RBAC High Level Design

Document Owner(s)	@Jay Jiang
Work Package	
Status	REVIEWED
Consulted	@Yolandi Jardim @Ben Sykes (Deactivated) @Vishnu Devarajan (Unlicensed) @Java Mukherjee @Davood Shaiek
List of Approvers	

Design Summary

Access to databricks is controlled by a combination of 3 groups:

- an individual's function in a team (Functional Group); this controls which environments the individual has access to and what actions can be performed within that environment.
- an individual's data classification clearance (Classification Group); this controls what data the individual can see.
- an individual’s data compliance clearance (Compliance Group); this controls additional special legal requirements for data handling (requires additional mandatory training).

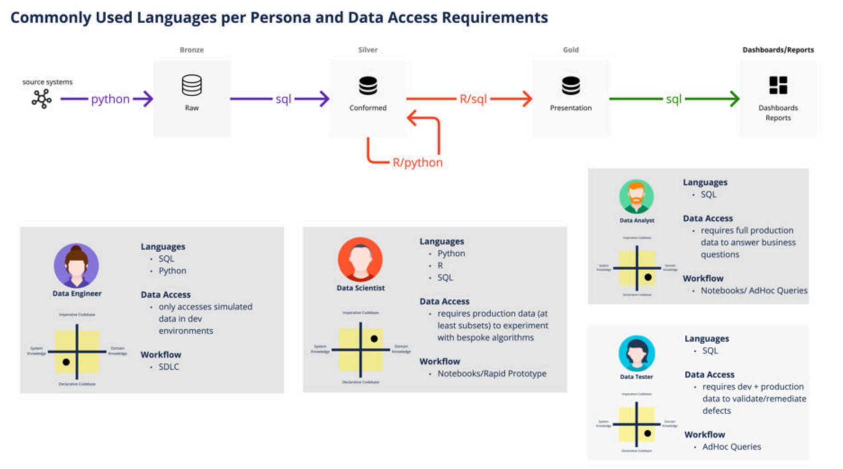
RBAC Implementation - Future Networks - Data Hub and Analytics Project - Confluence for details on implementation.

Functional Groups

These denote the job function of an individual; or personas within a team.

Data Product Team Functions (Personas)

- data viewer (business users)
- data analyst
- data engineer
- data scientist
- data tester



Commonly Used Languages per Persona and Data Access Requirements

Data Engineer

Languages
- SQL
- Python

Data Access
- only accesses simulated data in dev environments

Workflow
- SDLC

Data Scientist

Languages
- Python
- R
- SQL

Data Access
- requires production data (at least subsets) to experiment with bespoke algorithms

Workflow
- Notebooks/Rapid Prototype

Data Analyst

Languages
- SQL

Data Access
- requires full production data to answer business questions

Workflow
- Notebooks/ AdHoc Queries

Data Tester

Languages
- SQL

Data Access
- requires dev + production data to validate/remediate defects

Workflow
- AdHoc Queries

For smaller data product teams, these job functions collapse into a single person; for example a single developer might be developing the *ingestion pipeline from source system* (i.e. serving the function of a data engineer) and also *developing the business logic for how to process that source data into reports* (i.e. serving the function of a data analyst).

In this example, the 2 functions (engineer vs. analyst) actually have different requirements for data access (refer to diagram above). From an RBAC design perspective, these 2 functions' data access should be separately controlled via different AD groups with different permissions associated to the respective AD group (via access groups, see below). Practically, the same developer may still be able to be assigned to both AD groups, and thereby inheriting all necessary permissions required to perform both functions.

Data Platform Team Functions

> ⚠ By the separation of duty principle, *data product team* functions and *data platform team* functions are to be treated differently with access stacks defined separately.

- platform engineer
- platform tester

These data platform functions/personas look after the data platform from an infrastructure perspective. They do not necessarily need access to actual data on the platform. Their responsibilities are largely to develop additional features of the platform itself, e.g. cluster policies, bucket policies, catalog/workspace configuration, as opposed to using the data platform to produce data products/output. (Refer to `linear service lineage` in design principles).

⌄ Technical Implementation Details

Note that these are technical implementation details, a future implementation may deprecate the usage of `Access Groups`.

Access Groups

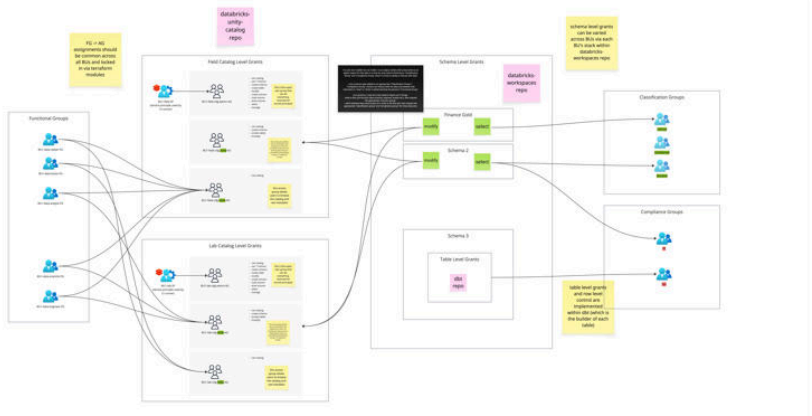
Access groups are defined per each databricks workspace, `grants` and other `permission` related databricks resources are typically assigned to `access groups` rather than `functional` `grops` ; this is to make implementation more repeatable across environments.

> ⚠ Users are never to be assigned to **Access Groups** directly.
>
> Only **Service Principals** may be assigned to **Access Groups** directly.

Functional x Access Association

Permissions are inherited by `functional group` when they are assigned as members of `access groups` .

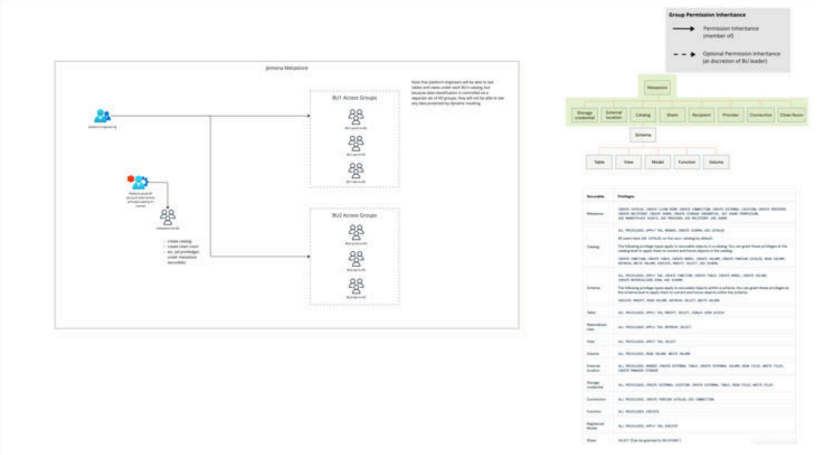
Detailed Design - Data Product Team Functional Groups


FG -> AG assignments should be common across all BUs and locked in via terraform modules
databricks-unity-catalog repo
Field Catalog Level Grants
Lab Catalog Level Grants
Functional Groups
Schema Level Grants
databricks-workspaces repo
schema level grants can be varied across BUs via each BU's stack within databricks-workspaces repo
Finance Gold
Schema 2
Schema 3
modify
select
Table Level Grants
dbt repo
Classification Groups
Compliance Groups
table level grants and row level control are implemented within dbt (which is the builder of each table)

[🔗 A private Miro board]

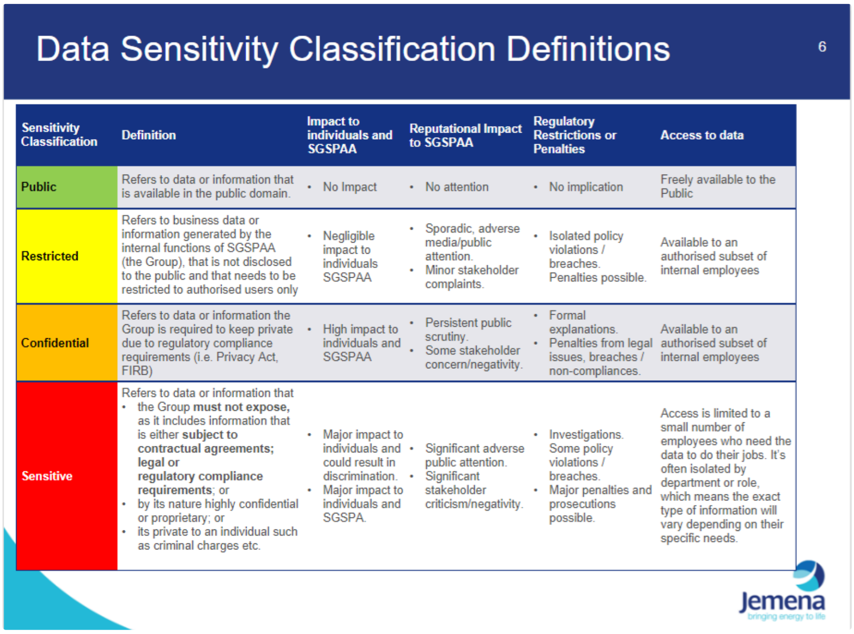
Detailed Design - Data Platform Team Functional Groups

Platform team (`cadm-*`) has a different set of permissions than Data Product teams.



Classification Groups

This design follows Jemena data governance team’s data sensitivity classification definition.



Data Sensitivity Classification Definitions

6

Sensitivity Classification	Definition	Impact to individuals and SGSPAA	Reputational Impact to SGSPAA	Regulatory Restrictions or Penalties	Access to data
Public	Refers to data or information that is available in the public domain.	• No Impact	• No attention	• No implication	Freely available to the Public
Restricted	Refers to business data or information generated by the internal functions of SGSPAA (the Group), that is not disclosed to the public and that needs to be restricted to authorised users only	• Negligible impact to individuals SGSPAA	• Sporadic, adverse media/public attention. • Minor stakeholder complaints.	• Isolated policy violations / breaches. Penalties possible.	Available to an authorised subset of internal employees
Confidential	Refers to data or information the Group is required to keep private due to regulatory compliance requirements (i.e. Privacy Act, FIRB)	• High impact to individuals and SGSPAA	• Persistent public scrutiny. • Some stakeholder concern/negativity.	• Formal explanations. • Penalties from legal issues, breaches / non-compliances.	Available to an authorised subset of internal employees
Sensitive	Refers to data or information that • the Group **must not expose,** as it includes information that is either **subject to contractual agreements; legal or regulatory compliance requirements**; or • by its nature highly confidential or proprietary; or • its private to an individual such as criminal charges etc.	• Major impact to individuals and could result in discrimination. • Major impact to individuals and SGSPA.	• Significant adverse public attention. • Significant stakeholder criticism/negativity.	• Investigations. Some policy violations / breaches. • Major penalties and prosecutions possible.	Access is limited to a small number of employees who need the data to do their jobs. It’s often isolated by department or role, which means the exact type of information will vary depending on their specific needs.

A property address is **public** information.

Information about the property’s electricity meter is **restricted** .

Information about the property’s electricity usage is **confidential** (to be clarified)

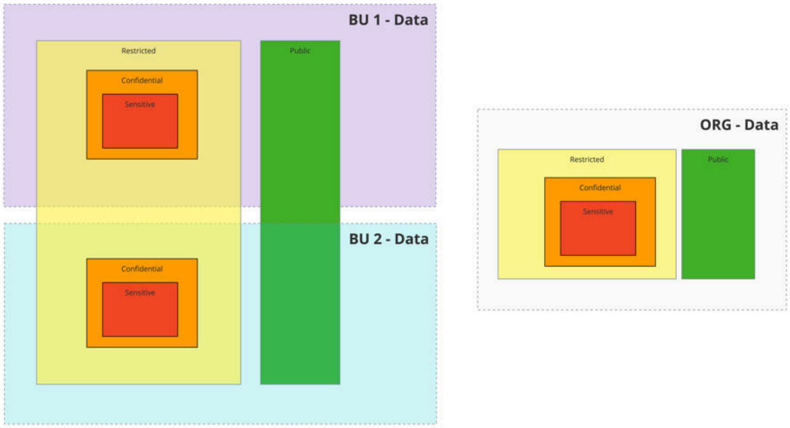
Who lived in this property over the years is **confidential.**

Information private to the resident of that individual is **sensitive.** *(In this example, Jemena may not have collected any sensitive information).*

Classification across BUs

Below is a possible topological implementation across 2 different business units.

- Each business unit’s `sensitive` and `confidential` are visible only to members of that business unit
- Each business unit’s `restricted` (a.k.a internal) data may be shared internally within Jemena to other business units if approved by the owning business unit.
- Each business unit’s `public` data may be shared publicly but is internal to Jemena by default.



Compliance Groups

Compliance groups correspond to additional attributes that describe how data should be handled (separate to sensitivity) in order to satisfy Jemena’s legal compliance obligations.

Based on information in Hopex, we introduced 2 Compliance Groups to ensure compliance with the relevant regulations.

- APP-DataHub-SOI
  - **SOI, Sensitive Operational Information,** as defined per relevant regulation by FIRB
- APP-DataHub-PII
  - **PII, Personally Identifiable Information,** as defined per Privacy Act (or any additional regulation Jemena needs to adhere to, e.g. GDPR)

Compliance is applied holistically Org wide, regardless of BU; the reasoning is that there should only be a single canonical decision on whether any particular data is SOI or not (by Jemena legal team). While BUs can have an input over the decision, once applied, it is applied across all of Jemena.

APP-DataHub-SOI group is also used to implement device level restrictions by Jemena security to prevent data exfiltration.

Masking Design Choice Details

Option 1 - Enforce masking at storage level

This option applies and enforces data classification + compliance at each medallion layer, as soon as data lands onto datahub.

It is typically used for organisations that handle large amounts of sensitive

Option 2 - Enforce masking at consumption (PREFERRED)

This option applies and enforces data classification + compliance at **gold** layer when user accesses the data (either by executing a query or viewing a dashboard).

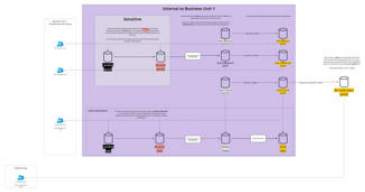
data from unknown external upstream systems (e.g. purchased from data brokers) or have development teams offshore. Landing treatment is sometimes automated using tools such as [AWS Macie]() to avoid exposure to human developers.

Pros

- Granular control of who can access data down to the developer groups/personas.

Cons

- Significantly more operational overhead (~30% additional ongoing effort in FTE terms required).
- Difficult to modify if classification rules/grades change.
- Access to production `landing` and `bronze` in lower environments is not possible; development must be carried out using simulated data in lower environments.



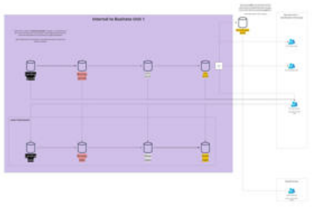
It is typically used for organisations that handle only internal data and have full governance of developer teams.

Pros

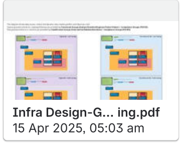
- Simpler to implement and operate, flexible to change.

Cons

- Developers have access to unmasked data in `landing`, `bronze`, `silver` layers; as they need to see the data to apply classification for `gold` views.



For details on how Compliance Group and Classfication Groups interact across BUs, please refer to attached PDF.



Infra Design-G... ing.pdf
15 Apr 2025, 05:03 am