pravidha

Pramana

Security and Trust

Governance evidence for Snowflake — classification, policy coverage, drift, audit packs

Document	Pramana — Security and Trust
Version	v1.0
Date	17 November 2026
Author	Pravidha — Snowflake Select Service Partner
Status	Released
Classification	Public

© 2026 Pravidha. This document and the software it describes are Pravidha intellectual property.

Contents

The short answers
What Pramana can and cannot reach
Where it runs
What it asks for
What it reads
What it stores
What it never does
What leaves your account
Who can do what inside it
Software and supply chain
Limits you should know about
Removing it
Pravidha as a supplier

© 2026 Pravidha · Public

For security architects, information security officers and anyone carrying out a vendor or third-party risk review. It states what Pramana can reach, what it stores, what leaves your account and what does not. About twelve minutes.

The short answers

Question	Answer
Where does it run?	Inside your Snowflake account, as a Snowflake Native App. There is no Pravidha-hosted service
Does it read the data in our tables?	No. It holds no privilege on your databases and asks for none
What does it read?	Eight of Snowflake's `ACCOUNT_USAGE` metadata views, through three Snowflake viewer roles that you grant
Can it change our objects?	No. It writes only inside itself. Fixes are produced as statements for you to run
Does anything leave our account?	Not through Pramana. Error logs are shared with Pravidha only if you switch that on. The consoles' fonts are fetched by the viewer's browser from `fonts.gstatic.com`
Can Pravidha see our installation?	No. Pravidha cannot log in to it, query it or read what it holds
Does it make network calls?	No. It has no external access integration and requests none
What does it create in our account?	One application and, once you start the schedule, one serverless task inside it. No warehouse, database, role or integration
How is it removed?	`DROP APPLICATION PRAMANA CASCADE;` removes the application and everything it holds

The rest of this document gives the detail behind each answer, so that it can be checked.

© 2026 Pravidha · Public

What Pramana can and cannot reach

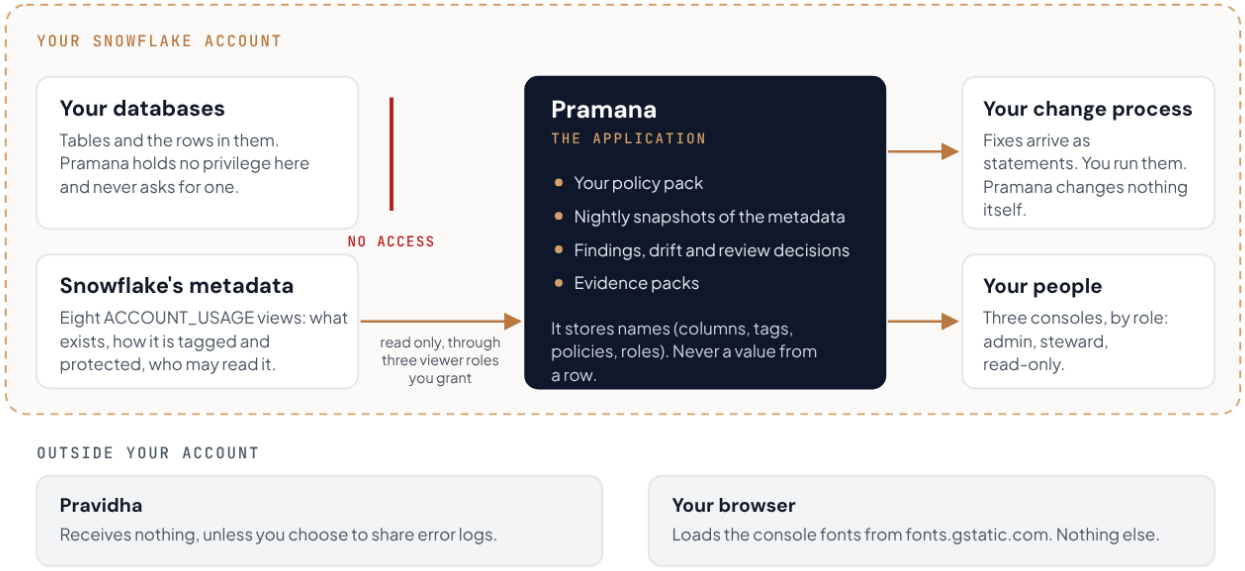


Figure 1 · *What Pramana can and cannot reach: no access to your databases, read-only metadata, statements out, nothing leaves the account*

Where it runs

Pramana is a Snowflake Native App. Installing it creates an application object in your account. Its code, a Python stored procedure and three Streamlit consoles, runs on Snowflake's compute within your account boundary, under Snowflake's Native App security model. The nightly job is a Snowflake serverless task, so no warehouse of yours is used and none is created.

Pravidha publishes the application; Snowflake delivers it. Before a version can be released to customers, Snowflake runs its automated security scan on the package. Upgrades arrive the same way and are applied in place by Snowflake.

What it asks for

Everything Pramana requests is declared in its manifest and shown to you by Snowsight at install time.

Request	Purpose	When
`EXECUTE TASK`, `EXECUTE MANAGED TASK`	Run its one scheduled job on serverless compute	Granted at install. Nothing is scheduled until an administrator starts it
`OBJECT_VIEWER` (SNOWFLAKE database role)	Read which tables and columns exist	You grant it, before the first scan of your account
`GOVERNANCE_VIEWER` (SNOWFLAKE database role)	Read tags, policies and classification results	As above
`SECURITY_VIEWER` (SNOWFLAKE database role)	Read who has been granted what	As above
`READ SESSION`	Record the name of the user who takes a review decision	Optional. Without it, decisions are recorded as `UNRECORDED`

© 2026 Pravidha · Public

It does **not** ask for:

- any privilege on your databases, schemas, tables or views;
- `IMPORTED PRIVILEGES` on the SNOWFLAKE database, which would open all of `ACCOUNT_USAGE`;
- `USAGE_VIEWER`, which covers usage and cost history;
- a warehouse, an external access integration, a notification integration or a network rule;
- privileges to apply tags or policies, or to manage grants.

The built-in sample needs none of the grants above. It is a fictional estate shipped inside the application.

What it reads

Eight views in `SNOWFLAKE.ACCOUNT_USAGE`, restricted to the databases you place in scope:

View	What Pramana takes from it
`TABLES`, `COLUMNS`	Names of tables and columns, data types, owners
`TAG_REFERENCES`	Which tag sits on which object, and its value
`POLICY_REFERENCES`	Which masking and row access policies are attached where
`MASKING_POLICIES`, `ROW_ACCESS_POLICIES`	Which policies exist
`DATA_CLASSIFICATION_LATEST`	The categories Snowflake's own classifier assigned, where your account has run it
`GRANTS_TO_ROLES`	Which roles hold which privileges on the tables in scope

These views describe your account. They do not contain the rows of your tables.

What it stores

Everything Pramana stores stays inside the application, in your account:

- **Your policy pack:** scope, taxonomy, bindings, governed tags, exemptions, settings.
- **Snapshots of metadata:** names of databases, schemas, tables and columns; tag values; policy names; role names and privileges; classifier categories.
- **Results:** findings, proposals and the decisions taken on them, drift events, the run log.
- **Evidence packs.**

Three points are worth knowing for a privacy review:

1. **Names can be informative.** Pramana stores object names, not values. A column called `DIAGNOSIS` reveals that such a column exists, to anyone you allow to read Pramana.
2. **People are named in decisions.** With `READ SESSION` granted, the Snowflake user name of whoever approves a proposal, acknowledges a drift event or adds an exemption is recorded with it. The approver recorded on an exemption is the user who added it, or the name given in your policy pack files.
3. **One optional feature can copy rows.** An *evidence source* of type `VIEW` lets you attach a view of your own to every pack. Pramana then reads that view and places a sample of up to twenty of its rows in the pack. It is off unless an administrator adds such a source and grants

© 2026 Pravidha · Public

the application access to the view. Point it only at views whose contents you are content to see in an evidence pack.

What it never does

- **It never reads your tables.** It has no privilege to.
- **It never changes your objects.** It cannot set a tag, attach a policy or revoke a grant. Remediation is a list of statements; a person or your pipeline runs them.
- **It never calls out.** No network request is made by the application.
- **It never sends e-mail or alerts on its own.** If you want to be told about weakening drift, you create the alert in your own schema with your own notification integration. Operations and Troubleshooting has a ready-made one.

What leaves your account

From the application: nothing by default.

- **Logs.** Pramana writes one informational line per run and a warning when a run finds weakening drift. They go to your account's event table, if you have one. They contain run identifiers and counts.
- **Sharing with Pravidha.** Snowflake lets a consumer choose to share an application's errors and warnings with its provider, to help with a support case. Pramana declares this as optional. It is off until you turn it on in the application's settings in Snowsight, and you can turn it off again.
- **Fonts.** The consoles use three typefaces. The viewer's browser fetches them from `fonts.gstatic.com`, a Google service, which therefore sees the browser's IP address and user agent, as for any web font. No data from Pramana is part of that request. Where the host is blocked, the consoles fall back to system fonts and work as before.

Who can do what inside it

Access to Pramana is given through four application roles, which you grant to your own account roles.

Role	Intended for	Can
`PRAMANA_ADMIN`	Platform or governance engineering	Everything: maintain the policy pack, run the engine, start and stop the schedule, load and clear the sample
`PRAMANA_STEWARD`	Data stewards	Read everything; approve and reject proposals, acknowledge drift, add and close exemptions
`PRAMANA_AUDITOR`	Audit, risk, security	Read findings, drift, evidence packs and the policy pack. Change nothing
`PRAMANA_EVIDENCE_READER`	The service role of a GRC tool	Read one view, `PRAMANA_EVIDENCE_V`, and nothing else

Administrators, stewards and auditors each have a console of their own, so what a person can do is decided by which console their role may open, not by anything they type. The evidence reader has no console.

Two properties matter for the integrity of the record:

© 2026 Pravidha · Public

- **History cannot be edited.** Administrators can change the ten policy pack tables. No role you can grant has `INSERT`, `UPDATE` or `DELETE` on snapshots, findings, drift events or evidence packs. Only the application writes those. What an administrator can do is run the retention purge, which removes whole snapshots older than the period you set and never the two newest.
- **Every pack is sealed.** Each evidence pack carries a SHA-256 of its own content. A copy taken away can be checked against it; the Evidence Guide shows how.

The application's procedures run with the application's own rights, which are limited to what is listed under "What it asks for". Granting a person an application role never gives them more access to your account than the application itself has.

Software and supply chain

Component	Detail
Engine	One Python 3.11 stored procedure. Its only package is `snowflake-snowpark-python`, from Snowflake's own channel
Consoles	Streamlit in Snowflake, pinned to Streamlit 1.39.0, with pandas, from Snowflake's own channel
Build	The application package is generated from source by script and checked in continuous integration; nothing in it is edited by hand
Tests	An automated suite runs on every change without needing a Snowflake account, and an acceptance run on a real account precedes each release
Release	Each version passes Snowflake's security scan before it can be released to customers

Limits you should know about

These are limits of the evidence, set out so that reliance on it is informed.

- **Metadata runs behind.** `ACCOUNT_USAGE` is typically up to two or three hours behind the account. Where policies are attached is the slowest part to arrive, and for the first policies an account ever attaches it has taken more than seven hours. Pramana says so in its results when it sees the signs.
- **Presence, not behaviour.** Pramana proves that the policy you bound is attached. It does not evaluate what the policy's logic returns to a given role.
- **Scope is what you set.** Only databases you place in scope are examined.
- **Classification depends on its inputs.** A sensitive column is found when Snowflake's classifier has run on its table, or when its name matches one of your rules. A column that neither catches is not flagged.

Removing it

```
CALL PRAMANA.CORE.GOV_STOP();
DROP APPLICATION PRAMANA CASCADE;
```

This removes the application, its schedule, your policy pack and every snapshot and evidence pack it holds. Nothing remains in your account. Export the packs you need first; the Evidence Guide shows how.

© 2026 Pravidha · Public

Pravidha as a supplier

Pravidha is a Snowflake Select Services Partner. Because Pramana runs wholly inside your account and Pravidha has no access to it, Pravidha does not receive, store or process your data in providing it. If you choose to share error logs, Pravidha receives those log lines and nothing else.

Pravidha's privacy policy is at pravidha.com/privacy-policy.

Pramana is provided by Pravidha Solutions Limited. The Marketplace listing is offered under Snowflake's Standard Agreement for Marketplace Products; the full version is supplied under terms agreed with Pravidha when you request it.

Support is by e-mail at **support@pravidha.com**, on any day and at any hour, and we aim to send a first reply within two hours. A suspected security issue is reported to the same address.

Questions about this document: **info@pravidha.com**.

© 2026 Pravidha · Public