SOLUTIONS

TECHNOLOGY

Data Strategy **Framework**

<<YOUR COMPANY>> realizes its Business Goals with the following Use Cases…

UC 1

UC 2

UC 3

DATA PRODUCTS

… which need the following Data Objects …

Object 1

Object 2

Object 3

…Created in the following processes and tools…

… having the following Properties and Issues…

No Access

Low DQ

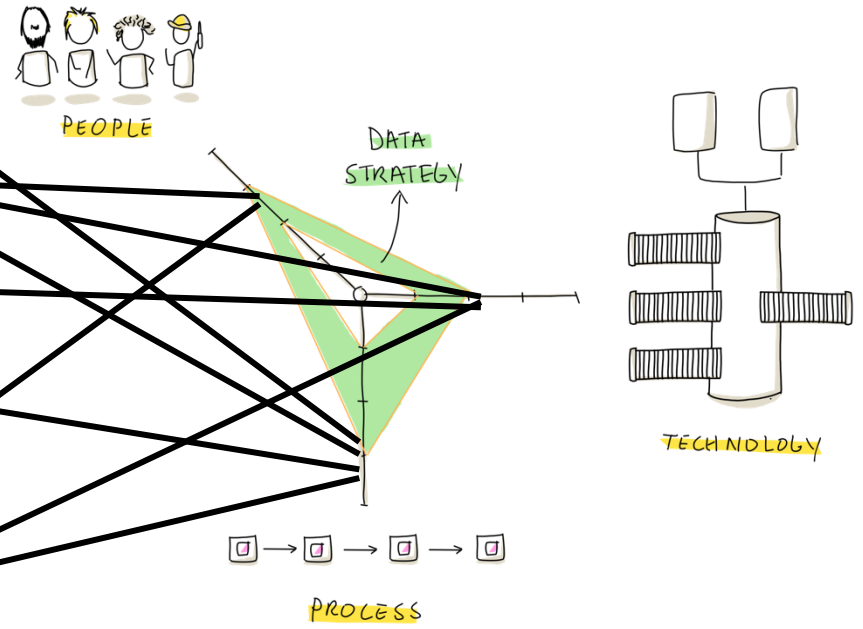
High Volume

No Definition

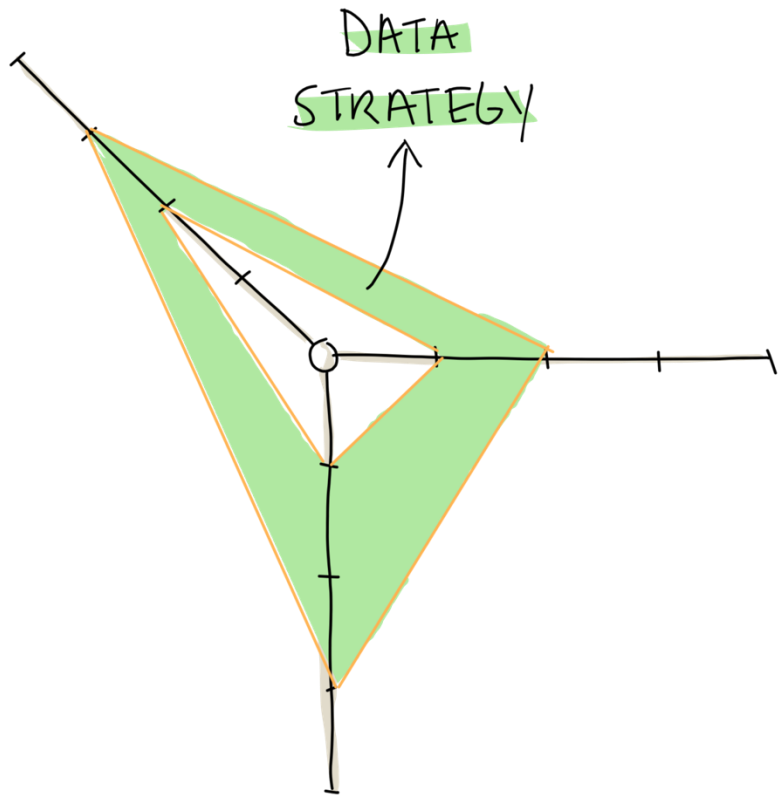
No Ownership

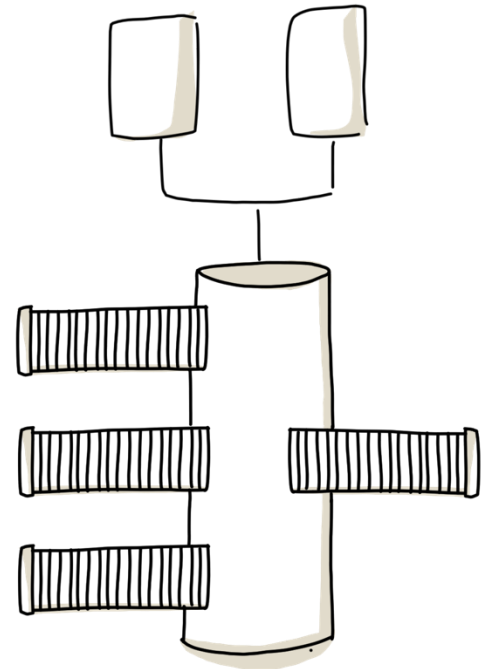
Not integrated

… which can be solved by PEOPLE, PROCESS & TECHNOLOGY INITIATIVES…

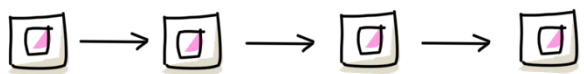


PEOPLE





TECHNOLOGY



PROCESS

Table of Contents

Table of Contents

BEFORE WE DIVE INTO TECHNOLOGY

DATA PLATFORM projects can become LONG RUNNING & EXPENSIVE projects

BE AWARE OF DEAD HORSES

Source: Oscar Berg

THE DEAD HORSE THEORY

ILLUSTRATED

Source: Oscar Berg

Author of:

Illustrations are part of his new book (to be announced)

The tribal wisdom of the Indians, passed on from generation to generation, says that, “When you discover that you are riding a dead horse, the best strategy is to dismount.”

Source: Oscar Berg

The Dead Horse Theory
goes on to say
that in modern business,
education and government,
far more advanced
strategies are often
employed, such as:

Source: Oscar Berg

1. Buying a stronger whip.

Source: Oscar Berg

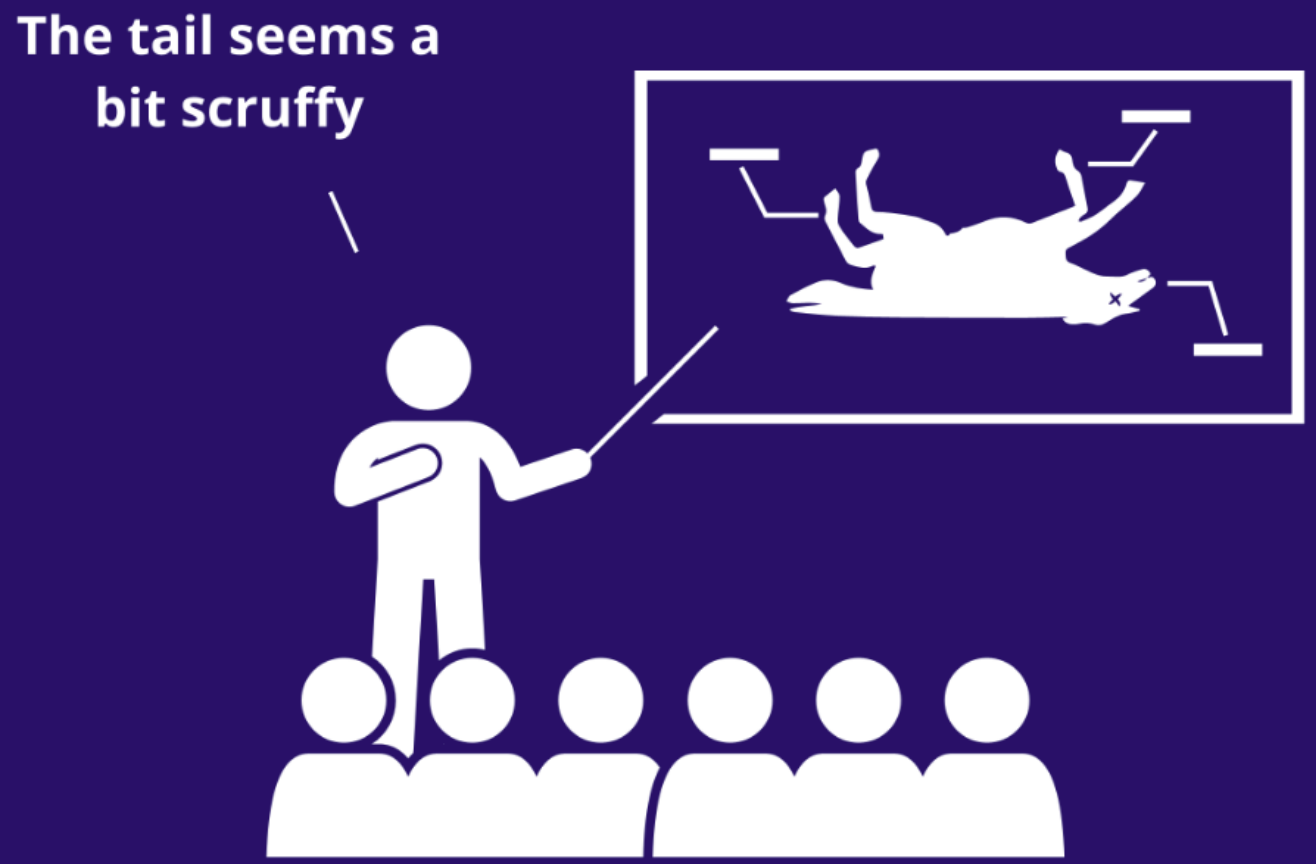
2. Changing riders.

Source: Oscar Berg

3. Threatening the horse with termination.

Source: Oscar Berg

4. Appointing a committee to study the horse.



Source: Oscar Berg

5. Arranging to visit other countries to see how others ride dead horses.

Source: Oscar Berg

6. Lowering the standards so that dead horses can be included.

Source: Oscar Berg

7. Re-classifying the dead horse as ‘living-impaired’.

Source: Oscar Berg

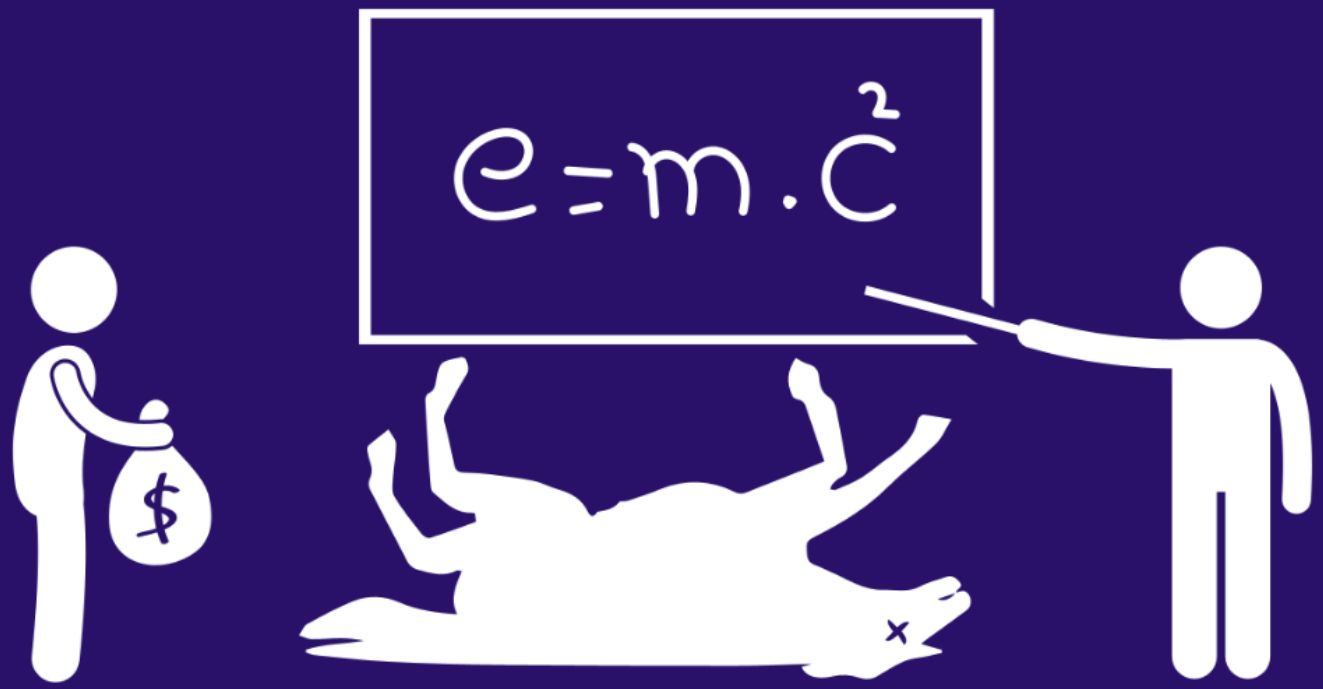
8. Hiring outside contractors to ride the dead horse.

Source: Oscar Berg

9. Harnessing several dead horses together to increase the speed.

Source: Oscar Berg

10. Providing additional funding and/or training to increase the dead horse's performance.



Source: Oscar Berg

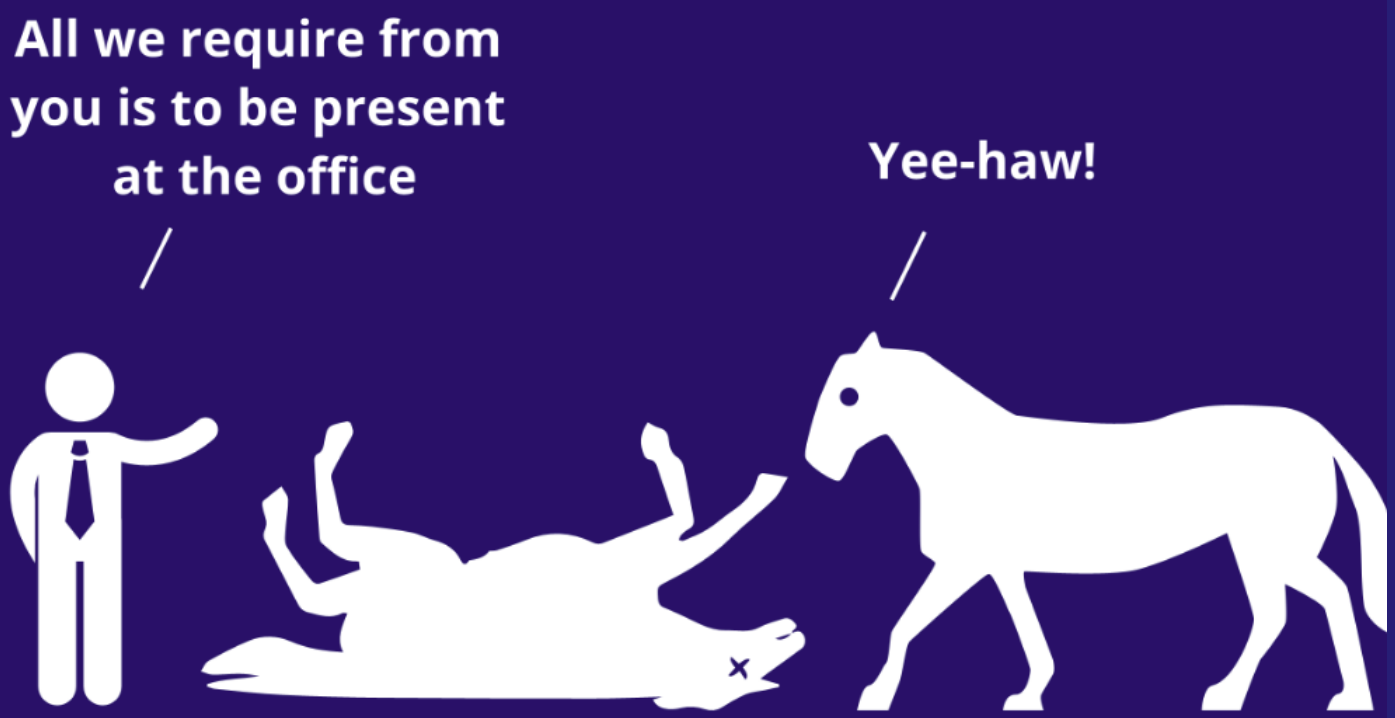
11. Doing a productivity study to see if lighter riders would improve the dead horse's performance.

Source: Oscar Berg

12. Declaring that as the dead horse does not have to be fed, it is less costly, carries lower overhead and, therefore, contributes substantially more to the bottom line of the economy than do some other horses.

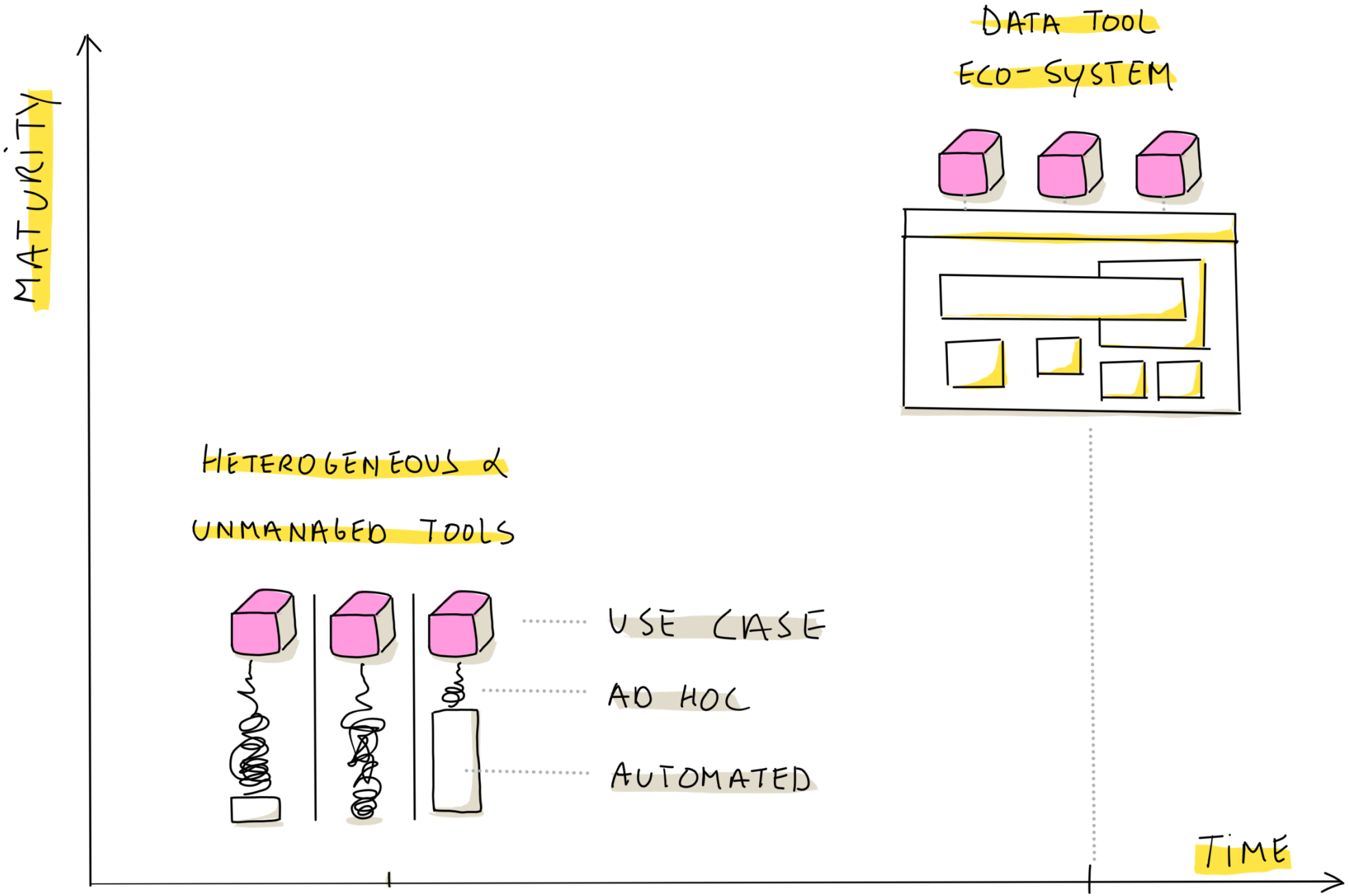
Source: Oscar Berg

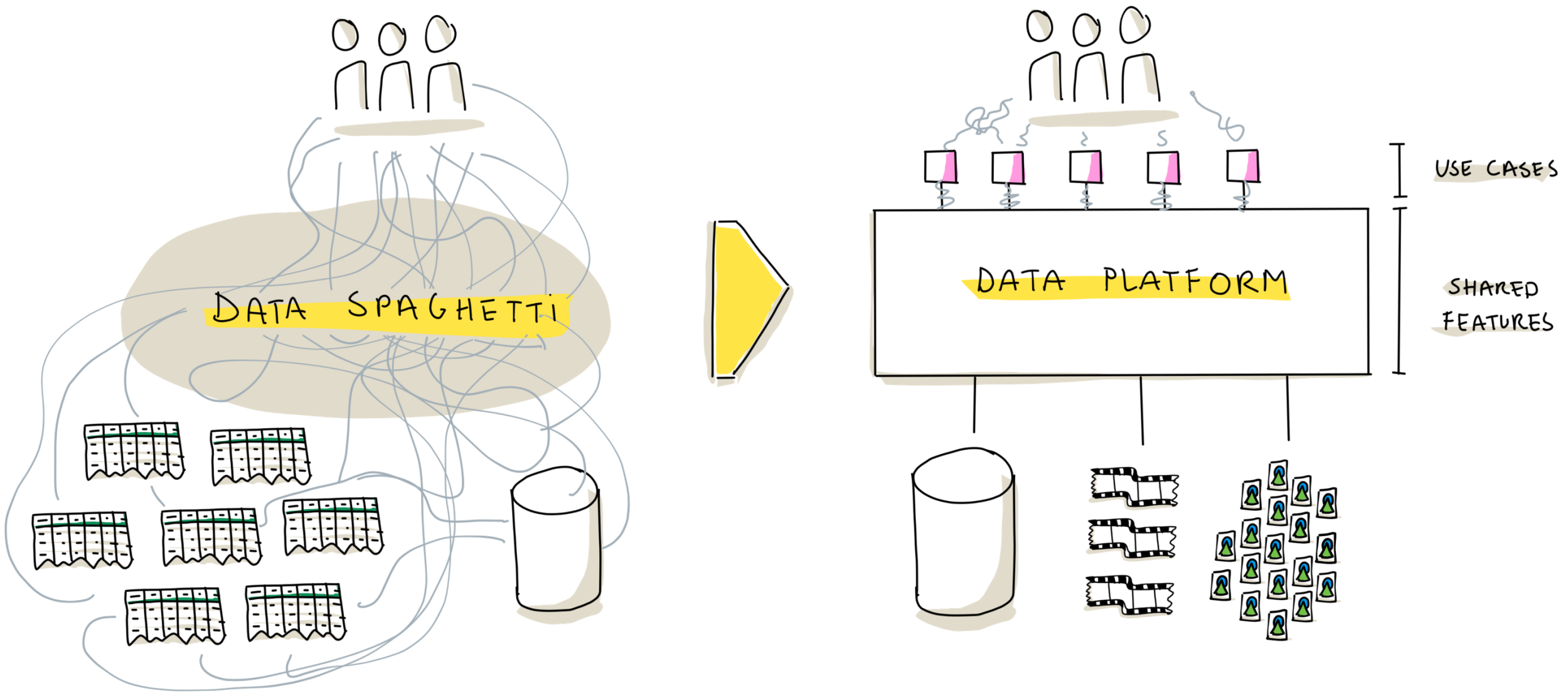
13. Re-writing the expected performance requirements for all horses.

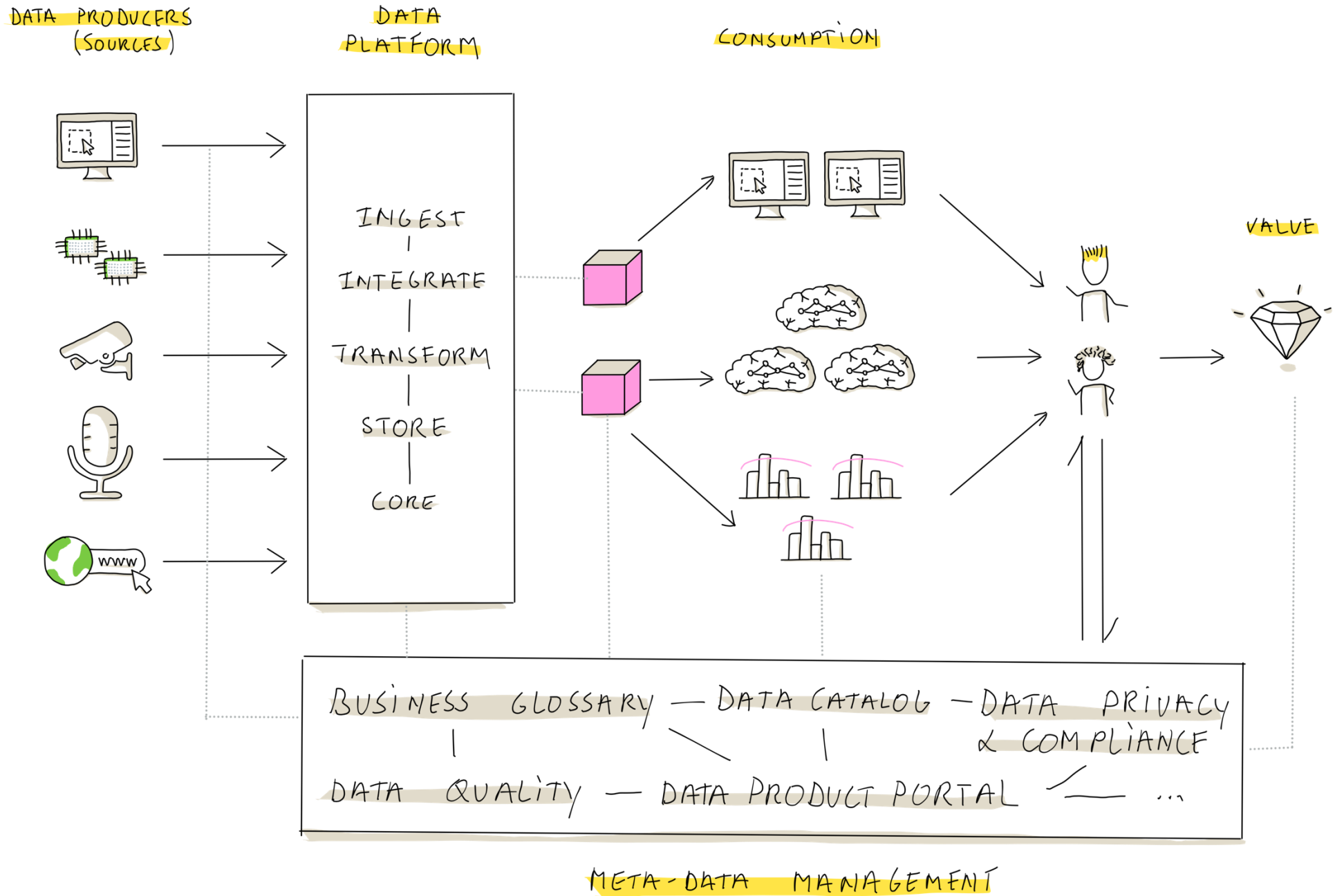


Source: Oscar Berg

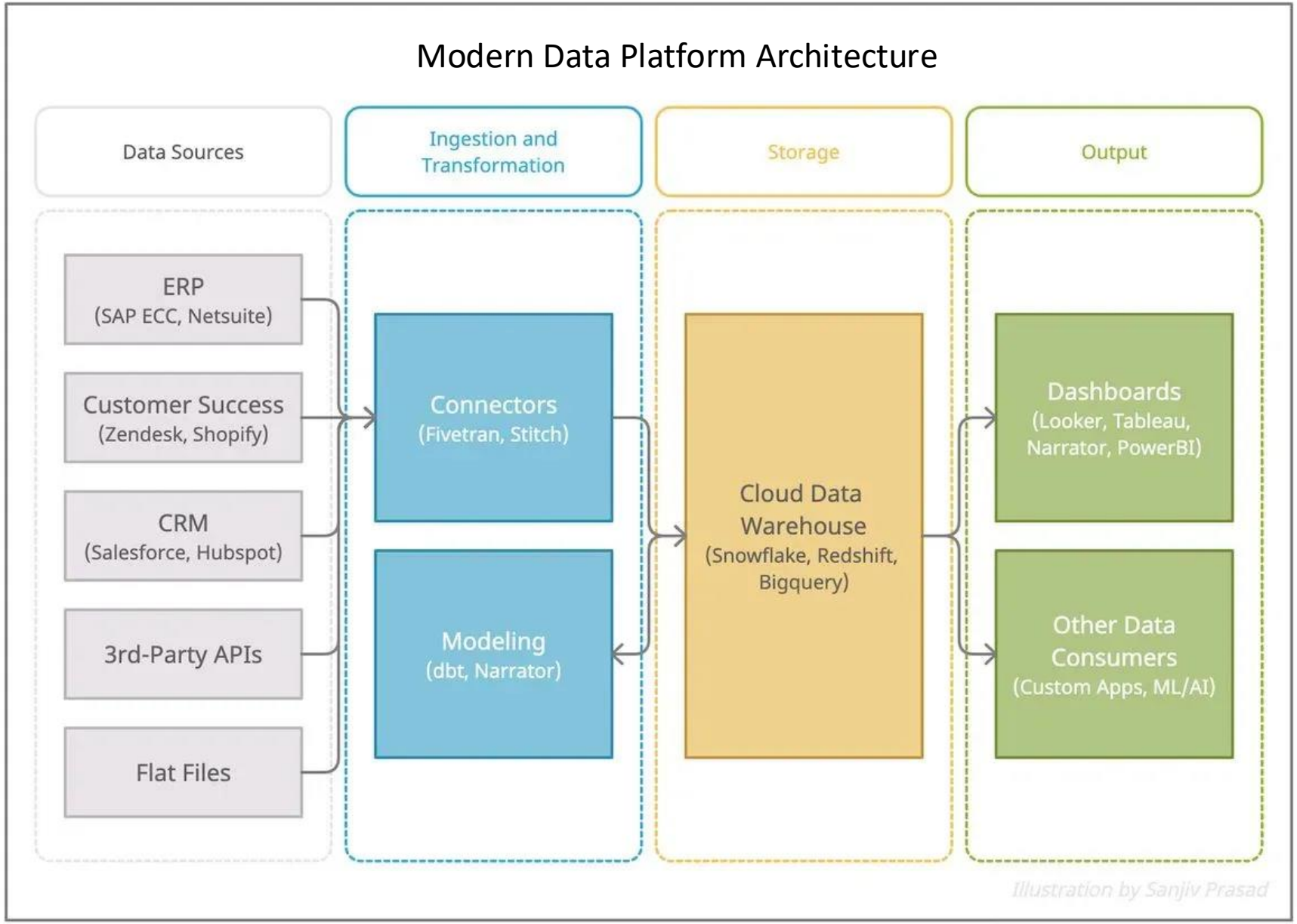
Table of Contents

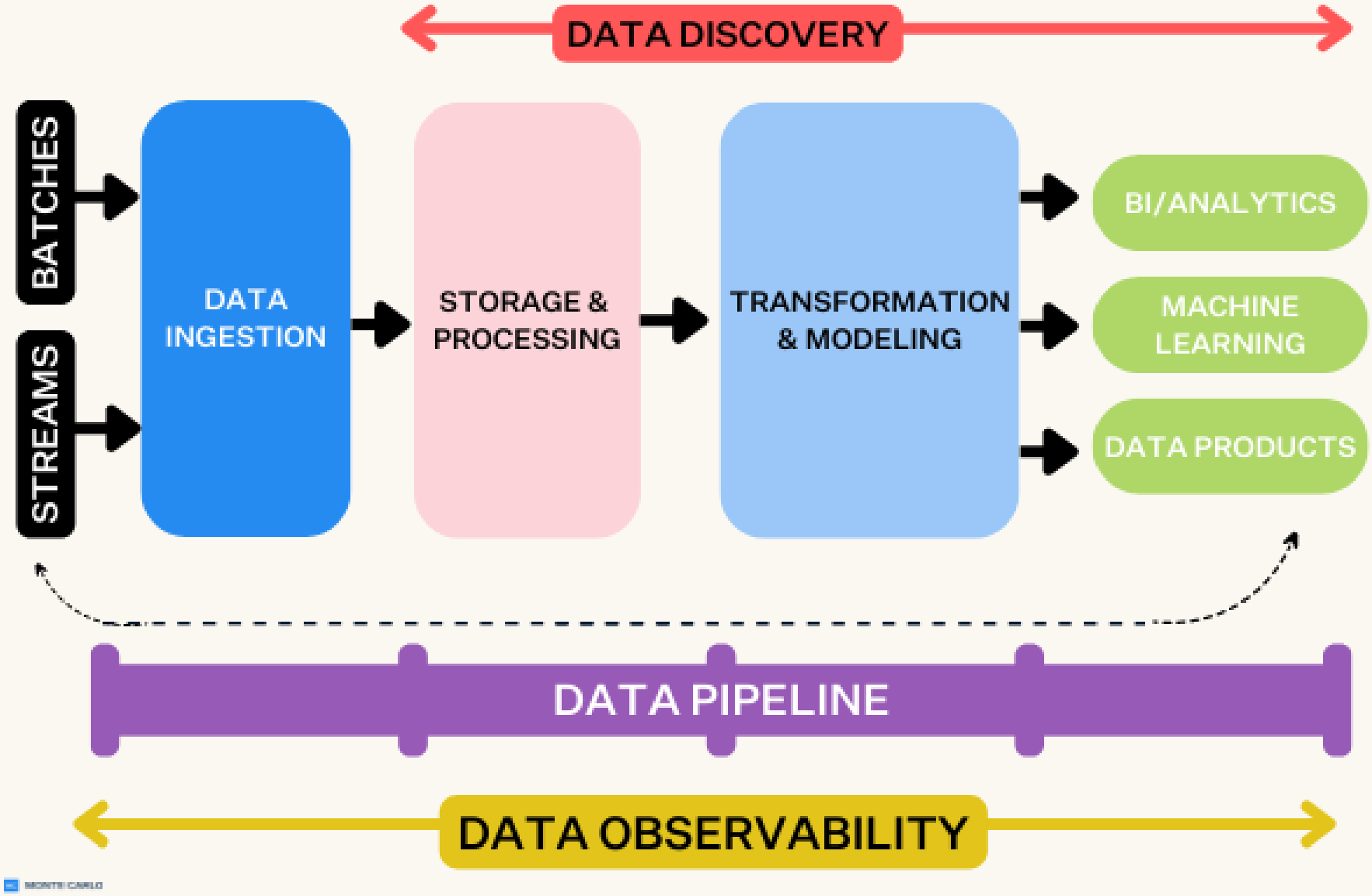

MATURITY
DATA TOOL
ECO-SYSTEM
HETEROGENEOUS &
UNMANAGED TOOLS
USE CASE
AD HOC
AUTOMATED
TIME


DATA SPAGHETTI
DATA PLATFORM
USE CASES
SHARED FEATURES

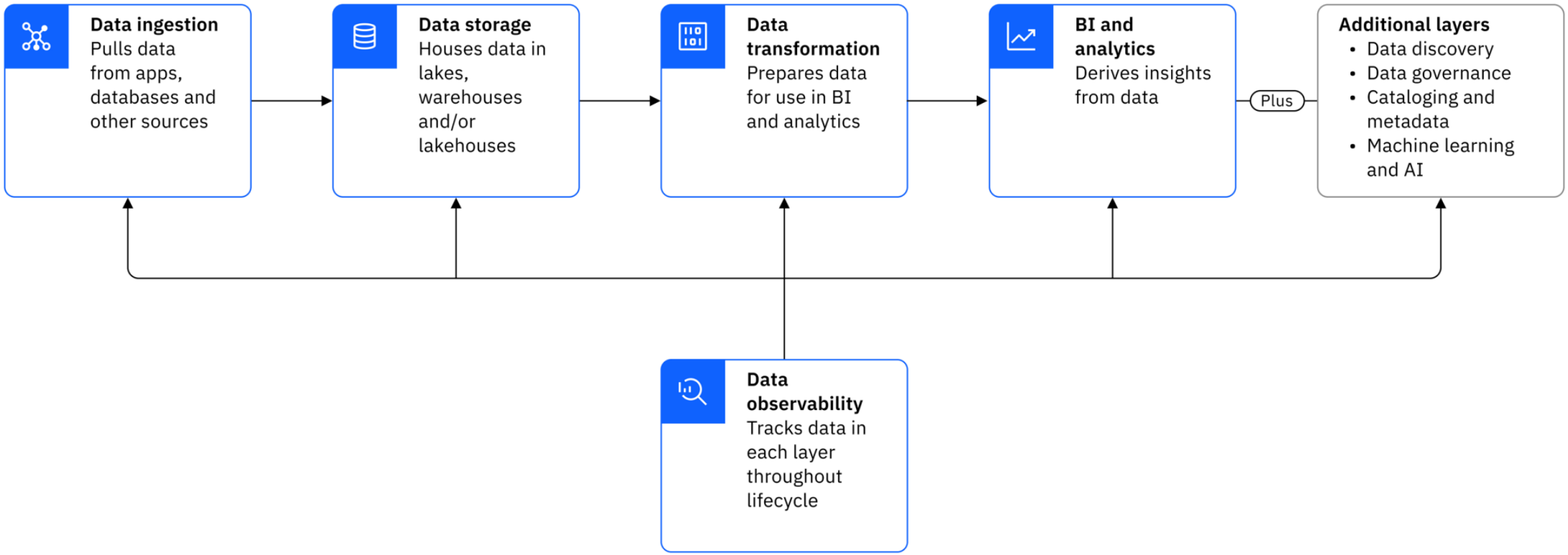

DATA PRODUCERS (SOURCES)
DATA PLATFORM
CONSUMPTION
VALUE
INGEST
INTEGRATE
TRANSFORM
STORE
CORE
www
BUSINESS GLOSSARY
DATA CATALOG
DATA PRIVACY & COMPLIANCE
DATA QUALITY
DATA PRODUCT PORTAL
...
META-DATA MANAGEMENT

Modern Data Platform Architecture

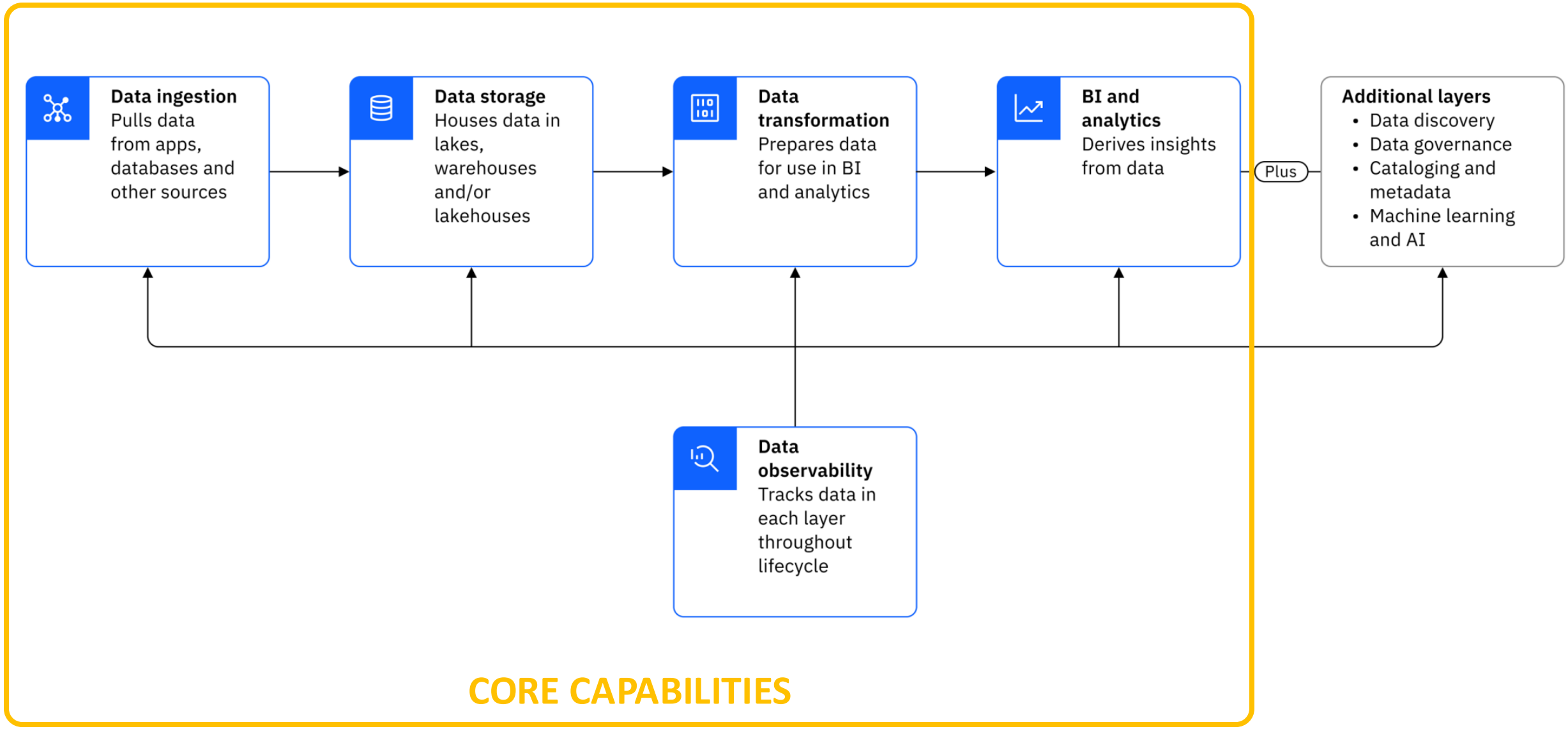




[Montecarlo – What is a Data Platform and how to Build One?]



[IBM: Data Platforms]



[IBM: Data Platforms]

[Atlan: What is a Data Platform]

[Atlan: What is a Data Platform]

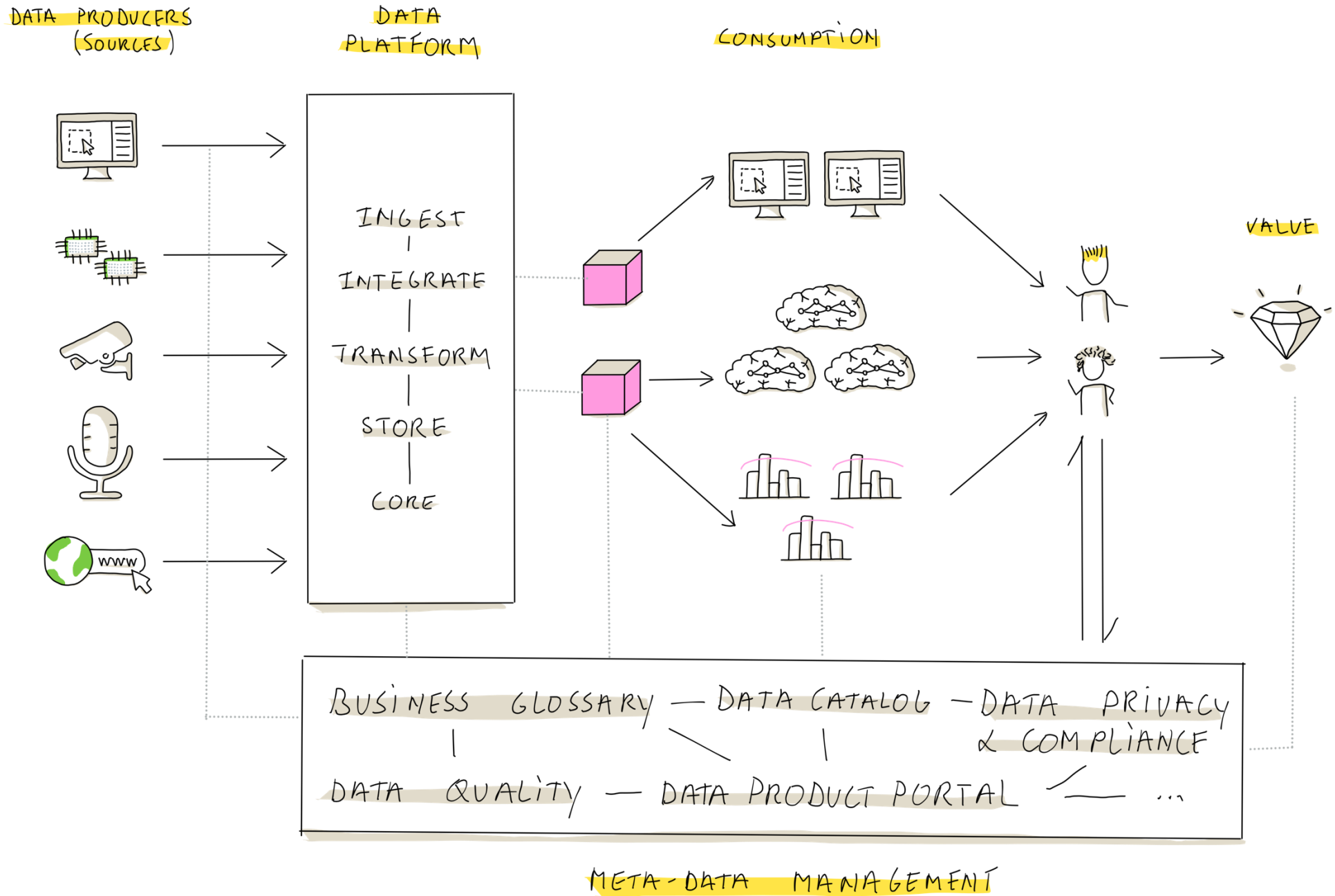
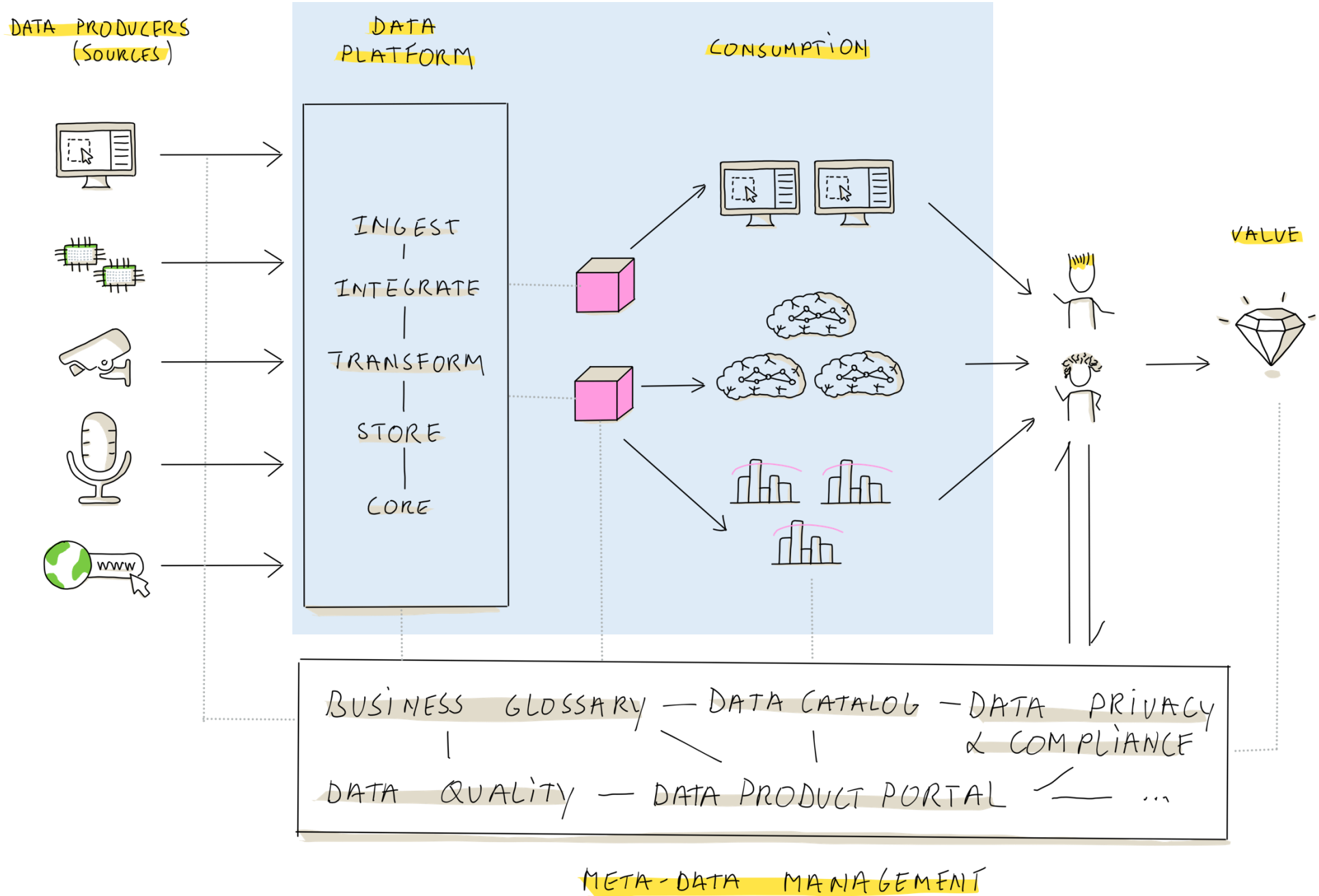
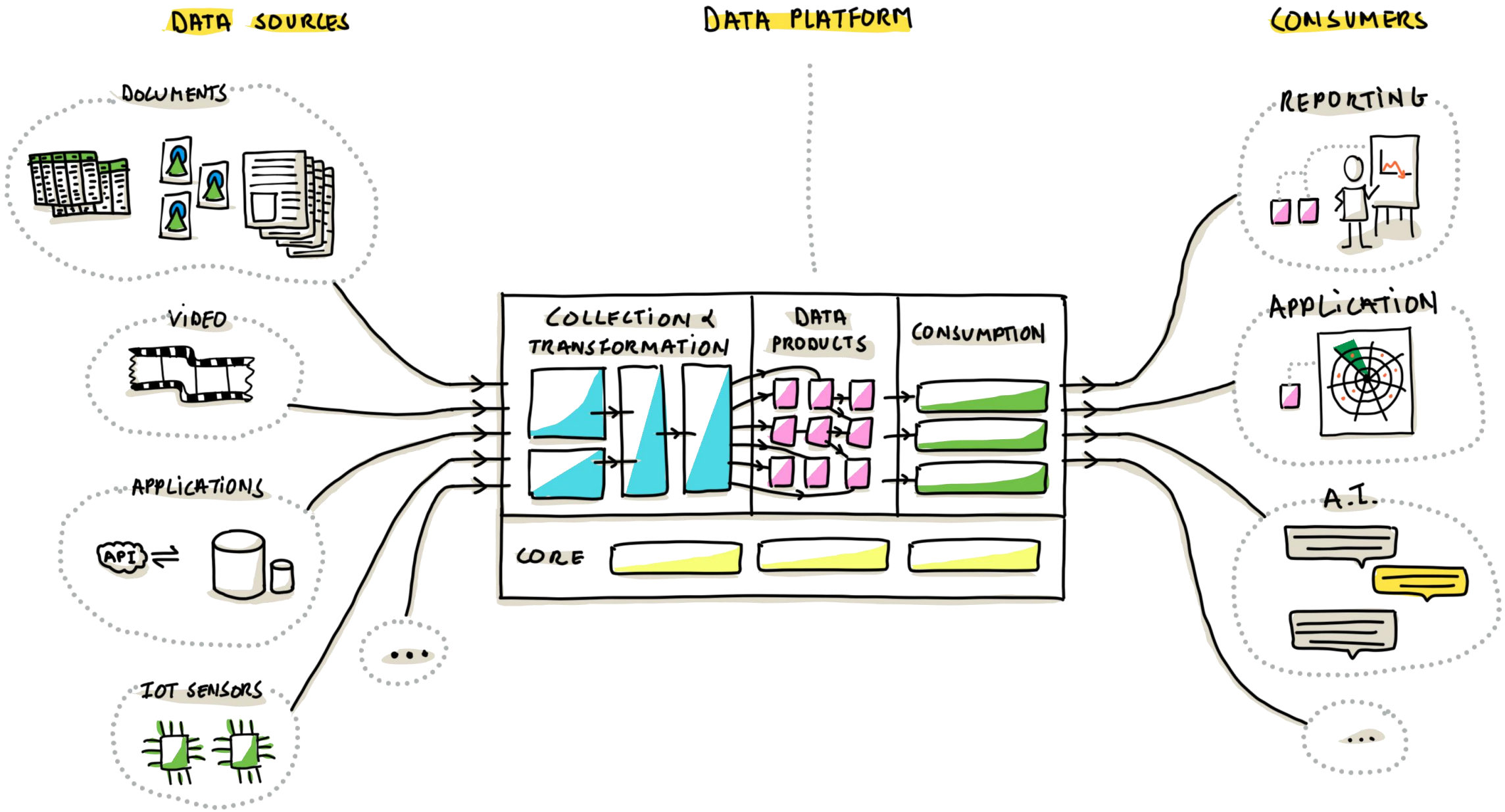
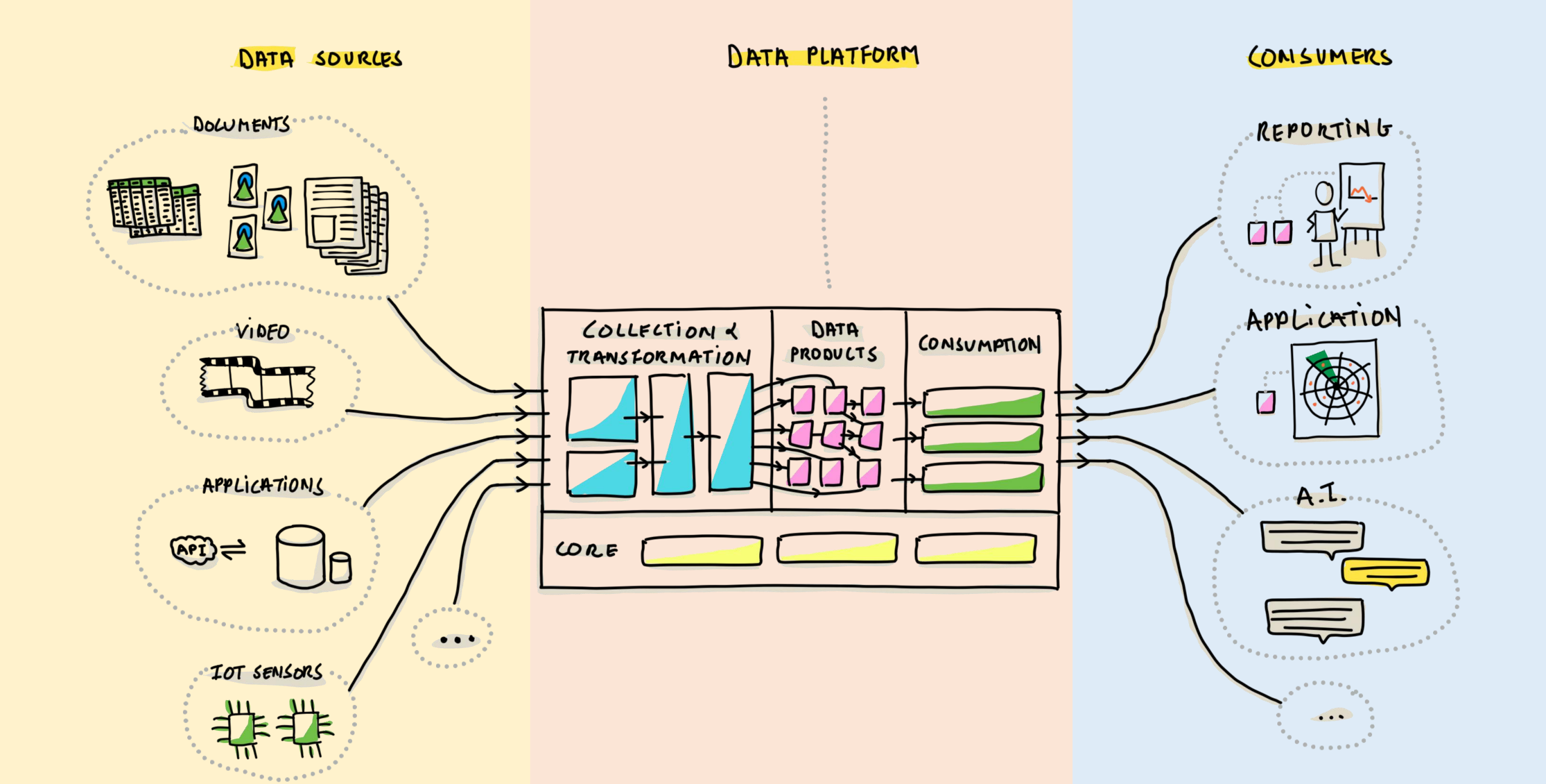

DATA PRODUCERS (SOURCES)
DATA PLATFORM
CONSUMPTION
VALUE
INGEST
INTEGRATE
TRANSFORM
STORE
CORE
www
BUSINESS GLOSSARY
DATA CATALOG
DATA PRIVACY & COMPLIANCE
DATA QUALITY
DATA PRODUCT PORTAL
...
META-DATA MANAGEMENT

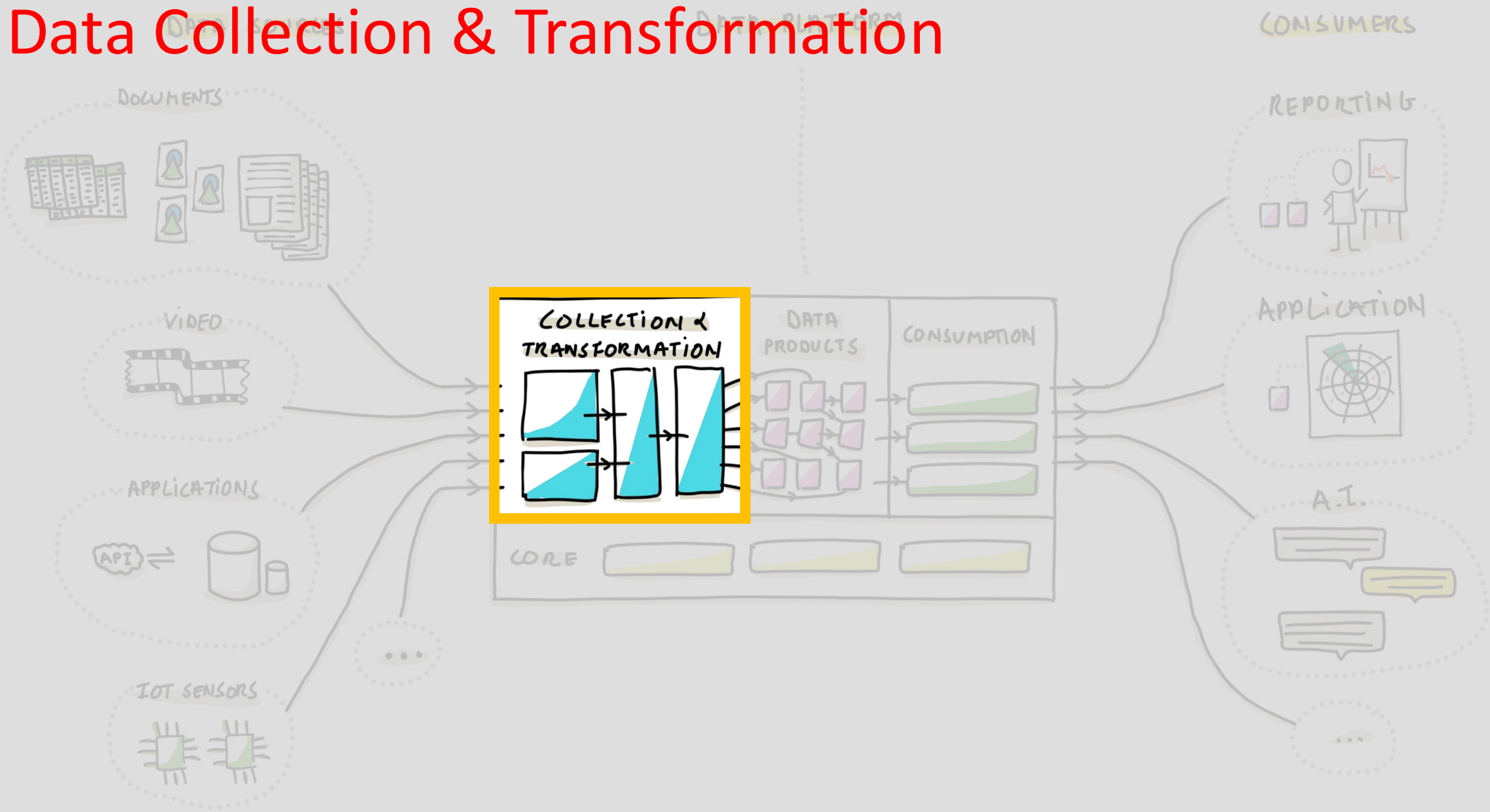
Table of Contents


DATA PRODUCERS (SOURCES)
DATA PLATFORM
CONSUMPTION
INGEST
INTEGRATE
TRANSFORM
STORE
CORE
www
VALUE
BUSINESS GLOSSARY
DATA CATALOG
DATA PRIVACY & COMPLIANCE
DATA QUALITY
DATA PRODUCT PORTAL
...
META-DATA MANAGEMENT


DATA SOURCES
DATA PLATFORM
CONSUMERS
DOCUMENTS
VIDEO
APPLICATIONS
API
IOT SENSORS
...
COLLECTION & TRANSFORMATION
DATA PRODUCTS
CONSUMPTION
CORE
REPORTING
APPLICATION
A.I.
...


DATA SOURCES
DATA PLATFORM
CONSUMERS
DOCUMENTS
VIDEO
APPLICATIONS
API
IOT SENSORS
COLLECTION & TRANSFORMATION
DATA PRODUCTS
CONSUMPTION
CORE
REPORTING
APPLICATION
A.I.
"Operational Plane"
"Analytical Plane"
"Operational / Analytical Plane"

1. Data Collection & Transformation



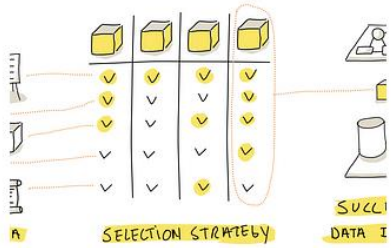
"Operational Plane" "Analytical Plane" "Operational / Analytical Plane"

See also my Medium Blog

janmeskens in The Modern Scientist

Data Ingestion—Part 2: Tool Selection Strategy

This article is the second one in my series on data ingestion. For an introduction to the topic and to explore ‘data ingestion...

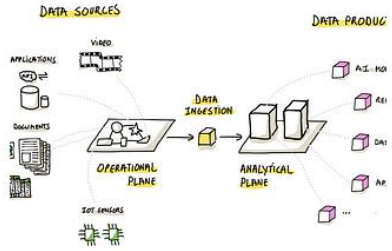


Jan 17 715 5

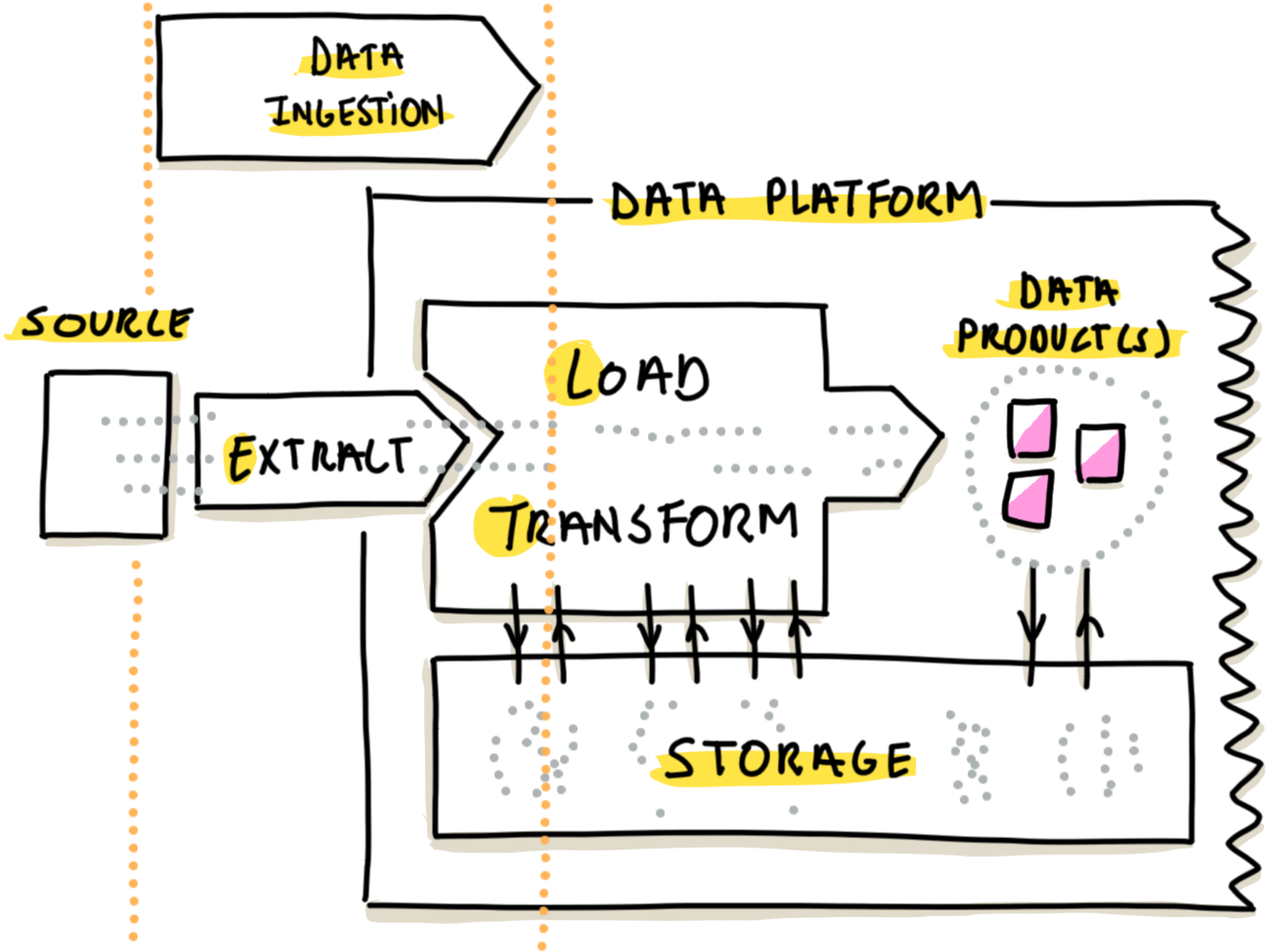
janmeskens in The Modern Scientist

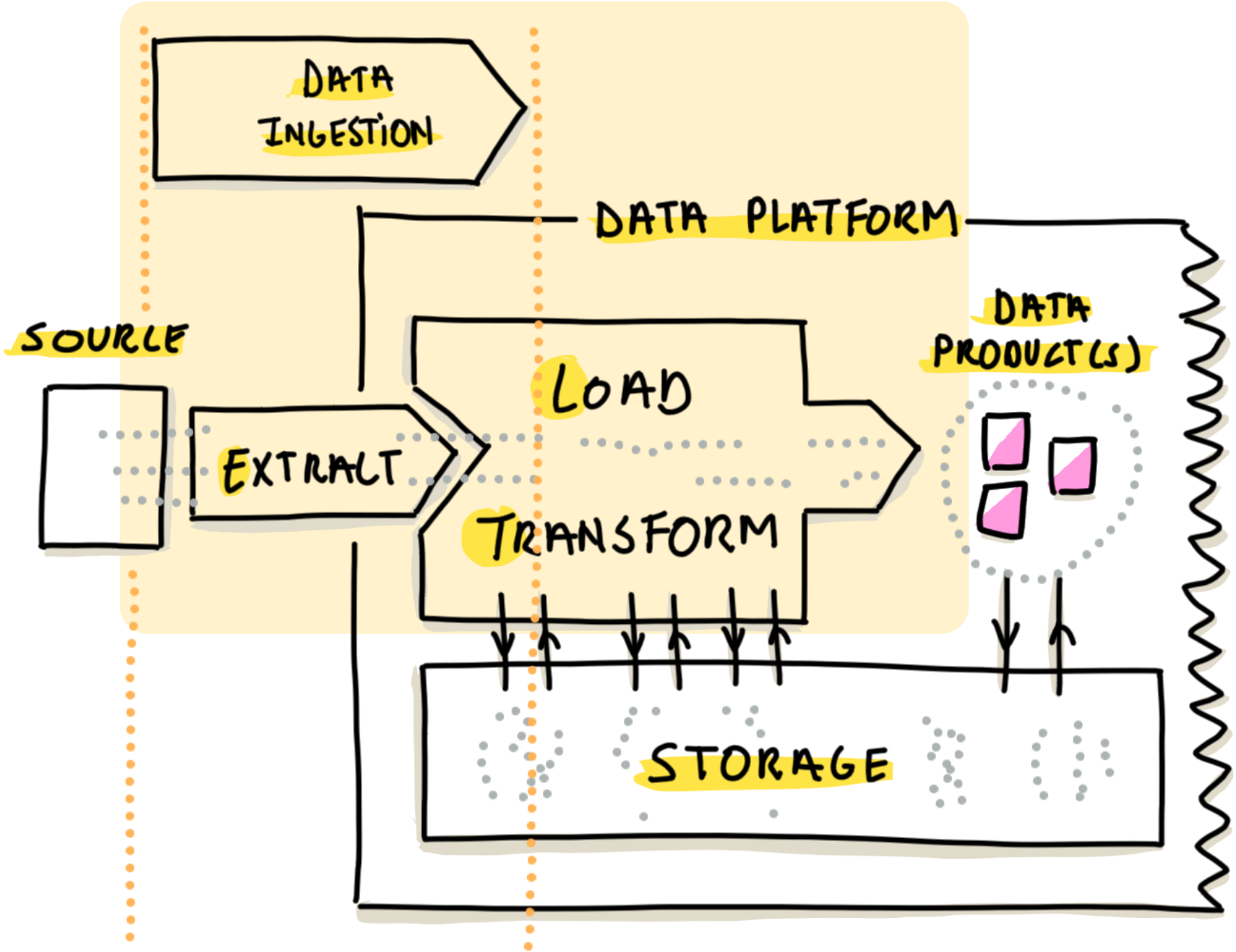
Data Ingestion — Part 1: Architectural Patterns

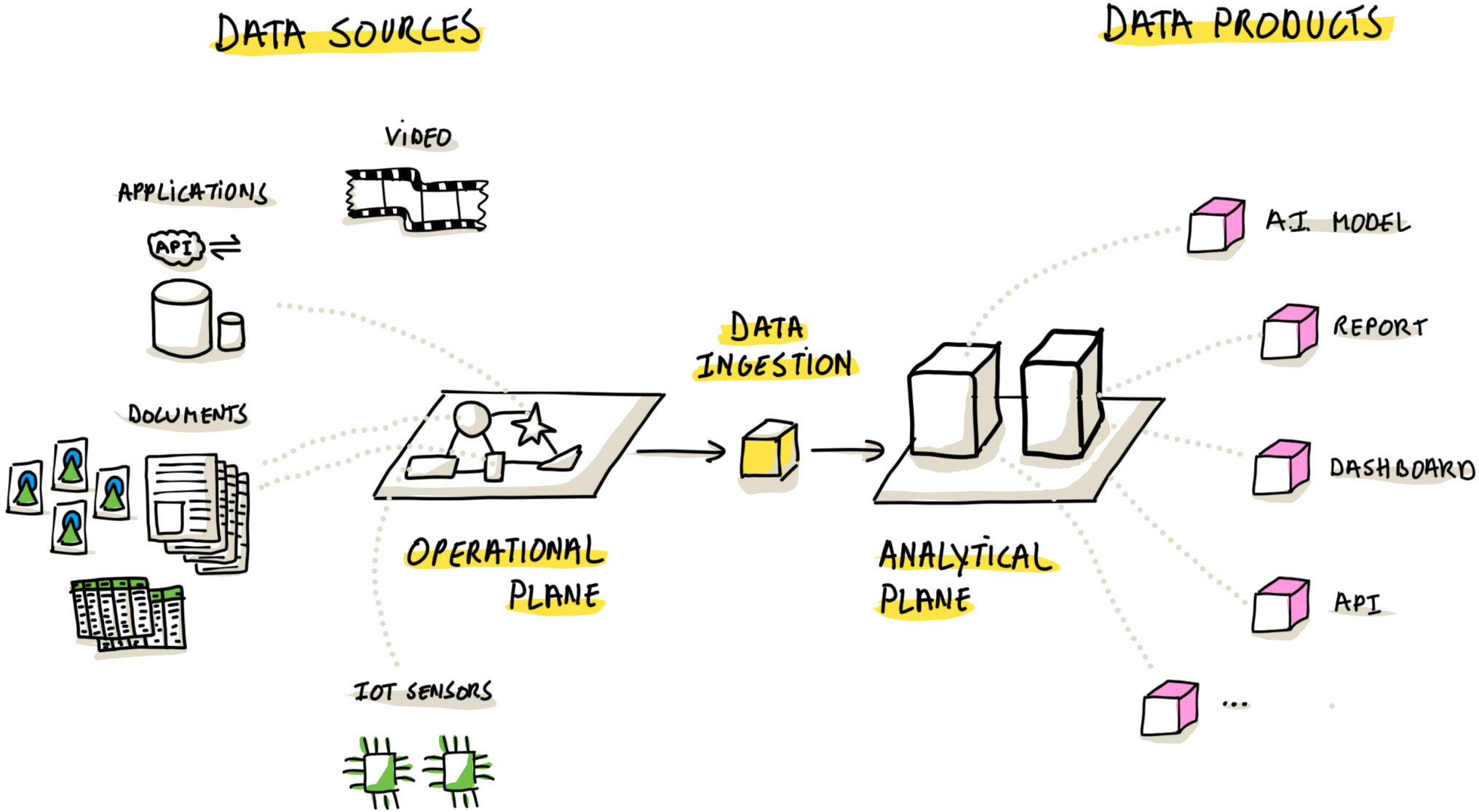
Over the course of two articles, I will thoroughly explore data ingestion, a fundamental process that bridges the operational...

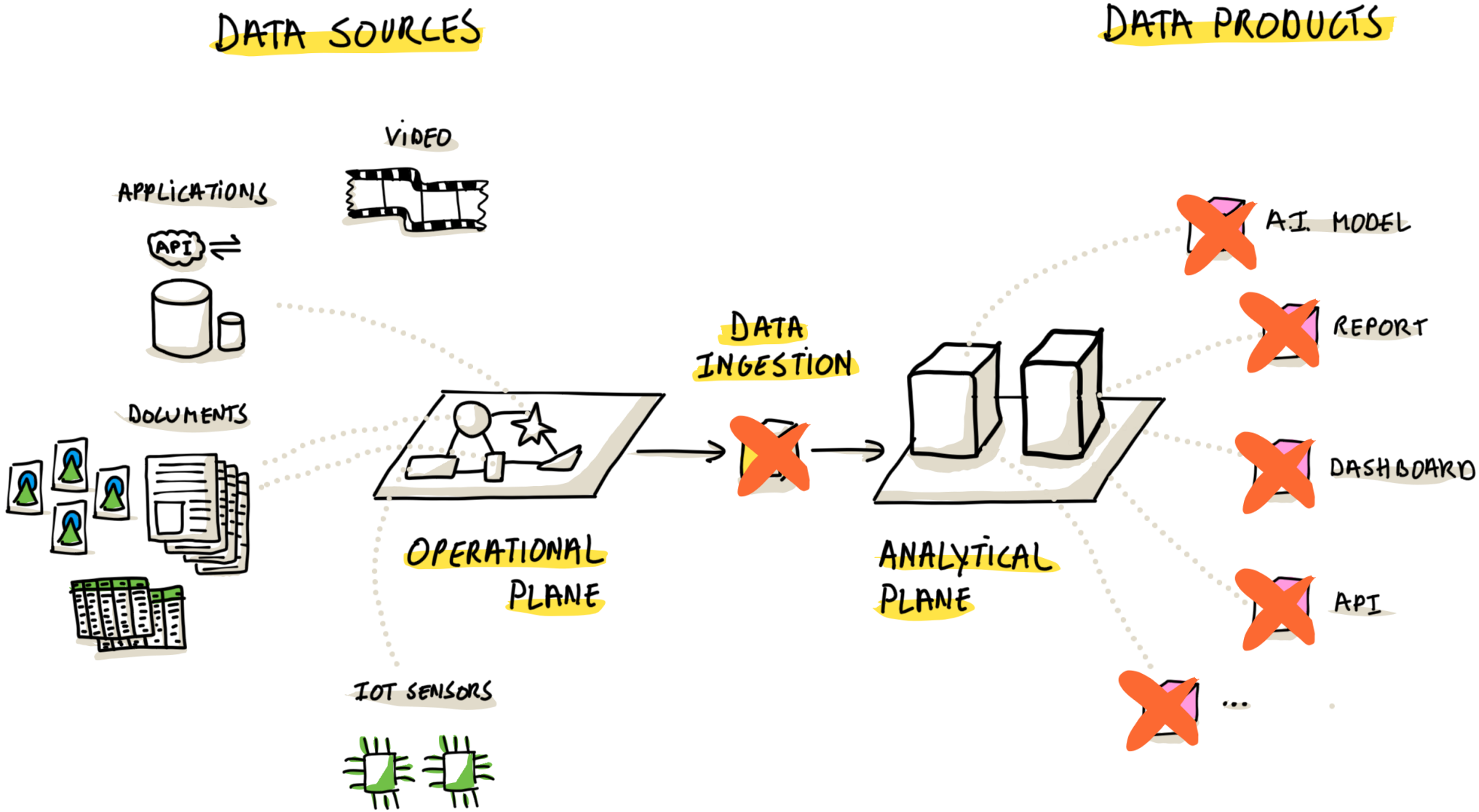


Nov 27, 2023 2.7K 26


DATA
INGESTION
DATA PLATFORM
SOURCE
EXTRACT
LOAD
TRANSFORM
DATA
PRODUCT(S)
STORAGE


DATA
INGESTION
DATA PLATFORM
SOURCE
EXTRACT
LOAD
TRANSFORM
DATA
PRODUCT(S)
STORAGE


DATA SOURCES
DATA PRODUCTS
VIDEO
APPLICATIONS
API
DOCUMENTS
IOT SENSORS
DATA INGESTION
OPERATIONAL PLANE
ANALYTICAL PLANE
A.I. MODEL
REPORT
DASHBOARD
API
...


DATA SOURCES
DATA PRODUCTS
VIDEO
APPLICATIONS
API
DOCUMENTS
IOT SENSORS
DATA INGESTION
OPERATIONAL PLANE
ANALYTICAL PLANE
A.I. MODEL
REPORT
DASHBOARD
API
...

Definition

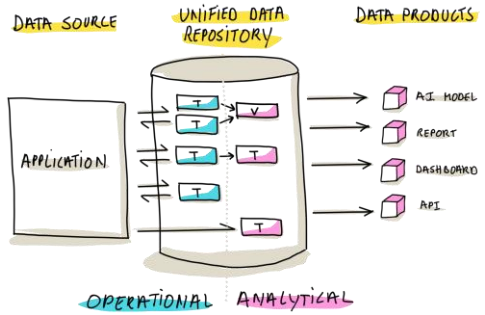
Data ingestion refers to the tools & processes used to **collect data from various sources and move it to a target site**, either in **batches or in real-time**. The data ingestion layer is **critical to your downstream** data science, BI, and analytics systems which depend on **timely, complete, and accurate data**.

[Qlik – Data Ingestion]

Data Ingestion Patterns

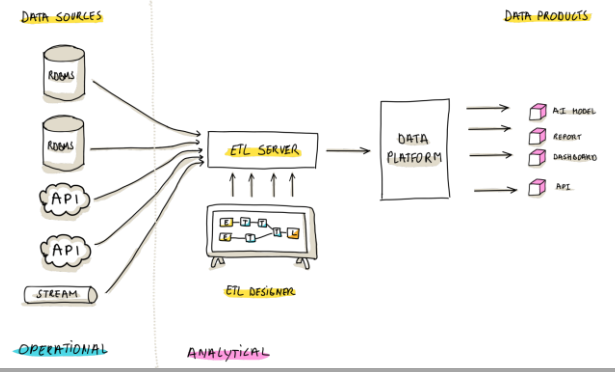
Pattern #1

UNIFIED DATA REPOSITORY



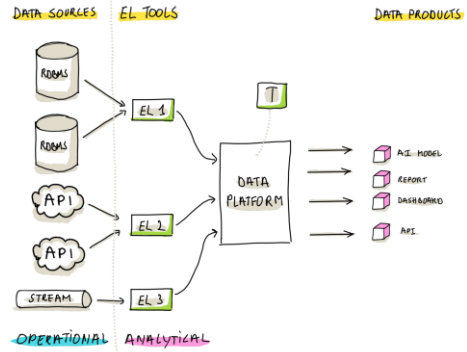
Pattern #2

ETL – Extract Transform Load



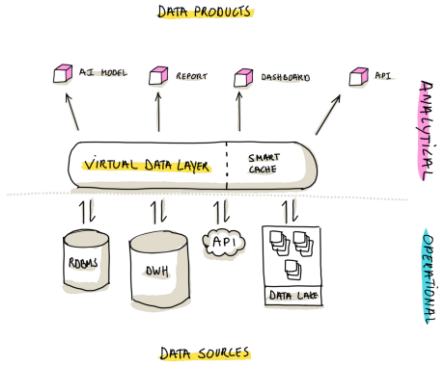
Pattern #3

ELT – Extract Transform Load



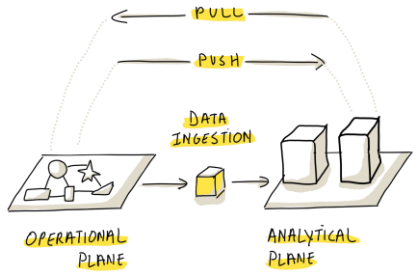
Pattern #4

Data Virtualization



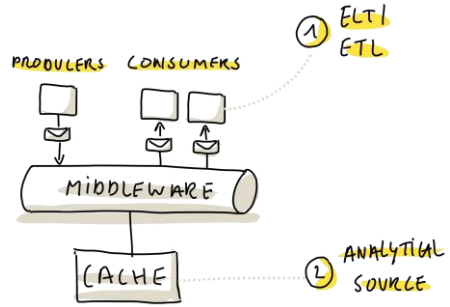
Pattern #5

Push vs Pull



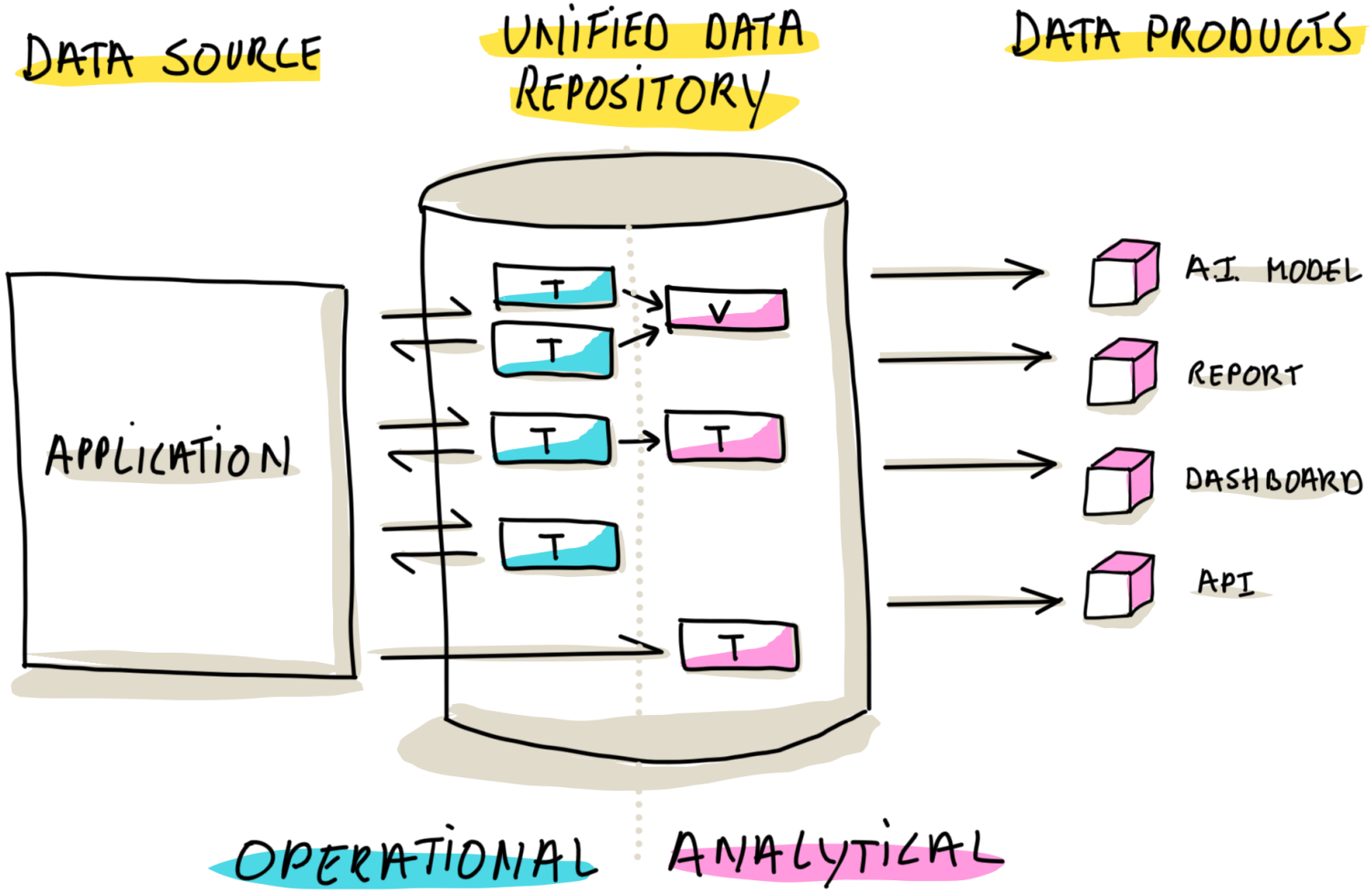
Pattern #6

Streaming Data



Pattern #1

UNIFIED DATA REPOSITORY



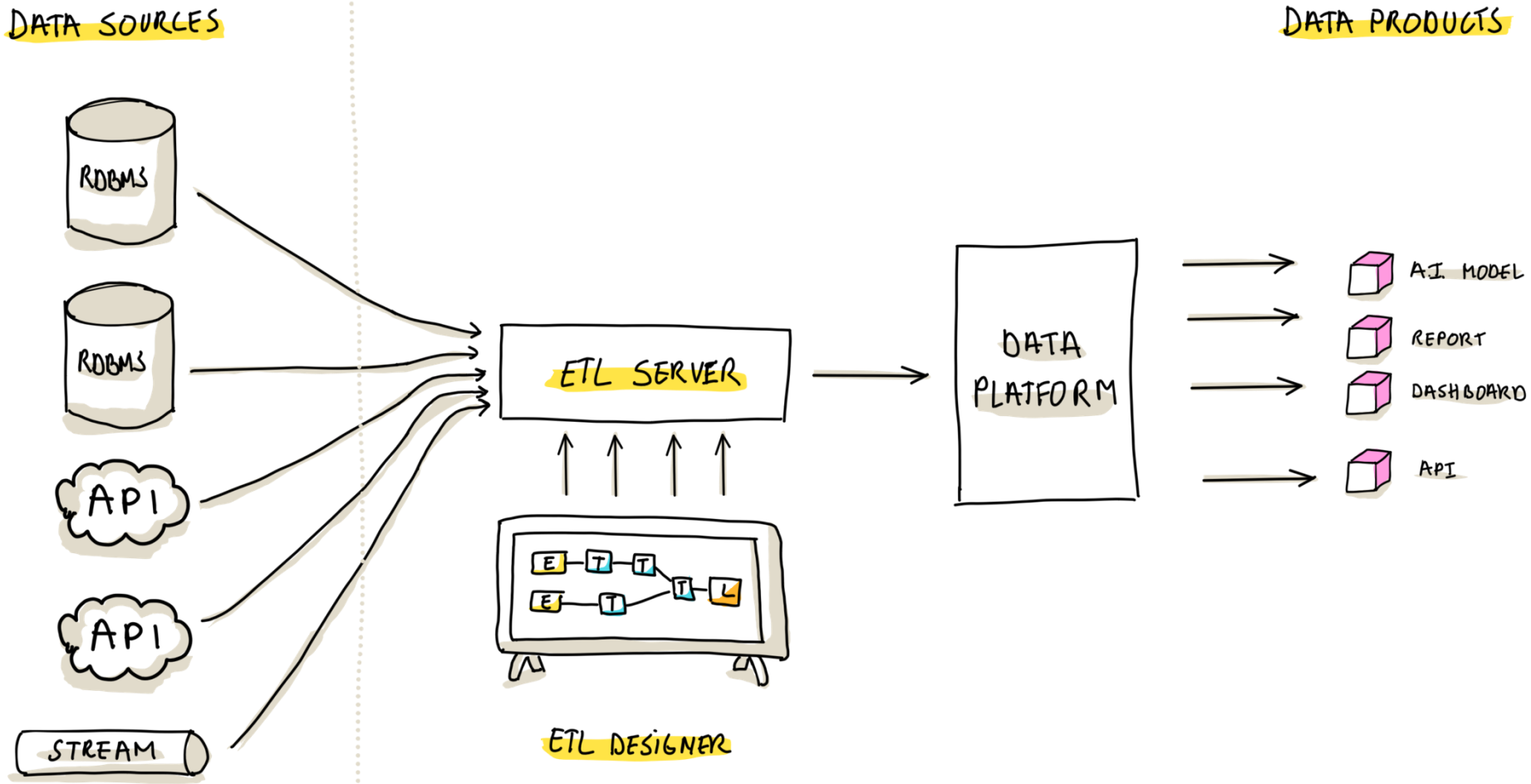
Unified Data Repository = A single storage system caters to both the operational application needs and analytical processing

Specificities:

- Typically a Relational Database Management System (RDBMS).
- The same database is utilized for both everyday operations and data analysis
- Two prevalent sub-patterns:
  1. **Virtualization**
  2. **Duplication and Transformation**

Pattern #2

ETL – **E**xtract **T**ransform **L**oad



ETL (Extract, Transform, Load) = *a well-established paradigm in data processing*

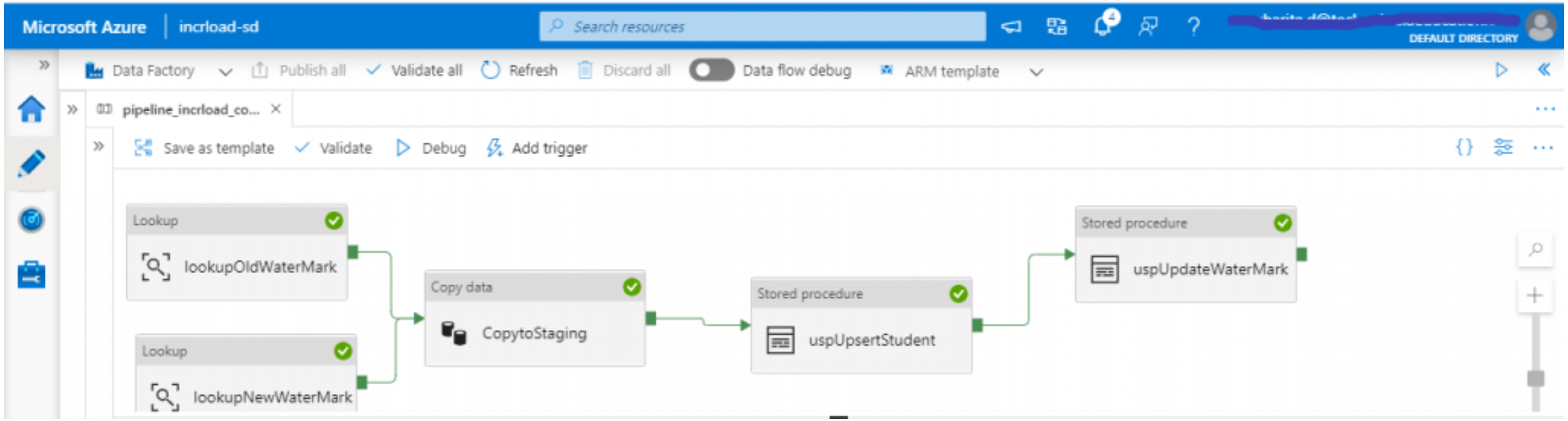
3 steps:

1. **Extract** : Data is harvested from its source
2. **Transform** : Refined on an ETL server
3. **Load** : The polished output is deposited into an analytics-focused database.

ETL tools have a graphical interface where users can interlink Extract, Transform, and Load operations within an intuitive visual workflow. These processes are often further customizable through scripting or direct SQL queries.

Pattern #2

ETL – Extract **T**ransform **L**oad

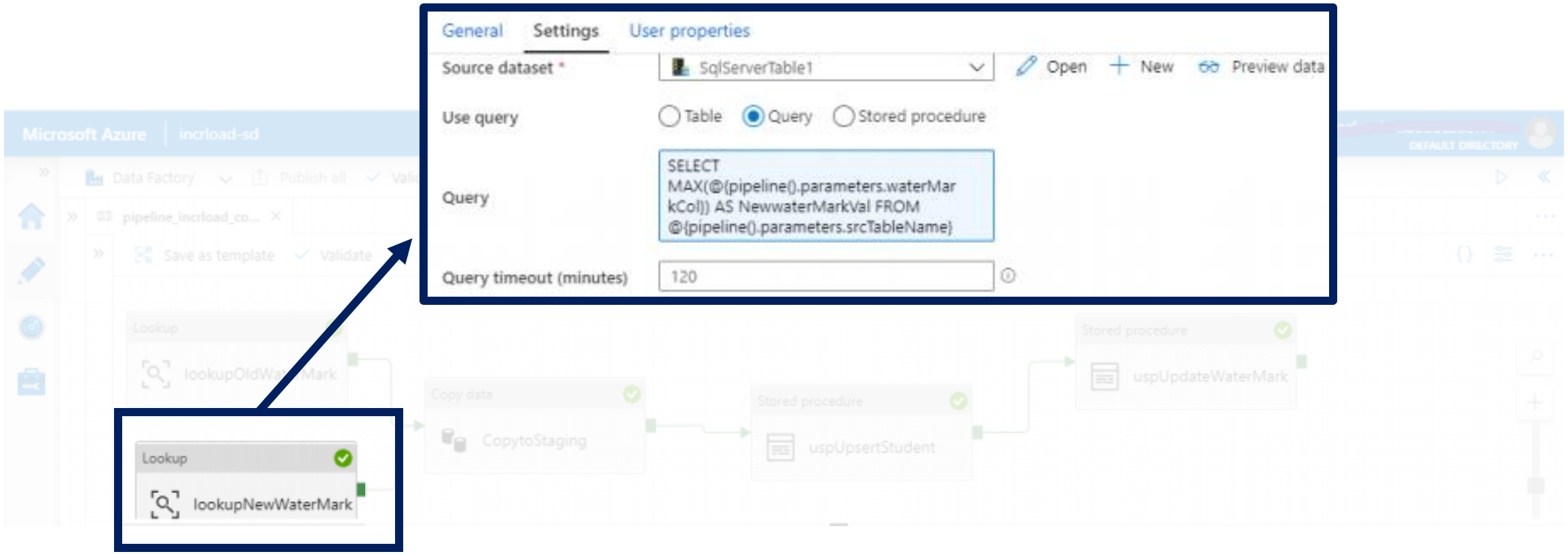


Extract Transform **Load**

[Data Loading using Azure Data Factory]

Pattern #2

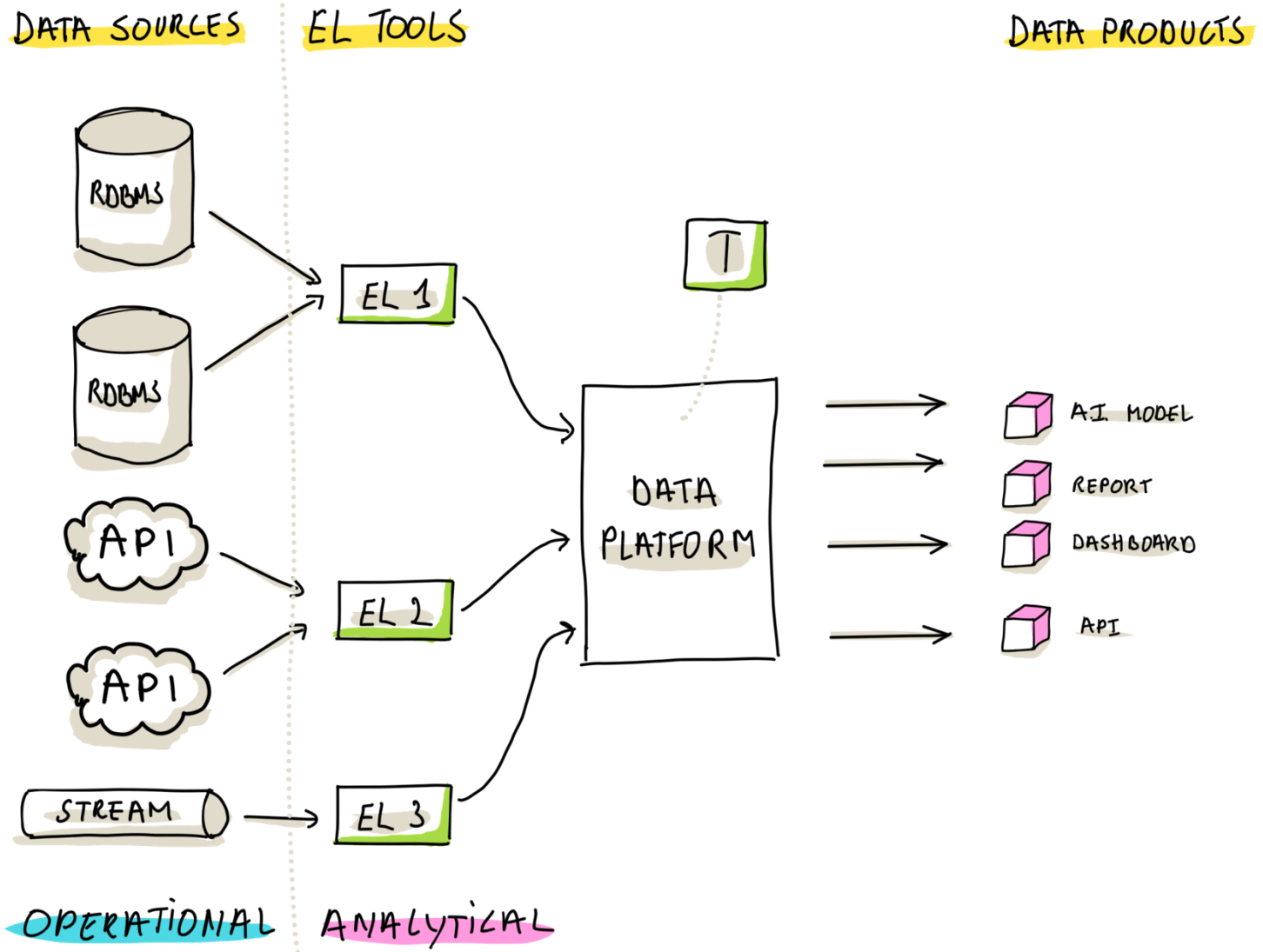
ETL – Extract Transform Load



[Data Loading using Azure Data Factory]

Pattern #3

ELT – Extract Load Transform



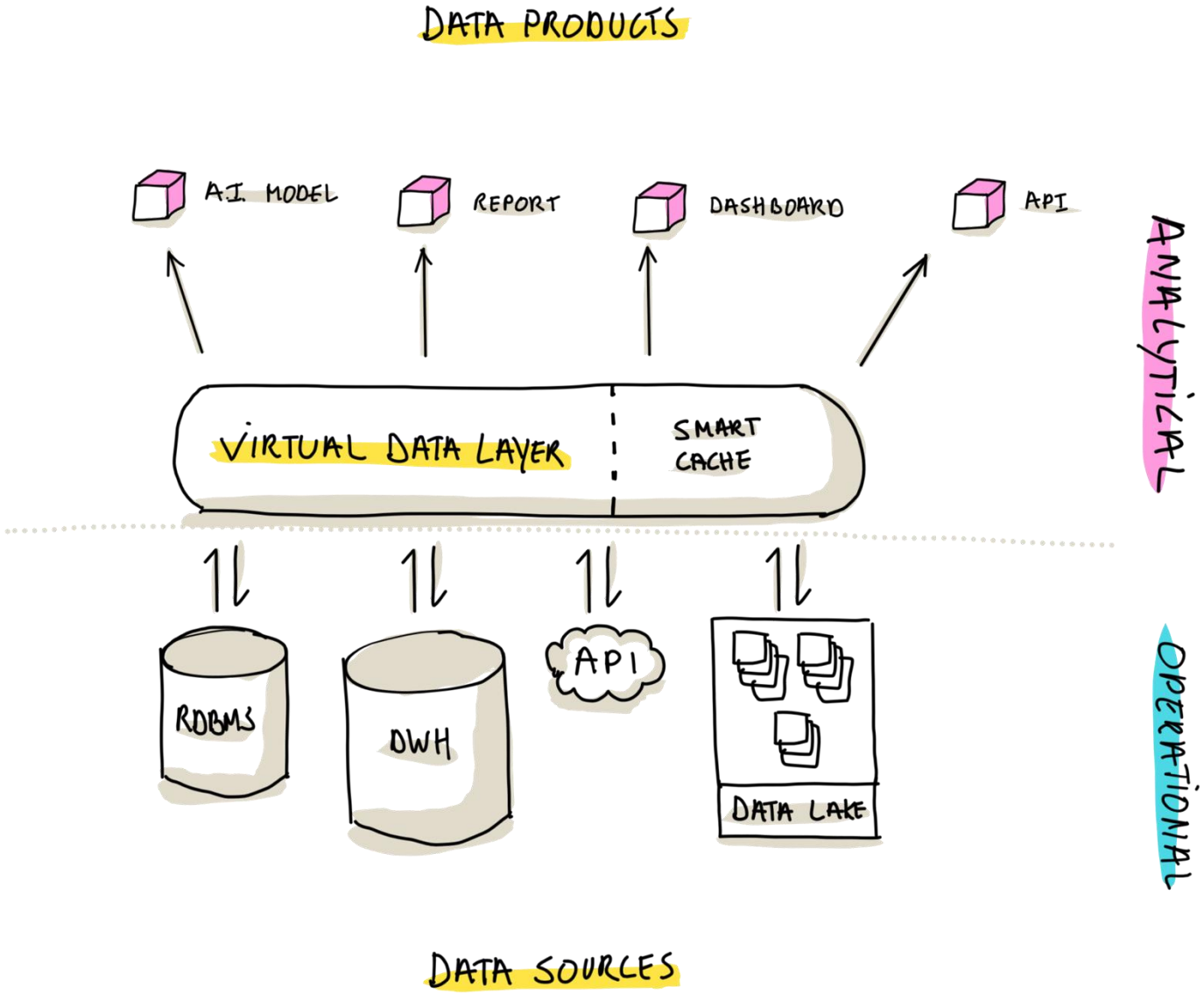
ELT, *sharing the basic steps of ETL, diverges by restructuring and redefining these processes.*

In ELT:

1. **EL — Extract and Load operations** are carried out first, transferring raw data directly to the data platform without immediate transformation.
2. **T — Transformation** occurs subsequently, converting raw data into actionable insights. Crucially, transformation tasks can operate independently and on different schedules from the extraction and loading.

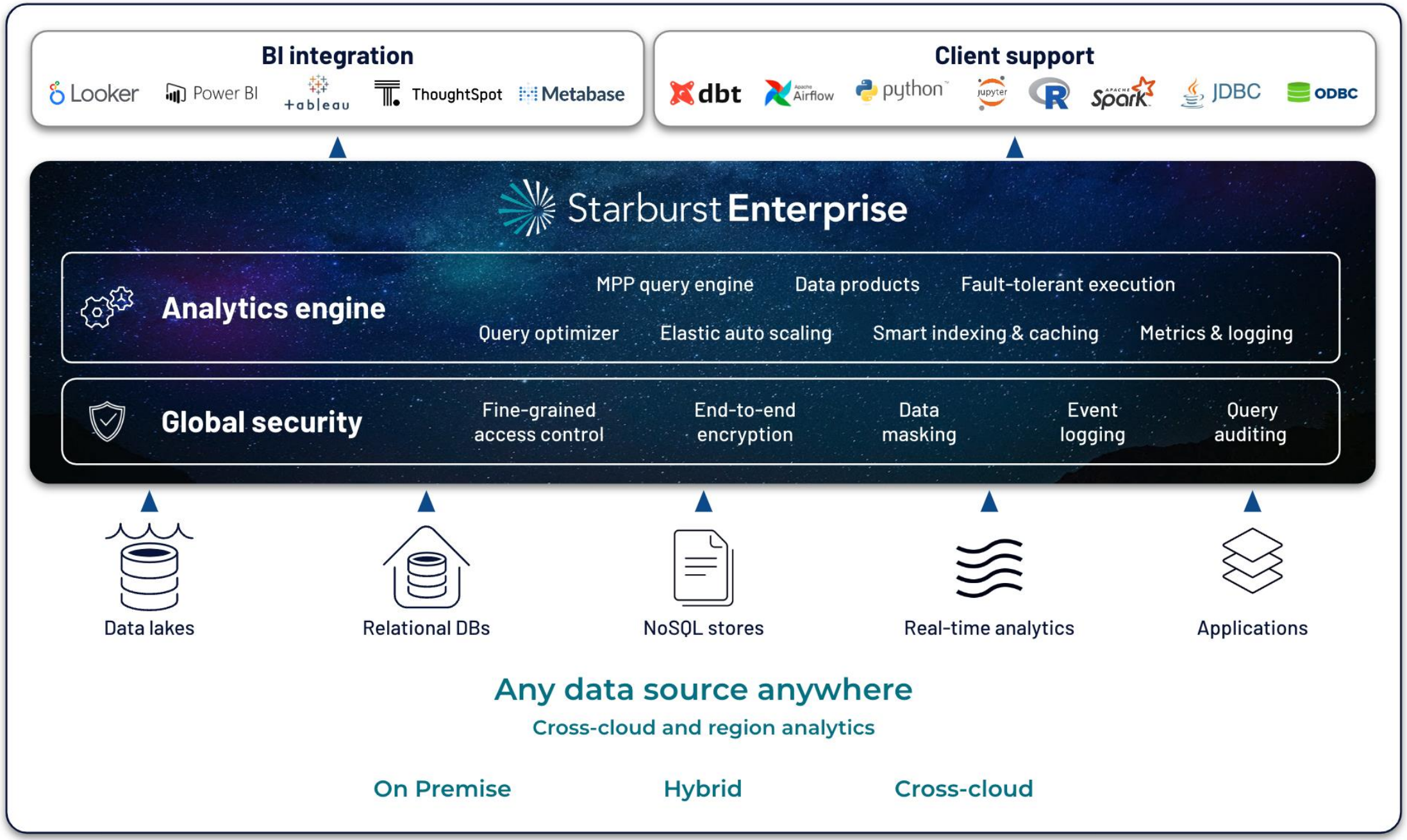
Pattern #4

Data Virtualization



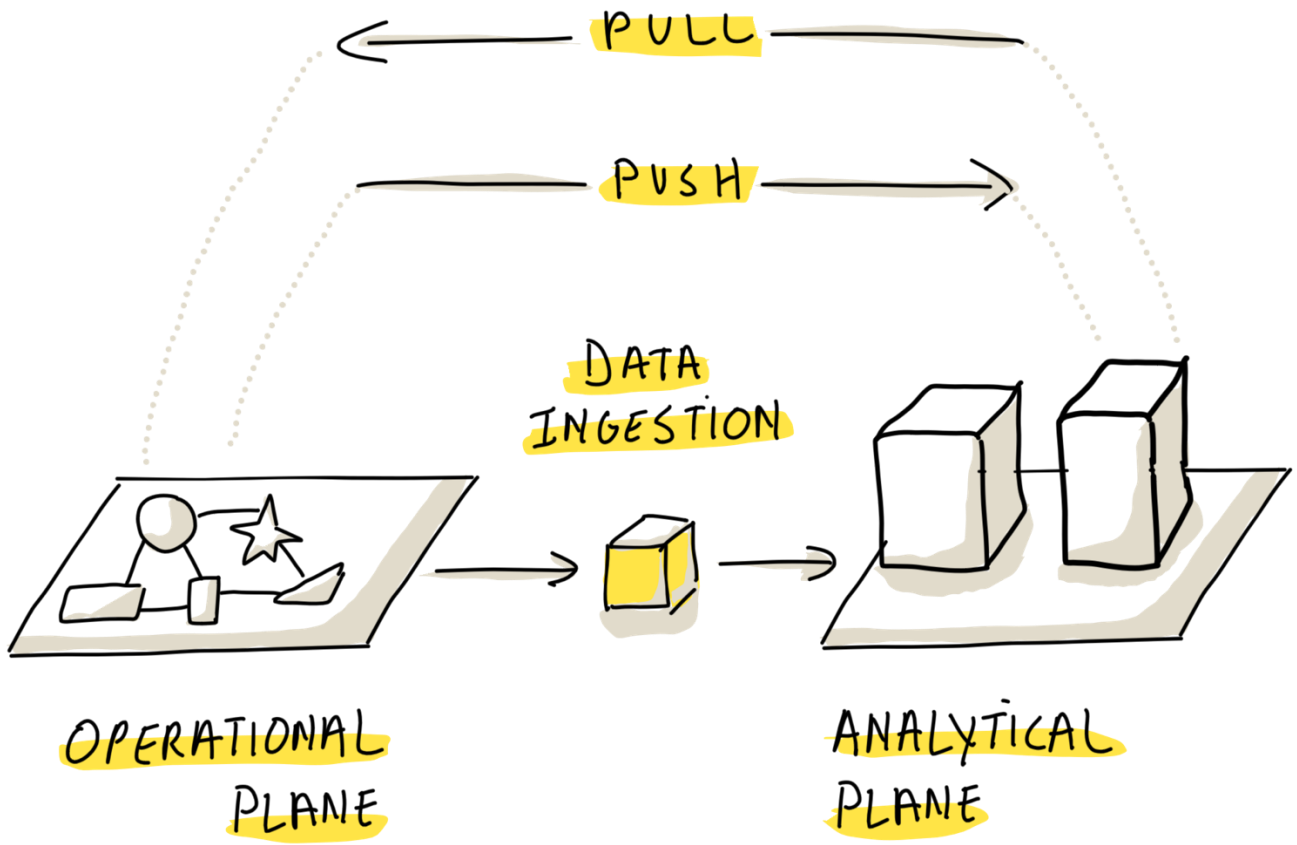
Data Virtualization = specialized software to establish a virtualized data layer over multiple underlying data sources. This intermediary layer allows for the execution of queries that are partially processed by the original data sources, integrating the results into a cohesive dataset for analysis.

- Inspired by the Unified Data Repository (Pattern #1)


BI integration
Looker
Power BI
tableau
ThoughtSpot
Metabase
Client support
dbt
Airflow
python
jupyter
Spark
JDBC
ODBC
Starburst Enterprise
Analytics engine
MPP query engine
Data products
Fault-tolerant execution
Query optimizer
Elastic auto scaling
Smart indexing & caching
Metrics & logging
Global security
Fine-grained access control
End-to-end encryption
Data masking
Event logging
Query auditing
Data lakes
Relational DBs
NoSQL stores
Real-time analytics
Applications
Any data source anywhere
Cross-cloud and region analytics
On Premise
Hybrid
Cross-cloud

Pattern #5

Push vs Pull



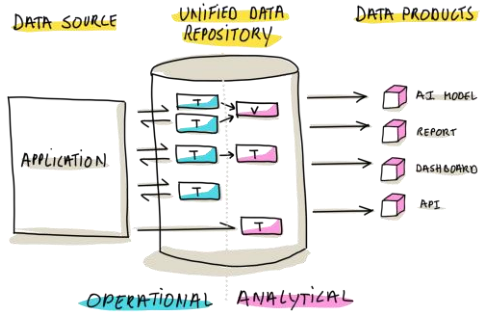
Push = The operational plane initiates data transfer to an endpoint designated by the analytical plane.

- Often found within streaming architectures (discussed next) but is not confined to them.
- Software development teams are mostly responsible to implement the push mechanism

EXERCISE : STRENGTS & WEAKNESSES

Pattern #1

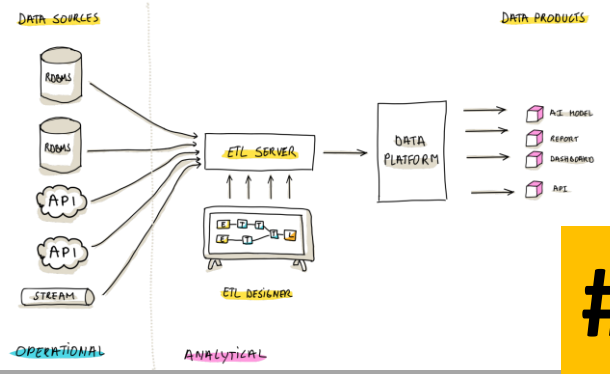
UNIFIED DATA REPOSITORY



#1

Pattern #2

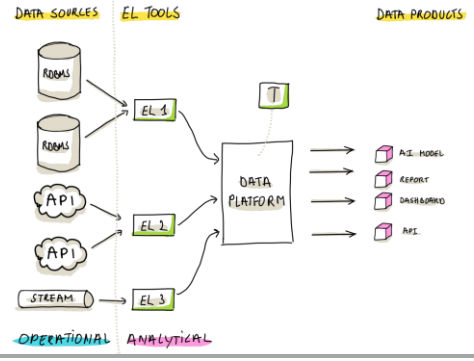
ETL – Extract Transform Load



#2

Pattern #3

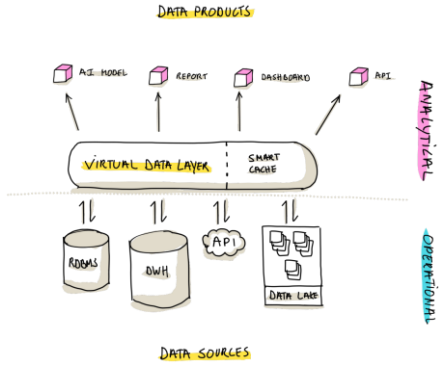
ELT – Extract Transform Load



#3

Pattern #4

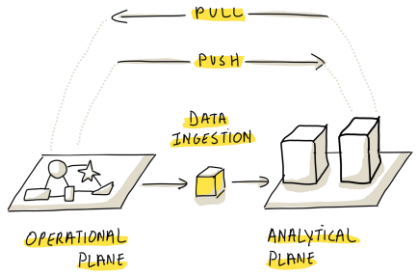
Data Virtualization



#4

Pattern #5

Push vs Pull



#5

Pattern #6

Streaming Data

Streams
record history

"The sequence of moves"

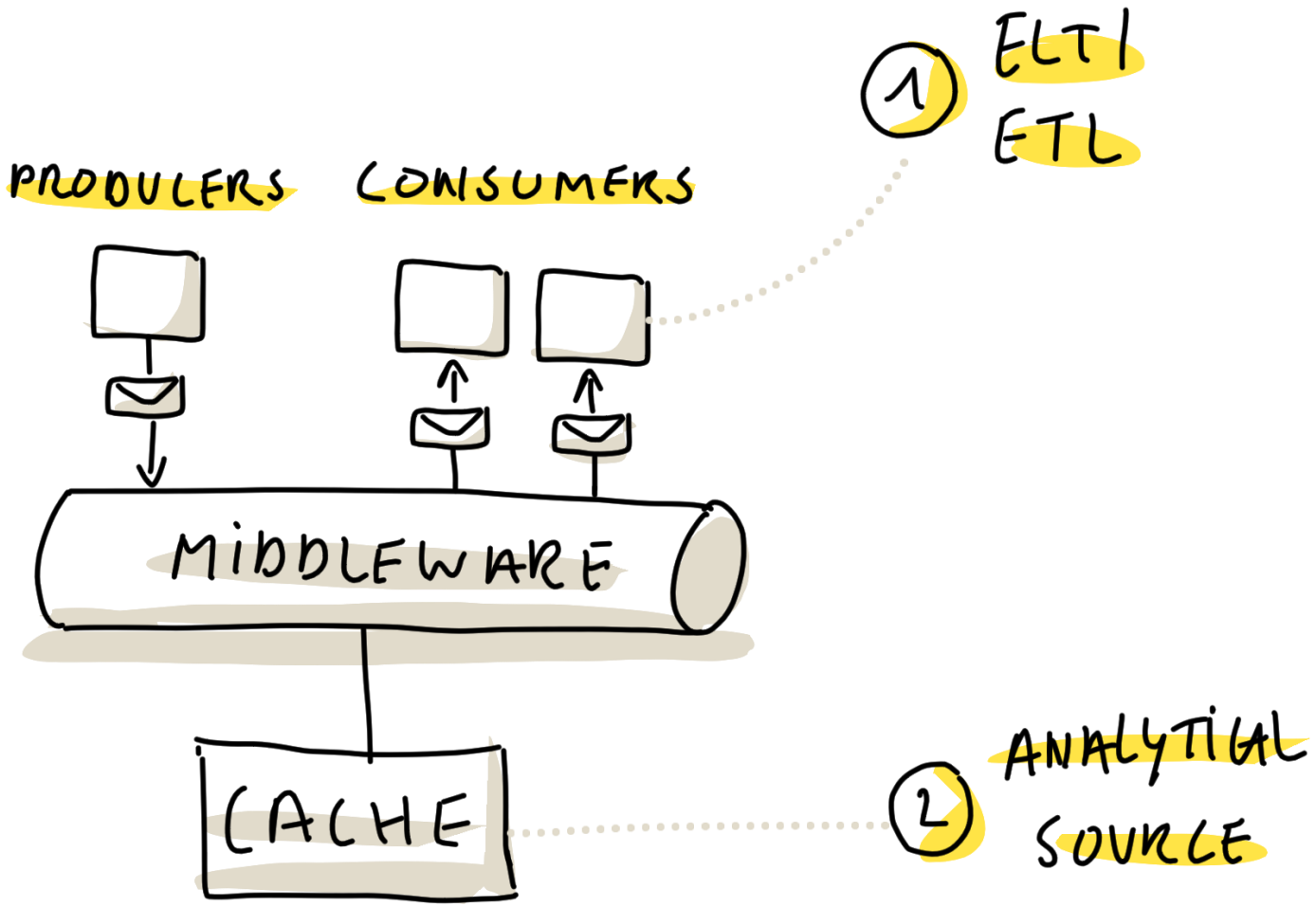
Tables
represent state

"The state of the board"

[Streams and Tables in Apache Kafka – A Primer]

Pattern #6

Streaming Data



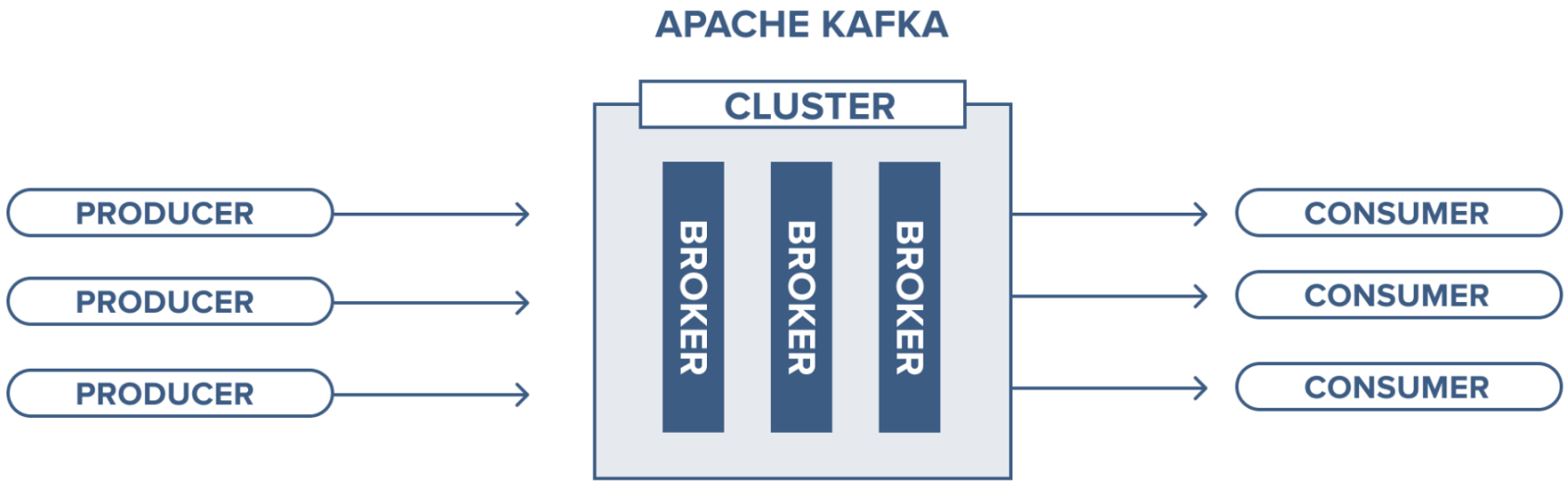
Stream processing = the continuous flow of data as it's generated, enabling real-time processing and analysis for immediate insights.

These systems are crucial for instant decision-making tasks and support high-volume, low-latency processing for activities like financial trades, real-time analytics, and IoT monitoring.

Two common approaches:

- ELT (or ETL) for streaming
- Leveraging streaming caches

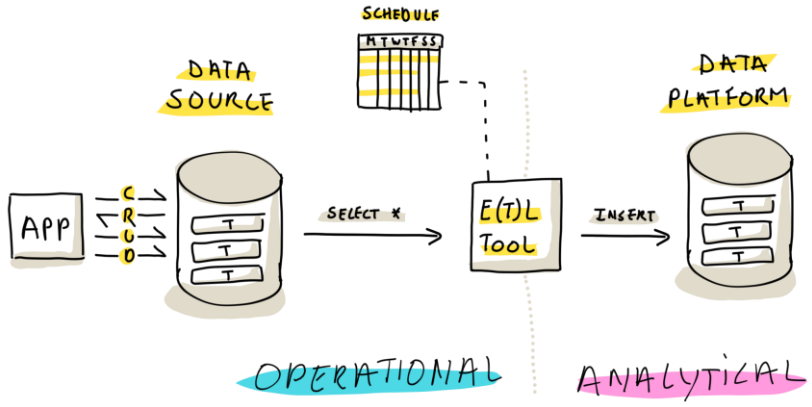
Example: Kafka



Data Ingestion Tool Flavors

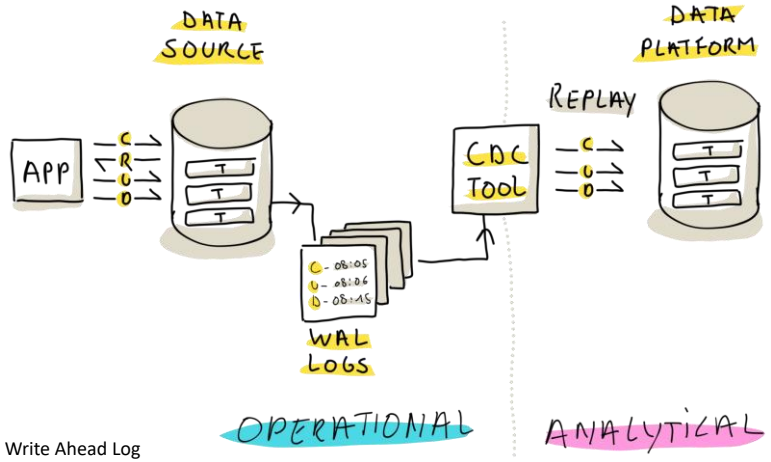
Flavor #1

Batch Loading



Flavor #2

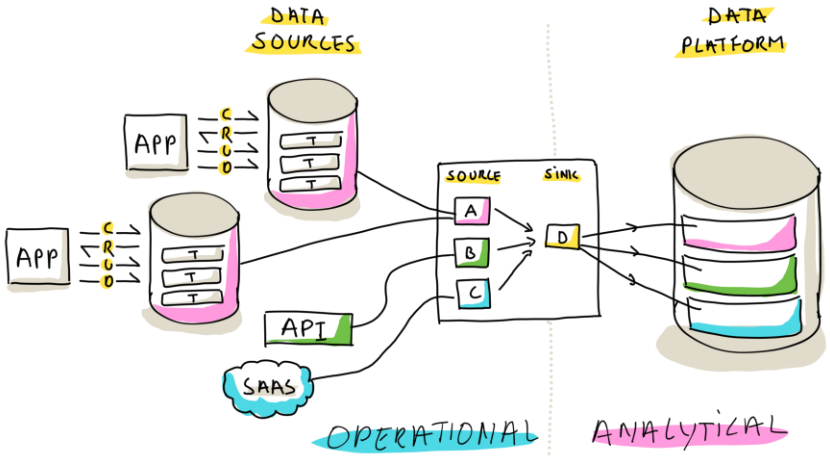
CDC – Change Data Capture



WAL = Write Ahead Log

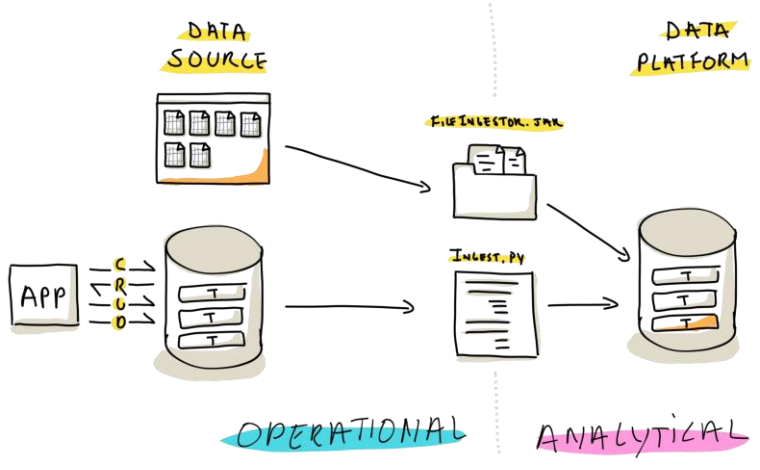
Flavor #3

Connector Based



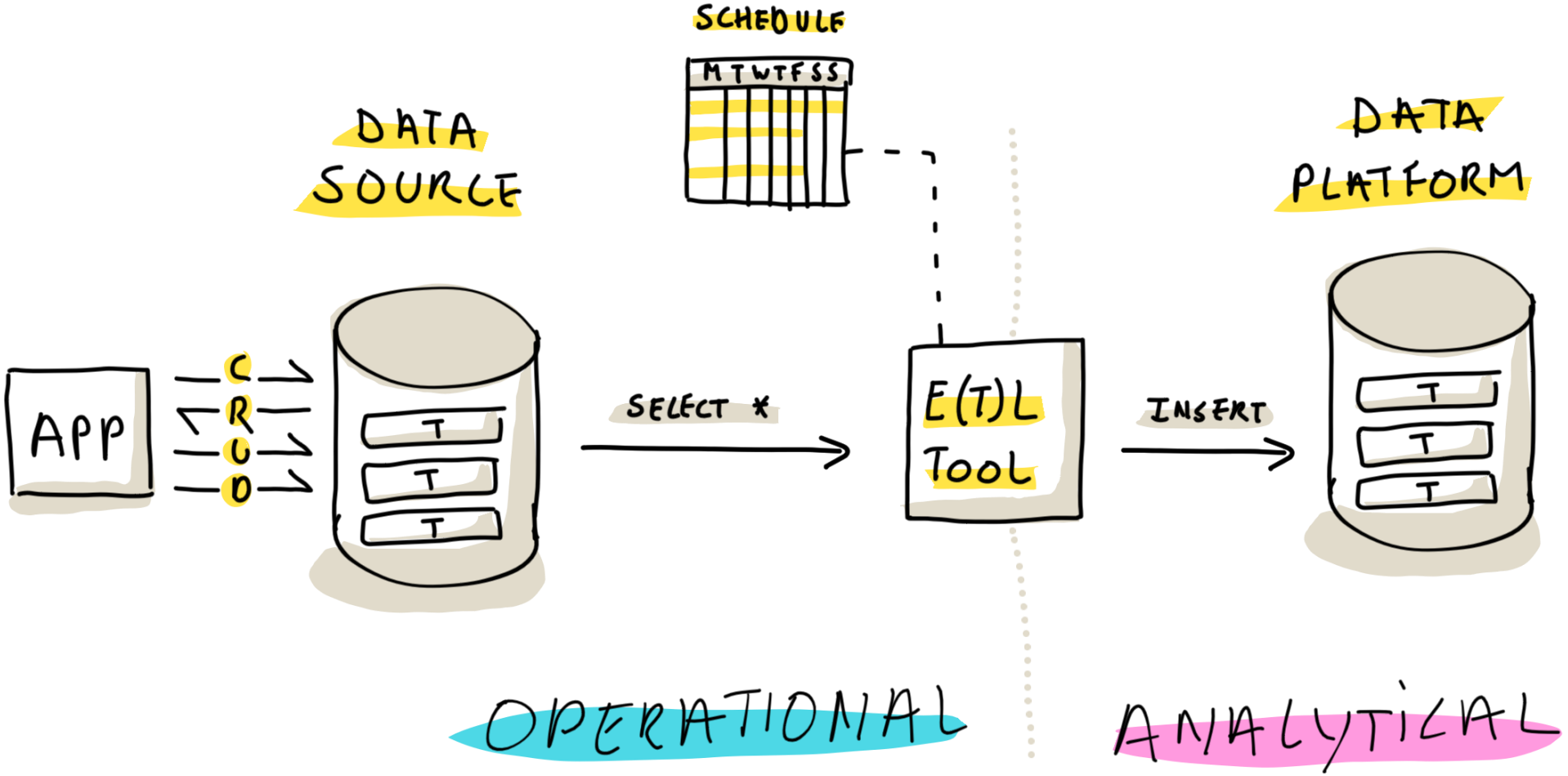
Flavor #4

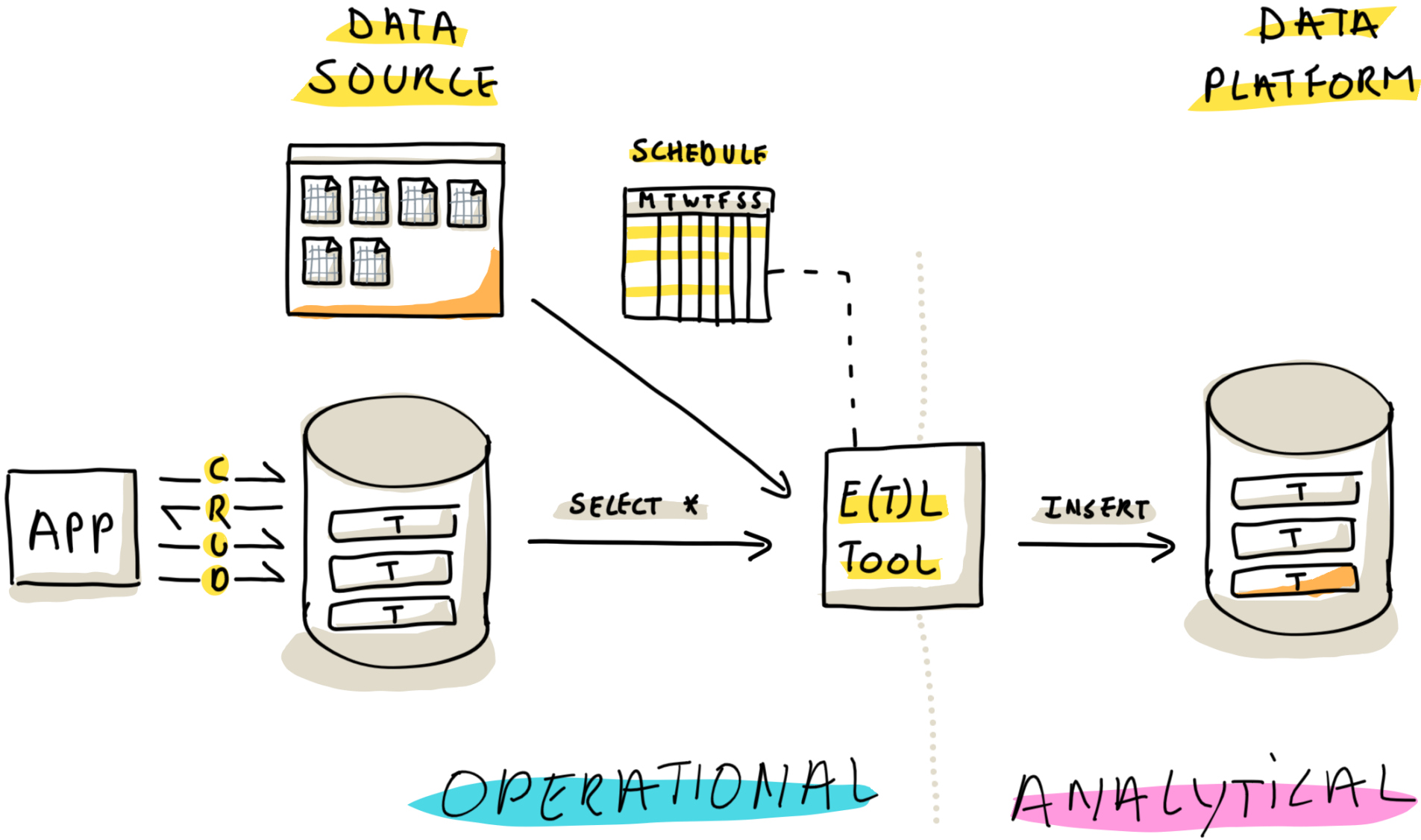
Custom Builds



Flavor #1

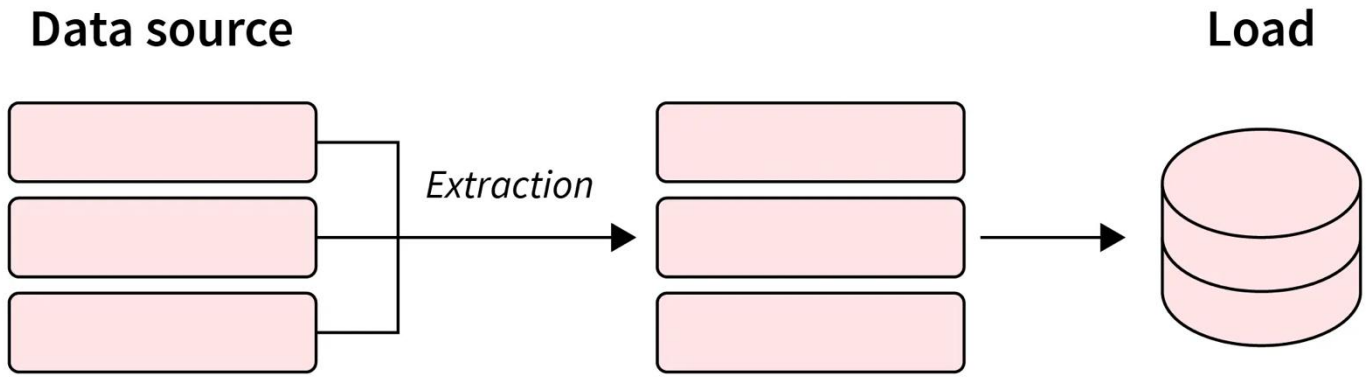
Batch Loading




DATA SOURCE
DATA PLATFORM
SCHEDULE
M T W T F S S
APP
C
R
U
D
T
T
T
SELECT *
E(T)L Tool
INSERT
T
T
T
OPERATIONAL
ANALYTICAL

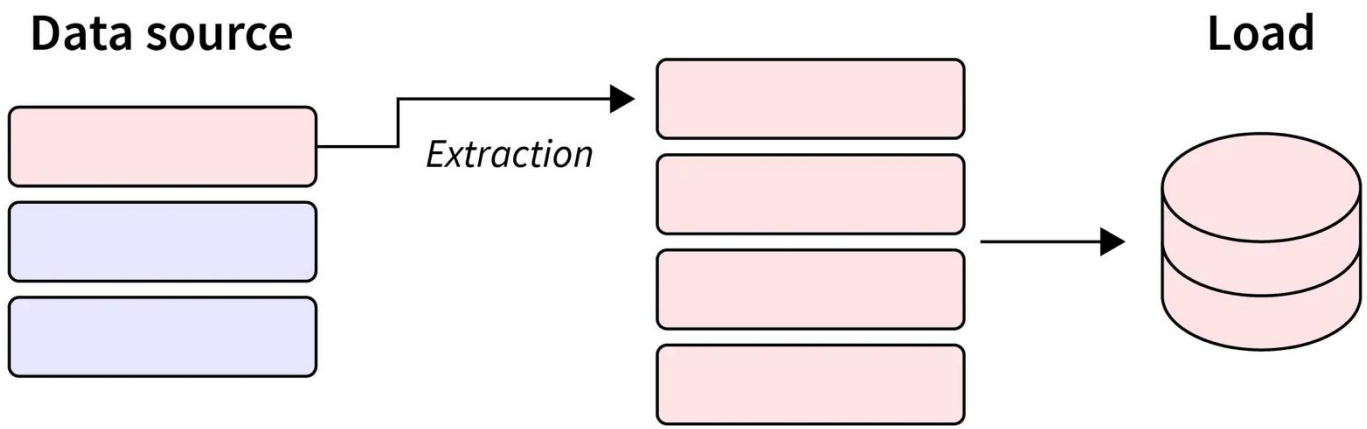
Full load

All available data is extracted at the same time



Incremental load

The occuring changes in the source data is incrementally extracted and loaded



[What is Data Extraction?]

EXERCISE: HOW CAN YOU INCREMENTALLY LOAD THIS TABLE?

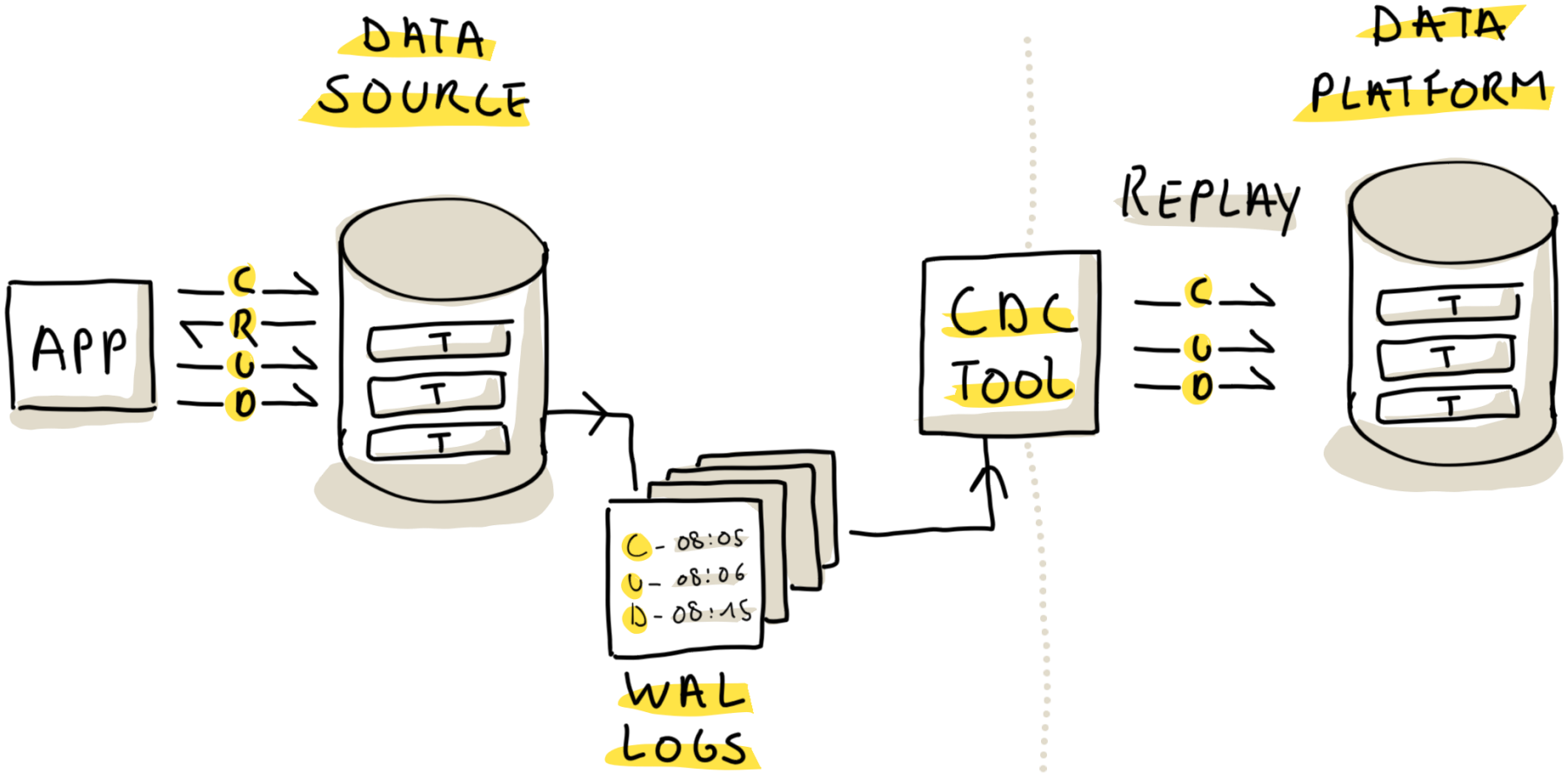
m_transaction

OFF

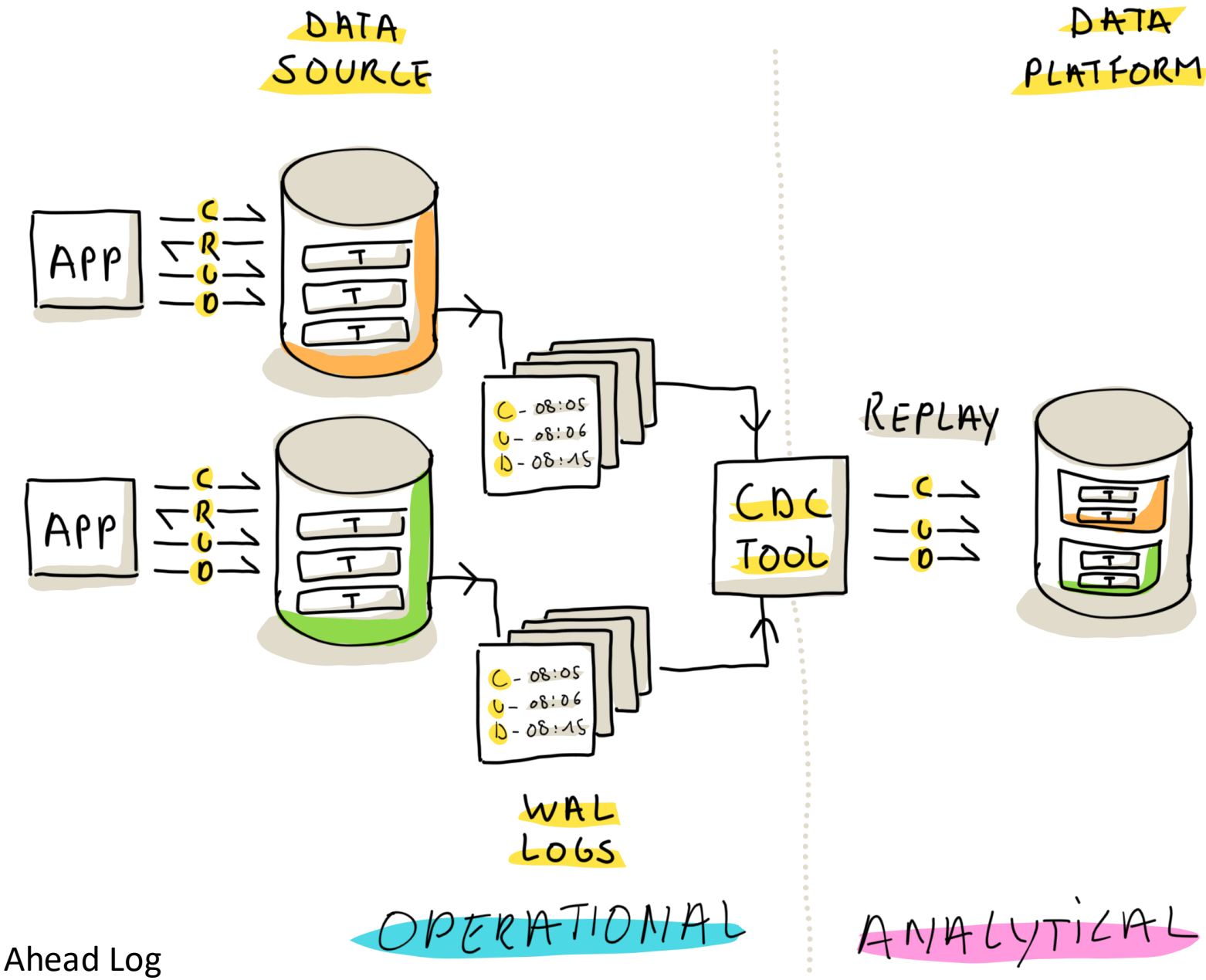
	m_transaction_id	ad_client_id	ad_org_id	isactive	created	createdby	updated
1	1002856	1000010	1000061	Y	2020-03-18 14:40:01.253	100	2020-03-18 14:40:01.253
2	1002857	1000010	1000061	Y	2020-03-18 14:53:40.809	100	2020-03-18 14:53:40.809
3	1002858	1000010	1000061	Y	2020-03-18 15:20:00.275	100	2020-03-18 15:20:00.275
4	1002859	1000010	1000061	Y	2020-03-18 17:42:27.395	1000405	2020-03-18 17:42:27.395
5	1002860	1000010	1000129	Y	2020-03-18 19:50:49.07	100	2020-03-18 19:50:49.07
6	1002861	1000010	1000129	Y	2020-03-18 19:59:42.211	100	2020-03-18 19:59:42.211
7	1002862	1000010	1000129	Y	2020-03-18 20:00:39.243	100	2020-03-18 20:00:39.243
8	1002863	1000010	1000129	Y	2020-03-18 20:02:20.357	100	2020-03-18 20:02:20.357
9	1002864	1000010	1000129	Y	2020-03-18 20:10:12.598	100	2020-03-18 20:10:12.598
10	1002865	1000010	1000061	Y	2020-03-18 20:25:26.384	100	2020-03-18 20:25:26.384

Flavor #2

CDC – Change **D**ata Capture

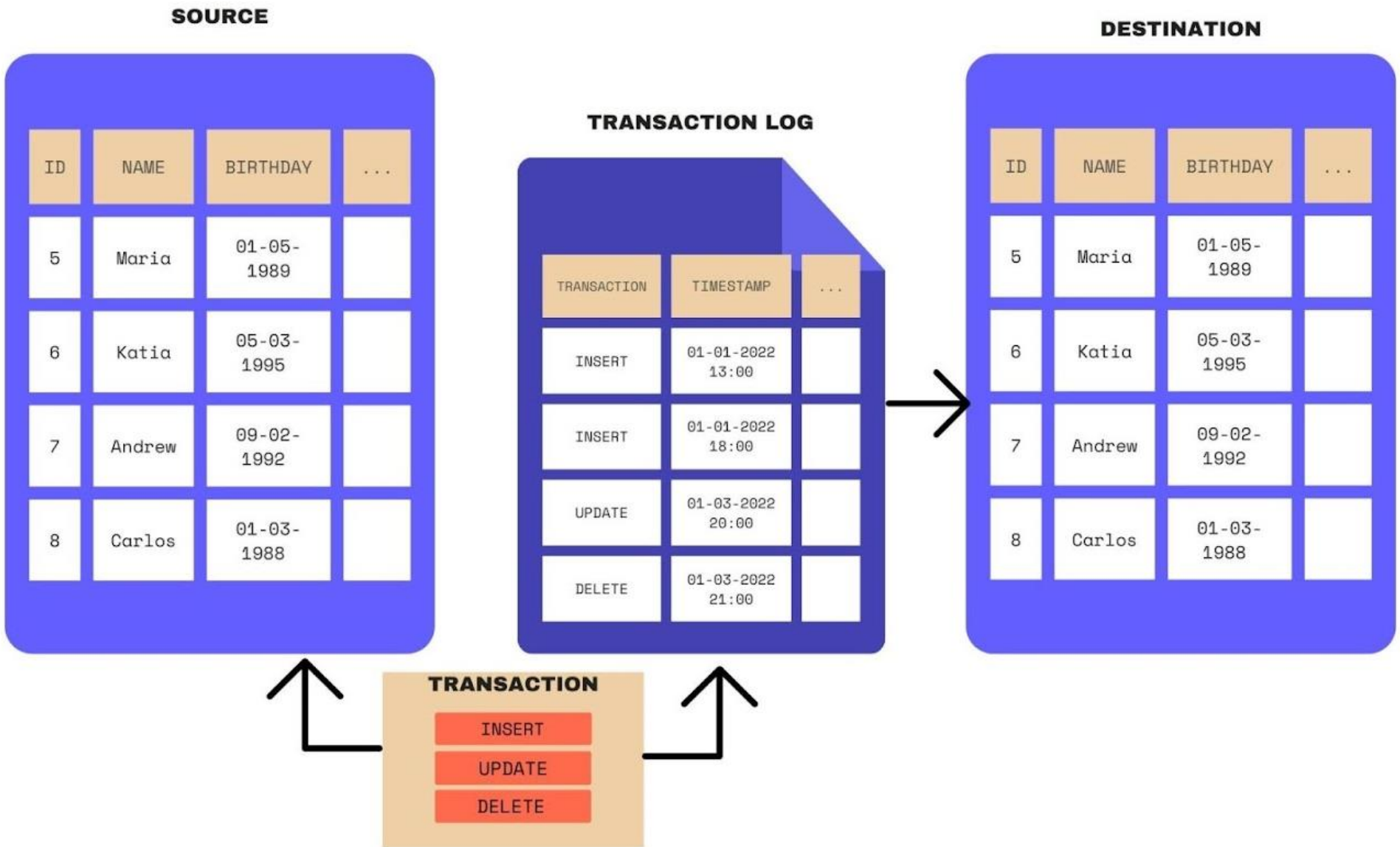


WAL = Write Ahead Log

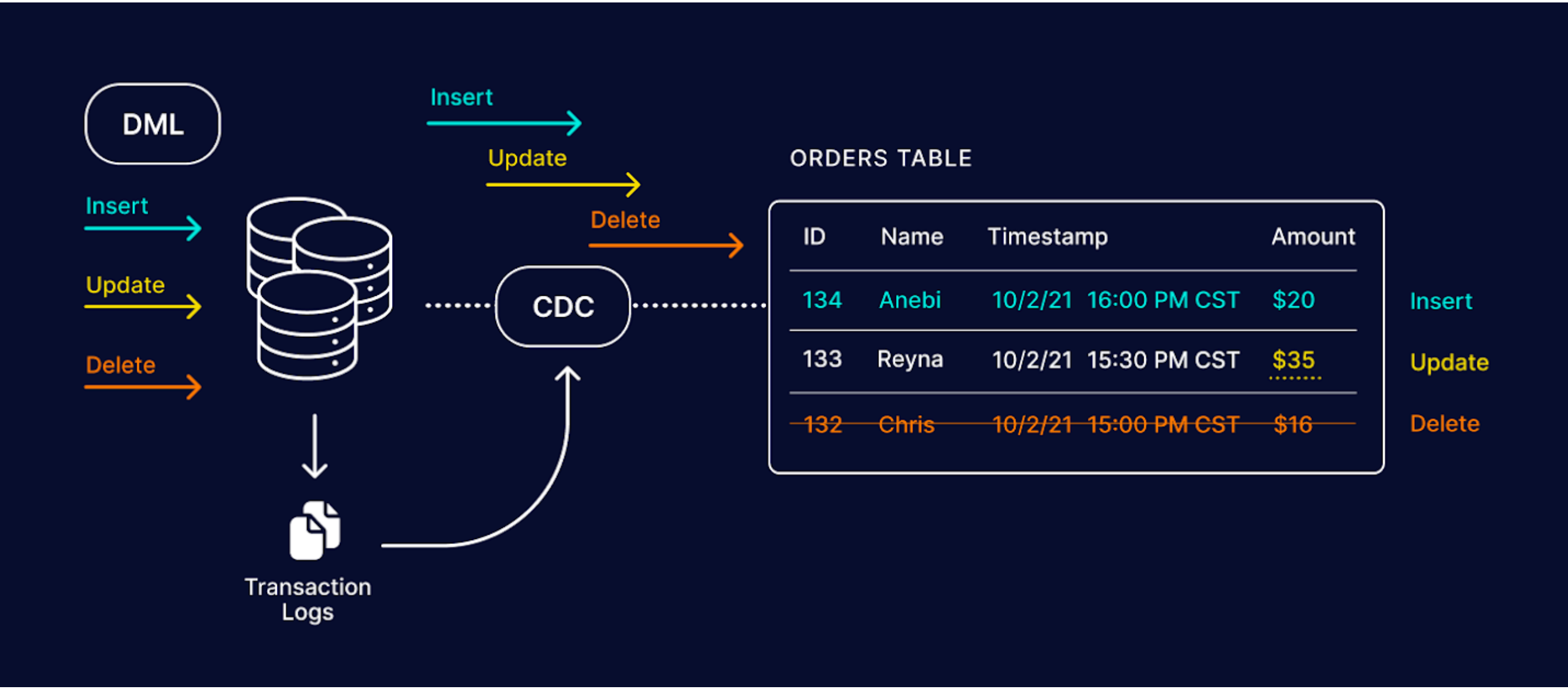


WAL = Write Ahead Log

Log-based CDC technique



[Understanding Change Data Capturing]



[How to Implement CDC with Apache Kafka]

Change Data Capture (CDC) = a software that allows detecting and capturing changes made to data in a database and sending these changes, sometimes in real-time, to a downstream process or system. More specifically, CDC entails recording INSERT, UPDATE, and DELETE transactions applied to a table.

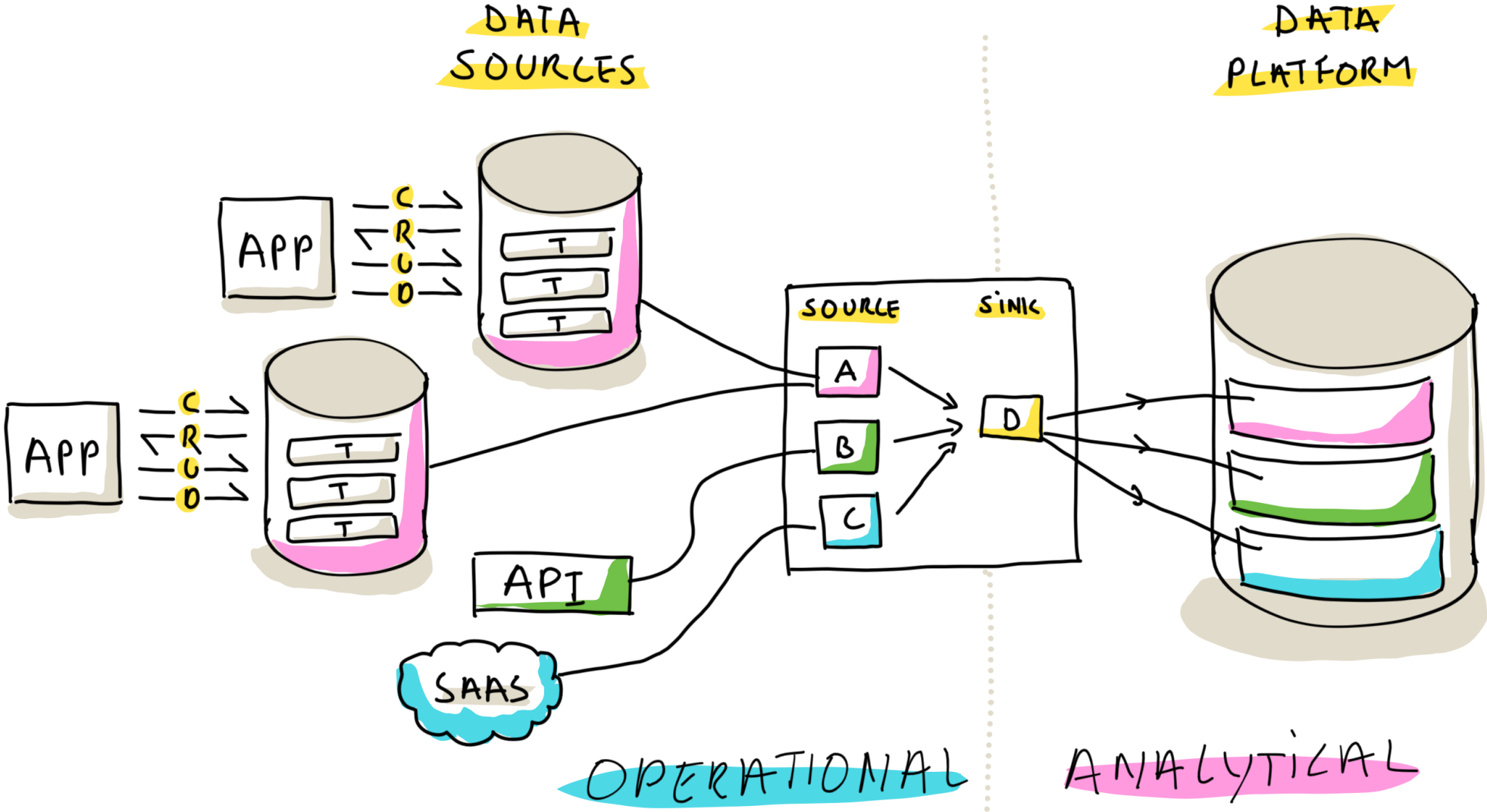
Various techniques exist: meta-data based, trigger based, **log based**

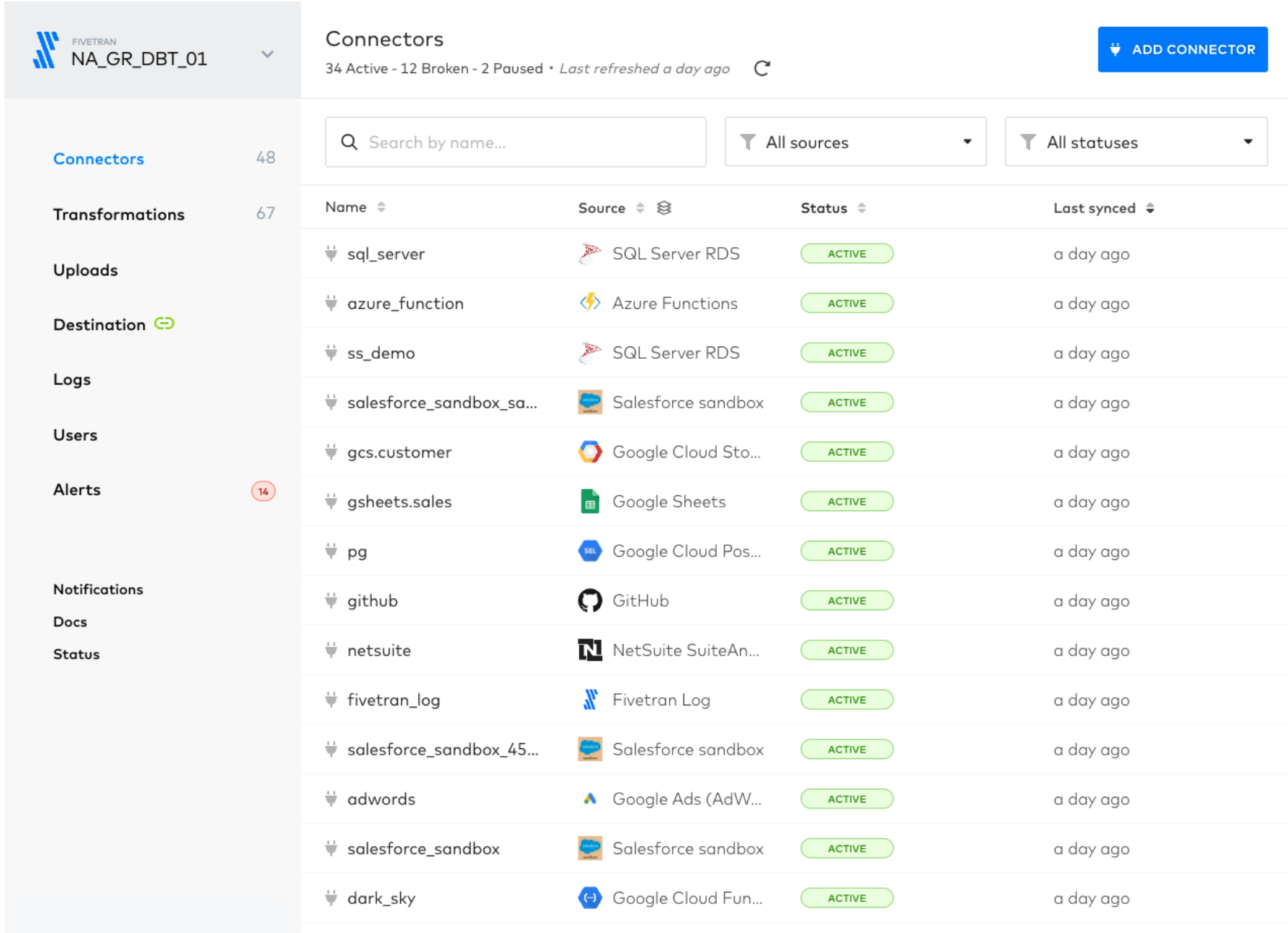
Log based CDC systems read data directly from the database Change Data Capture logs to identify changes in a database (not from the actual database)

[Understanding Change Data Capturing]

Flavor #3

Connector Based




FIVETRAN
NA_GR_DBT_01
Connectors
48
Transformations
67
Uploads
Destination
Logs
Users
Alerts
14
Notifications
Docs
Status
Connectors
34 Active - 12 Broken - 2 Paused • Last refreshed a day ago
ADD CONNECTOR
Search by name...
All sources
All statuses
Name
Source
Status
Last synced
sql_server
SQL Server RDS
ACTIVE
a day ago
azure_function
Azure Functions
ACTIVE
a day ago
ss_demo
SQL Server RDS
ACTIVE
a day ago
salesforce_sandbox_sa...
Salesforce sandbox
ACTIVE
a day ago
gcs.customer
Google Cloud Sto...
ACTIVE
a day ago
gsheets.sales
Google Sheets
ACTIVE
a day ago
pg
Google Cloud Pos...
ACTIVE
a day ago
github
GitHub
ACTIVE
a day ago
netsuite
NetSuite SuiteAn...
ACTIVE
a day ago
fivetran_log
Fivetran Log
ACTIVE
a day ago
salesforce_sandbox_45...
Salesforce sandbox
ACTIVE
a day ago
adwords
Google Ads (AdW...
ACTIVE
a day ago
salesforce_sandbox
Salesforce sandbox
ACTIVE
a day ago
dark_sky
Google Cloud Fun...
ACTIVE
a day ago

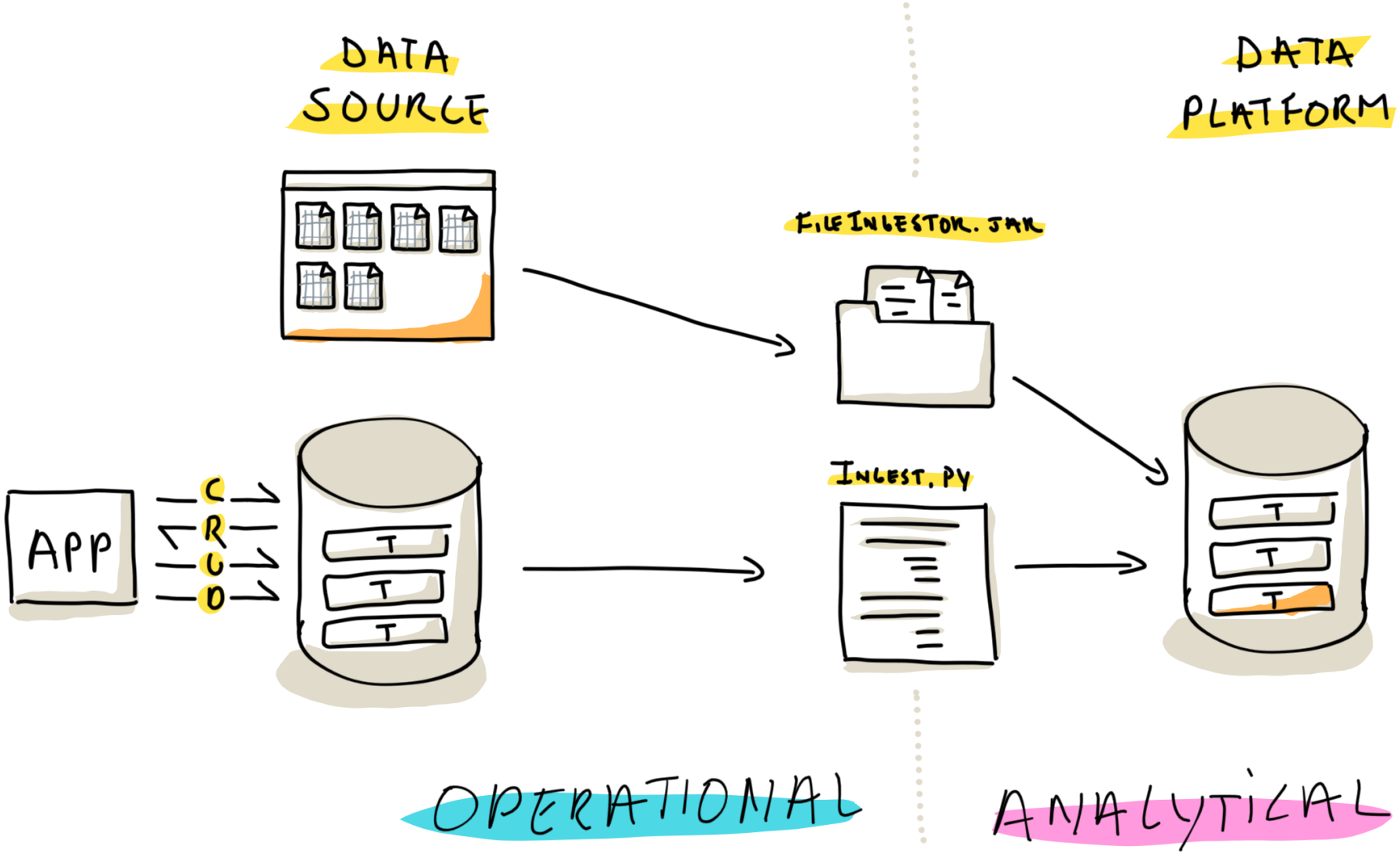
[Fivetran]

Pre-built source connectors are coupled with pre-built sink connectors, often using a graphical user interface.

- **Source connectors:** can be very diverse databases, APIs, SAAS-applications, applications, files, …
- **Sink connectors:** mostly limited to (analytical) databases or data lakes.
- Higly **flexible** but **No control** over the individual connectors

Flavor #4

Custom Builds

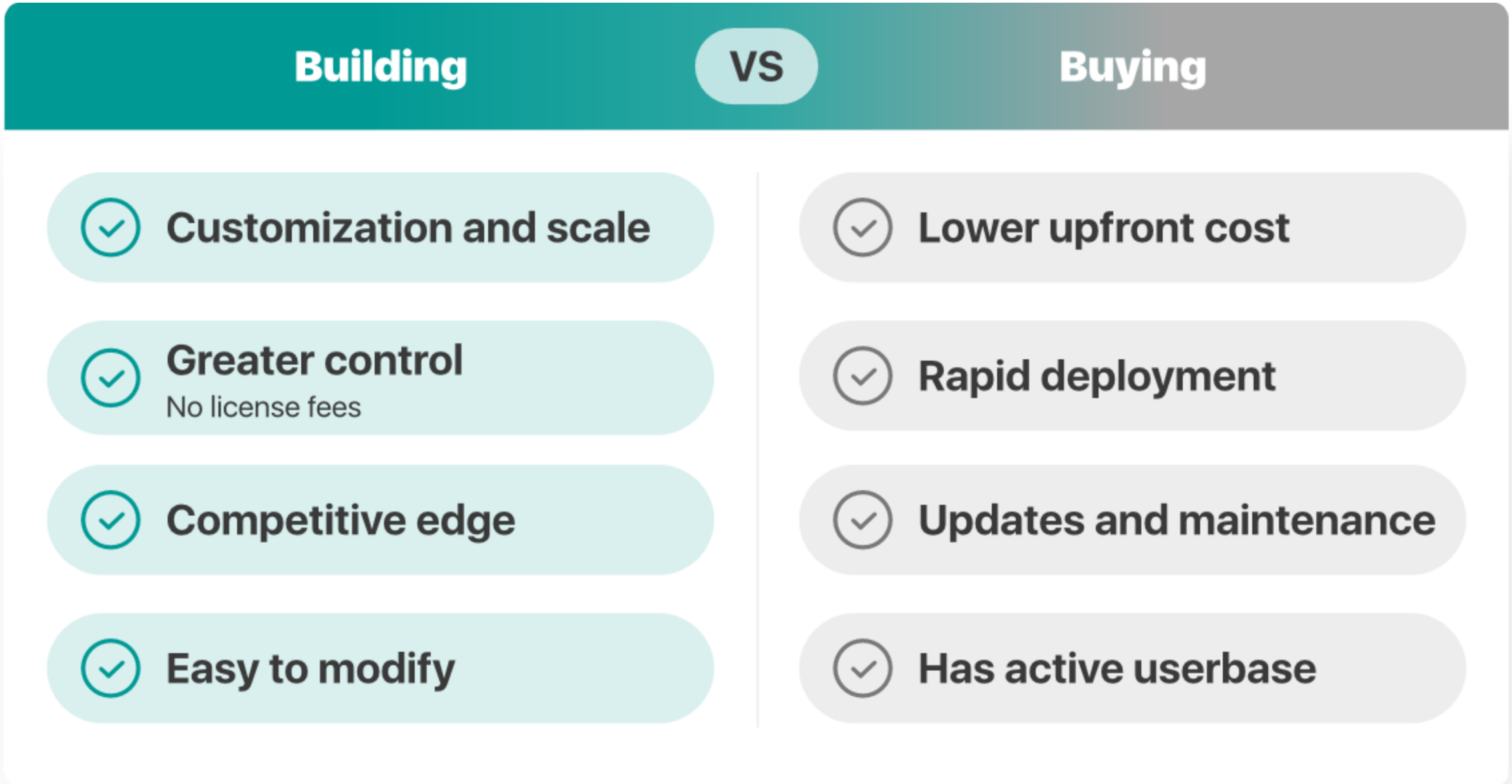


Disadvantages:

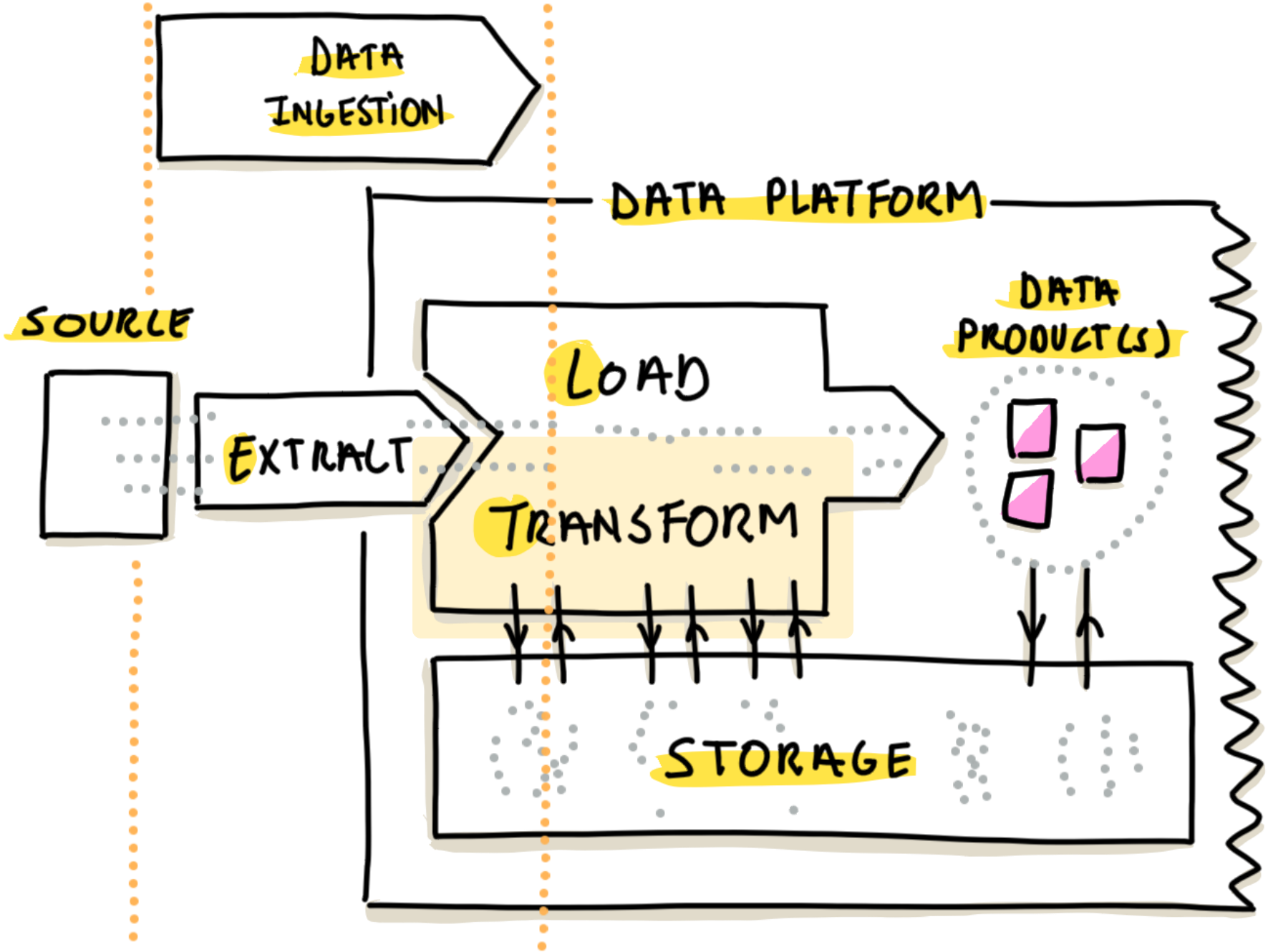
- Building ingestion pipelines is usually more expensive than expected
- Specific programming knowledge & team needed
- High maintenance cost

Advantages:

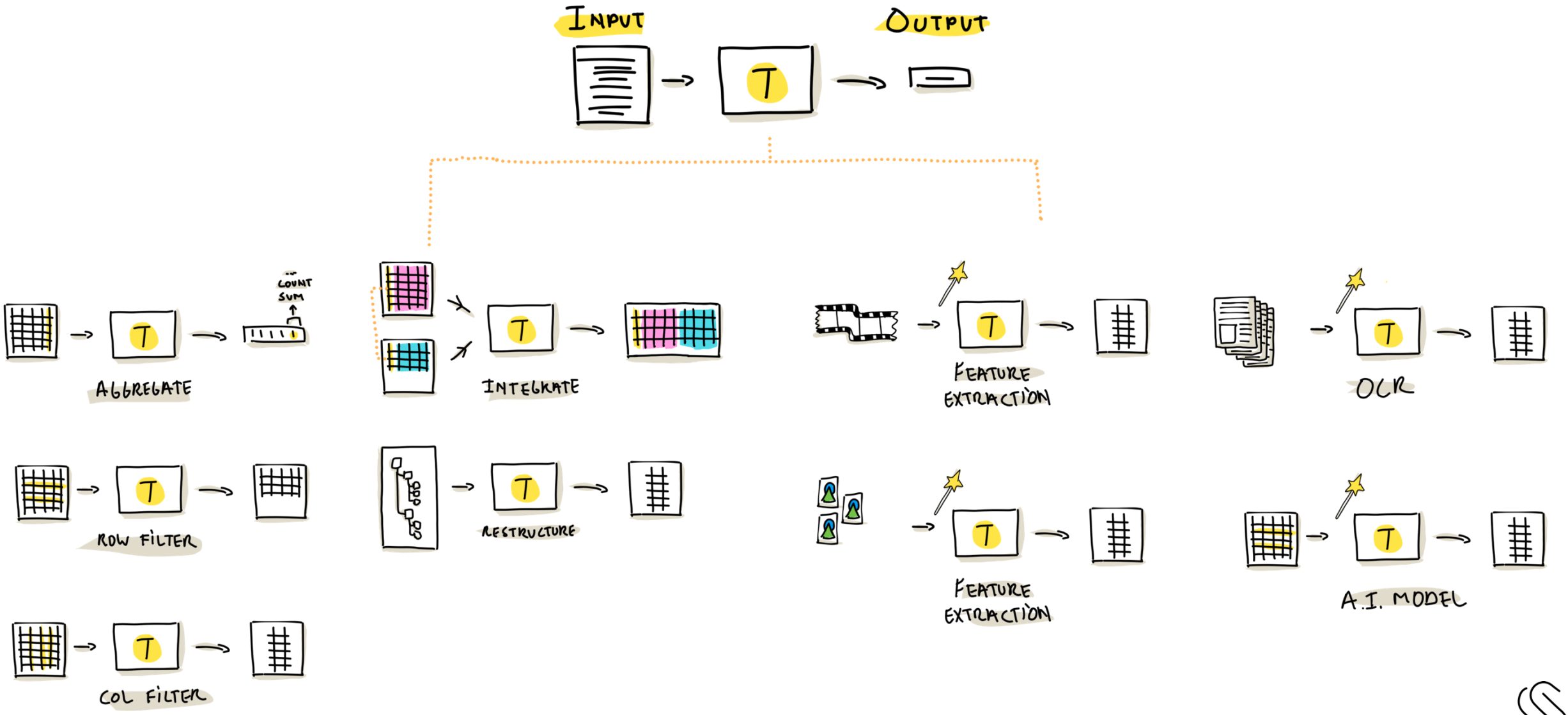
- Full control
- Allows to ingest very specific / unique / exotic data sources



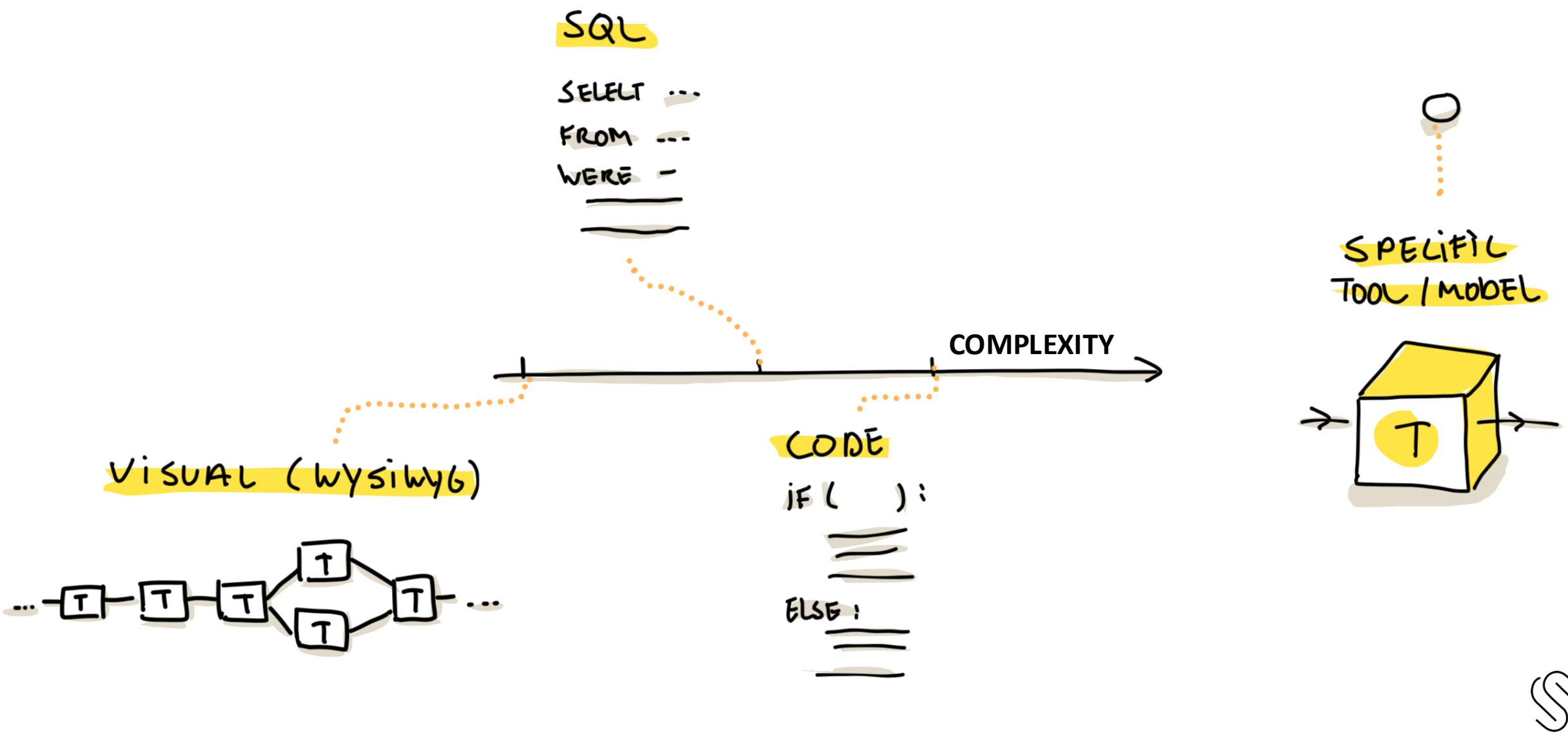
[Build vs Buy: The Pros and Cons]


DATA
INGESTION
DATA PLATFORM
SOURCE
EXTRACT
LOAD
TRANSFORM
DATA
PRODUCT(S)
STORAGE

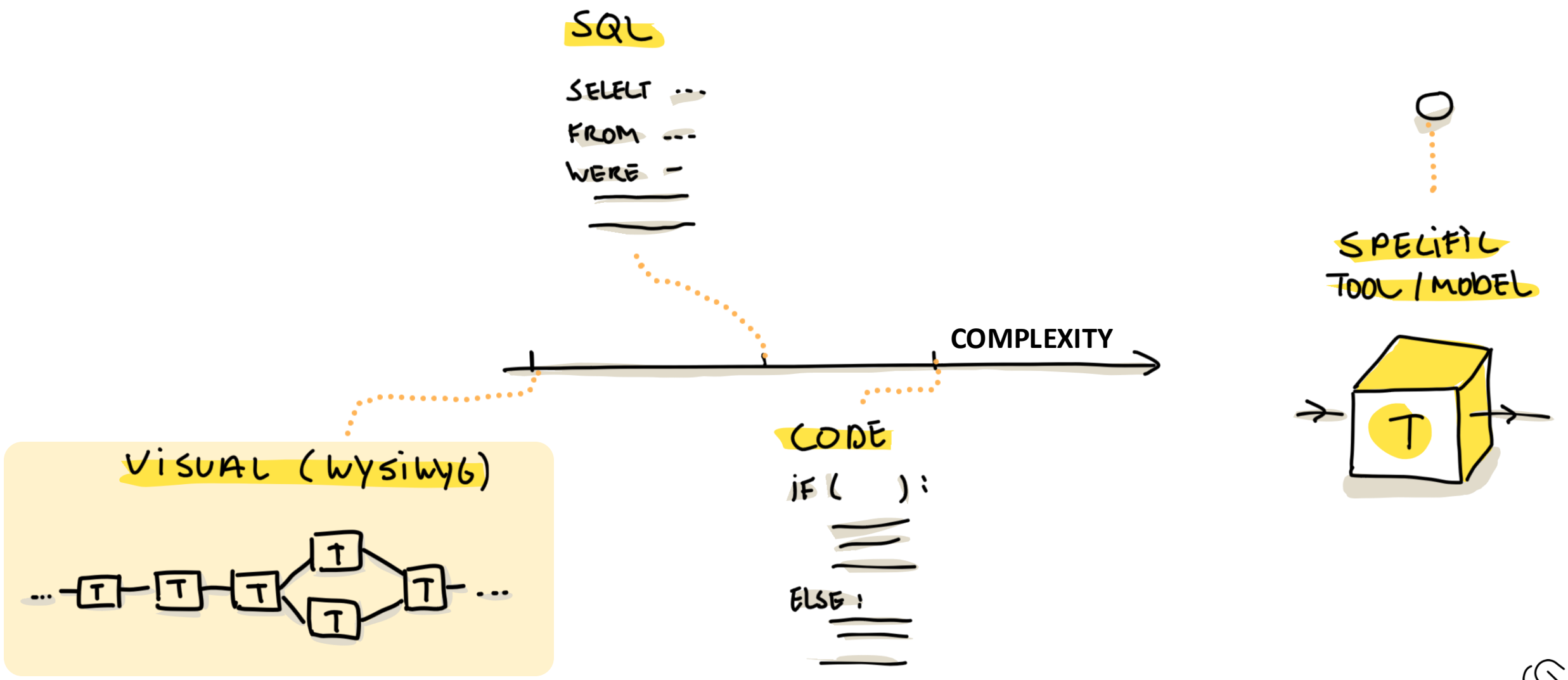
Data Transformations



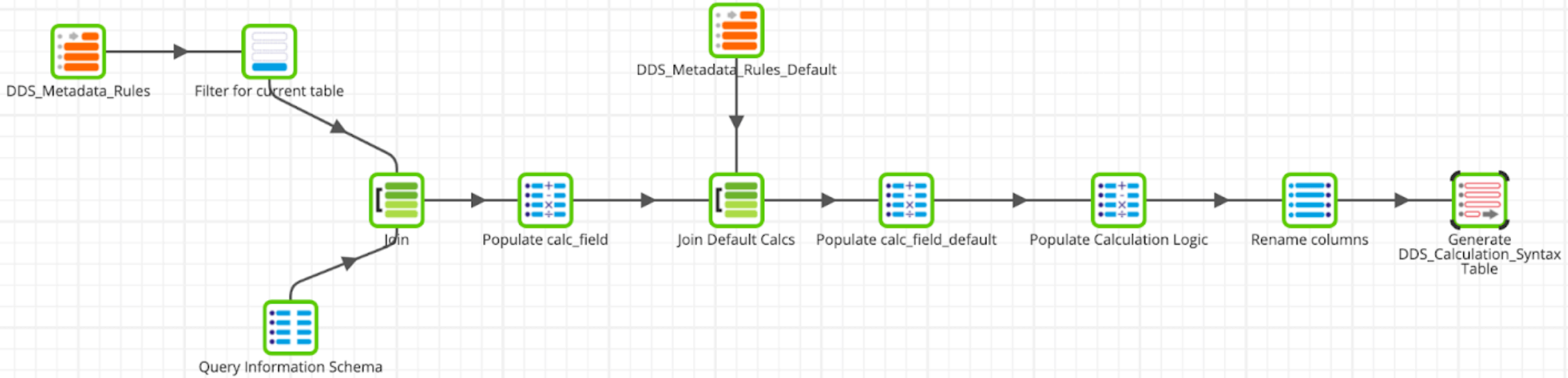
Data Transformation Tool Flavors



Data Transformation Tool Flavors



Purpose:
The purpose of this transformation step is to formulate the appropriate syntax that should be applied onto the columns during the transformation step of the process. All of the syntax will be dynamically populated into a syntax metadata table called 'DDS_Calculation_Syntax'.



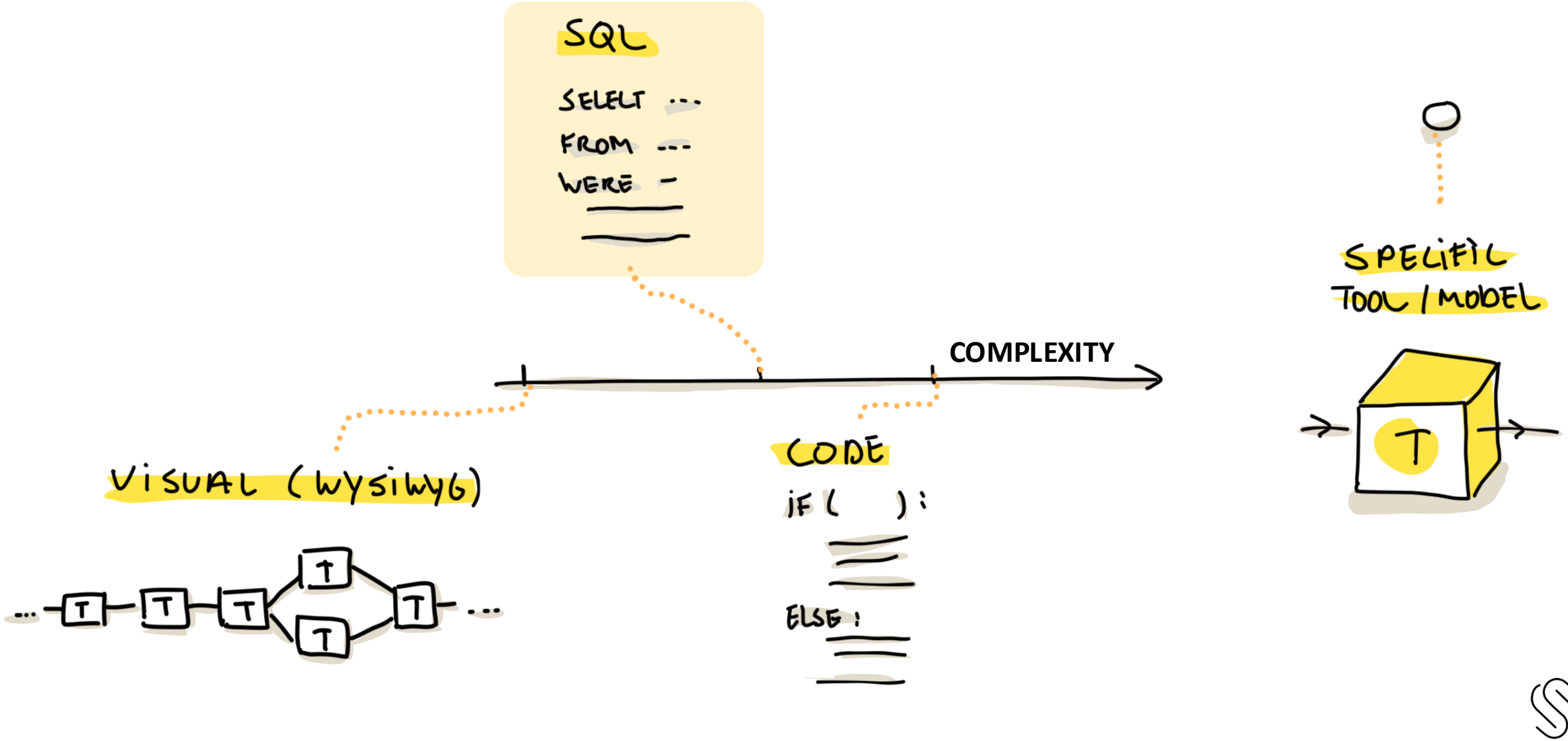
Properties | Export | Sample | Metadata | SQL | Plan | Help

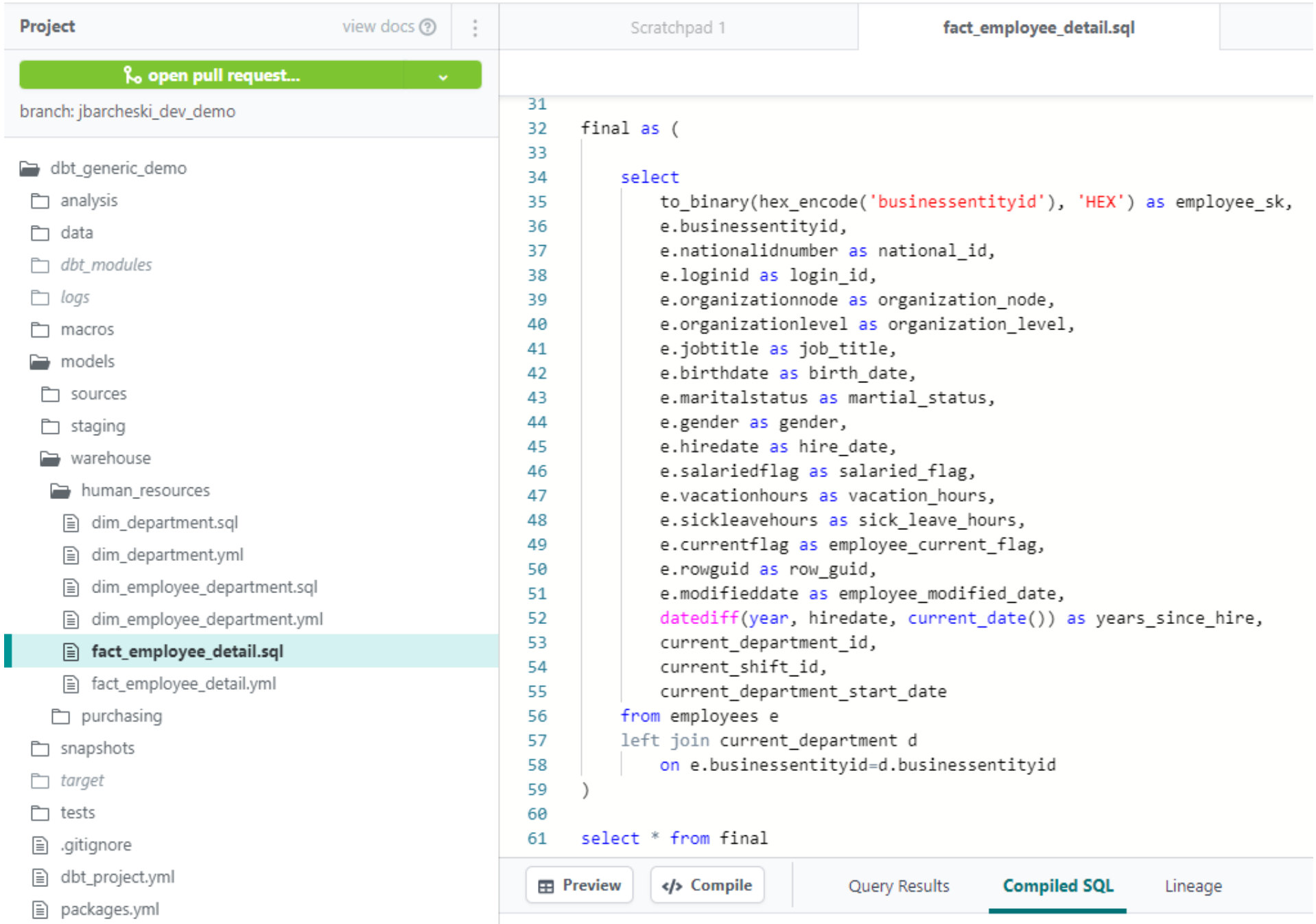
Data | Row Count | Filter Not Set | Export Sa

SCHEMA	TABLE_NAME	COLUMN_NAME	calculation
KB_AZURE_SNOW	DDS_STATE	STATEID	STATEID
KB_AZURE_SNOW	DDS_STATE	COUNTRYNAME	INITCAP(NVL(TRIM("COUNTRYNAME"),' '))
KB_AZURE_SNOW	DDS_STATE	STATECODE	UPPER(NVL(TRIM("STATECODE"),' '))
KB_AZURE_SNOW	DDS_STATE	FLAG	TO_BOOLEAN("FLAG")
KB_AZURE_SNOW	DDS_STATE	DATE	TO_TIMESTAMP_NTZ(TO_VARCHAR("DATE"), 'yyyymmdd')
KB_AZURE_SNOW	DDS_STATE	STATENAME	INITCAP(NVL(TRIM("STATENAME"),' '))

[Matillion]

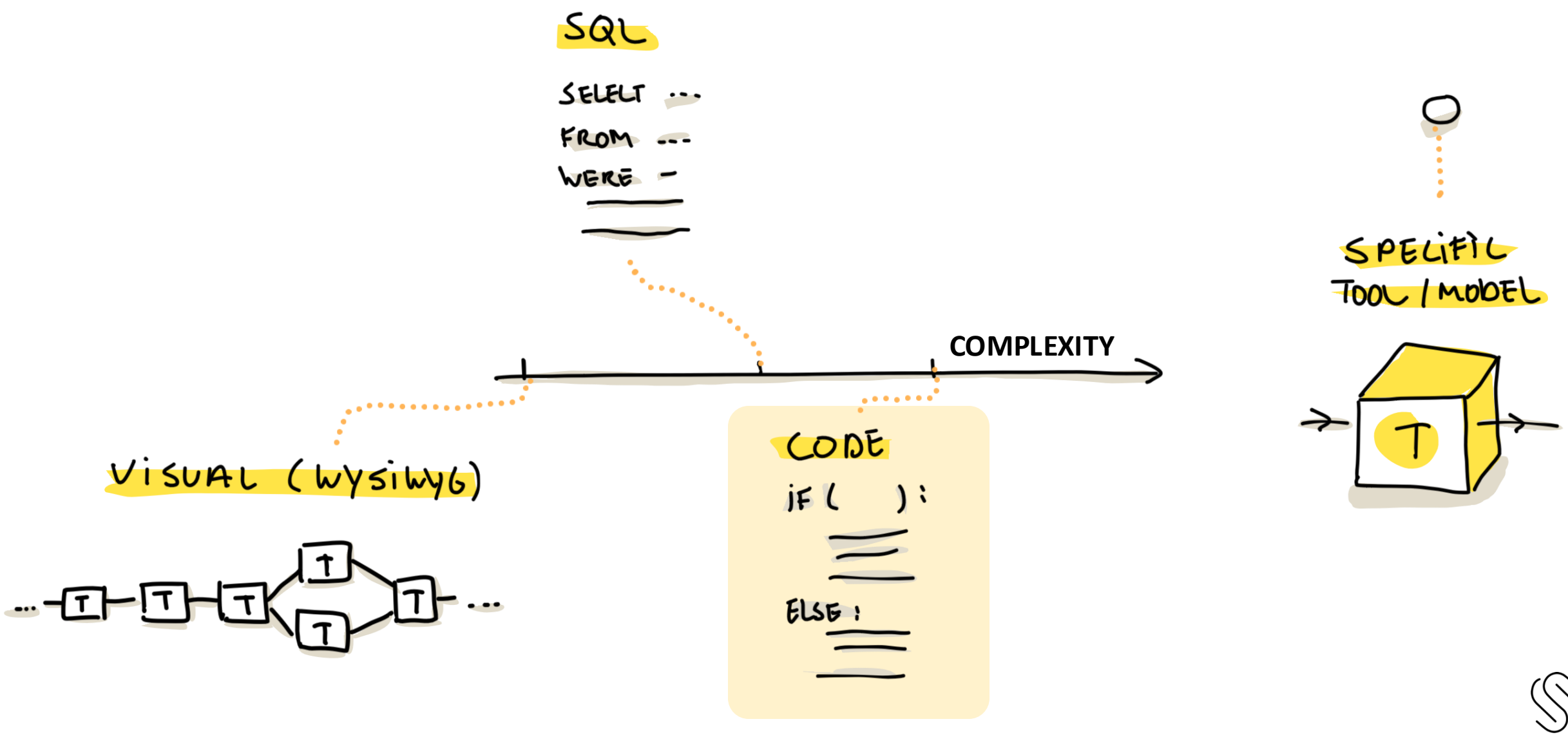
Data Transformation Tool Flavors




Project
view docs
open pull request...
branch: jbarcheski_dev_demo
dbt_generic_demo
analysis
data
dbt_modules
logs
macros
models
sources
staging
warehouse
human_resources
dim_department.sql
dim_department.yml
dim_employee_department.sql
dim_employee_department.yml
fact_employee_detail.sql
fact_employee_detail.yml
purchasing
snapshots
target
tests
.gitignore
dbt_project.yml
packages.yml
Scratchpad 1
fact_employee_detail.sql
31
32 final as (
33
34 select
35 to_binary(hex_encode('businessentityid'), 'HEX') as employee_sk,
36 e.businessentityid,
37 e.nationalidnumber as national_id,
38 e.loginid as login_id,
39 e.organizationnode as organization_node,
40 e.organizationlevel as organization_level,
41 e.jobtitle as job_title,
42 e.birthdate as birth_date,
43 e.maritalstatus as martial_status,
44 e.gender as gender,
45 e.hiredate as hire_date,
46 e.salariedflag as salaried_flag,
47 e.vacationhours as vacation_hours,
48 e.sickleavehours as sick_leave_hours,
49 e.currentflag as employee_current_flag,
50 e.rowguid as row_guid,
51 e.modifieddate as employee_modified_date,
52 datediff(year, hiredate, current_date()) as years_since_hire,
53 current_department_id,
54 current_shift_id,
55 current_department_start_date
56 from employees e
57 left join current_department d
58 on e.businessentityid=d.businessentityid
59)
60
61 select * from final
Preview
Compile
Query Results
Compiled SQL
Lineage

[DBT]

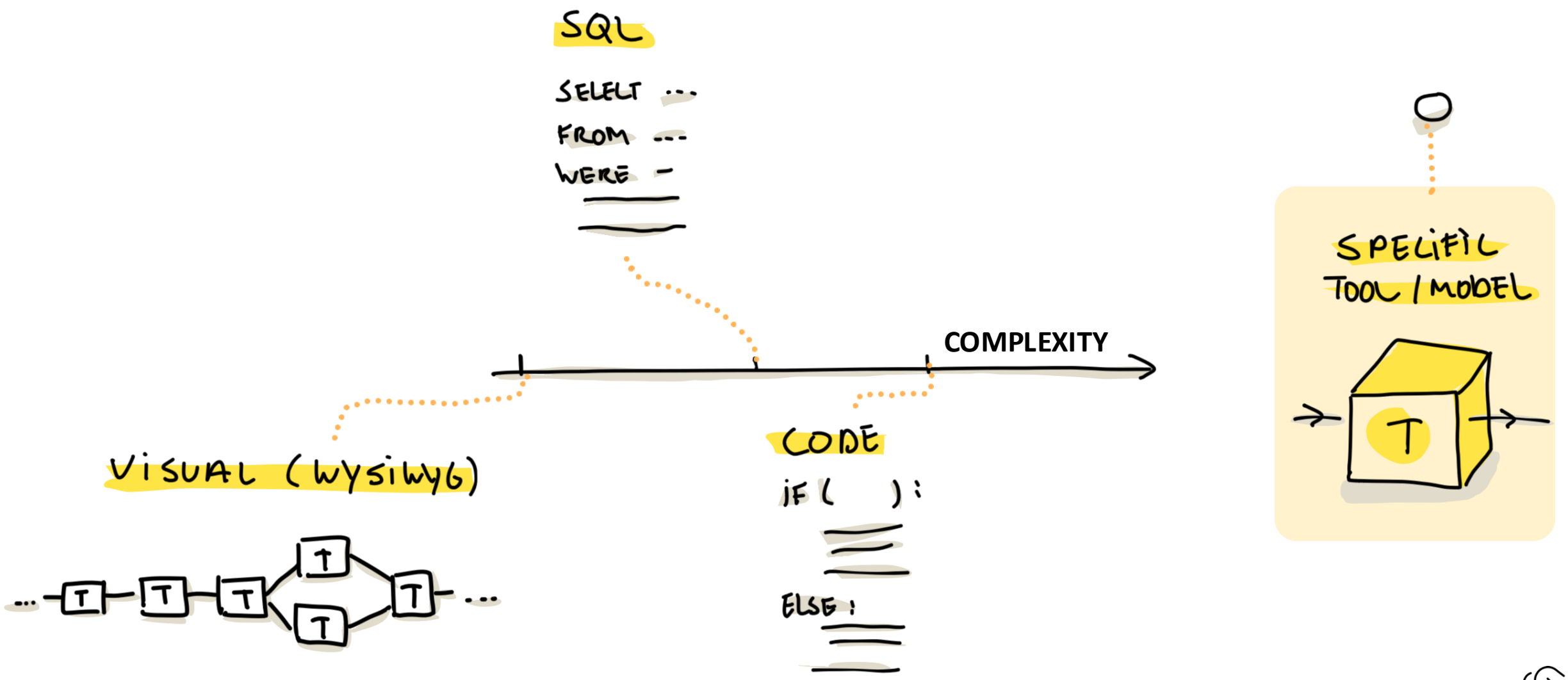
Data Transformation Tool Flavors



```python
import time
from sklearn.datasets import make_classification
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression

# Set training and validation sets
X, y = make_classification(n_samples=1000000, n_features=1000, n_classes = 2)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=10000)

# Solvers
solvers = ['liblinear', 'saga']

for sol in solvers:
    start = time.time()
    logreg = LogisticRegression(solver=sol)
    logreg.fit(X_train, y_train)
    end = time.time()
    print(sol + " Fit Time: ",end-start)
```

```
liblinear Fit Time:  2424.526051044464
saga Fit Time:  133.299968957901
```

Data Transformation Tool Flavors

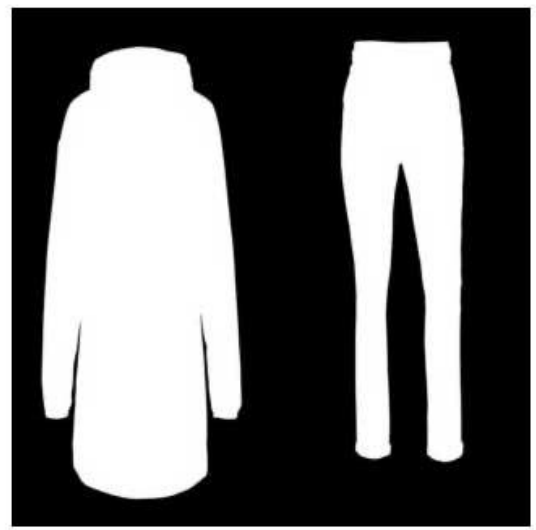


Background-color change

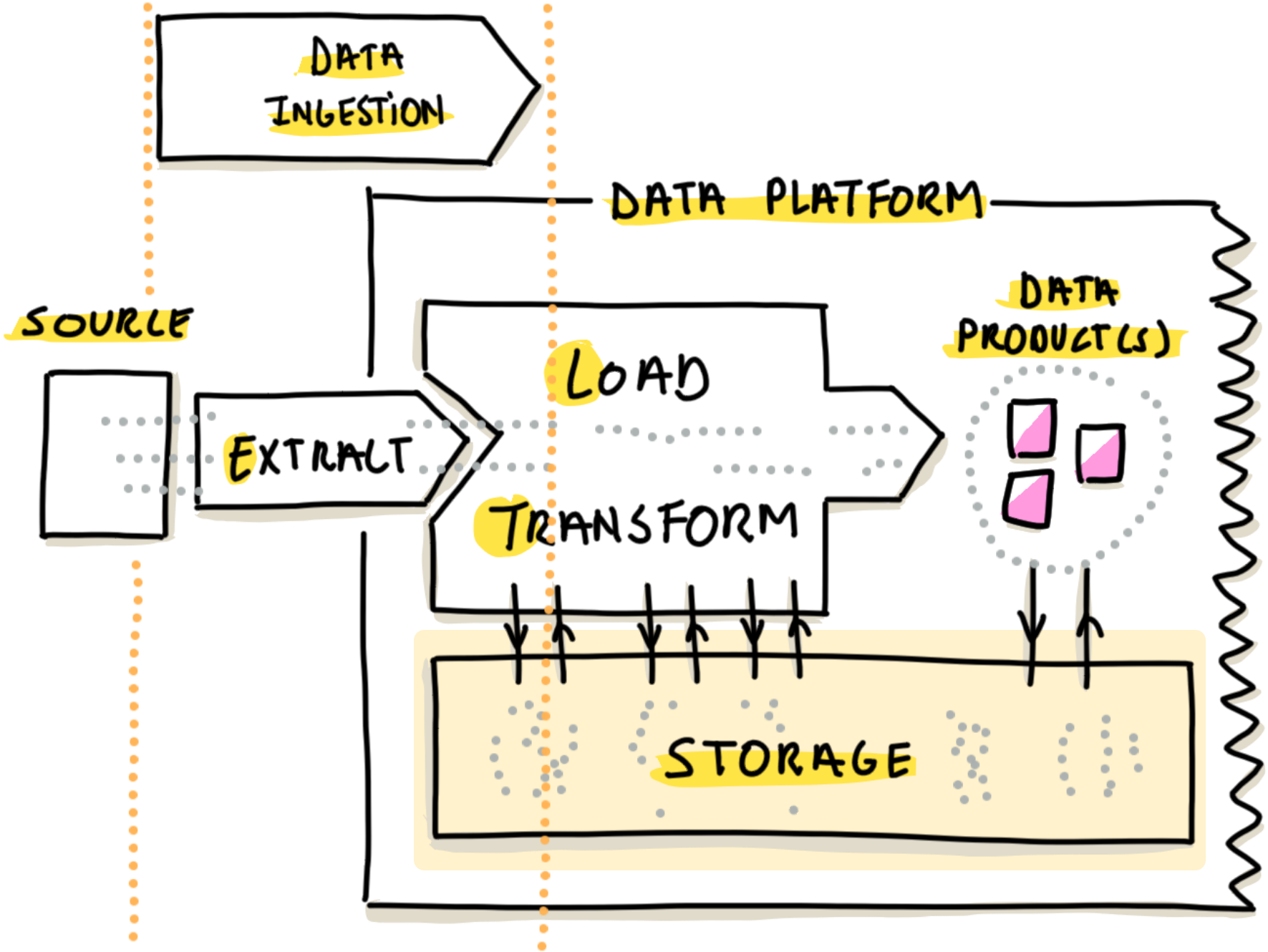
Background-color change

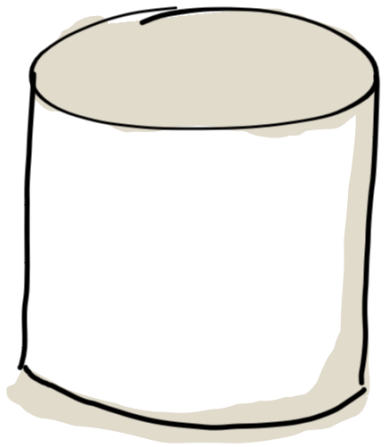
Cropping

Combined images

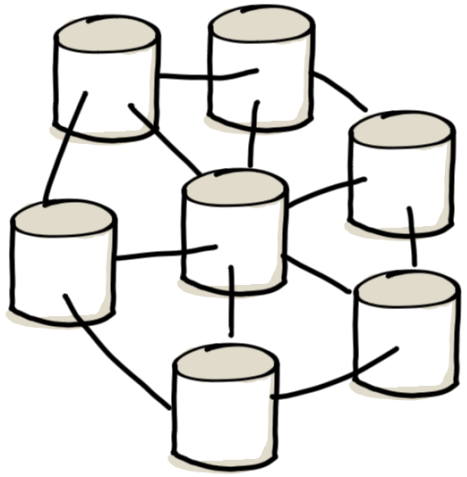


[Automated Background Removal in E-commerce Fashion Image Processing Using PyTorch on Databricks]

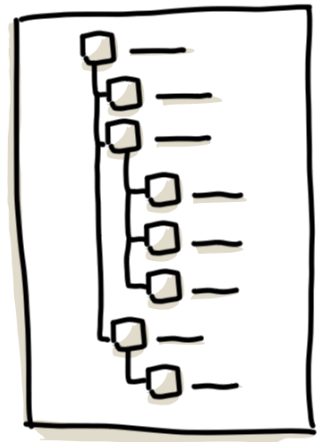

DATA INGESTION
DATA PLATFORM
SOURCE
EXTRACT
LOAD
TRANSFORM
DATA PRODUCT(S)
STORAGE



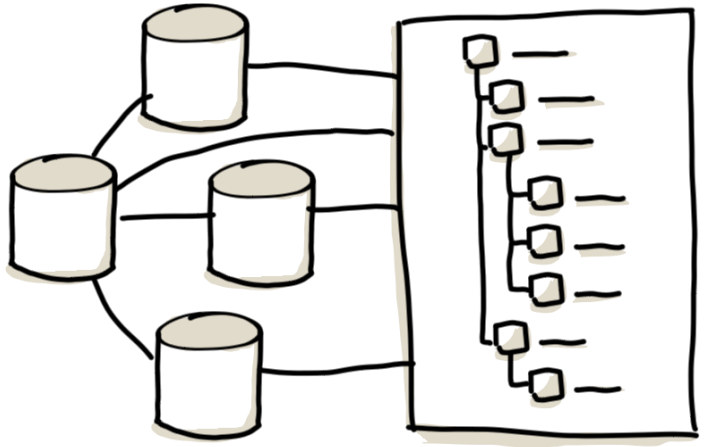
DATABASE



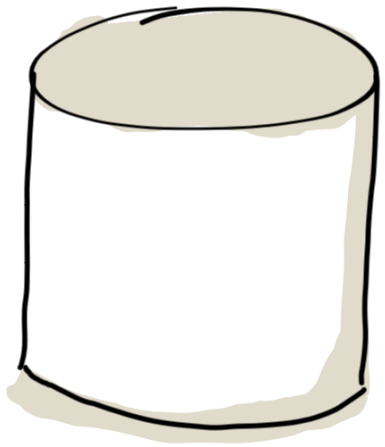
ANALYTICAL DATABASE



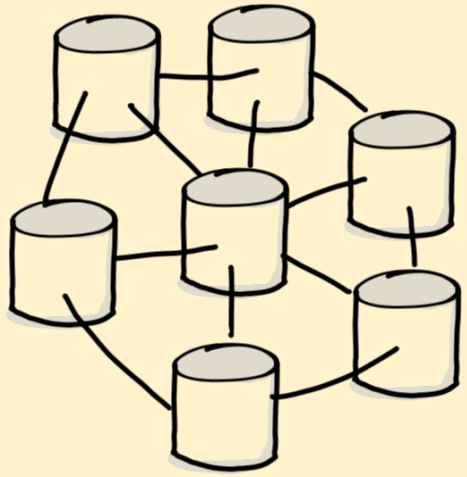
DATA LAKE



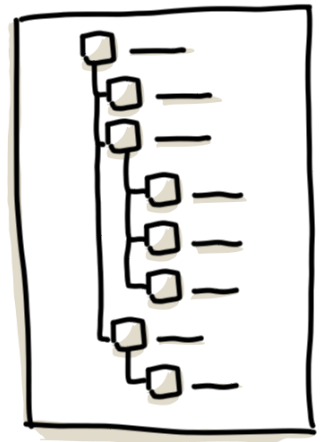
DATA LAKEHOUSE



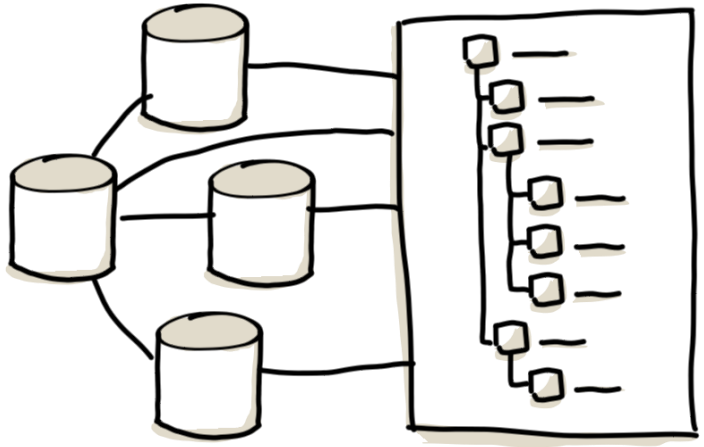
DATABASE



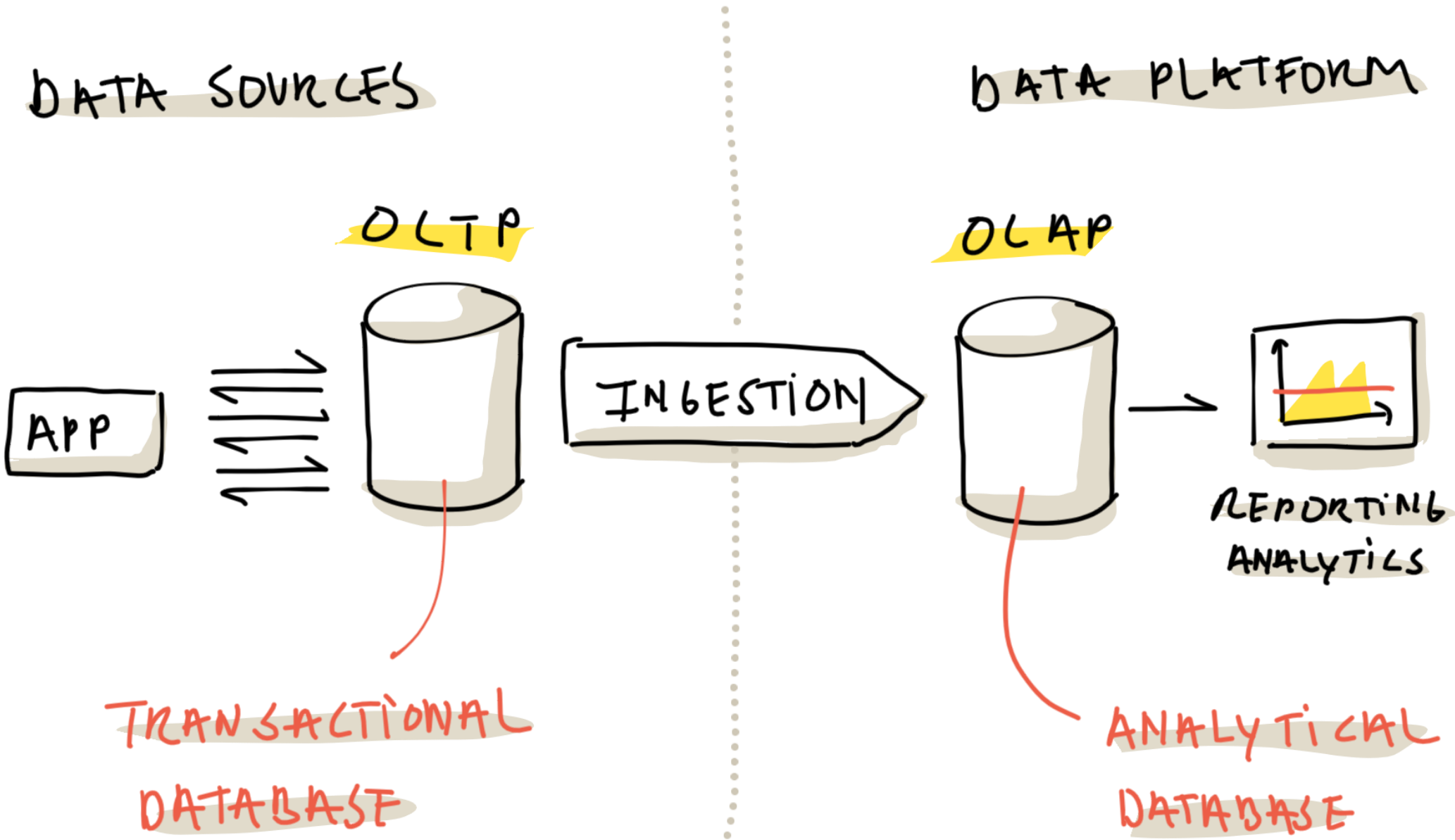
ANALYTICAL DATABASE



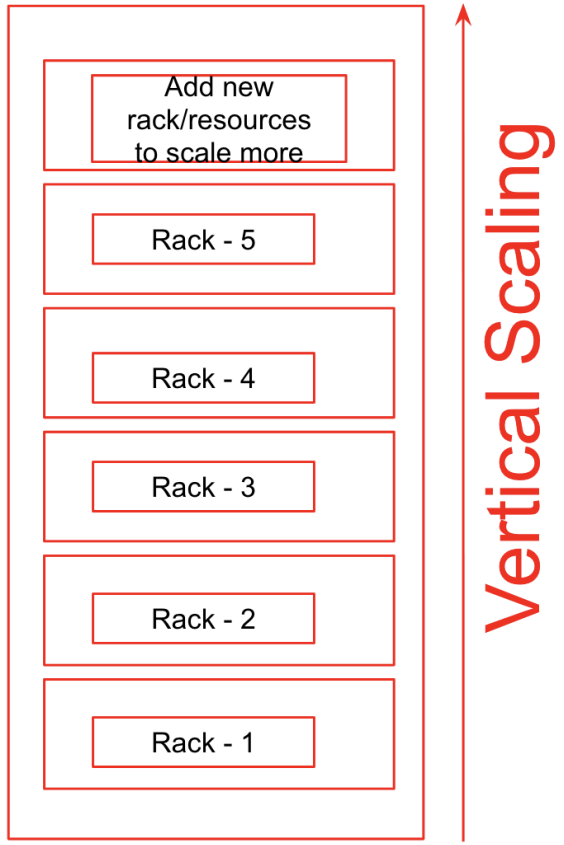
DATA LAKE



DATA LAKEHOUSE


DATA SOURCES
DATA PLATFORM
OLTP
OLAP
APP
INGESTION
REPORTING
ANALYTICS
TRANSACTIONAL
DATABASE
ANALYTICAL
DATABASE

A DWH Database (often called ‘Cloud DWH’) is tuned for **horizontal scaling**.

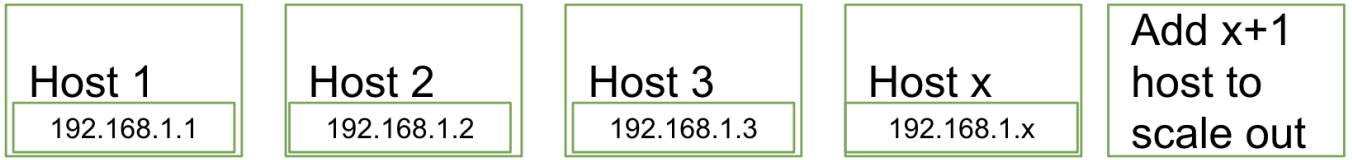


Vertical Scaling

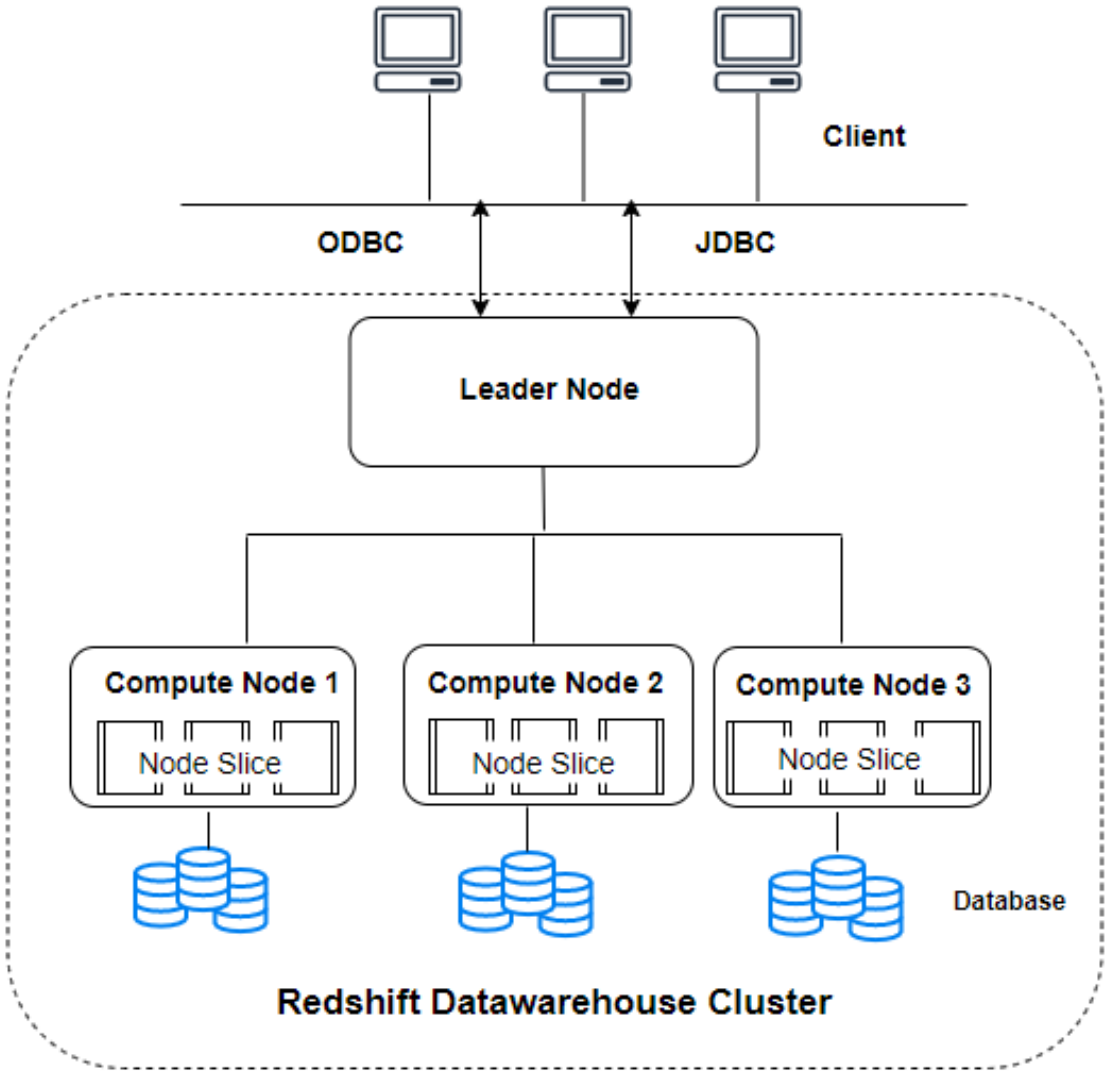
To scale more, Add more RAM, CPU, Memory to the **one existing machine**

Horizontal Scaling

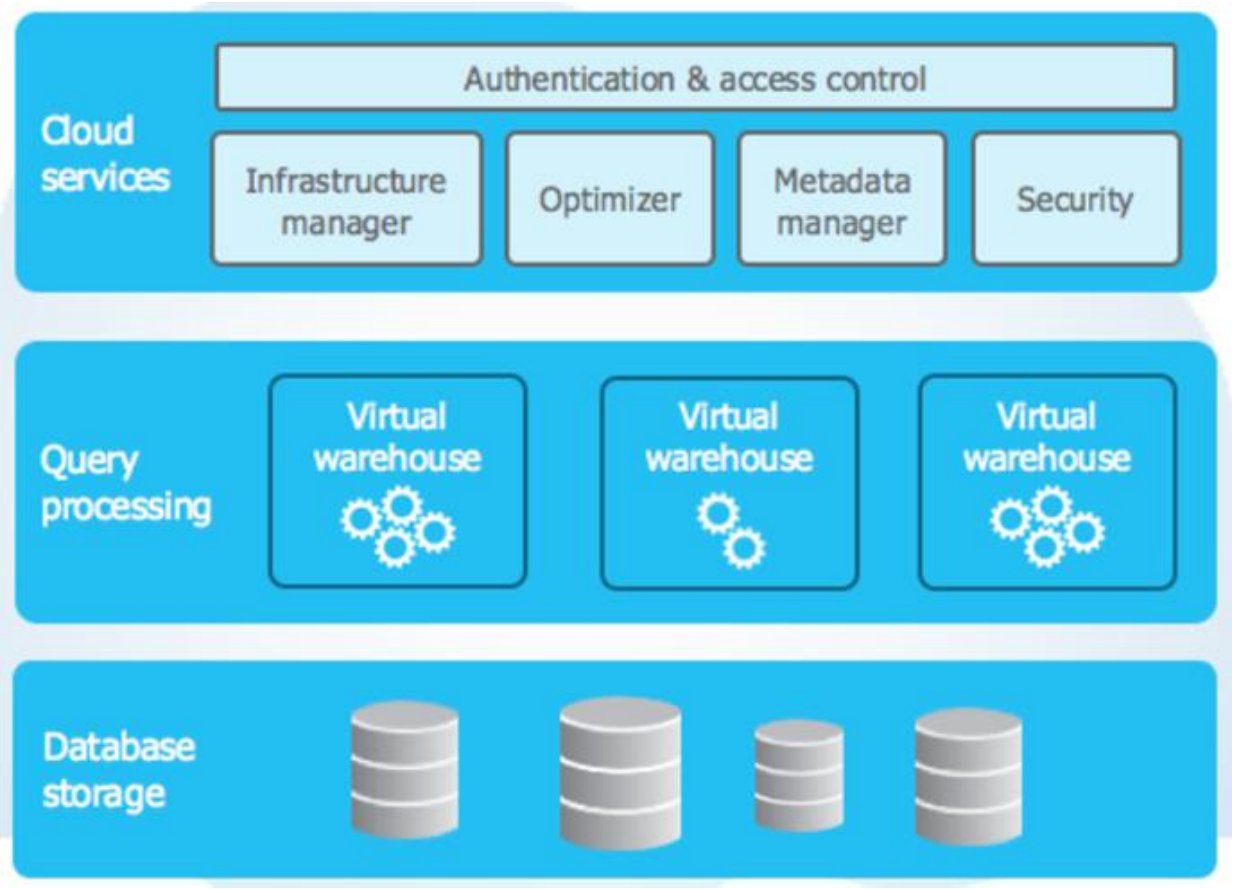
To scale more: Add more machines to existing **group of distributed system**



Example: Amazon Redshift



Example: Snowflake

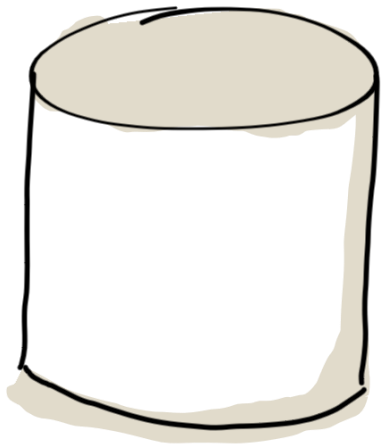



snowflake

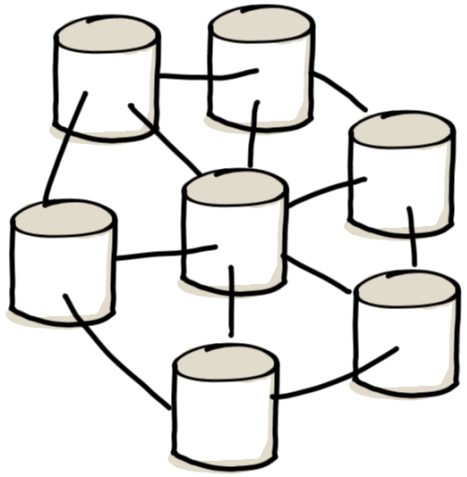

amazon
REDSHIFT


databricks

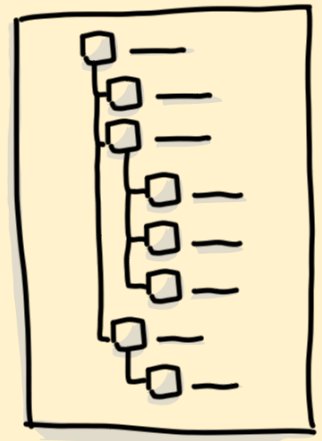

BigQuery



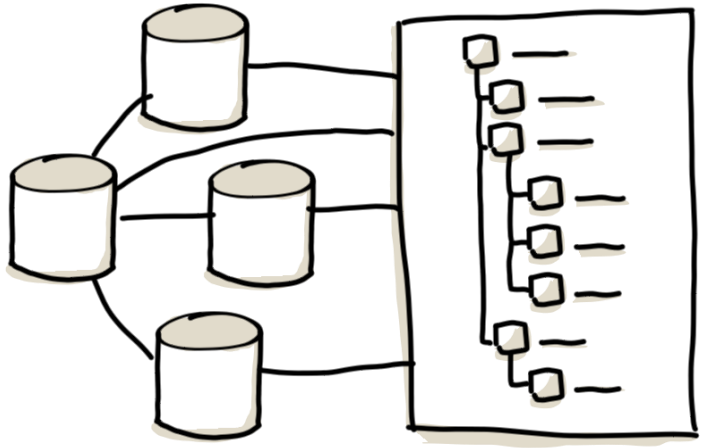
DATABASE



ANALYTICAL DATABASE

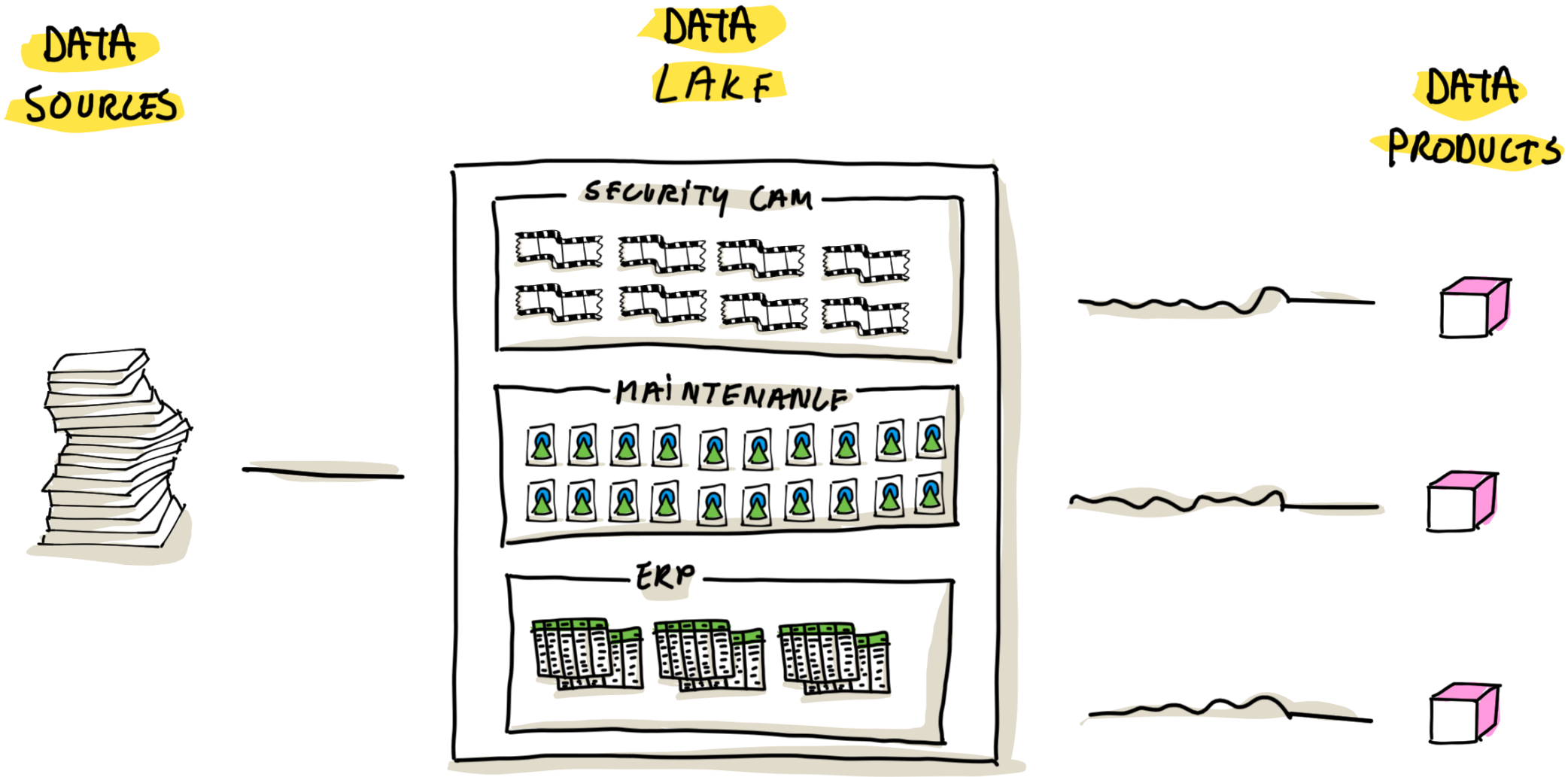


DATA LAKE

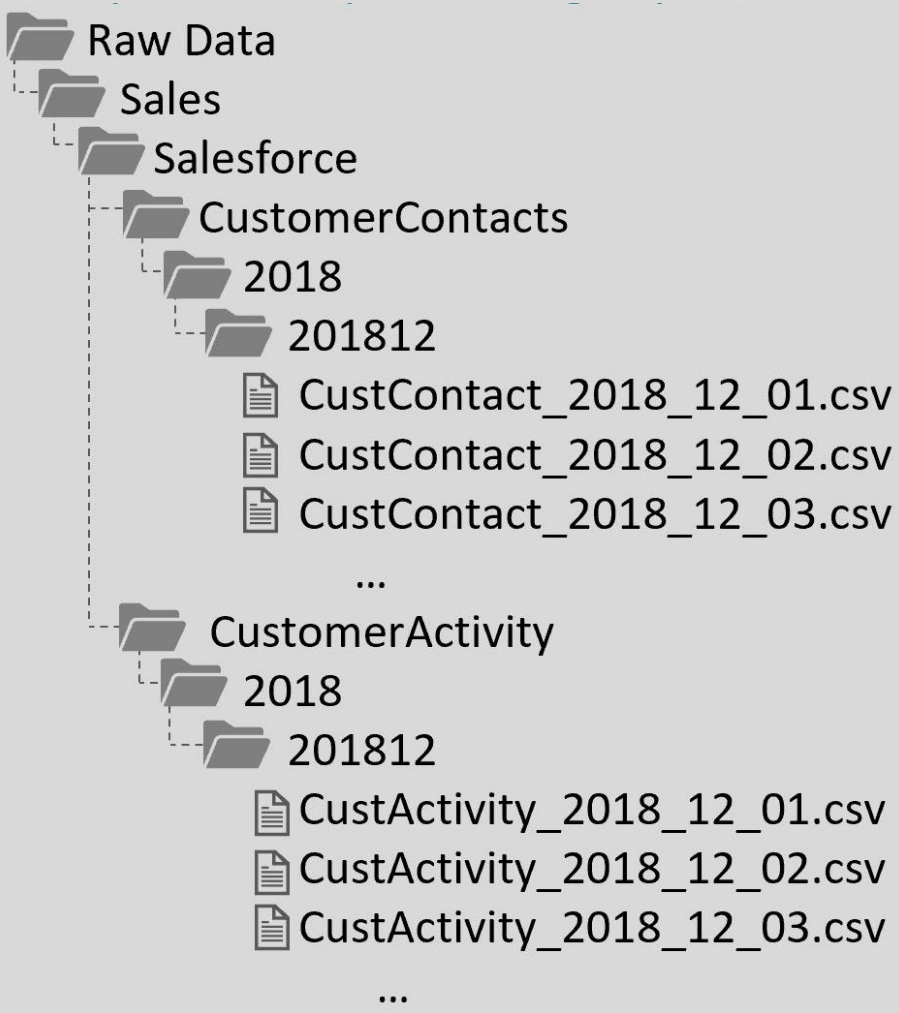


DATA LAKEHOUSE

Data Lake

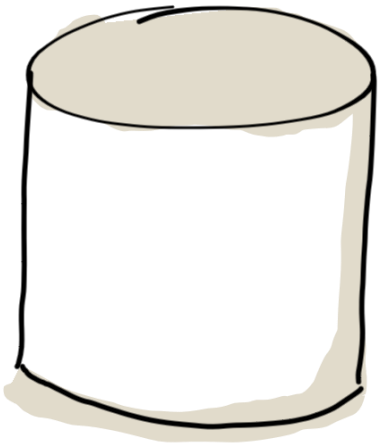

DATA SOURCES
DATA LAKE
DATA PRODUCTS
SECURITY CAM
MAINTENANCE
ERP

Data Lake: A **Structured** File Repository

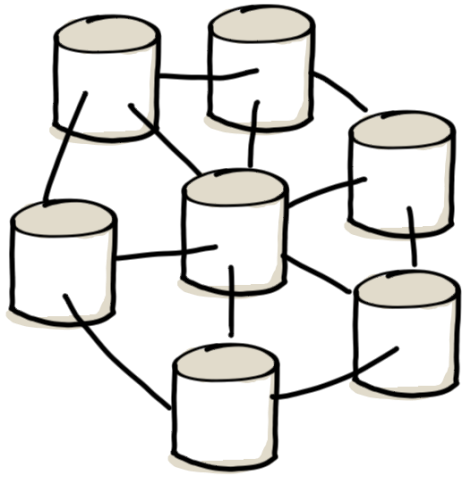


DELTA LAKE

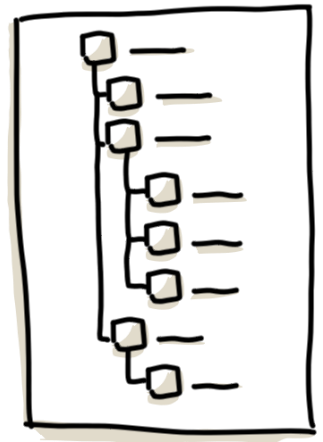
Parquet



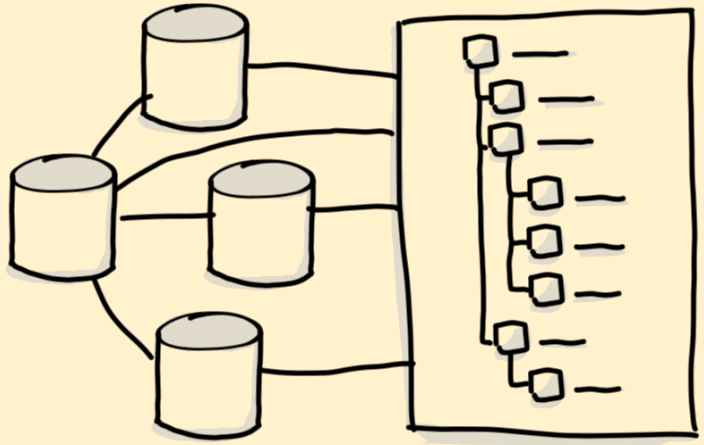
DATABASE



ANALYTICAL DATABASE

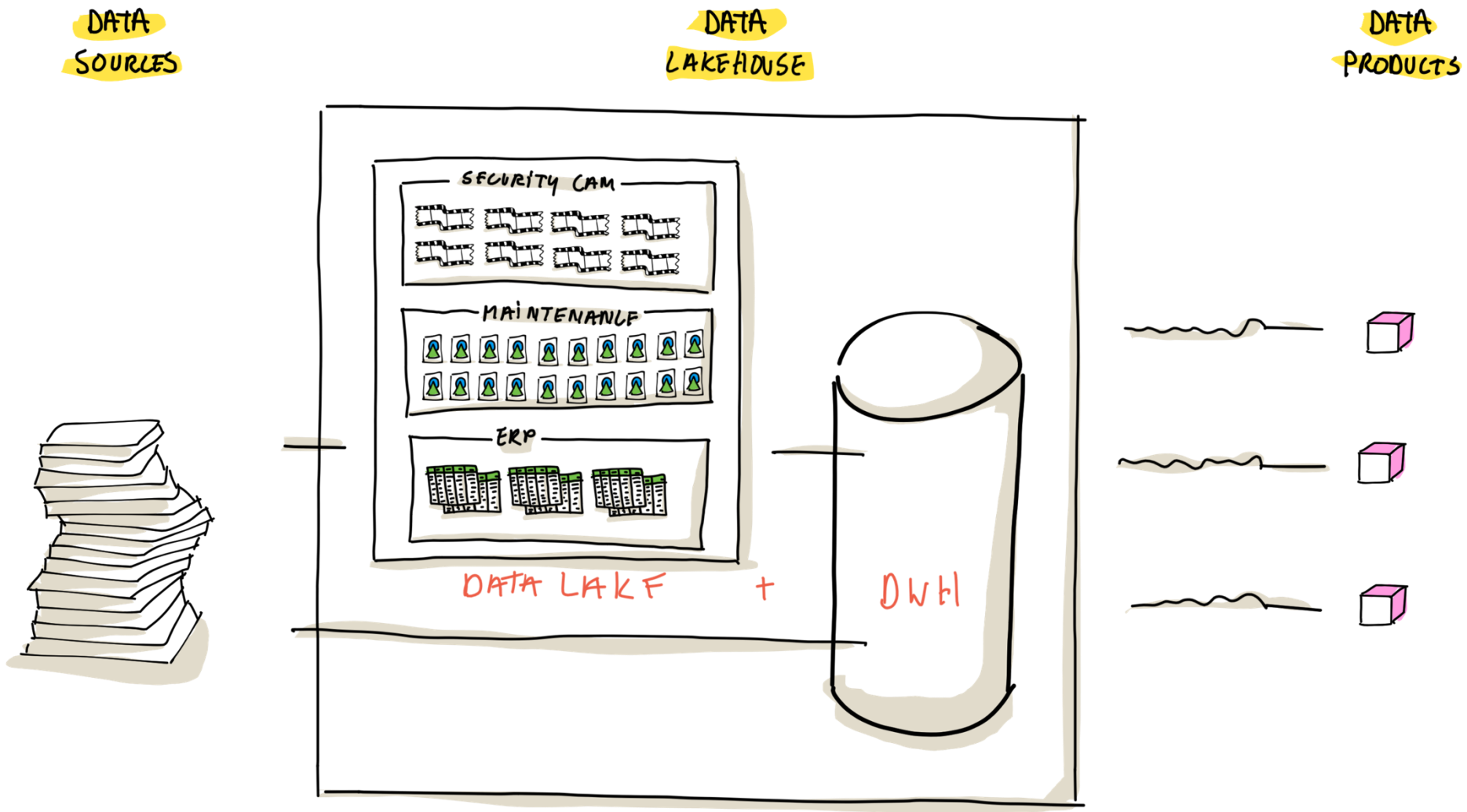


DATA LAKE



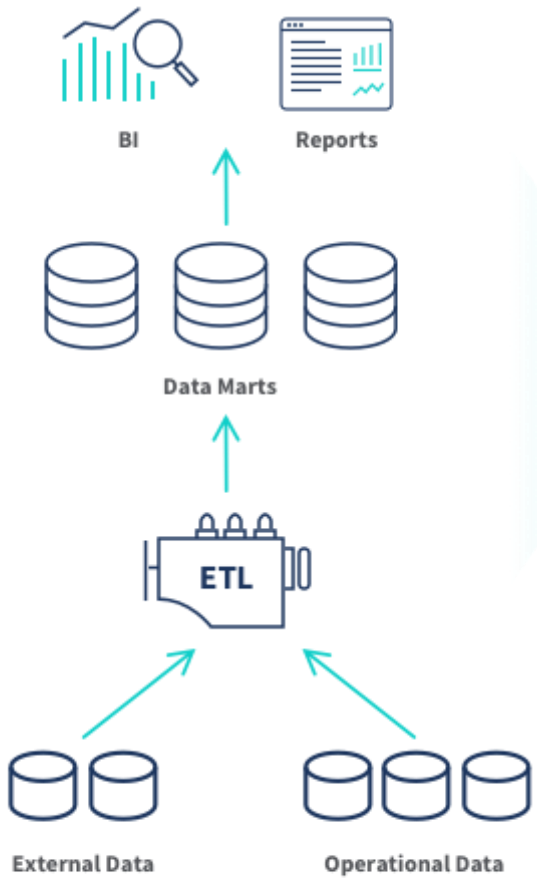
DATA LAKEHOUSE

Data Lakehouse



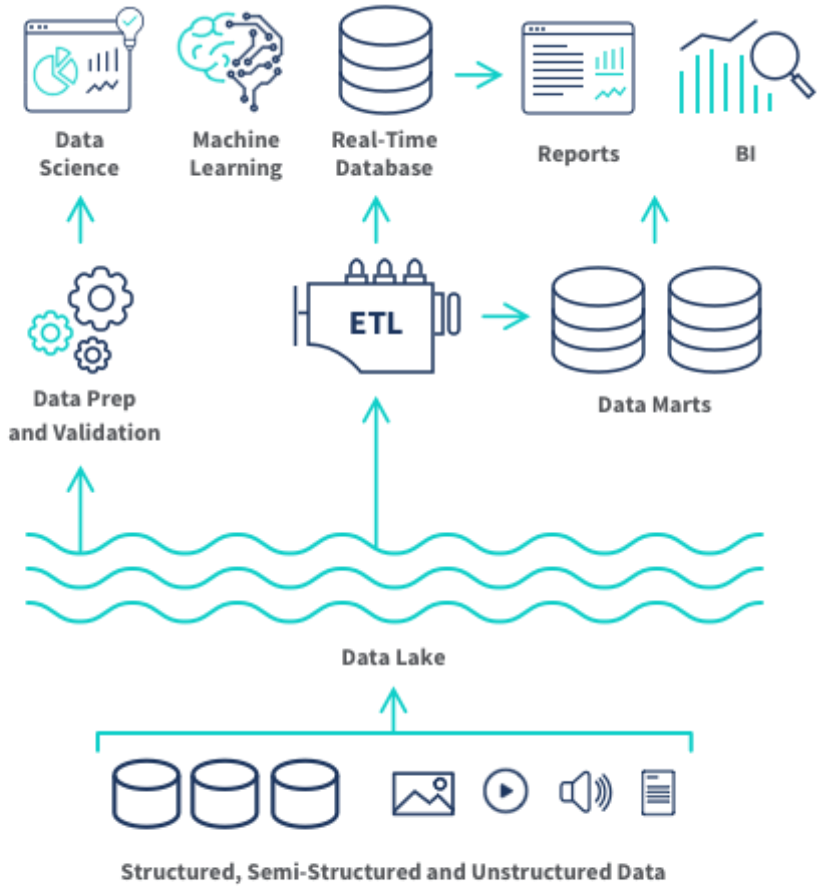
LATE 1980'S

Data Warehouse



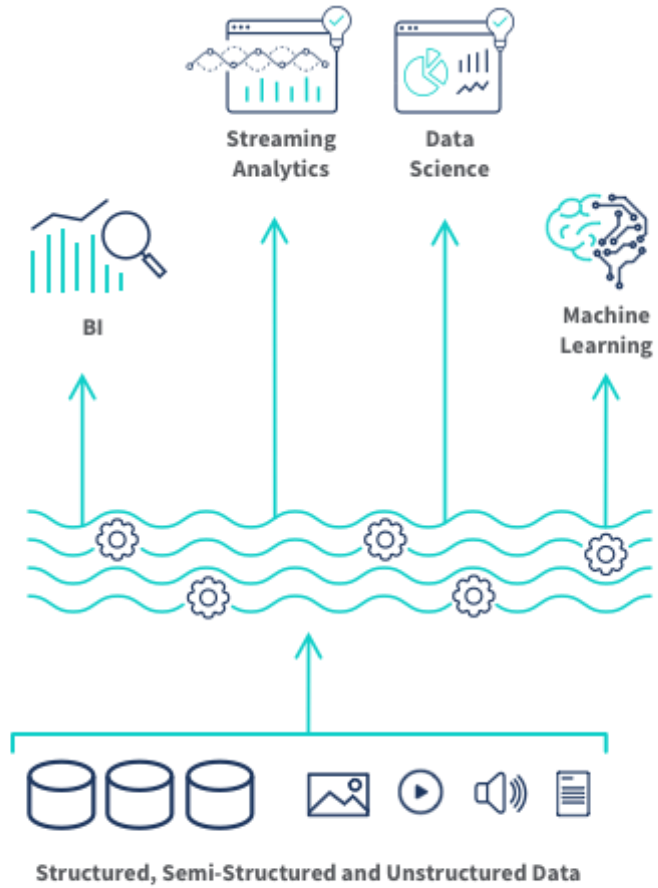
2011

Data Lake



2020

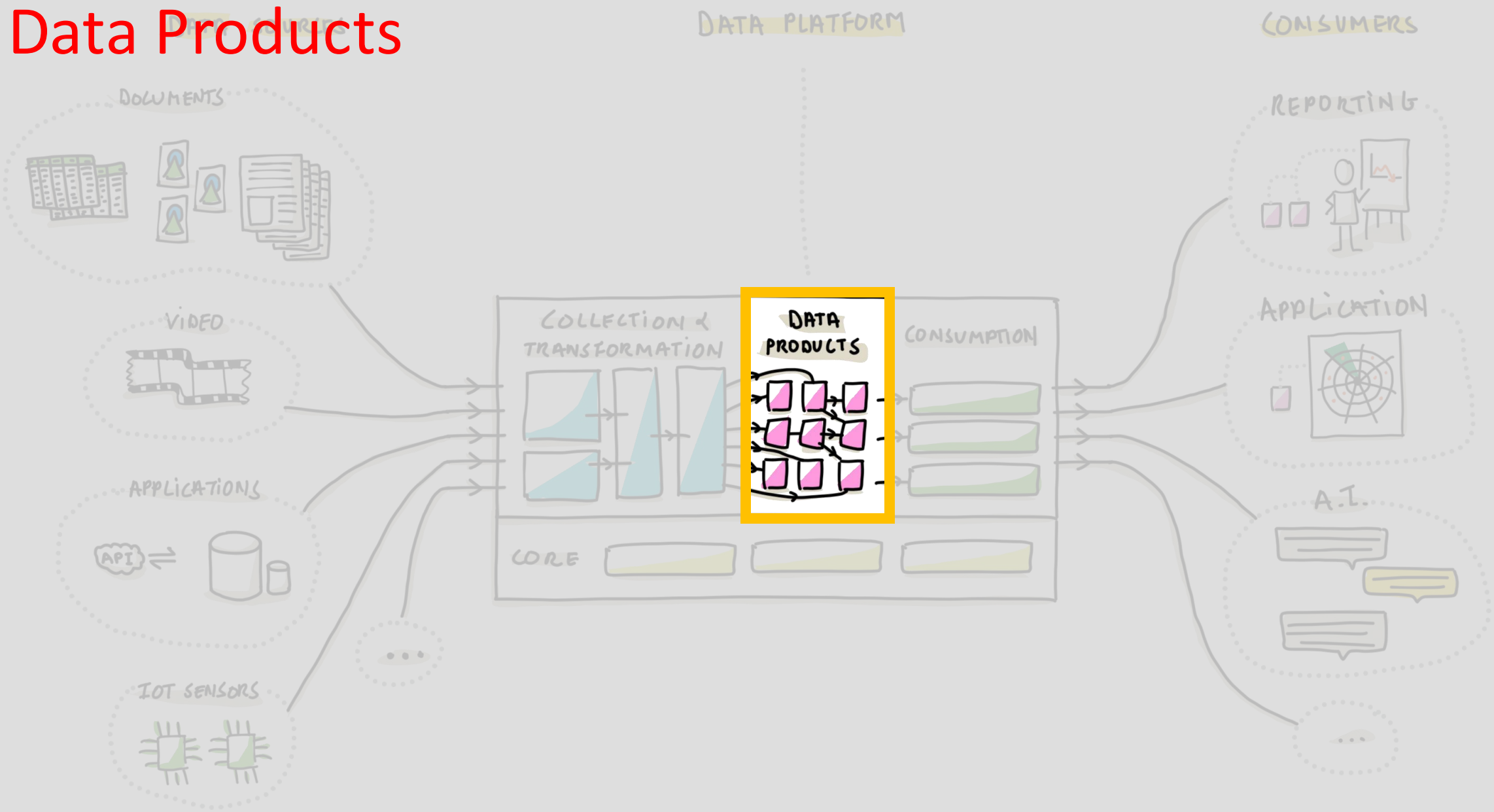
Lakehouse



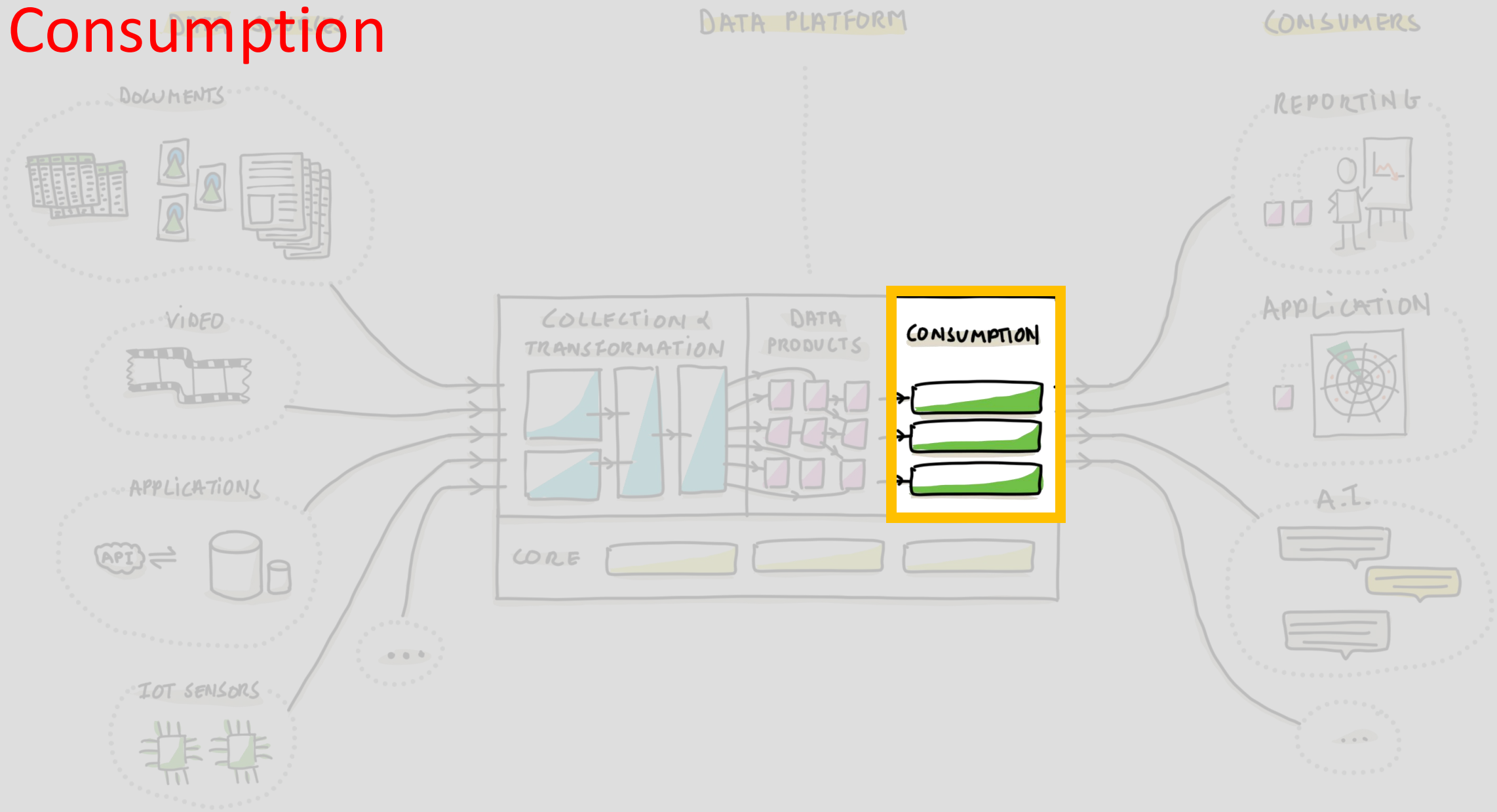
EXERCISE : DIFFERENCES BETWEEN STORAGE TYPES?

	Database	DWH Database	Data Lake	Data Lakehouse	?
Cost					
Scaling					
Volume					
Type of Data					
Performance					
Agility					
Users					
?					

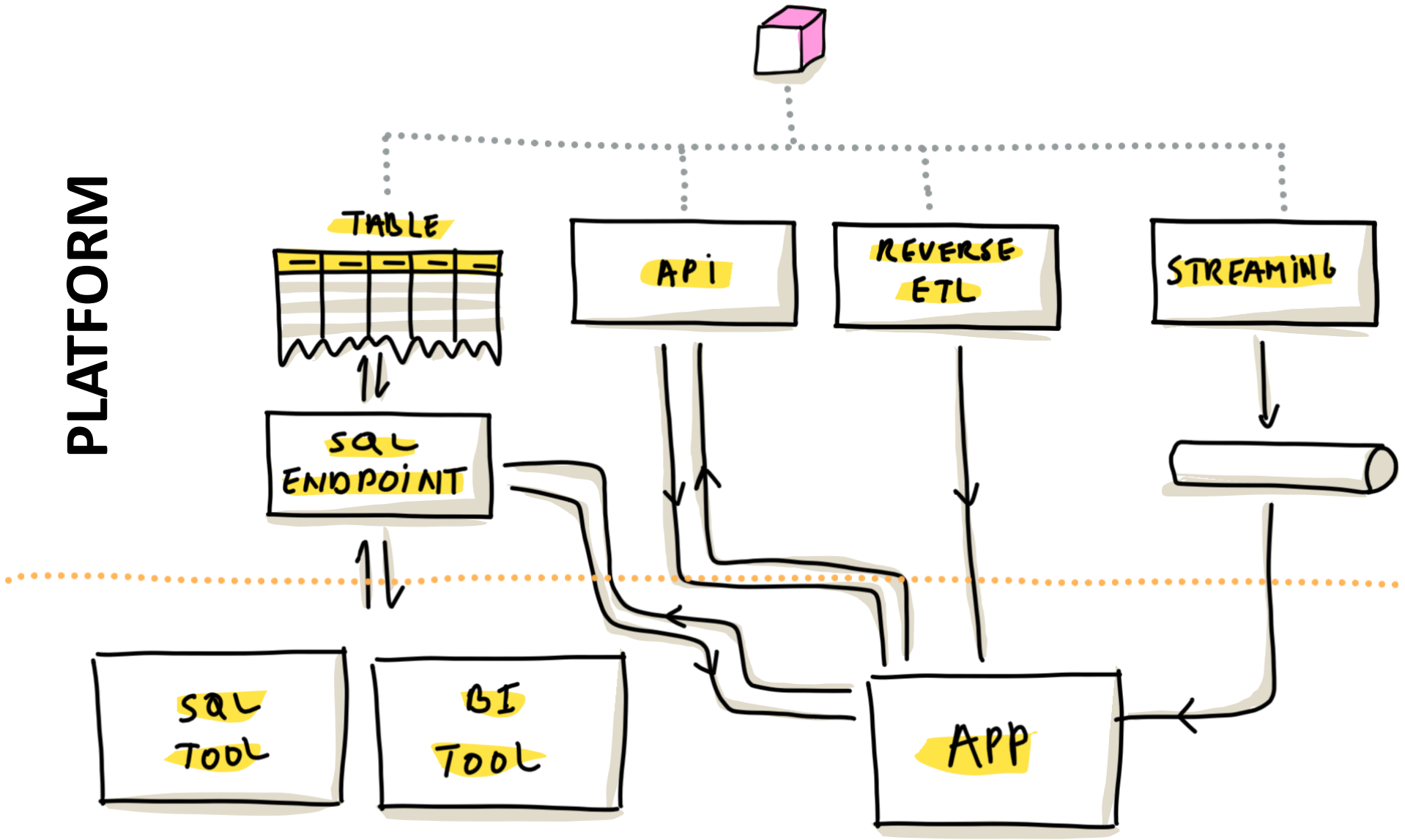
2. Data Products



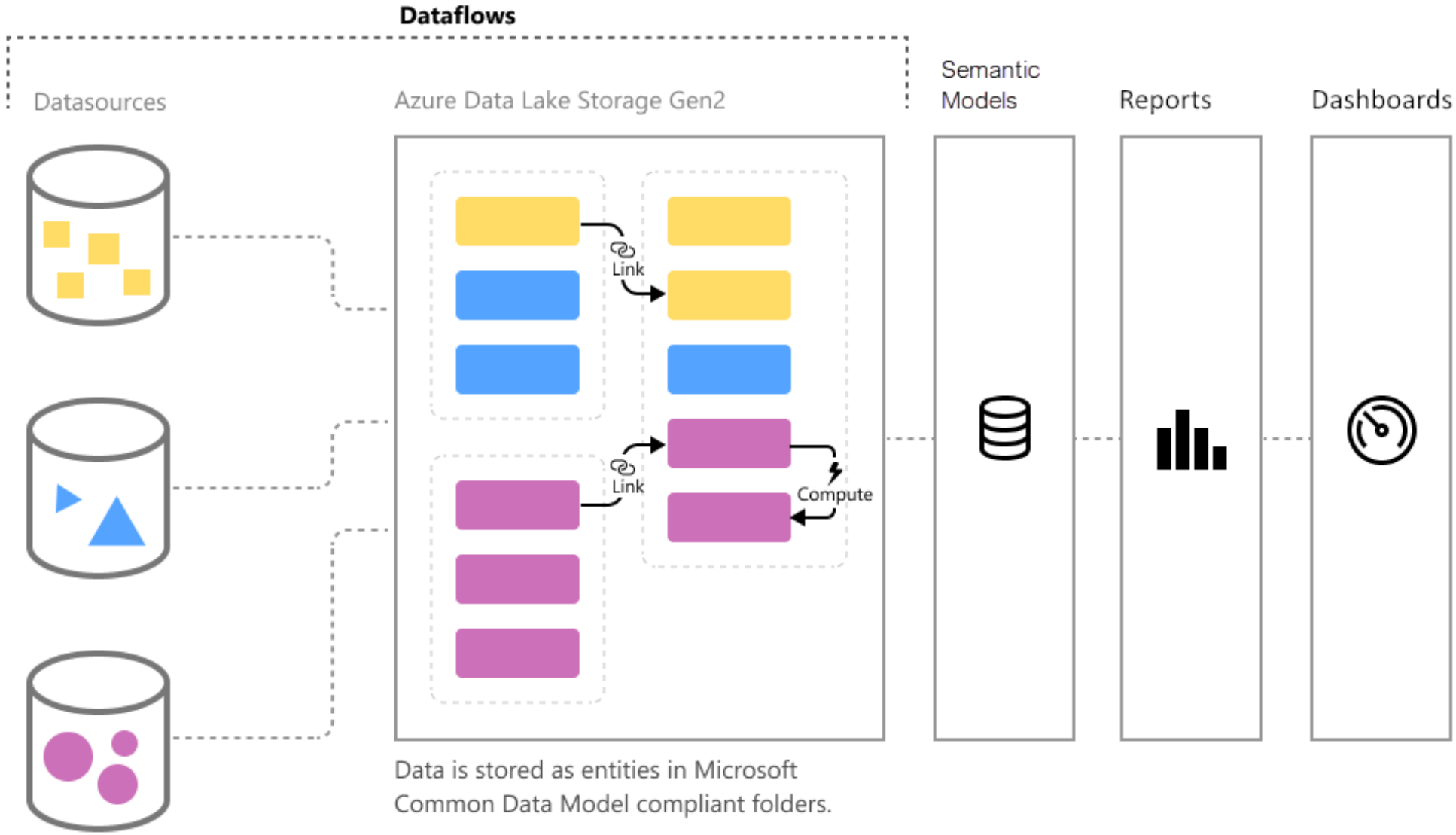
3. Consumption



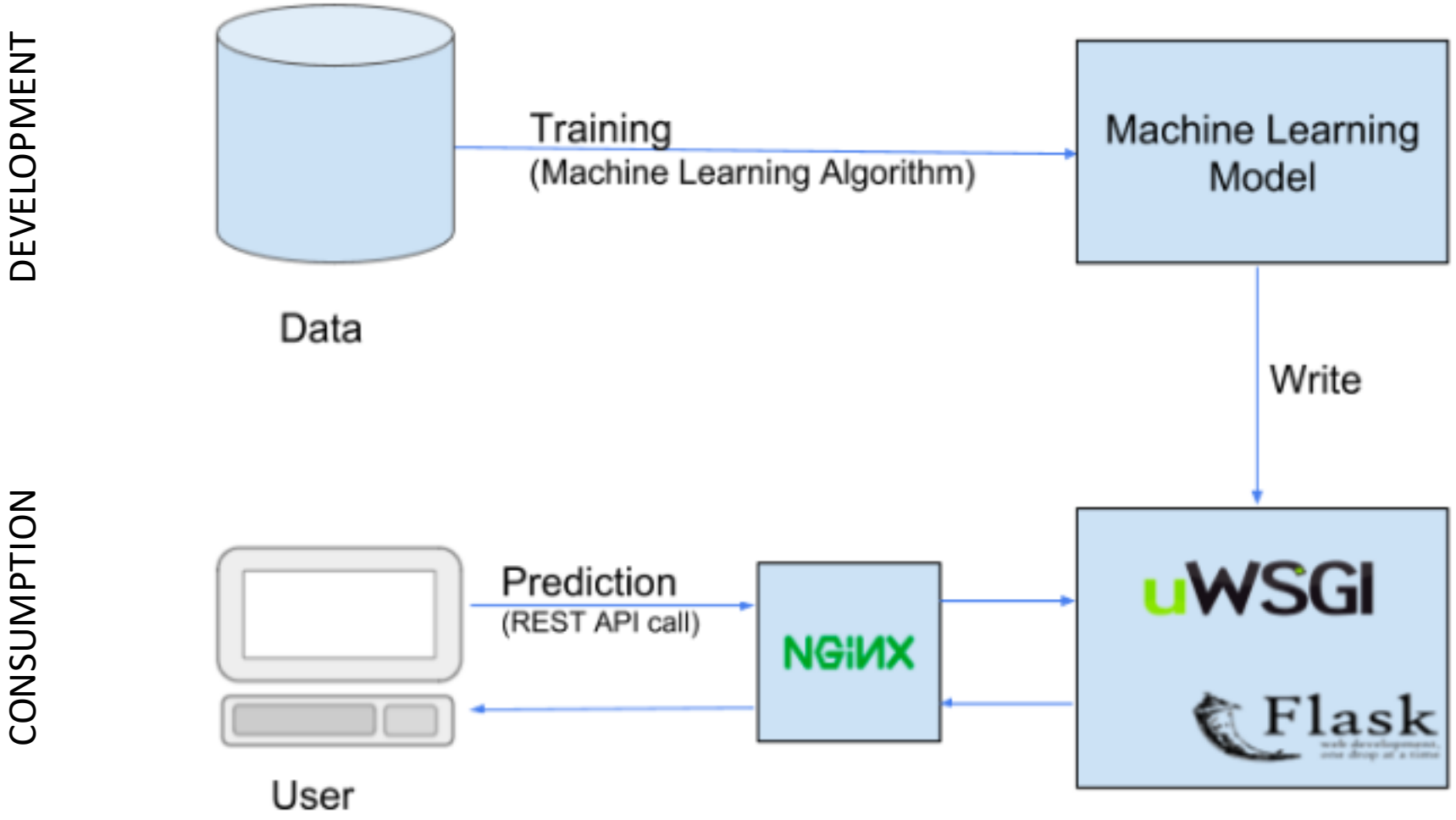
Common Consumption Patterns



Consumption Pattern: Power BI

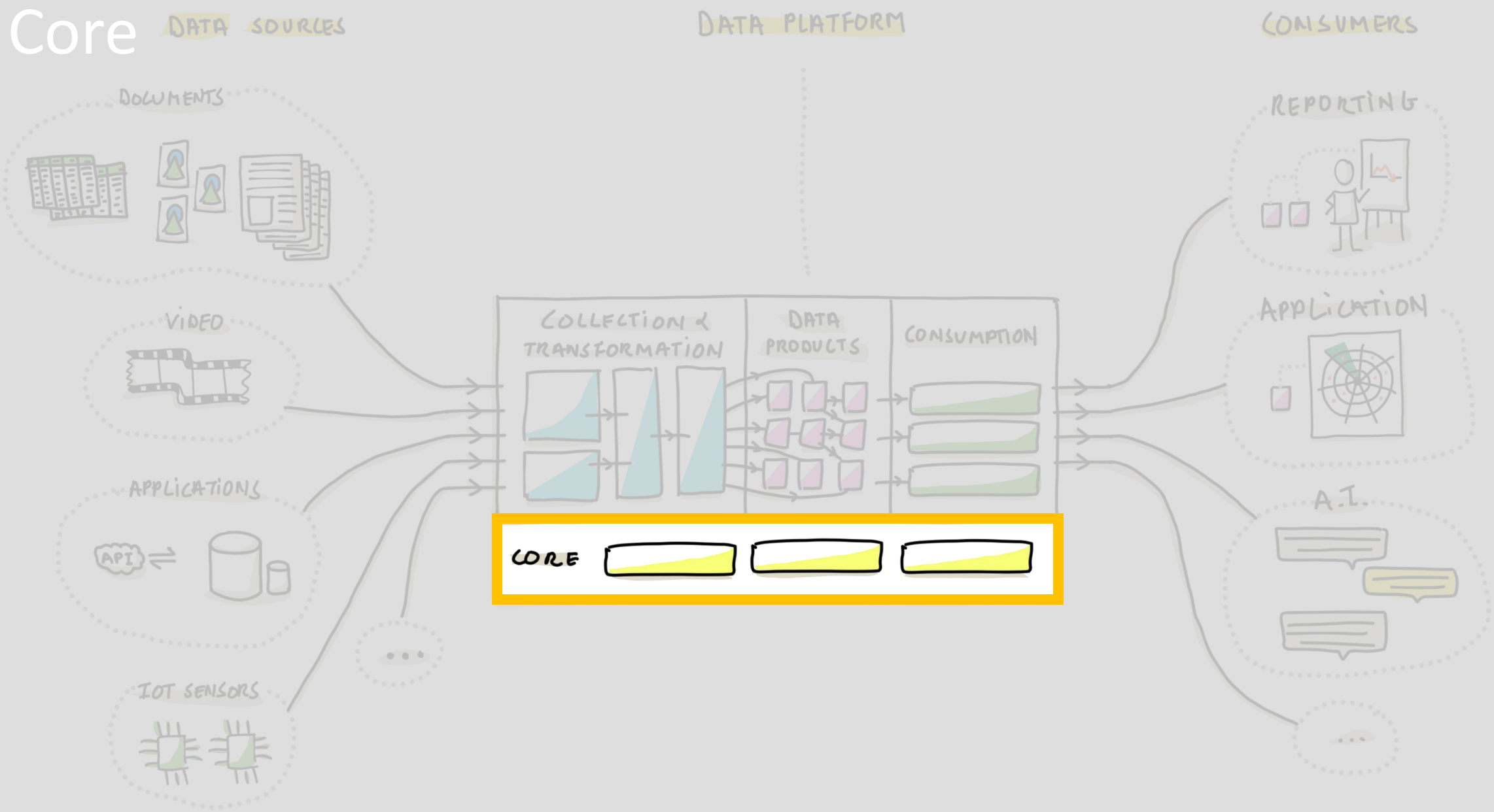


Deploy AI Models as an API

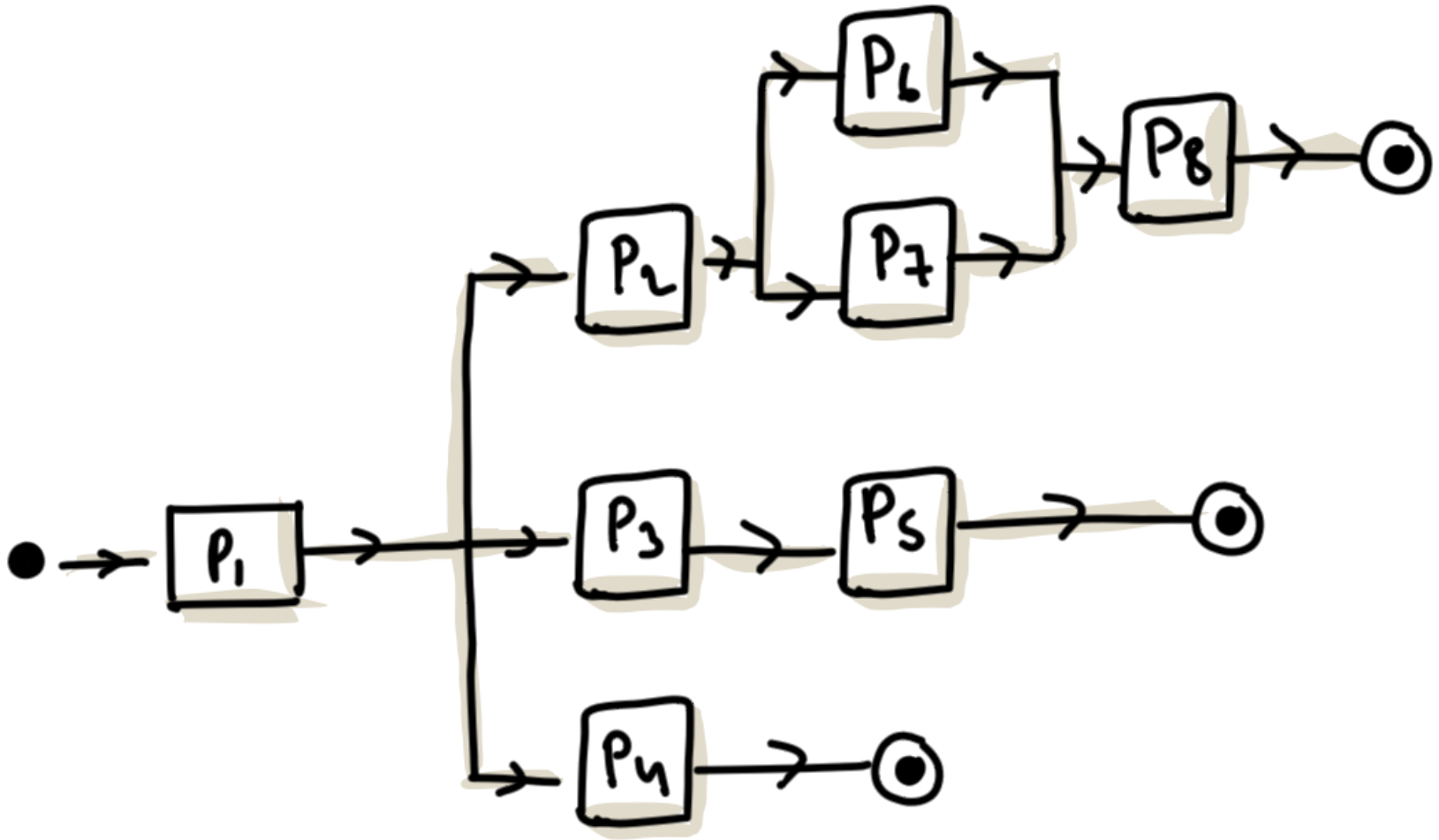


[Deploy ML Models using Flask]

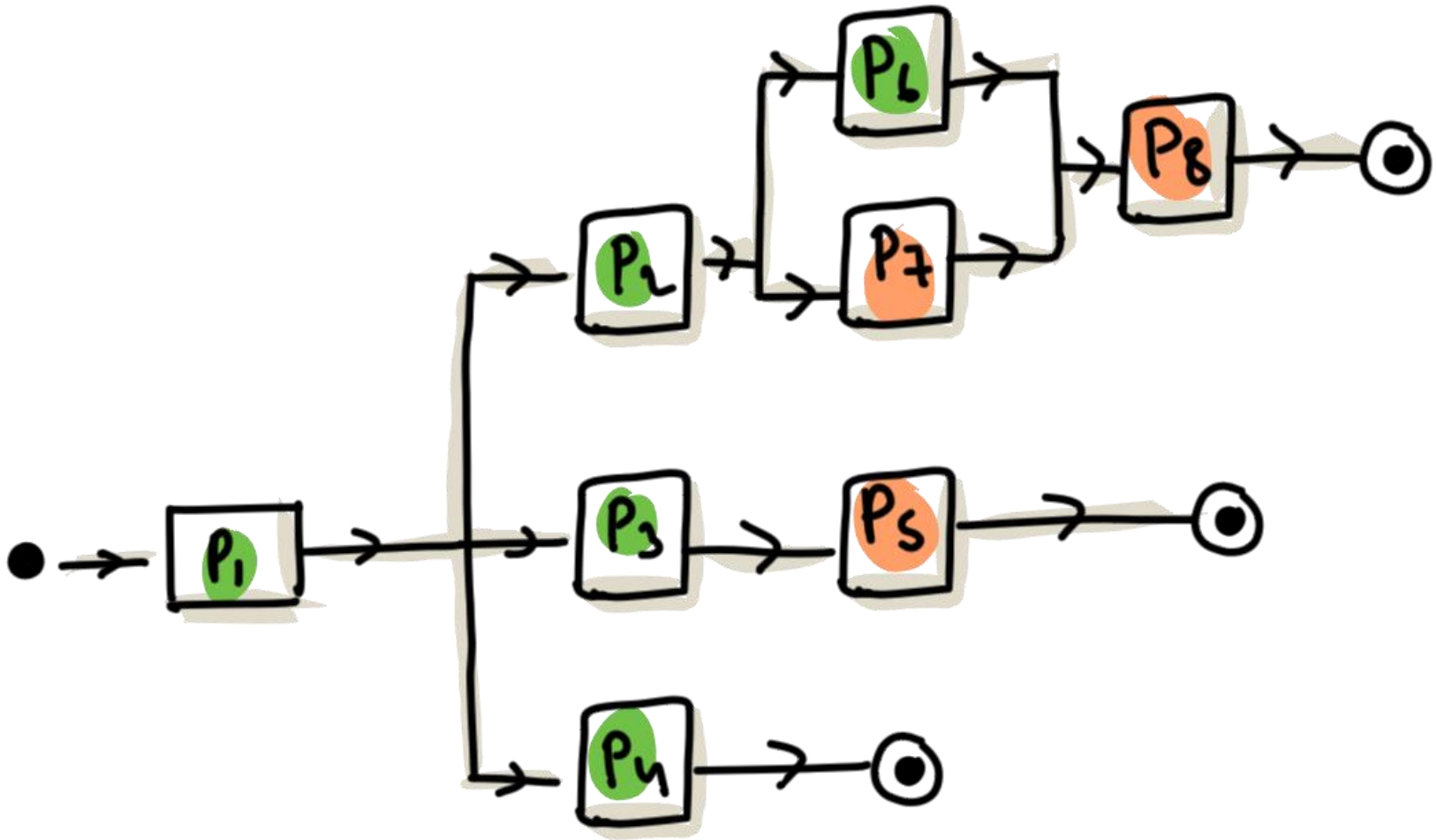
4. Core


DATA SOURCES
DATA PLATFORM
CONSUMERS
DOCUMENTS
VIDEO
APPLICATIONS
API
IOT SENSORS
COLLECTION & TRANSFORMATION
DATA PRODUCTS
CONSUMPTION
CORE
REPORTING
APPLICATION
A.I.
"Operational Plane"
"Analytical Plane"
"Operational / Analytical Plane"

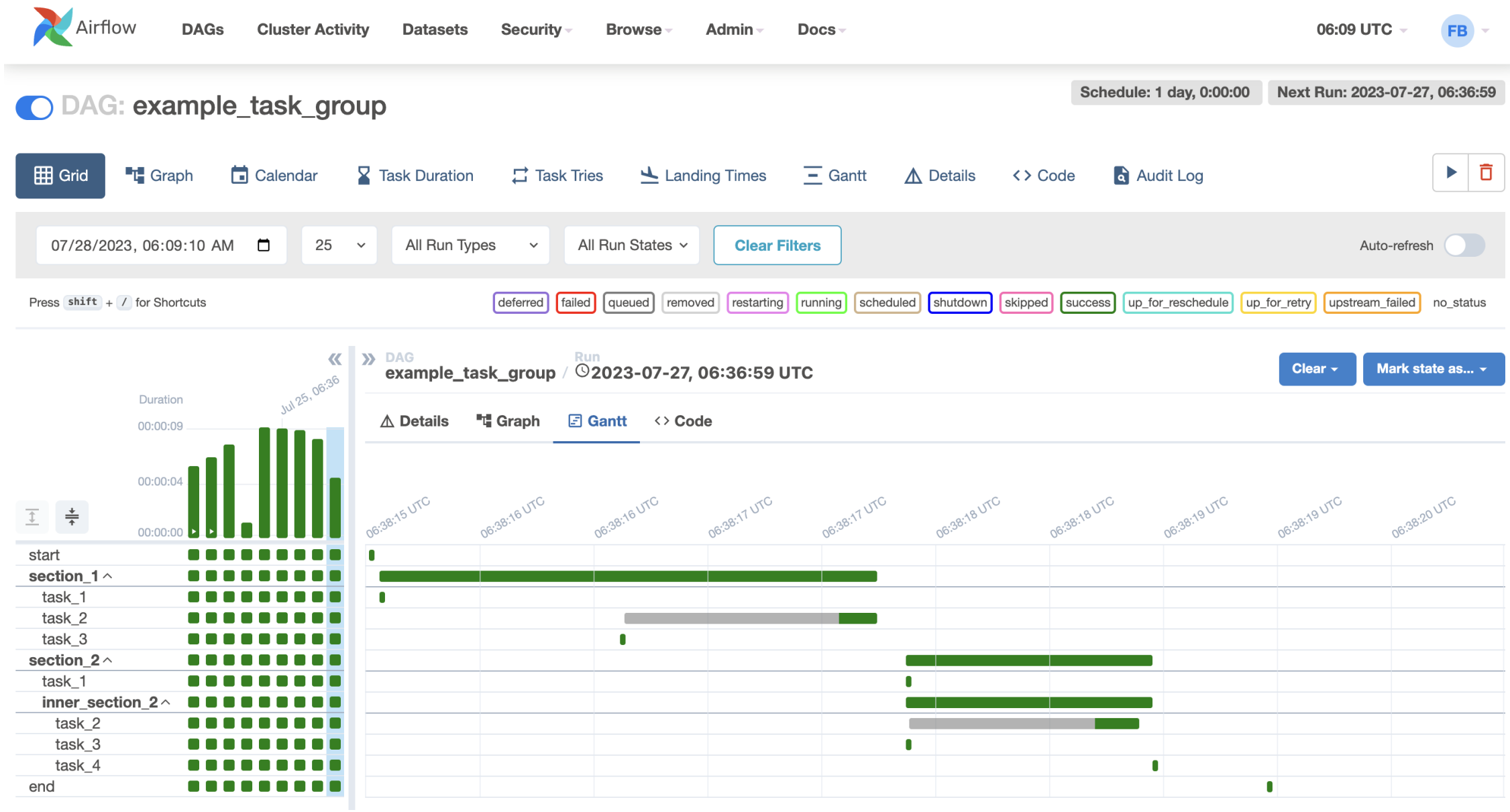
Orchestration



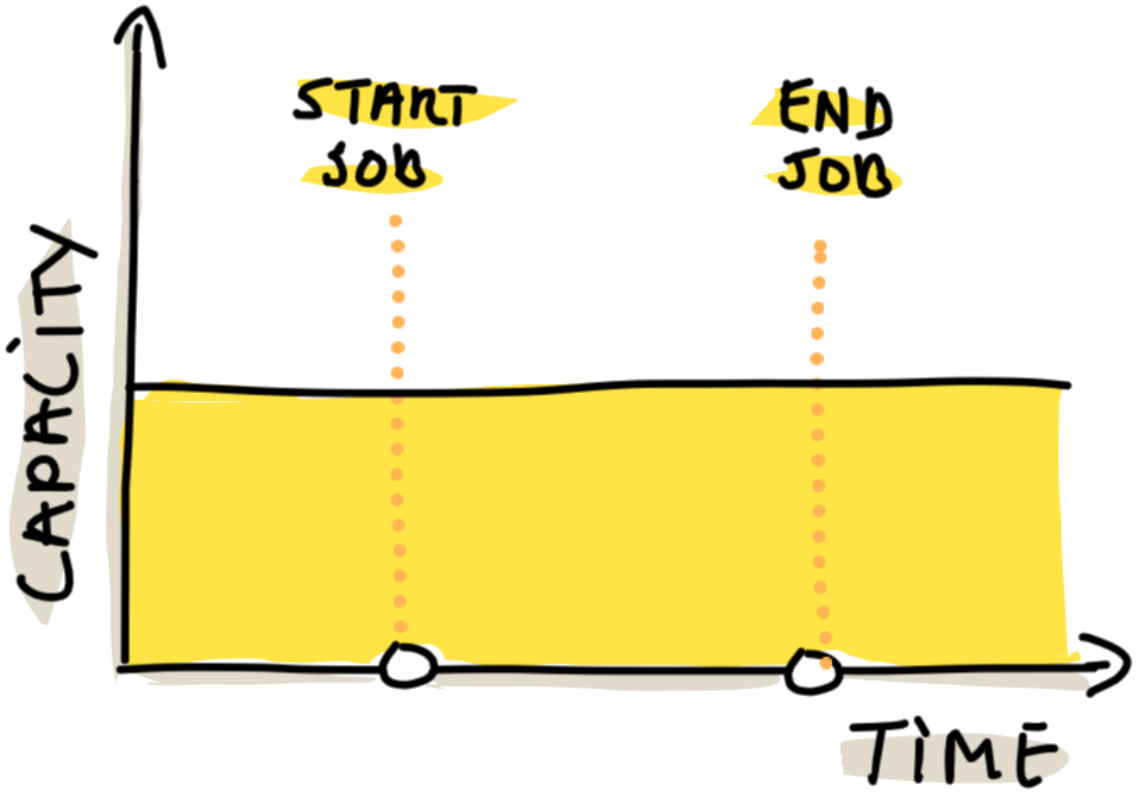
Orchestration



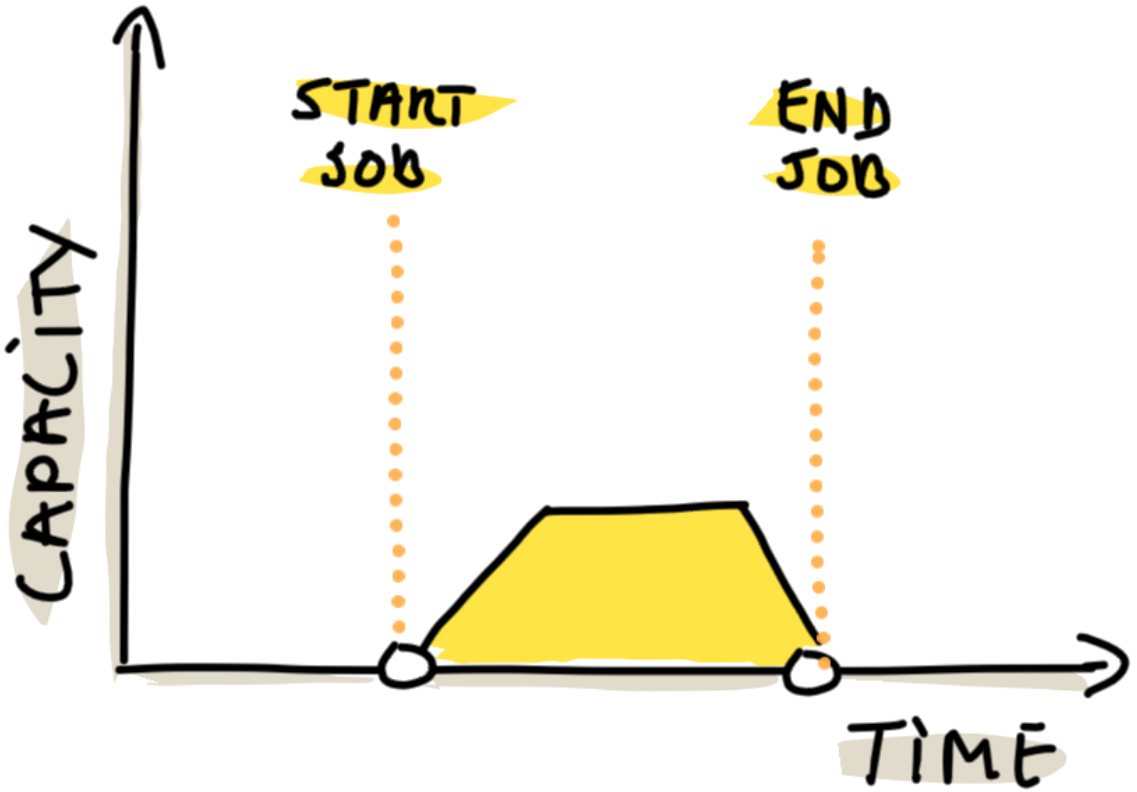
Example: Apache Airflow



Infrastructure Management (Compute)

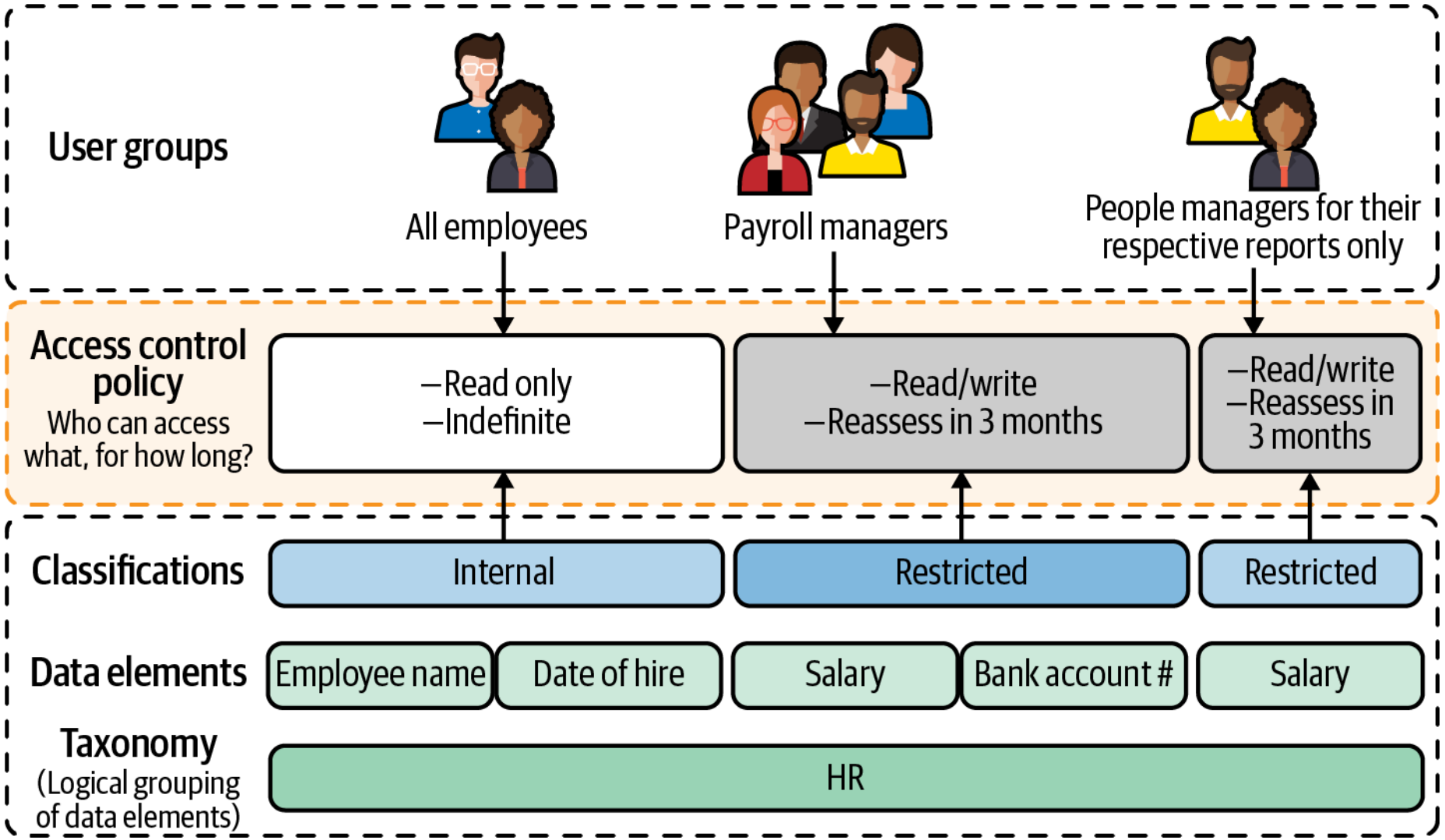


Static



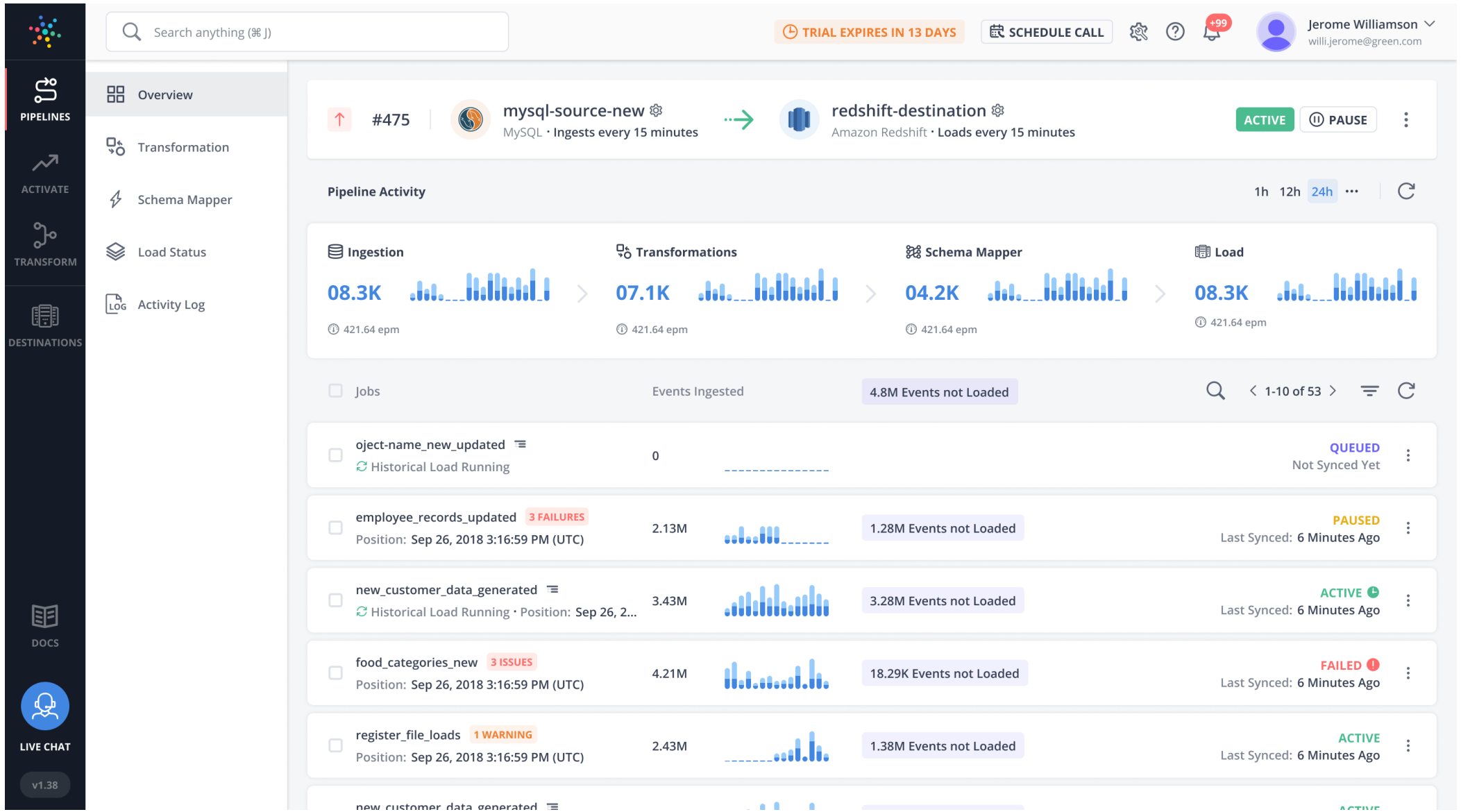
Dynamic

Security & Access Control



[Source]

Monitoring / Observability



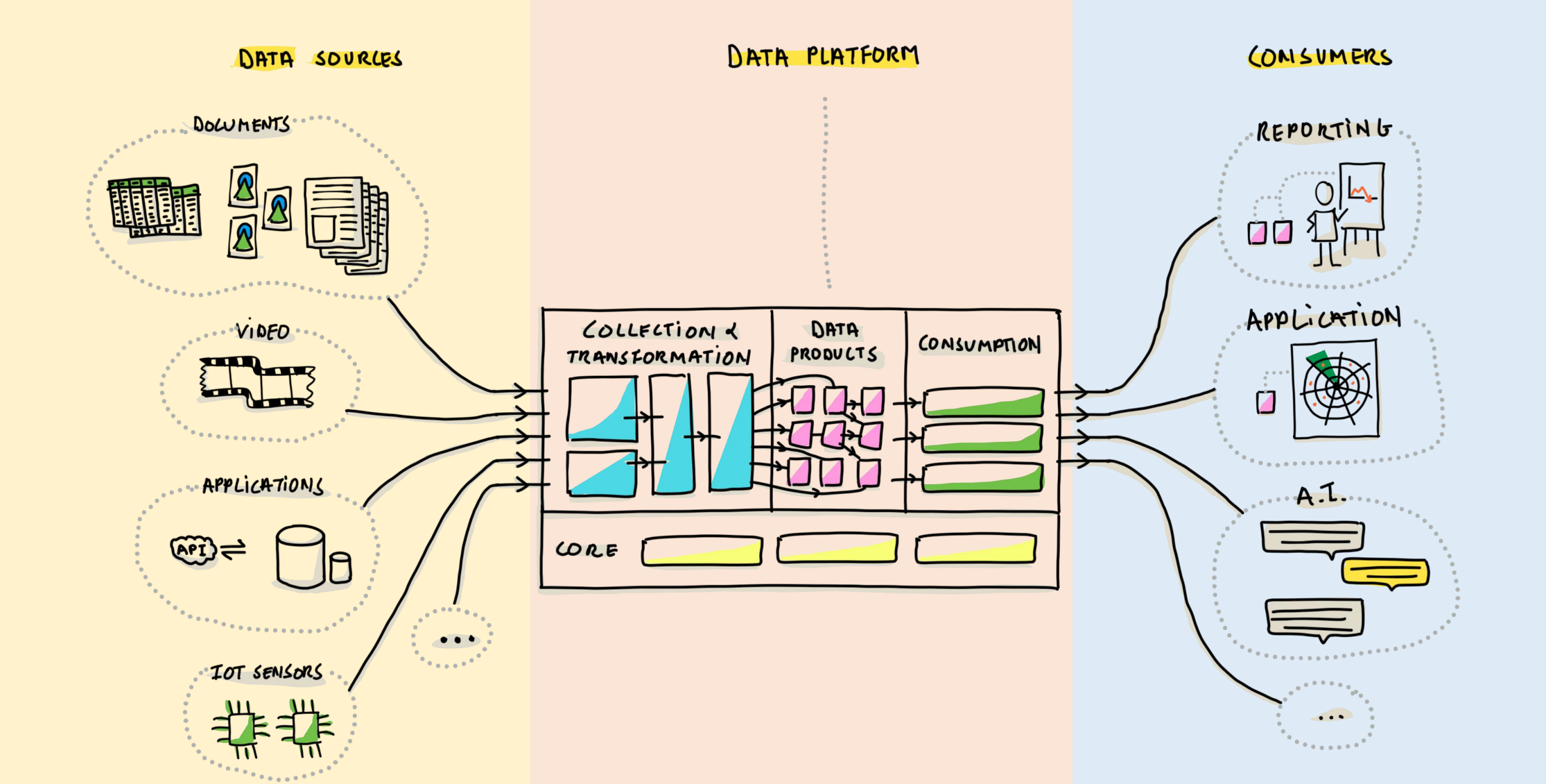
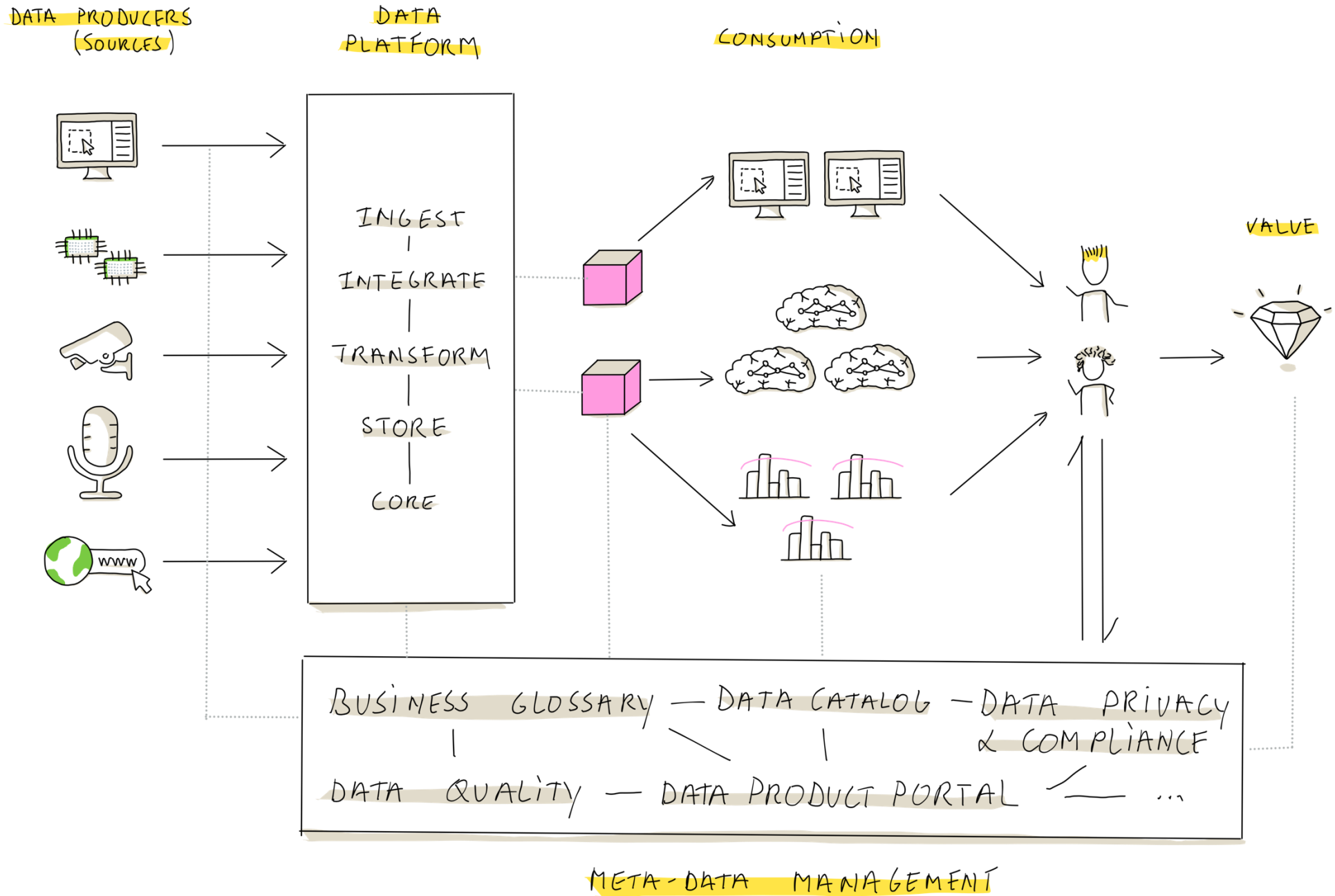
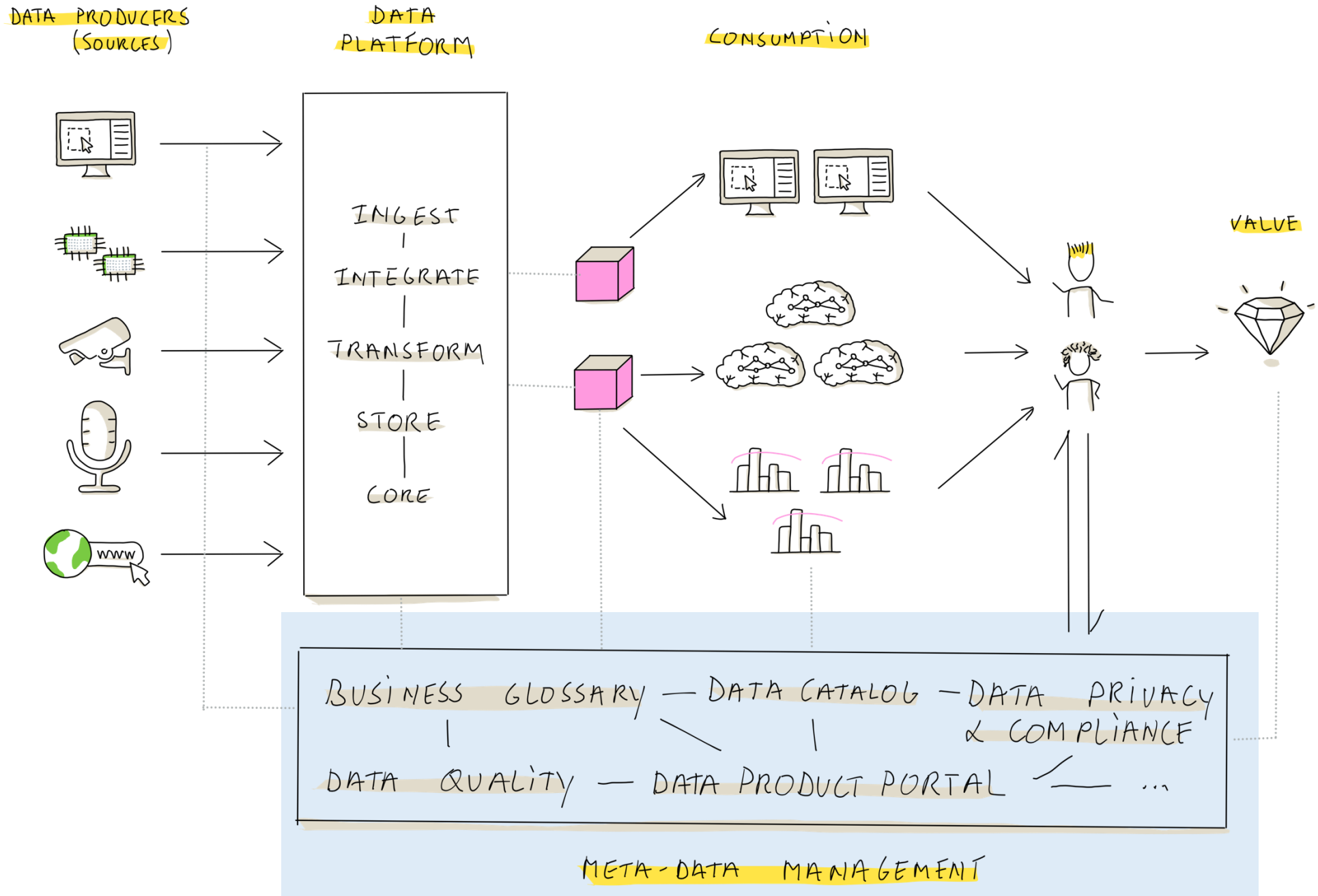
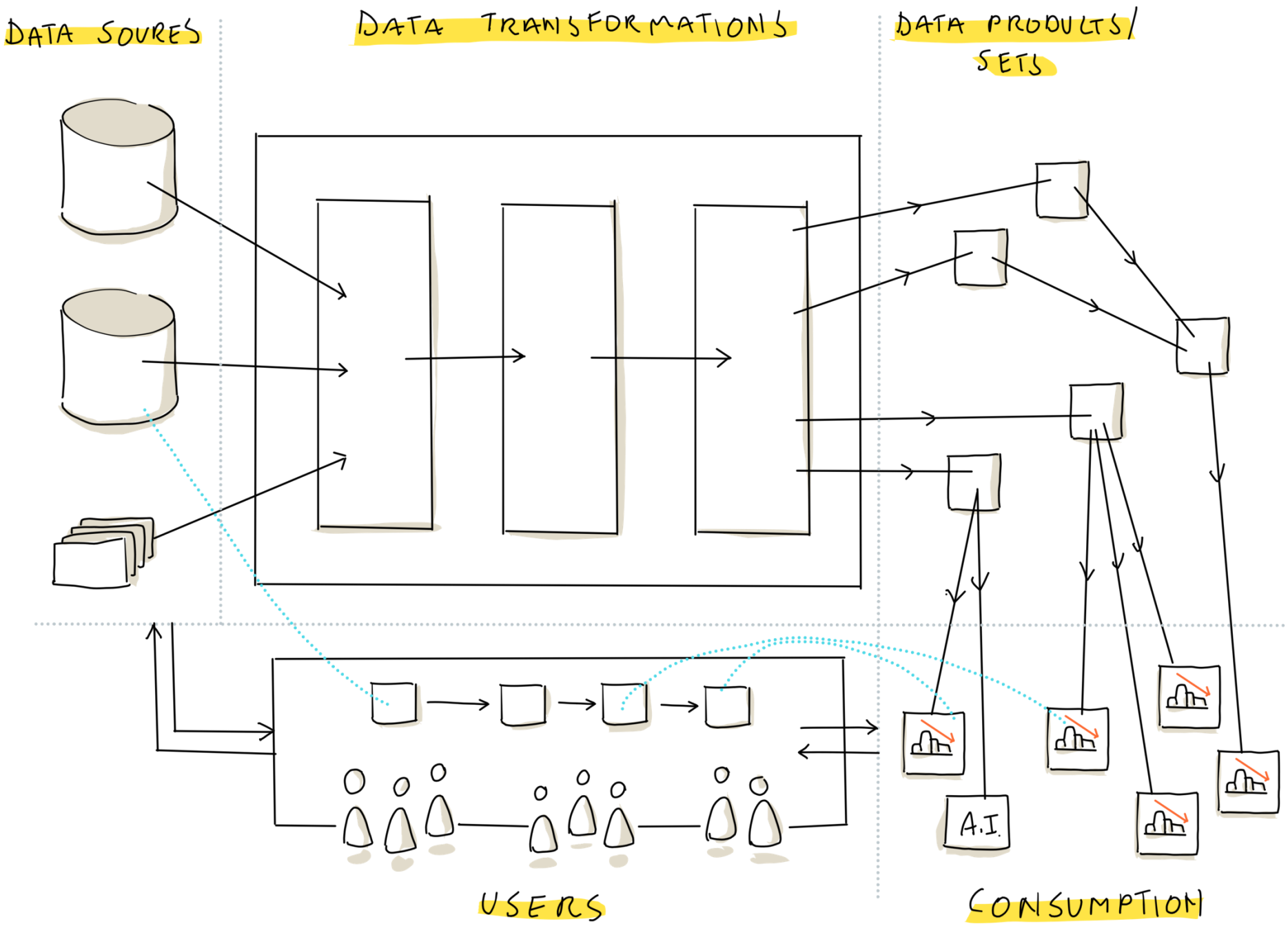

DATA SOURCES
DATA PLATFORM
CONSUMERS
DOCUMENTS
VIDEO
APPLICATIONS
API
IOT SENSORS
COLLECTION & TRANSFORMATION
DATA PRODUCTS
CONSUMPTION
CORE
REPORTING
APPLICATION
A.I.
"Operational Plane"
"Analytical Plane"
"Operational / Analytical Plane"

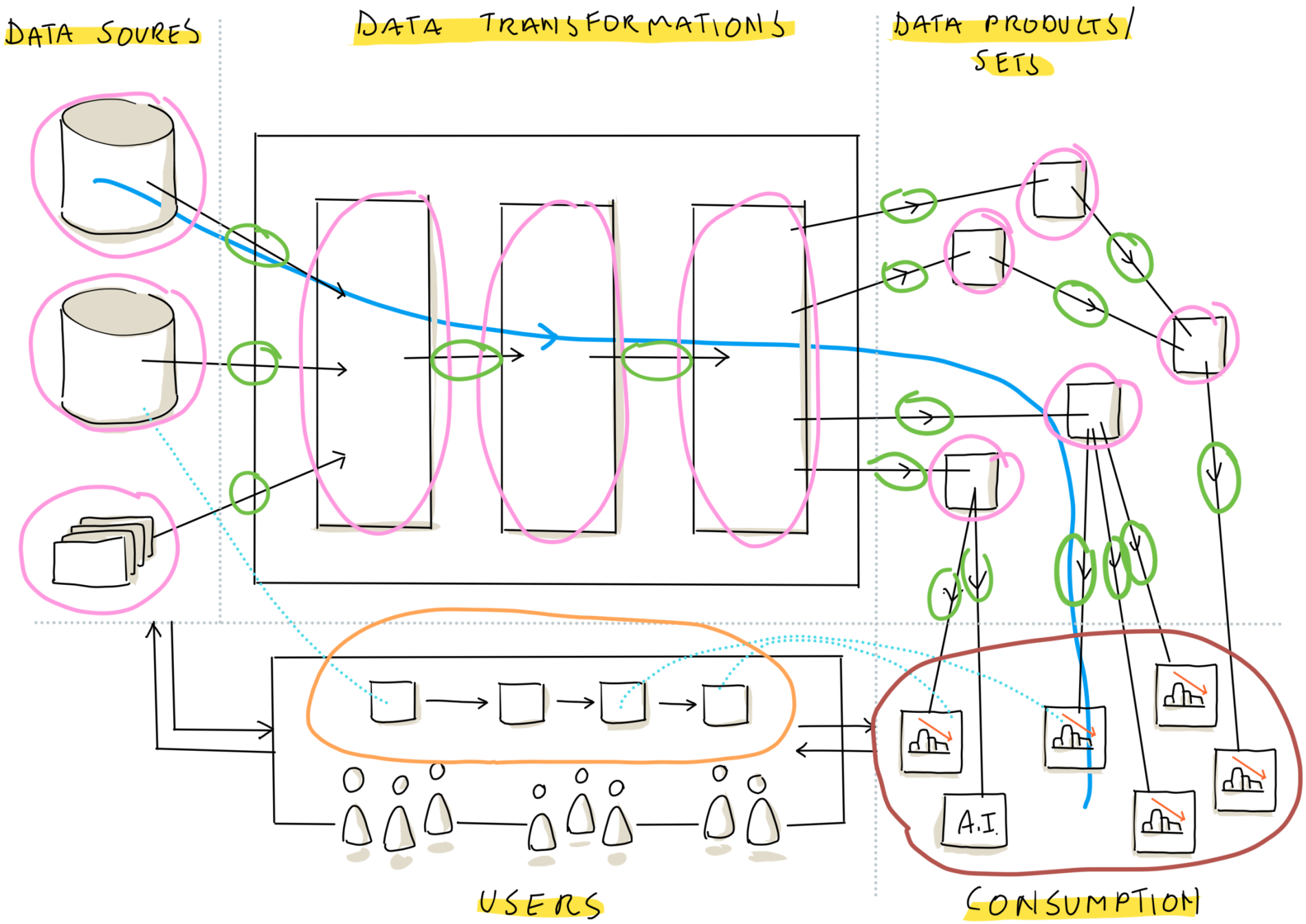
Table of Contents


DATA PRODUCERS (SOURCES)
DATA PLATFORM
CONSUMPTION
VALUE
INGEST
INTEGRATE
TRANSFORM
STORE
CORE
www
BUSINESS GLOSSARY
DATA CATALOG
DATA PRIVACY & COMPLIANCE
DATA QUALITY
DATA PRODUCT PORTAL
...
META-DATA MANAGEMENT


DATA PRODUCERS (SOURCES)
DATA PLATFORM
CONSUMPTION
VALUE
INGEST
INTEGRATE
TRANSFORM
STORE
CORE
www
BUSINESS GLOSSARY
DATA CATALOG
DATA PRIVACY & COMPLIANCE
DATA QUALITY
DATA PRODUCT PORTAL
...
META-DATA MANAGEMENT


DATA SOURES
DATA TRANSFORMATIONS
DATA PRODUCTS/ SETS
A.I.
USERS
CONSUMPTION

META-DATA



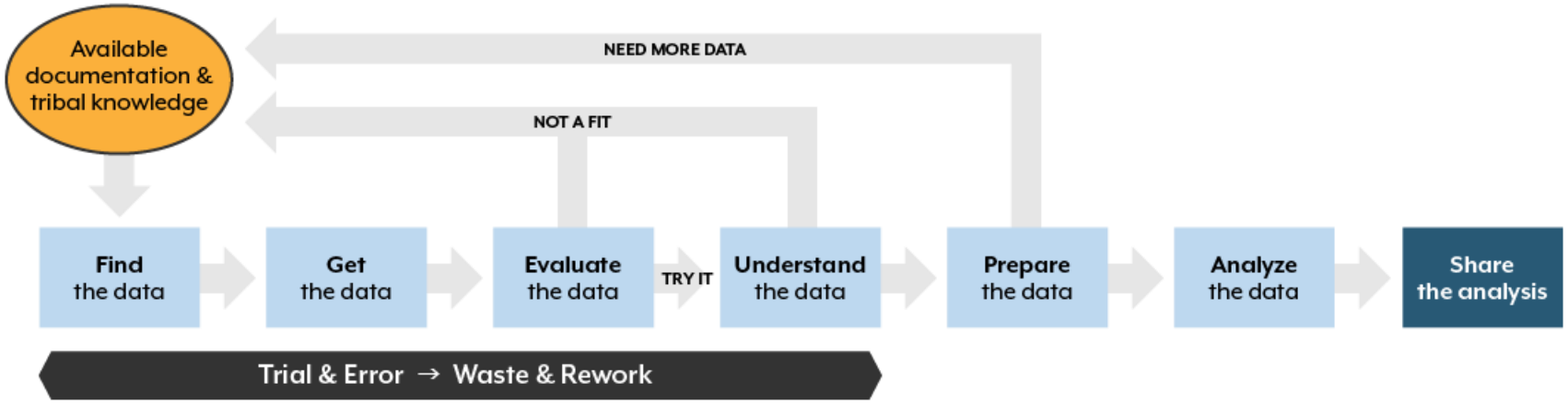
META-DATA

- Data Operations
- Stored Data
- Data Transformations
- Business Knowledge
- Data Consumption
- ...

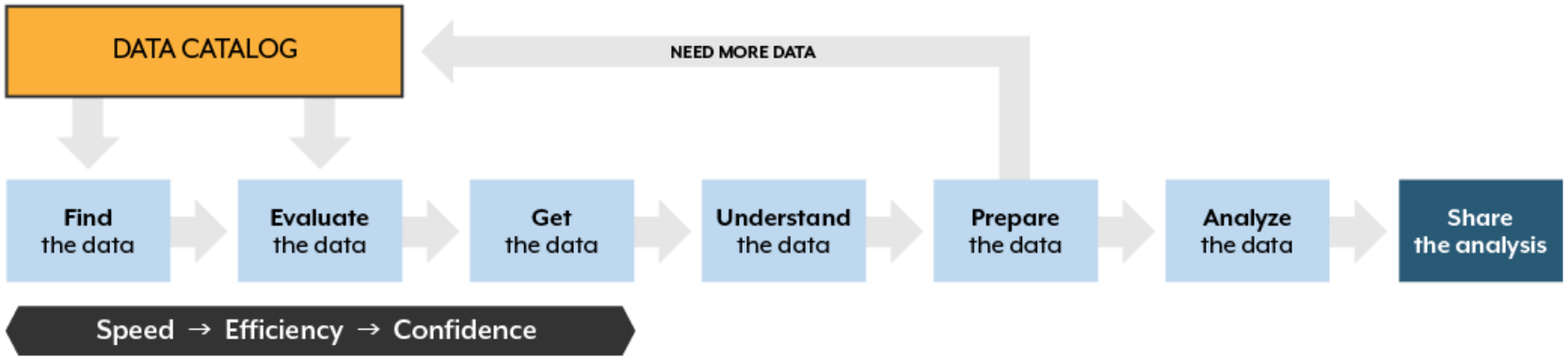

META-DATA
Data Operations
Stored Data
Data Transformations
Business Knowledge
Data Consumption
...
META-DATA TOOL FEATURES
Data Catalog
Data Lineage
Business Glossary
Reporting / Metric Catalog
...

DATA CATALOG

Without Data Catalog

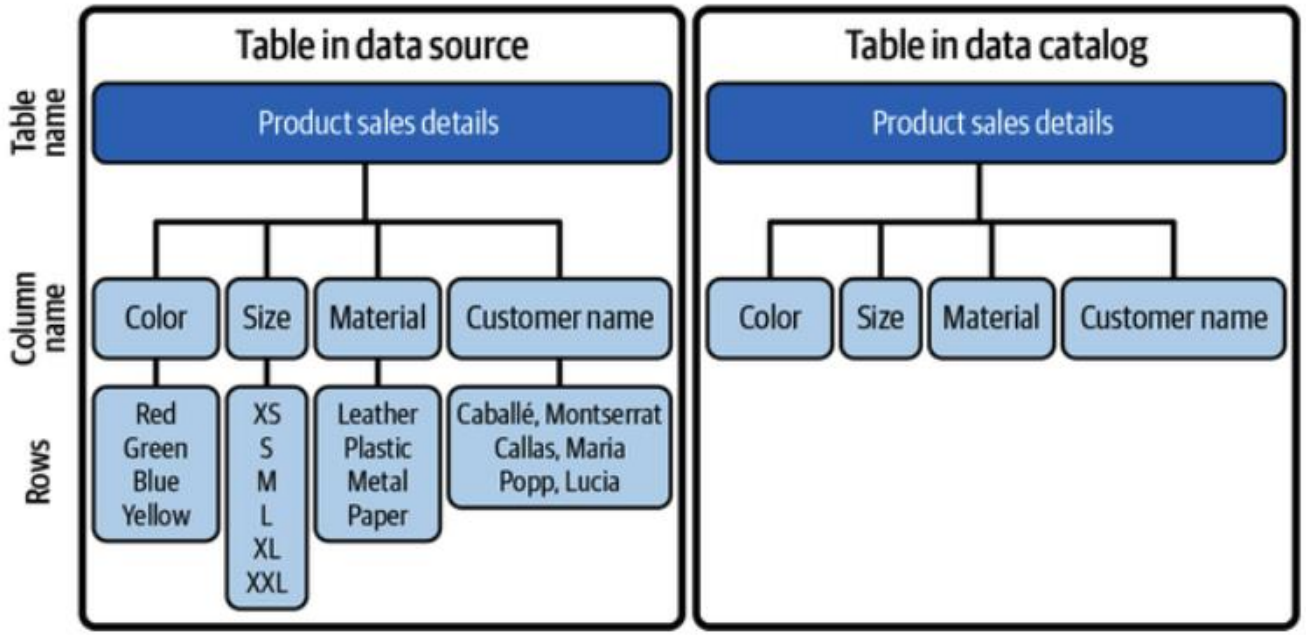


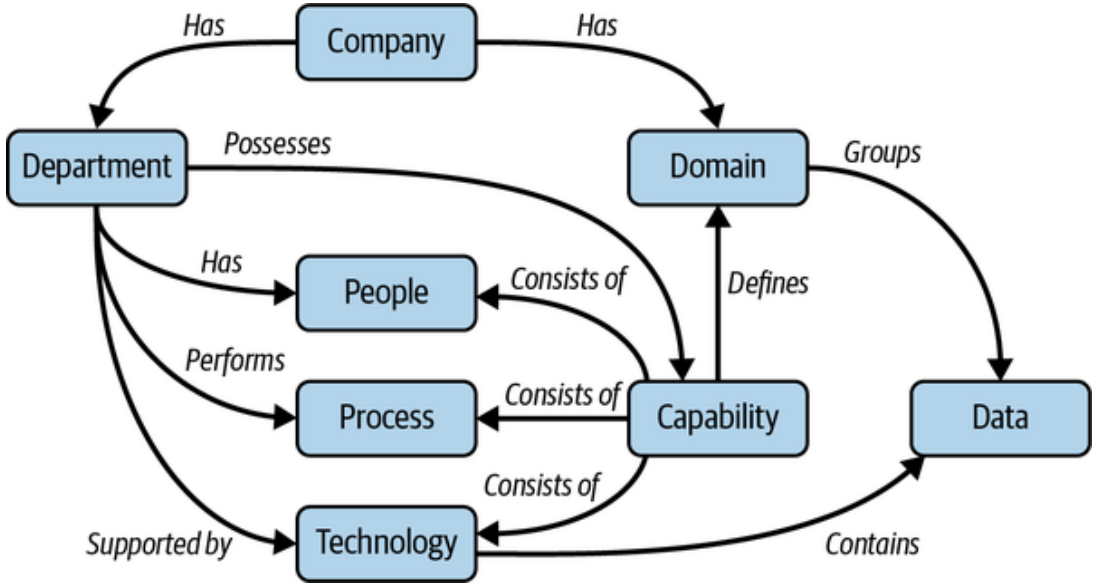
With Data Catalog



[What is a Data Catalog]

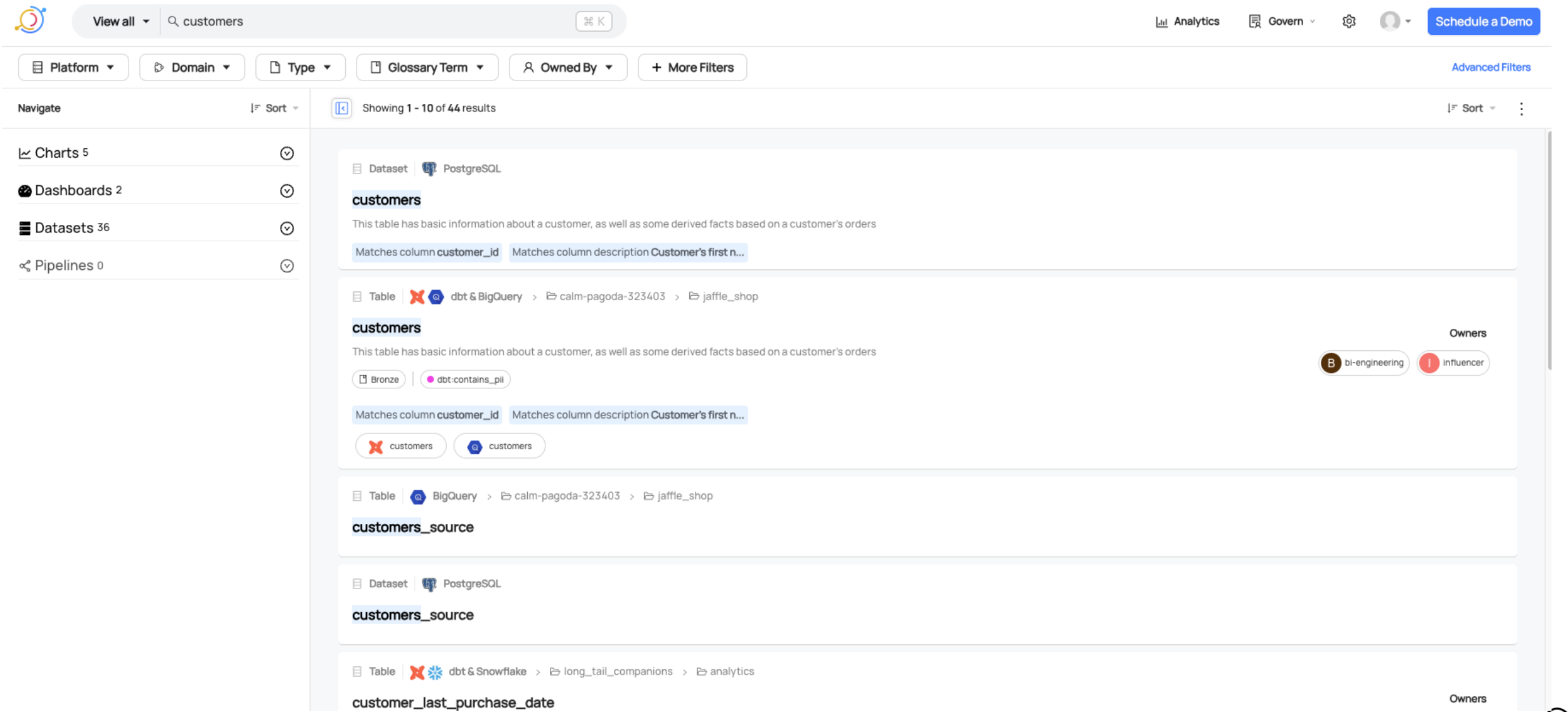
DATA CATALOG





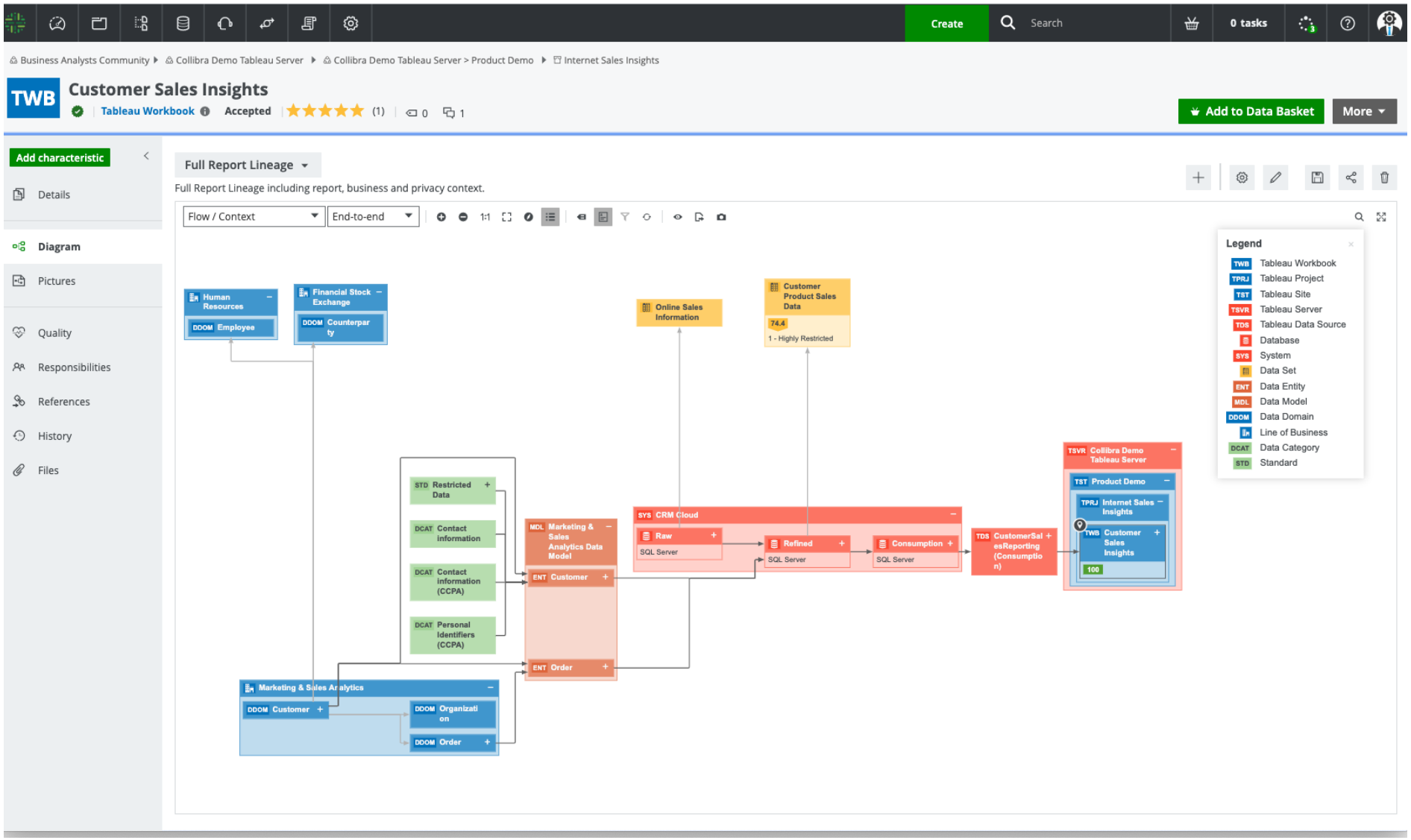
[The Enterprise Data Catalog, Ole Olesen-Bagneux]

DATA CATALOG



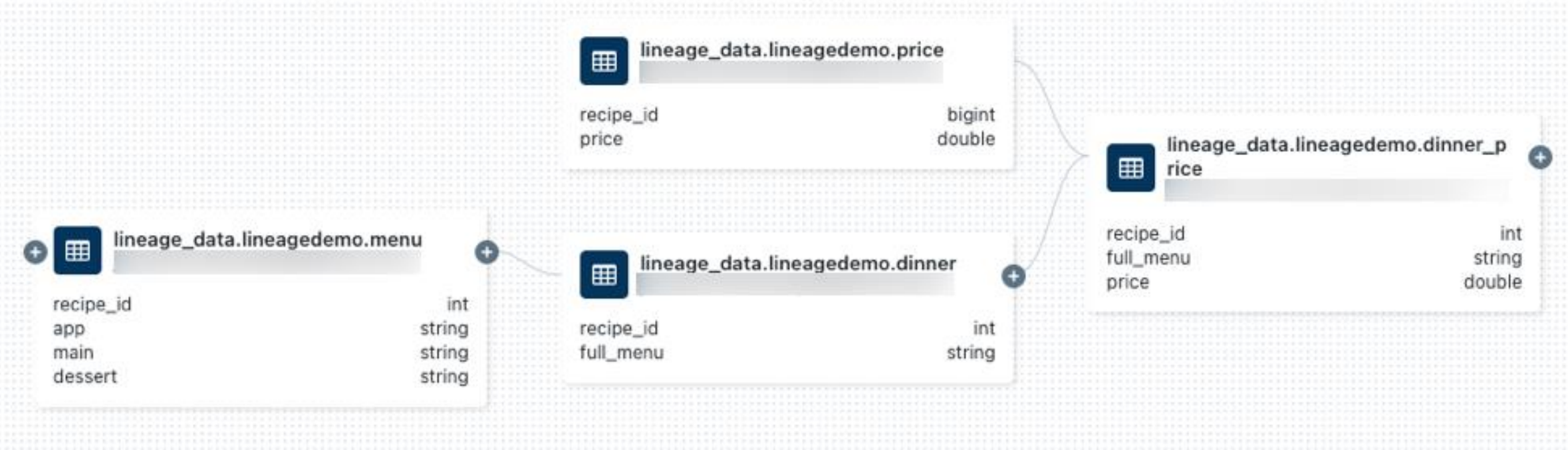
[Datahub (LinkedIn)]

DATA LINEAGE



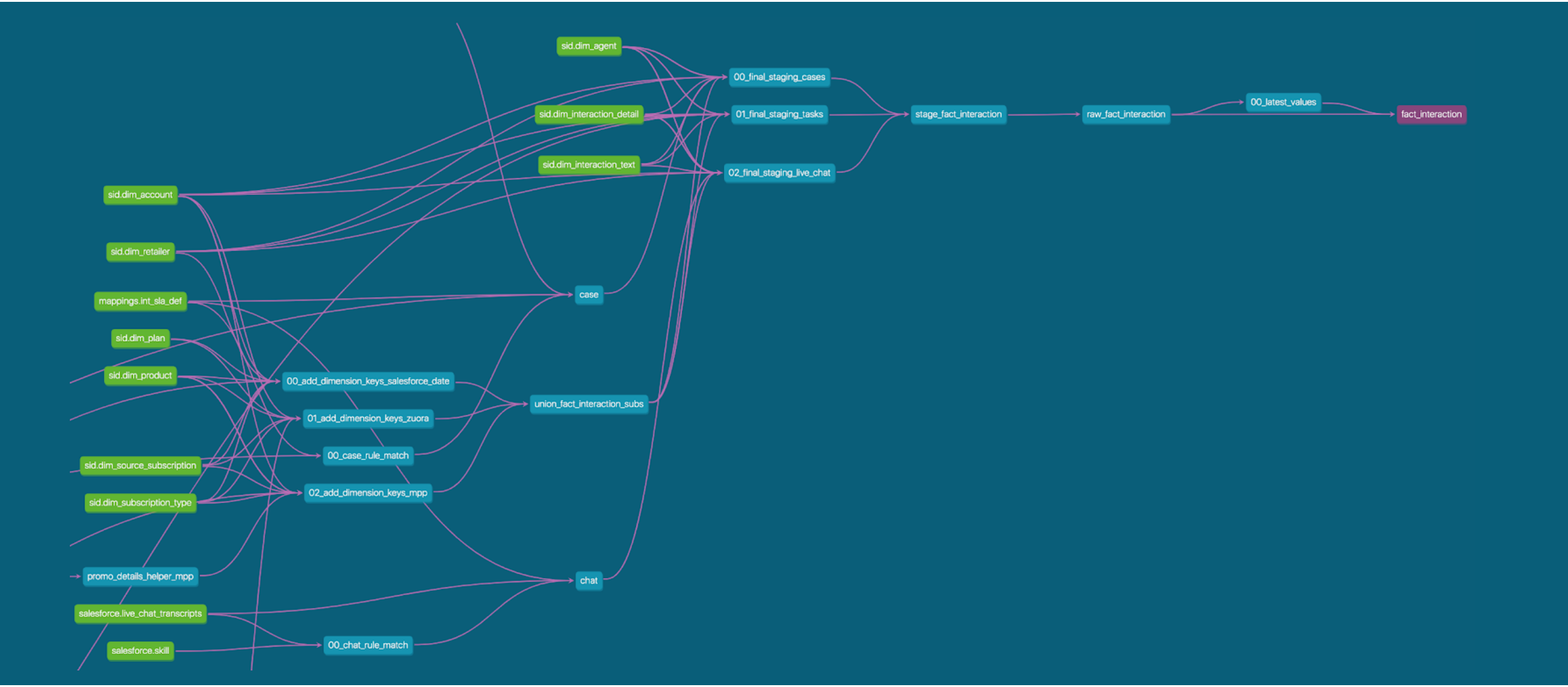
[Data Lineage in Collibra]

DATA LINEAGE



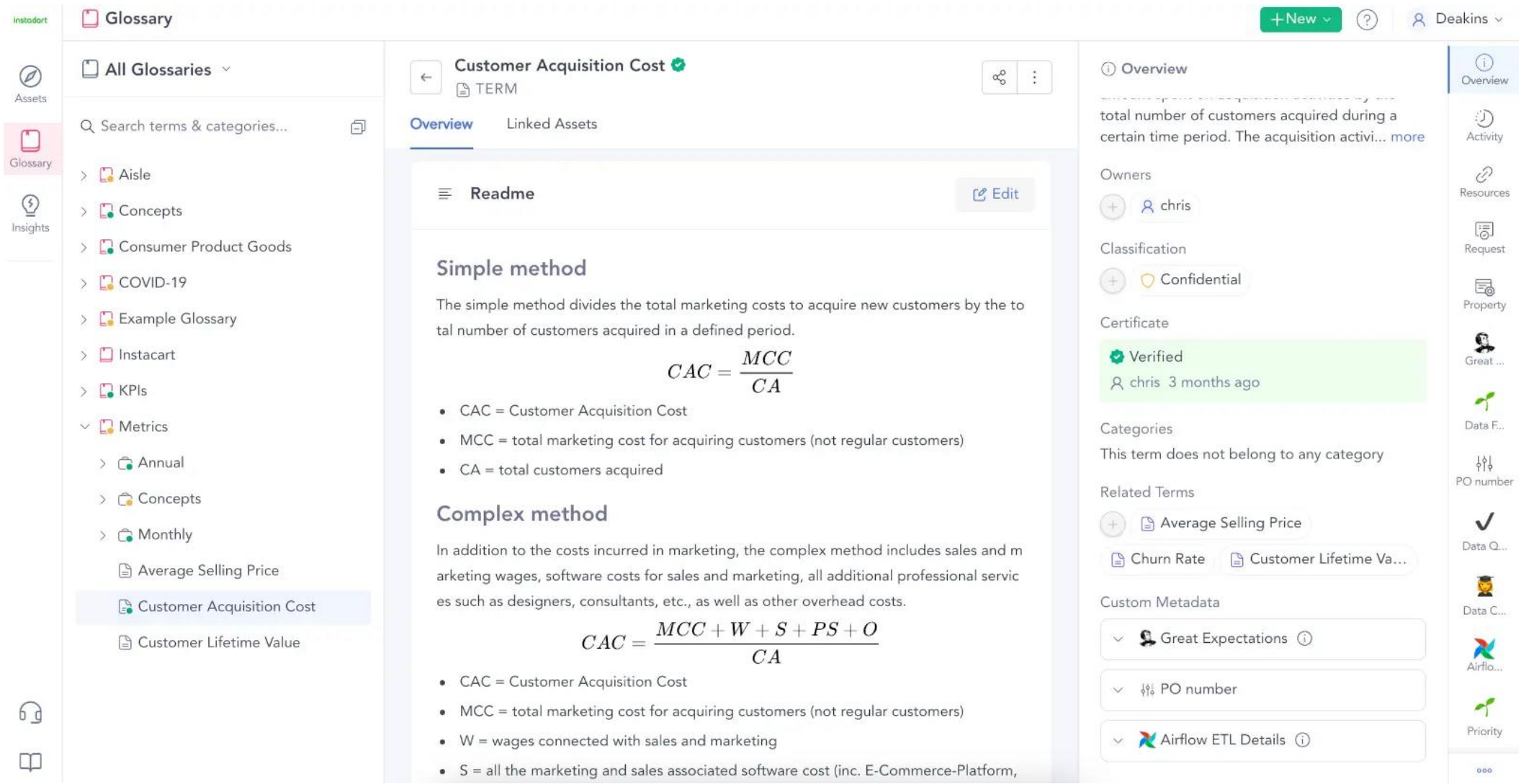
[Data Lineage in Databricks Unity Catalog]

DATA LINEAGE



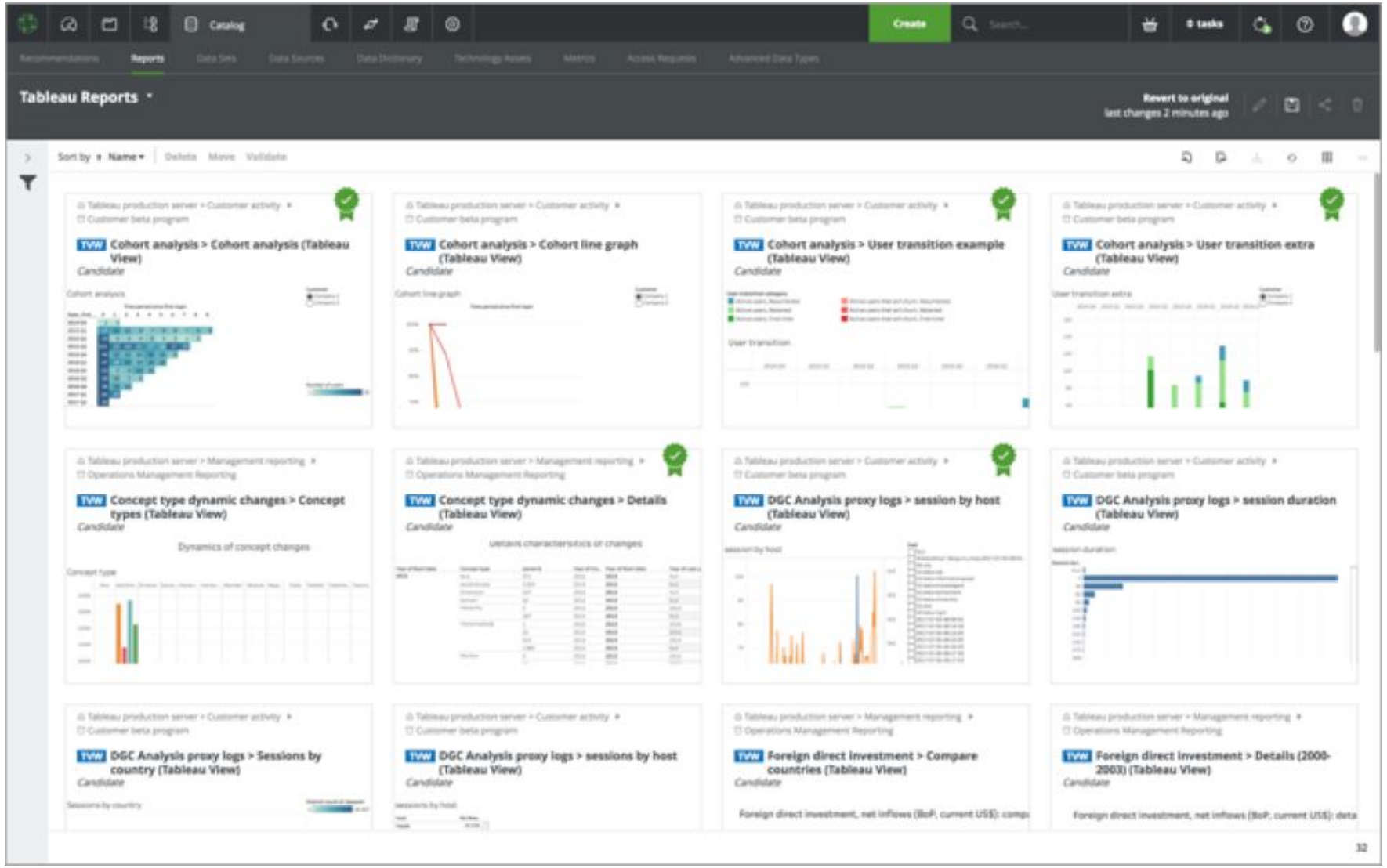
[Data Lineage in DBT]

BUSINESS GLOSSARY



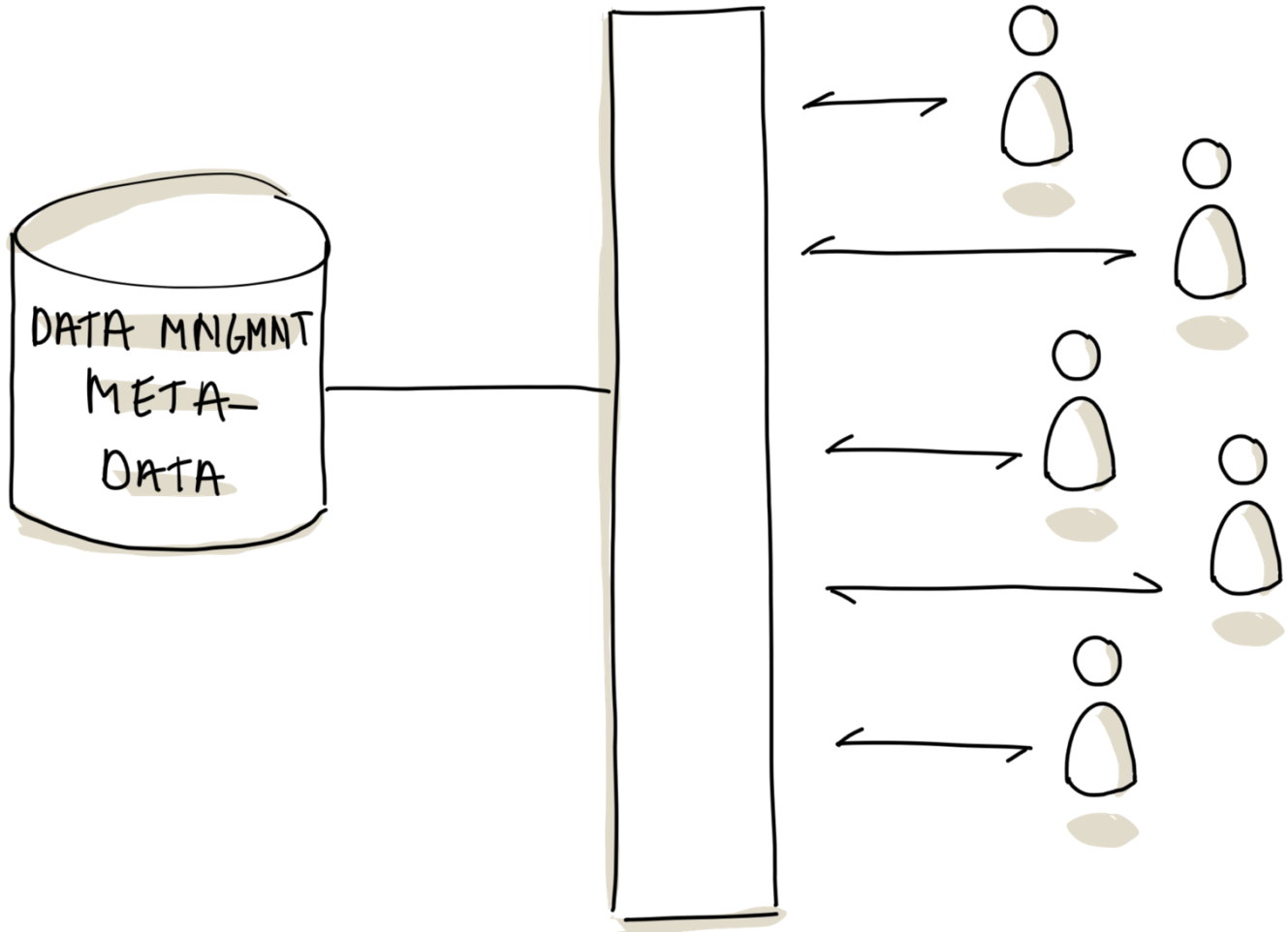
[Business Glossary in Atlan]

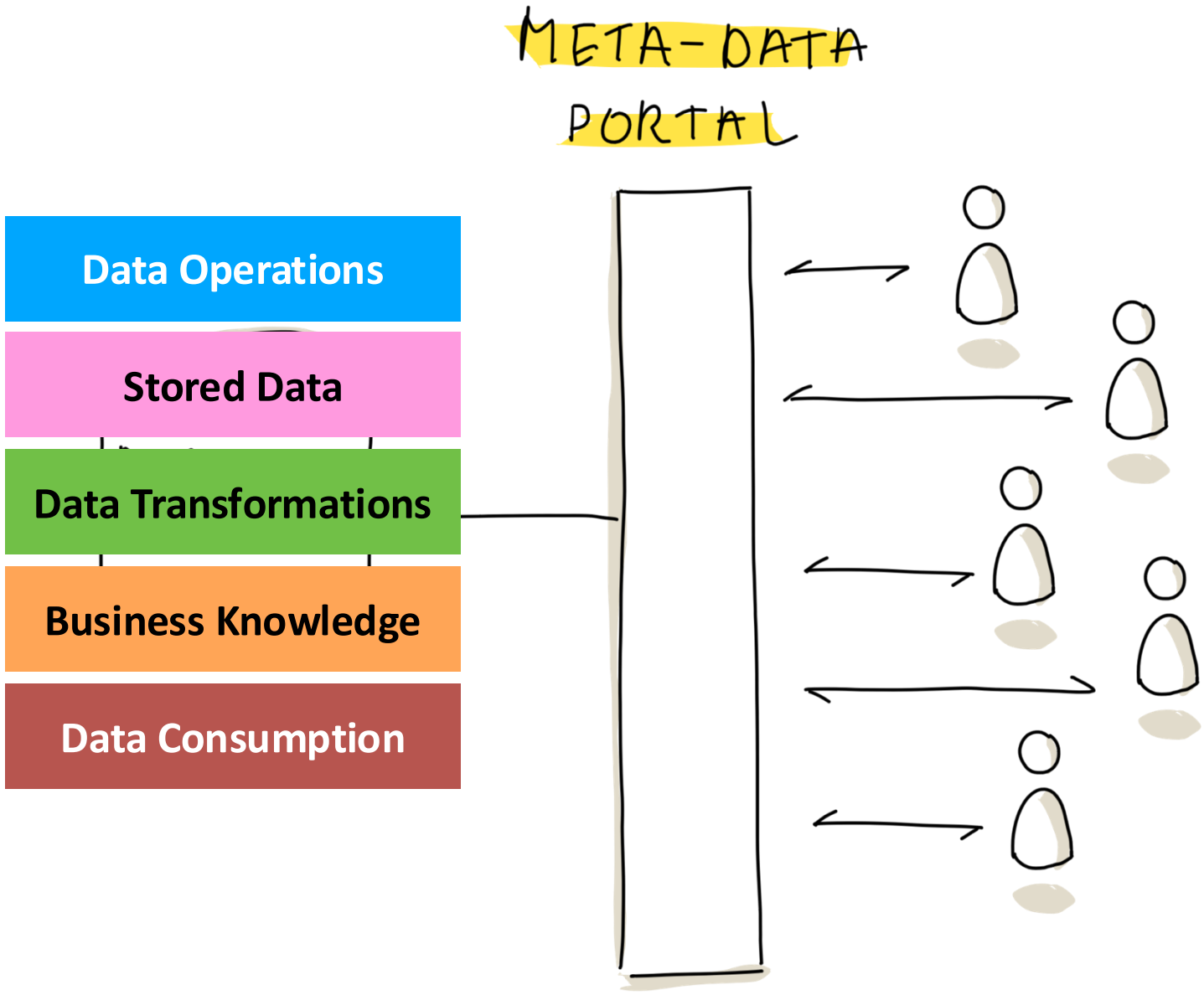
REPORTING / METRIC CATALOG



[Collibra (Screenshot deprecated – now integrated in the 'Data Catalog')]

META-DATA PORTAL




META-DATA
PORTAL
Data Operations
Stored Data
Data Transformations
Business Knowledge
Data Consumption

Data models

Conceptual		Logical		Physical		Canonical	
Entity	Relationship	Entity	Attribute	File	Table	Type	Namespace
Glossary	Identifier	Model	Relationship	Column	Document	Schema	Attribute

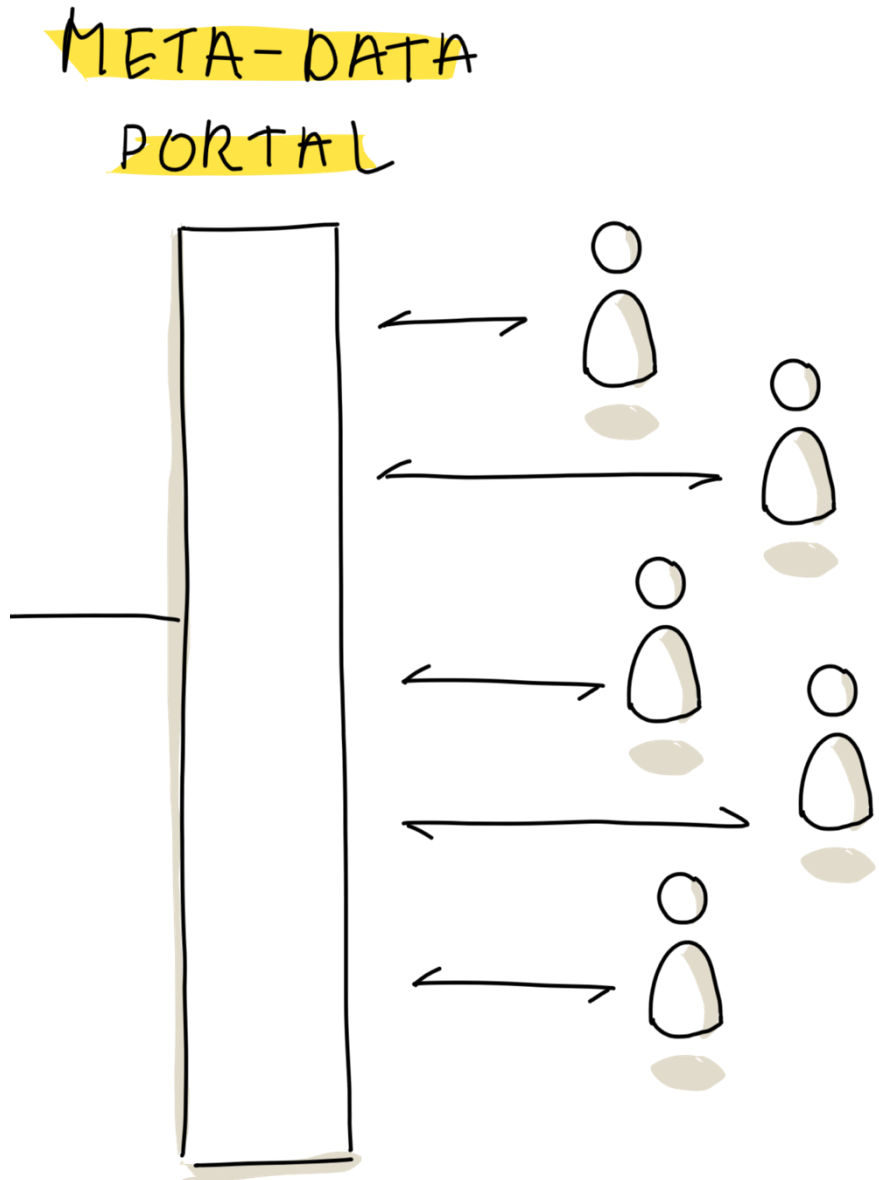
Business capabilities			Semantics		
Process	Organization	Application function	Business rule	Code definition	Attribute

Data objects		Dataset		Collaboration		
Attributes		Elements		Comment	Tag	Like
Topic	Database	Data domain	Data owner	Rating	Label	Note
API	File	Relationships	Definitions			

Data quality		Lineage			Technology	
Metric	Dimension	Process	Dataset	Attribute	Platform	Datastore
Score	Attribute	Source:target			Application	Data flow

Organization								
Process	Roles	Agreement	Owner	Purpose	Sensitivity	Confidentiality	Organization	SLA

Figure 10-2. An overview showing all the major metadata management subject areas for the enterprise. These subject areas are nonexhaustive. Each organization uses the areas differently.



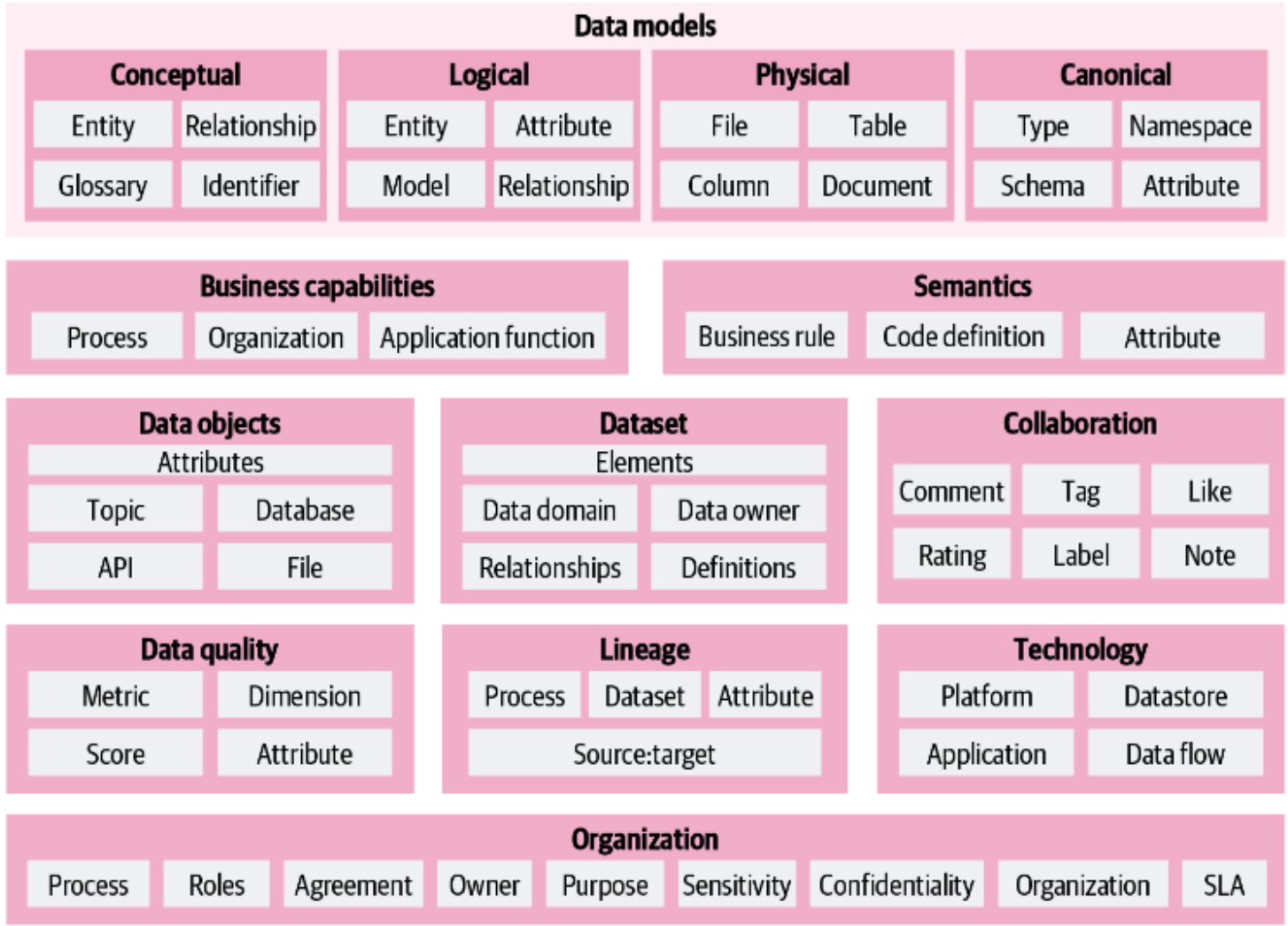
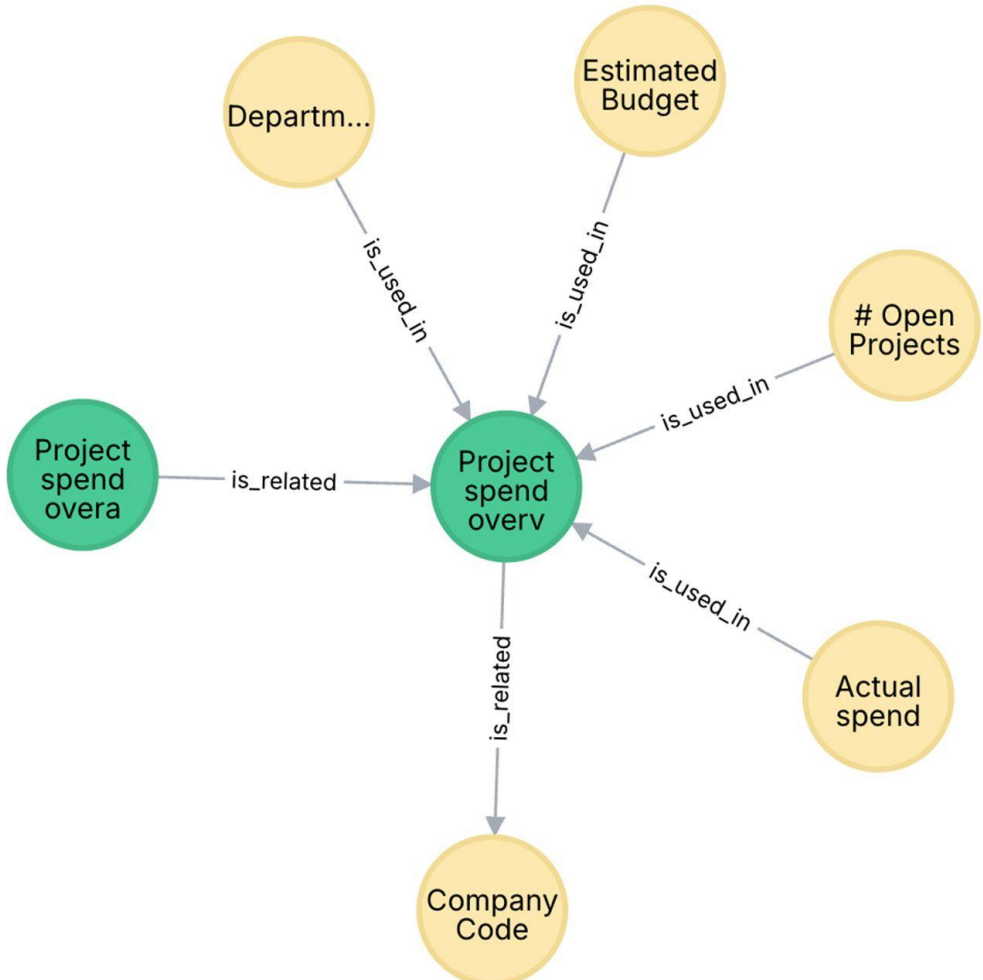


Figure 10-2. An overview showing all the major metadata management subject areas for the enterprise. These subject areas are nonexhaustive. Each organization uses the areas differently.



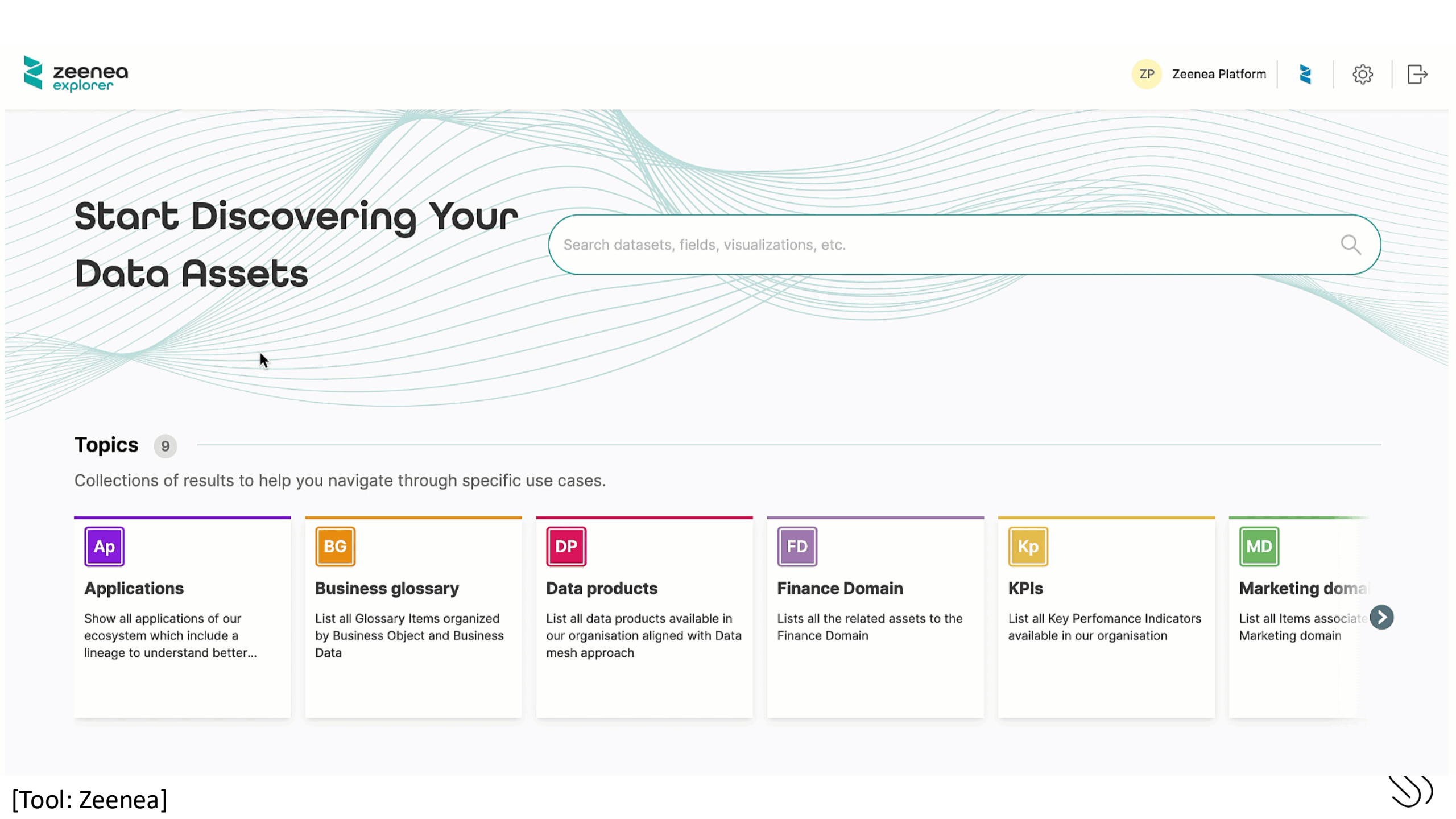
Asset Types

- [x] Report
- [x] Definition
- [x] Webpage

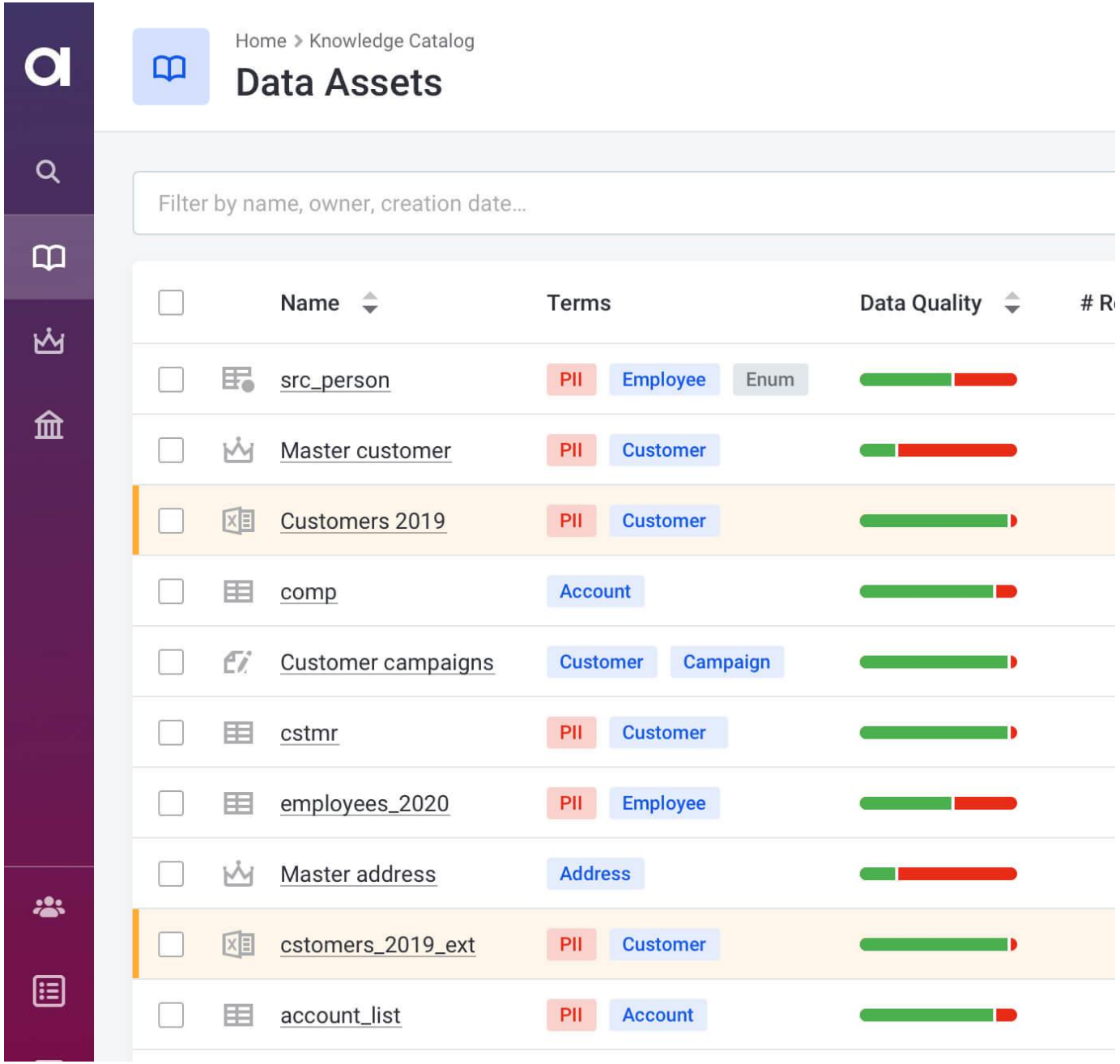
Relation Type

- [x] is used in / uses
- [x] is related to
- [x] is parent of / is child of
- [x] mentions / is mentioned in

[Tool: Dscribe]

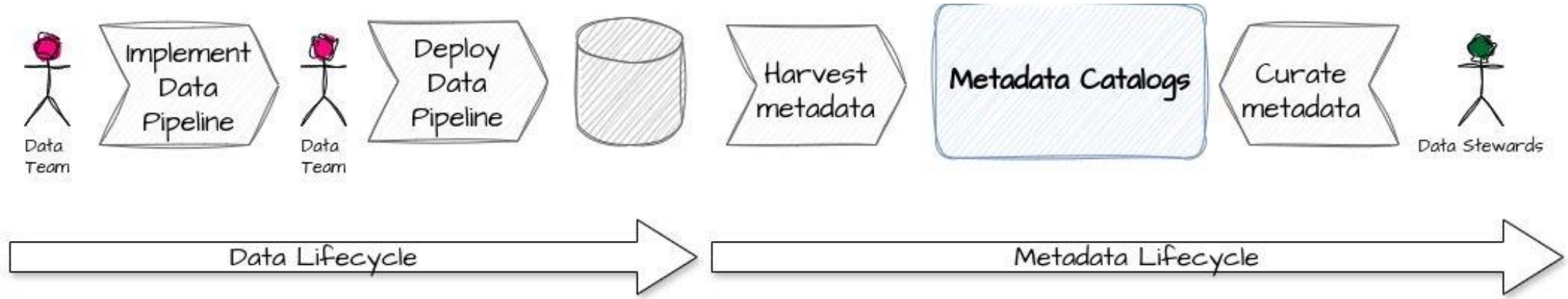

zeenea explorer
ZP
Zeenea Platform
Start Discovering Your Data Assets
Search datasets, fields, visualizations, etc.
Topics 9
Collections of results to help you navigate through specific use cases.
Ap
Applications
Show all applications of our ecosystem which include a lineage to understand better...
BG
Business glossary
List all Glossary Items organized by Business Object and Business Data
DP
Data products
List all data products available in our organisation aligned with Data mesh approach
FD
Finance Domain
Lists all the related assets to the Finance Domain
Kp
KPIs
List all Key Perfomance Indicators available in our organisation
MD
List all Items associate
Marketing domain

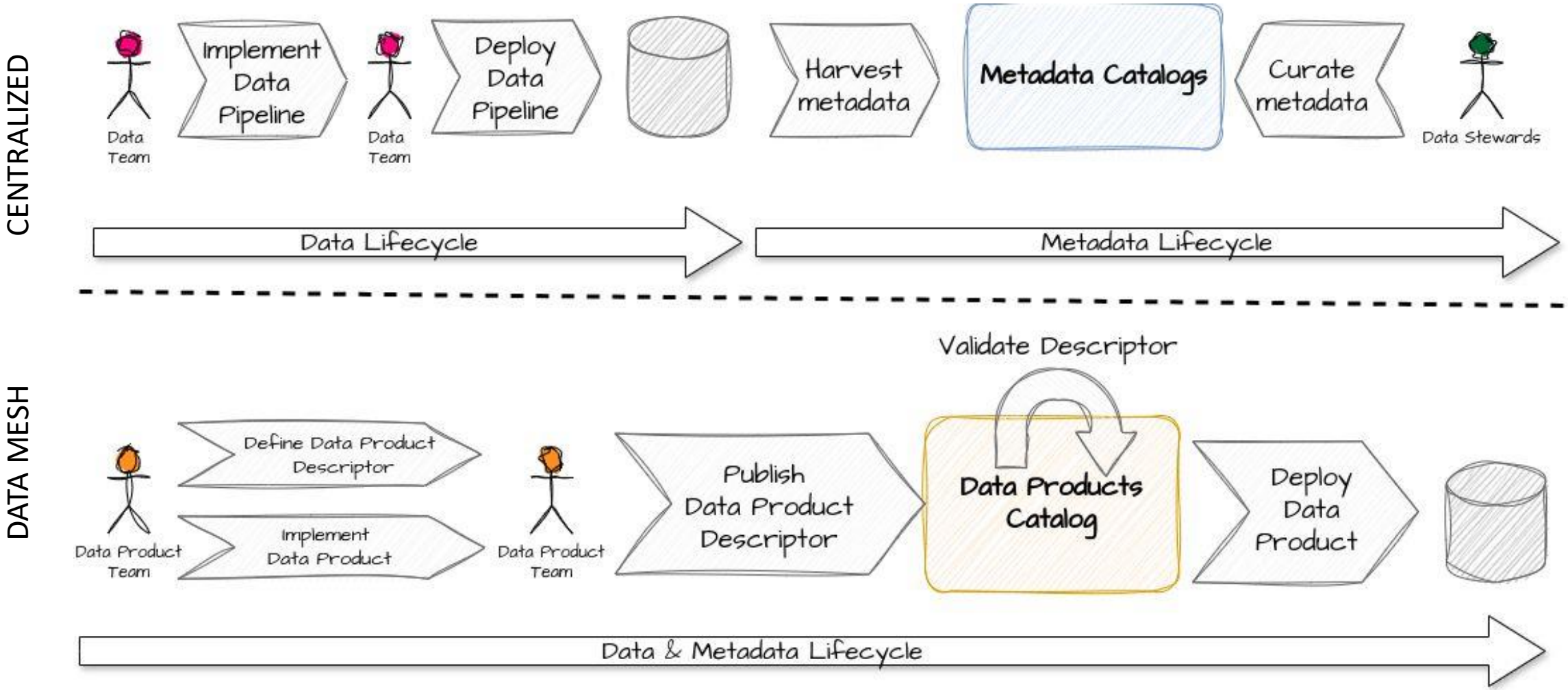
[Tool: Zeenea]


Home > Knowledge Catalog
Data Assets
Filter by name, owner, creation date...
Name
Terms
Data Quality
R
src_person
PII
Employee
Enum
Master customer
PII
Customer
Customers 2019
PII
Customer
comp
Account
Customer campaigns
Customer
Campaign
cstmr
PII
Customer
employees_2020
PII
Employee
Master address
Address
cstomers_2019_ext
PII
Customer
account_list
PII
Account

[Alation]

CENTRALIZED





[Copyright: Andrea Gioia]

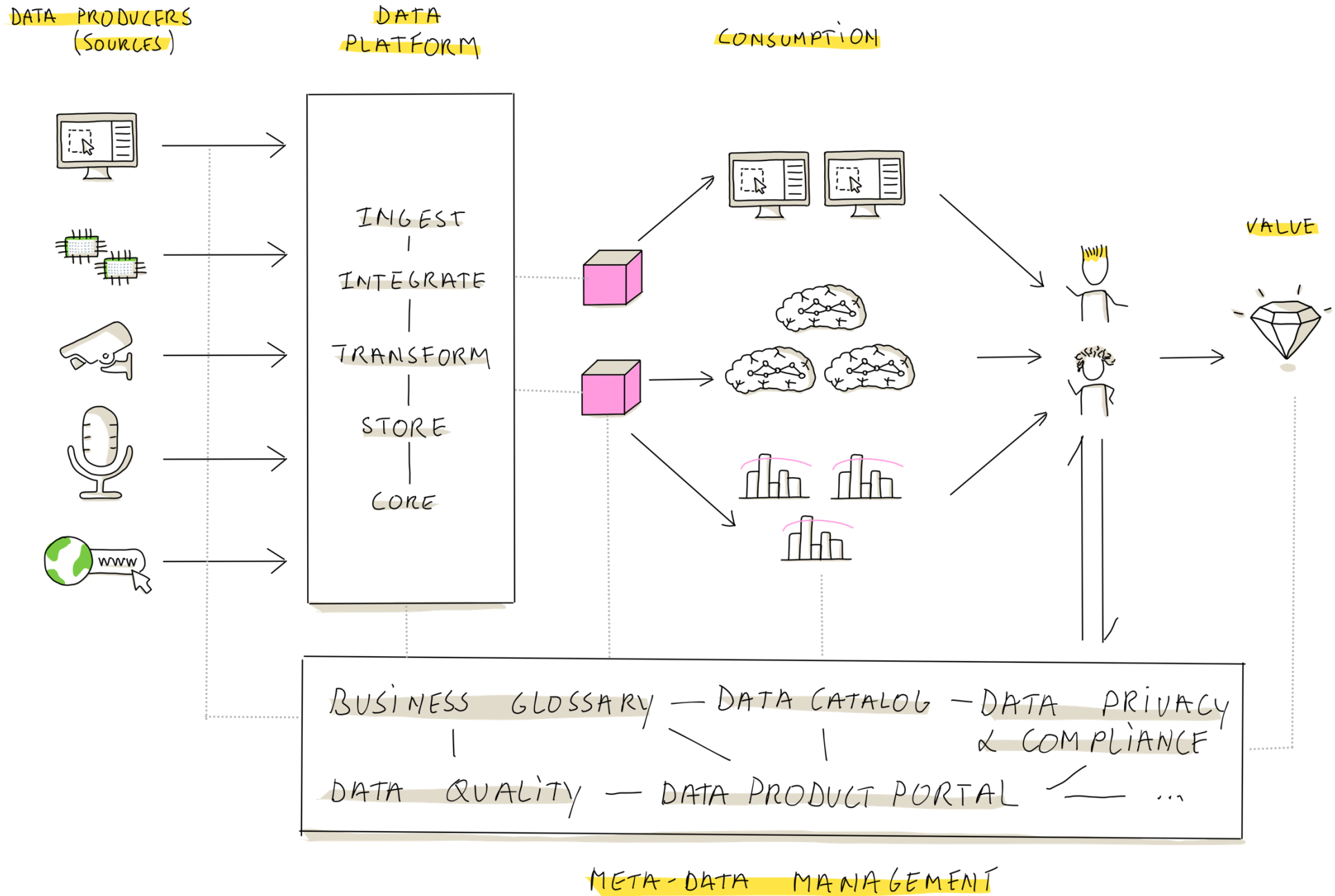
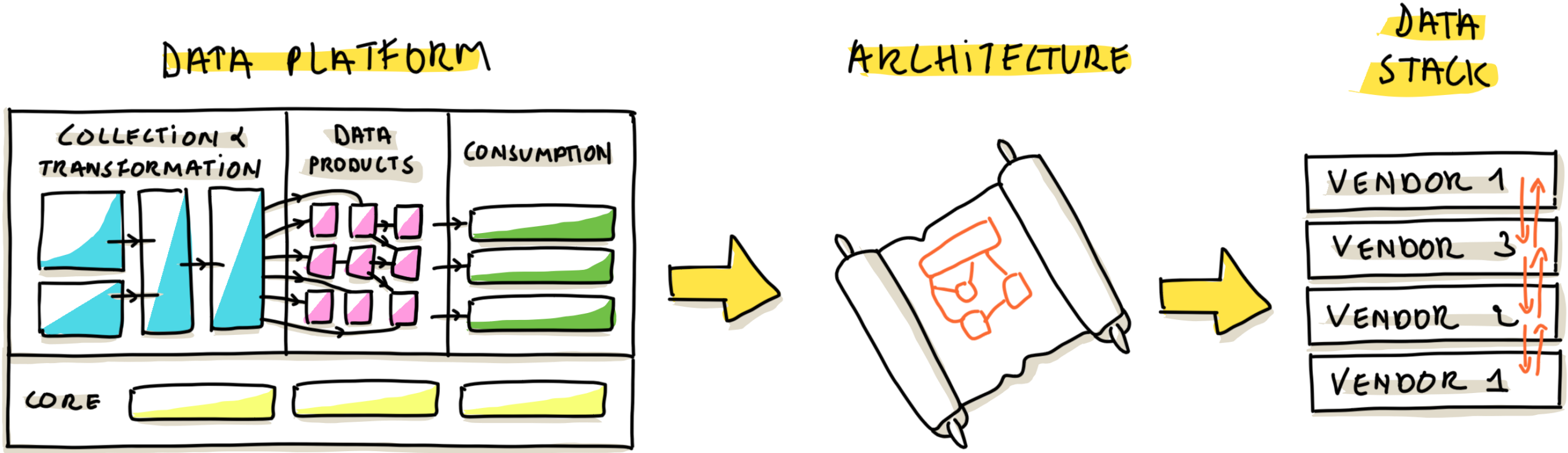
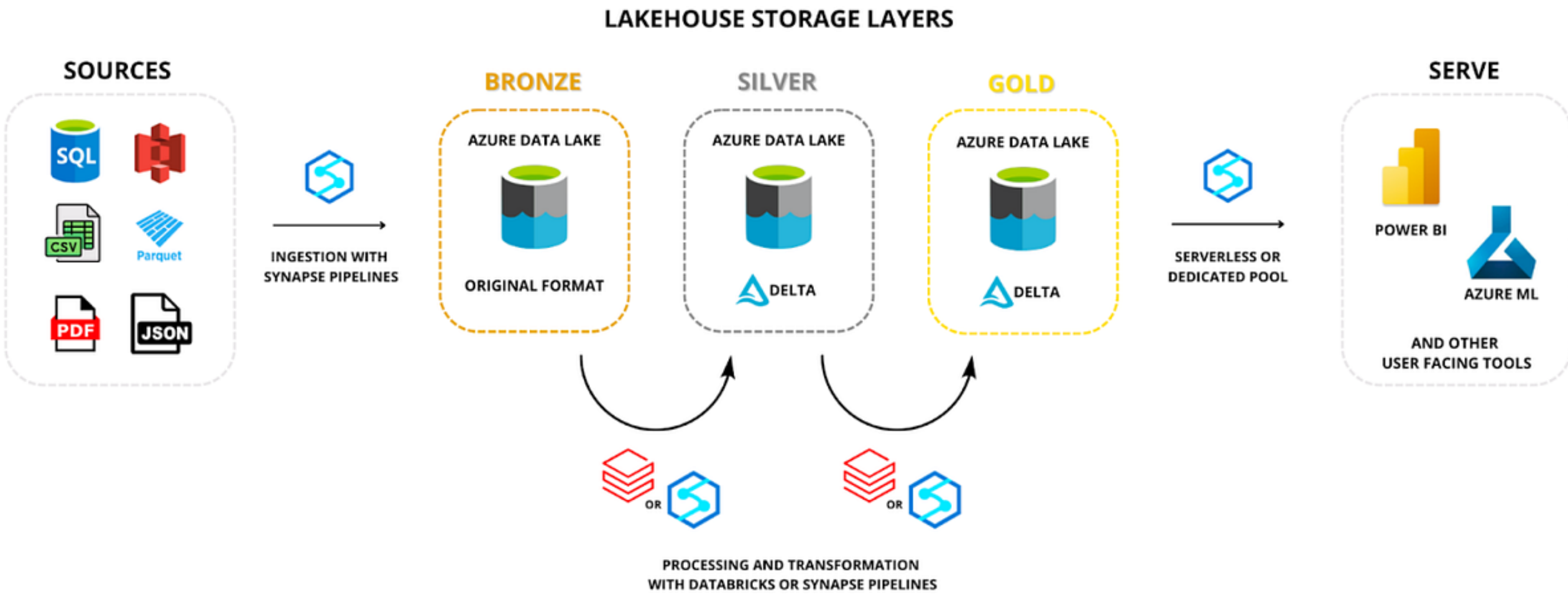

DATA PRODUCERS (SOURCES)
DATA PLATFORM
CONSUMPTION
VALUE
INGEST
INTEGRATE
TRANSFORM
STORE
CORE
www
BUSINESS GLOSSARY
DATA CATALOG
DATA PRIVACY & COMPLIANCE
DATA QUALITY
DATA PRODUCT PORTAL
...
META-DATA MANAGEMENT

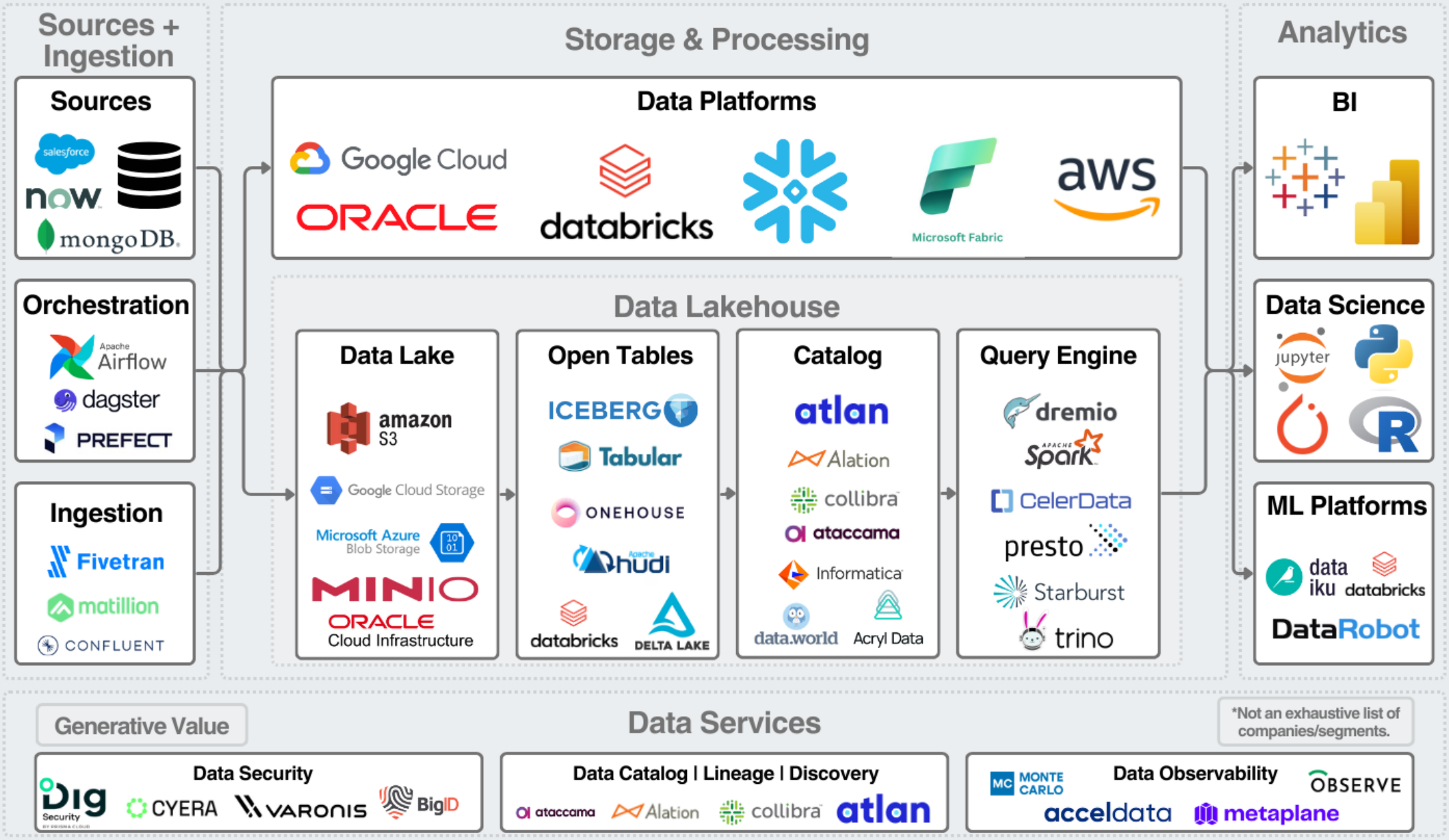
Table of Contents

From Platform to Implementation

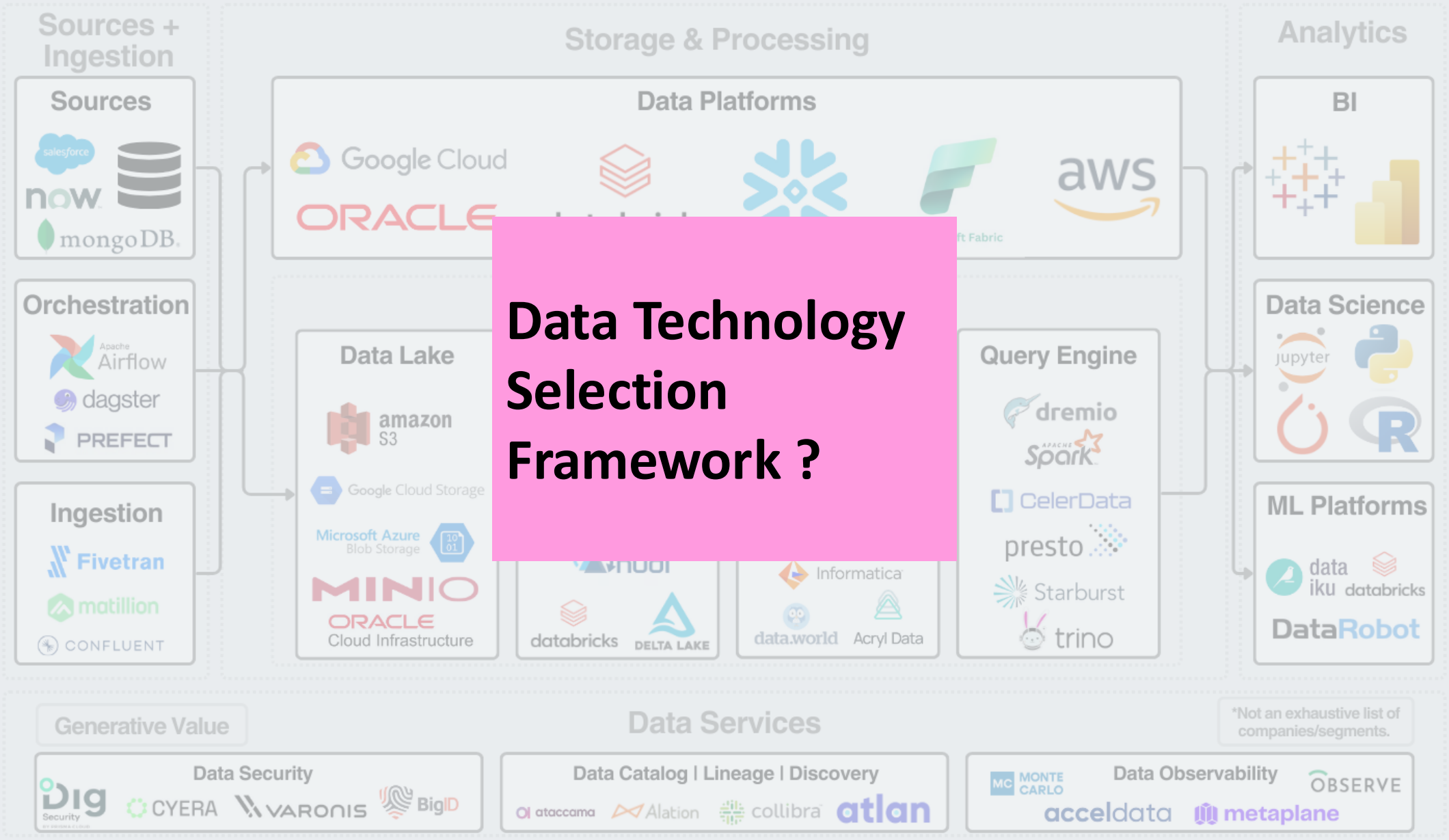


Medaillon Reference Architecture


LAKEHOUSE STORAGE LAYERS
SOURCES
SQL
CSV
Parquet
PDF
JSON
INGESTION WITH
SYNAPSE PIPELINES
BRONZE
AZURE DATA LAKE
ORIGINAL FORMAT
SILVER
AZURE DATA LAKE
DELTA
GOLD
AZURE DATA LAKE
DELTA
OR
OR
PROCESSING AND TRANSFORMATION
WITH DATABRICKS OR SYNAPSE PIPELINES
SERVERLESS OR
DEDICATED POOL
SERVE
POWER BI
AZURE ML
AND OTHER
USER FACING TOOLS

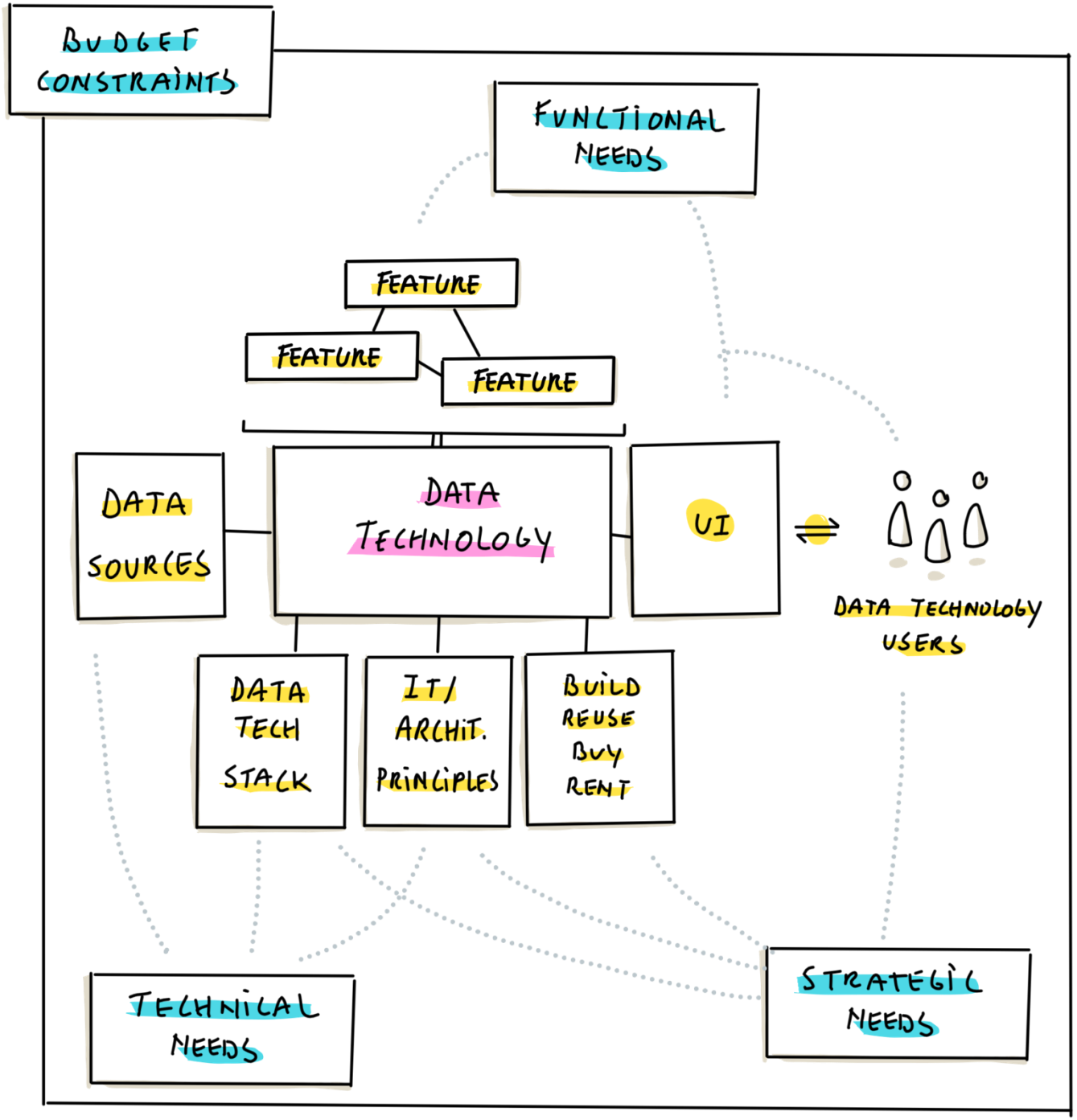


[A Primer on Data Lakehouses]



[A Primer on Data Lakehouses]

Data Technology Selection Framework



[Framework Under Review]

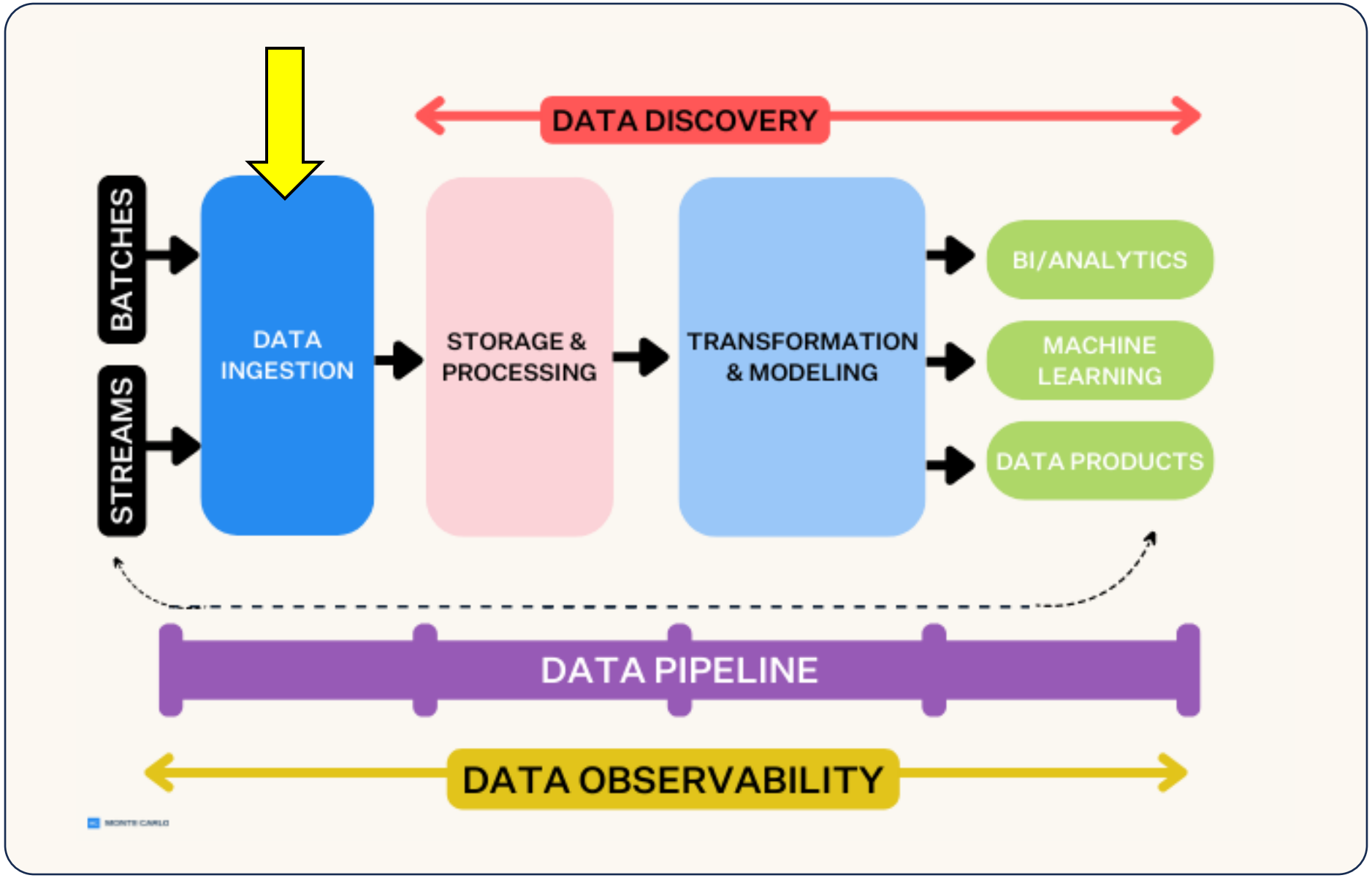
Data Technology Selection Framework

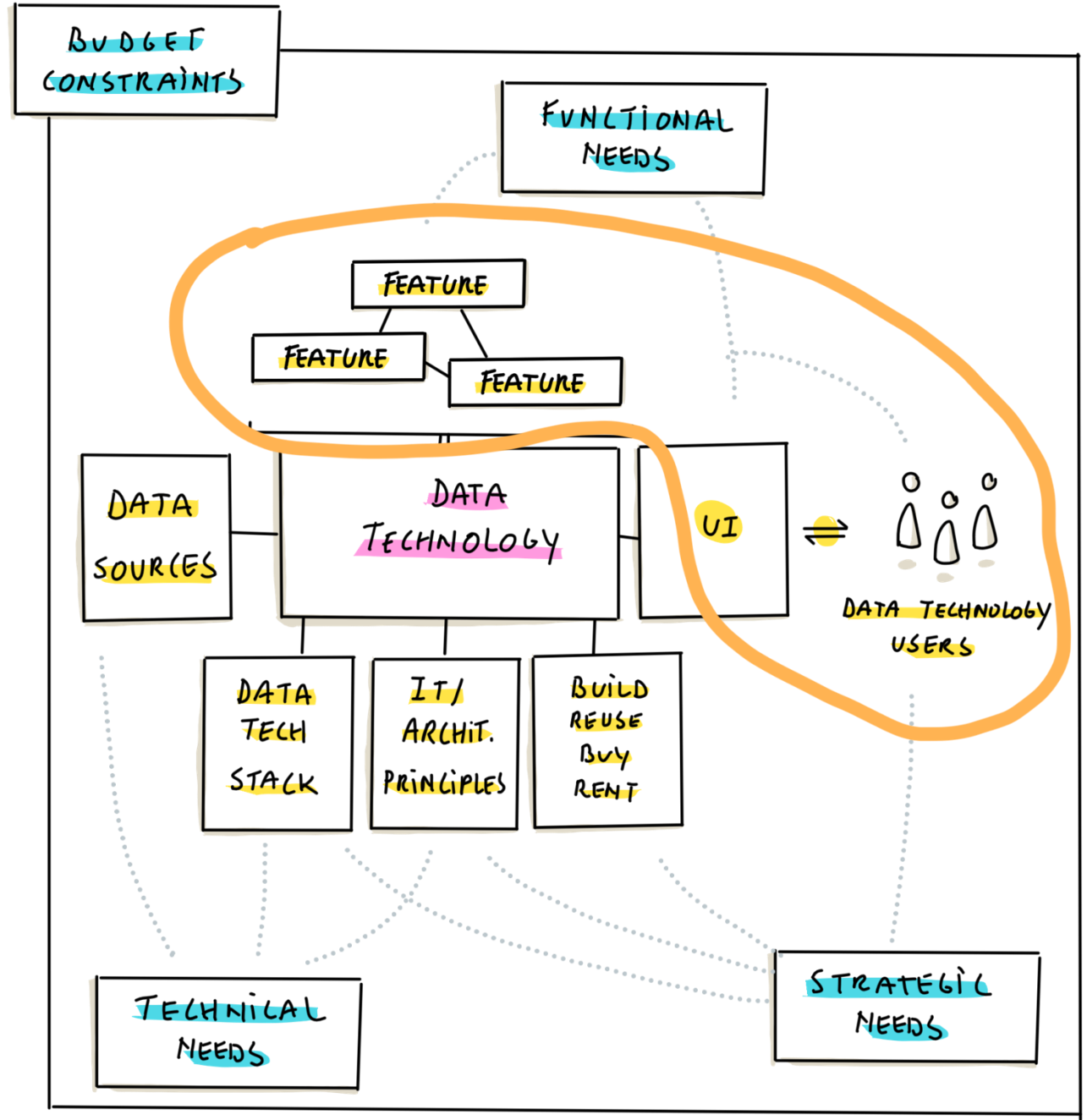
Complete Data Platform

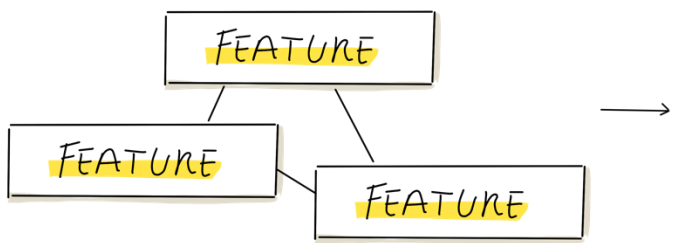
1 or More Data Platform Component(s)

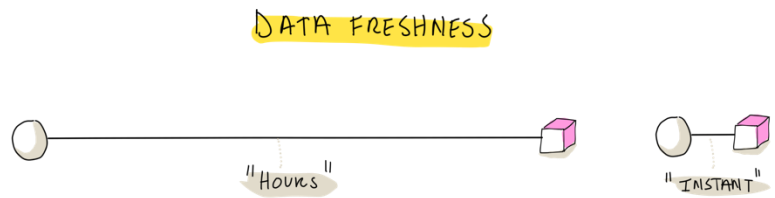
[Framework Under Review]

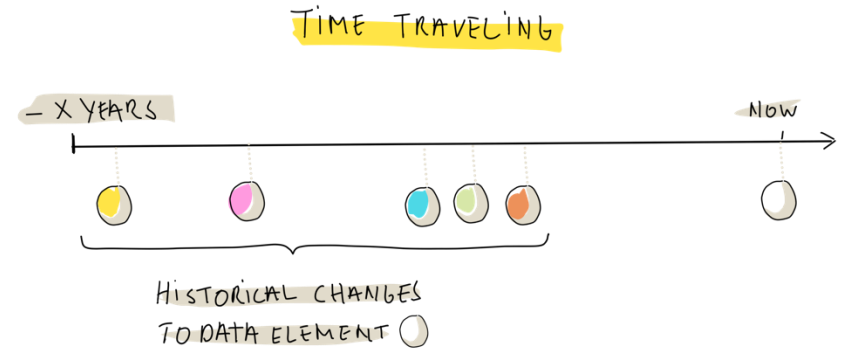
Example: Data Ingestion

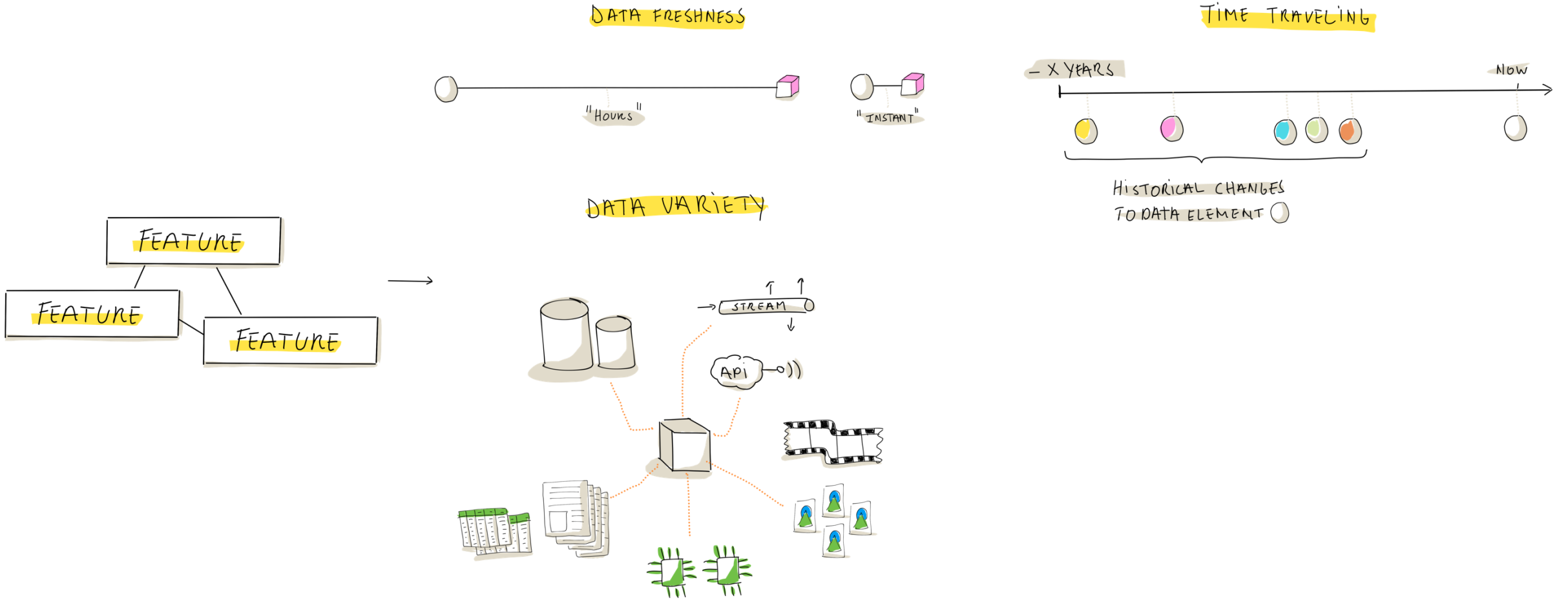


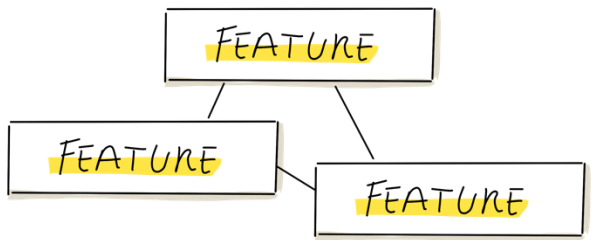

BUDGET CONSTRAINTS
FUNCTIONAL NEEDS
FEATURE
FEATURE
FEATURE
DATA SOURCES
DATA TECHNOLOGY
UI
DATA TECHNOLOGY USERS
DATA TECH STACK
IT/ ARCHIT. PRINCIPLES
BUILD REUSE BUY RENT
TECHNICAL NEEDS
STRATEGIC NEEDS


FEATURE
FEATURE
FEATURE


DATA FRESHNESS
"HOURS"
"INSTANT"

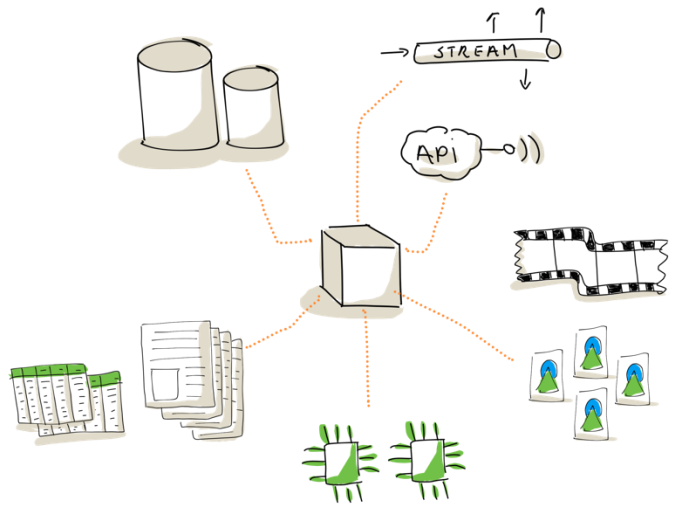

TIME TRAVELING
- X YEARS
NOW
HISTORICAL CHANGES
TO DATA ELEMENT


DATA FRESHNESS
"HOURS"
"INSTANT"
TIME TRAVELING
- X YEARS
NOW
HISTORICAL CHANGES
TO DATA ELEMENT
FEATURE
FEATURE
FEATURE
DATA VARIETY
STREAM
API

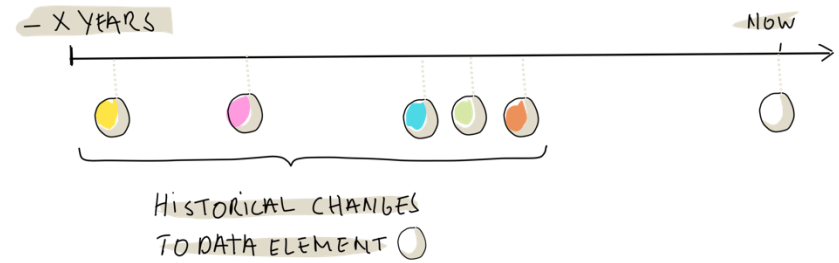


DATA FRESHNESS

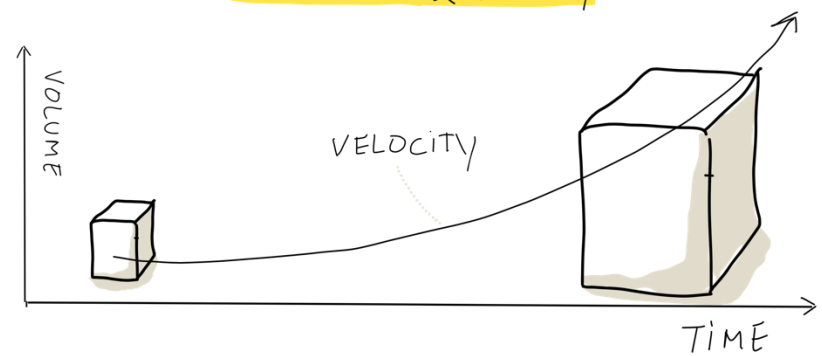
DATA VARIETY



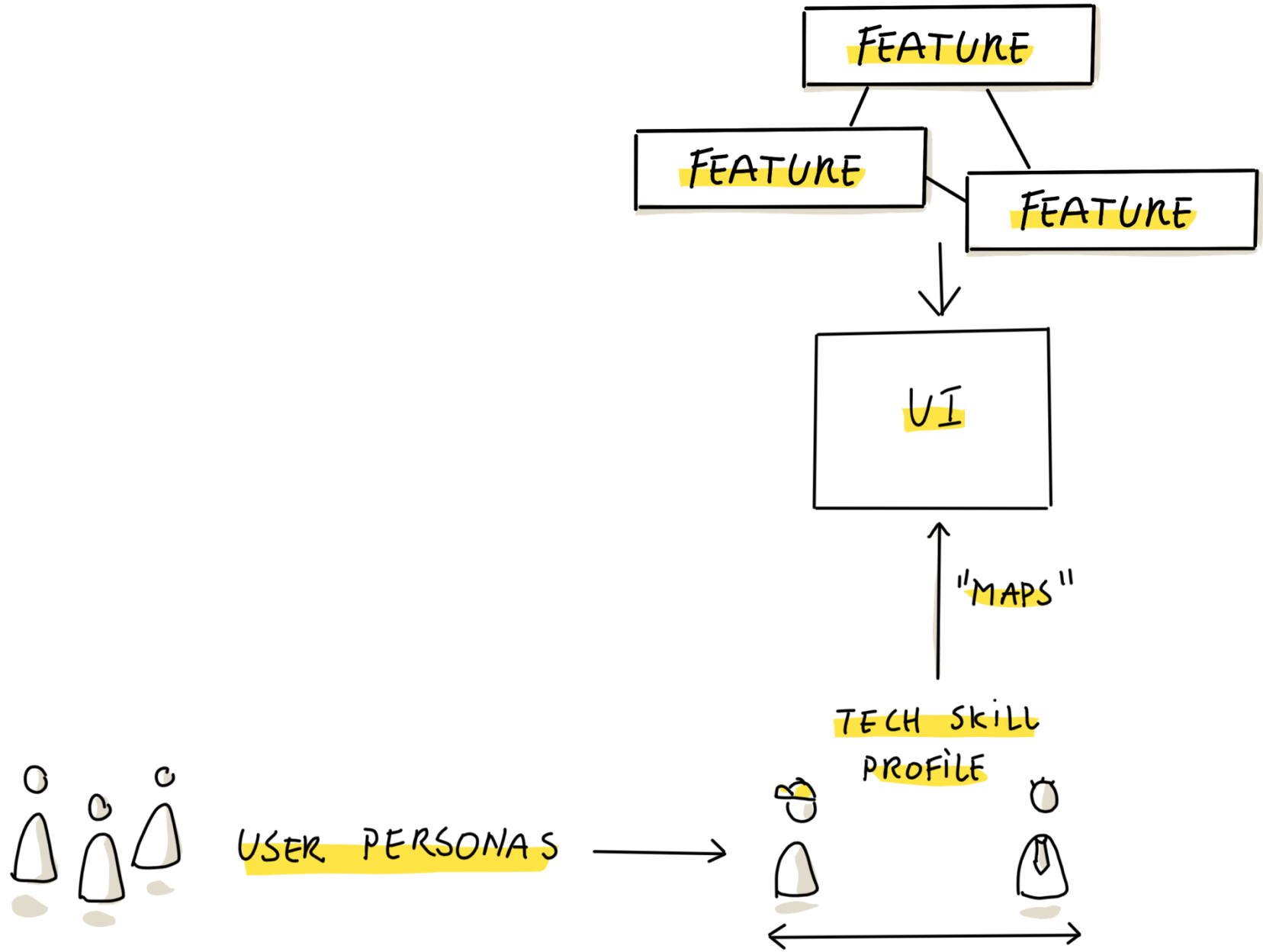
TIME TRAVELING

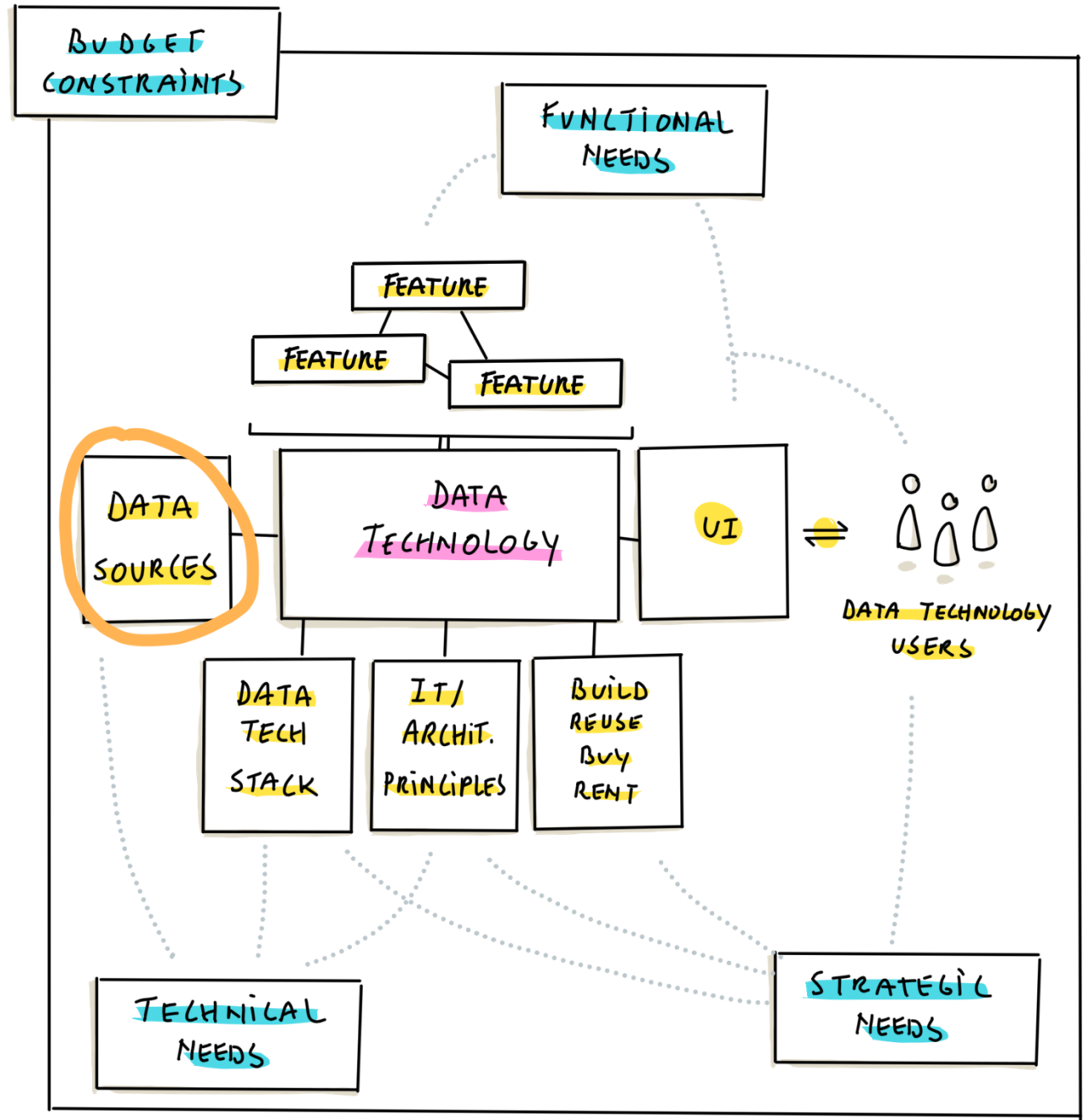


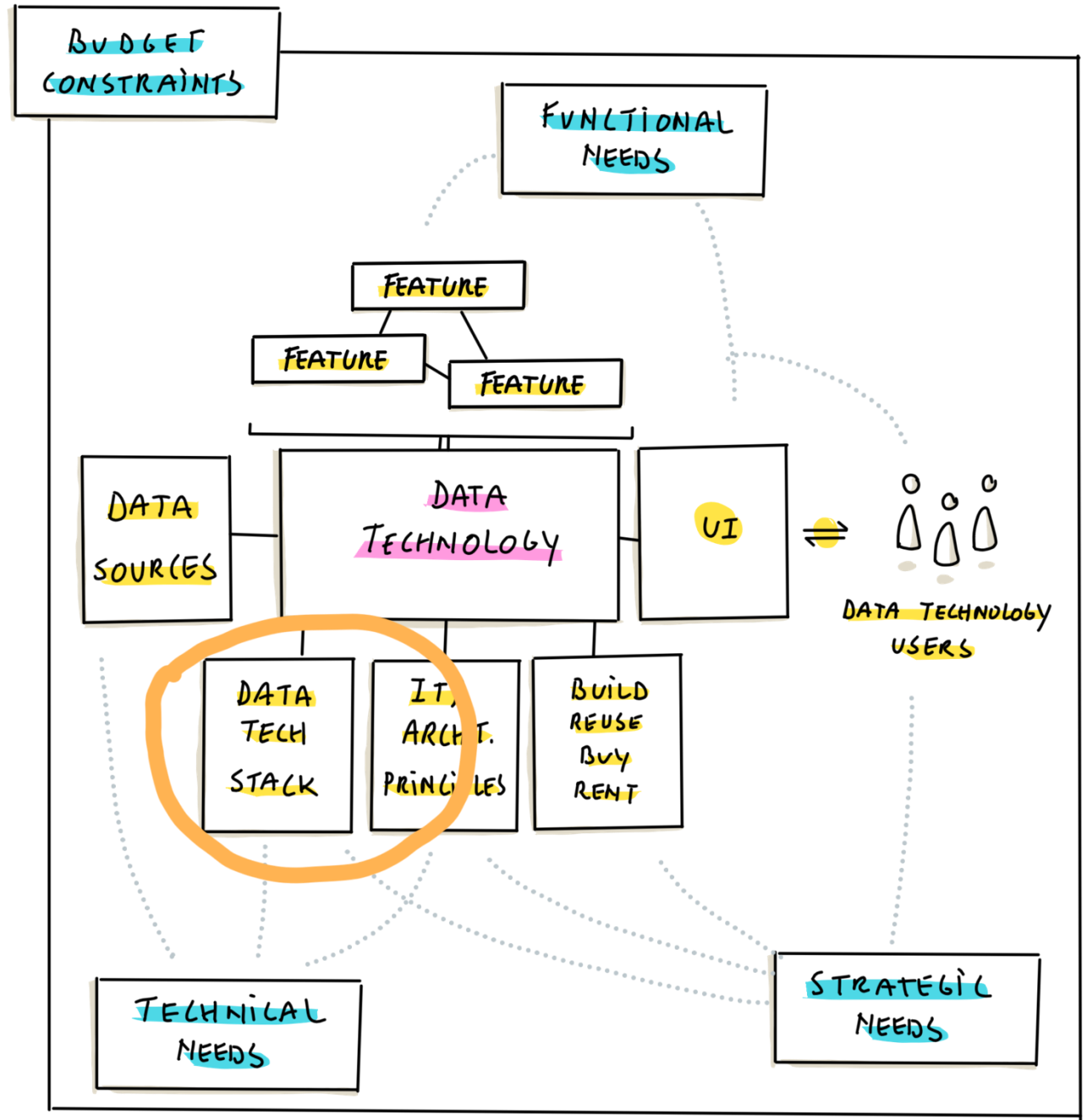
DATA VOLUME & VELOCITY



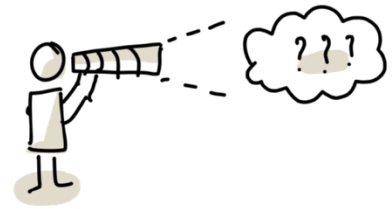
USER PERSONAS


FEATURE
FEATURE
FEATURE
UI
"MAPS"
TECH SKILL PROFILE
USER PERSONAS


BUDGET CONSTRAINTS
FUNCTIONAL NEEDS
FEATURE
FEATURE
FEATURE
DATA SOURCES
DATA TECHNOLOGY
UI
DATA TECHNOLOGY USERS
DATA TECH STACK
IT/ ARCHIT. PRINCIPLES
BUILD REUSE BUY RENT
TECHNICAL NEEDS
STRATEGIC NEEDS

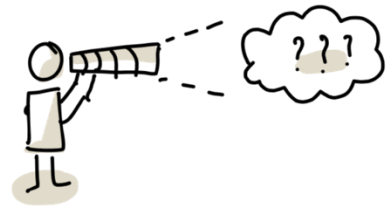

BUDGET CONSTRAINTS
FUNCTIONAL NEEDS
FEATURE
FEATURE
FEATURE
DATA SOURCES
DATA TECHNOLOGY
UI
DATA TECHNOLOGY USERS
DATA TECH STACK
IT ARCH. PRINCIPLES
BUILD REUSE BUY RENT
TECHNICAL NEEDS
STRATEGIC NEEDS

DATA PLATFORM ARCHITECTURE



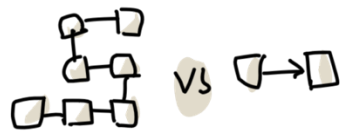
IT STRATEGY

DATA PLATFORM ARCHITECTURE

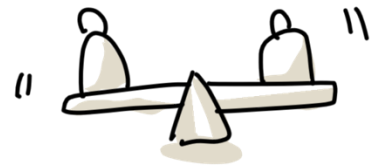


IT STRATEGY

VENDOR LOCK-IN

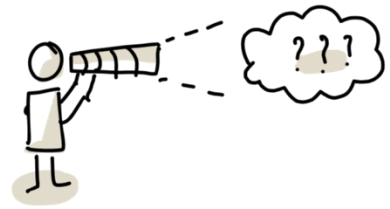


SIMPLICITY



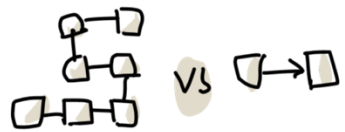
MAINTENANCE VS CONTROL

DATA PLATFORM ARCHITECTURE

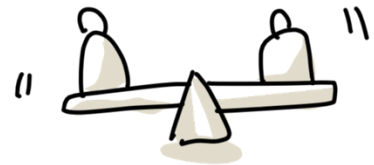


IT STRATEGY

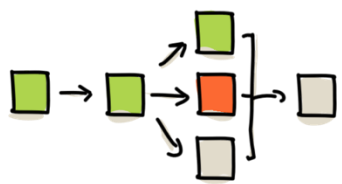
VENDOR LOCK-IN



SIMPLICITY

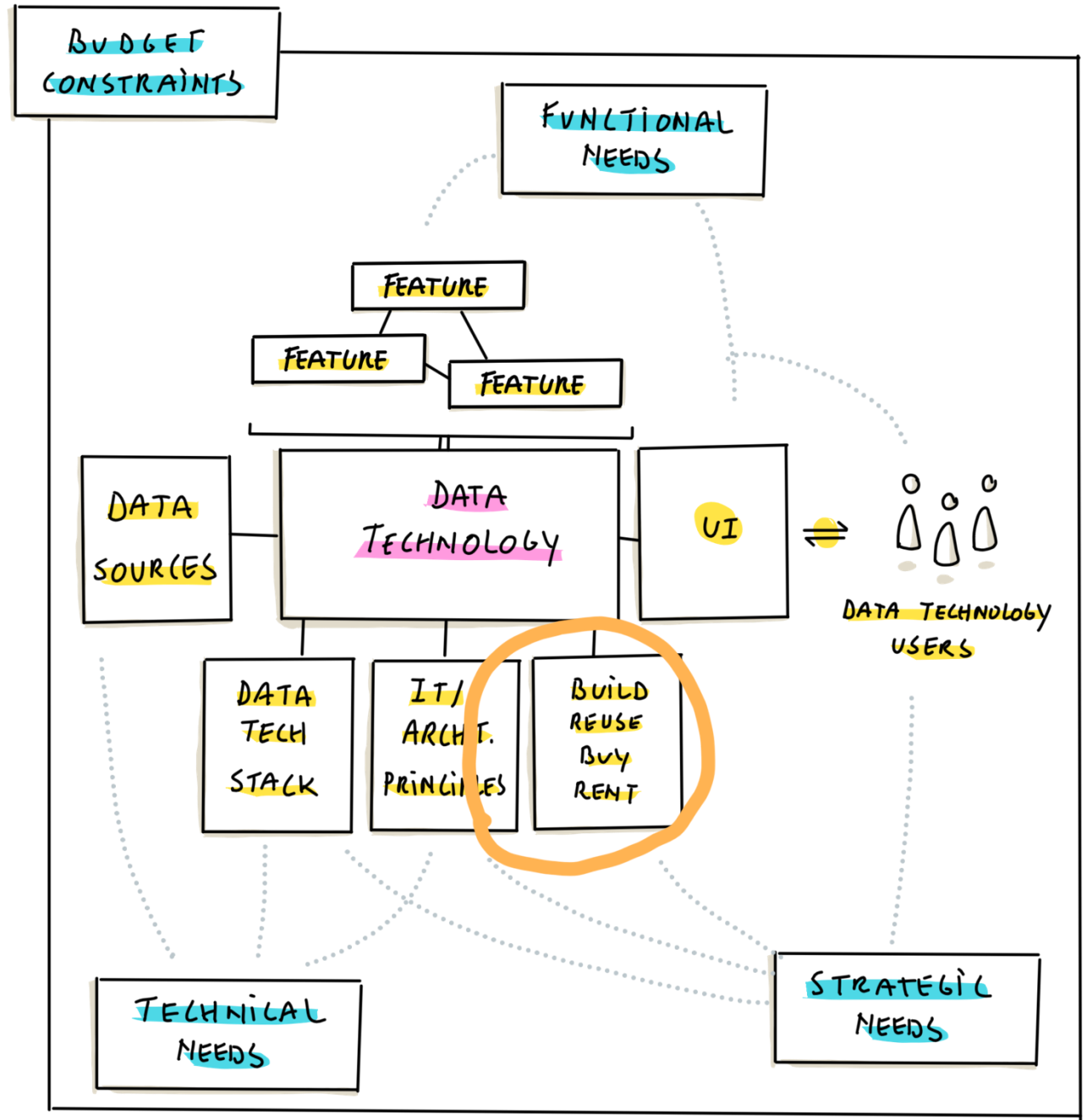


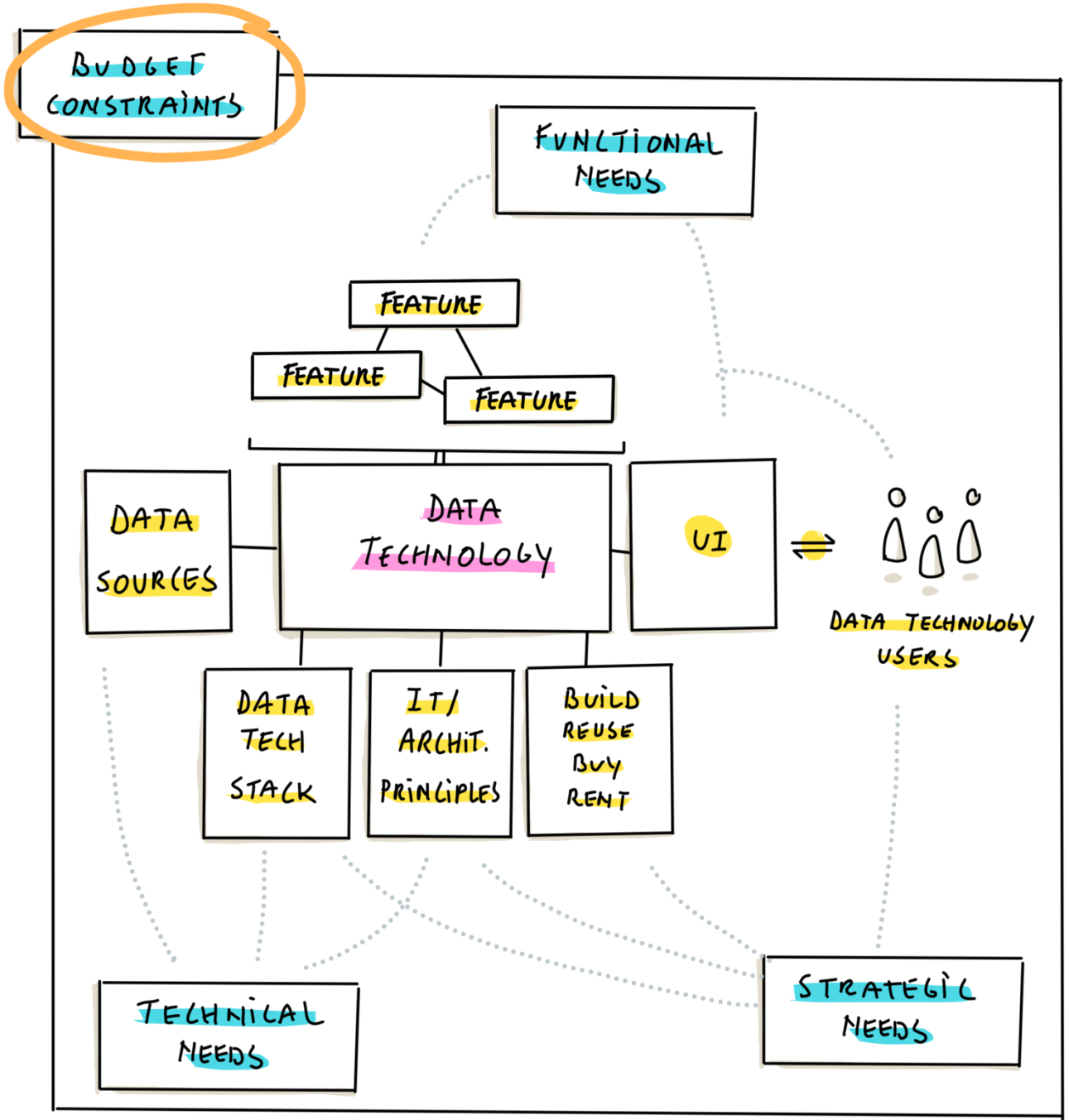
MAINTENANCE VS CONTROL

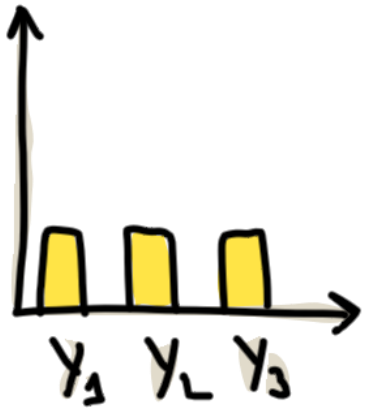


ORCHESTRATION & MONITORING

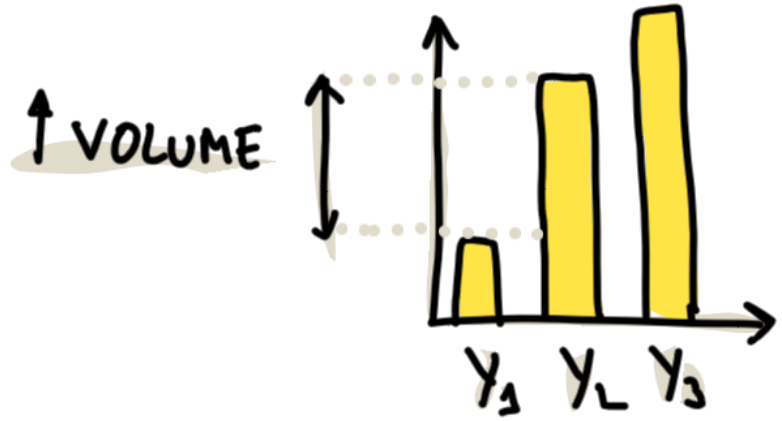
TOOL ECOSYSTEM


BUDGET CONSTRAINTS
FUNCTIONAL NEEDS
FEATURE
FEATURE
FEATURE
DATA SOURCES
DATA TECHNOLOGY
UI
DATA TECHNOLOGY USERS
DATA TECH STACK
IT/ ARCHT. PRINCIPLES
BUILD REUSE BUY RENT
TECHNICAL NEEDS
STRATEGIC NEEDS

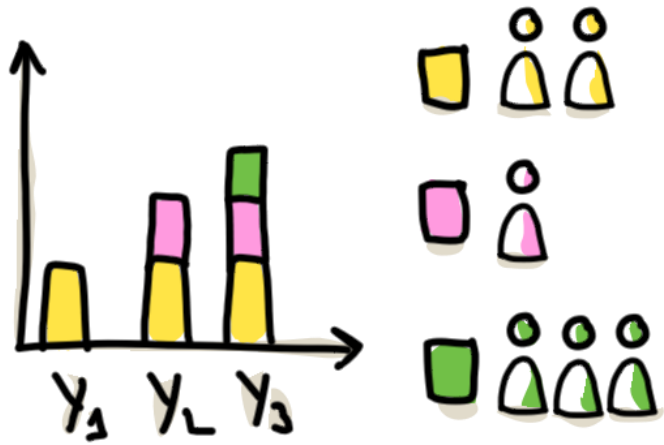

BUDGET CONSTRAINTS
FUNCTIONAL NEEDS
FEATURE
FEATURE
FEATURE
DATA SOURCES
DATA TECHNOLOGY
UI
DATA TECHNOLOGY USERS
DATA TECH STACK
IT/ ARCHIT. PRINCIPLES
BUILD REUSE BUY RENT
TECHNICAL NEEDS
STRATEGIC NEEDS



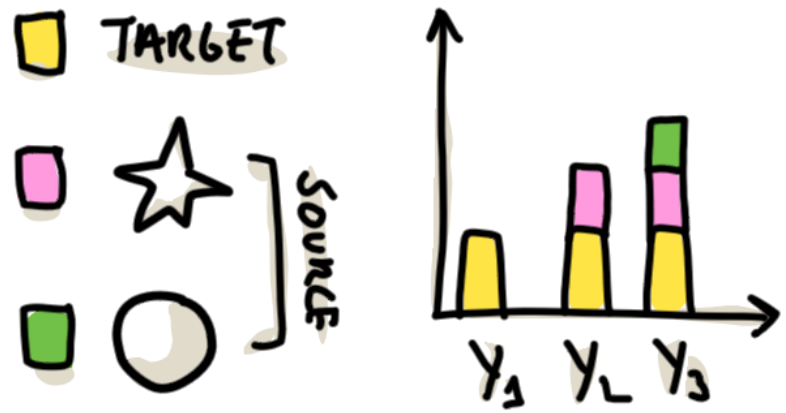
FIXED PRICE



PER VOLUME



PER USER



PER SOURCE / TARGET

PART	CRITERIUM	DESCRIPTION
Functional / Strategic	Future users	What will be the main profile of the future technology users?
Functional	Features	Which technology features are important?
Technical	Data Sources	What are the exact properties of the data sources?
Technical	Data Tech Stack	In which data architecture should the new technology be implemented?
Technical	Eco System	Which data tool eco-system is in use and will we keep using?
Technical / Strategic	IT Strategy	Cloud? Open Source? Micro Services? …
Technical / Strategic	Simplicity	Choos the most simple (= direct) solution
Technical / Strategic	Vendor Lock-In	Avoid lock-in
Technical / Strategic	Monitoring / Orchestration	Integration in overall monitoring / orchestration
Budget Constraints	Pricing strategy	
Budget Constraints	TCO	

PART	CRITERIUM	DESCRIPTION
Purchasing / Strategic	Turnover vendor	What's the turnover of the vendor in the last X years?
Purchasing / Strategic	Support in EU	Is there a support organization in the EU?
Purchasing / Strategic	Product Maturity & Stability	What's the maturity of the product? (Can also be evaluated for OS)
Purchasing / Strategic	Community	What's that response time to resolve critical issues of an OS product?

EXERCISE : SELECTION CRITERIA WITHIN YOUR USE CASE?

Pick a needed technology

PART	CRITERIUM	DESCRIPTION
Functional / Strategic	Future users	What will be the main profile of the future technology users?
Functional	Features	Which technology features are important?
Technical	Data Sources	What are the exact properties of the data sources?
Technical	Data Tech Stack	In which data architecture should the new technology be implemented?
Technical	Eco System	Which data tool eco-system is in use and will we keep using?
Technical / Strategic	IT Strategy	Cloud? Open Source? Micro Services? …
Technical / Strategic	Simplicity	Choos the most simple (= direct) solution
Technical / Strategic	Vendor Lock-In	Avoid lock-in
Technical / Strategic	Monitoring / Orchestration	Integration in overall monitoring / orchestration
Budget Constraints	Pricing strategy	
Budget Constraints	TCO	

How to use in Practice

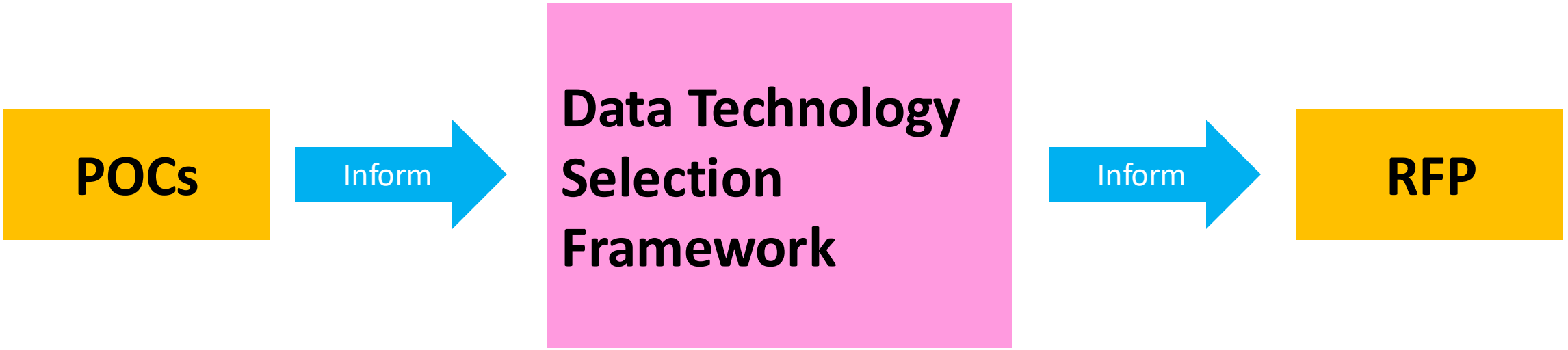


Table of Contents