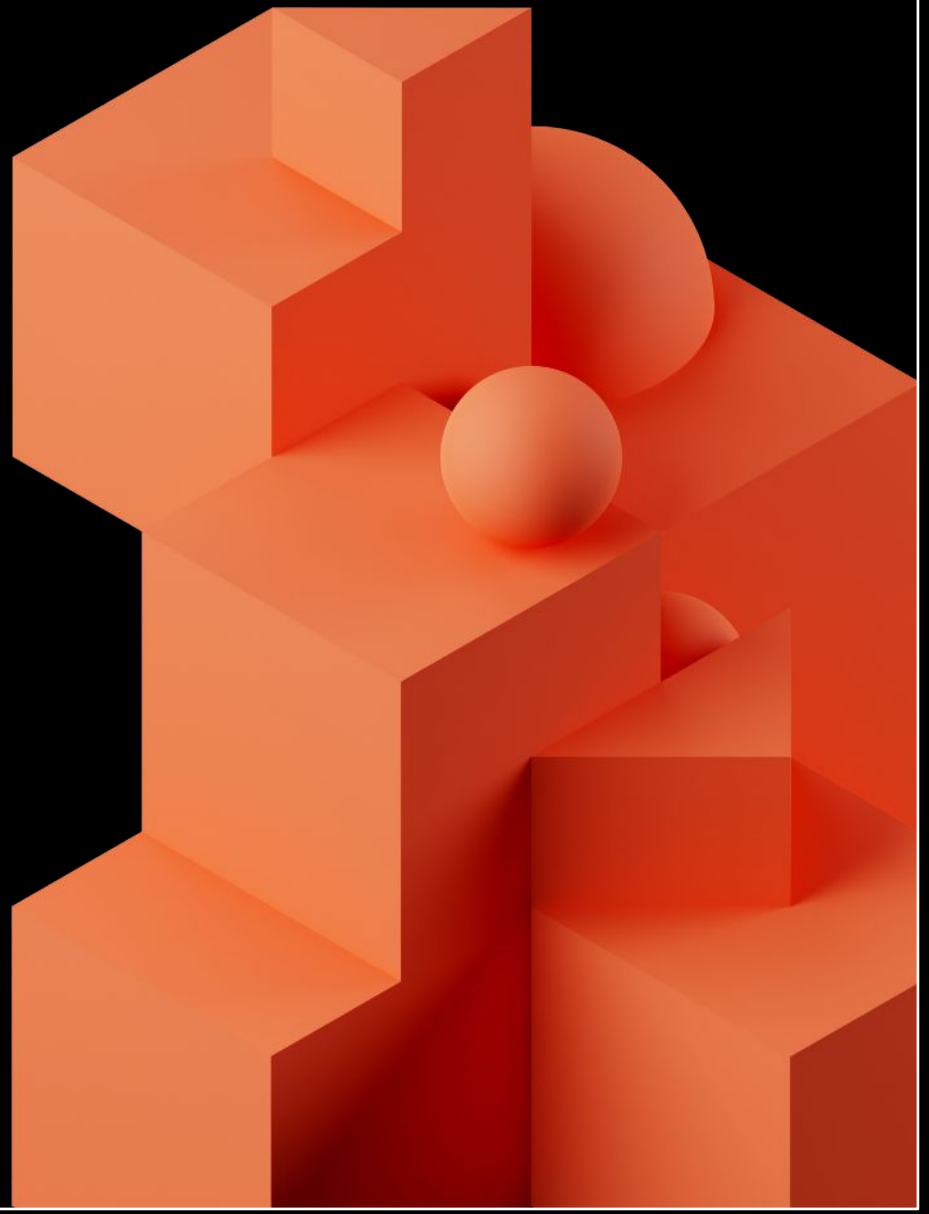
DATA+AI SUMMIT
BY databricks

Engineers Shouldn't Write Data Governance Policies

Kieran Taylor | Instacart

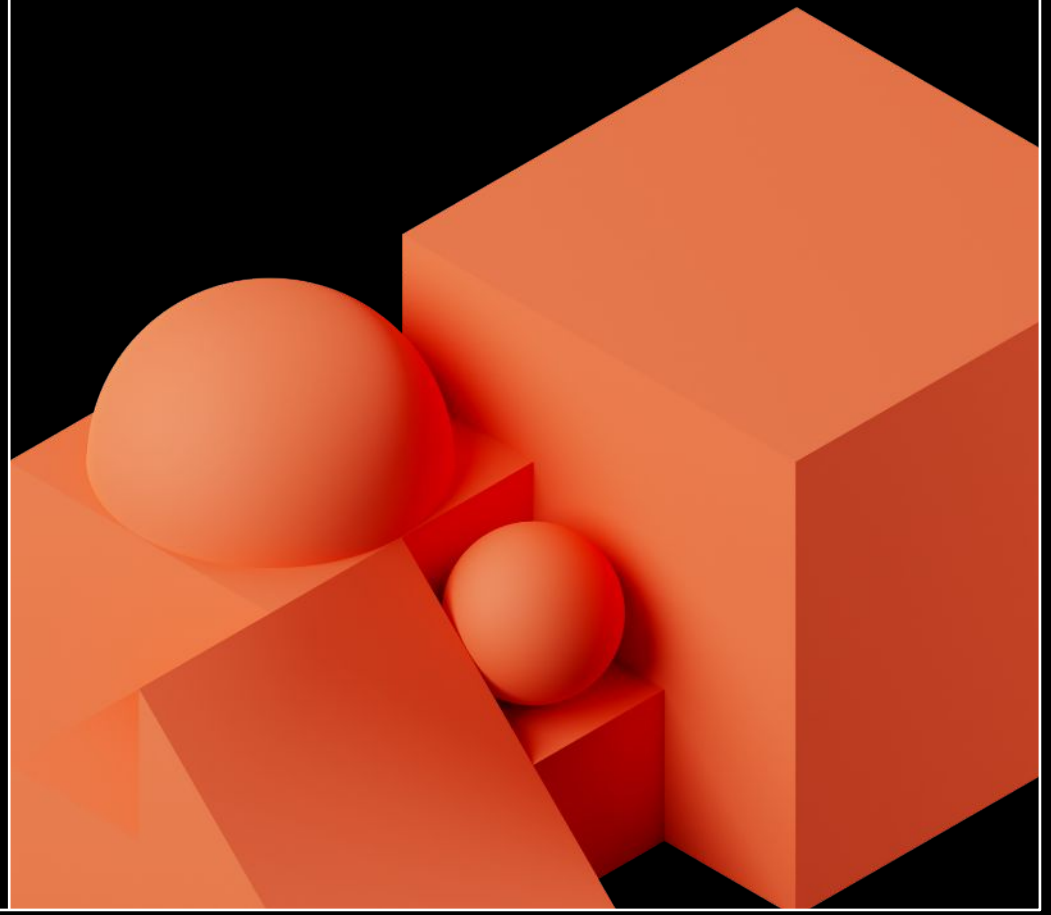


Databricks
2023

Agenda

- How we got here
- Answering: “Should this consumer be able to access this data?”
- Our solution
  aka leaving “Get a managers approval” flow

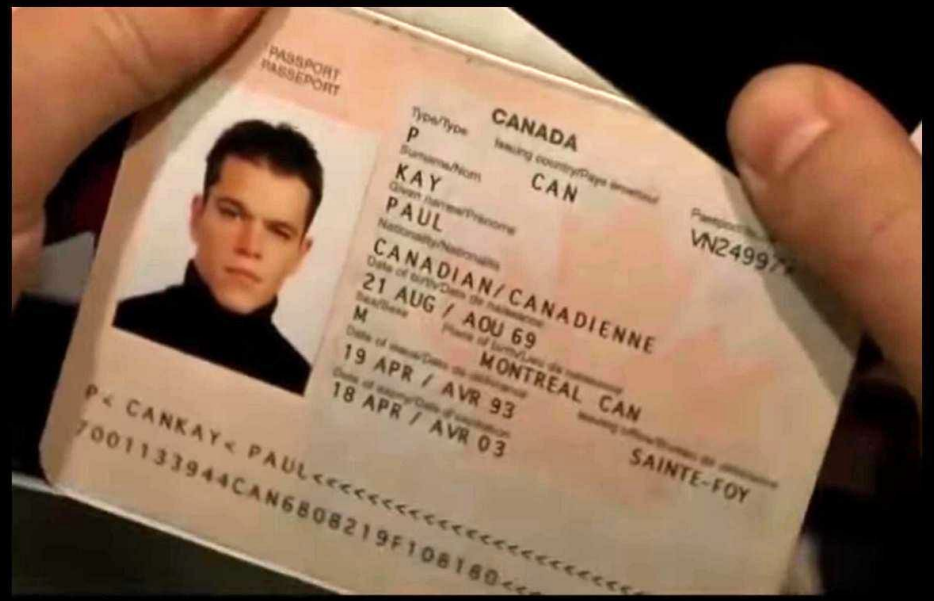
How we got here

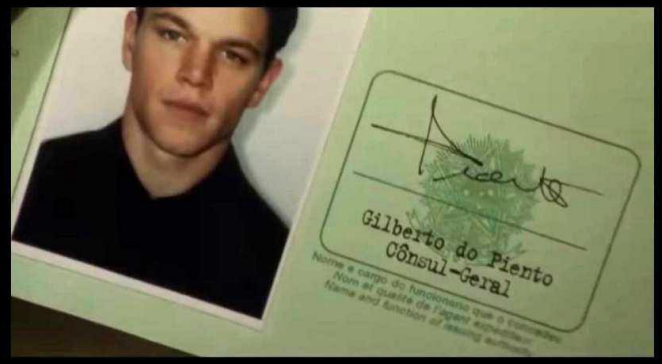


How we got here

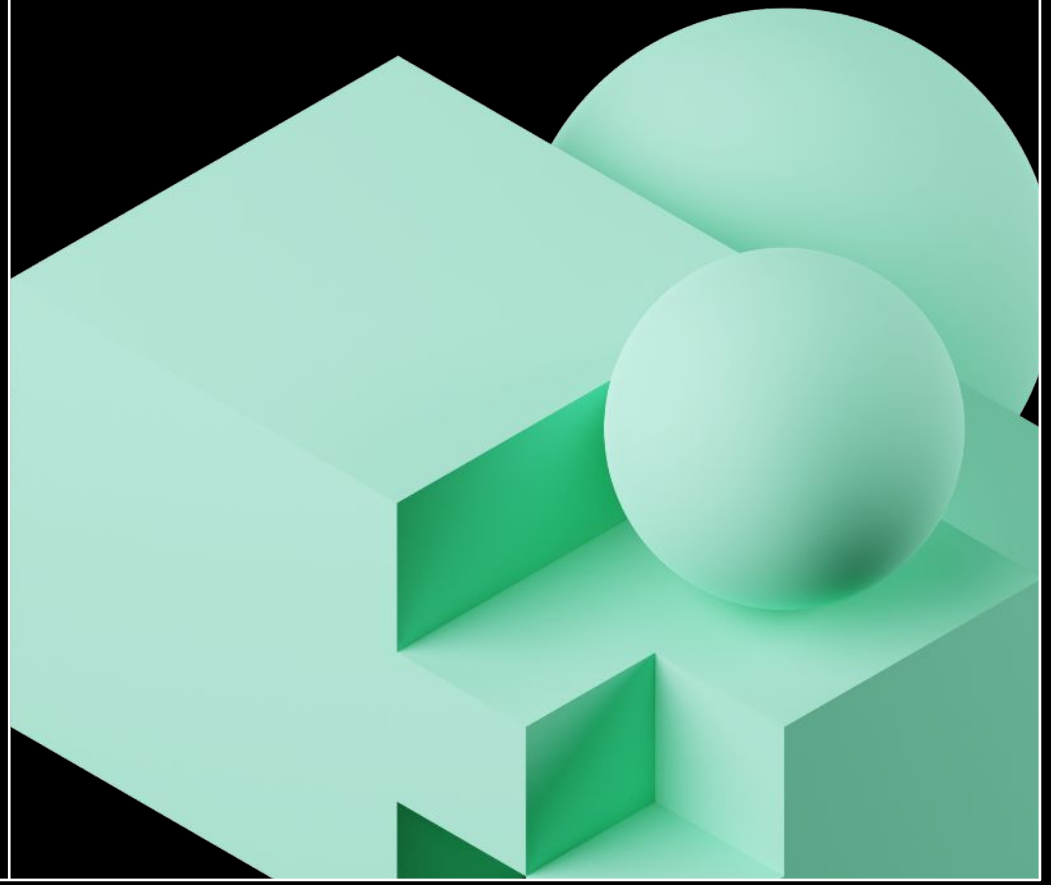
- 250k+ tables/views... in one DB
- Custom agreements with retailers
- Data living in multiple warehouses

How we got here





Answering: “Should this consumer be able to access this data?”



Answering: “Should this consumer be able to access this data?”

What information do we need?

- **What are the business meaningful attributes of this data?**
- **What are the business meaningful attributes of the user / use case?**
- **What are the business rules around access?**

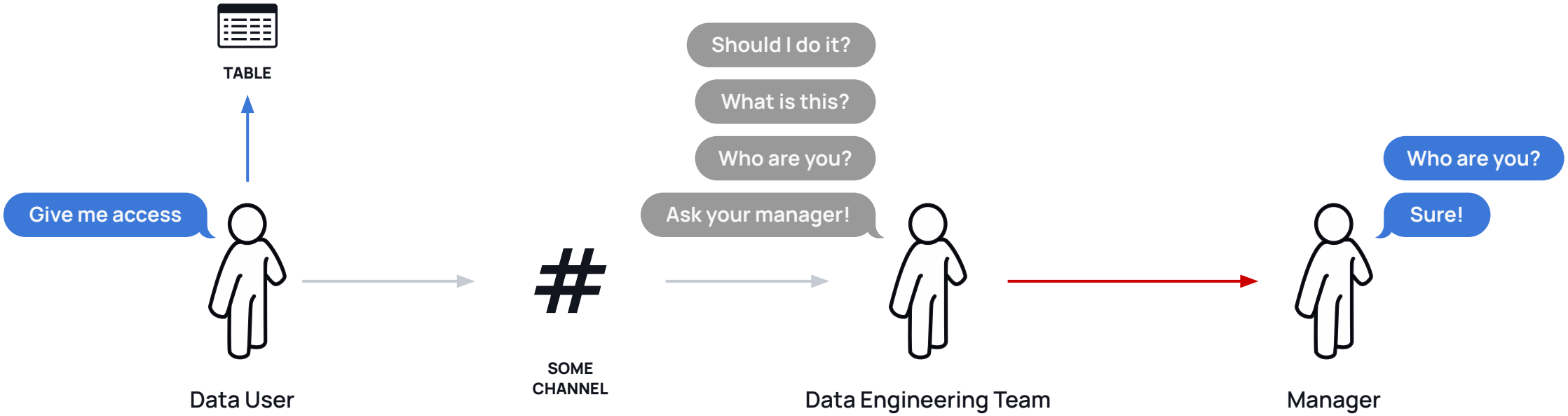
Answering: “Should this consumer be able to access this data?”

Who is best placed to answer this?

- **What are the business meaningful attributes of this data?**
  Data owner
- **What are the business meaningful attributes of the user / use case?**
  HR + Management
- **What are the business rules around access?**
  Governance / Compliance / Legal + Data Owner

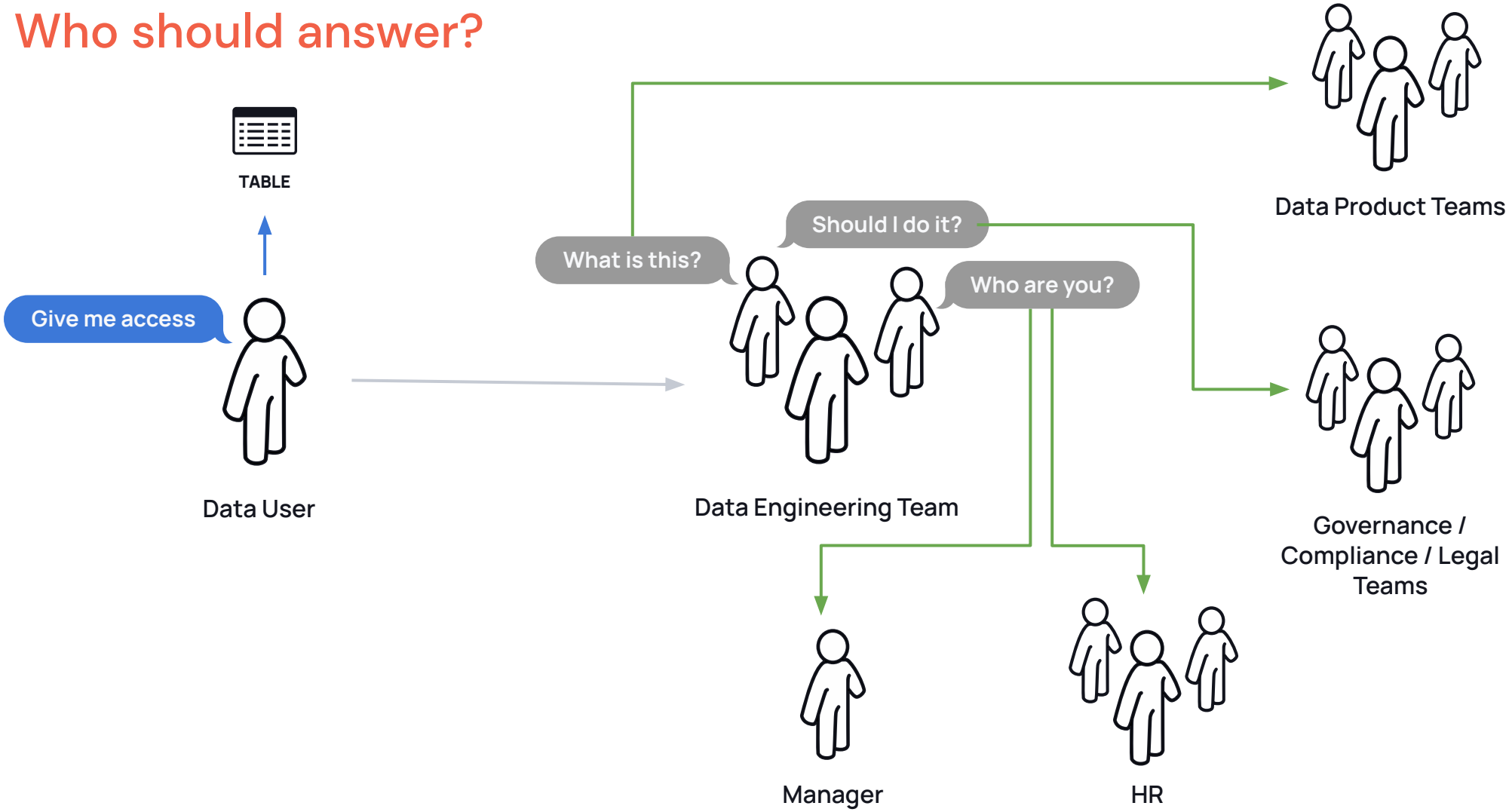
Answering: “Should this consumer be able to access this data?”

Typical process flow

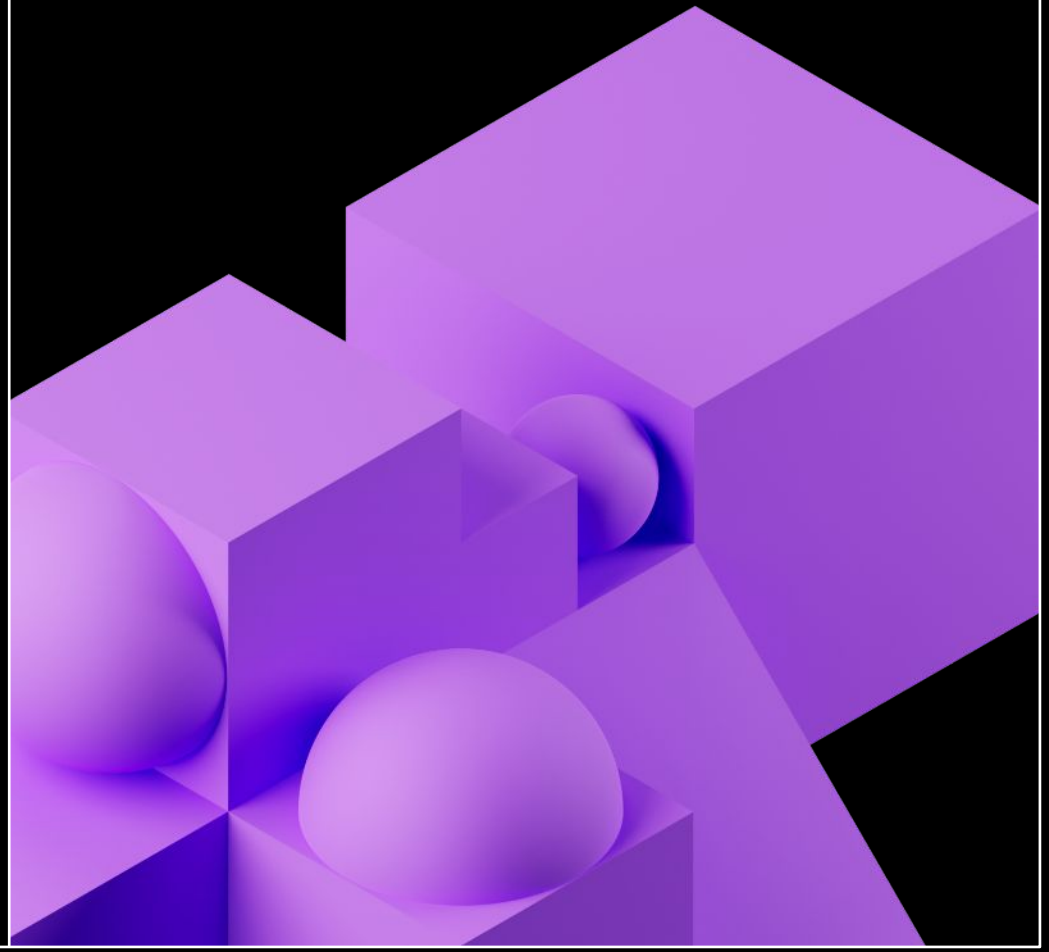


Answering: “Should this consumer be able to access this data?”

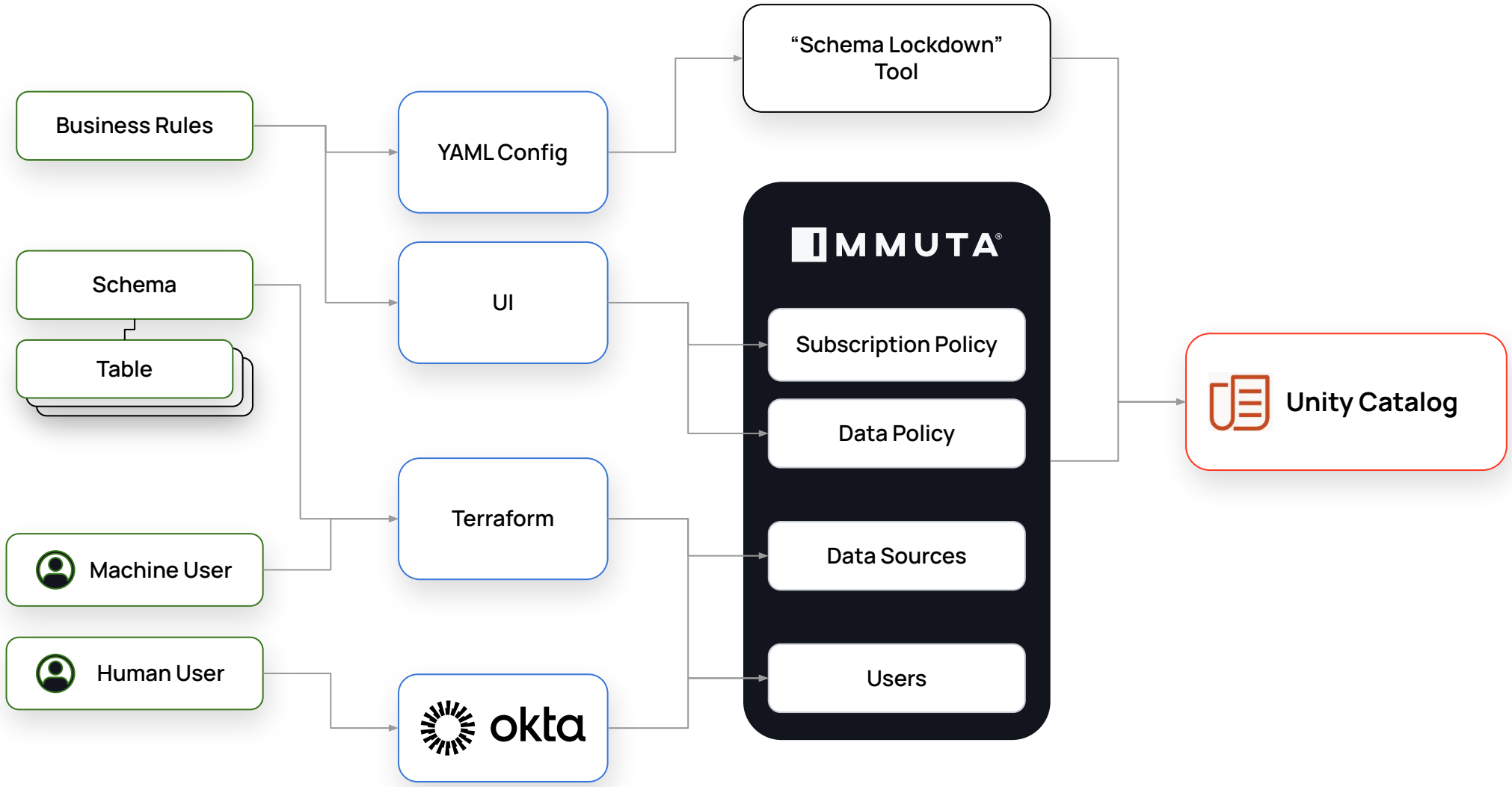
Who should answer?



Our solution



Our solution



Our solution

Subscription Policy
Allow users to subscribe when user possesses attribute in organization Instacart on data sources tagged `sensitivity.low`

Data Policy
Mask columns tagged `sensitive.pi` using hashing for everyone except when a user is a member of group `PII_USERS`

Our solution

Instacart + Immuta

Instacart owns:

- Tagging data
- Tagging users
- Writing our policies

Immuta owns:

- Translating policies to source system primitives
- Reporting on current state
- Giving context (purpose) of access

Our solution

Why is this difficult to achieve today?

- Reimplementing on various systems is very difficult
- No upper bound on access
- Visibility is poor

Our solution

Goals

- Separation of concerns
- Single pane of glass
- Visibility baked in

Benefits

- Cross-team requirements are reduced
- Time-to-data is reduced
- Rules are maintainable

Our solution

Challenges to Address

- A lot of legacy sources to ingest
- Intermediate state can be confusing
- Downstream tools (e.g. BI) where user identity is lost


databricks