User Guide

Privitar Data Security Platform, version 2.0.0

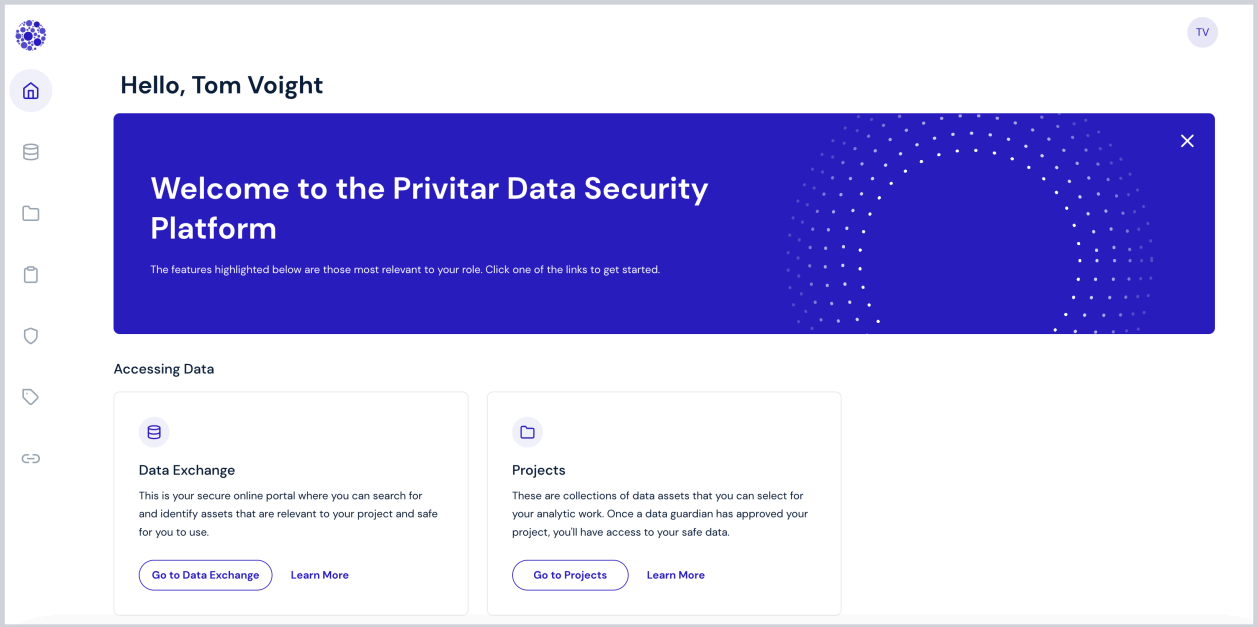
Publication date October 12, 2023

Privitar Data Security Platform, version 2.0.0

Table of Contents

1. Welcome

Welcome to the user guide for the Privitar Data Security Platform. Here you will find explanations of all the concepts specific to the platform and instructions on how to use its features.



1.1. What's the Privitar Data Security Platform?

The Privitar Data Security Platform is a proven data security solution for responsible data analytics. It builds collaborative workflows and policy-based data protections into data operations to enable efficient, effective, and responsible data use.

The platform enables self-service access to data for all users and applications. It streamlines access to data, privacy policy enforcement, and lineage reporting across an organization. Businesses can use their data to innovate and achieve greater efficiency while maintaining customer trust and navigating data privacy, data sovereignty, and industry data protection regulations.

1.2. Who Uses the Privitar Data Security Platform?

The Privitar Data Security Platform serves the following user types:

- Data guardians
- Data owners
- Data consumers

To see which role or roles you have on the platform, click the avatar symbol in the top right corner of any page.

To see which version of the platform you are using, click the avatar symbol in the top right corner of any page, and click **About**.

1.2.1. Data Guardians

Data guardians are users on the Privitar Data Security Platform who develop and maintain company policies and rules that govern data usage, including how the organization adheres to regulatory and compliance guidelines and requirements.

Data guardians are responsible for approving all data requests, including requests to register data on the platform and requests to access data outside the platform.

In your organization, data guardians might have job titles similar to security officer, information officer, or data officer. They might work in legal, risk, or compliance departments.

1.2.2. Data Owners

Data owners are users on the Privitar Data Security Platform who register and classify data on the platform. Data owners understand where the data comes from, its quality, its meaning, and for what purposes it can be used.

In your organization, data owners might have job titles similar to data architect, data analyst, or business intelligence manager.

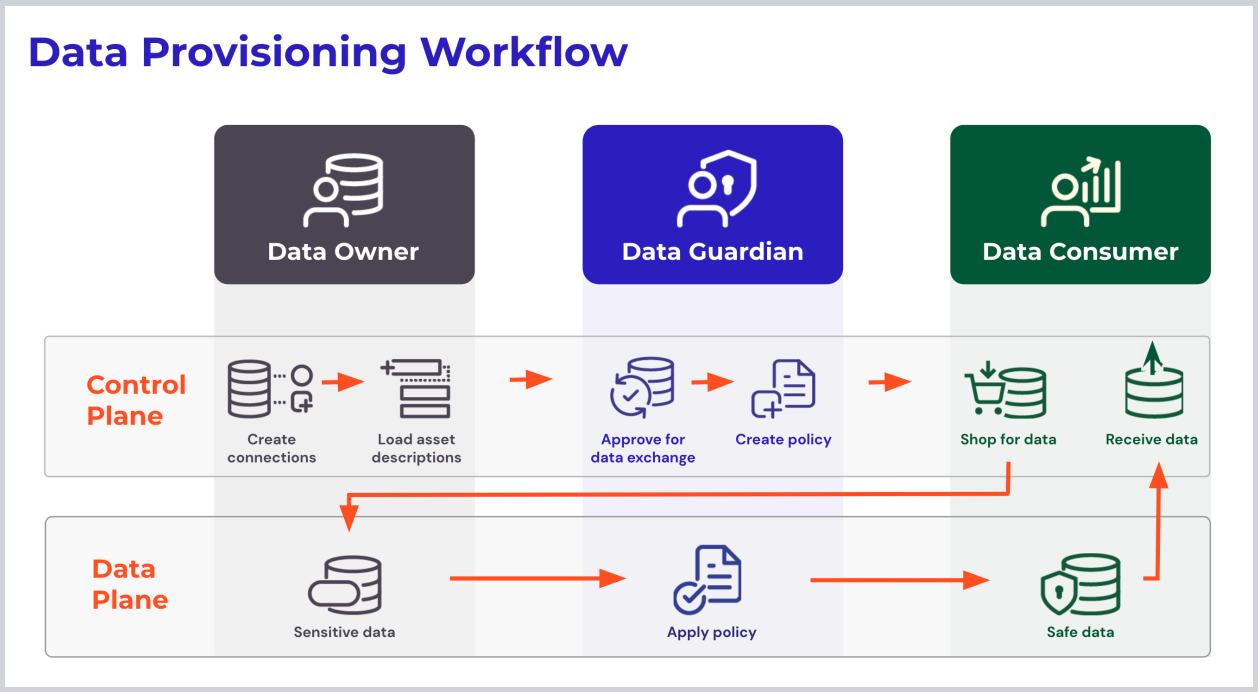
1.2.3. Data Consumers

Data consumers are users on the Privitar Data Security Platform who request and consume data from the platform. Data consumers require direct access to data as part of their job responsibilities.

Data consumers search for data, understand its meaning, its intended business use, and the responsibility that comes with handling that data.

In your organization, data consumers might have job titles similar to financial analyst, marketing analyst, business analyst, data analyst, data engineer, or data scientist.

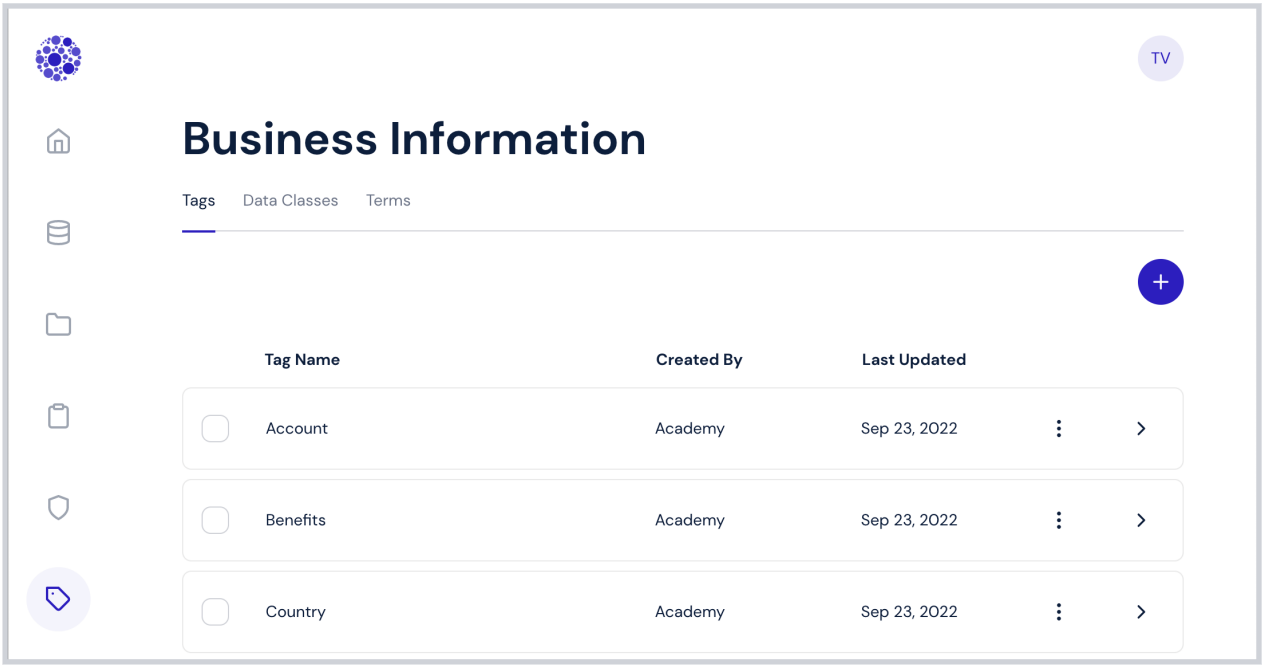
1.3. High-Level Data Provisioning Workflow



Users in your organization will perform the following tasks as they provision data through the Privitar Data Security Platform:

1. Data guardians create business information:
    a. Create tags
    b. Create terms
    c. Create data classes
2. Data guardians create policies, rules, and transformations:
    a. Create policies
    b. Create rules
    c. Create transformations
3. Data owners add data to the data exchange:
    a. Create a dataset
    b. Create a connection to the data source
    c. Add an asset to a dataset
    d. Describe an asset and submit it for approval
4. Data consumers search for and provision data:
    a. Create a project
    b. Search for data
    c. Submit a request to access data
    d. Access the data
5. Data guardians manage request tasks:
    a. View asset registration and project request tasks

2. Business Information



As a data guardian, you create *business information* for others in your organization to use as they add, search for, and access data. In the Privitar Data Security Platform, business information includes:

- Data Classes
- Tags
- Terms
- Purposes

2.1. Privitar's Approach to Organizing Business Information

Here are a some guidelines that may be helpful when creating business information, such as tags, terms, and data classes.

Model only what you wish to manage. While the temptation is to include your entire business glossary, the platform is not meant to replace your data catalog. Instead, it should contain at most only a subset, and only the portion that actually helps you categorize, classify, and manage the data that needs to be provisioned and protected.

Don't use the platform as a modeling design tool. While it may be useful to create some terms and tags while exploring the platform, don't rush into creating tags and terms only to replace all of them when you get new ideas. Instead, attend Privitar's training to understand the concepts, and use the training environment as a means to develop your initial design thoughts.

Consider simplifying and optimizing existing workflows. Avoid having to maintain context in multiple systems, or worse, having to introduce additional new systems or mechanisms, such as separate spreadsheets outside the platform just to conform to your existing workflow.

2.2. Data Classes

A data class is a categorization that data owners apply to fields within data assets to indicate the category of data. Within the Privitar Data Security Platform, data owners can apply a data class to identify the data's category and ensure that that kind of data is classified consistently throughout your organization. For example, data classes can classify birth dates, national identifiers, and postal codes.

As a data guardian, you create the data classes that data owners apply to fields when registering an asset.

2.2.1. Best Practices for Data Classes

The platform allows you to build logical policies, rules, and transformations based on data classes so you don't have to have rules for every single asset and field.

Some key points:

- **Every field does not need a unique data class.** Every field *planned for use* should have an assigned data class. That can be a generic data class. Fields that will never be used for subsequent processing or analysis do not need a data class assigned, but remember that if you do not map a data class to a field, then the default transformation will apply.
- **When data consumers request access to data, they do not know which policies apply.** Within the platform, categorizing fields into data classes helps to manage how the platform applies transformation rules, but this is the job of data owners and data guardians. Data class activity is transparent to data consumers.

  When data owners add assets to a data exchange, they should work with potential data consumers to determine proper assignment of data classes to fields that data consumers are likely to use.
- **Transformations only act on data classes, not on tags or terms.** Data owners should keep in mind as they are preparing assets that data guardians assign transformations only to data classes within a transformation rule.

You can also use data classes to classify similar kinds of data consistently. For example:

- You can use a data class called `Birth Date` to put Customer Birth Date and Employee Birth Date fields into the same data class.
- You can use a data class called `Address` to put Billing Address and Mailing Address fields into the same data class.

Good candidates for classification under a single data class are:

- fields with similar content, such as names and email addresses
- fields with similar format, such as phone numbers and postal codes

However, if such fields have different data types, then they should be in their own data class because data classes are specific to the data type. For example, a numeric phone number can be classified as "Phone Number (Numeric)," but a string-based contact number will need another data class, like "Phone Number (String)." This is important because the platform uses data classes to determine which transformations to apply, and those transformations are data type specific.

Data classes are used within the platform, but you may find a similar concept, such as data categories, is currently used in your organization.

Determine the level of specificity needed when creating data classes and mapping them to physical fields within the platform. If there are multiple physical fields of the same type that contain the same general content, it can be simpler to create a multi-purpose, generic data class, such as “Generic String” that maps to various "description" fields and other text fields. That way, the platform will transform those fields in the same way. This means that the results could be dropped, redacted, or replaced as part of a data request.

To help ensure that the platform can transform data consistently and ensure that a broad array of useful analytic fields are available, your set of generic data classes might include:

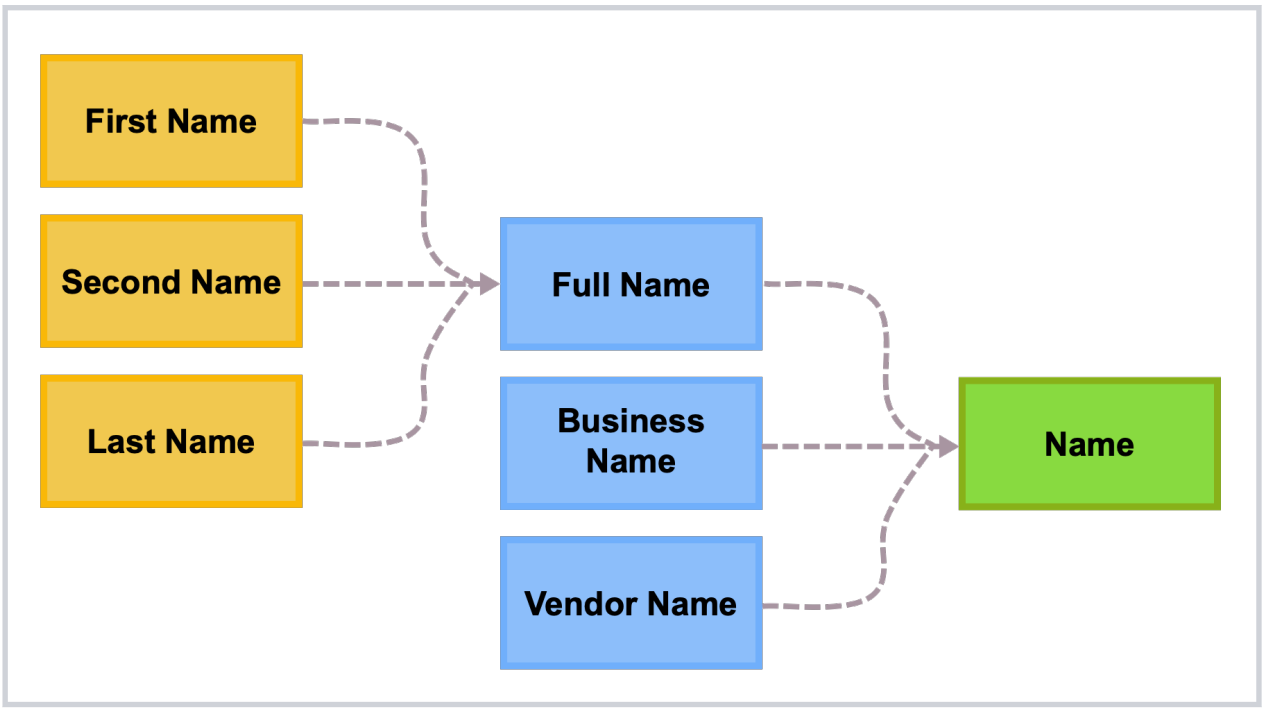
- BLOB (binary large object)
- Boolean [generally use string data type unless the table uses numeric 0/1] (Binary pairs, with only two values, such as T or F, Y or N, and 0 or 1.)
- Code (This might include short-length, alphanumeric fields, which might indicate gender, postal codes, and more. There are a finite number of entries.)
- Date [string data type]
- Date [date data type]
- Identifier [string data type] (Unique alphanumeric fields that identify an object. Examples include Social Security Numbers, credit card numbers, customer numbers, and so on.)
- Identifier [integer data type] (Unique numeric fields that identify an object. Examples include Social Security Numbers, credit card numbers, customer numbers, and so on.)
- Quantity [string data type]
- Quantity [numeric data type]
- Text (This might include names, addresses, product descriptions, comment fields, and more.)

Note

Data discovery and data catalog tools typically provide data classification capabilities. Their data classifications can be brought in (manually or via API) and used as the foundation for data classes in the platform. If they use such data classifications, ensure that you understand how they classify fields, such as quantities and dates. With some tools, both specific data types (for example, integer and date) AND string representations of the same are classified the same. This can create issues, as the platform requires that a data class is represented by a single data type (for example, integer OR string, but not both). In these cases, use the provided data classification (for example, Date_of_Birth) as one specific data class/data type combination and add a second data class with a clearly labeled name (for example, Date_of_Birth_string) to support the other data class/data type combination.

Generalizing Data Classes to Support Broader Policy Definitions

Generalization is the collecting of fields into a more generic group by considering the contents of the fields and summarizing their context. For example, you can generalize "First Name," "Second Name," and "Last Name" into a data class called "Full Name." Or, you can further generalize "Full Name," "Business Name," "Building Name," and "Vendor Name" into a data class called "Name."



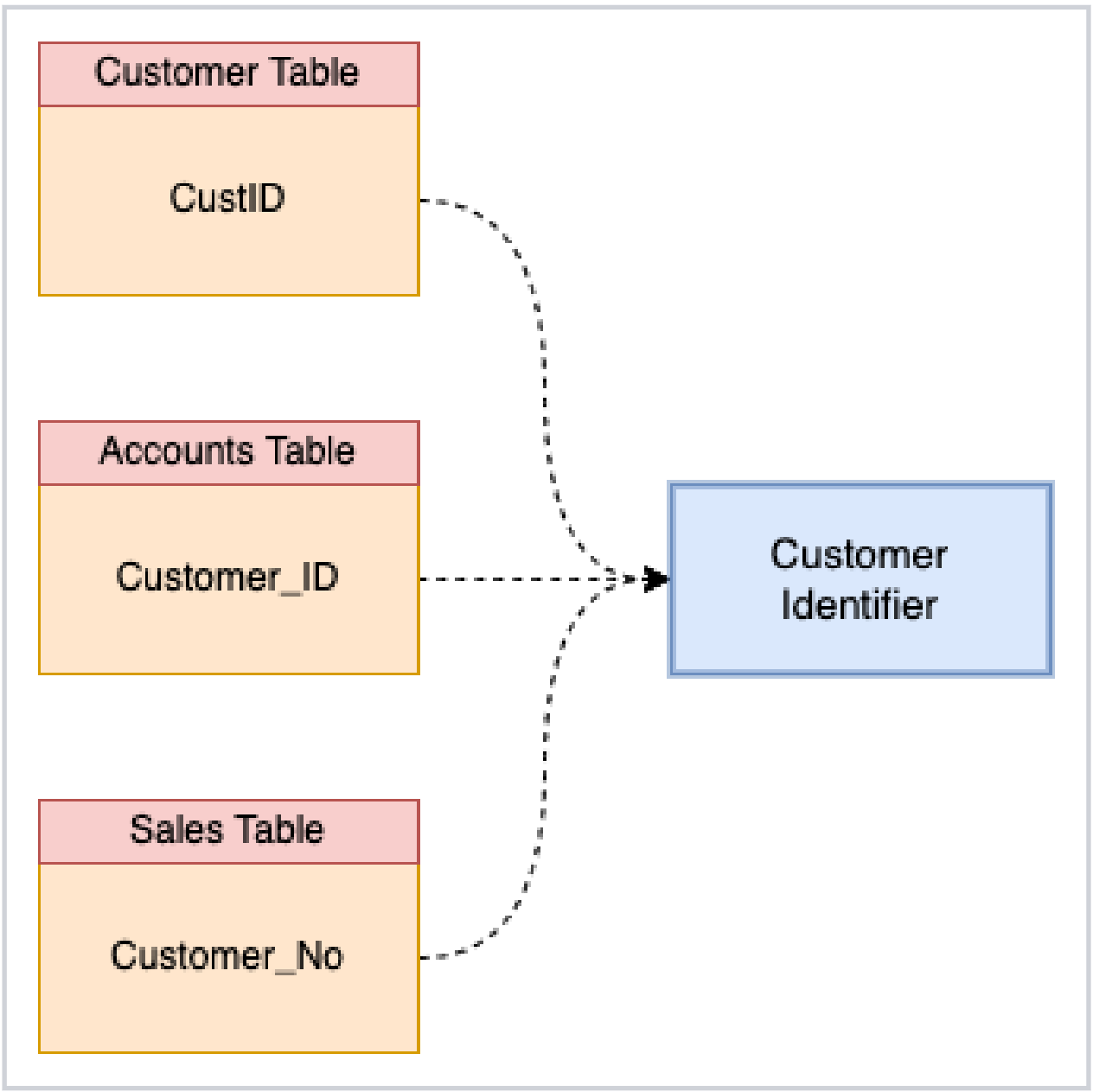
Generalization can help data owners simplify the task of assigning data classes to fields for assets. It also works best when the fields under the generalized data class require the same transformation.

Data guardians can assign tags to fields to apply more fine-grained decisions in policies. This is useful when you need to apply different transformations to the same data class in different contexts.

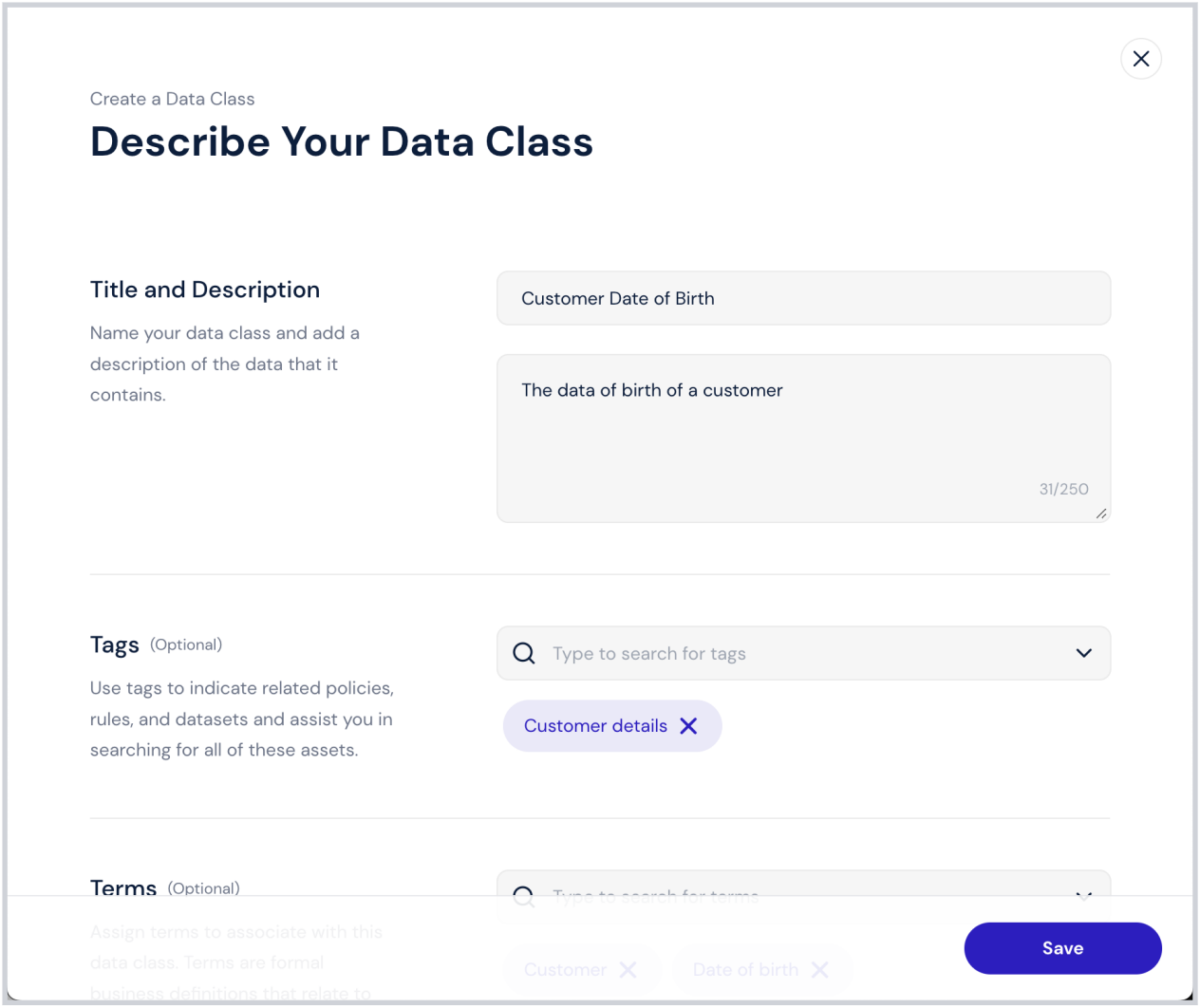
Similarly, it is usually unnecessary to create separate data classes for different datasets. In cases where you wish for consistency in how the platform transforms data, you must have a single data class. In cases where they wish for different transformations or a lack of consistency, you can use rules and project settings to control this.

Using Data Classes to Address Aliasing

When something has an alias, it means it is also known by another name. Data consumers can apply aliasing to fields that are similar but have different names in different tables. For example, fields named "CustID," "Customer_ID," and "Customer_No" might actually represent the same data with the same format. Hence, you can assign them all to a data class called "Customer_Identifier."


Customer Table
CustID
Accounts Table
Customer_ID
Sales Table
Customer_No
Customer Identifier

2.2.2. Create a Data Class



1. Click **Business Information** in the left navigation.
2. Click the **Data Classes** tab.
3. Click the plus sign (+) or click **Create a Data Class.**

   The Describe Your Data Class page appears.
4. **Title and Description—**Enter a name for this data class and describe its meaning.
5. **Tags—**Select tags that apply to this data class.

> **Note**
>
> If you cannot find a tag that matches your criteria, you can create a new tag. See Create a Tag.

6. **Terms—**Select terms to associate with the data class.

   Terms define the business meaning or context for the logical data construct.
7. **Data Types—**Select a data type that describe this data class, such as integer, string, decimal, and so on.

A data type is the data's categorization that is read from the source. Examples include: integer and string. The data type references how data is stored in a database, and each data type can have a different corresponding transformation. For example, you can store a person's age as an integer or a string.

8. Click **Save.**

2.3. Tags

A tag is a keyword that you can define to describe objects, such as when you want to group objects together or add context to those objects. For example, you might want to define tags that correspond to geography, line of business, project names, or applications. Tags help facilitate search and filtering.

For example, you might create a tag for each of your organization's office locations, such as "New York," "London," and "Hong Kong."

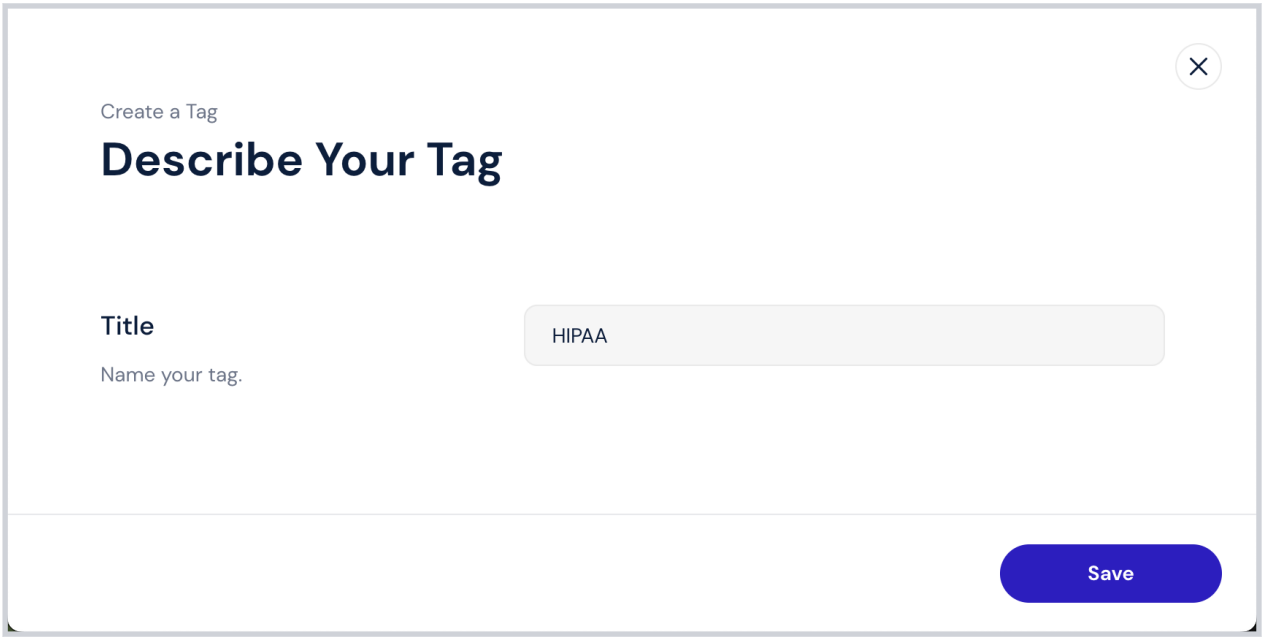
As a data guardian, you will have the opportunity to apply tags when you create policies, rules, and transformations. As a data owner, you apply tags when you create datasets and register assets.

2.3.1. Best Practices for Tags

You can use tags in the following ways:

- **Grouping—**You can assign tags to entities that belong to the same logical group. For example, you can assign a tag like "PII" to fields like "Mailing Address," "Last Name," or "Social Security Number." Or you can assign tags like "Finance" and "HR" to assets owned by those lines of business.
- **Searching—**You can assign tags to entities so that when you search for the tag, these entities will appear in the search results. For example, you can use a tag like "PII" to tag fields that are personal identifiers. Note that tags added for grouping also work well for searching.
- **Meaning—**You can assign tags to entities that have similar classifications or labels. For example, if you classify data according to data sensitivity, you might create tags like "Sensitivity - RESTRICTED," "Sensitivity - CONFIDENTIAL," or "Sensitivity - NONE," and assign them to relevant assets.
- **Context—**You can assign tags to fields to apply more specific decisions in policies. This is useful when you need to apply different transformations to the same data class in different contexts. However, because you should apply tags to the broadest (or most diverse) set of information (such as PII), they require careful consideration when used as policy triggers.

2.3.2. Create a Tag



1. Click **Business Information** in the left navigation.
2. Click the **Tags** tab.
3. Click the plus sign (+) or click **Create a Tag.**

   The Describe Your Tag page appears.
4. **Title—**Enter the text of the tag.
5. Click **Save.**

2.4. Terms

Terms are words used within your organization to describe business concepts in plain language. Adding them to the platform ensures consistent use of those words throughout your organization. Terms also lend meaning to physical assets and their fields and give them context. When data consumers are browsing assets, terms allow them to understand the business meaning and semantics of the physical asset. Examples of terms could be “account type,” “customer level,” or “credit risk rating.”

As a data guardian, you create terms for others in your organization to use as they add and search for data.

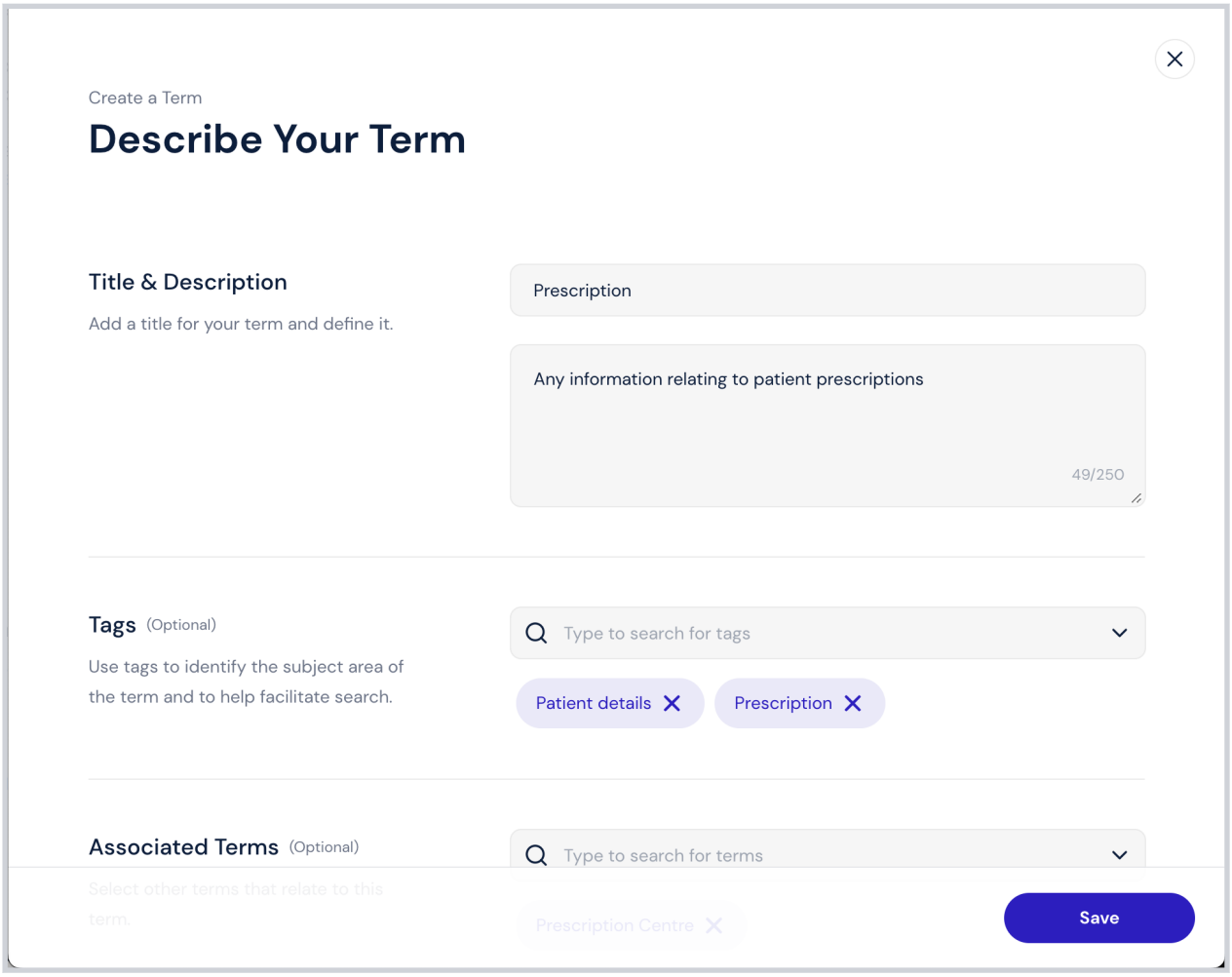
2.4.1. Best Practices for Terms

You do not need to model every business concept in your organization as a term. It's more useful to identify only those concepts that help the data provisioning process or those that give guidance on how the data protection applies to the data assets.

Note

We do not recommend that you import your whole existing business glossary to establish a set of useful terms. Filter the set that you want to work with based on relevant classifications or tags established in the business glossary. For example, you might want to only include terms that represent confidential, sensitive, or PII data.

2.4.2. Create a Term



1. Click **Business Information** in the left navigation.
2. Click the **Terms** tab.
3. Click the plus sign (+) or click **Create a Term**.

   The Describe Your Term page appears.
4. **Title—**Enter a title for the term.
5. **Description—**Explain the term's purpose.
6. **Tags—**Select tags to associate with the term.

   Use tags to identify the subject area of the term and to help facilitate search.

> **Note**
>
> If you cannot find a tag that matches your criteria, you can create a new tag. See Create a Tag.

7. **Associated Terms—**Select other terms to associate with this term.
8. **Data Classes—**Select data classes to associate with the term.

> **Note**
>
> If you cannot find a data class, you can create it. See Create a Data Class.

9. Click **Save**.

   A confirmation message appears.

2.5. Business Information FAQ

The following are frequently asked questions regarding the creation of tags, terms, and data classes:

Should we convert our entire business glossary or data catalog into tags, terms, and data classes?

If your organization has a business glossary or data catalog, then we recommend that you only import privacy-relevant information from it. We do not recommend bringing all content into the platform.

Should we assign the codes used with our existing business taxonomy to tags, terms, and data classes?

In order to align the use of these entities across the organization in a consistent manner, and also to relate to similar entities defined as part of data governance, you might have already defined a comprehensive taxonomy of these entities and assigned each entity a unique code to identify them in your organization. An example code might be DS993AI - Data Science and AI.

In the platform, such a code can be applied as a tag on a business entity to allow users to search by the code. For example, you can create and apply a tag like `DS993AI - Data Science and AI` to entities.

Should we apply tags and terms at the top level (datasets) or the lower levels (assets and fields)?

You can apply tags to most objects on the platform, including datasets, assets, fields, projects, and more. You can apply terms to data classes, assets, and fields. When deciding at which level to apply tags and terms, consider how your organization will use them within the platform.

For example, if you need to control access to data by line of business, such as HR or Finance, then apply line-of-business tags to datasets and projects. This allows you to form policies that match concepts between projects and datasets at a high level. It also allows you to more easily manage policy triggers.

Another example is if users in your organization have a need to search for related fields across assets and datasets. If so, apply tags at the field level. Finally, if you need to express rule conditions down to the field level, you should also apply tags and terms to fields.

Should we model hierarchy with tags or terms?

In general, be prudent when venturing into deeper levels. Consider applying tags and terms only at the highest level in which they are required, and proceed one level downwards only if tags and terms applied at the current level are unable to satisfy the rule conditions required. On the platform, assets and fields do not inherit the tags and terms associated with their datasets.

However, certain business concepts might require that you implicitly apply tags and terms at a lower level to correspond with those at a higher level. In this case, if you apply tags to datasets, then you should also apply these tags to assets and fields since they are part of the same dataset. Similarly, if you apply a term to assets, you should apply the same term to the fields in those assets.

How should we represent hierarchical business concepts?

You might have certain business concepts that are hierarchical. For example, their definition might include having three levels: Level 1, Level 2, or Level 3. When working in the platform, you will want to flatten these out. For example, a business concept for Customer Account Number, might include:

- L1: Personal Data
- L2: Customer
- L3: Customer Account Number

You can represent this in a single tag on the platform as `Personal Data  Customer  Customer Account Number`. Splitting them into three tags might not work well in all cases, and you will need to make sure entities are not stuck with orphaned tags. For example, an L2 tag without an L1 tag.

Ask yourself:

- Does creating or expressing this hierarchy help to make search or grouping of entities easier?
- Does it help you create better conditions for filtering data or applying transformations?
- Do you really need to represent the expanded subtree for each entity in the searching, grouping, or creation of conditions in policies?

3. Policies, Rules, and Transformations

3.1. Privitar's Approach to Building Data Protection Policies

The Privitar Data Security Platform offers a system of policies and rules that can meet all sorts of requests for data consumption. This system provides a scalable solution that allows for automated data requests. Rather than creating specific policies for each dataset, you create reusable policies based on business context. These policies execute automatically on an unlimited number of data requests.

Let's start by looking at the result you want to achieve and see how the platform helps you achieve it.

Your goal is to de-identify data while still maintaining that data's usefulness. This is where the platform's transformations come in.

A transformation defines a set of behaviors (privacy enhancing technologies) for the platform to execute on a field in a dataset to de-identify it, while still preserving data utility.

However, you don't want the platform to apply the same transformations to every field in every dataset regardless of the user, the type of data, and the purpose. You want these transformations to be conditional on the metadata context. You use rules to ensure that the platform applies a transformation only when you want.

Rules are building blocks of policies. Rules are conditional based on attributes, such as user groups, terms, tags, locations, and so on. Rules also take actions specific to data classes and transformations.

While rules apply to specific conditions, you might need multiple rules to meet the needs of a broader use case. You need to group this set of rules into a policy that defines the use case for which the platform should apply them.

Note

Policies, rules, and transformations are constrained within the context of a data exchange, so if you wish to use them in multiple data exchanges, you will need to recreate them in each data exchange.

3.2. Policies

A policy is a set of rules that serves a specific data provisioning use case. A policy is a flexible construct that allows you to apply rules in the way that best meets your use cases. For example, you can write a policy around a specific regulation (such as HIPAA or GDPR) or around a specific business use case (such as provisioning data for marketing analytics).

The platform offers two types of policies—access control policies and transformation policies.

As a data guardian, you use access control policies to control access to assets, and you use transformation policies to define a set of transformations to apply to those assets to de-identify them for a given use case. You can reuse policies across different environments or for multiple data releases to ensure consistent treatment for the same asset.

The platform processes policies from top to bottom, and within each policy, the platform processes transformation rules from top to bottom. If multiple transformation rules match the set of conditions for a given request, the platform executes the first rule in order and prevents any other rules from executing on the same field.

3.2.1. Best Practices for Policies

The platform consistently applies policies to all assets and projects across data exchanges. Where data exchanges are kept physically distinct, you can export policies to the different environments to ensure consistent treatment for data assets.

Ignore Order with Access Control Policies

Record-level access controls (RLACs) use a compounding approach to rule execution. This means that the platform will evaluate all rules. There cannot be any conflict between rules since any rule that applies will allow or deny access. This differs from transformation rules, in which the platform only evaluates rules until it executes the first applicable transformation in the list of all transformations for a given data class, while it ignores the rest of the transformations for the same class.

Since multiple access control policies can apply to the same record, the policies do not need to be in any specific order.

Start with Absolute Transformation Policies

When the Privitar Data Security Platform evaluates transformation policies, it does so in priority order from top to bottom. Due to this top-to-bottom processing of policies, data guardians should place absolute policies at the top. These are policies that govern data conditions that you always want enforced.

For example, you might want to grant auditors access to all data, regardless of conditions. In this case, your top policy should contain a transformation rule with a conditional trigger that determines whether the user group is Auditors. If true, the rule is active and the policy applies for all relevant data. If false, the platform skips that policy.

Your next absolute policy might be for data transfer controls for those countries where your organization can never transfer data. In this case, your policy would have no conditional trigger. It would always be active and enforced.

3.2.2. Examples of Policy Models

The following are example templates that you might use to craft access control policies and transformation policies.

Case One

Consider the following access control policy:

Rule	Rule Condition	Action (Fixed)
1	If A is true	Deny access where data class X = x
2	If A is true and B is true	Deny access where data class Y = y
3	If C is true	Deny access where data class X = z

1. If A is true, the user will be denied access to fields mapped to data class X with values x (due to Rule 1).
2. If A and B are both true, the user will be denied access to fields mapped to data class X with values x (due to Rule 1) and data class Y with values y (due to Rule 2).
3. If A and C are both true, the user will be denied access to fields mapped to data class X with values x (due to Rule 1) and values z (due to Rule 3).
4. If A, B, and C are false, the user can access all rows since no rows were caught by any of the above rules.

Case 2

Consider the following transformation policy:

Rule	Rule Condition	Action
1	If D is true.	Transform data class P Retain data class Q
2	If E is true and F is true.	Drop data class P Transform data class Q Retain data class R
3	If E is true.	Retain data class P Drop data class Q Transform data class R
4	If D is true.	Retain data class P Transform data class S

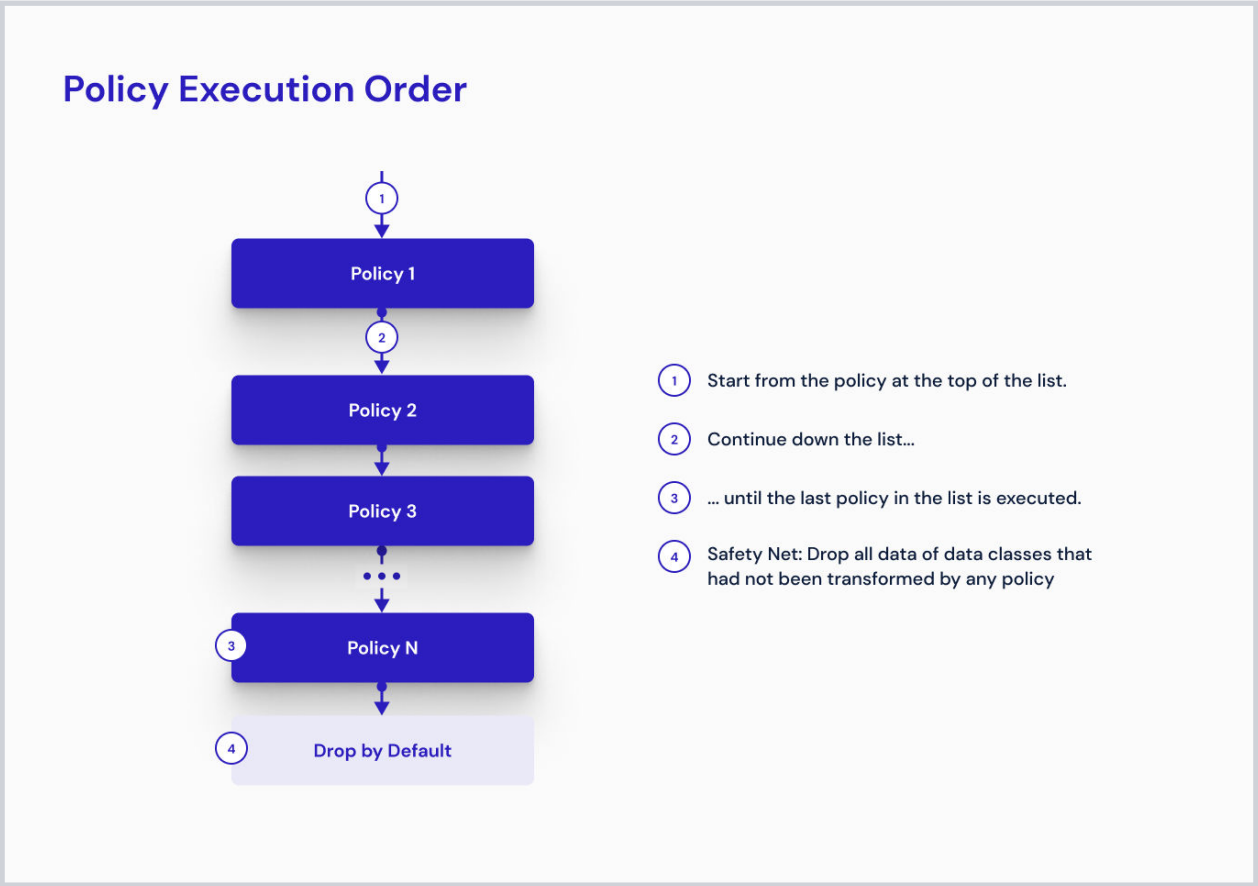
If only D and E are true, then data class P will be transformed (due to Rule 1), data class Q will be retained (due to Rule 1), data class R will be transformed (due to Rule 3), and data class S will be transformed (due to Rule 4). Note that Rules 1 and 4 have the same conditions.

This also means that you only know the final set of fields that a user can access across the data exchange, only after the platform has evaluated all the rules.

3.2.3. Data Privacy by Design

If, after executing all transformation policies and rules, a given data class has not been processed by any rule, the default transformation is to drop the data. The purpose of this is to ensure safe data by design. However, you can change the default transformation.

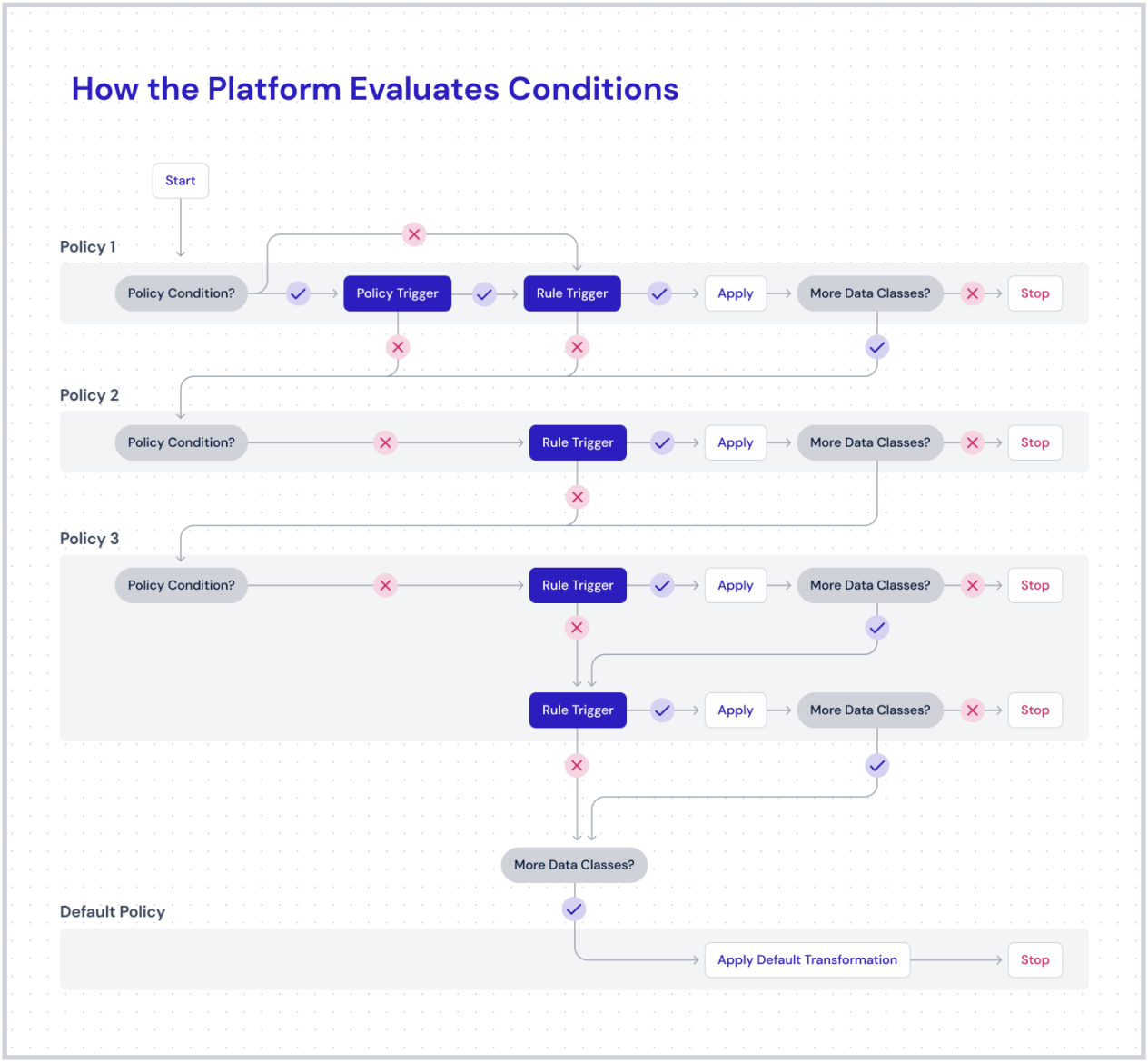
Dropping the data by default enables data guardians to build the system of transformation policies incrementally, without risking data leakage as a result of an incomplete policy system. Also, if a data guardian was to forget to write a rule for a class, the default transformation to drop that data would ensure that the raw data is not accidentally exposed.



3.2.4. How the Platform Evaluates Conditions

When the Privitar Data Security Platform evaluates a transformation policy, it first checks whether the policy has a conditional trigger. If the policy has a trigger, the platform evaluates that trigger first. If the policy trigger applies in that context, the platform then evaluates any rule conditions within that policy.

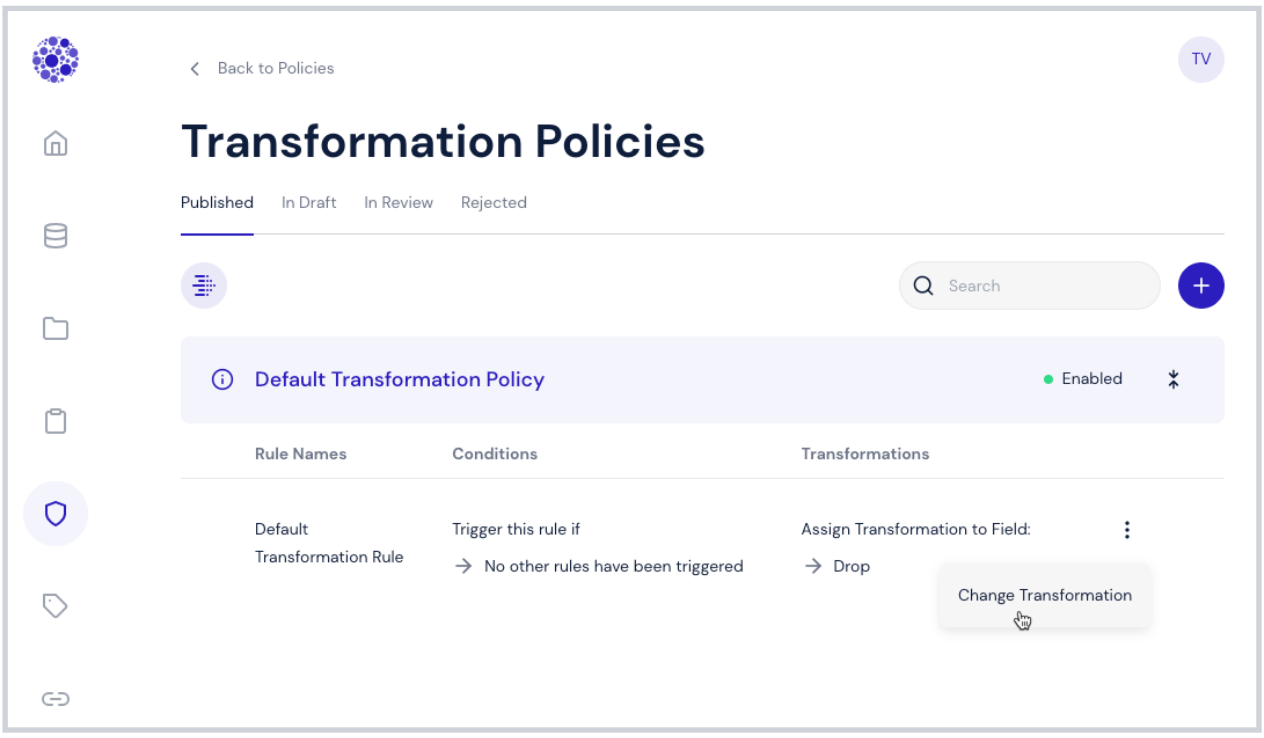
If the transformation policy does not have a conditional trigger, the platform evaluates the rule conditions within that policy. If the policy trigger does not apply in that context, the platform does not evaluate any rule conditions within that policy.



3.2.5. Set the Default Transformation Policy

By default, the platform uses the drop field transformation type when no other rules apply to prevent accidentally sharing data. However, as a data guardian, you can change the default transformation policy to the redact with NULL transformation type, which allows users to see the column header, and return a NULL value within a data request.

The redact with NULL transformation type is useful when you want to remove field values while preserving the schema of the asset after the data transformation. Maintaining the original schema is often critical to ensure correct processing by data integration, data replication, or data virtualization tools. Maintaining the original schema might be required to support software development and testing and when working with some third-party applications.



1. Click **Policies** in the left navigation.
2. Click **Transformation Policies**.
3. Next to "Default Transformation Policy," click the expand icon to display the details.
4. Click **More** (the three vertical dots).
5. Click **Change Transformation**.

   The Change Default Transformation window appears.
6. Select the transformation type.

   Your request to change the default transformation type goes to another data guardian for approval.

Note

The platform uses the default transformation:

- On any columns identified in a data asset for which a data guardian has not associated any transformation policies.
- On any "unknown" columns, that is columns that exist in the data source but are not part of the data asset.

3.2.6. Create a Policy

1. Click **Policies** in the left navigation.
2. Click **Access Control Policies** or **Transformation Policies**.
3. Click the plus sign (**+**).

   The Describe Your Policy page appears.

4. **Title—**Enter a title for the policy.
5. **Description—**Explain the policy's purpose.
6. **Tags—**Select tags to assign to the policy.

   Use tags to indicate related policies, rules, and datasets and assist you in searching for all of these assets. For example, you might wish to tag all of these assets with the name of the data privacy rule with which they help your organization comply, such as GDPR or HIPAA.

   > **Note**
   >
   > If you cannot find a tag that matches your criteria, you can create a new tag. See Create a Tag.

7. Click **Save** to return to the list of policies.

   Or click **Edit Policy** to add rules or a conditional trigger (for transformation policies).

You can now add rules to this policy. See Create a Transformation Rule or Create an Access Control Rule.

This policy will stay in **In Draft** status until you submit it for approval. See Submit a Policy for Approval.

3.2.7. Add a Trigger to a Transformation Policy

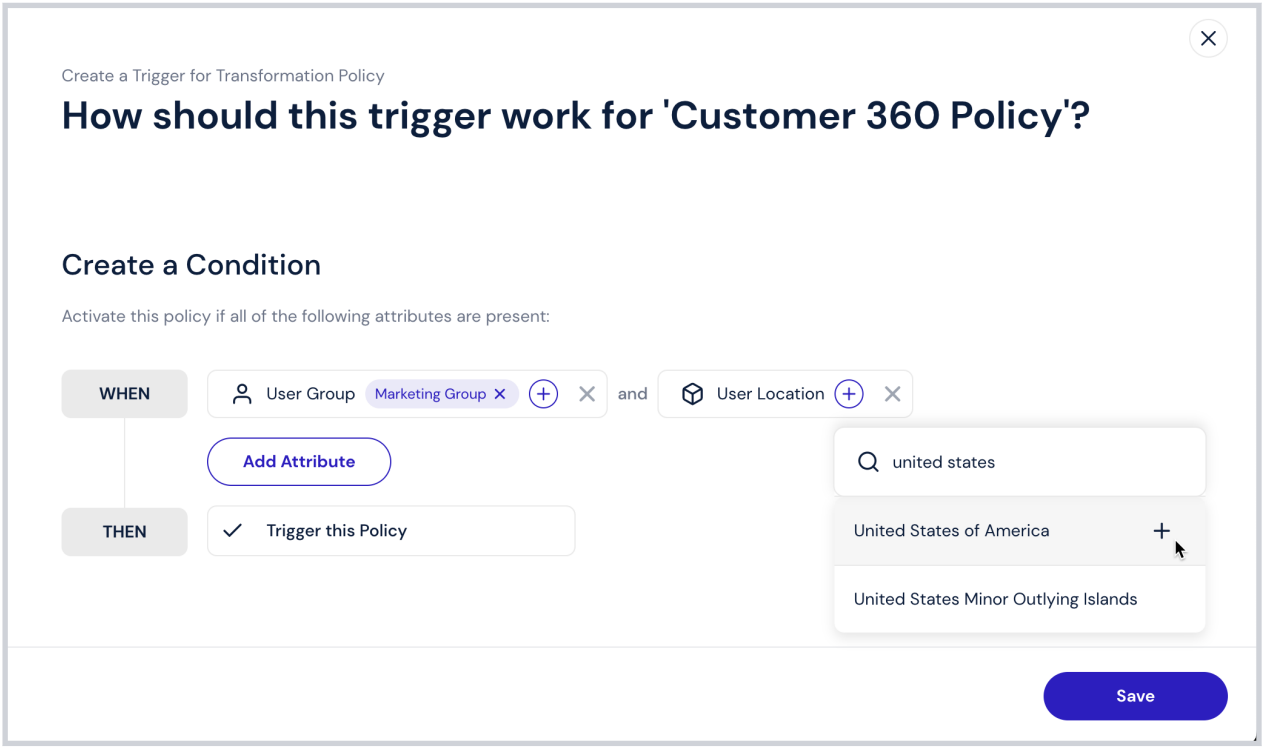
As a data guardian you can add a trigger to a transformation policy so that it only applies under conditions specified by you. If you do not add a trigger, the policy always applies. See How the Platform Evaluates Conditions to learn more.

1. Follow the steps in Create a Policy.
2. Click **Back to Policies** to return to the Policies page.
3. Click the name of the policy that you just created.
4. Click the **Triggers** tab.
5. Click **Create Policy Trigger**.

   The Create a Trigger page appears.
6. Click **Add Attribute**.

   The Add Attribute window appears.
7. Select one or more attributes.
8. Select an operator, such as `Is any of` or `Is none of`.
9. Click the plus sign (+) next to each attribute to add one or more values for those attributes.
10. Click **Save**.

    A confirmation message appears.



3.2.8. Edit and Enable a Policy

As a data guardian, you can edit a policy.

1. Click **Policies** in the left navigation.
2. Click **Access Control Policies** or **Transformation Policies**.
3. Click the title of the policy that you want to edit or enable.
4. Click **More** (the three vertical dots).
5. Select **Edit Policy**.
6. Make changes to the policy details, triggers, or rules, as described in Create a Policy.
7. Make changes to the state of the policy, such as enabling or disabling it if it is in **Draft** status. You cannot change the state of a policy in **Live** status.
8. To exit edit mode, click the banner at the top of the page that reads, "Click to exit edit mode," or click **More** (the three vertical dots) and select **Exit Edit Mode**.
9. Click **Submit Changes**.

   A confirmation page appears.
10. Check the box to confirm that you understand and would like to proceed.
11. Click **OK**.

    The policy moves into **In Review** status. You can now submit it for approval. See Submit a Policy for Approval.

3.2.9. Submit a Policy for Approval

As a data guardian, once you have created a policy, you can submit it for approval.

1. Click the **Drafts** tab on the Policies page.

A list of policies not yet submitted for approval appears.

2. Click the title of the policy that you want to submit for approval.

   A preview of the policy appears.

3. Click **Submit for Approval**.
4. Click **OK**.

   The policy remains in "In Review" status until a data guardian approves it.

5. To see the status of your request, click **Tasks** in the left navigation.
6. Click the **My Requests** tab.

   Policies can have the following statuses:

   - **In Draft—**A data guardian has not yet submitted the policy for approval.
   - **In Review—**A data guardian has submitted the policy for approval.
   - **Rejected—**A data guardian or platform administrator has rejected the policy.
   - **Published—**A data guardian or platform administrator has approved the policy.

Once a data guardian or platform administrator approves a policy, it appears on the **Published** tab.

If a data guardian declines a policy, it appears on the **Rejected** tab.

Important

Only the data guardian who submitted the policy for approval may view and act on a declined policy that is on the **Rejected** tab, including editing it or deleting the request.

To learn more about this review process, see Approve Policy Tasks.

3.3. Rules

Rules are building blocks of policies. Rules are conditional based on attributes, such as user groups, terms, tags, locations, and so on. Rules also take actions specific to data classes and transformations.

3.3.1. Best Practices for Rules

Keep in mind that data owners assign a single data class to multiple fields, which may be from different assets within the same exchange. In this case, the platform will process all fields assigned with the same data class according to the transformation specified in the rule, if the rule applies under that context.

For this reason, when designing rules, avoid thinking in terms of which tables the user can access. Instead think about which fields (data classes) the user can access across all tables.

The platform evaluates transformation policies and rules, it does so in priority order from top to bottom. Due to this top-to-bottom processing of policies and rules, data guardians should place absolute policies at the top.

When a transformation rule is satisfied for a given data class, the noted transformation is the transformation that the platform will apply. If the same data class appears in a subsequent transformation rule or policy, the platform will ignore it. This top-to-bottom processing continues until either all data classes have an assigned transformation, or the platform has processed all transformation policies and rules, at which point the platform will apply the default transformation policy (either drop field or redact with NULL) to any unassigned data classes.

3.3.2. Rules Set the Context

As a data guardian, you define rule conditions by using metadata, such as user groups, data classes, tags, locations, and purposes, to explain the context in which the data is provisioned. It's the who, what, where, and why of data provisioning.

Rules support the following metadata:

- Who
  - user group
- What
  - asset tag
  - asset term
  - dataset tag
  - field tag
  - field term
  - project tag
- Where
  - source location
- Why
  - project purpose

The platform maps user identities to rules through user groups (from integration through LDAP or an internal registry) and user locations.

3.3.3. How Record-Level Access Controls Work

Record-level access controls (RLACs) are rules that act as access filters. They determine who can access specific data records, unlike transformation rules, which determine how users access data.

Depending on how your exchange administrator set the default, RLACs either use a "deny list" or "allow list" approach to access. The default is that all users have access, and RLAC rules filter out access by exception. Alternatively, the default is an "allow list" approach in which dropping data is the default.

By either allowing or denying access by default:

- Data owners do not need to know who will need access to the data that they register.
- Data guardians can allow or deny access to records in an asset on a case-by-case basis.
- Data consumers do not have to request access to every single record.

3.3.4. How the Platform Executes Access Control Rules

Record-level access controls (RLACs) use a compounding approach to rule execution. This means that the platform will evaluate all rules. There cannot be any conflict between rules since any rule that applies will allow or deny access. This differs from transformation rules, in which the platform only evaluates rules until it executes the first applicable transformation in the list of all transformations for a given data class, while it ignores the rest of the transformations for the same class.

Since multiple access control policies can apply to the same record, the policies do not need to be in any specific order.

Access control policies execute before transformation policies. For example, consider a source database table with columns and rows. If there are applicable RLAC policies available for this table, then these apply before any transformation policies apply to any of the columns. Thus, the rows that the user does not have access to are never fetched from the underlying database. The transformation policies are only applied to the rows that are still available to the user.

Note

It is also possible to have access control policies with filters that are based on columns that may be dropped later.

3.3.5. How the Platform Executes Transformation Rules

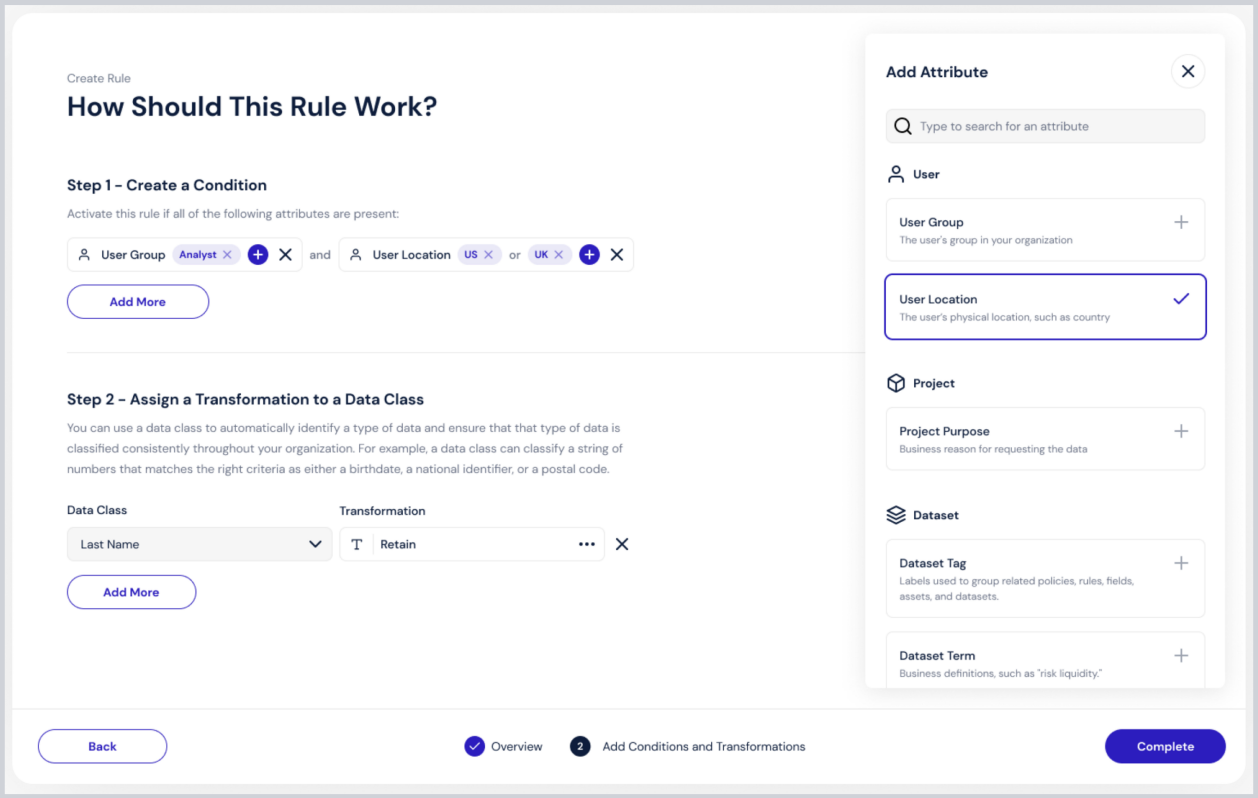
When the Privitar Data Security Platform goes through each transformation rule in a policy individually, in priority order, it first checks whether the criteria match the metadata, such as data classes and tags in the query.

The platform employs user groups to control the access to resources and select the appropriate transformations. Users are represented in the rules through the user groups to which they belong.

- If the metadata attributes match those in the rule, the platform then performs the transformation that is associated with that rule, and the process completes.
- If the metadata attributes do not match those in the rule, the platform then moves to the next rule in that policy. This process continues for each rule from top to bottom until the attributes match those in the rule.
- If the metadata attributes do not match those in any of the rules, the platform performs the default transformation, and the process completes.

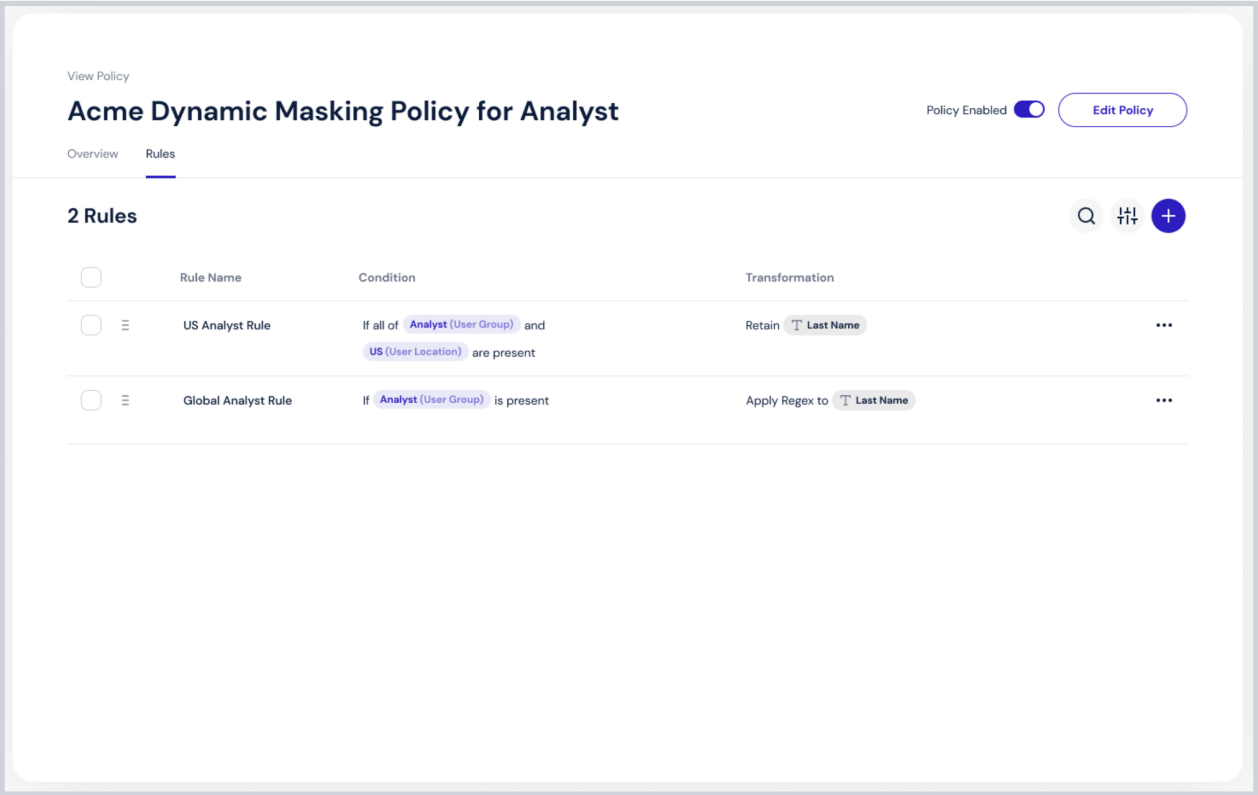
> **Tip**
>
> Due to this ordered, top-to-bottom processing of transformation rules, we recommend that you place your most specific transformation rule (the one that has the most conditions) first, your second-most specific rule second, and so on, until your most generic rule (one that has only one condition) is last.

When a transformation rule contains multiple metadata attributes, the platform treats this as an `and` condition. That is, the attributes must match all of the rule's attributes in order for the platform to apply the associated transformations of the rule. For example, if a transformation rule has a `user group` attribute and a `user location` attribute, the user's attributes must match both in order for the platform to apply the associated rule. For example:



In this example, the user must both be an analyst and be in the US or UK to access the last name data under this rule.

However, to prevent an "all or nothing" approach to data access, it is also possible to condition a user's access based on partial metadata attribute matches. So, rather than dropping a column (if that is your default transformation) you could instead apply a transformation so that the user has access to tokenized data rather than no data. For example:



In this example, the user must both be an analyst and be in the US in order for the rule to retain the last name data.

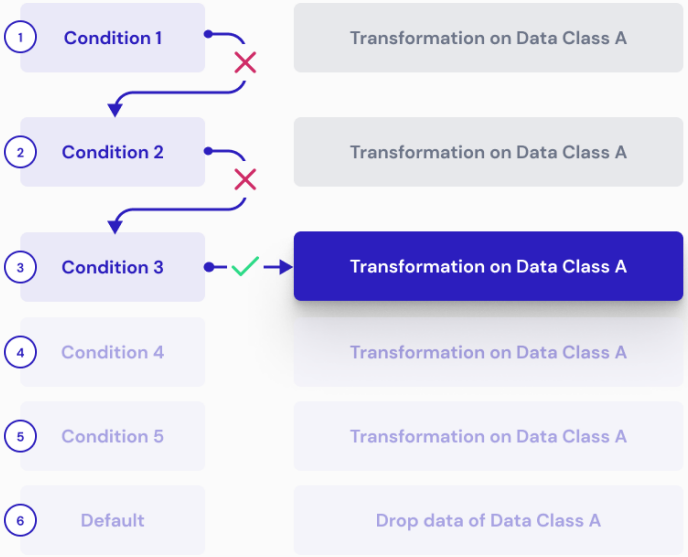
If instead, the user is an analyst, but is not in the US, the second rule would match, and it would therefore apply the regular expression (regex) tokenization to the last names, giving the user access to that tokenized data.

Transformation Rules Execution Order Examples

The platform processes policies from top to bottom, and within each policy, the platform processes transformation rules from top to bottom. If multiple transformation rules match the set of conditions for a given request, the platform executes the first rule in order and prevents any other rules from executing on the same field.

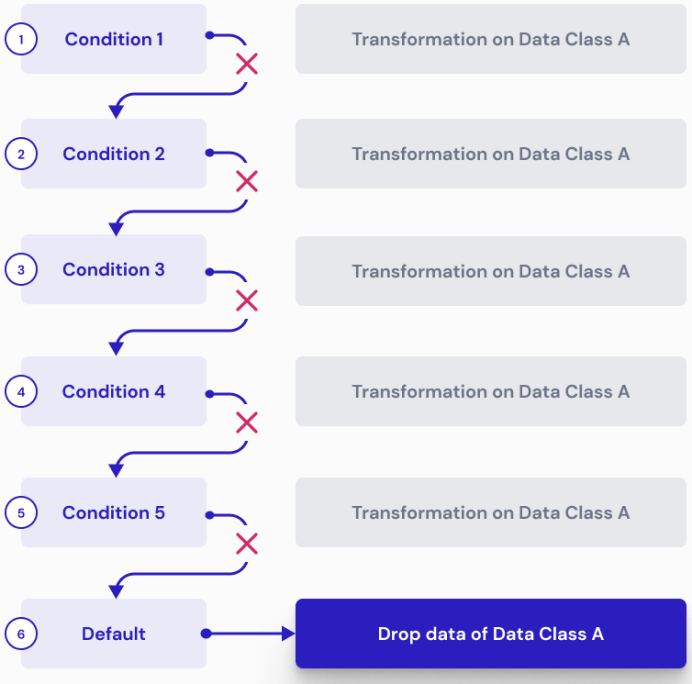
The following examples illustrate this ordered approach.

Rule Execution Order



1. Condition 1 is False; go to next rule for Class A
2. Condition 2 is False; go to next rule for Class A
3. Condition 3 is True; transform all data of Class A
4. Skip
5. Skip
6. Skip the default action

No Matching Rule in Any Policy



1. Condition 1 is False; go to next rule for Class A
2. Condition 2 is False; go to next rule for Class A
3. Condition 3 is False; go to next rule for Class A
4. Condition 4 is False; go to next rule for Class A
5. Condition 5 is False; go to next rule for Class A
6. Default rule triggers; DROP all data of Class A

3.3.6. Create an Access Control Rule

1. Select an access control policy.

   If you have not yet created one, see Create a Policy.
2. Click the **Rules** tab.
3. Click **Add Rule** if the policy is already in *In Draft* status.

   Otherwise click **More** (the three vertical dots) and select **Edit Policy**, and then click **Update Policy** to move it to *In Draft* status. Then click **Add Rule**.
4. Click the plus sign (**+**).

   The Describe Your Rule page appears.
5. **Title—**Enter a title for the rule.
6. **Description—**Explain the rule's purpose.
7. **Tags—**Select tags to assign to the rule.

   Use tags to indicate related policies, rules, and datasets and assist you in searching for all of these assets. For example, you might wish to tag all of these assets with the name of the data privacy rule with which they help your organization comply, such as GDPR or HIPAA.

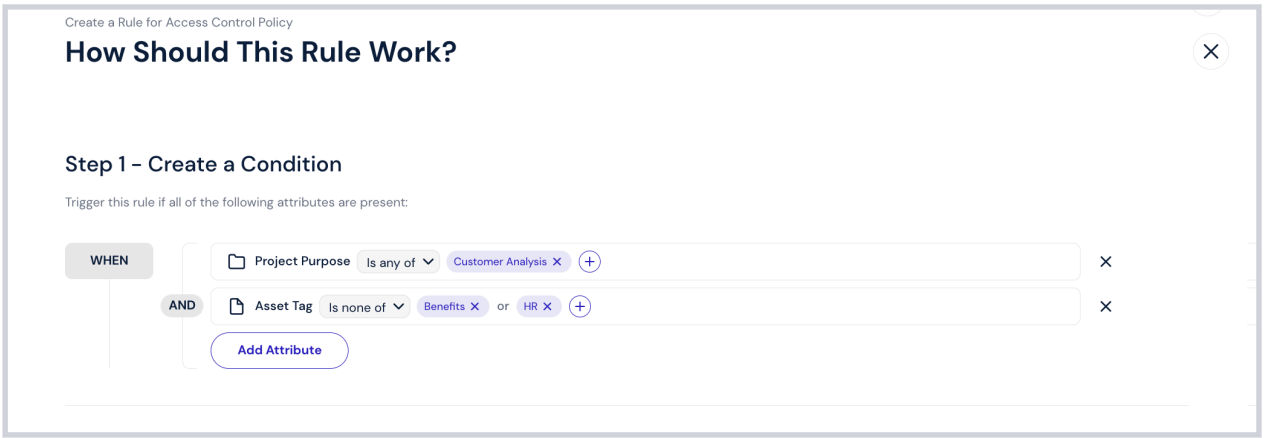
   > **Note**
   >
   > If you cannot find a tag that matches your criteria, you can create a new tag. See Create a Tag.

8. Click **Next**.

   The How Should This Rule Work? page appears. On this page, you specify under which conditions the platform will deny access to data records (such as table rows).

   > **Note**
   >
   > To learn more about how access controls function, see How Record-Level Access Controls Work.

Create Conditions



1. In the Create a Condition section, click **Add Attribute**.

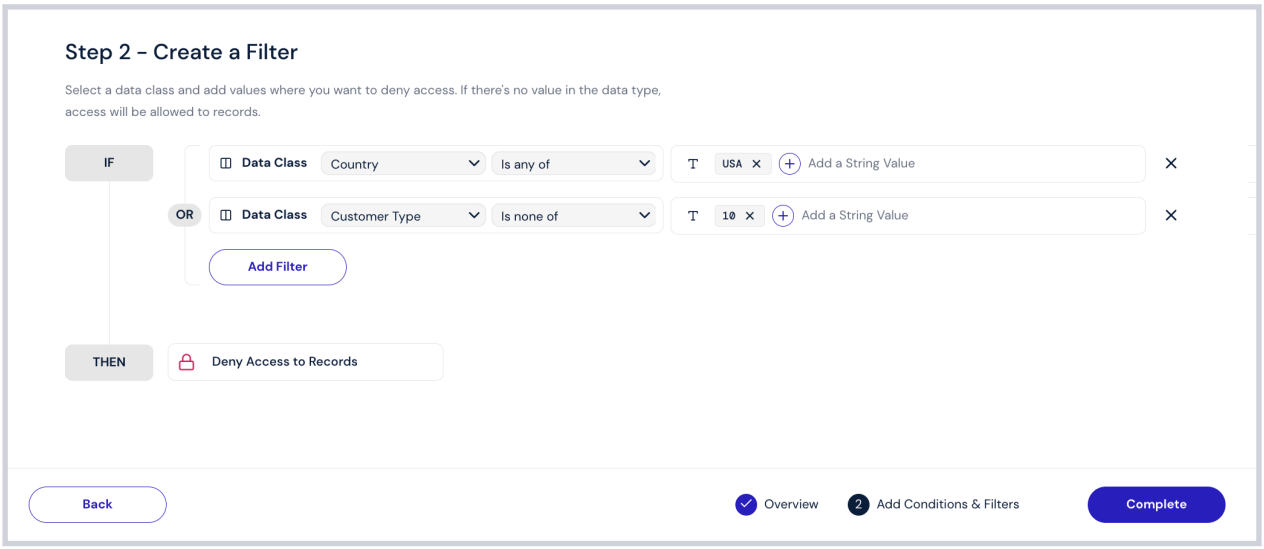
   The Add Attribute window appears.
2. Select all of the attributes that must be present for this rule to activate. These include tags, terms, purposes, user groups, and more.

   Each attribute appears.
3. Select an operator, such as `Is any of` or `Is none of`.
4. Click the plus sign (**+**) next to each attribute to select values for each.

> **Note**
>
> If you select **User Location**, you can select from a list of countries.

5. Click **Add Attribute** again to create another condition.
6. Select an operator, such as `Is any of` or `Is none of`.
7. Click the plus sign (**+**) next to each attribute to select values for each.
8. Continue this process until you have created all of the conditions necessary.

Create Filters



1. In the Create a Filter section, select a data class to which you want to deny access.

> **Note**
>
> If you cannot find a data class that matches your criteria, you can create a new data class. See Create a Data Class.

2. Select an operator, such as `Is any of` or `Is none of`.
3. Click the plus sign (**+**) to add a value.

> **Important**
>
> If you enter a text string, it is case sensitive. This means that the capitalization in your entry must exactly match the capitalization in the actual values. For example, if you enter "Smith," but the actual value in the data source is either "SMITH" or "smith," your filter will not apply to that record.

4. Click the plus sign (**+**) again to add more values.

   For example, if you select a data class called "Last Name," select the "Is any of" operator, and enter a string of "Smith," the platform will drop any record with "Smith" in the "Last Name" column.

   If instead, you selected the "Is none of" operator, the platform will keep any record that has the value "Smith" in the "Last Name" column and drop all other records.

   > **Note**
   >
   > If you select **User Location**, you can select from a list of countries.

5. Click **Add Filter** to select another data class.
6. Assign attributes and filters to that data class.
7. Continue this process until you have specified conditions and access controls for the required data classes.

   The platform denies access to a field if it meets any one of the filters.

   > **Note**
   >
   > The order of the filters does not affect the level of access because the platform applies all filters. Any filter that applies to a field denies access to that field. If more than one filter applies, the result is the same—the platform denies access.

8. Click **Complete**.

   A confirmation message appears.
9. Click the **X** at the top-right of the page to return to the Policies tab.

   You need to submit the policy associated with this rule for approval for this new rule to take effect. See Submit a Policy for Approval.

3.3.7. Create a Transformation Rule

1. View a transformation policy.

   See Create a Policy.
2. Click the **Rules** tab.
3. Click **Add Rule** or click **More** (the three vertical dots) and select **Edit Policy**.

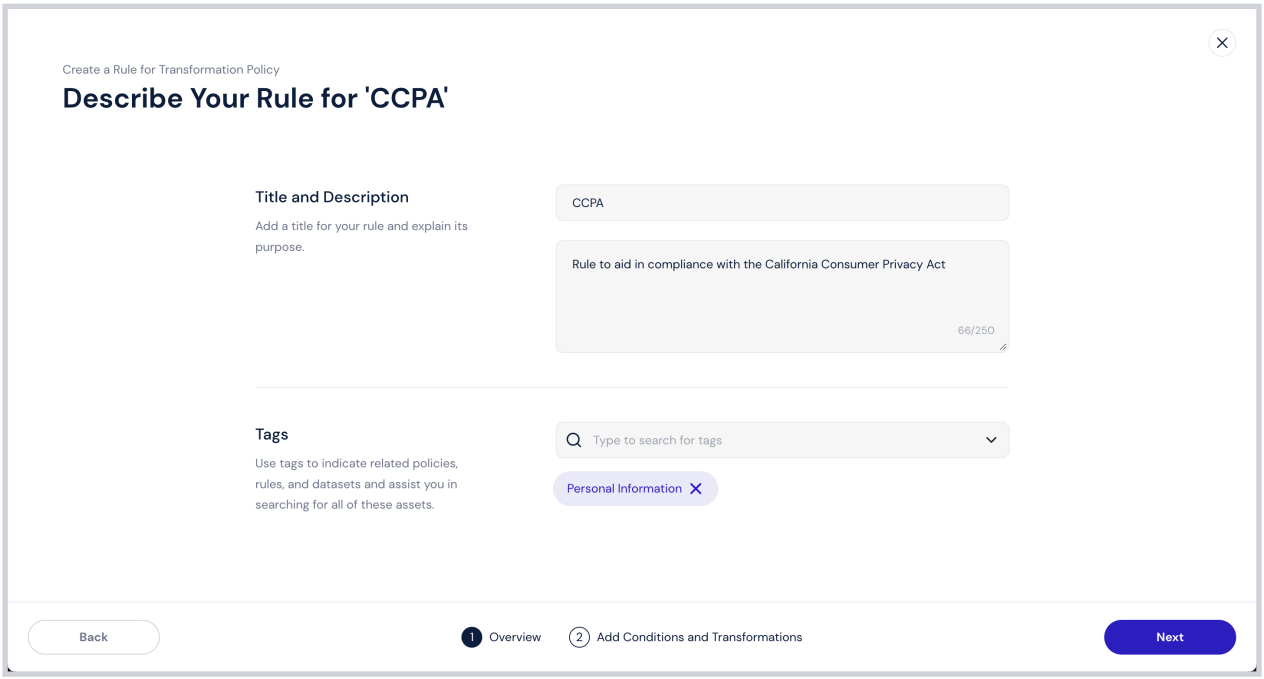
4. Click the plus sign (**+**).

   The Describe Your Rule page appears.
5. **Title—**Enter a title for the rule.
6. **Description—**Explain the rule's purpose.
7. **Tags—**Select tags to assign to the rule.

   Use tags to indicate related policies, rules, and datasets and assist you in searching for all of these assets. For example, you might wish to tag all of these assets with the name of the data privacy rule with which they help your organization comply, such as GDPR or HIPAA.

   > **Note**
   >
   > If you cannot find a tag that matches your criteria, you can create a new tag. See Create a Tag.

8. Click **Next**.



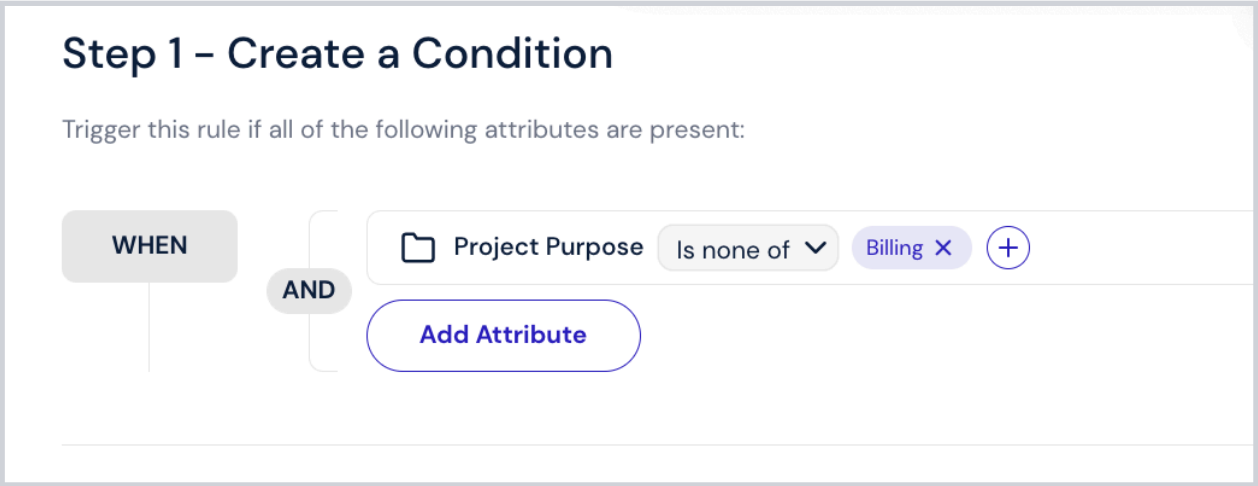
The How Should This Rule Work? page appears. On this page, you specify under which conditions transformations will apply to data classes.

1. In the Step 1 - Create a Condition section, click **Add Attribute**.

   The Add Attribute window appears.
2. Select all of the attributes that must be present for this rule to activate. These include tags, terms, purposes, user groups, and more.

   Each attribute appears. Note that these are joined with an `and` operator, which means that they all must be present for the condition to apply.
3. Select an operator, such as `Is any of` or `Is none of`.

4. Click the plus sign (**+**) next to each attribute to select values for each.
5. Click **Add Attribute** again to create another condition.
6. Continue this process until you have created all of the conditions necessary.



In the Step 2 - Assign a Transformation to a Data Class section, you can assign transformations at the field level or at the cell level. You can organize both into logical groups. By default, transformations are assigned at the field level. If you do not want to create a field-level transformation, click the trash can icon next to Field-Level Transformation.

1. Click **Add More**.
2. Select **Add Field-Level Transformation** or **Add Cell-Level Transformation**.

To add a *field-level transformation*:

1. Select a data class to which you want to assign a transformation.

> **Note**
> If you cannot find a data class that matches your criteria, you can create a new data class. See Create a Data Class.

> **Note**
> If you select **User Location**, you can select from a list of countries.

2. Assign a transformation to that data class.
3. Click **Add Transformation** to select another data class.
4. Continue this process until you have specified all field-level transformations relevant to the rule.

> **Note**
> You can only have one field-level transformation group per rule.

To add a *cell-level transformation*:

1. In the IF section, start creating a condition by selecting a data class.

   > **Note**
   >
   > During provisioning, if a data consumer uses an asset that has multiple fields with the same assigned data class that is used in a cell-level transformation IF statement, the platform will not know which one to evaluate. When executing a query, the platform will generate an error and stop provisioning.

2. Select an operator, such as `Is greater than`, `Is less than`, `Is any of`, or `Is none of`.
3. In the Add Value box, enter a string value for the data class, such as an age or a country.

   > **Important**
   >
   > If you enter a text string, it is case sensitive. This means that the capitalization in your entry must exactly match the capitalization in the actual values. For example, if you enter "Smith," but the actual value in the data source is either "SMITH" or "smith," your filter will not apply to that record.

4. Click **Add a Condition** to add another condition to the transformation and repeat the previous steps.
5. In the THEN section, select a data class to which you want to assign a transformation.
6. Select a transformation.

   > **Note**
   >
   > Cell-level transformations do not support the drop transformation type. If you'd like to remove data, you can use the redact with NULL transformation type.

7. Click **Add a Transformation** to add another transformation to this condition.
8. Optional:

   a. In the ELSE IF section, begin adding an alternative to your first condition by selecting a data class.

      > **Note**
      >
      > In the ELSE IF and ELSE sections, you cannot select new data classes. The data classes that appear are the ones that you chose in the IF statement.

b. Select an operator.
c. Select or enter a value.
d. In the THEN section, assign a transformation to each data class.

9. In the ELSE section, instruct the platform how to transform any data classes that you identified as part of your initial THEN condition that do not satisfy the condition in this group. The default transformation type is redact with NULL.
10. Click **Complete**.

    A confirmation message appears.
11. Click the **X** at the top-right of the page to return to the Policies page.

    You need to submit the policy associated with this rule for approval for this new rule to take effect. See Submit a Policy for Approval.

3.4. Transformations

A transformation defines a set of behaviors (privacy enhancing technologies) for the platform to execute on a field in a dataset to de-identify it, while still preserving data utility.

As a data guardian, you create transformations and add them to transformation rules, which determine when to apply the transformations to data.

3.4.1. Create a Transformation

1. Click **Policies** in the left navigation.
2. Click the **Transformations** tab.
3. Click **Create Transformation**.

   The Select a Transformation Type page appears.
4. Click **Select a Transformation**.

   A list of available transformation types appears.
5. Select a type of transformation.
6. Select the criteria for the transformation type.

   Tokenization Behavior—For regular expression transformation types, the criteria include whether to preserve data consistency when tokenizing data. Data consistency means that the same input value always results in the same token. This is important when considering relationships in the data. Consistent tokenization is required to maintain relationships between tables because the same value (that is, the same random token) must be present in several de-identified tables. If tokenization is not consistent, a new token will be created for each occurrence of a value. This means there will be multiple tokens for the same input value if it occurs more than once in the data.
7. Click **Next**.

   The Describe Your Transformation page appears.
8. **Title and Description—**Name the transformation and describe its function.
9. **Tags—**Select tags to associate with the transformation.

10. Click **Save**.

3.4.2. Transformation Types

The platform supports the following transformation and *tokenization* types:

- Constant Text Value—Replaces all input values with the same user-defined value.
- Drop Field—Removes the column entirely. It does not appear in the output.
- Generalize Date—Replaces input values with a generalized version of the date or a constant date if the input date is outside of a user-defined date range.
- Redact with NULL—Replaces field values with NULL.
- Regular Expression (Regex) Number Generation—Replaces input values with a randomly generated numeric token that matches a user-defined regular expression.
- Regular Expression (Regex) Text Generation—Replaces input values with randomly generated text-based token matching a user-defined regular expression.
- Retain Value—The input value is unchanged. The output value matches the input value.
- Truncate Text—Removes or retains part of the input value. The tokenized output is a truncated version of the input.

As a data guardian, you can take advantage of the platform's unique ability to configure tokens to maintain data consistency (called "consistent tokenization"), reversibility, and formatting of the data as needed.

Tokenization

Tokenization refers to replacing raw values with generated tokens.

The platform supports two types of tokenization:

- **Random Tokenization**—The platform replaces raw values with randomly generated tokens.
- **Derived Tokenization**—The platform replaces raw values with a token derived from the encrypted value of the input.

The process of tokenization may be consistent, meaning that the same input value always results in the same token whenever it is processed by the same rule within the same project, depending on the project collaboration mode.

Consistent tokenization is required if a keyed relationship is to be maintained, because the same token must be present in several de-identified tables. For rules that are configured to mask or tokenize consistently, you might optionally allow for unmasking of original values. That is, enable re-identication of data that was de-identified through that rule.

If tokenization is not consistent, a new token will be created for each occurrence of a value. This means there will be multiple tokens for the same input value if it occurs more than once in the data.

Constant Text Value

This section provides a comprehensive description of the constant text value transformation type.

Data Types

The constant text value transformation supports the following data types:

- String

Description

The constant text value transformation type allows you to mask all string values of a given class with the same piece of text.

Masking Behavior

The input value is always replaced with the constant text value.

Examples

Here are examples of constant text values that you might use.

Field	Formats	Constant Text Value	Output
Password	P@ssw0rd1 MyP@$$w0rd!2345	********	********
Postal Code	12345 AB1 2CD	00000	00000

Drop Field

This section provides a comprehensive description of the drop field transformation type.

Data Types

The drop field transformation supports the following data types:

- All

Description

The selected field is dropped from the output schema.

Masking Behavior

No masking behavior takes place.

Generalize Date

This section contains a comprehensive description of the generalize date transformation type.

Data Types

The generalize date transformation supports the following data types:

- Date
- Timestamp

Description

The input date is replaced by a generalized version of the date or a constant date if the input date is outside of a user-defined date range. For example:

`23-12-2018`

would become `01-01-2018` if the date is generalized to only preserve the year.

For timestamp and datetime data types the format is `HH:mm:ss`, and the output time is always set to midnight (`00:00:00`). For example:

`16-09-2018T11:33:00`

would become `01-01-2018T00:00:00` if the date is generalized to only preserve the year.

Generalization Behavior

The following table describes the generalization options that you can use to define the date to be returned to replace the input value:

Option	Description
Preserve year (01-01-YYYY)	Select this option to retain the year from the input values but generalize the day and month to `01-01`. Note that the platform might still change the year depending on the settings that you select in **Protect elderly individuals** and **Protect minor or underage individuals**.
Preserve month and year (01-MM-YYYY)	Select this option to retain the month and year from the input values but generalize the day to `01`. Note that the platform might still change the month and year depending on the settings that you select in **Protect elderly individuals** and **Protect minor or underage individuals**.
Preserve day, month, and year (DD-MM-YYYY)	Select this option to retain the day, month, and year from the input values. Note that the platform might still change the day, month, and year depending on the settings that you select in **Protect elderly individuals** and **Protect minor or underage individuals**.

Option	Description
Protect elderly individuals	Select this option to protect individuals older than a specified age, then enter that age in the box provided. Note that the platform calculates the data subject's age as of the date that it provisions the data. Also note that this setting works in conjunction with the "Preserve" setting that you select. That is, the platform will generalize all dates using the "Preserve" setting, but it will return constant values for any dates outside of the range created by the age value set here. For example, if you select "Preserve month and year," and you specify to generalize the birth date for all individuals older than 90 years old, this would result in the following: If the platform provisions the data on December 31, 2022, then the platform will generalize any date that is after December 31, 1932 (2022-90=1932) to `01-[original month value]-[original year value]`. The platform will generalize any date that is before December 31, 1932 to `01-12-1932`. For example, the platform would change the input date of June 2, 1929 (`02-06-1929`) to `01-12-1932`. The platform would not generalize any date that is equal to December 31, 1932.
Protect minor or underage individuals	Select this option to protect individuals younger than a specified age, then enter that age in the box provided. Note that the platform calculates the data subject's age as of the date that it provisions the data. Also note that this setting works in conjunction with the "Preserve" setting that you select. That is, the platform will generalize all dates using the "Preserve" setting, but it will return constant values for any dates outside of the range created by the age value set here. For example, if you select "Preserve month and year," and you specify to generalize the birth date for all individuals younger than 16 years old, this would result in the following: If the platform provisions the data on December 31, 2022, then the platform will generalize any date that is before December 31, 2006 (2022-16=2006) to `01-[original month value]-[original year value]`. The platform will generalize any date that is after December 31, 2006 to `01-12-2006`. For example, the platform would change the input date of June 2, 2007 (02-06-2007) to `01-12-2006`. The platform would not generalize any date that is equal to December 31, 2006.

Note

This behavior of this transformation type is designed to help you comply with the US Health Insurance Portability and Accountability Act (HIPAA) requirement for the storage of ages and dates contained in patient healthcare data.

Redact with NULL

This section contains a comprehensive description of the redact with NULL transformation type.

Data Types

The redact with NULL transformation supports the following data types:

- All

Description

The field values in a column are replaced with NULL, but the column remains. All users see the column name, but each field in the column only includes NULL values.

This differs from the drop field transformation type, which removes the column and all of its metadata, altering the schema.

The redact with NULL transformation type is useful when you want to remove field values while preserving the schema of the asset after the data transformation. Maintaining the original schema is often critical to ensure correct processing by data integration, data replication, or data virtualization tools. Maintaining the original schema might be required to support software development and testing and when working with some third-party applications.

Masking Behavior

The field values are replaced with NULL values.

Regular Expression Number Generator

This section provides a comprehensive description of the regular expression (regex) number generation transformation type.

Data Types

The regex number generator supports the following data types:

- Byte
- Integer
- Long
- Short

Description

The value is replaced by a randomly generated number that matches the supplied regular expression. For example, take an initial value of:

```
1234
```

To replace this with a randomly generated 4-digit number, you could use the following regular expression:

```
[1-9]{4}
```

which could produce a value such as:

```
4159
```

A more complex example, would be:

```
[1-9]{1}[0-9]{4}
```

The first part of the expression (`[1-9]{1}`) generates one number between `1` and `9`. The second part of the expression (`[0-9]{4}`) generates four numbers between `0` and `9`.

The regular expression specifies a pattern that the generated number should match. Only expressions generating integers are accepted, and the range of potential minimum and maximum values that can be generated based on the expression has to be compatible with the data types the regex number generator is applied to.

Masking Behavior

The options are described in the following table:

Option	Description
Regular expression	The pattern that the generated number should match.

You can unmask fields that have been masked with the regex number generator.

Examples

Here are examples of regular expressions that you can use to match some example fields and formats:

Field	Format	Expression
Employee ID (For example, 52938486)	12345678	[1-9]{8}
US ZIP Code (For example, 93612)	12345	[1-9]{5}

Regular Expression Text Generator

This section provides a comprehensive description of the regular expression (regex) text generation transformation type.

Data Types

The regex text generator supports the following data types:

- Text

Description

The value is completely replaced by a randomly generated string that matches the supplied regular expression. For example, for an initial value:

```
abcdef
```

you could use the following regular expression:

```
[a-z]{6}
```

This would produce an output such as:

`mvskyc`

Masking Behavior

The options are described in the following table:

Option	Description
Regular Expression	The pattern that the generated text should match.

You can unmask fields that have been masked with the regex text generator.

> **Note**
>
> When transforming data using a consistent regular expression (regex), it is expected that all data values match the regular expression to return consistent results. If all the data values do not match the regular expression, the request will fail and return an error. For example, if a regex specifies a capital letter `[A-Z][a-z]{1,15}`, and the the data doesn't include a capital letter, the platform returns an error.

Examples

Here are examples of regular expressions that you could use to match some example fields and formats:

Field	Format	Expression
Email Address	xxxxxxx@xxxxx.com	[a-z]{7}\@[a-z]{5}\.com
Last Name	xxxxxxxxxxxxxxx	[a-z]{15}

Retain Value

This section provides a comprehensive description of the retain value transformation type.

Data Types

The retain value transformation supports the following data types:

- All

Description

The input value is unchanged. The output value matches the input value.

Masking Behavior

No masking behavior takes place.

Truncate Text

This section provides a comprehensive description of the truncate text (sometimes called "clip text") transformation type.

Data Types

The truncate text privacy enhancing technology supports the following data types:

- Text

Description

Truncate text removes part of the input value. You can select the transformation to start from the left of the value (default) or from the right of the value.

For example, if the number of characters to clip or retain is three characters, starting from the left:

```
SW12 7AH
```

becomes:

```
SW1 (if Preserve selected characters is checked)
2 7AH (if Preserve selected characters is not checked)
```

Masking Behavior

The options are described in the following table:

Options	Description
Number of characters to remove	The number of characters to clip from the input value.
Start from the left	Remove the number of specified characters starting from the left. This is the default.
Start from the right	Remove the number of specified characters starting from the right.
Preserve select characters	The action to take with the characters that have been selected from the input to derive the final output value. The truncated (clipped) output value is obtained by either retaining the specified number of selected characters (if **Preserve selected characters** is checked) while removing all other characters, or removing those selected characters while retaining the rest (if **Preserve selected characters** is not checked).

You cannot unmask fields that have been masked with truncate text.

3.5. Policies, Rules, and Transformations FAQ

The following are frequently asked questions regarding policies, rules, and transformations:

Since policies determine which assets a user can access, how do we know which asset the user is actually requesting?

Assets are identified through addition to projects. Projects have an ID, a purpose, and possibly tags. When a data guardian approves a project, the data consumer receives a proxy connection string for their queries. When data consumers access assets through that connection string, the platform applies and resolves policies. Each query for a given asset generates an audit event, which specifically includes the SQL request, the assets requested, and the policies and transformations applied.

Both access control policies and transformation policies act on a variety of factors (including user groups, user location, project purpose, and source location associated with the asset). As specific conditions are triggered and satisfied, access control policies filter out identified records from view, and transformation policies apply transformations to specific columns or cells.

Policies define what should happen when the user tries to access data, but nothing happens until the user runs a query. The asset that the user requests does not come into play in any policy or project because the platform evaluates this dynamically when the user submits a query. In other words, transformation policies and projects do not know which asset the user requests at query time.

Does the approval of projects imply that users already have access to the requested assets? If so, do we still need access control policies?

Projects specify the assets to which a given user or user group may gain access through a Privitar-specified URL by which the platform applies policies when the data consumer makes queries.

This means that once a data guardian approves a project, all users and groups with access to the project can query the contained assets. However, the platform then applies the access control and transformation policies to the query before retrieving data based on the characteristics of the project, the user, and the assets in combination as defined in the policies by the data guardian. The returned data for the query is limited and transformed based on the relevant policies.

Access control policies remain a key part of the project-based process. These filter or limit the data returned to the user based on the identified conditions.

Important

If an organization grants access directly to a database to a specific user role and provides a user with those credentials to connect directly to said database, then they have bypassed all policy controls. As part of the adoption of a data security platform, organizations must remove such direct access.

When do data guardians need to review policies?

Your organization should set up an ongoing process for review and approval of policies.

As a data guardian, you should periodically review access control policies to determine who should have access as your organization adds new users and user groups.

To minimize this type of work, as new assets are onboarded and new data classes are added, review the new asset requests to determine whether the existing platform policies already cover all the noted data classes or whether policies require additions or updates.

As you and other data guardians add new tags and terms to the platform, you might need to incorporate them into policy conditions.

Should each asset have its own transformation policy?

Consider a situation where all your users have similar access to all assets. For example, for Employee Asset, you might be thinking of creating a transformation policy with transformation rules similar to the following:

Number	Rule Condition	Action
1	If `User.Group` = (“Team1” or “Team2”) AND `Asset.Tag` = “Employee”	Transform Data Class A Retain Data Class B Transform Data Class C
2	If `User.Group` = (“Team2” or “Team3”) AND `Asset.Tag` = “Employee”	Drop Data Class D Transform Data Class E Retain Data Class F Retain Data Class G Retain Data Class H
3	If `User.Group` = (“Team1” or “Team2” or “Team3”) AND `Asset.Tag` = “Employee”	Retain Data Class A Retain Data Class B Retain Data Class C Retain Data Class D Retain Data Class E Retain Data Class F Retain Data Class G Retain Data Class H

Then, for the next asset:

Number	Rule Condition	Action
1	If `User.Group` = (“Team1” or “Team2”) AND `Asset.Tag` = “Accounts”	Transform Data Class A Retain Data Class J Retain Data Class K

While the above is one prescriptive way to manage your policies, it will require at least one policy for each asset or dataset, if you do this at the dataset level. However, by doing so, you degrade the ability of the platform to handle new assets. You will end up with many policies and rules in which the bulk of the actions involve similar or even duplicated actions across policies. For example, retaining or dropping the same data classes.

Instead, decide which data classes need to be protected, and then define policies for the required transformations. Leave the rest to flow through. You can simplify the above example into rules that govern which data classes the platform needs transform or drop. The last flow-through rule will cover everything else.

In the following example, the actual assets do not matter anymore. Since the data classes apply across all assets, you just need to ensure that every data class has been considered in at least one policy rule. Adding new assets becomes easy, because you only need to discover which new data classes do not currently exist and handle them in the policies. However, if you need to transform a specific data class differently across differing assets, you will need to create additional policies to manage those specific deviations.

Number	Rule Condition	Action
1	If `User.Group` = (“Team1” or “Team2”)	Transform Data Class A Transform Data Class C
2	If `User.Group` = (“Team2” or “Team3”)	Drop Data Class D Transform Data Class E
3	If `User.Group` = (*)	Retain Data Class A Retain Data Class B Retain Data Class C Retain Data Class D Retain Data Class E Retain Data Class F Retain Data Class G Retain Data Class H Retain Data Class J Retain Data Class K

Should each project have its own access control policy or transformation policy?

A project is a request for access to assets. Over the lifecycle of an asset, there are likely to be multiple projects from different users that may request the same assets.

A well-designed platform prevents the need to modify policies every time data consumers submit a new project. This allows faster access to the data once a data guardian approves the project, which results in happier users.

Should each data class have its own transformation rule?

There is no need to have a transformation rule for each data class. How you write transformation rules to group data classes depends on how similar the data classes are.

If many of the data classes have similar conditions, focus on writing rules for data classes that you want retained or dropped.

If many of the data classes have dissimilar conditions, focus on writing rules for data classes that you want transformed.

This will ensure that you have a minimal set of rules. This makes the platform more efficient because the platform will not need to evaluate each rule in order to determine the final set of data classes that the data consumer can access.

Instead of managing access control policies, can we just put users in LDAP groups and build access control policies using these groups in the conditions?

While this is possible, remember that you mainly use LDAP groups to manage:

- users' access to the platform itself
- user roles in the data exchange

If you intend to deny users' access to entire assets, you could use LDAP groups to exert a broad level of control. However, if you want to deny users' access to specific records, you will need to use access control policies.

Should we try to limit the number of policies and rules?

Yes, having too many policies and rules complicates the logic that you must manage, makes troubleshooting more challenging, and also impacts performance of the platform, as it will take more time to evaluate all the policies and rules.

You should also keep the set of data classes small. Map fields that do not need to be transformed or dropped to generic data classes.

How can we grant access to a user who was previously denied access in an access control policy?

You will have to revisit each access control policy to ensure that the user will not be denied access by any policy. If the user is denied access by any policy, you will need to update the rules to avoid denying access to that user. Next, add or modify transformation policies where necessary to verify that the user can access the right set of data classes.

Can we incorporate multiple business terms into our policy conditions?

For organizations with a rich business term taxonomy, there might be a temptation to include as many terms as possible to align them to existing business terminology. However, you will find it a challenge to represent different business concepts in just

terms because not every entity in the platform can have associated terms. For example, datasets only have tags, while assets and fields can have both terms and tags.

Instead, you should use a combination of terms and tags to represent your business concepts. Use terms to indicate the business area, and use tags to add specificity. For example, you might use the term "Data Subject" paired with the tags "Customer," "Vendor," and "Employee" to align with your organization's concept of customer, vendor, and employee data subjects.

Also note that you can use each entity only once within each rule. For example, you can use only one condition that relates to an asset's term and one condition that relates to an asset's tag.

Allowed in Same Rule	Not Allowed in Same Rule
If `Asset.Term` is A or B or C	If `Asset.Term` is A or B or C AND `Asset.Term` is D or E or F
If `Asset.Term` is A or B or C AND `Asset.Tag` is G or H	If `Asset.Term` is “Data Subject: Customer” or “Data Subject: Vendor” or “Data Subject: Employee” AND `Asset.Term` is “Data Classification: RESTRICTED” or “Data Classification: NON-RESTRICTED”

4. Adding Data

As a data owner, you can add data to the data exchange. You do this by taking the following actions:

1. Create a dataset
2. Create a connection to the data source
3. Add an asset to a dataset
4. Describe an asset and submit it for approval

4.1. Datasets

A dataset is a logical container of assets that is also known as a "data product." Its purpose is to group and facilitate an easier search experience. Data owners make datasets available to data consumers.

4.1.1. Create a Dataset

1. Click **Data Exchange** in the left navigation.
2. Click the plus sign (**+**).

   The Dataset Details page appears.
3. **Title—**Enter a title for the dataset.
4. **Description—**Explain the dataset's purpose.
5. **Dataset Members—**Dataset membership allows you to restrict who may view the dataset and its assets. Restricting access means that only those specified may view the dataset and add its assets to a project.

   Select one of the following options:

   a. **Unrestricted: Anyone in your organization may view the dataset**

      This option allows any user with any role in this data exchange to view the dataset and its assets.

   b. **Restricted: Only dataset members may view the dataset**

      This option restricts dataset visibility to the dataset members that you select. If you do not select any members, you are the only person who can view the dataset and its assets.
6. **Add Members—**If you selected **Restricted**, you can select which users and user groups can be members of this dataset. (If you selected **Unrestricted**, **Add Members** does not appear.)

   a. Click the plus (+) sign.

      The Add Members window appears.

   b. Search for users and user groups.

   c. Click each user or user group to add as dataset members.

   d. Click **Add**.
7. **Tags—**Select tags to assign to the dataset.

Use tags to identify the subject area or domain of the dataset and to help facilitate search.

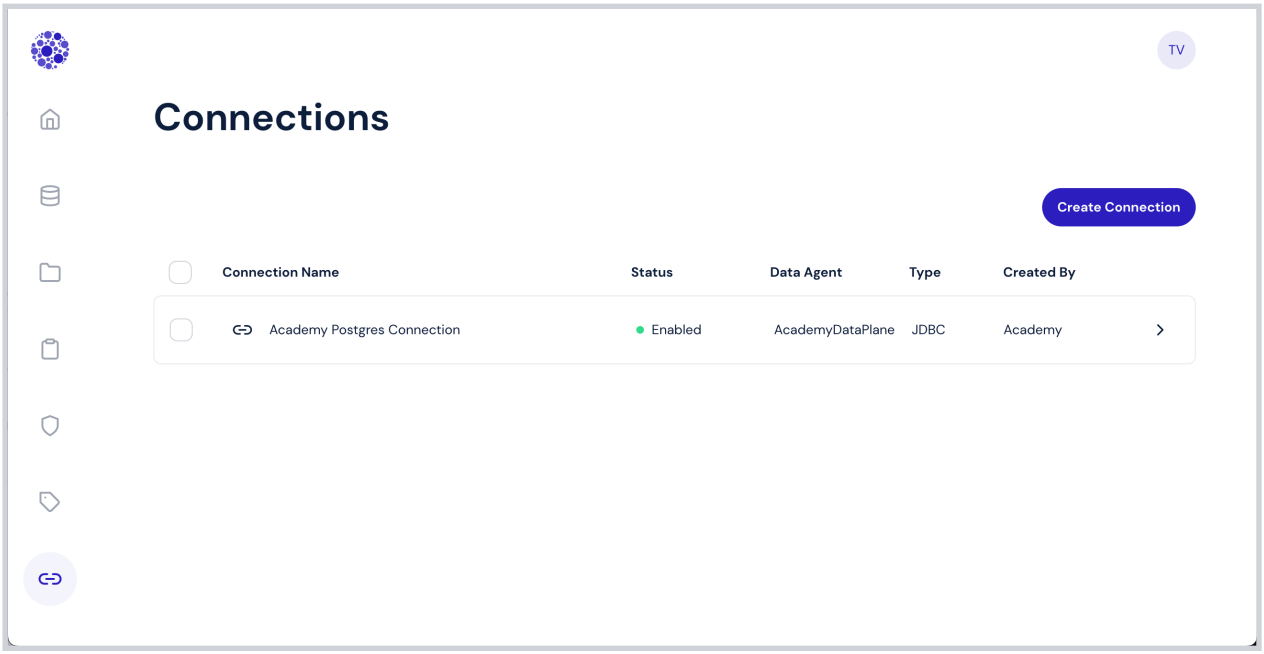
> **Note**
>
> If you cannot find a tag that matches your criteria, you can create a new tag. See Create a Tag.

8. Click **Create.**

   A confirmation message appears, and then the Data Exchange page refreshes, and the newly created dataset appears.

> **Important**
>
> Regardless of whether a dataset has restricted or unrestricted membership, only an exchange administrator or the data owner who created the dataset may edit or delete it.

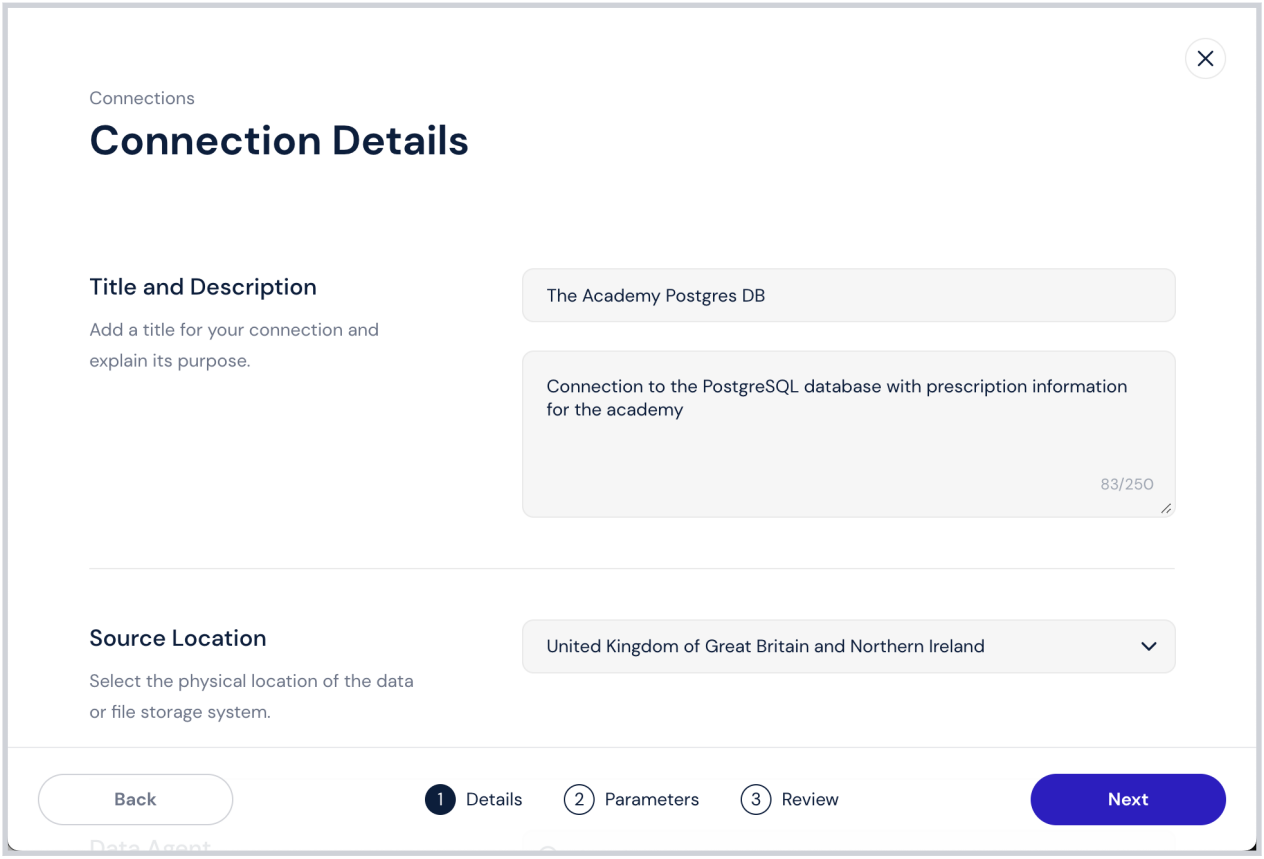
4.2. Connections



A connection is a configuration for connecting to and reading data from a data source, such as a JDBC connection string. The platform uses this connection information to read metadata attributes from a data asset, to read the data itself, and to write the processed data to the target location.

As a data owner, you set up a connection to ingest data and metadata from a data source into the platform.

4.2.1. Create a Connection to the Data Source



As a data owner, you can create a connection to a data source.

1. Click **Connections** in the left navigation.
2. Click **Create Connection**.

   The Connections Details page appears.
3. **Title—**Enter a name for the connection.
4. **Description—**Explain the connection's purpose.
5. **Source Location—**Select the physical location of the data or file storage system.

   For example, if the data is stored on a database that resides in your organization's office in London, select "United Kingdom." If the data is stored in an Amazon S3 bucket that resides in the region us-east-1, select "United States of America."
6. **Data Agent—**Select which *data agent*s can use this connection.

   > **Note**
   >
   > The selected data agents must be able to connect to the infrastructure that you define in the following steps.
7. **Connection Status—**Select **Enabled** when you are ready to use this connection. Select **Disabled** if the connection is not yet ready for use.
8. Click **Next**.

   The Connection Parameters page appears.

9. Click **Select a Connection Type**.
10. **Connection Type—**Select the type of connection.
11. **JDBC Connection String—**Enter the JDBC connection string required to connect to your database.

    Important

    The string must have a specific format depending on the database type. For details about the connection string to use with specific drivers, see your database vendor's documentation.

12. **Username—**Enter the username of the system user needed to authenticate the connection.

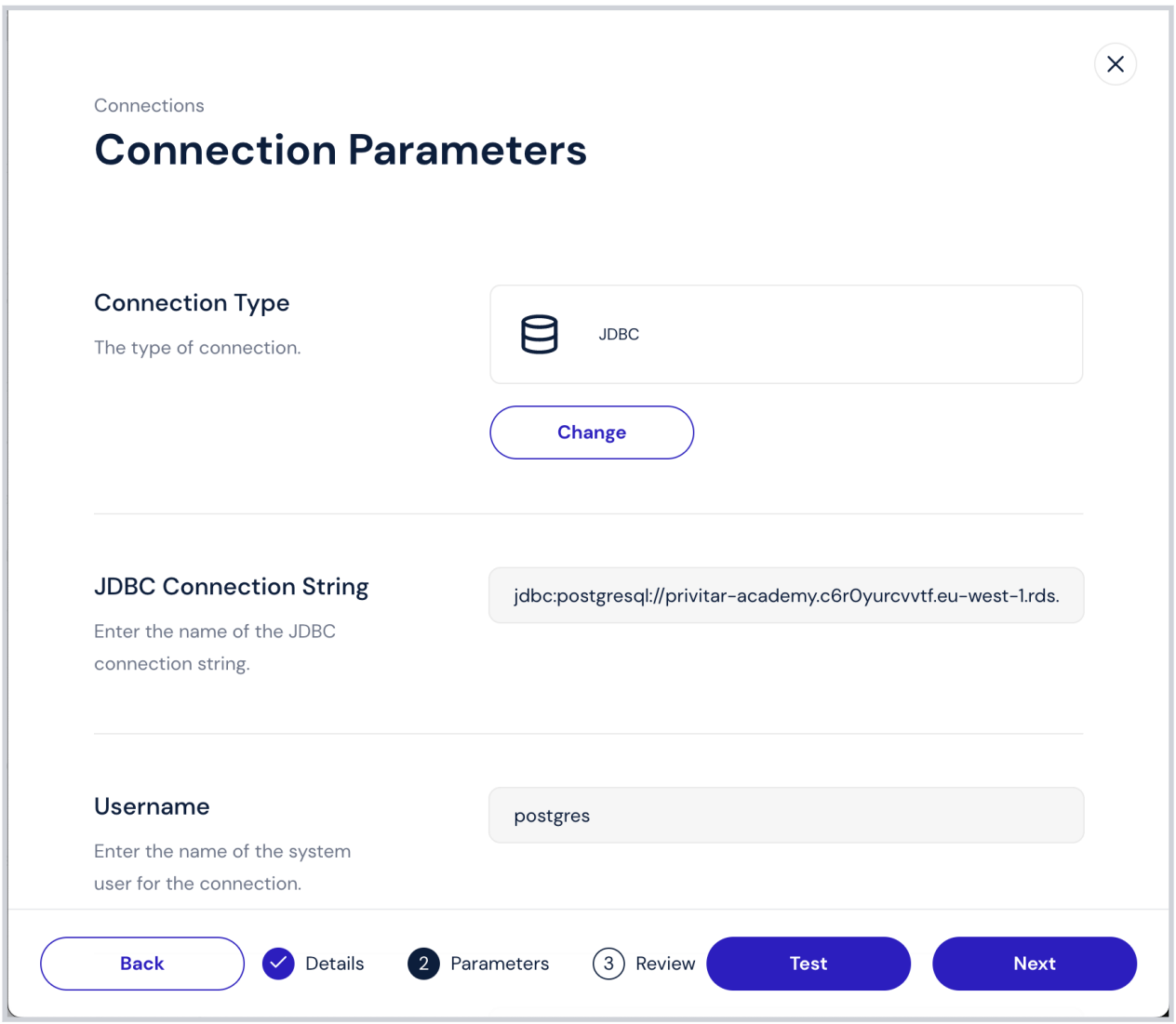
    Note

    We suggest that you use a shared username and password rather than your own username and password because this username and password will be shared by all users using this connection.

13. **Password—**Enter the password needed to authenticate the connection.
14. **Extra Parameters—**If your connection requires parameters not specified in the JDBC connection string, click **Add Parameters** to add them here. Check your database vendor's documentation to learn more.
15. Click **Test** to test the connection from each of the data agents that you selected.
16. Review the results of the tests.
17. When you are ready to save the connection details, click **Next**.

    The Review and Confirm page appears.
18. Review the connection details.
19. Click **Save**.

    A confirmation message appears.



4.2.2. Create a Connection to Apache Hive

You can use Apache Hive as a data source with Privitar Data Security Platform.

To connect to Apache Hive, you must:

1. Meet the Apache Hive Connection Prerequisites
2. Build an Apache Hive Connection String
3. Authenticate to Apache Hive

Meet the Apache Hive Connection Prerequisites

Before you connect to Apache Hive, you must:

1. Have a system user that is able to authenticate to Apache Hive using a username and password and has read access to the relevant databases and tables
2. Have access to the SSL certificate used to encrypt the connection (or the relevant certificate authority certificates)

If your Secure Sockets Layer (SSL) source uses privately-signed server certificates, you must modify the truststore of your data plane in order to trust the server certificates as follows:

1. Obtain the SSL certificate from the data source.
2. Convert the SSL certificate to a JKS truststore.
3. Copy the truststore into the `shared/truststores/` location of your data plane configuration mounted volume (the volume used to store JDBC drivers).

   > **Note**
   >
   > You will need to refer to this truststore when configuring the SSL JDBC properties. By default, the truststore is mounted on `/config/shared/truststores/truststore.jks`.

   The mounted volume's directory structure should look similar to the following:

```
shared/
|  jdbc-drivers/
|       hive-42.2.23.jar
|  truststores/
|       truststore.jks
data-agent
|  EMPTY
 data-proxy
|  EMPTY
```

4. Download the JDBC JAR driver that you will use to connect to the data source.
5. Place the JDBC JAR driver into the `shared/jdbc-drivers/` location of your data plane configuration mounted volume (the volume used to store JDBC drivers).

For example, the SSL settings for Hive might look like the following:

```
jdbc:hive2://ip-172-31-26-172.eu-west-2.compute.internal:10000/
default;ssl=true;sslTrustStore=/config/shared/truststores/
truststore.jks;trustStorePassword=changeit
```

Build an Apache Hive Connection String

The following is an example of a complete Apache Hive connection string:

```
jdbc:hive2://localhost:10000/database1
```

To build an Apache Hive connection string, follow this example. Note that it has the following segments:

```
jdbc:hive2://<host>:<port>/<dbName>;<sessionConfs>?<hiveConfs>#<hiveVars>
```

If you have configured to use SSL in the previous section, the SSL settings for Hive might look like the following:

```
jdbc:hive2://ip-172-31-26-172.eu-west-2.compute.internal:10000/
default;ssl=true;sslTrustStore=/config/shared/truststores/
truststore.jks;trustStorePassword=changeit
```

The following table includes a description of each segment.

Table 1. Apache Hive Connection String

String Segment	Description
`host`	The HiveServer hosting node. Required.
`port`	The port that HiveServer listens to. The default port number is `10000`. Required. Your Hive instance might not use the default port number. To confirm the port number, check the `hive.server2.thrift.port` property the `hive-site.xml` configuration file.
`dbName`	The name of the Hive database. Required.
`sessionConfs`	Key-value pairs for the JDBC driver in the format `<key1>=<value1>;<key2>=<value2>;` Optional.
`hiveConfs`	Key-value pairs for Hive in the format `<key1>=<value1>;<key2>=<value2>;` Optional.
`hiveVars`	Key-value pairs for Hive variables in the format `<key1>=<value1>;<key2>=<value2>;` Optional.

Authenticate to Apache Hive

The Privitar Data Security Platform currently supports username/password authentication for Apache Hive.

Enter the system user's Apache Hive credentials in the Username and Password fields on the platform's Connections page.

4.2.3. Create a Connection to Apache Spark

You can use Apache Spark as a data source with Privitar Data Security Platform.

To connect to Apache Spark, you must:

1. Meet the Apache Spark Connection Prerequisites
2. Build an Apache Spark Connection String
3. Authenticate to Apache Spark

Meet the Apache Spark Connection Prerequisites

Note

Most of the settings for the Spark Thrift server are the same as those for HiveServer2. To learn more, see https://spark.apache.org/docs/latest/sql-distributed-sql-engine.html

Before you connect to Apache Spark, you must:

1. Have a system user that is able to authenticate to Apache Spark using a username and password and has read access to the relevant databases and tables
2. Have access to the SSL certificate used to encrypt the connection (or the relevant certificate authority certificates)

If your Secure Sockets Layer (SSL) source uses privately-signed server certificates, you must modify the truststore of your data plane in order to trust the server certificates as follows:

1. Obtain the SSL certificate from the data source.
2. Convert the SSL certificate to a JKS truststore.
3. Copy the truststore into the `shared/truststores/` location of your data plane configuration mounted volume (the volume used to store JDBC drivers).

   > **Note**
   >
   > You will need to refer to this truststore when configuring the SSL JDBC properties. By default, the truststore is mounted on `/config/shared/truststores/truststore.jks`.

   The mounted volume's directory structure should look similar to the following:

```
shared/
|  jdbc-drivers/
|       hive-42.2.23.jar
|  truststores/
|       truststore.jks
data-agent
|  EMPTY
 data-proxy
|  EMPTY
```

4. Download the JDBC JAR driver that you will use to connect to the data source.
5. Place the JDBC JAR driver into the `shared/jdbc-drivers/` location of your data plane configuration mounted volume (the volume used to store JDBC drivers).

For example, the SSL settings for Spark might look like the following:

```
jdbc:hive2://ip-172-31-26-172.eu-west-2.compute.internal:10000/
default;ssl=true;sslTrustStore=/config/shared/truststores/
truststore.jks;trustStorePassword=changeit
```

Build an Apache Spark Connection String

> **Note**
>
> Most of the settings for the Spark Thrift server are the same as those for HiveServer2. To learn more, see https://spark.apache.org/docs/latest/sql-distributed-sql-engine.html

The following is an example of a complete Apache Spark connection string:

```
jdbc:hive2://localhost:10000/database1
```

> **Note**
>
> The Spark Thrift server uses the same JDBC driver as HiveServer2.

To build an Apache Spark connection string, follow this example. Note that it has the following segments:

```
jdbc:hive2://<host>:<port>/<dbName>;<sessionConfs>?<hiveConfs>#<hiveVars>
```

If you have configured to use SSL in the previous section, the SSL settings for Spark might look like the following:

```
jdbc:hive2://ip-172-31-26-172.eu-west-2.compute.internal:10000/
default;ssl=true;sslTrustStore=/config/shared/truststores/
truststore.jks;trustStorePassword=changeit
```

The following table includes a description of each segment.

Table 2. Apache Spark Connection String

String Segment	Description
`host`	The Spark server hosting node. Required.
`port`	The port that the Spark server listens to. Required.
`dbName`	The name of the Hive database. Required.
`sessionConfs`	Key-value pairs for the JDBC driver in the format `<key1>=<value1>;<key2>=<value2>;` Optional.
`hiveConfs`	Key-value pairs for Hive in the format `<key1>=<value1>;<key2>=<value2>;` Optional.
`hiveVars`	Key-value pairs for Hive variables in the format `<key1>=<value1>;<key2>=<value2>;` Optional.

Authenticate to Apache Spark

The Privitar Data Security Platform currently supports username/password authentication for Apache Spark.

Enter the system user's Apache Spark credentials in the Username and Password fields on the platform's Connections page.

4.2.4. Create a Connection to Google BigQuery

You can use Google BigQuery as a data source with Privitar Data Security Platform.

To connect to Google BigQuery, you must:

1. Meet the Google BigQuery Connection Prerequisites
2. Build a Google BigQuery Connection String
3. Authenticate to Google BigQuery

Meet the Google BigQuery Connection Prerequisites

Before you connect to Google BigQuery, you must:

1. Have a Google account
2. Set credentials appropriate for the type of authentication you're using.

   See our instructions here: Authenticate to Google BigQuery

   See Google's instructions here: https://cloud.google.com/storage/docs/authentication.
3. Set permissions for your BigQuery and Google Cloud accounts

Build a Google BigQuery Connection String

To build a Google BigQuery connection string, follow the instructions in the Google BigQuery documentation.

The following is an example of a complete Google BigQuery connection string:

```
jdbc:bigquery://https://www.googleapis.com/bigquery/
v2:443;ProjectId=privitar-123456;OauthType=3;EnableSession=1;Location=europe-
west2;
```

The following table includes descriptions of some of the important segments.

Note

These descriptions assume that you are using one of the official drivers for BigQuery. Consult the Google BigQuery documentation for more information about drivers.

Table 3. Google BigQuery Connection String

String Segment	Description
`jdbc:bigquery://https://www.googleapis.com/bigquery/v2`	The host name is the URL to Google's BigQuery web services API.
`[443]`	The port number is 443.
`ProjectId=[name]-[nnnnnn]`	The name of the Google BigQuery project (not a project within the Privitar Data Security Platform), for example `privitar-123456`.
`OauthType=[n]`	Add the number corresponding to the Oauth type. `0`: The connector uses service-based OAuth authentication, such as Service Account Key File. (See "Service Account Key File" under Authenticate to Google BigQuery.) 3: The connector authenticates using Workload Identity federation. (See "Workload Identity" under Authenticate to Google BigQuery.)
`OAuthPvtKeyPath=[pathToSecretsFile]`	If you are using a service account key file to authenticate, include the path to the secrets file. (See "Service Account Key File" under Authenticate to Google BigQuery.)
`EnableSession=[n]`	Enter `1` for "true" (enable session).
`Location=[location]`	Enter the location for your Google BigQuery container (see https://cloud.google.com/bigquery/docs/locations).

Authenticate to Google BigQuery

Important

Google BigQuery does not support username/password authentication methods. For this reason, leave the Username and Password fields empty on the Connections page.

The Privitar Data Security Platform currently supports the following types of authentication:

Service Account Key File

- 1. Pass in a path to a secrets file accessible by data plane services as part of the JDBC connection string. For example:

```
jdbc:bigquery:<host>;ProjectId=<projectId>;OAuthPvtKeyPath=[pathToSecret
sFile];...
```

a. Ensure that the secrets file is in JSON format and contains the private key and certificates. For example:

```
{
"type": "service_account",
"project_id": <PROJECT_ID>,
"private_key_id": <PRIVATE_KEY_ID>,
"private_key": <PRIVATE_KEY>,
"client_email": <EMAIL>,
"client_id": "101273788015860915068",
"auth_uri": "https://accounts.google.com/o/oauth2/auth",
"token_uri": "https://oauth2.googleapis.com/token",
"auth_provider_x509_cert_url": "https://www.googleapis.com/
oauth2/v1/certs",
"client_x509_cert_url": <CERTS_URL>
}
```

2. In your JDBC connection string, set up the correct Oauth type in accordance with your configuration. For example, `OauthType=0`.

Workload Identity

1. Set up Workload Identity on your GKE cluster, granting access from the data proxy pods to Google BigQuery as described in Google BigQuery documentation.
2. In your JDBC connection string, set up the correct Oauth type in accordance with your configuration. For example, `OauthType=3`.

4.2.5. Create a Connection to Trino

You can use Trino as a data source with Privitar Data Security Platform.

To connect to Trino, you must:

1. Meet the Trino Connection Prerequisites
2. Build a Trino Connection String
3. Authenticate to Trino

Meet the Trino Connection Prerequisites

Before you connect to Trino, you must:

1. Have a system user that is able to authenticate to Trino using a username and password and has read access to the relevant databases and tables
2. Have access to the SSL certificate used to encrypt the connection (or the relevant certificate authority certificates)

If your Secure Sockets Layer (SSL) source uses privately-signed server certificates, you must modify the truststore of your data plane in order to trust the server certificates as follows:

1. Obtain the SSL certificate from the data source.
2. Convert the SSL certificate to a JKS truststore.
3. Copy the truststore into the `shared/truststores/` location of your data plane configuration mounted volume (the volume used to store JDBC drivers).

   > **Note**
   >
   > You will need to refer to this truststore when configuring the SSL JDBC properties. By default, the truststore is mounted on `/config/shared/truststores/truststore.jks`.

   The mounted volume's directory structure should look similar to the following:

```
shared/
|  jdbc-drivers/
|       trino-422.jar
|  truststores/
|       truststore.jks
data-agent
|  EMPTY
 data-proxy
|  EMPTY
```

4. Download the JDBC JAR driver that you will use to connect to the data source.
5. Place the JDBC JAR driver into the `shared/jdbc-drivers/` location of your data plane configuration mounted volume (the volume used to store JDBC drivers).

For example, the SSL settings for Trino might look like the following:

```
jdbc:trino://example.net:443/hive/?
user=test&password=secret&SSL=true;sslTrustStore=/config/shared/truststores/
truststore.jks;trustStorePassword=changeit
```

Build a Trino Connection String

The following is an example of a complete Trino connection string:

```
jdbc:trino://example.net:8080/hive/
```

To build a Trino connection string, follow this example. Note that it has the following segments:

```
jdbc:trino://${host}:<port>/<catalog>/
```

If you have configured to use SSL in the previous section, the SSL settings for Trino might look like the following:

```
jdbc:trino://example.net:443/hive/?
user=test&password=secret&SSL=true;sslTrustStore=/config/shared/truststores/
truststore.jks;trustStorePassword=changeit
```

The following table includes a description of each segment.

Table 4. Trino Connection String

String Segment	Description
`host`	The Trino hosting node. Required.
`port`	The port that Trino listens to. The default port number is 443. Required.
`catalog`	The name of the database catalog. Required.

Authenticate to Trino

The Privitar Data Security Platform currently supports username/password authentication for Trino.

Enter the system user's Trino credentials in the Username and Password fields on the platform's Connections page.

4.3. Assets

Assets are data structures; for example the tables in an Oracle® or PostgreSQL database.

As a data owner, you can add assets to datasets to make them available to data consumers.

4.3.1. Add an Asset to a Dataset

Important

If the dataset membership is restricted, only the data owner who created a dataset or members of that dataset may add, edit, delete, or submit assets for approval.

1. Click **Data Exchange** in the left navigation.
2. Click a dataset.

   The View Dataset page appears.
3. Click **Add Asset**.
4. Click **Next**.

   The Describe Asset page appears.

4.3.2. Describe and Register an Asset

As a data owner, after you add an asset to a dataset, the Describe Asset page appears.

Important

If the dataset membership is restricted, only the data owner who created a dataset or members of that dataset may add, edit, delete, or submit assets for approval.

1. **Title—**Enter a title for the asset.

   This does not have to match the table name.
2. **Description—**Describe the asset and explain the asset's purpose.

   This provides the data consumers with helpful context.
3. **Selected Connection Type—**Click **Select a Connection**.

   The Select a Connection window appears.

   Any previously created connections appears here.

   If no connections or no relevant connections appear, click the plus sign **(+)** and follow the instructions in To Create a Connection.
4. Select a connection.
5. Click **Select <CONNECTION NAME>**.

   The Schema Name and Table Name fields appear.
6. **Schema Name—**Enter the name of the schema containing the asset that you wish to register.

   > **Important**
   >
   > The schema name is case sensitive. Ensure that the capitalization in this entry matches the actual schema name to successfully add the asset.

7. **Table Name—**Enter the name of the database table that you want to register.

   > **Important**
   >
   > The table name is case sensitive. Ensure that the capitalization in this entry matches the actual table name to successfully add the asset.

8. **Tags—**Select *tags* to assign to the asset.

   Use tags to identity the subject area or domain of the asset and to help facilitate search. Rules rely on these tags to determine which transformation to use on data.

   > **Note**
   >
   > If you cannot find a tag that matches your criteria, you can create a new tag. See Create a Tag.

9. **Terms—**Select *terms* to assign to the asset.

   Use terms to identify the asset's business meaning.
10. Confirm that the connection details are correct.
11. Click **Save & Continue**.

    The Privitar Data Security Platform reads the structural details of the table (asset), such as its columns and their data types.

    The Describe Fields page appears.

12. Click plus sign (**+**), and then search for and select a data class, term, and tag to assign to this field.

    Rules use data classes as part of the determination for which privacy enhancing technologies to apply to each field. The default transformation is to drop the field so that it is not available to data consumers. If you do not assign a data class to a field, the platform will drop that field.

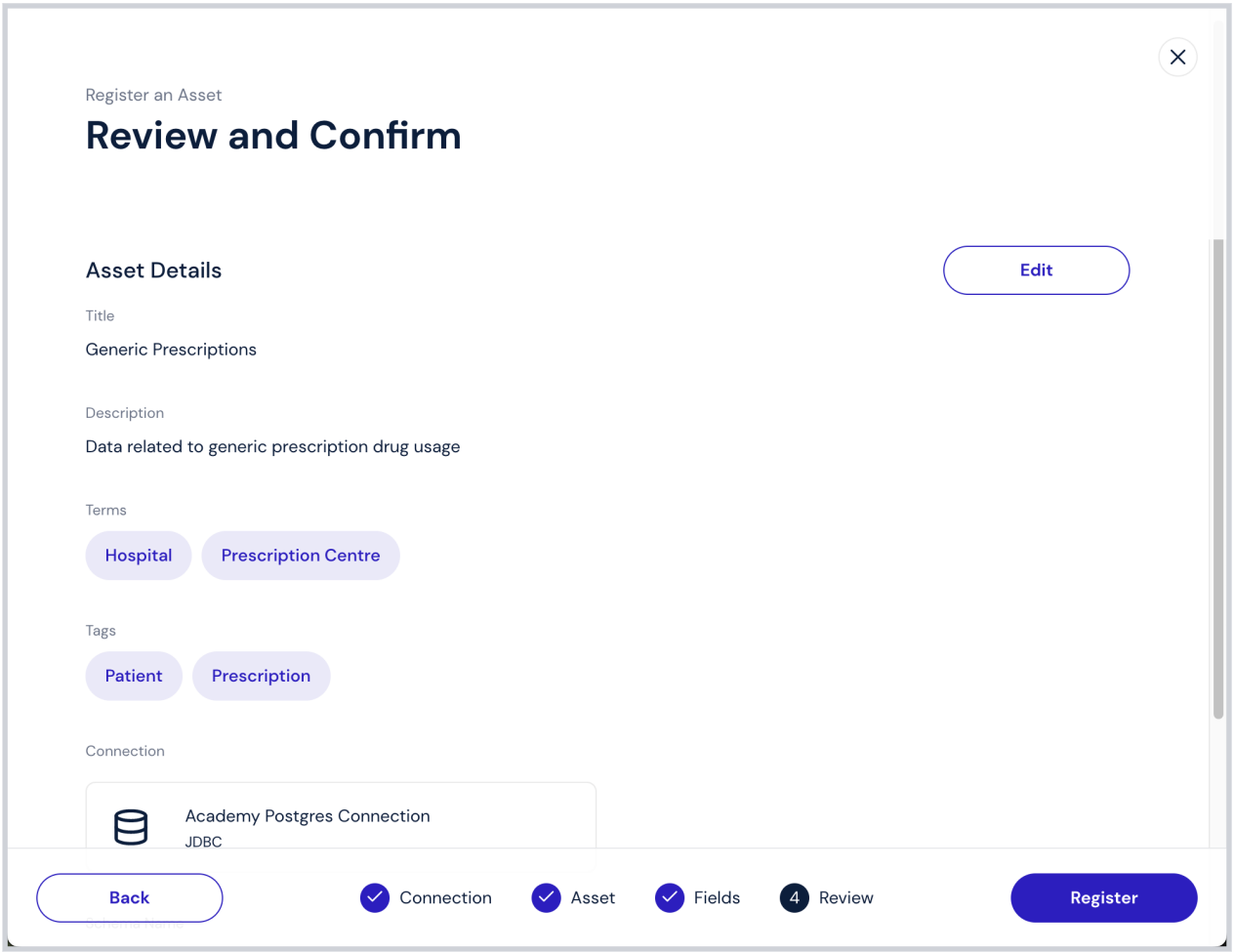
    Rules use tags and terms as part of the determination for which privacy enhancing technologies to apply to the data in each field.

    > **Tip**
    >
    > To edit multiple fields simultaneously, follow the steps in Edit Multiple Fields.

13. Click **Save** to save the assignment to the field.
14. Click **Next**.

    The Review and Confirm page appears.
15. Confirm that all of the asset's details are correct.
16. Click **Register**.

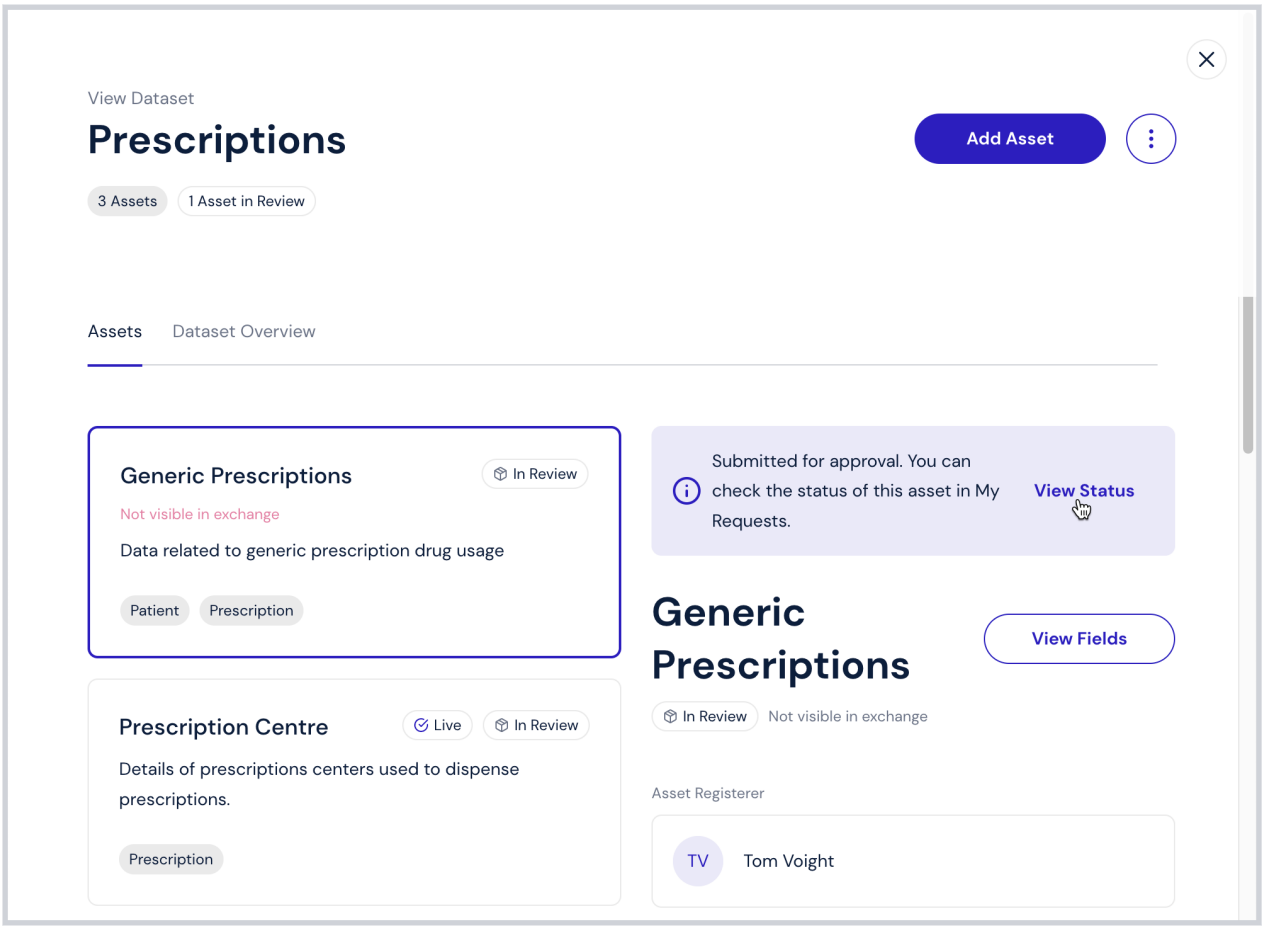
    A confirmation message appears.



The View Dataset page appears.

The asset remains in "In Review" status until a data guardian approves it. To learn more about this approval process, see Approve Asset Registration Tasks.

Click **View Status** to see the approval status of the asset.



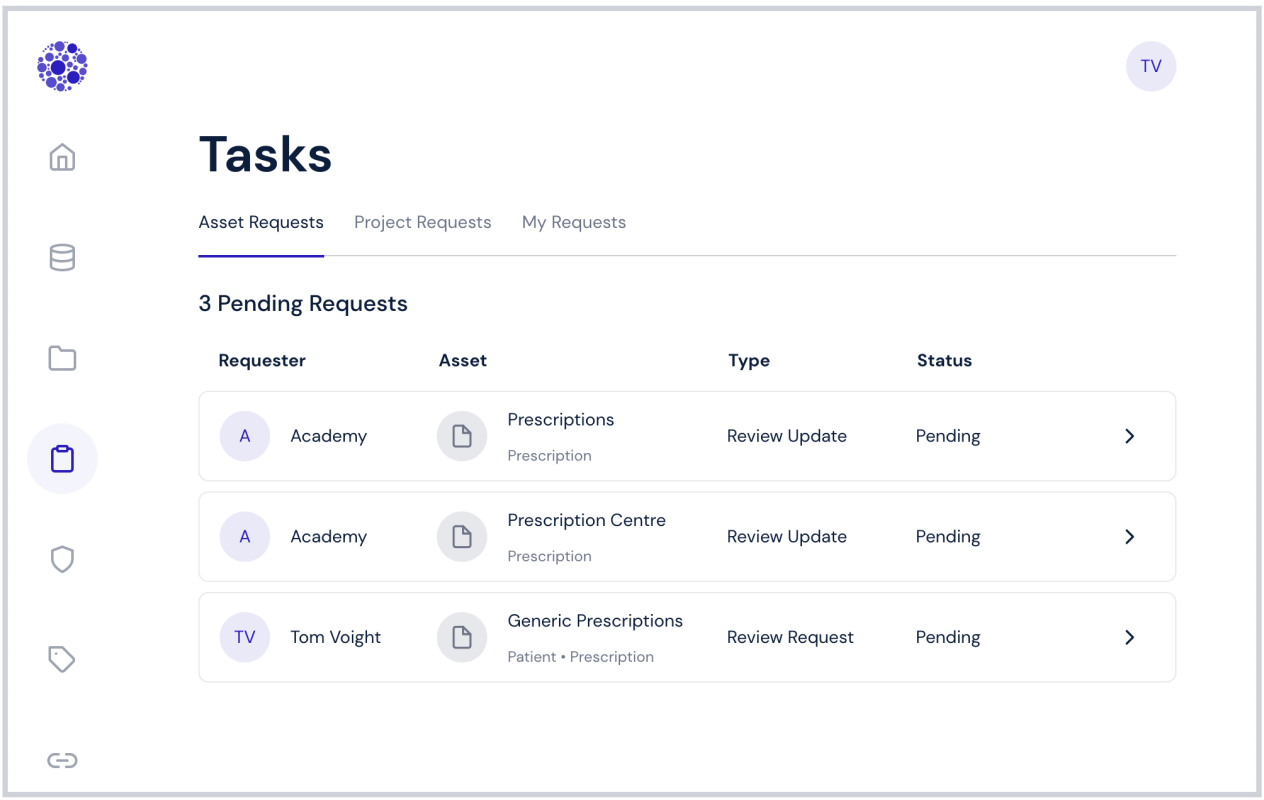
Until a data guardian has reviewed your request, the status is "In Review," and the asset is not yet available on the platform.

If a data guardian has approved your request, the status is "Live," and the asset is available on the platform.

If a data guardian has declined your request, the status is "Rejected," and the asset is not available on the platform. You can delete the request, or you can modify it and resubmit it.

4.3.3. Review an Asset Registration Request

As a data guardian, you review and approve asset registration requests submitted by data owners. You ensure that the data owner properly classified the asset and that there are appropriate rules for the protection of the data asset.



1. Click Tasks in the left navigation.
2. Click the **Asset Requests** tab.
3. Click a pending request.

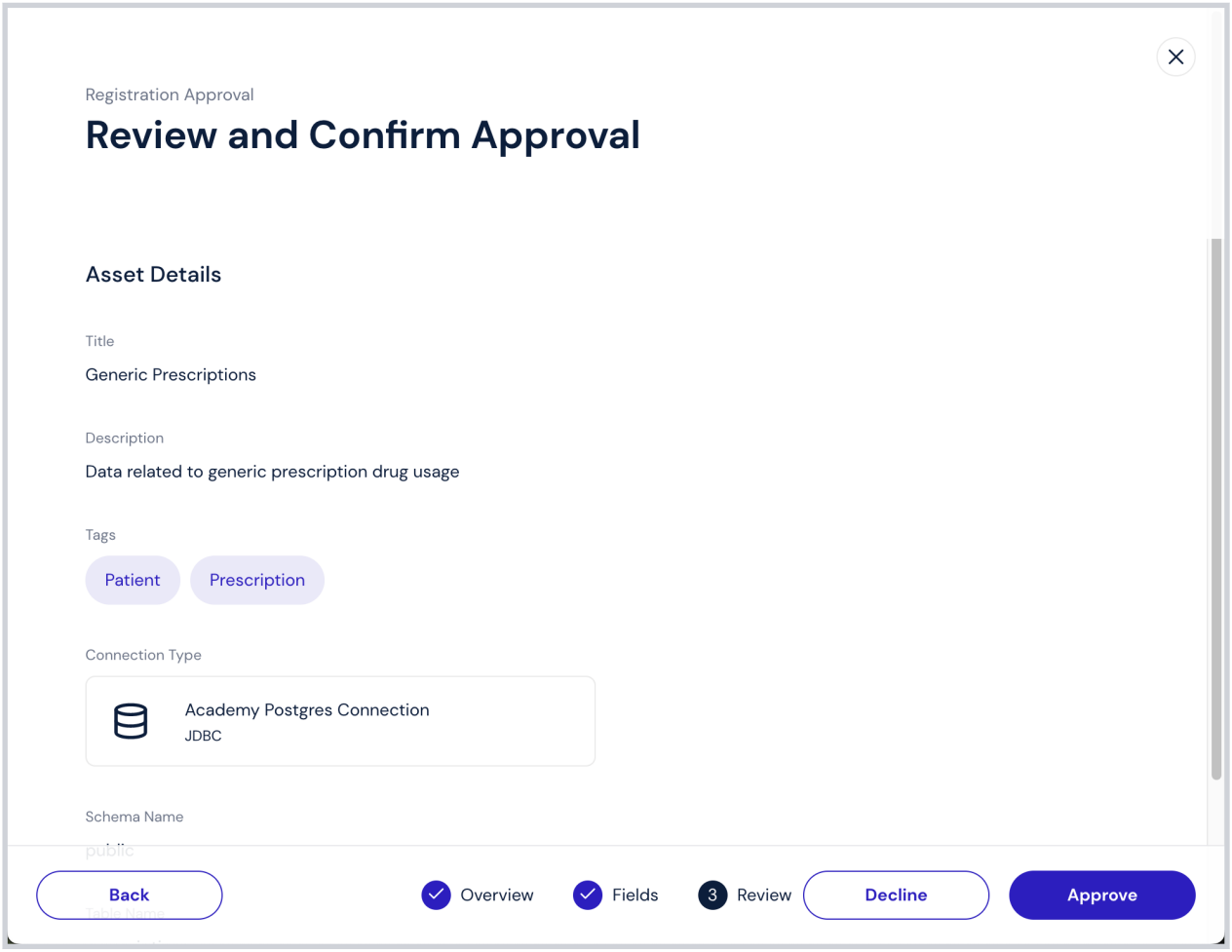
   The Asset Registration Request page appears.
4. Review the request details.
5. Click **Next**.

   The Review Fields page appears.
6. Review the data classes and tags assigned to each field.
7. Click the link next to a field to view details about that field.
8. Click **Next**.

   The Review and Confirm Approval page appears.
9. Review the asset details.
10. To accept the registration request, click **Approve**. The asset's status changes from In Review to Live.

    To reject the registration request, click **Decline**.

    The Tasks page appears.



To learn more about reviewing asset registration requests in bulk, see Approve Asset Registration Tasks.

4.3.4. Edit Multiple Fields

To edit multiple fields simultaneously:

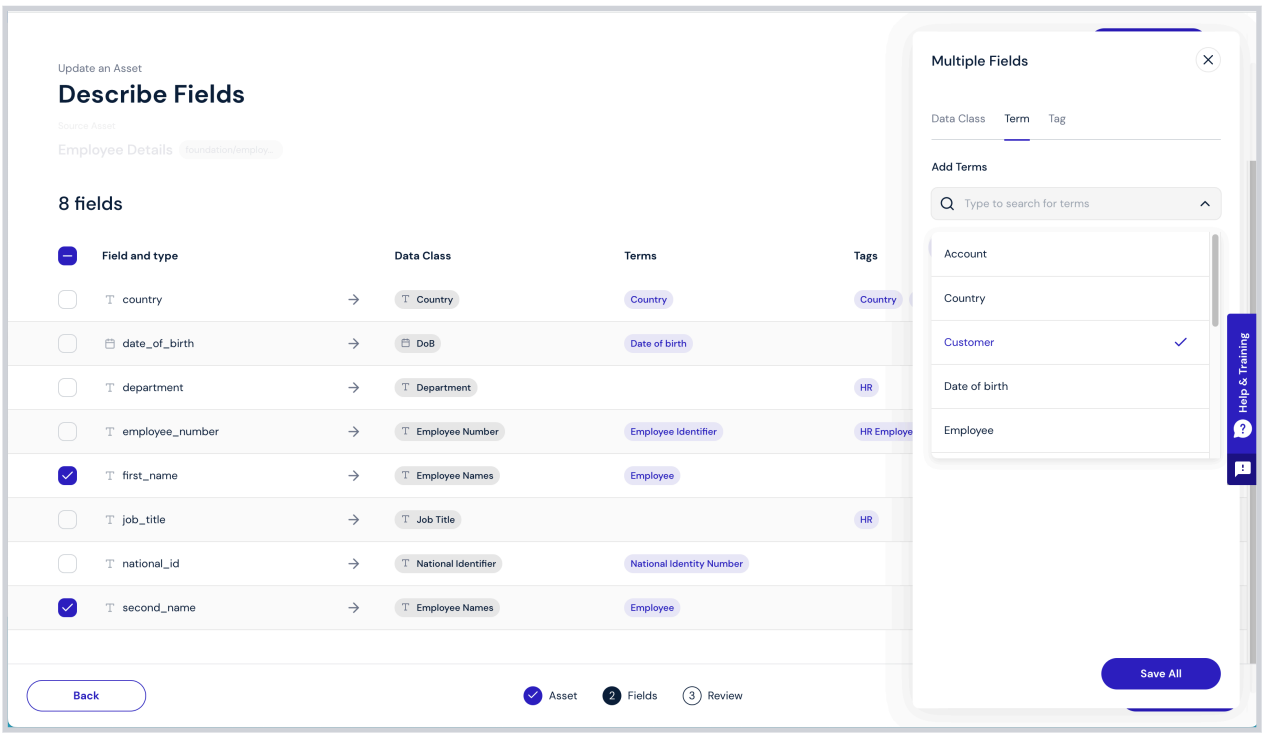
1. Check any combination of fields in the **Field and type** column.

   Alternatively, check the box above the list of fields next to **Field and type**.

   This selects all fields. You can now deselect any fields.
2. Click **Edit [n] Fields**.

   The Multiple Fields window appears on the right.
3. Click each tab to add that metadata type.
4. Search for and select each term, tag, or other metadata.
5. Click **Save All**.

   The new metadata attributes appear in the list of fields for each field that you selected.


Update an Asset
Describe Fields
Employee Details
8 fields
Field and type
Data Class
Terms
Tags
country
Country
Country
Country
date_of_birth
DoB
Date of birth
department
Department
HR
employee_number
Employee Number
Employee Identifier
HR Employe
first_name
Employee Names
Employee
job_title
Job Title
HR
national_id
National Identifier
National Identity Number
second_name
Employee Names
Employee
Back
Asset
2 Fields
3 Review
Multiple Fields
Data Class
Term
Tag
Add Terms
Type to search for terms
Account
Country
Customer
Date of birth
Employee
Save All
Help & Training

5. Accessing Data

5.1. Consumption Projects

A consumption project is a collection of data assets that a team of data consumers wishes to provision safely. While data owners manage the data assets themselves, data consumers manage consumption projects, including linkability between assets. However, data consumers will not have access to the data within a consumption project until a data guardian approves their access.

As a data consumer, you can create a consumption project to request access to data.

5.1.1. Best Practices for Consumption Projects

Although consumption projects do not expire by default, your organization should take full advantage of the expiration feature by creating a standard practice of setting consumption projects to expire after a specified time frame.

Your organization should codify standard collaboration mode settings for consumption projects.

Where necessary, your organization should also create a defined set of consumption project members per project to control and restrict access to data according to your organization's needs.

5.1.2. Create a Consumption Project

1. Click **Projects** in the left navigation.
2. Click **Create** or **Create a Project**.

   The Describe Your Project page appears.
3. **Title—**Enter a title for the consumption project.
4. **Description—**Explain the goal of the consumption project.
5. **Purpose—**Select the consumption project purpose from the list of preapproved purposes. (Required)

   The purpose identifies the intended use of the assets in the consumption project. The purpose is one of the parameters used to match the data consumer's data request to one or more data protection policies and rules that are conditioned on the same purpose.
6. **Collaboration Mode—**Project collaboration mode allows you to define the extent to which the consumption project members can collaborate with each other by linking their query results. Data is linkable when the same values are consistently de-identified across the different datasets.

   Select one of the following options:

   a. **Allow Collaboration between All Members of the Project**

      This option allows all consumption project members to collaborate with any other project member by linking their consistently de-identified data fields.

b. **Restrict Collaboration within User Groups**

This option restricts the collaboration (data linkability) only within individual user groups. Data cannot be linked by members of different groups. (For example, members of Group A cannot link data with members of Group B, only with other members of Group A who are members of the project.) Consumption project members who are not listed in any of the selected user groups cannot collaborate with any other project member.

c. **Prevent Collaboration between the Project Members**

Consumption project members cannot collaborate with other project members by linking their de-identified data fields. Each project member has their unique view of the de-identified data.

7. **Project Members—**Consumption project membership allows you to restrict which users may access the data. Restricting access means that only those specified may access the data.

   Select one of the following options:

   a. **Unrestricted: Any data consumer in organization may access the data**

   This option allows any user with the data consumer role in this data exchange to access the data in this consumption project.

   b. **Restricted: Only project members may access the data**

   This option restricts access to the the data in this consumption project to the project members that you select. If you do not select any members, you are the only person with access to the data in this consumption project.

8. **Add Members—**If you selected **Restricted**, you can select which users and user groups can be members of this consumption project. (If you selected **Unrestricted**, **Add Members** does not appear.)

   a. Click the plus (+) sign.

   The Add Members window appears.

   b. Search for users and user groups.

   c. Click each user or user group to add as project members.

   d. Click **Add**.

9. **Tags—**Select tags to assign to the consumption project.

   Use tags to identify the subject area of the consumption project and to help facilitate search.

> **Note**
>
> If you cannot find a tag that matches your criteria, you can create a new tag. See Create a Tag.

10. **Expiration Date—**Select whether access to data in the consumption project expires and if so, select the expiration date.

    By default consumption projects do not expire.

    If you select an expiration date for this consumption project, it will be inactive after that date, preventing access to the data.

If you want to access data in a consumption project after its expiration date, you can go through another approval process. You do this by editing the inactive consumption project to move it into **In Draft** status. Once you resubmit it, a data guardian can then approve the consumption project, granting you access to the data once again.

11. Click **Create**.

    A confirmation message appears.

Once you have created a consumption project, you can now look for data to add to that consumption project.

5.1.3. See Statuses of Consumption Projects

As a data consumer, you can view the status of consumption projects that you submitted for approval.

1. Click **Projects** in the left navigation.
2. Click each tab to see projects with the status indicated on the tab.

Consumption projects can have the following statuses:

- **In Draft—**A data consumer has not yet submitted the consumption project for approval. A data consumer can edit or delete a project in this status. It remains in "In Draft" status until a data consumer submits for approval.
- **In Review—**A data consumer has submitted the consumption project for approval. A data consumer cannot edit or delete a project in this status. It remains in "In Review" status until a data guardian approves or rejects it.
- **Rejected—**A data guardian has rejected the data access request. A data consumer can edit a rejected consumption project and re-submit it for approval by a data guardian.
- **Published—**A data guardian has approved the data access request. A data consumer can use this consumption project to access data. A data consumer can also edit a published consumption project, but doing so moves the project to "In Draft" status, preventing the data consumer from being able to consume data until a data guardian re-approves the consumption project.

5.1.4. Edit a Consumption Project

As the data consumer who created a consumption project, you can edit or delete that project when it is in any status (Published, In Draft, In Review, or Rejected).

Important

Regardless of whether a consumption project has restricted or unrestricted membership, only an exchange administrator or the data consumer who created the consumption project may edit it, delete it, add assets to it, or submit it for approval.

1. Click **Projects** in the left navigation.

2. Click the consumption project that you want to edit.
3. Click **More** (the three vertical dots).
4. Click **Edit Details**.
5. Follow the steps in Create a Consumption Project.

5.1.5. Delete a Consumption Project

As the data consumer who created a consumption project, you can edit or delete that project when it is in any status (Published, In Draft, In Review, or Rejected).

> **Important**
>
> Regardless of whether a consumption project has restricted or unrestricted membership, only an exchange administrator or the data consumer who created the consumption project may edit it, delete it, add assets to it, or submit it for approval.

1. Click **Projects** in the left navigation.
2. Click the consumption project that you want to delete.
3. Click **More** (the three vertical dots).
4. Click **Delete Project**.

   A confirmation page appears.
5. Check the **I understand and would like to proceed** box, and click **Delete**.

   A confirmation message appears.

5.2. Search for Data to Consume

As a data consumer, you can search for data to add it to a consumption project, which you will later submit for access approval.

1. Click **Data Exchange** in the left navigation.
2. Review the list of datasets.

   > **Note**
   >
   > The list only contains datasets of which you are a member.
3. To search for a dataset, enter search text in the search box.

   > **Tip**
   >
   > You can click the filter icon and select a tag to see all datasets with that tag. You can then search within those datasets.

4. Click a dataset's title to select it.

   A list of the assets within that dataset appears.
5. Click an asset to review its details.
6. Click **Add to Project**.
7. Select the consumption project to which you want to add this asset.

   > **Note**
   >
   > As a data consumer, you may only add data to consumption projects that you created. An exchange administrator may add data to any project.

8. Repeat these steps to add more assets to the consumption project.

   > **Note**
   >
   > You can add assets from different datasets.

   Next, you'll return to the consumption project and submit it for approval.

5.3. Submit a Request for Data

As the data consumer who created a consumption project and added data to that project, you can submit a request to access the assets.

> **Important**
>
> Regardless of whether a consumption project has restricted or unrestricted membership, only an exchange administrator or the data consumer who created the consumption project may edit it, delete it, add assets to it, or submit it for approval.

1. Click **Projects** in the left navigation.

   Consumption projects can have the following statuses:

   - **In Draft—**A data consumer has not yet submitted the consumption project for approval. A data consumer can edit or delete a project in this status. It remains in "In Draft" status until a data consumer submits for approval.
   - **In Review—**A data consumer has submitted the consumption project for approval. A data consumer cannot edit or delete a project in this status. It remains in "In Review" status until a data guardian approves or rejects it.
   - **Rejected—**A data guardian has rejected the data access request. A data consumer can edit a rejected consumption project and re-submit it for approval by a data guardian.
   - **Published—**A data guardian has approved the data access request. A data consumer can use this consumption project to access data. A data consumer can

also edit a published consumption project, but doing so moves the project to "In Draft" status, preventing the data consumer from being able to consume data until a data guardian re-approves the consumption project.

2. Click the **Drafts** tab.

   A list of consumption projects not yet submitted for approval appears.
3. Click the title of the consumption project that contains the assets to which you want access.
4. Click **Submit for Approval**.
5. Click **OK**.

   The project's status changes to "In Review." You cannot edit or delete a project in this status. It remains in "In Review" status until a data guardian approves or rejects it.
6. To see the status of your request, click **Projects** in the left navigation.
7. Click the **In Review** tab to view a list of consumption projects that data guardians are reviewing, including the project that you just created.

   To learn more about this review process, see Approve Project Request Tasks.

Once a data guardian approves a project, it appears on the **Published** tab.

If a data guardian rejects a project, it appears on the **Rejected** tab.

5.4. Access Data

As a data consumer, you can access data external to the Privitar Data Security Platform through a query from a SQL query or business intelligence (BI) tool using a custom connection.

> **Important**
>
> Only an exchange administrator, the data consumer who created the consumption project, or a member of the project may access the data in that project.

Make sure that you have a copy of the supported data proxy JDBC driver installed in your SQL query or business intelligence (BI) tool for the version of the Privitar Data Security Platform that you are using.

1. Click **Projects** in the left navigation.
2. Click the title of a project on the **Published** tab.
3. Click the **Assets and Access** tab.
4. Click **Copy Proxy URL** to copy the address of the proxy to your computer's clipboard.
5. Launch your SQL query or business intelligence (BI) tool of choice.

   Examples of SQL query tools include Microsoft SQL Server, dbForge, and RazorSQL. Examples of BI tools include Tableau® and SAS® Visual Analytics.
6. Refer to your SQL query or BI tool's documentation for creating a custom data driver for connecting to data. You will need to supply the following information:

a. The path to the location of the data driver.
b. Your Privitar Data Security Platform user credentials (the username and password that you use to log in to the platform).
c. The driver class name, which is: `com.privitar.dataplane.integrations.dynamic.jdbc.DataProxyDriver`
d. The proxy URL that you copied in the previous steps.

Ensure that when creating the custom data connection, the parameter's TLS (sometimes called "SSL") flag is set to True in your tool.

7. Using the supplied data proxy driver, use the proxy URL to connect to the data assets in the project and to retrieve de-identified data based on the policies and rules defined on the platform.

Note

If there is an error when authenticating a connection, the error message may not contain enough information to indicate the exact cause of the problem. Please validate the connection URL, credentials, and TrustStore certificate are correct.

Note

Some data source drivers don't provide a default schema. For these, you need to fully qualify your table (`schema.table`) in order to query it.

5.5. Troubleshooting SQL Queries

This table attempts to assist you when you experience issues querying data. To learn about any release-specific issues with querying data, see "Known Issues and Limitations" in the DSP Platform Release Notes.

Table 5. Troubleshooting SQL Queries

Issue	Cause
No data returned with error	If an asset you are connecting to has multiple fields with the same assigned data class, it's possible that the platform will not know which one to evaluate. It will then generate an error and stop provisioning.
No data returned without error	It is possible that you are attempting to query data through a consumption project to which you do not have authorized access. Alternatively, if the default policy is drop field, it might be tha no policies or rules apply to the data that you are querying.
Error message references unsupported data types	Your query contains unsupported data types.

Issue	Cause
Error message stating object <table name> not found	Possible causes include: • Some drivers don't provide a default schema • Table does not exist on the schema that you are querying • Incorrect capitalization for databases that are case sensiti
This error message appears: "Remote driver error: AvaticaRuntimeException: [Messsage: 'Proxy is not configured to accept correlationIds. Messsage: 'Proxy is not configured to accept correlationIds. `privitar.jdbcproxy.allowCorrelationId` must be set to true or set to empty correlationID JDBC property on the driver', Error code: '400', SQL State: '0000', Severity: 'ERROR']. Error 500 (00000) ERROR `allowCorrelationId` must be set to true or set to empty correlationID JDBC property on the driver', Error code: '400', SQL State: '0000', Severity: 'ERROR']. Error 500 (00000) ERROR"	This message appears in the following scenario: • JDBC Driver property `correlationId={string}` • The data proxy is set with `privitar.jdbcproxy.allowCorrelationId=fal`

6. Migrating Data

When migrating data, users in your organization will perform the following tasks:

1. A data owner registers raw assets with the platform.

   > **Note**
   >
   > Our recommended best practice is to create a separate dataset for the raw assets and mark it as restricted.

2. A data guardian approves the raw assets.
3. The data owner creates and submits a migration project.
4. A data guardian approves the migration project, and this automatically registers the corresponding masked assets to the data exchange.
5. The data owner reviews and submits masked assets.
6. A data guardian approves the masked assets.
7. A data integration developer executes a data migration pipeline, which references the migration project. This process completes the data migration and transformation of raw assets to masked assets, allowing data consumers to request and access the data.
8. A data consumer creates a consumption project to request access to the masked assets.
9. A data guardian approves the consumption project.
10. The data consumer copies the proxy URL to use in their business intelligence (BI) tool, for example.

6.1. Migration Projects

A migration project is a collection of raw data assets that a data owner wishes to mask and move to cloud storage or cross jurisdiction. A data consumer with appropriate authorization may consume the masked assets from the new location where, according to policy, masked data may be reversed.

The Privitar Data Security Platform (DSP) provides a platform for creating reusable policies for the protection and application of privacy enhancing technologies. Most importantly, the platform affords access to the data, including unmasking (sometimes called "re-identification" or "re-ID") of masked data when appropriate.

6.1.1. Best Practices for Migration Projects

As a data owner, when you register raw assets with the platform, our recommended best practices include:

- Create a separate dataset for the raw assets.
- Mark datasets that contain raw assets as restricted. Mark datasets that contain the masked assets as unrestricted and made available for data consumers to request for use in a consumption project.

- Use a different connection for the raw and masked assets.

> **Note**
>
> You may not reuse a connection that you previously used with another migration project.

- The platform ensures that you only select and include raw assets within a migration project. Masked assets cannot be masked again, nor can they be added to a migration project.
- Use only consistent numeric or text regular expression transformation types for data that you wish to allow a data consumer to reverse to its original value, according to their request and applicable policy. Other transformation types (such as Constant Text Value or Date Generalization) will not allow data to be reversed. The platform will return other transformation types in their transformed form, rather than the original values. For example, if Constant Text Value is the applied transformation type, the platform will return the constant text.

6.1.2. Create a Migration Project

As a data owner, you can create a migration project to mask and move raw data assets.

1. Click **Projects** in the left navigation.
2. Click **Create** or **Create a Project**.

   The Describe Your Project page appears.
3. **Title—**Enter a title for the project.
4. **Description—**Explain the goal of the project.
5. **Purpose—**Select the project purpose from the list of preapproved purposes. (Required)

   The purpose identifies the intended use of the assets in the project. The purpose is one of the parameters used to match the data migration request.
6. **Destination—**Indicate the connection and dataset that the platform will use when it registers the masked (target) assets.
   a. **Destination Connection—**Select a connection from the list of available connections.
   b. **Destination Dataset—**Select a dataset from the list of available datasets.

> **Important**
>
> The dataset must either be unrestricted, or you must be the owner or a member of the dataset.

7. **Expiration Date—**Select whether migration of the data in the project expires and if so, select the expiration date.

   By default, projects do not expire.

   If you select an expiration date for this project, it will be inactive after that date, preventing migration of the data.

8. **Tags—**Select tags to assign to the project.

   Use tags to identify the subject area of the project and to help facilitate search.

   > **Note**
   >
   > If you cannot find a tag that matches your criteria, you can create a new tag. See Create a Tag.

9. Click **Create**.

   A confirmation message appears.

10. Click **Projects** in the left navigation.
11. Click the **Draft** tab.

    A list of projects in Draft status appears.

12. Click the name of the project that you just created.

    The Project Details page appears.

13. Add raw assets to the project.
14. Submit the project for approval by a data guardian.

> **Note**
>
> Once you submit a project for approval, you cannot edit it.

6.1.3. See Statuses of Migration Projects

As a data owner, you can view the status of migration projects that you submitted for approval.

1. Click **Projects** in the left navigation.
2. Click each tab to see projects with the status indicated on the tab.

Migration projects can have the following statuses:

- **In Draft—**A data owner has not yet submitted the migration project for approval. The data owner who created the project can edit or delete that project in this status. It remains in "In Draft" status until a data owner submits it for approval.
- **In Review—**A data owner has submitted the migration project for approval. A data owner cannot edit or delete a project in this status. It remains in "In Review" status until a data guardian approves or rejects it.
- **Rejected—**A data guardian has rejected the data migration request. A data owner can edit a rejected migration project and re-submit it for approval by a data guardian. Alternatively, a data owner can delete a rejected migration project.
- **Published—**A data guardian has approved the data migration request. A data integration developer can use this migration project to move data. A data owner cannot edit or delete a published migration project.

6.1.4. Delete a Migration Project

As the data owner who created a migration project, you can edit or delete that project when it is in Draft or Rejected status. You cannot edit or delete a Published or In-Review project.

1. Click **Projects** in the left navigation.
2. Click the project that you want to delete.
3. Click **More** (the three vertical dots).
4. Click **Delete Project**.

   A confirmation page appears.
5. Check the **I understand and would like to proceed** box, and click **Delete**.

   A confirmation message appears.

6.2. Search for Data to Migrate

As a data owner, you can search for data to add it to a migration project, which you will later submit for approval.

1. Click **Data Exchange** in the left navigation.
2. Review the list of datasets.

   Note

   The list only contains the following types of datasets:

   - unrestricted datasets
   - datasets of which you are a member
   - datasets that you created
3. To search for a dataset, enter search text in the search box.

   Tip

   You can click the filter icon and select a tag to see all datasets with that tag. You can then search within those datasets.
4. Click a dataset's title to select it.

   A list of the assets within that dataset appears.
5. Click an asset to review its details.
6. Click **Add to Project**.
7. Select the migration project to which you want to add this asset.

> **Note**
>
> As a data owner, you may only add data to migration projects in In Draft status that you created. An exchange administrator may add data to any project.

8. Repeat these steps to add more assets to the migration project.

> **Note**
>
> You can add assets from different datasets, but you may not add already masked assets.

Next, you'll return to the migration project and submit it for approval.

6.3. Submit a Request for Data to Migrate

As the data owner who created a migration project and added data to that project, you can submit a request to access the assets.

> **Important**
>
> Only an exchange administrator or the data owner who created the migration project may edit it, delete it, add assets to it, or submit it for approval.

1. Click **Projects** in the left navigation.
2. Click the **Drafts** tab.

   A list of projects not yet submitted for approval appears.
3. Click the title of the migration project that contains the assets you want to migrate.
4. Click **Submit for Approval**.
5. Click **OK**.

   The project's status changes to "In Review." You cannot edit or delete a project in this status. It remains in "In Review" status until a data guardian approves or rejects it.
6. To see the status of your request, click **Projects** in the left navigation.
7. Click the **In Review** tab to view a list of migration projects that data guardians are reviewing, including the project that you just created.

   To learn more about this review process, see Approve Project Request Tasks.

If a data guardian rejects a project, it appears on the **Rejected** tab.

Once a data guardian approves a project, it appears on the **Published** tab. This process automatically adds a draft of the masked assets to the dataset you indicated when you created the project.

1. Return to the data exchange.
2. Select the dataset containing the draft masked assets.
3. Review and update the asset's details and classifications and submit it for approval by a data guardian.

   For reference, see Describe and Register an Asset.

If the platform cannot automatically add a draft of the masked assets to the dataset, an error appears on the project page. Go to the **Published** tab and click on the project. If an alert appears, click **Add Assets**.

7. Approving Requests

Data guardians perform the following tasks in the Privitar Data Security Platform:

- Approving asset registration requests made by data owners

  An asset registration request is an inquiry made by a data owner to add a data asset (a database table, for example) to a dataset. A data guardian approves or denies asset registration requests.

  The data guardian ensures that the data owner properly classified the asset and that there are appropriate rules for the protection of the data asset.
- Approving project requests made by data consumers

  A project request is an inquiry made either by a data consumer to use the assets in a data consumption project or by a data owner to create a migration project. Data guardians approve or deny all project requests.

  The data guardian ensures that the fields in each asset within a project are properly classified and that there are appropriate rules for the protection of the data assets used by the project.

7.1. Approve Asset Registration Tasks

As a data guardian, you review and approve asset registration requests submitted by data owners. You ensure that the data owner properly classified the asset and that there are appropriate rules for the protection of the data asset.

To approve an *asset registration request*:

1. Click **Tasks** in the left navigation.
2. Click **Asset Requests**.
3. Click the name of an asset.

   The Asset and Dataset Information page appears.
4. Click **Next**.
5. Review the data classification and tags that the data owner associated with each field.
6. Click **Next**.
7. Click **Approve** or **Decline**.

   A confirmation message appears.

7.2. Approve Policy Tasks

As a data guardian or a platform administrator, you can review and approve a policy task submitted by another data guardian if all of the following are true:

- You did not submit the policy for approval.
- You did not create the policy or any of its rules.
- You did not modify the policy or any of its rules.

All data guardians who contributed to a policy or any of its rules are not be able to approve or decline a request related to that policy.

To approve a policy task:

1. Click **Tasks** in the left navigation.
2. Click the **Policy Requests** tab.
3. Click the name of a policy.

   The Policy Information page appears.
4. Click **Next**.
5. Review the policy that the data guardian requested to create or modify.
6. Click **Approve** or **Decline**.

   A confirmation message appears.

> **Important**
>
> Only the data guardian who submitted the policy for approval may view and act on a declined policy that is on the **Rejected** tab, including editing it or deleting the request.

7.3. See Statuses of Policy Tasks

As a data guardian, you can view the status of a policy that you submitted for approval.

1. Click **Tasks** in the left navigation.
2. Click the **Policies** tab.

   A list of policies that you have submitted for approval appears.

Policies can have the following statuses:

- **In Draft—**A data guardian has not yet submitted the policy for approval.
- **In Review—**A data guardian has submitted the policy for approval.
- **Rejected—**A data guardian has rejected the policy.
- **Published—**A data guardian has approved the policy.

7.4. Approve Project Request Tasks

As a data guardian, you review and approve project requests submitted by data consumers and data owners.

To approve a *project request*:

1. Click **Tasks** in the left navigation.
2. Click **Project Requests**.
3. Click the name of a project.

   The Project Information page appears.
4. Click **Next**.
5. Review which data assets the data consumer or data owner requested as part of the project.

6. Click **Approve** or **Decline**.

   A confirmation message appears.

8. Viewing Audit Logs

As a data guardian, you can view audit logs in the platform user interface (UI) to see details of logged events, such as policy resolutions.

8.1. View Audit Logs

1. Click Audit Log in the left navigation.

   A list of audit events appears, including the following types:

   - authentication events for policy resolution
   - policy resolution events (See View Policy Resolution Audit Logs .)

   > **Note**
   >
   > Policy resolution events record whether or not the policies were applied to the query. They do not record the final state of the query. For the latter, review the query log from your query tool.

   The list of audit events appears in date order from most recent to oldest.
2. Enter search criteria in the Search box.

   You can enter any text to search for member name or correlation ID.
3. Click the Filter icon.

   The Refine Results window appears.

   a. Select specific criteria within available categories, specifically:

      - member (not case sensitive)
      - date range
4. Click an event to view its details.

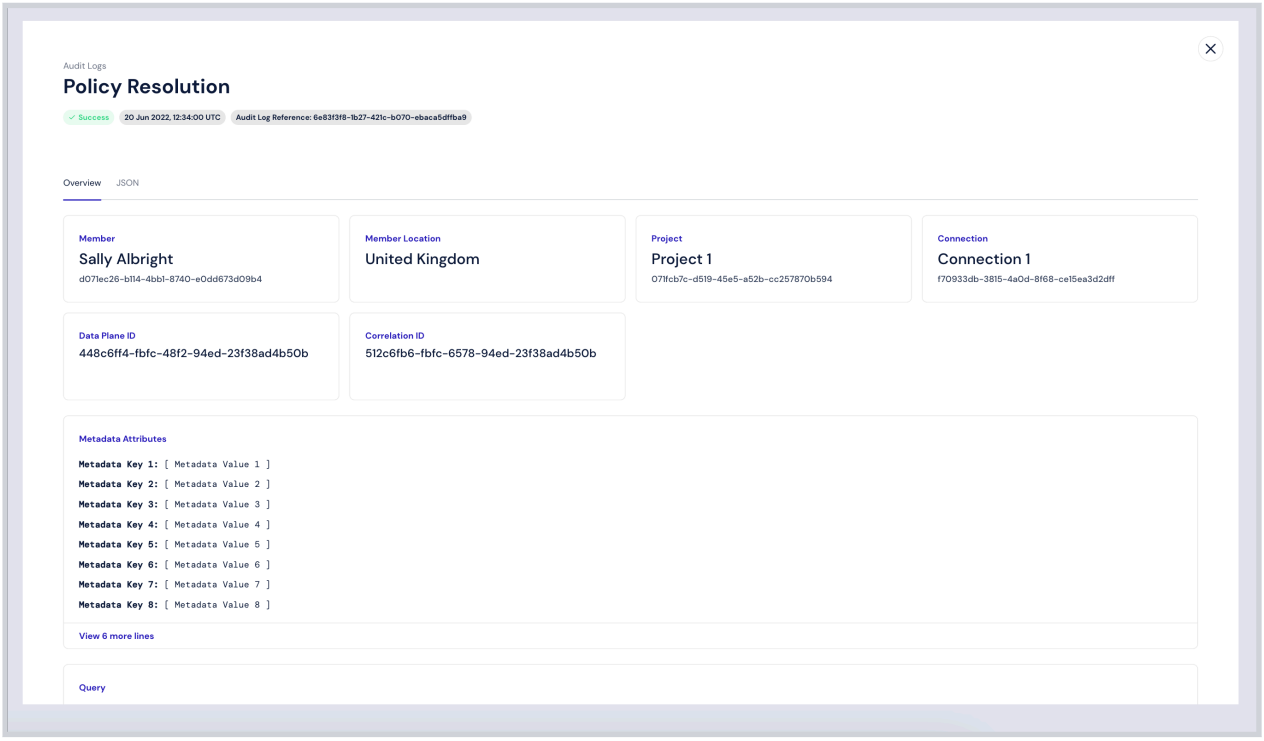
   The **Overview** tab appears, showing some high-level information about the event.
5. Click the **JSON** tab to view the JavaScript Object Notation (JSON) version of that event's details.

8.1.1. View Policy Resolution Audit Logs

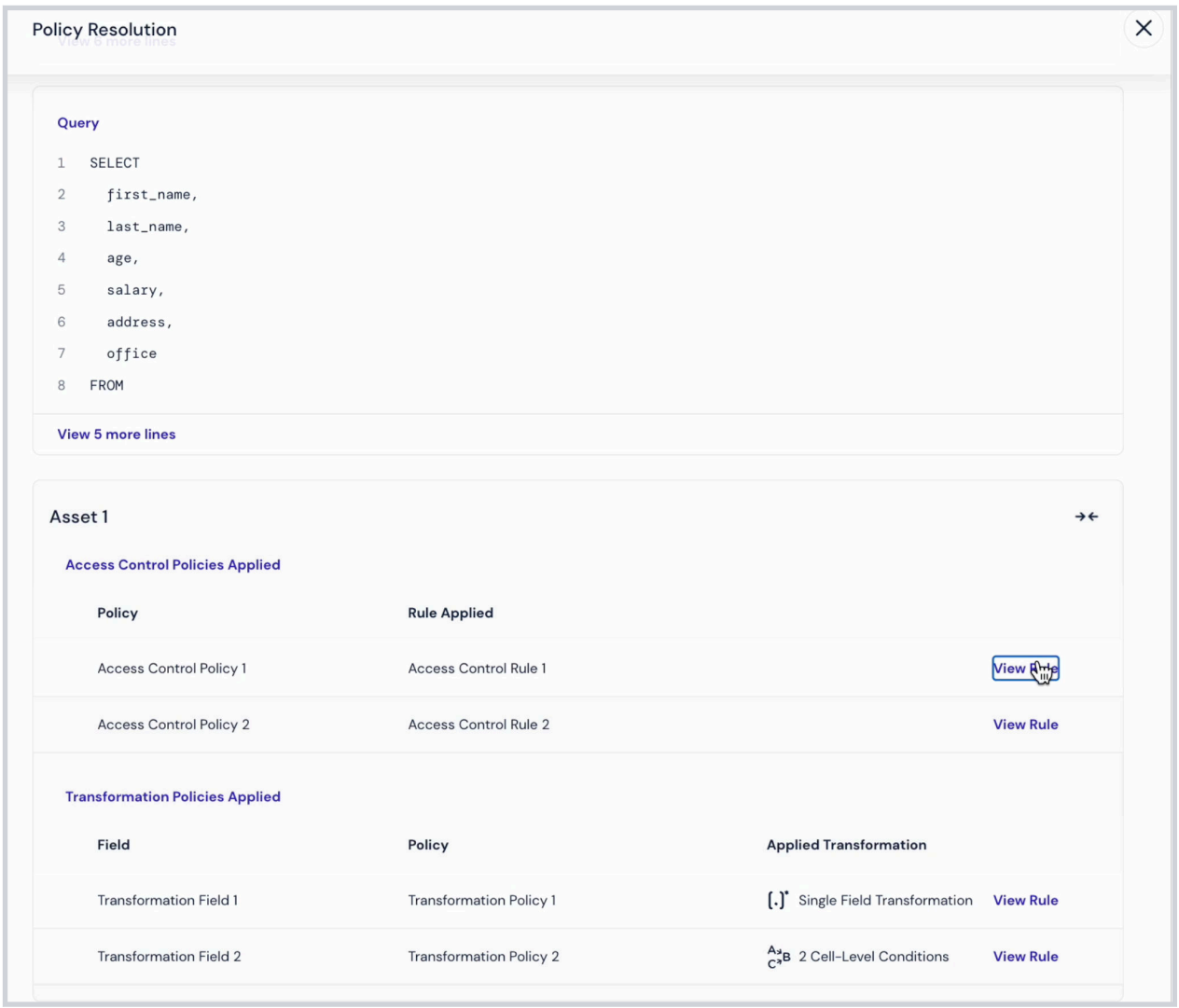
The Policy Resolution Detail page includes an Overview tab and a JSON tab.

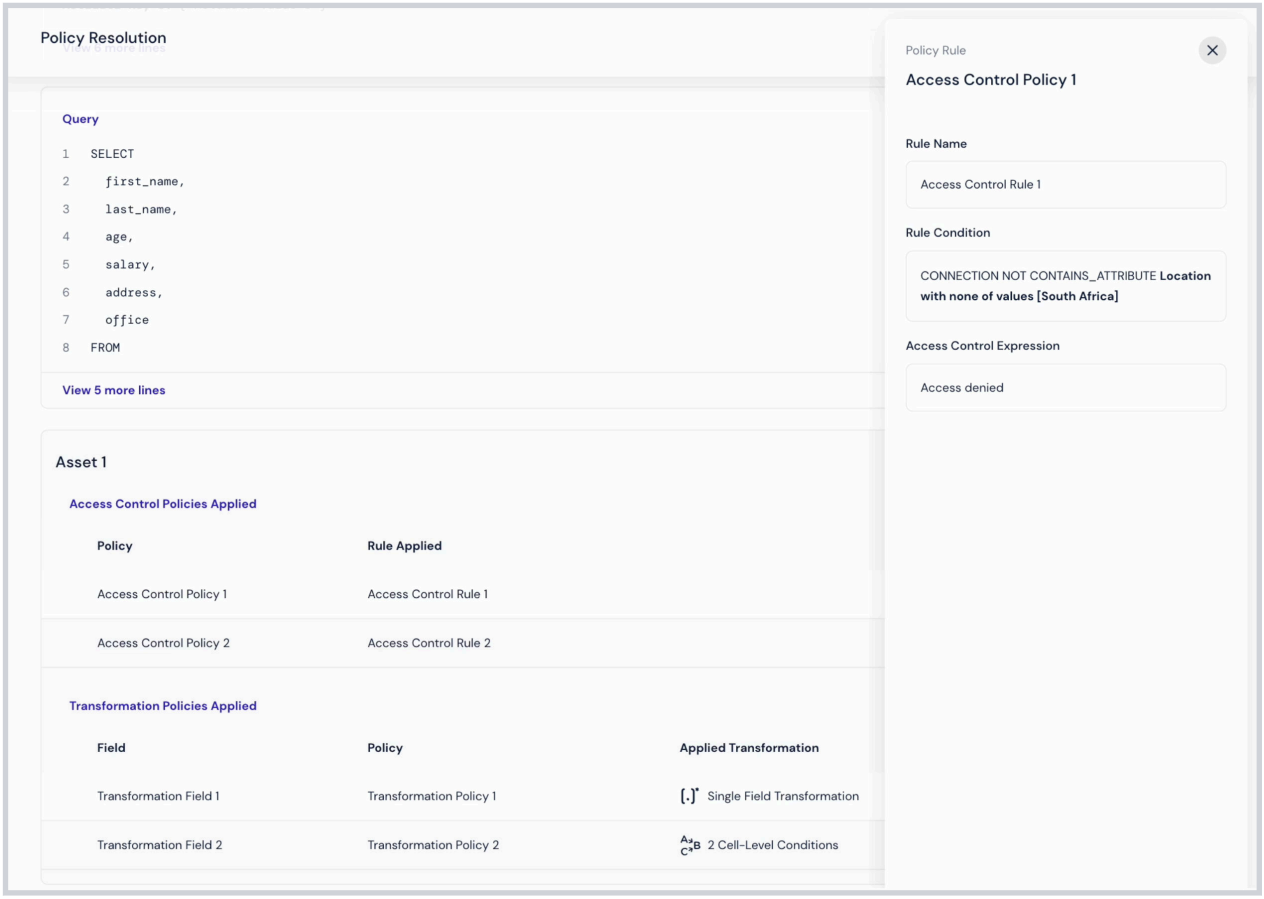
The Overview tab includes stylized information, including:

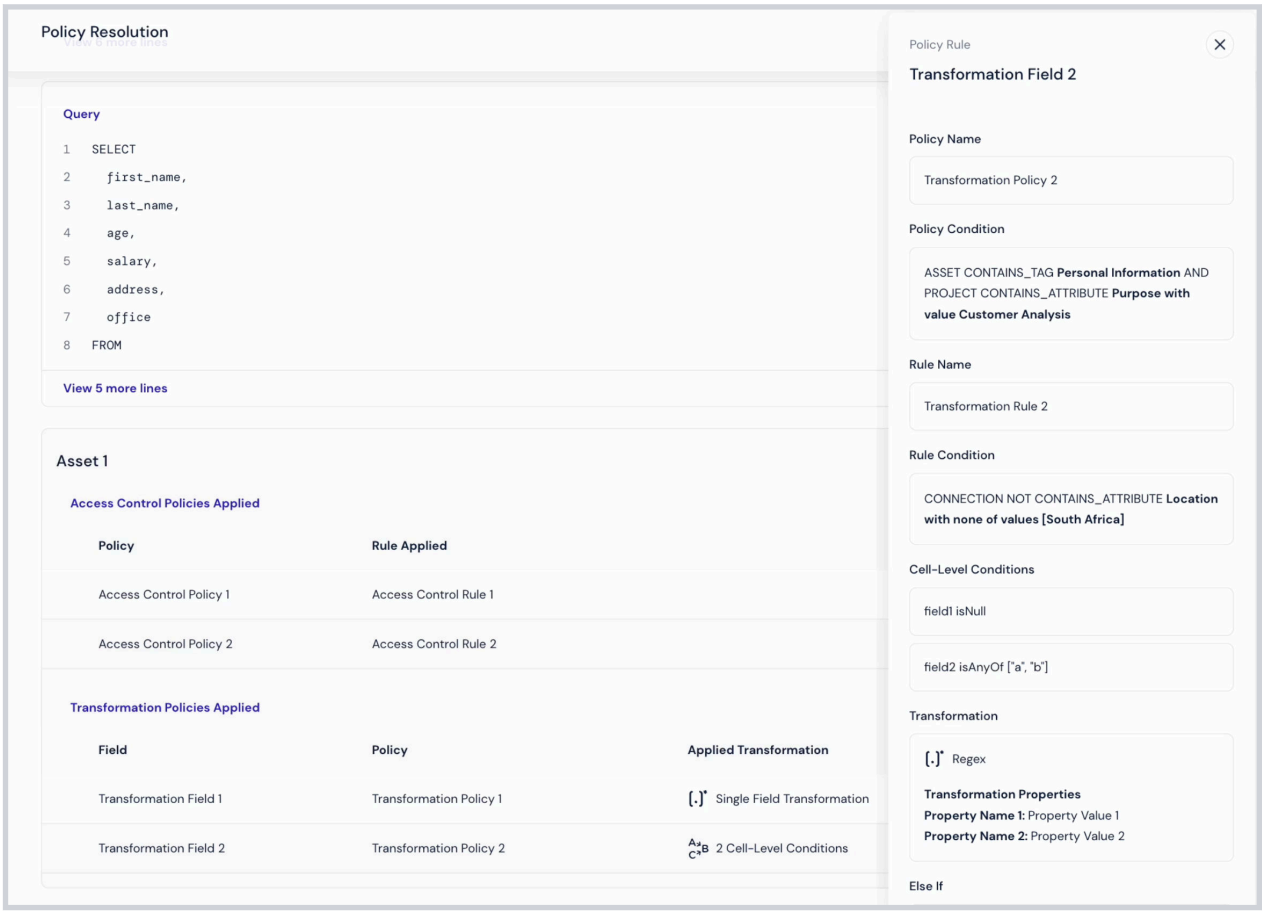
- header content
- metadata context, if available
- the query that triggered the policy
- the queried assets
- the policies that were triggered
- the filters, controls, and transformations that were applied



1. For each asset, click **View Rule** to view details about the applied rule, including:
   - policy name
   - policy trigger condition, if applicable
   - rule name
   - rule condition
   - transformation applied


Policy Resolution
Query
1 SELECT
2 first_name,
3 last_name,
4 age,
5 salary,
6 address,
7 office
8 FROM
View 5 more lines
Asset 1
Access Control Policies Applied
Policy
Rule Applied
Access Control Policy 1
Access Control Rule 1
View Rule
Access Control Policy 2
Access Control Rule 2
View Rule
Transformation Policies Applied
Field
Policy
Applied Transformation
Transformation Field 1
Transformation Policy 1
Single Field Transformation
View Rule
Transformation Field 2
Transformation Policy 2
2 Cell-Level Conditions
View Rule


Policy Resolution
Query
1 SELECT
2 first_name,
3 last_name,
4 age,
5 salary,
6 address,
7 office
8 FROM
View 5 more lines
Asset 1
Access Control Policies Applied
Policy
Rule Applied
Access Control Policy 1
Access Control Rule 1
Access Control Policy 2
Access Control Rule 2
Transformation Policies Applied
Field
Policy
Applied Transformation
Transformation Field 1
Transformation Policy 1
Single Field Transformation
Transformation Field 2
Transformation Policy 2
2 Cell-Level Conditions
Policy Rule
Access Control Policy 1
Rule Name
Access Control Rule 1
Rule Condition
CONNECTION NOT CONTAINS_ATTRIBUTE Location with none of values [South Africa]
Access Control Expression
Access denied


Policy Resolution
Query
1 SELECT
2 first_name,
3 last_name,
4 age,
5 salary,
6 address,
7 office
8 FROM
View 5 more lines
Asset 1
Access Control Policies Applied
Policy
Rule Applied
Access Control Policy 1
Access Control Rule 1
Access Control Policy 2
Access Control Rule 2
Transformation Policies Applied
Field
Policy
Applied Transformation
Transformation Field 1
Transformation Policy 1
Single Field Transformation
Transformation Field 2
Transformation Policy 2
2 Cell-Level Conditions
Policy Rule
Transformation Field 2
Policy Name
Transformation Policy 2
Policy Condition
ASSET CONTAINS_TAG Personal Information AND PROJECT CONTAINS_ATTRIBUTE Purpose with value Customer Analysis
Rule Name
Transformation Rule 2
Rule Condition
CONNECTION NOT CONTAINS_ATTRIBUTE Location with none of values [South Africa]
Cell-Level Conditions
field1 isNull
field2 isAnyOf ["a", "b"]
Transformation
Regex
Transformation Properties
Property Name 1: Property Value 1
Property Name 2: Property Value 2
Else If

2. Click the **JSON** tab to view the JavaScript Object Notation (JSON) version of that event's details.
    a. Click the **View** (eye) icon to toggle between:
        - a tree view in which you can expand and collapse sections of the tree
        - a fully expanded code view

9. Glossary of Data Security Terminology

This glossary defines terms that relate to the Privitar Data Security Platform.

A

access control policy	An access control policy is a reusable set of access control rules that serves a business context. An access control policy is a flexible construct that allows you to apply access control rules according to desired conditions. For example, you can write access control policies to define rules that examine and drop rows (records) according to the business condition and the actual data in those records.
access control rule	Access control rules act on the field level. Access control rules examine the actual data and discard each record being queried (requested) according to the rule's conditions.
access request	See project request.
asset	Assets are data structures; for example the tables in an Oracle® or PostgreSQL database.
asset registration request	An asset registration request is an inquiry made by a data owner to add a data asset (a database table, for example) to a dataset. A data guardian approves or denies asset registration requests.
attribute-based access control (ABAC)	Attribute-based access controls (ABACs) are conditional policies and rules that regulate how users' access fields or rows, based on specific attributes, such as location, terms, and tags. ABACs determine how the platform applies policies and rules. In contrast, field-level access controls and record-level access controls determine where (on which assets, rows, or fields) the platform applies the policies and rules.

B

business information	Business information provides definition, structure, and clarity to data assets, *consumption projects*, *policies*, and *rules* by representing the context and semantics of an organization. Business information includes *data classes*, *tags*, *terms*, and *purpose*. Business information assists users to find and understand content on the platform and guides when to apply transformations based on attributes and conditions.

C

cell-level transformation	Cell-level transformations allow you to select a different transformation for each distinct record of a specified field (column), that is, a cell, based on varying (logical) conditions. For example, you can instruct the platform to apply different transformations to an identity number or postal code in a given record based on the value of country of residence in a specific cell.
connection	A connection is a configuration for connecting to and reading data from a data source, such as a JDBC connection string. The platform uses this connection information to read metadata attributes from a data asset, to read the data itself, and to write the processed data to the target location.
control plane	The control plane is a logical perimeter that does not have direct access to data but may host components that drive operations in the data plane. The control plane is where policies, rules, projects, and assets are created and managed. The architectural split between the control plane and the data plane allows for configuration, orchestration, and administration (control) without the need to access data, but the ability to process data close to the source within a given jurisdiction. The control plane allows for this by using metadata, data classes, and other representations of the data.

D

data agent	The data agent provides access to the data plane whenever required by the control plane, for example to retrieve the schema for a data asset. It makes a long-lived connection to the *data bridge* on startup.
data bridge	The data bridge is the component in the control plane that handles communication with the data plane. It acts as a Google Remote Procedure Call (gRPC) server. It is replicated, and it sits behind an ingress with a load-balancer.
data class (class)	A data class is a categorization that data owners apply to fields within data assets to indicate the category of data. Within the Privitar Data Security Platform, data owners can apply a data class to identify the data's category and ensure that that kind of data is classified consistently throughout your organization. For example, data classes can classify birth dates, national identifiers, and postal codes.

data consumer (consumer)	Data consumers are users on the Privitar Data Security Platform who request and consume data from the platform. Data consumers require direct access to data as part of their job responsibilities.
data exchange (exchange)	A data exchange is a secure online portal where data owners can classify sensitive datasets, and data consumers can access them, without compromising data safety. Each data exchange is separate and different from other data exchanges, being a discrete entity within an enterprise.
data guardian (guardian)	Data guardians are users on the Privitar Data Security Platform who develop and maintain company policies and rules that govern data usage, including how the organization adheres to regulatory and compliance guidelines and requirements. Data guardians are responsible for approving all data requests, including requests to register data on the platform and requests to access data outside the platform.
data owner (owner)	Data owners are users on the Privitar Data Security Platform who register and classify data on the platform. Data owners understand where the data comes from, its quality, its meaning, and for what purposes it can be used.
data plane	A data plane is a set of services used for the reading, writing, and processing of data. It contains a data agent and services capable of provisioning data, such as a data proxy or an integration using the Privitar SDK.
data proxy (proxy)	The data proxy is a Java Database Connectivity proxy (JDBC proxy) that allows data consumers to access sensitive data to which de-identification policies have been applied. It makes calls to the *data bridge* to fetch the information it needs, for example the details of how to connect to the sensitive data and the policies to be applied.
dataset	A dataset is a logical container of assets that is also known as a "data product." Its purpose is to group and facilitate an easier search experience. Data owners make datasets available to data consumers.
data type (type)	A data type is the data's categorization that is read from the source. Examples include: integer and string. The data type references how data is stored in a database, and each data type can have a different corresponding transformation. For example, you can store a person's age as an integer or a string.

E

encryption	Encryption is the act of using a cryptographic algorithm to derive a value that is applied to a value in a dataset in such a way that only authorized parties can access the original value. In an encryption scheme, the original value, referred to as plaintext, is encrypted using an encryption algorithm to generate ciphertext that authorized parties can only read if it is decrypted. Encryption can be used as a de-identification technique. It is good practice to encrypt data at rest and in transit. However, while encryption can help protect against unauthorized access, it does not protect the privacy of individuals' data when it's used by people who are authorized. This is known as an insider attack.
enterprise administrator (enterprise admin)	Enterprise administrators are users who perform operations within the Privitar Data Security Platform, such as creating a data exchange, creating a data plane, and configuring a data plane.
exchange	See data exchange.
exchange administrator (exchange admin)	Exchange administrators are users who perform tasks within a data exchange, such as creating and editing a data plane, managing users and groups, and performing everyday administration tasks.

F

field-level access control	Field-level access controls are conditional policies and rules that regulate users' ability to access individual fields of a data asset. Field-level access controls determine which fields of the original dataset the platform retrieves prior to applying data transformation rules. Field-level access controls are implemented through drop field transformation, conditioned on attributes (ABAC), data consumer roles (RBAC), or purpose (PBAC). Field-level access controls determine where (on which fields) the platform applies policies and rules.
field-level transformation	Field-level transformations apply the same transformation to the entire field (column). The platform determines whether to apply a field-level transformation based on the data class of the column.

H

HashiCorp® Vault Key Management System (HashiCorp® Vault KMS)	The HashiCorp® Vault KMS is a key management system (KMS) used to create and control encryption keys, which you use to encrypt data. A KMS is a system for the management (generation, distribution, storage, and more) of cryptographic keys and their metadata.

K

key format	The Privitar Data Security Platform uses "asymmetric" (or public key) encryption, which uses a pair of distinct, yet related keys. One key (the public key) is used for encryption, while the other in the pair (the private key) is used for decryption by an authenticated recipient (user).

L

linkability	"Linkability" is the probability of inferring the original value of transformed data by linking values from different datasets. Applying different tokens to the same value in different datasets reduces the ability to re-identify or de-anonymize data.

M

migration project	A migration project is a collection of raw data assets that a data owner wishes to mask and move to cloud storage or cross jurisdiction. A data consumer with appropriate authorization may consume the masked assets from the new location where, according to policy, masked data may be reversed.

P

consumption project	A consumption project is a collection of data assets that a team of data consumers wishes to provision safely. While data owners manage the data assets themselves, data consumers manage consumption projects, including linkability between assets. However, data consumers will not have access to the data within a consumption project until a data guardian approves their access.
policy	A policy is a reusable set of rules that serves a business context. Users of the platform can utilize the following types of policies: • *access control policies*

- *transformation policies*

privacy enhancing technology (PET)	A privacy enhancing technology is a transformation type used to modify raw data to remove sensitive data elements. The Privitar Data Security Platform offers many PETs. These are the transformation types that data guardians select when building policies.
Privitar NOVLT	Privitar NOVLT is a feature of the Privitar Data Security Platform that applies consistent tokenization without a token vault. NOVLT allows for data linkability across regions. NOVLT also offers faster throughput and less latency than most vaulted solutions.
Privitar Query Engine	The Privitar Query Engine retrieves relevant policies and applies them to assets. The Query Engine transforms SQL queries, and the data retrieved with them, in compliance with the applicable policies.
project request (request)	A project request is an inquiry made either by a data consumer to use the assets in a data consumption project or by a data owner to create a migration project. Data guardians approve or deny all project requests.
provision	Provisioning is the act of making data available in a secure way to users and applications.
purpose	A purpose is the data consumer's intended use for the data in a consumption project. Data guardians use purposes as attributes in rules. Examples might include, "to find sources of bad loans" or "to build customer 360 profiles."
purpose-based access control (PBAC)	Purpose-based access controls (PBACs) are conditional policies and rules that regulate how users' access fields, rows, or entire data assets, based on a consumption project purpose selected by a data consumer. PBACs determine how the platform applies policies and rules. In contrast, field-level access controls, and RLACs determine where (on which fields, rows, or assets) the platform applies policies and rules.

R

record-level access control (RLAC)	Record-level access controls (RLACs) are conditional policies and rules that regulate users' ability to access individual records of an asset based on the values of selected fields of the same record. Record-level access controls determine which records of the original dataset the platform retrieves prior to applying transformation rules. Unlike data transformation rules, which are based solely on metadata,

	record-level access control rules are based on a combination of the data itself and metadata. Record-level access controls (RLACs) determine where (on which records) the platform applies policies and rules. Attribute-based access controls (ABACs), purpose-based access controls (PBACs), and role-based access controls (RBACs) determine how the platform applies those policies and rules.
region	1. In the Privitar Data Security Platform, a region is a name for the geographical location, such as the location of a data exchange or a data agent. This is closely tied to jurisdiction. Some regulations require that data must remain within certain jurisdictions. 2. In cloud computing a region, (aka "geography"), is a named set of cloud resources in the same geographical area. A region is comprised of availability zones.
regular expression (regex)	A regular expression is a series of characters that specifies a pattern to match text and numeric data formats. The Privitar Data Security Platform uses regular expressions to replace text strings and numbers with random characters. For example, for an initial value of `abcdef`, you could use the following regular expression `[a-z]{6}` to produce an output such as `mvskyc`.
request	See project request and *asset registration request*.
role-based access control (RBAC)	Role-based access controls (RBACs) are conditional policies and rules that regulate how users access fields or rows, based on specific roles provided as user groups. RBACs determine how the platform applies policies and rules. In contrast, field-level access controls, and record-level access controls determine where (on which fields, rows, or assets) the platform applies policies and rules.
rule	Rules are building blocks of policies. Rules are conditional based on attributes, such as user groups, terms, tags, locations, and so on. Rules also take actions specific to data classes and transformations. Users of the platform can utilize the following types of rules: • *access control rules* • *transformation rules*

S

source connection	A source connection is from where a data owner reads data.
system administrator (SysAdmin)	System administrators are users who perform activities to install and set up the Privitar Data Security Platform. Most of these activities are external to the platform, such as deploying the platform, managing secrets required for installation, performing backup and restore activities, and performing updates to the platform.

T

tag	A tag is a keyword that you can define to describe objects, such as when you want to group objects together or add context to those objects. For example, you might want to define tags that correspond to geography, line of business, project names, or applications. Tags help facilitate search and filtering.
target connection	A target connection is to where a data consumer provisions data.
term	Terms are words used within your organization to describe business concepts in plain language. Adding them to the platform ensures consistent use of those words throughout your organization. Terms also lend meaning to physical assets and their fields and give them context. When data consumers are browsing assets, terms allow them to understand the business meaning and semantics of the physical asset. Examples of terms could be “account type,” “customer level,” or “credit risk rating.”
tokenization	Tokenization is a form of fine-grained data protection that replaces a clear value with a randomly generated synthetic value that stands in for the original as a "token." The pattern for the tokenized value is configurable and can retain the same format as the original, which means fewer down-stream application changes, enhanced data sharing, and more meaningful testing and development with the protected data.
token vault	A token vault is a secure database (for example, PostgreSQL or Amazon DynamoDB) where you can store tokens generated during the de-identification of a dataset. Token vaults are only used for consistent tokenization (always returning the same token for the input value). Each token in a token vault is unique. That is, each token is only returned for one value. Token vaults allow for re-identification. That is, you are able to take a token from a de-identified dataset and look up the original input value.

transformation	A transformation defines a set of behaviors (privacy enhancing technologies) for the platform to execute on a field in a dataset to de-identify it, while still preserving data utility.
transformation policy	A transformation policy is a reusable set of transformation rules that serves a business context. A transformation policy is a flexible construct that allows you to apply transformation rules in the way that best meets your needs. For example, you can write a policy around a regulation (such as HIPAA or GDPR) or around a business context (such as provisioning data for marketing analytics). The order of transformation policies matters. The platform applies them in the order that they are arranged by the data guardian.
transformation rule	Transformation rules are conditional based on attributes, such as user group, terms, tags, location, and so on. Transformation rules apply pre-defined transformations to data classes.

W

watermark	A watermark is a unique digital pattern created by the Privitar platform that is added into the records of de-identified datasets for traceability reasons. The platform adds watermarks to the data during the process of de-identification. They are invisibly embedded and distributed throughout the data, and as a result are robust against tampering and operations, such as filtering or reorganizing of the data. In the event of a leak or data breach, watermarks can be used to identify the data and plug potential security holes faster. Watermarks can also act as a deterrent to anyone handling the data, encouraging them to take the security of the dataset seriously when they know that the data can be traced.