

# Data Architect - Security System


João Fonseca	0.1	Initial Version	26/12/2022

## Summary

[1- Introduction](#)

[1.1- Purpose of the document](#)

[1.2- Intended Audience](#)

[2- How it works](#)

[2.1- Description](#)

[2.2- Security Process](#)

[3- Security Model](#)

[3.1- Description](#)

[3.2- Model](#)

[3.2- Tables](#)

[4- Talend Jobs](#)

## 1- Introduction

### 1.1- Purpose of the document

This document describes how the security access will work.

## 1.2- Intended Audience

This document is intended for the Data Architectures, Data Engineering, operational and data Visualization team. It will be used as reference for any project or domain for the developments of the models.

## 2- How it works

### 2.1- Description

The objective of this security system is to control the access to the data independently to the tool or project. This model needs to guarantee access by data, if we want to give access only to a specific dataset like GBU or Materials and give access by object, if we want to give access to a report or folder. This needs to work for all the projects and data in Solvay GCP.  
PRIORITARIO

- Pricing quais são as tabelas de segurança

### 2.2- Security Process

The model will be splitted in 3 parts to be able to cover all the types of access by data or objects.

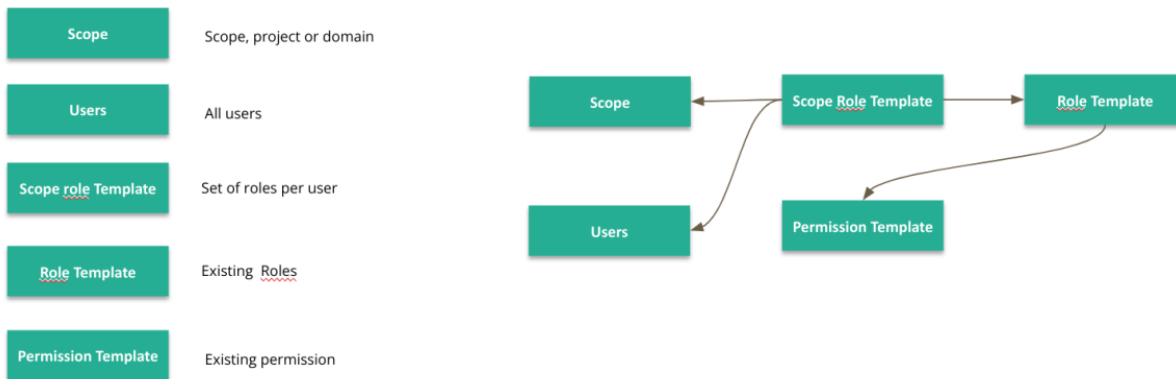
**The first part - Template** is the creation of templates to parametrize a project or a domain access, so we will create a set of tables that will allow us to parametrize the project or domain and then load the data to the final tables where this will match with the users, project, objects and data.

**The second part - Object access** is the control access to the objects (report; table etc..) this model is a set of tables that will be used to give the correct access, this will be described further.

**The third part - Data access** is the control access to the data (by GBU; Material etc..) . This model is a set of tables that will be used to give the correct access, this will be described further.

All these 3 parts will join together to provide one single model for all the Solvay access.

### First Part - Template



#### Scope:

- Contain information about the project we want to grant.
- This table will be a common table for the 3 different steps: template, object and data access
- Parametrized only one time for each project / domain.
- Every time we have a new project, this table needs to be filled.

#### User:

- Contain information about all the users in Solvay, including the GBU that person belongs to.
- This table comes from the HR data.
- This table will be a common table for the 3 different steps template, object and data access and will be parametrized only one time.

#### Scope Role template:

- This table is a template with what we want to include in the table User scope role.
- It contains information about the roles that i user can have (Relational Table).
- Every time we want to include a new user for a project or domain this table needs to be filled

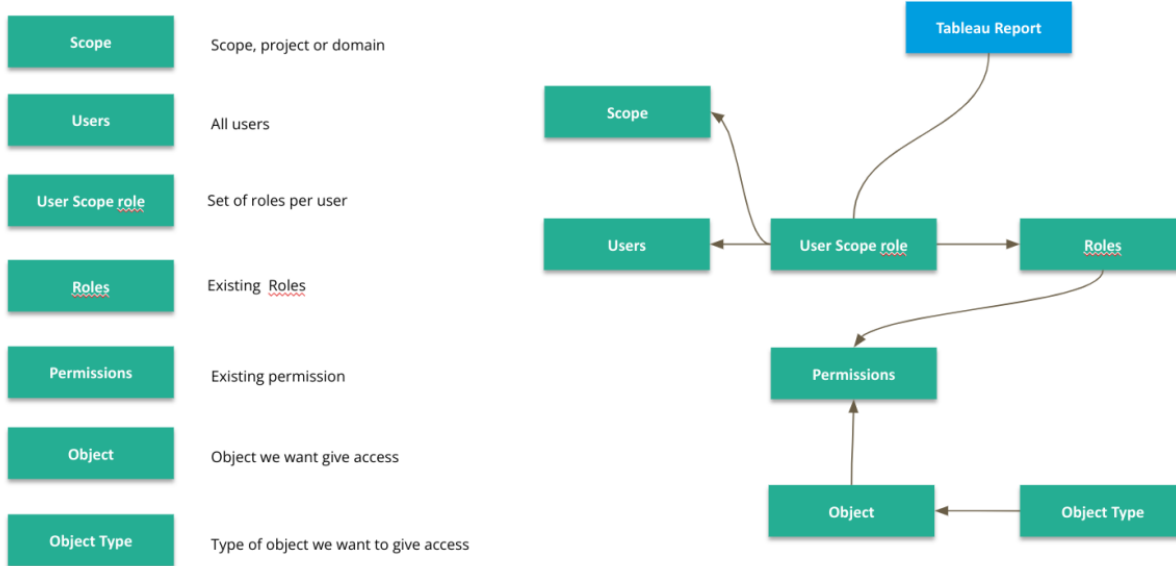
### Role Template:

- Contains information about the solvay roles (Manager, administrator, Controller etc..), each of these roles are associated with a set of permissions.
- Every time we have a new role this table needs to be filled.

### Permission Template -

- Contain information about all the permissions (Read; write; update; delete etc..)
- Every time we have a new role this table needs to be filled.

## Second Part - Object Access



### Scope:

- Contain information about the project we want to grant.
- This table will be a common table for the 3 different steps: template, object and data access
- Parametrized only one time for each project / domain.
- Every time we have a new project, this table needs to be filled.

### User:

- Contain information about all the users in Solvay, including the GBU that person belongs to.
- This table comes from the HR data.
- This table will be a common table for the 3 different steps template, object and data access and will be parametrized only one time.

### User Scope Role:

- This table is a template with what we want to include in the table User scope role.
- It contains information about the roles that i user can have (Relational Table).
- Every time we want to include a new user for a project or domain this table needs to be filled

### Roles:

- Contains information about the solvay roles (Manager, administrator, Controller etc..), each of these roles are associated with a set of permissions.
- Every time we have a new role this table needs to be filled.

### Permission:

- Contain information about all the permissions (Read; write; update; delete etc..)
- Every time we have a new permission this table needs to be filled.

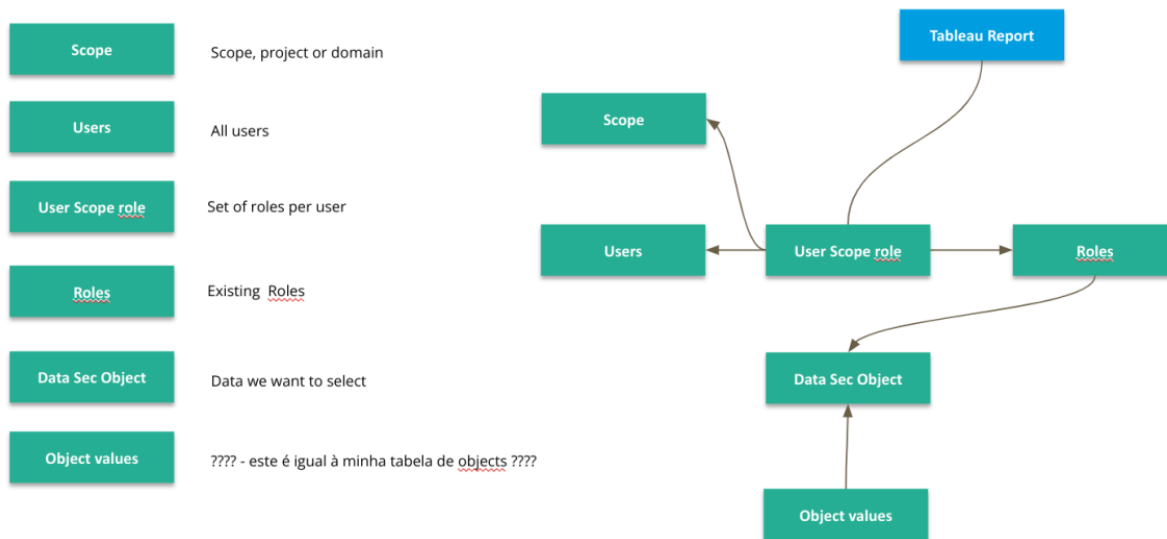
### Object:

- Objects instances
- Contains information about the objects we want to grant permissions to and the type of permission (relational table)
- Every time we want to include a new user for a project or domain this table needs to be filled

### Object type

- Type of objects, can be a report, a table, a file etc..
- Every time we have a new type of object we need to include in this table

## Third Part - Data Access



### Scope:

- Contain information about the project we want to grant.
- This table will be a common table for the 3 different steps: template, object and data access
- Parametrized only one time for each project / domain.
- Every time we have a new project, this table needs to be filled.

### Users:

- Contain information about all the users in Solvay, including the GBU that person belongs to.
- This table comes from the HR data.
- This table will be a common table for the 3 different steps: template, object and data access and will be parametrized only one time.

### User Scope Role:

- It contains information about the roles that i user can have (Relational Table).
- Every time we want to include a new user for a project or in a domain this table needs to be filled

### Roles:

- Contains information about the solvay roles (Manager, administrator, Controller etc..), each of these roles are associated with a set of permissions.
- Every time we have a new role this table needs to be filled.

### Data Sec Objects:

- Contain information about the data we want to select
- Every time we have a new role this table needs to be filled.

### Object Values:

- ????
- ???

Example:

User Table			User Scope Roles				Role Table	
user_id	user_login_id	user_full_name	user_login_id	user_scope_role_id	scope_id	role_id	role_id	role_name
1	fons1234	Joao Fonseca	fons1234	1	2	2	1	Controler
2	Madr5678	Joao Madrinha	fons1234	2	3	2	2	Data Architect
3	Gira9101	Fernando Gira	Gira9101	3	1	3	3	Manager
			Madr5678	4	1	4	4	Director
			Madr5678	5	2	4		
			Madr5678	6	3	4		

Permission			Object			Object Type		
per_id	permission_desc	Role_id	Obj_id	obj_id	obj_name	obj_type_id	obj_type_id	object_type
1	write	2	1	1	Sustainability	1	1	Folder
2	write	2	2	2	Sustainability KPI	2	2	Dashboard
3	read & write	3	3	3	Pricing Report	3	3	Single report
4	read	4	1				4	Table data
4	read	4	2					
4	read	4	3					

Scope		
Scope_id	Scope_desc	Scope_type
1	Pricing	Project
2	PCF	Project
3	Sustainability	Domain

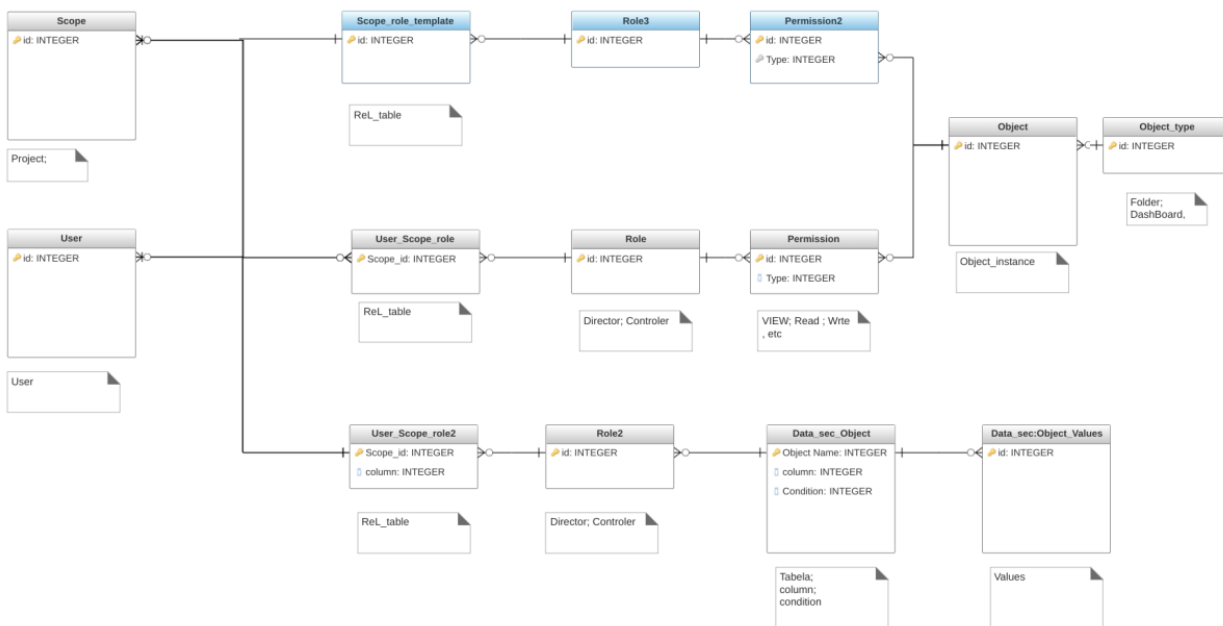
### 3- Security Model

#### 3.1- Description

Describe all the tables that need to have a full logging process.

#### 3.2- Model

The model is constituted by the 3 steps as explained, Template, object and data access



## 3.2- Tables

### Logging table in the staging (Step 1) - Users

001	user_id	Identification of the user	STRING	E.g. fd01cfb099ee11e798290000a8b263a
002	user_login_id		STRING	E.g. PRIVOAL
003	user_full_name	Name of the user	STRING	E.g. Sophie GALINAT
004	disabled	If the user exist or not exist	STRING	E.g. "F" false
005	department	Department of the user	STRING	E.g. S&T - LOF
006	e-mail	Email of the user	STRING	E.g. jp.fonse@solvay.com
007	role_id	Id of the role	STRING	E.g. bb9856605e8f11ea7e000006fb74a6
008	work_location	The physic place where the user is located	STRING	E.g. BRUXELLES (NOH)
009	gbu	GBU of the user	STRING	E.g. R&I- S&T LABS
011	hr_user_role	The job / role the person have in Solvay	STRING	E.g. R&I Engineer
012	start_date	Date when this record is valid	TIMESTAMP	E.g. 2023-01-04 17:34:04 UTC
013	end_date	Date when this record is no longer the last version	TIMESTAMP	E.g. 2023-01-04 17:34:04 UTC
014	current_flag	If this is the last version of the record	STRING	E.g. "YES"

### Logging table in the staging (Step 2) - Scope

001	role_id	Id of the role	STRING	E.g. 4ffdc2c087ab11eba3b1000096fb74a6
002	role_name	Name of the role	STRING	E.g. BatMat_PTF_ViewAccess
003	role_display_name		STRING	E.g. BatMat_PTF_ViewAccess
004	system_role	????	STRING	E.g. F
005	start_date		TIMESTAMP	E.g. 2022-11-28 14:28:15 UTC
006	end_date		TIMESTAMP	E.g. 2022-11-28 14:28:15 UTC
007	current_flag		STRING	E.g. YES

### Logging table in the staging (Step 2) - Roles

001	role_id	Id of the role	STRING	E.g. 4ffdc2c087ab11eba3b1000096fb74a6
002	role_name	Name of the role	STRING	E.g. BatMat_PTF_ViewAccess
003	role_display_name		STRING	E.g. BatMat_PTF_ViewAccess
004	system_role	????	STRING	E.g. F
005	start_date		TIMESTAMP	E.g. 2022-11-28 14:28:15 UTC
006	end_date		TIMESTAMP	E.g. 2022-11-28 14:28:15 UTC
007	current_flag		STRING	E.g. YES

### Logging table in the staging - User Scope Roles

001	role_id	Id of the role	STRING	E.g. 4ffdc2c087ab11eba3b1000096fb74a6
002	obj_per_id	Object permission id	STRING	
002	user_login_id		STRING	E.g. PRIVOAL
003	start_date		TIMESTAMP	E.g. 2022-11-28 14:28:15 UTC
004	end_date		TIMESTAMP	E.g. 2022-11-28 14:28:15 UTC
005	current_flag		STRING	E.g. YES

### Logging table in the staging - Data Set Object

001	obj_per_id		STRING	E.g. 4ffdc2c087ab11eba3b1000096fb74a6
002	obj_name	It's the name of the object we want to give access to, can be a dashboard, can be an excel file, access to a table etc...	STRING	E.g. BatMat_PTF_ViewAccess
003	group_id	Id of the group	STRING	E.g.
004	start_date		TIMESTAMP	E.g. 2022-11-28 14:28:15 UTC
005	end_date		TIMESTAMP	E.g. 2022-11-28 14:28:15 UTC
006	current_flag		STRING	E.g. YES

Logging table in the staging - **Permission**

001				
002				
003				
004				
005				

Logging table in the staging - **Object**

001				
002				
003				
004				
005				

Logging table in the staging - **Object Type**

001				
002				
003				
004				
005				

Logging table in the staging - **Data Set Object values**

001				
002				
003				
004				
005				
006				