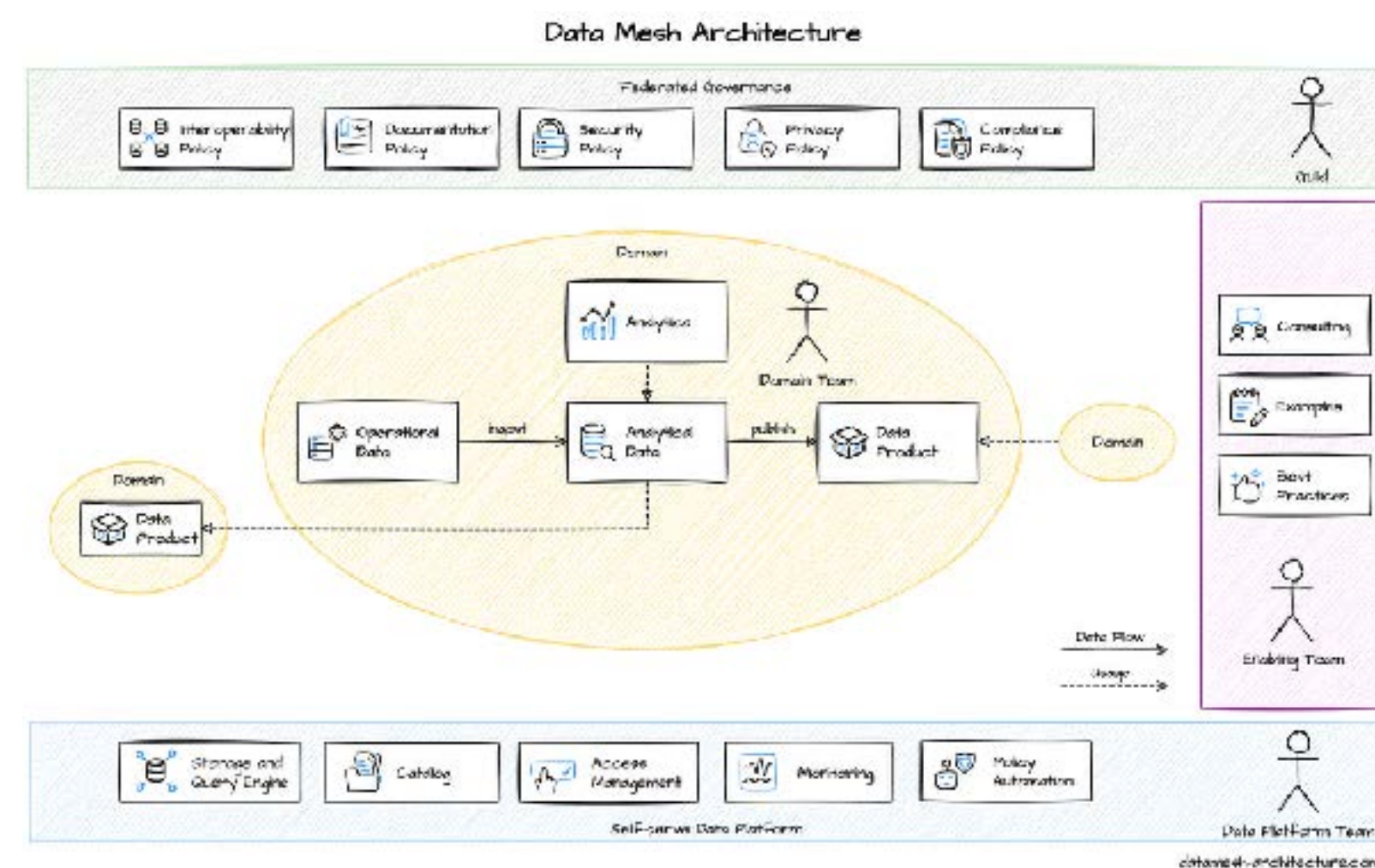
"Wir helfen bei der **Auswahl passender Technologien und Produkte** für die Data Platform und setzen diese anhand von **Best Practices** auch gerne mit auf."

</ad>

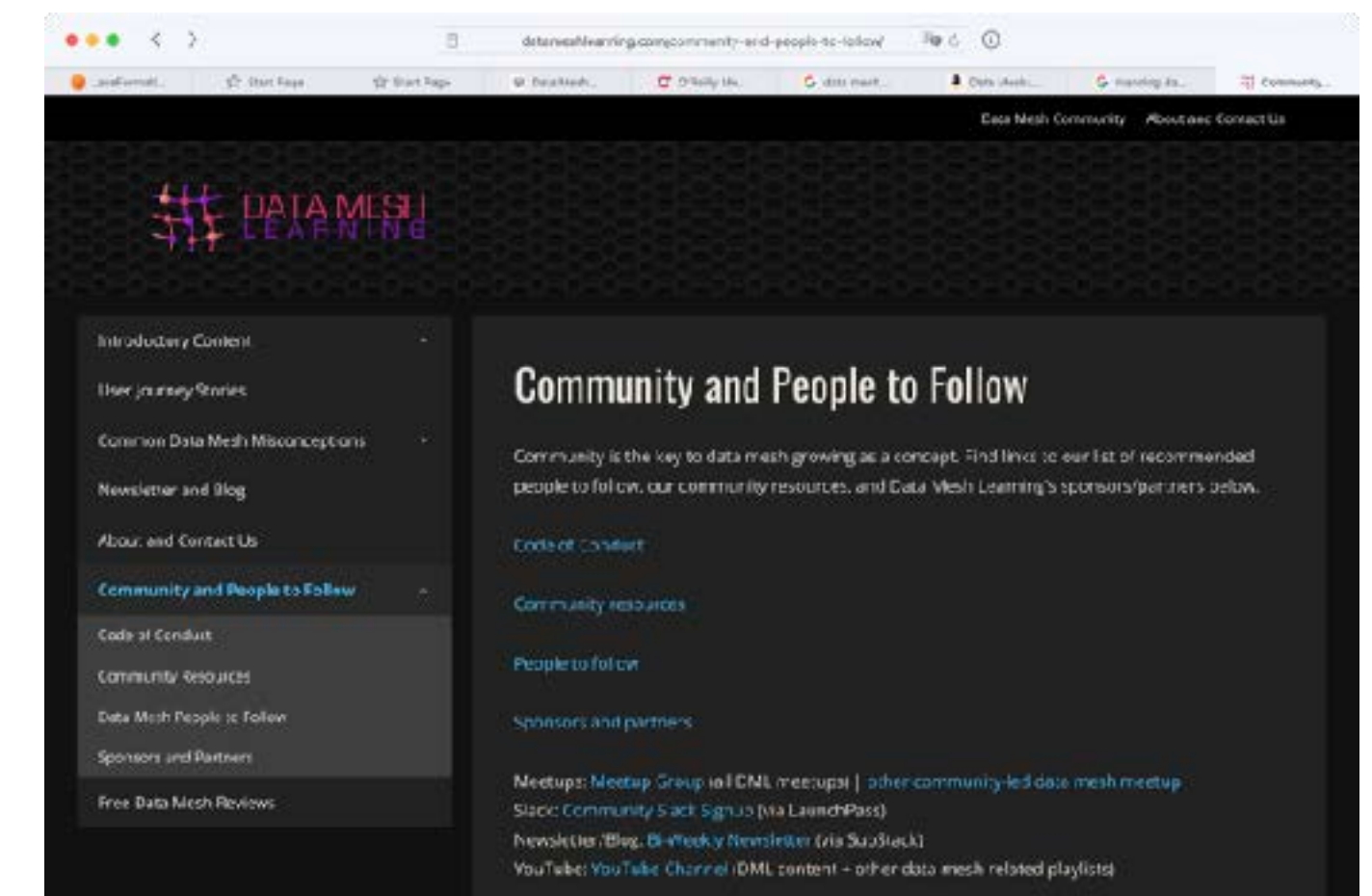
Learn more



Free* copy:
starburst.io/info/oreilly-data-mesh/
*) you pay with your data



datamesh-architecture.com



datameshlearning.com
Podcast, newsletter, slack channel

Data Mesh

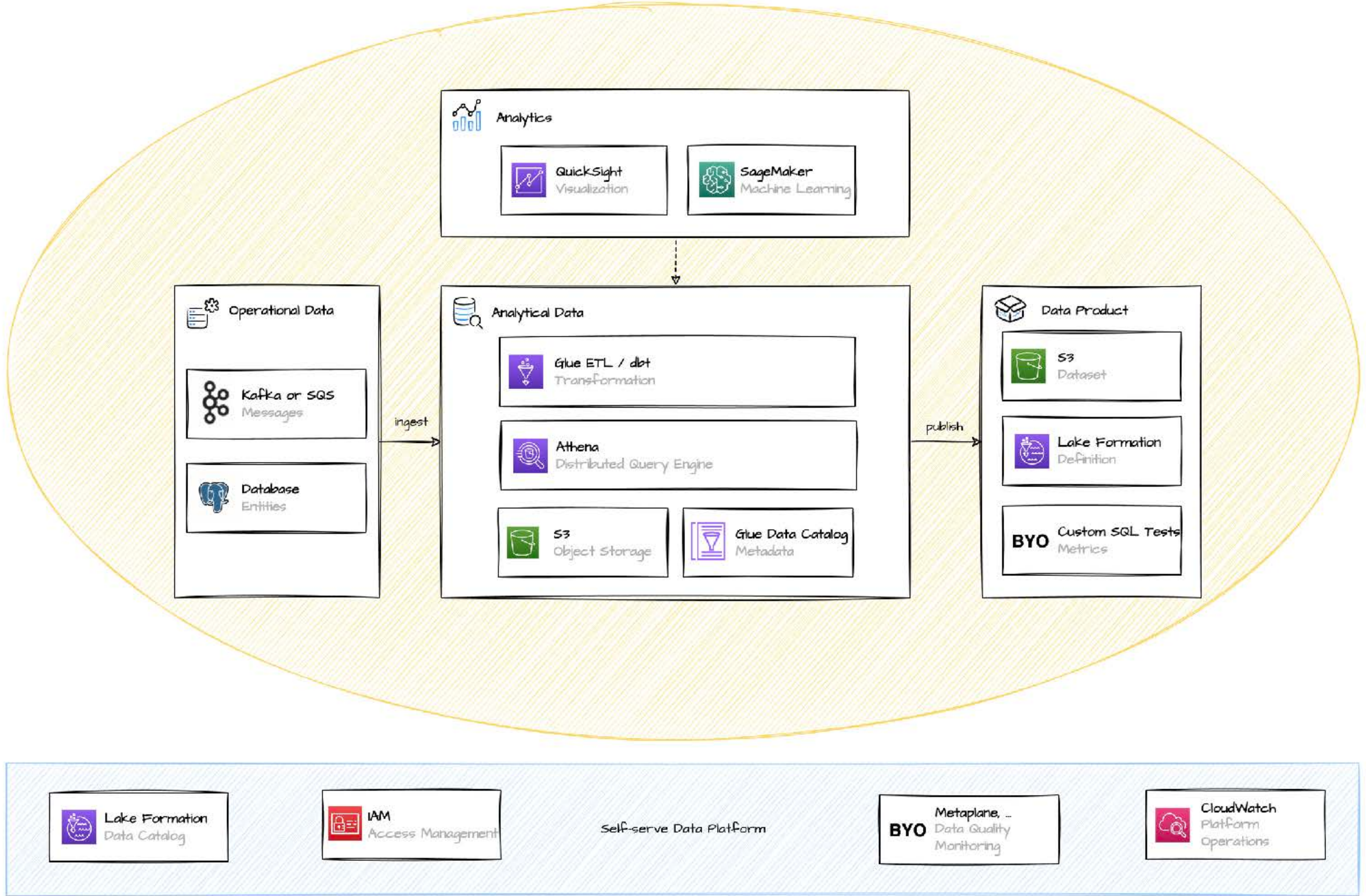
(für Entwickler:innen)



JOCHEN CHRIST
@JOCHEN_CHRIST

AWS Athena Example

Data Mesh Architecture with AWS S3 and Athena



>  Cluster overview

 Topics

>  Data integration

 Stream lineage

 ksqlDB

 Schema Registry

CLI and tools

Support

Topics

☒ Hide internal topics

+ Add topic

Topic name	Partitions	Production (last min)	Consumption (last min)	Schema
dlq-lcc-6ko7p6	1	--	--	Set a schema
dlq-lcc-7yym31	1	--	--	Set a schema
inventory	2	11.62K3/s	23.23KE/s	Set a schema

- >  **Cluster overview**
 -  **Topics**
 - >  **Data integration**
 -  **Stream lineage**
 -  **ksqlDB**

-  **Schema Registry**

inventory

[➡ See in Stream lineage](#)[Overview](#) [Messages](#) [Schema](#) [Configuration](#)

Producers

Bytes in/sec	11.62K
--------------	--------

Consumers

Bytes out/sec	23.23K
---------------	--------

Message fields

- topic
- partition
- offset
- timestamp
- timestampType
- ✓ headers
 - key
 - stringValue
 - key
 - stringValue
 - key
 - stringValue
 - key
 - stringValue
 - key
 - stringValue
- key

Filter by keyword

Jump to offset

▼

offset

[+ Produce a new message to this topic](#)

```

{"sku": "9520548895159", "location": "16", "available": 8, "updated_at": "2022-05-04T14:22:26.865382Z"}
Partition: 0      Offset: 2686189      Timestamp: 1651674146865

```

```

{"sku": "9520548895159", "location": "16", "available": 9, "updated_at": "2022-05-04T14:22:26.865374Z"}
Partition: 0      Offset: 2686188      Timestamp: 1651574146865

```

```

✓ {"sku":"9520548895159","location":"16","available":10,"updated_at":"2022-05-04T14:22:26.865365Z"}
Partition: 0      Offset: 2686187      Timestamp: 1651574146865

```

▼ {"sku":"9520548895159","location":"16","available":11,"updated_at":"2022-05-04T14:22:26.865357Z"}
Partition: 0 Offset: 2686186 Timestamp: 1651574146865

```

▼ {"sku":"9520548895159","location":"16","available":10,"updated_at":"2022-05-04T14:22:26.865348Z"}
  Partition: 0      Offset: 2686185      Timestamp: 1651574146865

```

```

{"sku":"9520548095159","location":"16","available":11,"updated_at":"2022-05-04T14:22:26.865341Z"}
Partition: 0      Offset: 2686184      Timestamp: 1651674146865

```


Amazon S3

Buckets

Access Points

Object Lambda Access Points

Multi-Region Access Points

Batch Operations

Access analyzer for S3

Block Public Access settings for this account

Storage Lens

Dashboards

AWS Organizations settings

Feature spotlight

3

AWS Marketplace for S3

Amazon S3

Buckets

datameshexample-fullfillment-inventory

topics/

inventory/

year=2022/

month=04/

day=20/

hour=00/

hour=00/

Copy S3 URI

Objects

Properties

To enable sorting in the table below, use the search to reduce the size of the list to 999 objects or fewer.

Objects (999+)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Refresh

Copy S3 URI

Copy URL

Download

Open

Delete

Actions

Create folder

Upload

Find objects by prefix

< 1 2 3 ... > ⚙

☐	Name	Type	Last modified	Size	Storage class
☐	 inventory+0+0001327496.json	json	April 20, 2022, 15:50:35 (UTC+02:00)	95.1 KB	Standard
☐	 inventory+0+0001328496.json	json	April 20, 2022, 15:50:36 (UTC+02:00)	95.1 KB	Standard
☐	 inventory+0+0001329496.json	json	April 20, 2022, 15:50:38 (UTC+02:00)	95.0 KB	Standard
☐	 inventory+0+0001330496.json	json	April 20, 2022, 15:50:38 (UTC+02:00)	95.1 KB	Standard
☐	 inventory+0+0001331496.json	json	April 20, 2022, 15:50:38 (UTC+02:00)	95.1 KB	Standard
☐	 inventory+0+0001332496.json	json	April 20, 2022, 15:50:40 (UTC+02:00)	95.1 KB	Standard
☐	 inventory+0+0001333496.json	json	April 20, 2022, 15:50:40 (UTC+02:00)	95.0 KB	Standard
☐	 inventory+0+0001334496.json	json	April 20, 2022, 15:50:40 (UTC+02:00)	95.0 KB	Standard
☐	 inventory+0+0001335496.json	json	April 20, 2022, 15:50:42 (UTC+02:00)	95.0 KB	Standard
☐	 inventory+0+0001336496.json	json	April 20, 2022, 15:50:42 (UTC+02:00)	94.9 KB	Standard
☐	 inventory+0+0001337496.json	json	April 20, 2022, 15:50:42 (UTC+02:00)	95.0 KB	Standard
☐	 inventory+0+0001338496.json	json	April 20, 2022, 15:50:44 (UTC+02:00)	95.0 KB	Standard
☐	 inventory+0+0001339496.json	json	April 20, 2022, 15:50:44 (UTC+02:00)	95.0 KB	Standard
☐	 inventory+0+0001340496.json	json	April 20, 2022, 15:50:44 (UTC+02:00)	95.0 KB	Standard
☐	 inventory+0+0001341496.json	json	April 20, 2022, 15:50:46 (UTC+02:00)	95.1 KB	Standard
☐	 inventory+0+0001342496.json	json	April 20, 2022, 15:50:46 (UTC+02:00)	95.1 KB	Standard
☐	 inventory+0+0001343496.json	json	April 20, 2022, 15:50:48 (UTC+02:00)	95.0 KB	Standard

Feedback

Looking for language selection? Find it in the new [Unified Settings](#)

© 2022, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences

eu-central-1.console.aws.amazon.com/athena/home?region=eu-central-1#/query-editor/history/2ad8f9

Search for services, features, blogs, docs, and more

[Option+S]

Frankfurt

jcchen @ 5281-1513-9298

Amazon Athena > Query editor

EditorRecent queriesSaved queriesSettings

Workgroupprimary

Data

Data Source

AwsDataCatalog

Database

fulfillment

Tables and views

Create

Filter tables and views

Tables (1)

1

inventory

Partitioned

Views (2)

1

inventory__available_total_by_location_latest

inventory_latest

Query 5 xQuery 6 xInventory Latest x

1WITH inventory_latest AS (
2SELECT *
3FROM (
4SELECT *,
5"row_number" OVER (
6PARTITION BY sku,
7location
8ORDER BY updated_at DESC
9) row_number
10FROM "inventory"
11)
12WHERE (row_number = 1)
13)
14SELECT sku,
15location,
16
17
18
19
20

SQLLn 10, Col 20

Run againCancelSaveClearCreate

Completed

Time in queue: 0.108 secRun time: 3.786 secData scanned: 246.24 MB

Results (100+)

Search rows

#skulocationavailableupdated_at

19521012022-03-17T14:35:40.918316Z

295200002279721902022-03-17T14:59:58.970737Z

395200005024841102022-03-17T15:01:14.768794Z

495200005195961102022-03-17T15:01:05.298968Z

595200006211181602022-03-17T14:59:43.609678Z

695200006264451002022-03-17T15:00:28.598760Z

Feedback

Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Web Services, Inc. or its affiliates. PrivacyTermsCookie preferences

AWS Glue

Data catalog

Databases

Tables

Connections

Crawlers

Classifiers

Schema registries

Schemas

Settings

ETL

AWS Glue Studio

Jobs - New

Jobs (legacy)

ML Transforms

Blueprints

Workflows

Triggers

Dev endpoints

Notebooks

Interactive Session - New

Security

Security configurations

Tutorials

Add crawler

Explore table

Add job

Resources

What's new 10+

Tables A table is the metadata definition that represents your data, including its schema. A table can be used as a source or target in a job definition.

Add tables

Action

Database : fulfillmentFilter or search for tables...

Save view

Showing: 1 - 3

Name	Database	Location	Classification	Last updated	Deprecated
☐ inventory_latest	fulfillment		Unknown	17 March 2022 4:53 AM UTC+1	
☐ inventory	fulfillment	s3://datameshexample-fulfillment-in...	json	17 March 2022 4:46 AM UTC+1	
☐ inventory__available_total_by_location_latest	fulfillment		Unknown	17 March 2022 5:13 AM UTC+1	

AWS Glue

Data catalog

Databases

Tables

Connections

Crawlers

Classifiers

Schema registries

Schemas

Settings

ETL

AWS Glue Studio

Jobs - New

Jobs (legacy)

ML Transforms

Blueprints

Workflows

Triggers

Dev endpoints

Notebooks

Interactive Session - New

Security

Security configurations

Tutorials

Add crawler

Explore table

Add job

Resources

What's new 10+

Tables > inventory

Last updated 17 Mar 2022 04:46 PM

Table Version (Current version)

Edit table

Delete table

Partitions and indices

View partitions

Compare versions

Edit schema

Name

inventory

Description

fulfillment

Database

fulfillment

Classification

json

Location

s3://datameshexample-fulfillment-inventory/topics/inventory/

Connection

Deprecated

No

Last updated

Thu Mar 17 16:46:32 GMT+100 2022

Input format

org.apache.hadoop.mapred.TextInputFormat

Output format

org.apache.hadoop.hive ql.io.HiveIgnoreKeyTextOutputFormat

Serde serialization lib

org.openx.data.jsonserde.JsonSerDe

Serde parameters

pathsavailable,location,sku,updated_at

Table properties

sizeKey258202545objectCount2654UPDATED_BY_CRAWLERinventoryCrawlerSchemaSerializerVersion1.0recordCount2660576

averageRecordSize97CrawlerSchemaDeserializerVersion1.0compressionTypenonetypeOfDatafile

Schema

Showing: 1 - 8 of 8

	Column name	Data type	Partition key	Comment
1	updated_at	string		
2	avaiable	int		
3	location	string		
4	sku	string		
5	year	string	Partition (0)	
6	month	string	Partition (1)	
7	day	string	Partition (2)	
8	hour	string	Partition (3)	

Amazon Athena > Query editor

Editor Recent queries Saved queries Settings

Workgroup primary

Data

Data Source
AwsDataCatalog

Database
fulfillment

Tables and views
Create Filter tables and views

Tables (2)
inventory Partitioned
inventory_latest_test1

Views (2)
inventory__available_total_by_location_latest
inventory_latest

Query 5 x Query 6 x Inventory Latest x Query 8 x

```
1 CREATE TABLE "fulfillment"."inventory_latest_test1"
2 WITH (
3   format='parquet',
4   external_location='s3://datameshexample-fulfillment-athena-results/inventory_latest_test1/'
5 ) AS
6 WITH inventory_latest AS (
7   SELECT *
8   FROM (
9     SELECT *,
10    "row_number" OVER (
11      PARTITION BY sku,
12      location
13      ORDER BY updated_at DESC
14    ) row_number
15   FROM "inventory"
```

SQL Ln 1, Col 1

Run again Cancel Save Clear Create

Completed Time in queue: 0.206 sec Run time: 4.099 sec Data scanned: 246.24 MB

Query successful.

eu-central-1.console.aws.amazon.com/athena/home?region=eu-central-1#/query-editor/history/2ad8f9

Search for services, features, blogs, docs, and more

[Option+S]

Frankfurt

jcchen @ 5281-1513-9298

Amazon Athena > Query editor

EditorRecent queriesSaved queriesSettings

Workgroupprimary

Data

Data Source
AwsDataCatalog

Database
fulfillment

Tables and views
Create

Filter tables and views

Tables (1)
inventory

Views (2)
inventory__available_total_by_location_latest
inventory_latest

Query 5 ×Query 6 ×Inventory Latest ×

```
1 WITH inventory_latest AS (  
2   SELECT *  
3   FROM (  
4     SELECT *,  
5     "row_number" OVER (  
6       PARTITION BY sku,  
7       location  
8       ORDER BY updated_at DESC  
9     ) row_number  
10    FROM "inventory"  
11  )  
12  WHERE (row_number = 1)  
13 )  
14 SELECT sku,  
15   location,
```

SQLLn 10, Col 20

Run againCancelSaveClearCreate

CompletedTime in queue: 0.108 secRun time: 3.786 secData scanned: 246.24 MB

Results (100+)CopyDownload results

Search rows

#	sku	location	available	updated_at
1	952	10	1	2022-03-17T14:35:40.918316Z
2	9520000227972	19	0	2022-03-17T14:59:58.970737Z
3	9520000502484	11	0	2022-03-17T15:01:14.768794Z
4	9520000519596	11	0	2022-03-17T15:01:05.298968Z
5	9520000621118	16	0	2022-03-17T14:59:43.609678Z
6	9520000626445	10	0	2022-03-17T15:00:28.598760Z

Feedback

Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Web Services, Inc. or its affiliates. PrivacyTermsCookie preferences



dbtTM

×

[Learn more](#)

Access analyzer for S3

AWS Organizations settings

► AWS Marketplace for S3

 Copy S3 URI

Properties

 Upload

< 1 > 

© 2022, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

AWS Lake Formation

×

Dashboard

▼ Data catalog

Databases

Tables

Data filters

Settings ⓘ

▼ Register and ingest

Data lake locations

Blueprints

Crawlers ↗

Jobs ↗

▼ Permissions

Administrative roles and tasks

LF-Tags

LF-tag permissions

Data lake permissions

Data locations

External data filtering

AWS Lake Formation

>

Permissions

ⓘ Too many permissions? Filter by database or table. In the navigation page, choose **Databases** or **Tables**. Then choose a database or table, and on the **Actions** menu, choose **View Permissions**.

Data permissions (3)

↻

Revoke

Grant

🔍 Filter permissions by property or value

<

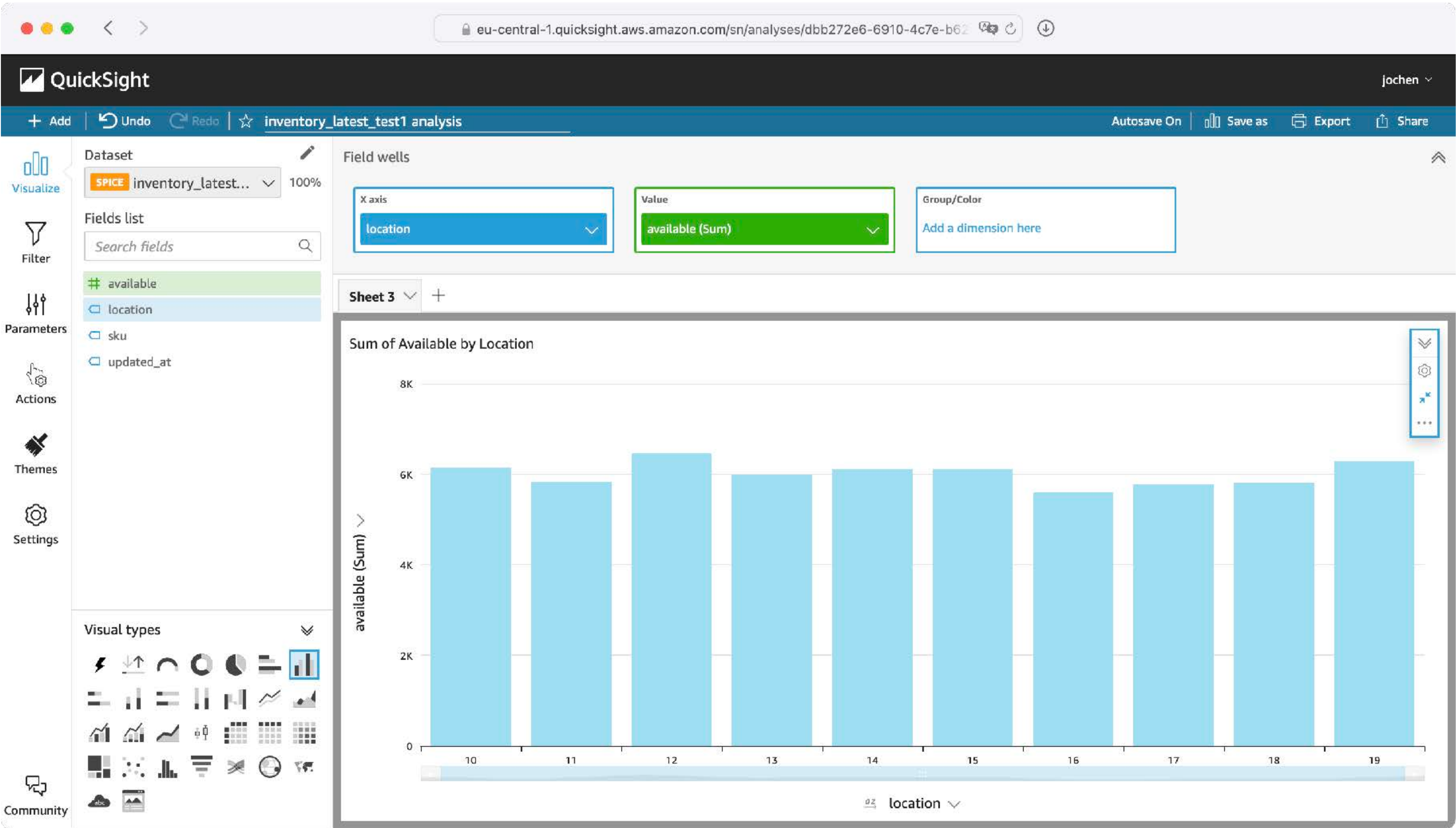
1

...

>

⚙

	Principal ▲	Principal type ▼	Resource type ▼	Database ▼	Table ▼	Resource ▼	Catalog ▼	LF-tag expression
☐	IAMAllowedPrincipals	Group	Table	fulfillment	inventory_latest_test1	inventory_latest_test1	528115139298	-
☐	jochen	IAM user	Table	fulfillment	inventory_latest_test1	inventory_latest_test1	528115139298	-
☐	jochen	IAM user	Column	fulfillment	inventory_latest_test1	Included: All	528115139298	-



Data Mesh Architecture with AWS S3 and Athena

