Trino’s New OPA Authorizer: *An Open Source Love Story*

Trino Summit 2023
December 14, 2023

Pablo Arteaga
Software Engineer, Reporting Apps Engineering, Bloomberg

Sönke Liebau
Chief Product Officer, Stackable GmbH

TechAtBloomberg.com

© 2023 Bloomberg Finance L.P. All rights reserved.

Stackable

An Open Source Love Story

Who am I?

Sönke Liebau
CPO Stackable

- Co-Founder Stackable
- Many years as a Big Data Consultant
- The man with the vision

“If you have visions, go see a doctor!”

Stackable

What made us embark upon this journey?

Stackable

https://www.pexels.com/photo/train-on-a-platform-in-black-and-white-18152212/

Platform ... what does that even mean?

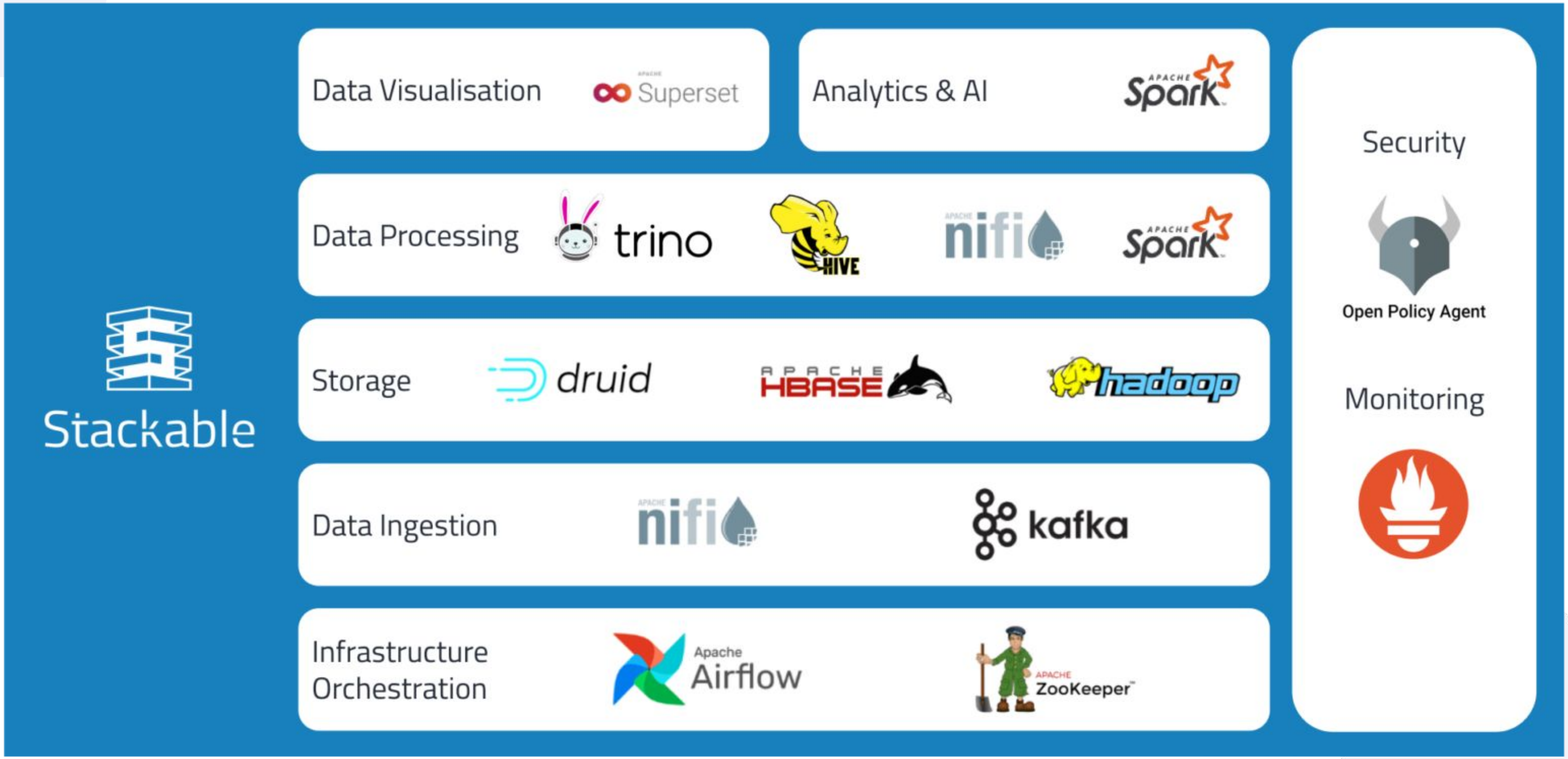
- Many things to many people ….
  - Ease of use
  - Support
  - Integrated everything
  - GUI
  - …

Stackable

Platform ... what does that even mean?

“A platform is a set of software and a surrounding ecosystem of resources that helps you to grow your business. A platform enables growth through connection: its value comes not only from its own features, but from its ability to connect external tools, teams, data, and processes.”

Source: https://blog.hubspot.com/marketing/software-platform


Stackable
Data Visualisation
Superset
Analytics & AI
Spark
Data Processing
trino
HIVE
nifi
Spark
Storage
druid
APACHE HBASE
hadoop
Data Ingestion
nifi
kafka
Infrastructure Orchestration
Apache Airflow
APACHE ZooKeeper
Security
Open Policy Agent
Monitoring

We still want things to feel the same ...

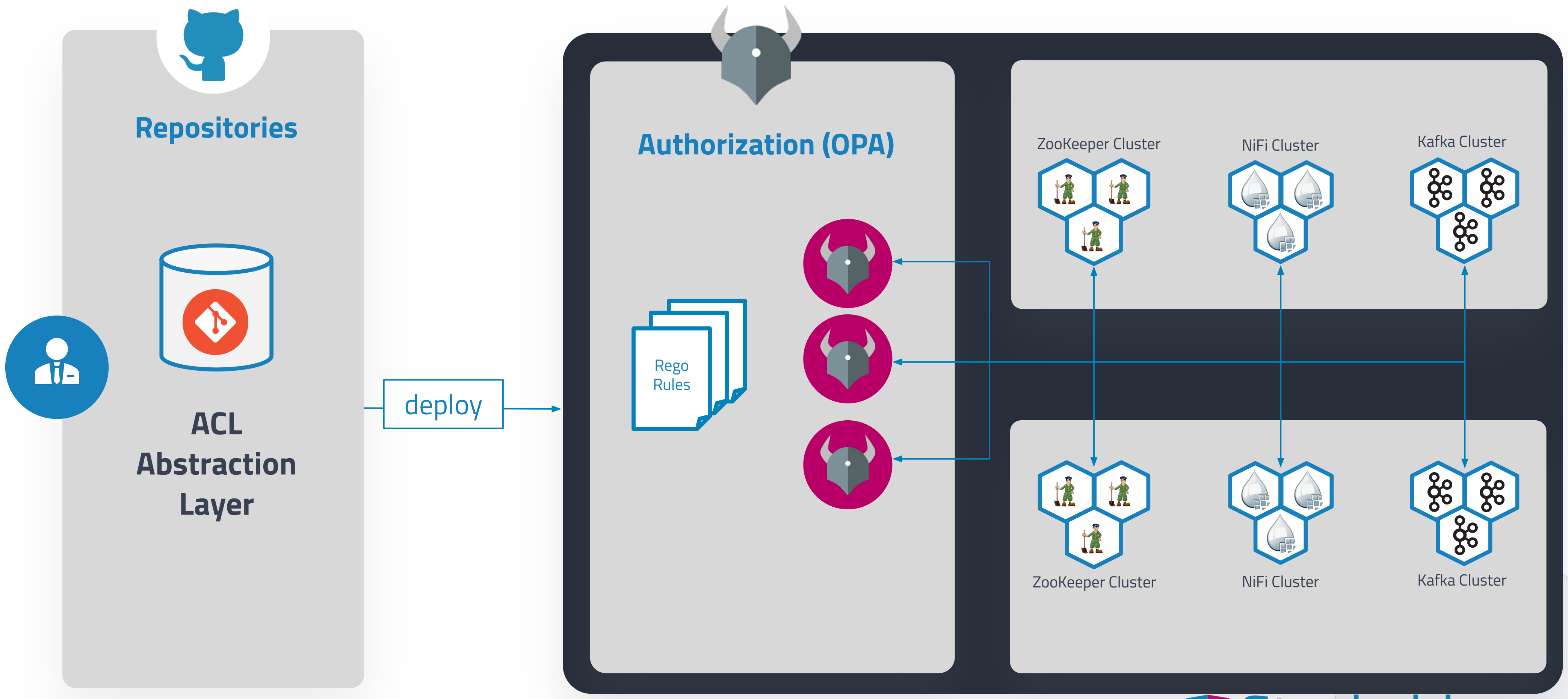
- Configure TLS the same way for every product
- Specify S3 backends only once and reuse them
- Specify your AD only once and reuse it
- Configure products to work with each other automatically ..
- …

Stackable

For the purpose of this talk ...

One place to specify who is allowed to access what!

What we want ...



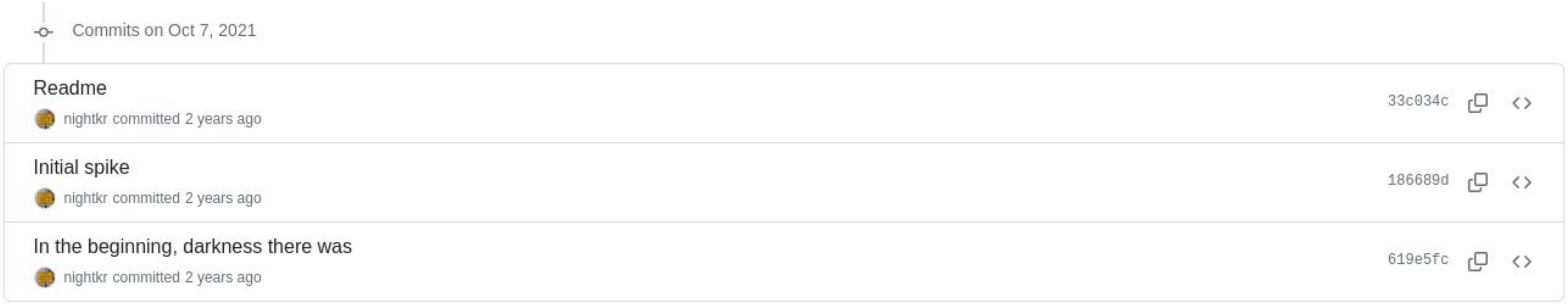
Stackable

Why not Ranger?

- Ranger has a fixed development model
- To add new systems you need to write new modules, compile and roll out Ranger
- OPA is all REST
  - Basically everything is configuration
- We can build the 80% abstraction layer easily
- Anybody else -> they can build whatever extra they need -> in config!

Stackable

The beginning ...

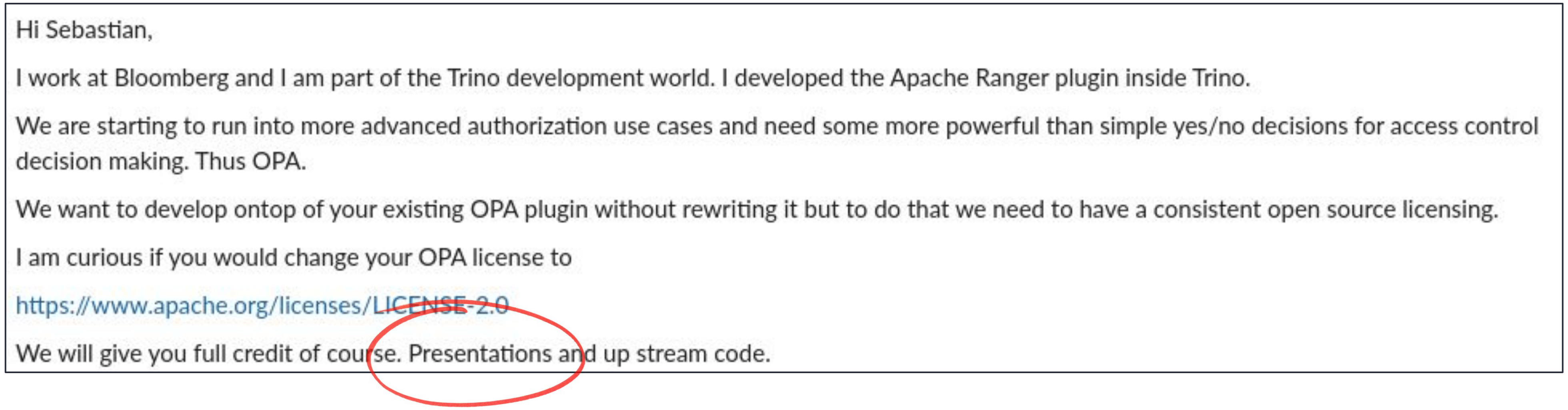

Commits on Oct 7, 2021

Readme
nightkr committed 2 years ago
33c034c

Initial spike
nightkr committed 2 years ago
186689d

In the beginning, darkness there was
nightkr committed 2 years ago
619e5fc

and then there was light ...



Hi Sebastian,

I work at Bloomberg and I am part of the Trino development world. I developed the Apache Ranger plugin inside Trino.

We are starting to run into more advanced authorization use cases and need some more powerful than simple yes/no decisions for access control decision making. Thus OPA.

We want to develop ontop of your existing OPA plugin without rewriting it but to do that we need to have a consistent open source licensing.

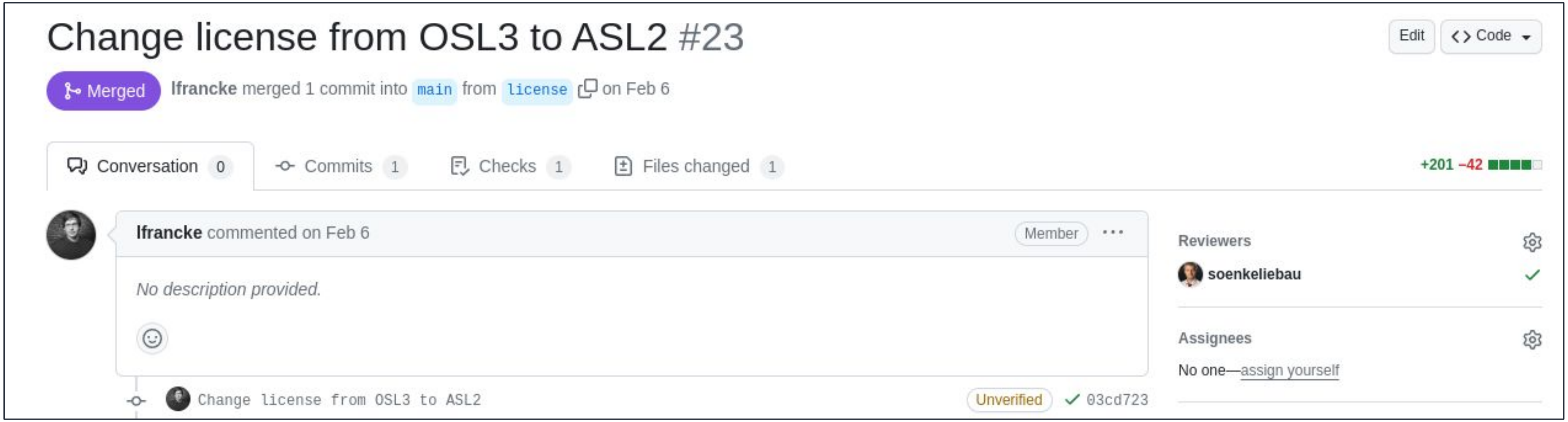
I am curious if you would change your OPA license to

https://www.apache.org/licenses/LICENSE-2.0

We will give you full credit of course. Presentations and up stream code.

Stackable

and then there was light …



Change license from OSL3 to ASL2 #23

Merged lfrancke merged 1 commit into `main` from `license` on Feb 6

lfrancke commented on Feb 6

No description provided.

`Change license from OSL3 to ASL2` Unverified ✓ 03cd723

Reviewers

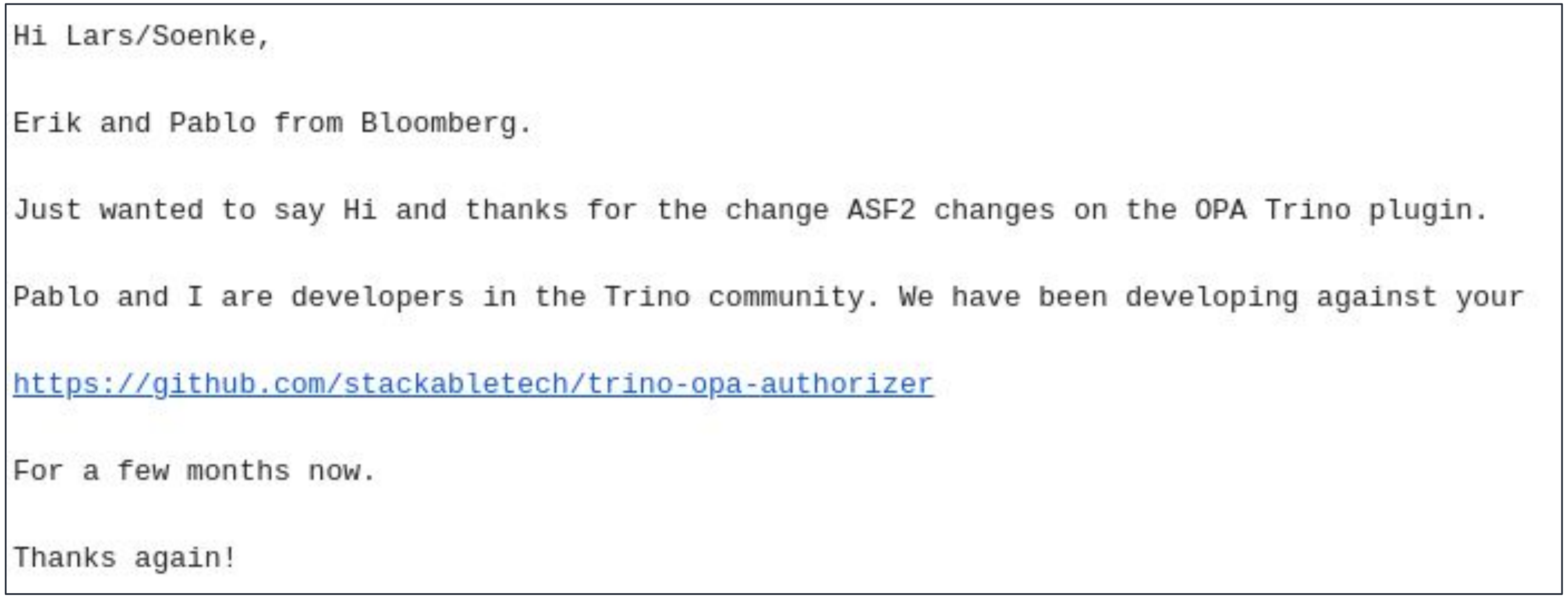
soenkeliebau ✓

Assignees

No one—assign yourself

Stackable

and then there was light ...



Hi Lars/Soenke,

Erik and Pablo from Bloomberg.

Just wanted to say Hi and thanks for the change ASF2 changes on the OPA Trino plugin.

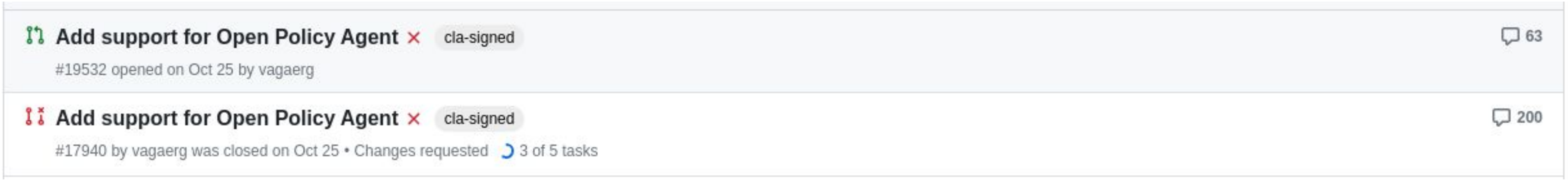
Pablo and I are developers in the Trino community. We have been developing against your

https://github.com/stackabletech/trino-opa-authorizer

For a few months now.

Thanks again!

getting closer ...



Add support for Open Policy Agent × cla-signed

63

#19532 opened on Oct 25 by vagaerg

Add support for Open Policy Agent × cla-signed

200

#17940 by vagaerg was closed on Oct 25 • Changes requested 3 of 5 tasks

Today’s speakers

Pablo Arteaga is a Software Engineer with Bloomberg's Data & Analytics Platform Engineering group. He is part of a team that is building a data mesh and the tooling around it to empower data owners to easily manage and share their datasets in a secure and scalable manner.

© 2023 Bloomberg Finance L.P. All rights reserved.

Our business mandate: Make data accessible

Why did we embark on this journey?

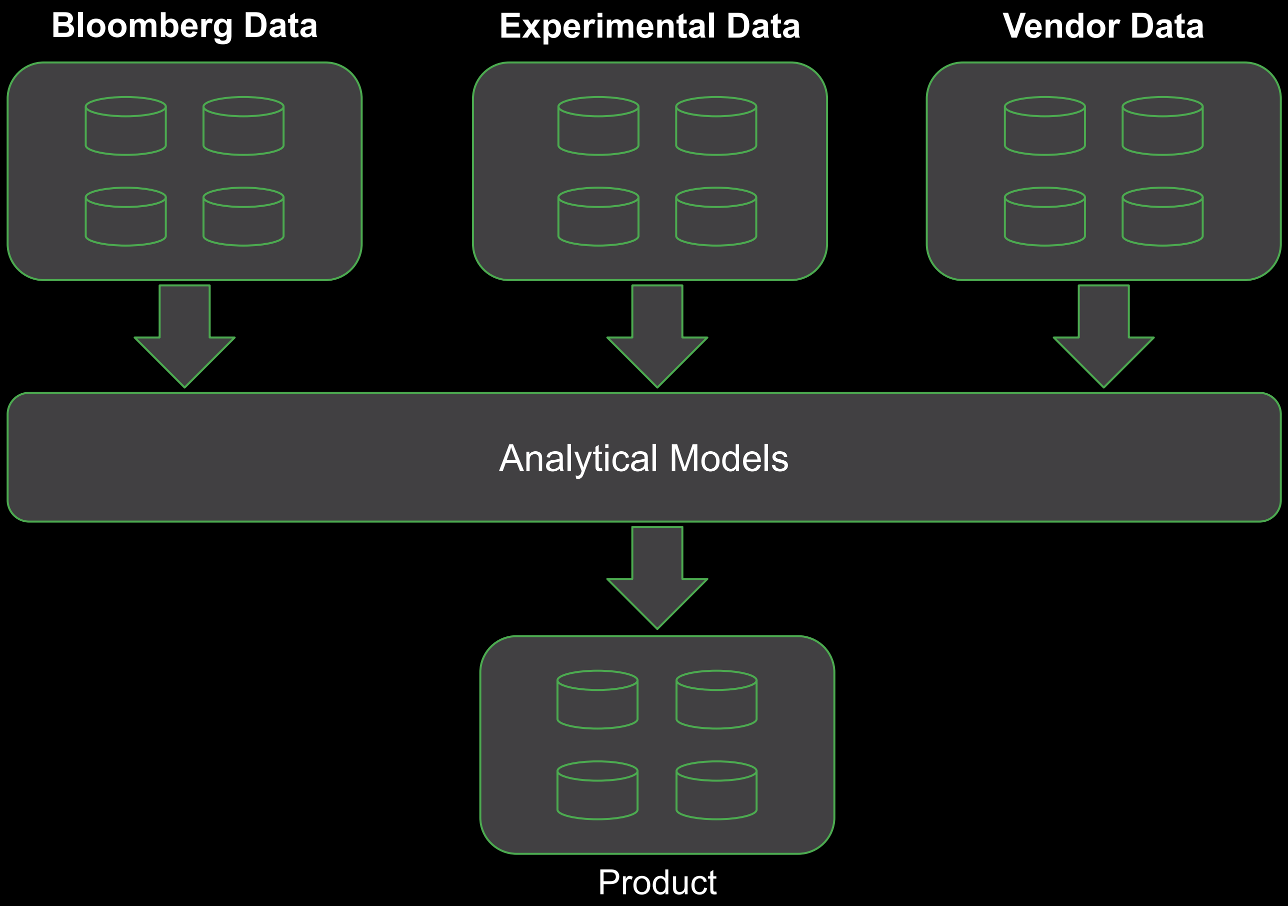
TechAtBloomberg.com

© 2023 Bloomberg Finance L.P. All rights reserved.

Bloomberg

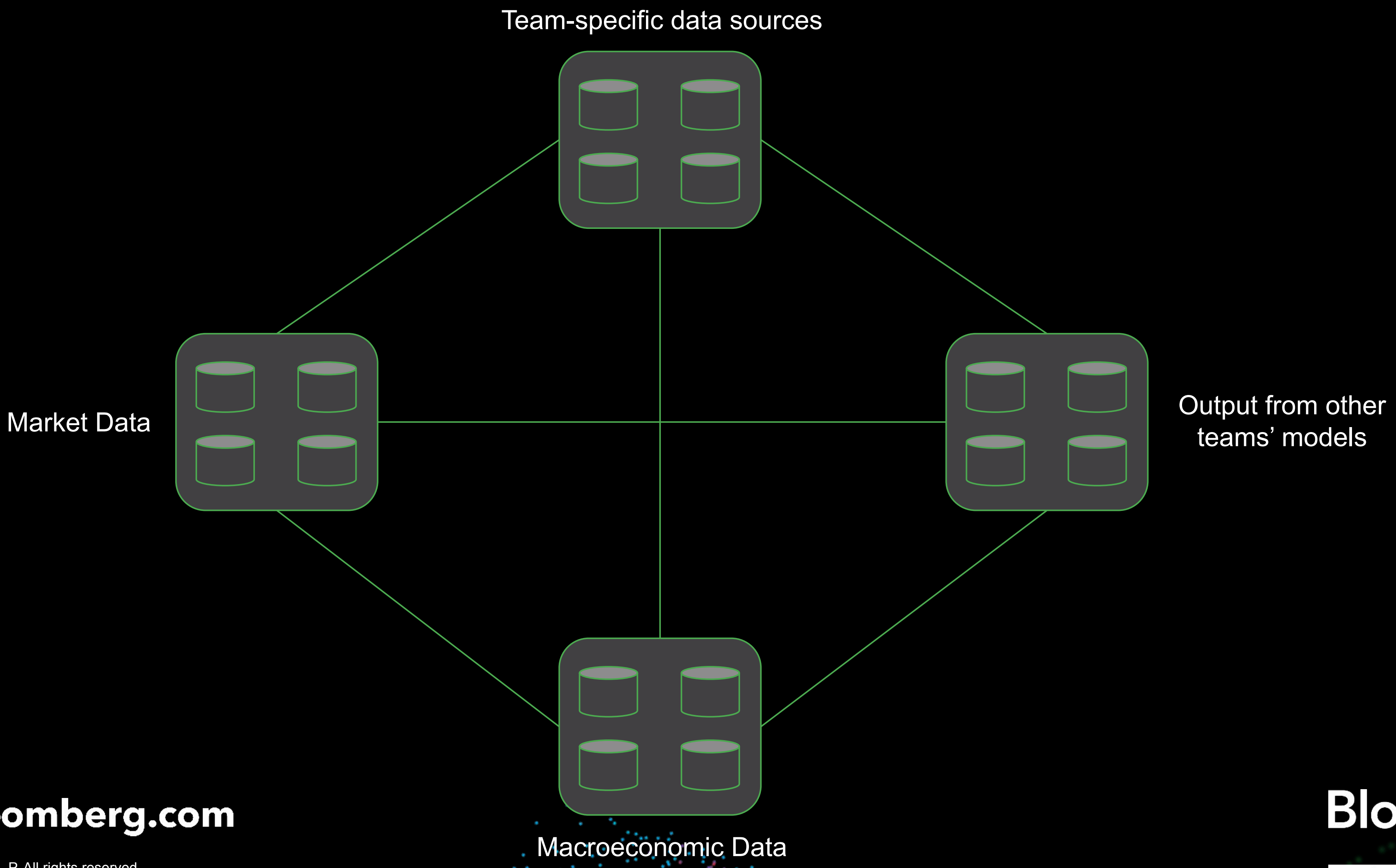
Engineering

Research Analysts: Where we started



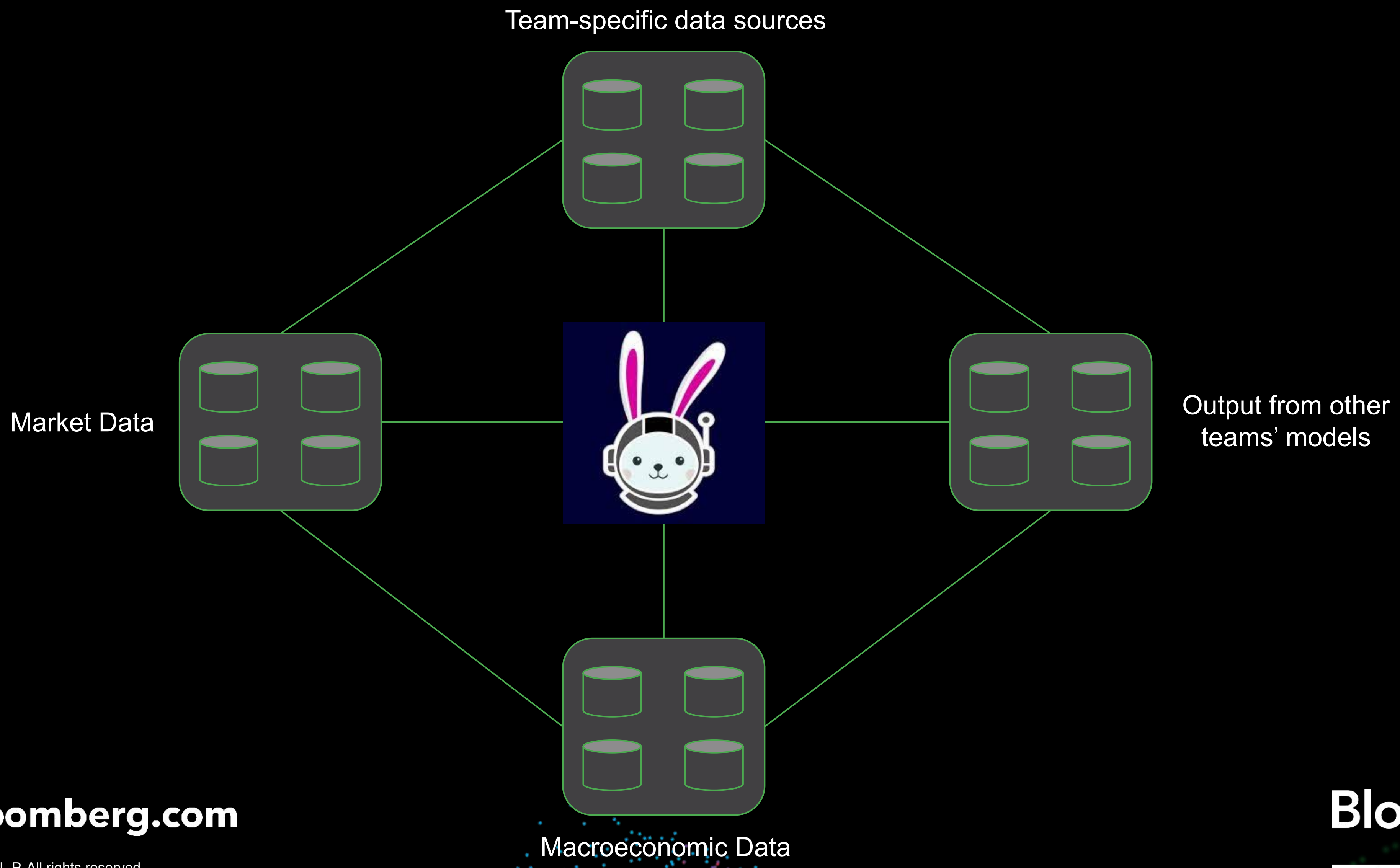
© 2023 Bloomberg Finance L.P. All rights reserved.

An interconnected maze



© 2023 Bloomberg Finance L.P. All rights reserved.

An interconnected maze



© 2023 Bloomberg Finance L.P. All rights reserved.

Our main requirements

- Catalogs are our federation point
- **Data owners** can **expose their datasets** and **control access** to them
- Catalogs must be:

Access-controlled

Data owners want to control access **themselves**

Highly available

Stakeholders demand **no downtime**, **high throughput** and **fast response times**

Auditable

Knowing ***who*** accessed ***what*** and ***when*** to empower diagnostics & traceability

TechAtBloomberg.com

© 2023 Bloomberg Finance L.P. All rights reserved.

Bloomberg

Engineering

Data security

Authorization

- Granular rules: From catalog-level down to row-level access, depending on the use case

Administration of policies - access control policy federation

- **Data owners** define access policies using granular rules
- Policies are flexible

Integration with Enterprise directories (e.g., LDAP)

Traceability: Query audit log enables analysis of usage patterns

© 2023 Bloomberg Finance L.P. All rights reserved.

Our initial approach to access control: Apache Ranger

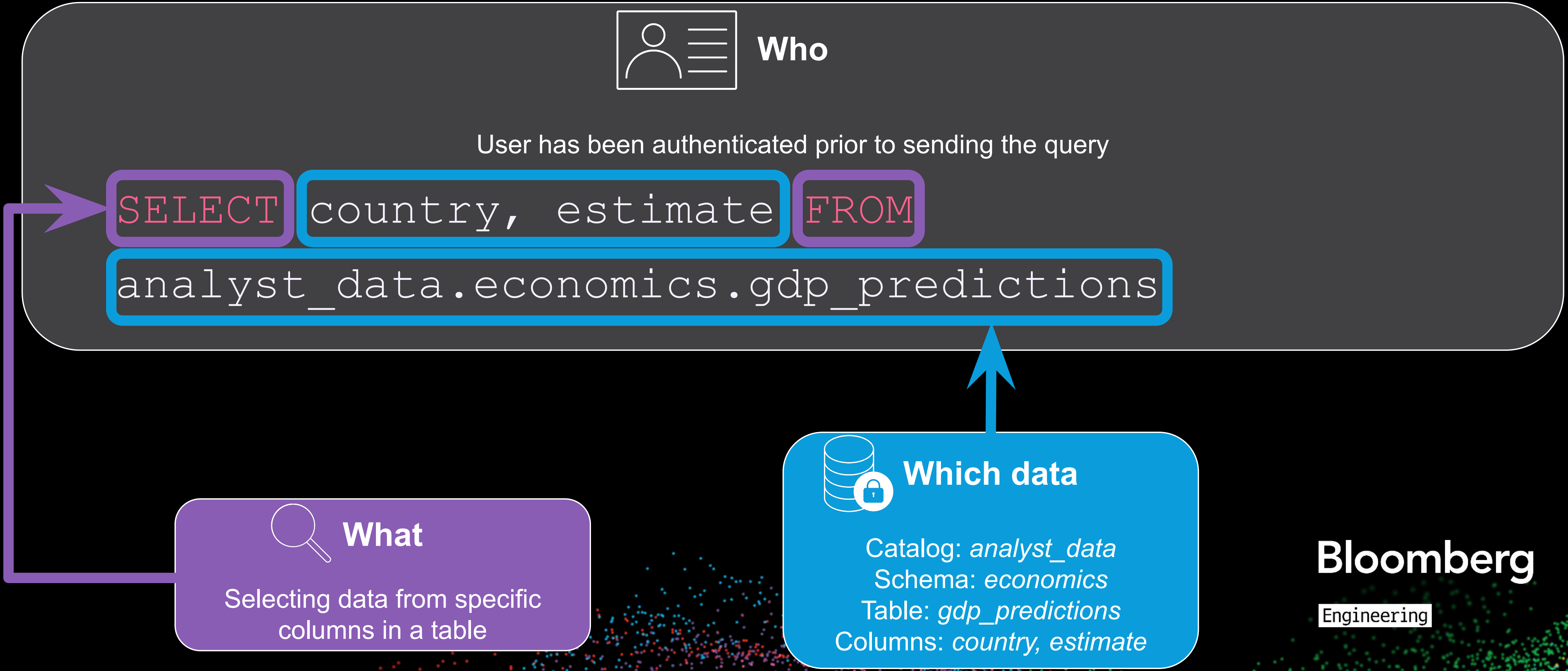
- Apache Ranger is:
  - Open source
  - Well known within the Apache Hadoop ecosystem
  - Actively maintained
  - Extensible

- It meets many of our requirements:
  - Directory integration: Through ranger-usersync
  - RBAC-style rules
  - Friendly self-service UI

© 2023 Bloomberg Finance L.P. All rights reserved.

Ranger’s authorization model

Ranger’s authorization model is intuitive: **who** is doing **what** upon **which data?**



Who

User has been authenticated prior to sending the query

```
SELECT country, estimate FROM
analyst_data.economics.gdp_predictions
```

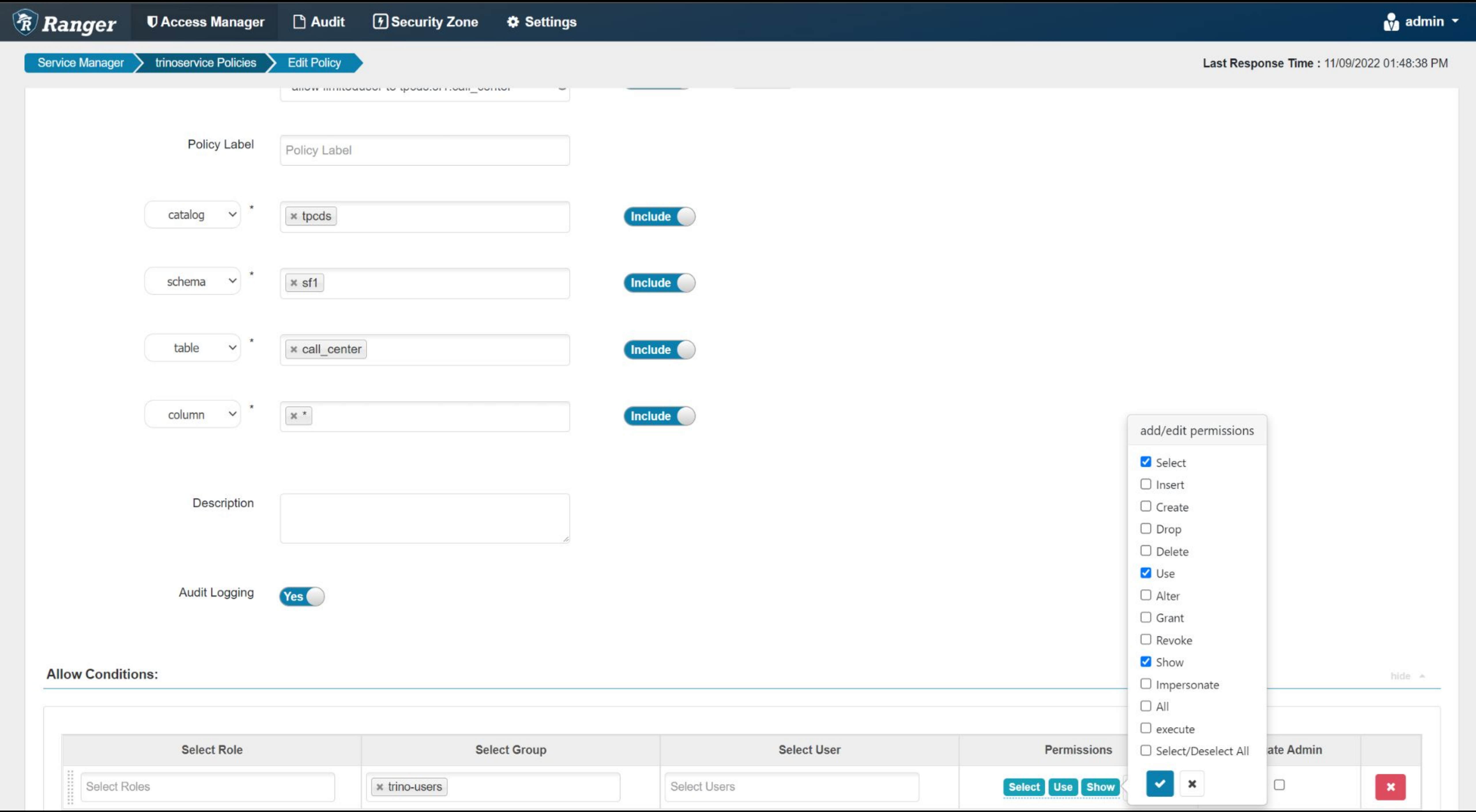
What

Selecting data from specific columns in a table

Which data

Catalog: *analyst_data*
Schema: *economics*
Table: *gdp_predictions*
Columns: *country, estimate*

Bloomberg
Engineering



Last Response Time : 11/09/2022 01:48:38 PM

Policy Label: Policy Label

catalog *: × tpcds — Include

schema *: × sf1 — Include

table *: × call_center — Include

column *: × * — Include

Description:

Audit Logging: Yes

Allow Conditions:

add/edit permissions

- [x] Select
- [ ] Insert
- [ ] Create
- [ ] Drop
- [ ] Delete
- [x] Use
- [ ] Alter
- [ ] Grant
- [ ] Revoke
- [x] Show
- [ ] Impersonate
- [ ] All
- [ ] execute
- [ ] Select/Deselect All

Select Role	Select Group	Select User	Permissions	ate Admin
Select Roles	× trino-users	Select Users	Select Use Show	☐

Where Ranger did not suffice for us

- Ranger’s RBAC system limitations:
  - Resource-based policies are static and may not be expressive enough
  - RBAC policy explosion, particularly in a multi-tenant system like Trino
- Ranger’s ABAC system limitations:
  - Tags are created and synced by external systems
  - Tricky to inspect & debug
- Delegation capabilities:
  - Delegation functionality is absolute - no capability to delegate specific permissions only

© 2023 Bloomberg Finance L.P. All rights reserved.

Where Ranger did not suffice for us

- No clear namespacing / ownership of rules:
  - Hard to know *why* a rule was created or what its purpose is

- Ranger rules are not peer reviewed:
  - Fine for simple rules, but complex Ranger rules may even involve JavaScript logic

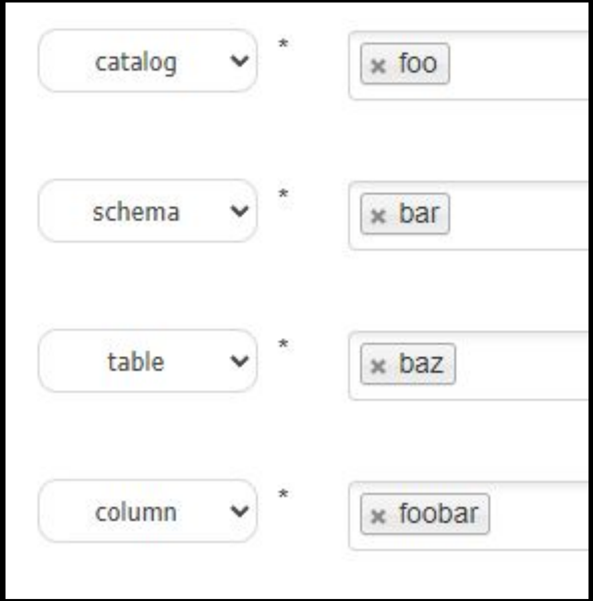
- Ranger is heavyweight and intertwined with applications that use it:
  - Hadoop dependencies make builds & artifacts larger
  - Complex to mock out or run locally for integration testing
  - Changing Ranger's policy evaluation logic requires rebuilding all applications that rely on it

© 2023 Bloomberg Finance L.P. All rights reserved.

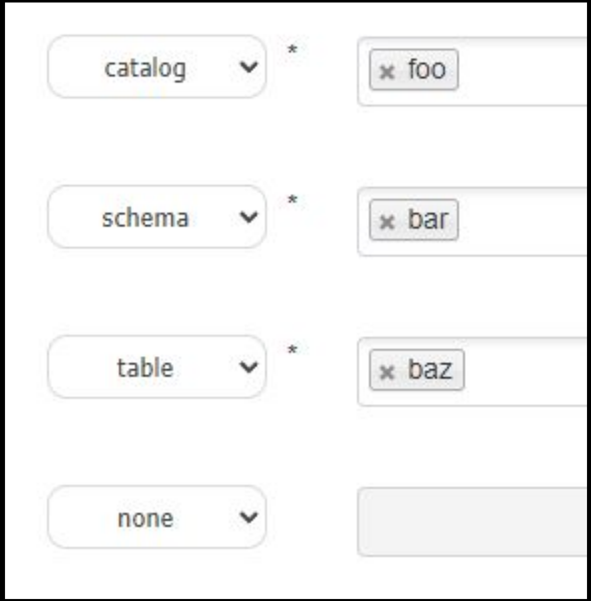
Where Ranger did not suffice for us

Ranger is intended to be generic, and this can result in ambiguous rules…
and developers prefer code over understanding app-specific logic :)

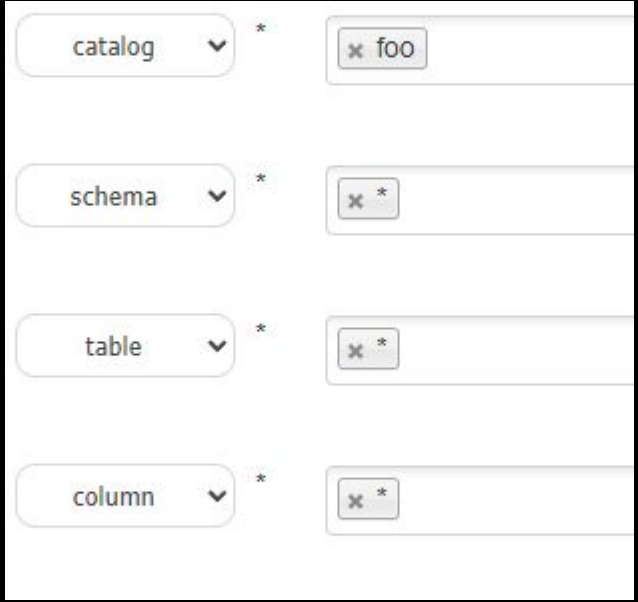
How do these differ for a query on “*foo.bar.baz*” column “*foobar*”?



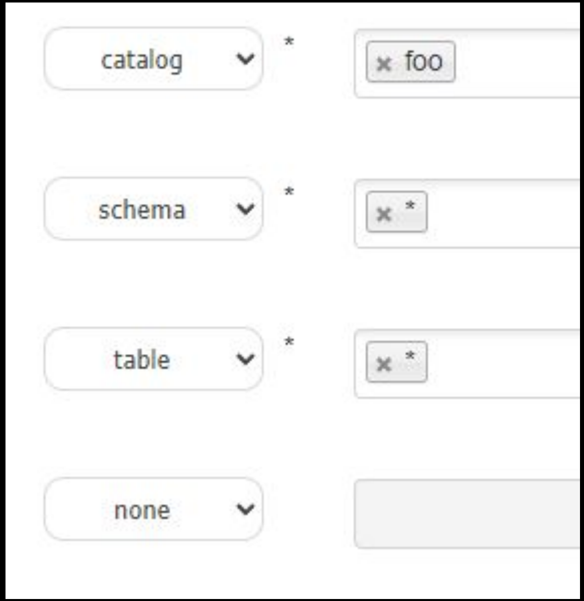
catalog	*	× foo
schema	*	× bar
table	*	× baz
column	*	× foobar



catalog	*	× foo
schema	*	× bar
table	*	× baz
none		



catalog	*	× foo
schema	*	× *
table	*	× *
column	*	× *



catalog	*	× foo
schema	*	× *
table	*	× *
none		

TechAtBloomberg.com

© 2023 Bloomberg Finance L.P. All rights reserved.

Bloomberg
Engineering

Ranger did not suffice for us… and that’s just fine

Ranger covers a specific set of requirements, and does this very well

However, building a multi-tenant, enterprise-ready datamesh requires more:

- Support for complex & pluggable RBAC and ABAC logic
- Peer review capabilities
- Staged policy deployment & policy testing
- Easy integration testing support for local development
- Extensive tooling for inspection & debugging of policies
- More modularity: Ranger’s *one-stop-shop* model makes it hard to integrate with other systems

© 2023 Bloomberg Finance L.P. All rights reserved.

Ranger’s RBAC limitations - Some rules just don’t play nicely

Organizationally-aware rules - For a given catalog, users can:

- Write data to any table *if they are the owner of said catalog*
- Read data from any table *if they are within the same team as the owner*
- Inspect the schema of tables (but not read data from them) *if they are within the same department as the owner*

To some extent, this *can* be done with Ranger ABAC rules, but things like traversing org charts (person > team > department) are not trivial without a powerful query language

TechAtBloomberg.com

© 2023 Bloomberg Finance L.P. All rights reserved.

Bloomberg

Engineering

Ranger’s RBAC limitations - Some rules just don’t play nicely

Dynamic rules, for instance:

- Matching resources based on regex expressions
- Dynamic attributes: time of day, IP addresses, time since user last logged in
- Logic-defined user grouping: Applying rules to users based on the *intersection* of several groups, for instance

Some of these *can* be done through frequent *ranger-tagsync* invocations and the creation of specific tags & groups for each required attribute, but it is not ideal and becomes hard to debug

© 2023 Bloomberg Finance L.P. All rights reserved.

Ranger's RBAC limitations - Some rules just don't play nicely

Global invariant enforcement, regardless of whatever other rules users have added

For instance, regardless of whatever rules exist…

- (Mutual exclusion) "*Users that can read confidential research data cannot write to public catalogs*"
- (Compliance enforcement) "*No rule can grant users in non-GDPR regions access to GDPR-sensitive data*"
- (Fail-safe invariants) "*Users should never be in more than one region*"

TechAtBloomberg.com

© 2023 Bloomberg Finance L.P. All rights reserved.

Bloomberg

Engineering

Ranger’s RBAC limitations - Some rules just don’t play nicely

Interaction between ABAC & RBAC:

- Ranger treats RBAC and ABAC rules as mostly separate
- This makes it hard to implement logic along the lines of:
  - ABAC rule: Grant access to unpublished research reports for users tagged “*researcher*”
  - + RBAC rule: For catalog “*energy*”, grant access to users in the “*commodities*” team only

This can be done by creating a resource-based rule and an attribute rule to separately enforce both conditions

However, the link between the two isn’t registered - they’re two entirely separate entities

© 2023 Bloomberg Finance L.P. All rights reserved.

Our next step: Open Policy Agent (OPA)

Open Policy Agent

*"OPA is a **lightweight general-purpose policy engine** that can be co-located with your service."*

OPA expresses policies as code, and allows us to:

- Move *all* policy evaluation logic away from Trino
- Implement arbitrarily complex policies
- Follow standard SDLC practices for security policies

https://www.openpolicyagent.org/docs/latest/philosophy/#what-is-opa

© 2023 Bloomberg Finance L.P. All rights reserved.

What makes an OPA policy?

Open Policy Agent

An OPA policy is a snippet of code written in a language called *Rego*, and may use additional data to make its decisions

https://www.openpolicyagent.org/docs/latest/policy-language/#what-is-rego

***Rego* code**

The *logic* of the policy is written as code, using *Rego*

For example

“Users can read data from any table as long as they are in the same team as the owner”

External data

Any data the policy needs to make its decisions

For example

Mappings between:

- Tables to owners
- Users and teams

TechAtBloomberg.com

© 2023 Bloomberg Finance L.P. All rights reserved.

Bloomberg Engineering

Why we like OPA: Lightweight & general-purpose

Open Policy Agent

*"OPA is a **lightweight general-purpose policy engine** that can be co-located with your service. You can **integrate** OPA as a sidecar, host-level daemon, or library."*

*"Services **offload policy decisions to OPA** by executing queries. OPA evaluates policies and data to produce query results [...]"*

https://www.openpolicyagent.org/docs/latest/philosophy/#what-is-opa

© 2023 Bloomberg Finance L.P. All rights reserved.

Why we like OPA: Decoupling of enforcement logic

Open Policy Agent

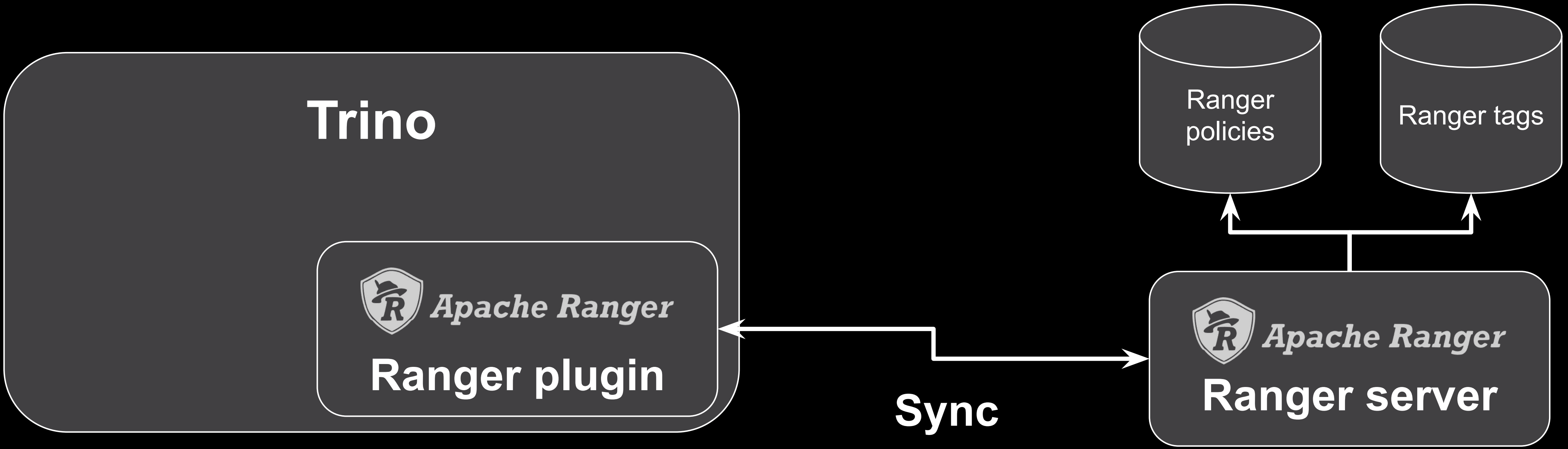
*“Software services should allow policies to be specified declaratively, **<u>updated at any time without recompiling or redeploying</u>**, and enforced automatically [...]”*

*“[...] The policies you write can adapt more easily to the external environment – to factors that the developer **<u>could never have imagined at the time the software service was designed</u>**.”*

https://www.openpolicyagent.org/docs/latest/philosophy/#policy-decoupling

© 2023 Bloomberg Finance L.P. All rights reserved.

Trino & Ranger architecture

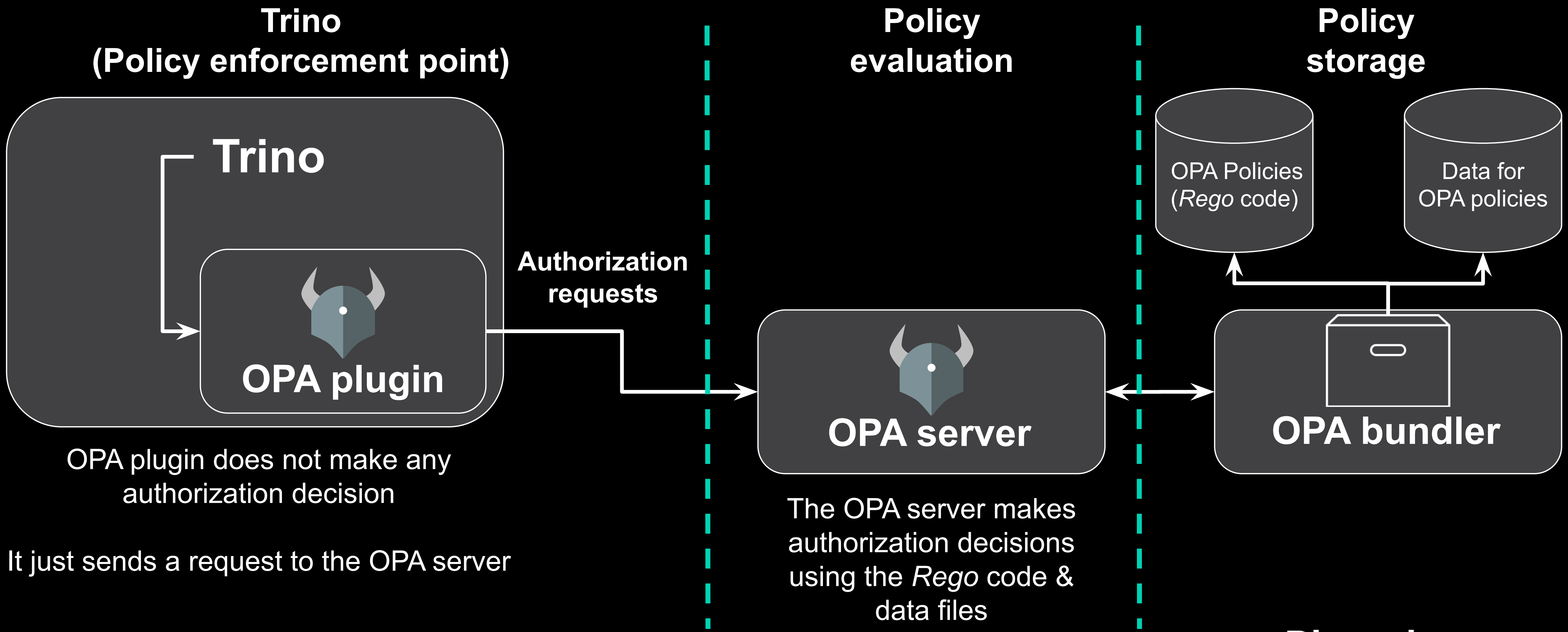


Authorization requests to the Ranger plugin running within the Trino JVM

All evaluation happens within the Ranger plugin, in-process

© 2023 Bloomberg Finance L.P. All rights reserved.

Trino & OPA architecture: Fully decoupled



OPA plugin does not make any authorization decision

It just sends a request to the OPA server

The OPA server makes authorization decisions using the *Rego* code & data files

TechAtBloomberg.com

© 2023 Bloomberg Finance L.P. All rights reserved.

Bloomberg

Engineering

Why we like OPA: Namespacing

Open Policy Agent

All parts of an OPA policy (the *Rego* code and any ancillary data) are **namespaced**; namespaces are hierarchical & multi-level

```
package example.trinosummit.policies

import input
import future.keywords.if
import data.example.trinosummit.some_json_file as json_data

default allow := false

allow if input.context.identity.user in json_data.allowed_users
```

TechAtBloomberg.com

© 2023 Bloomberg Finance L.P. All rights reserved.

Bloomberg

Engineering

What does an OPA request look like?

```
SELECT
    country,
    estimate
FROM
    analyst_data
    .economics
    .gdp_predictions
```

```
{
    "input": {
        "context": {
            "identity": {
                "user": "some-user",
                "groups": ["some-group"]
            }
        },
        "action": {
            "operation": "SelectFromColumns",
            "resource": {
                "table": {
                    "catalogName": "analyst_data",
                    "schemaName": "economics",
                    "tableName": "gdp_predictions",
                    "columns": ["country", "estimate"]
                }
            }
        }
    }
}
```

Who

Authenticated user

What

Selecting data from specific columns in a table

Which data

Catalog: *analyst_data*
Schema: *economics*
Table: *gdp_predictions*
Columns: *country, estimate*

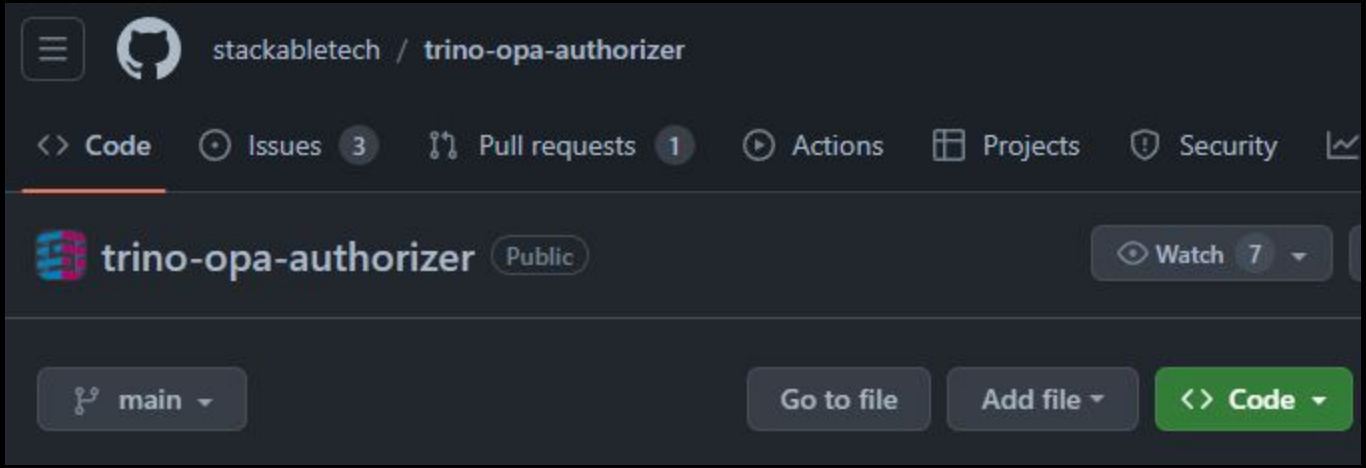
Why we like OPA: A summary

Open Policy Agent

- Policies as code:
  - Easy to test & integrate into SDLC processes
- Policy evaluation & enforcement fully decoupled
- Standard HTTP interface
- Lightweight: can be deployed alongside each Trino coordinator
- Extensible & modular:
  - Policies can use a variety of external data to make decisions
  - Policies can produce complex, non boolean answers

© 2023 Bloomberg Finance L.P. All rights reserved.

Stackable's Authorizer


stackabletech / trino-opa-authorizer

© 2023 Bloomberg Finance L.P. All rights reserved.

Where we are today

Initial PR: https://github.com/trinodb/trino/pull/17940
Superseded by: https://github.com/trinodb/trino/pull/19532

We are hoping to get this merged upstream soon!

But, please reach out to us if you have any experiences you would like to share

© 2023 Bloomberg Finance L.P. All rights reserved.

A workable migration path from Ranger to OPA

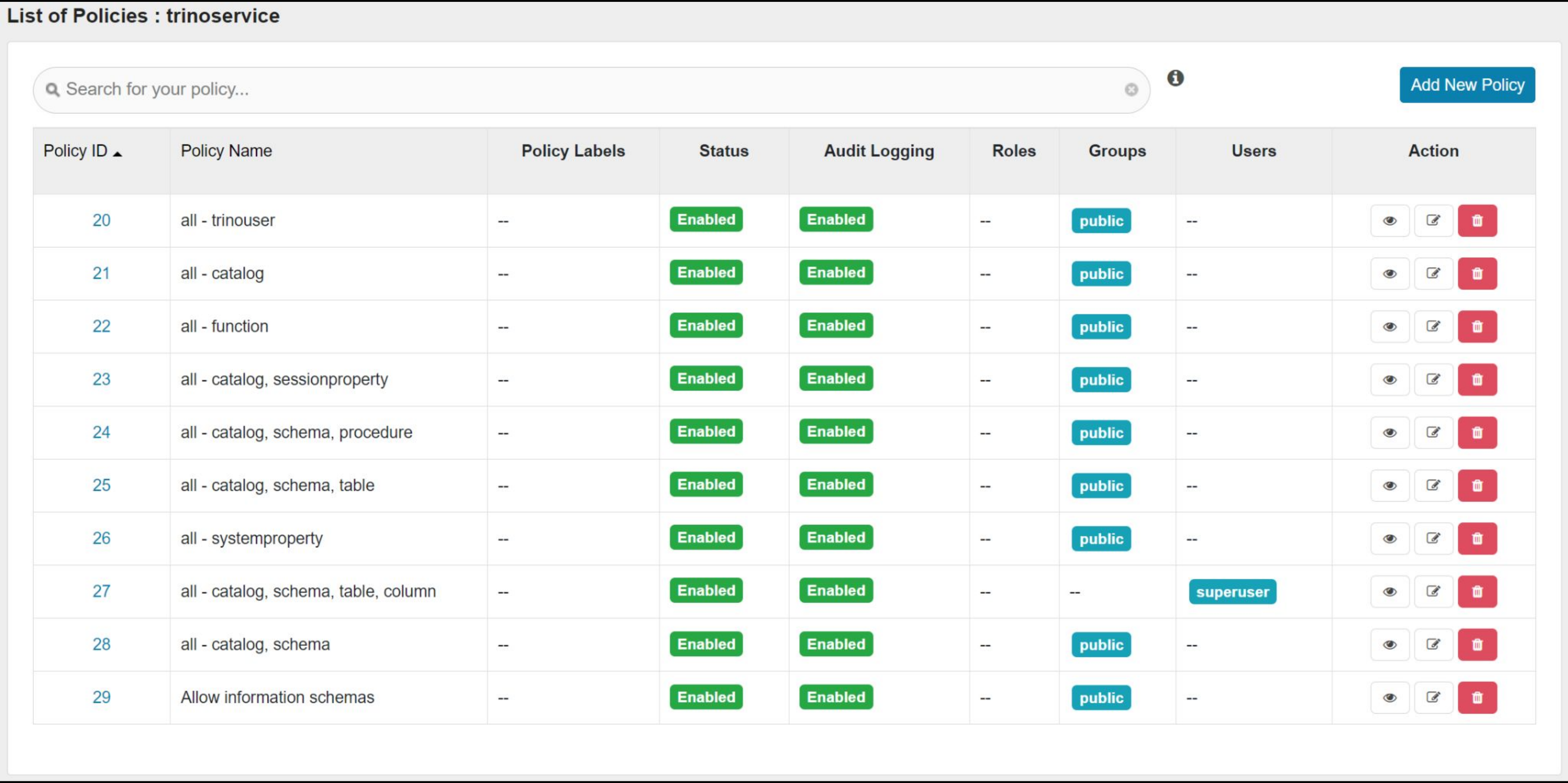
- We have many rules in Ranger that we need to ensure are still enforced
  - Running Ranger **and** OPA alongside it is tricky, so that's not an option

However, OPA policies are code!

- **We can *teach* OPA to behave *like* Ranger**
- Ranger policies are periodically exported using Ranger's REST API and pushed into OPA
- A custom OPA policy can then use these to *simulate* Ranger

We can leverage all the benefits of OPA, while keeping Ranger policies unchanged

© 2023 Bloomberg Finance L.P. All rights reserved.

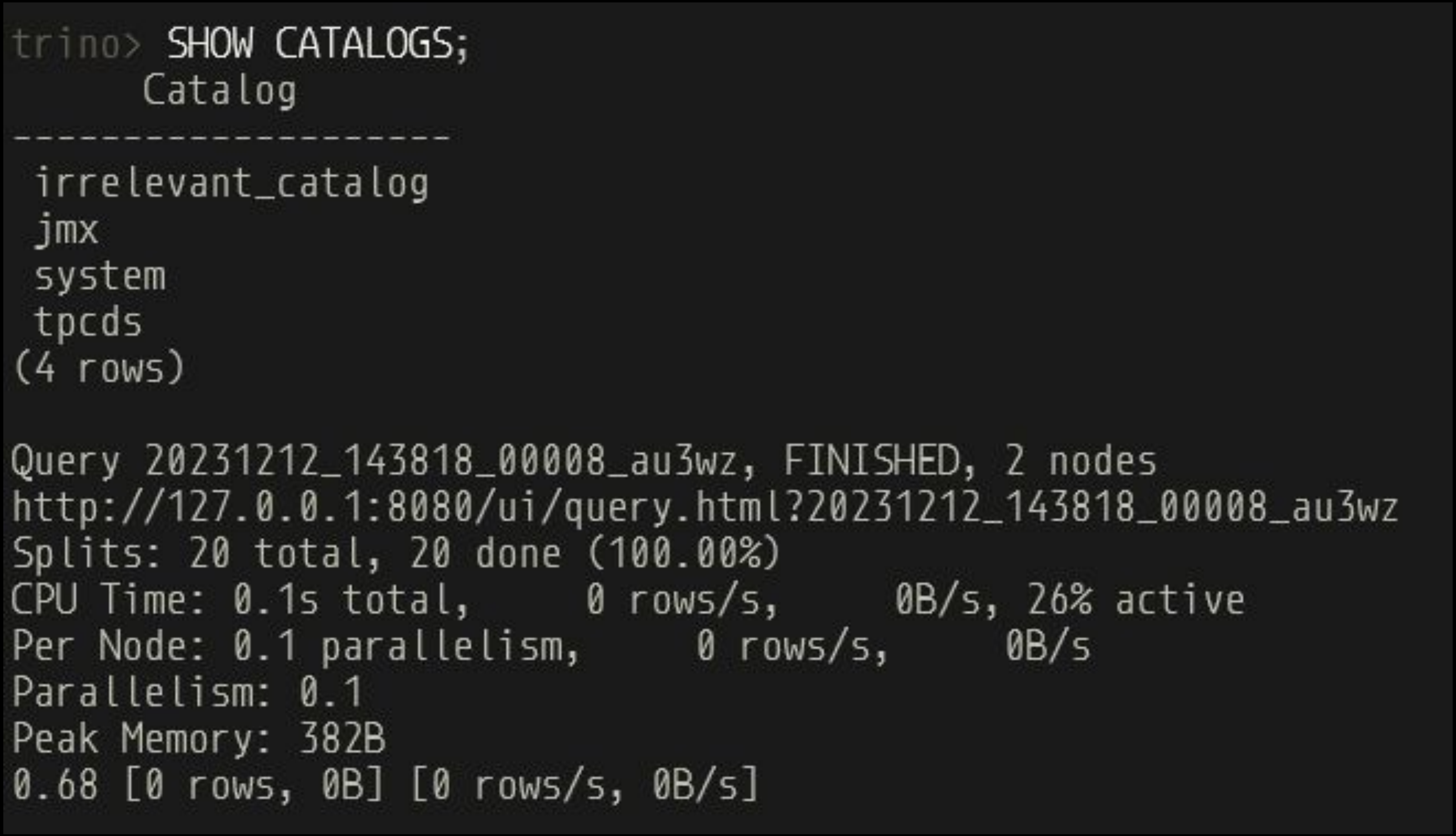


List of Policies : trinoservice

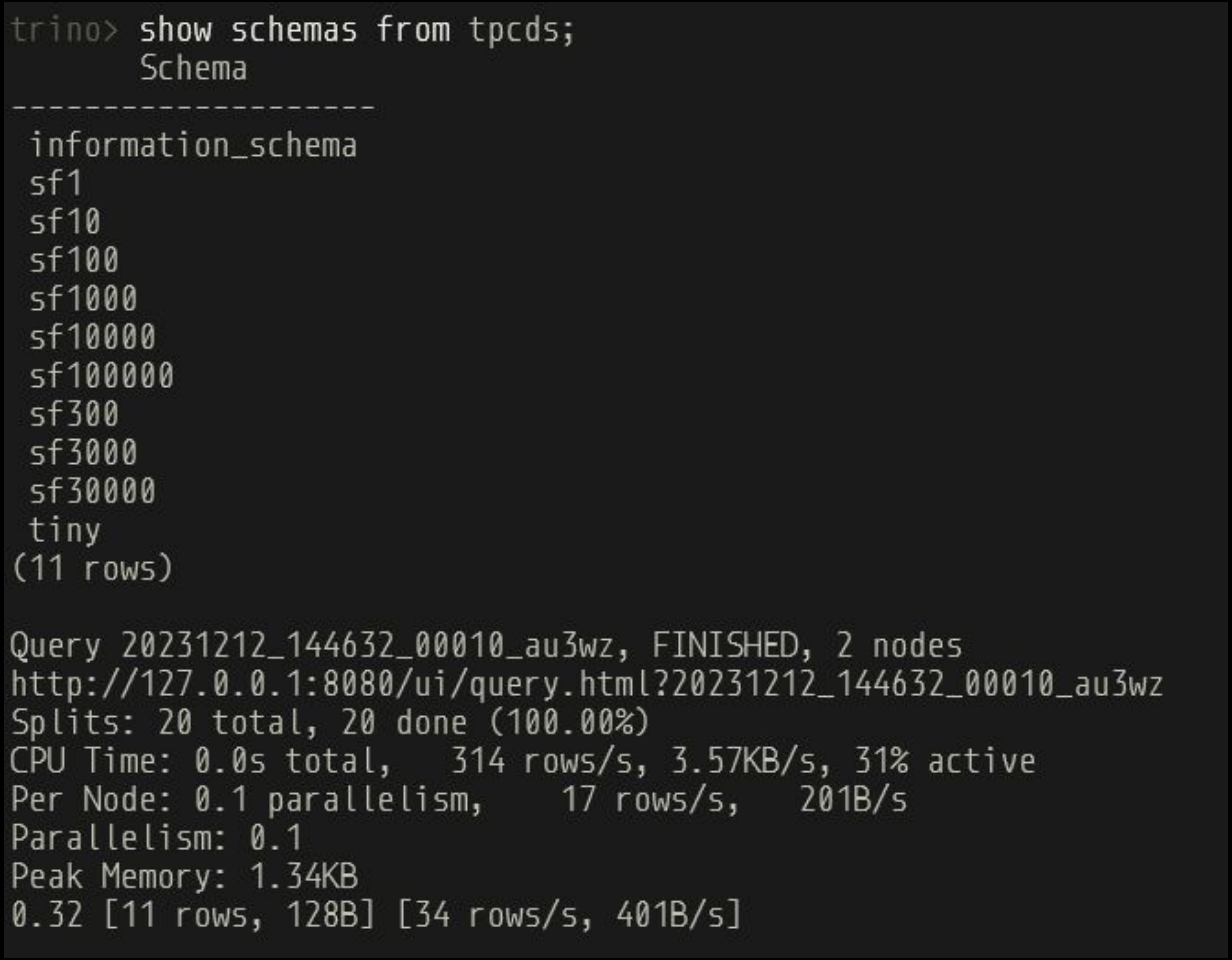
Search for your policy...

Add New Policy

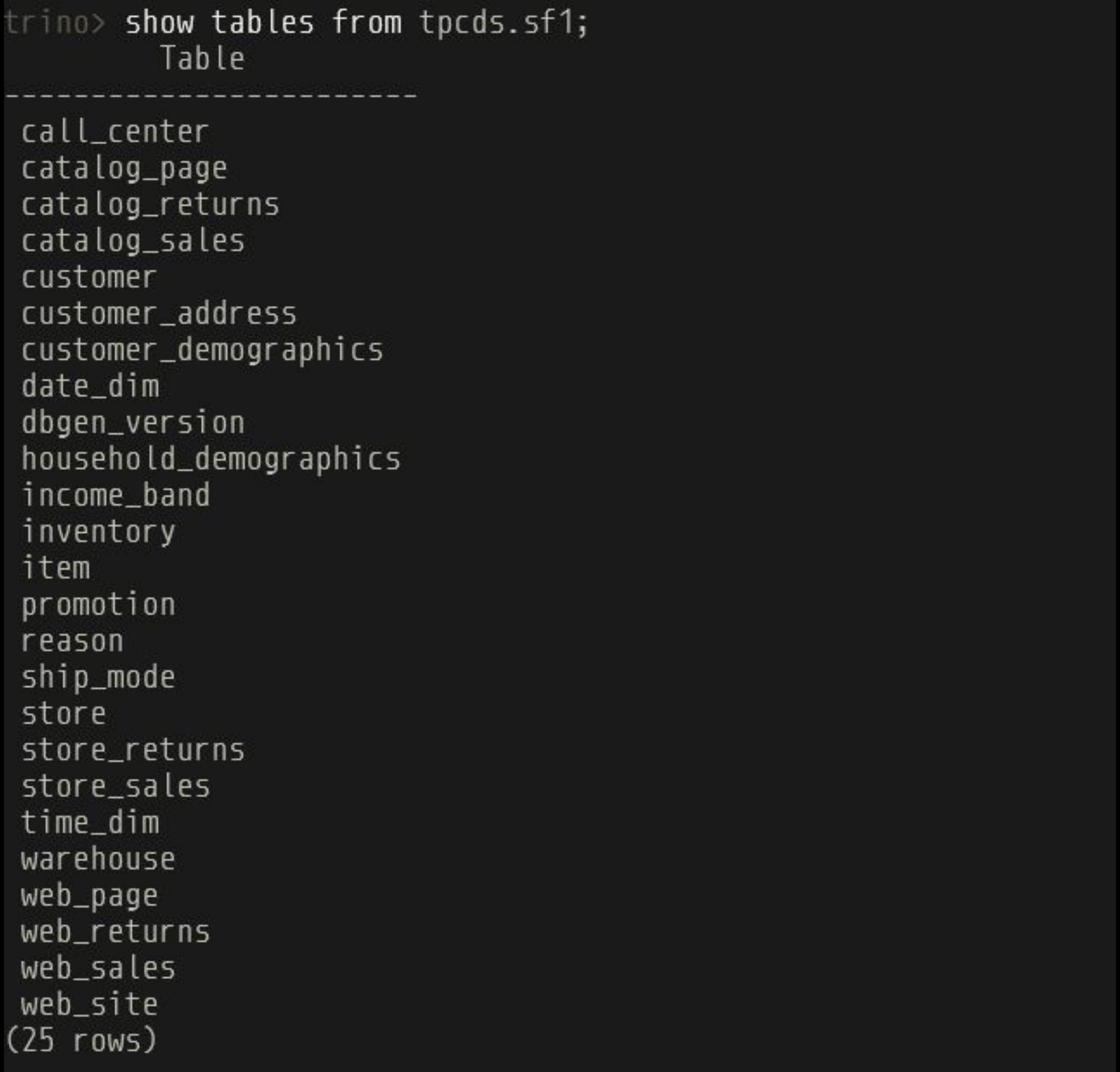
Policy ID ▲	Policy Name	Policy Labels	Status	Audit Logging	Roles	Groups	Users	Action
20	all - trinouser	--	Enabled	Enabled	--	public	--	
21	all - catalog	--	Enabled	Enabled	--	public	--	
22	all - function	--	Enabled	Enabled	--	public	--	
23	all - catalog, sessionproperty	--	Enabled	Enabled	--	public	--	
24	all - catalog, schema, procedure	--	Enabled	Enabled	--	public	--	
25	all - catalog, schema, table	--	Enabled	Enabled	--	public	--	
26	all - systemproperty	--	Enabled	Enabled	--	public	--	
27	all - catalog, schema, table, column	--	Enabled	Enabled	--	--	superuser	
28	all - catalog, schema	--	Enabled	Enabled	--	public	--	
29	Allow information schemas	--	Enabled	Enabled	--	public	--	



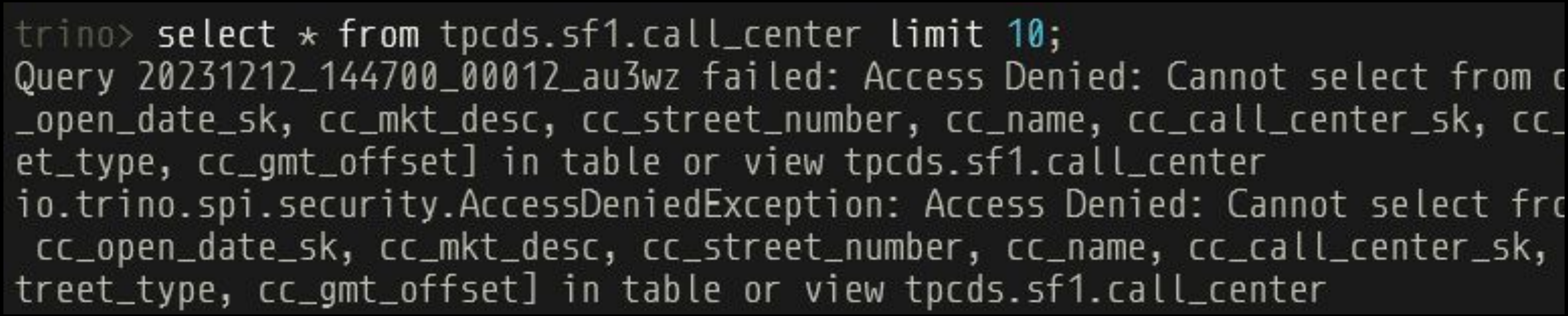
```
trino> SHOW CATALOGS;
      Catalog
--------------------
 irrelevant_catalog
 jmx
 system
 tpcds
(4 rows)

Query 20231212_143818_00008_au3wz, FINISHED, 2 nodes
http://127.0.0.1:8080/ui/query.html?20231212_143818_00008_au3wz
Splits: 20 total, 20 done (100.00%)
CPU Time: 0.1s total,     0 rows/s,     0B/s, 26% active
Per Node: 0.1 parallelism,     0 rows/s,     0B/s
Parallelism: 0.1
Peak Memory: 382B
0.68 [0 rows, 0B] [0 rows/s, 0B/s]
```



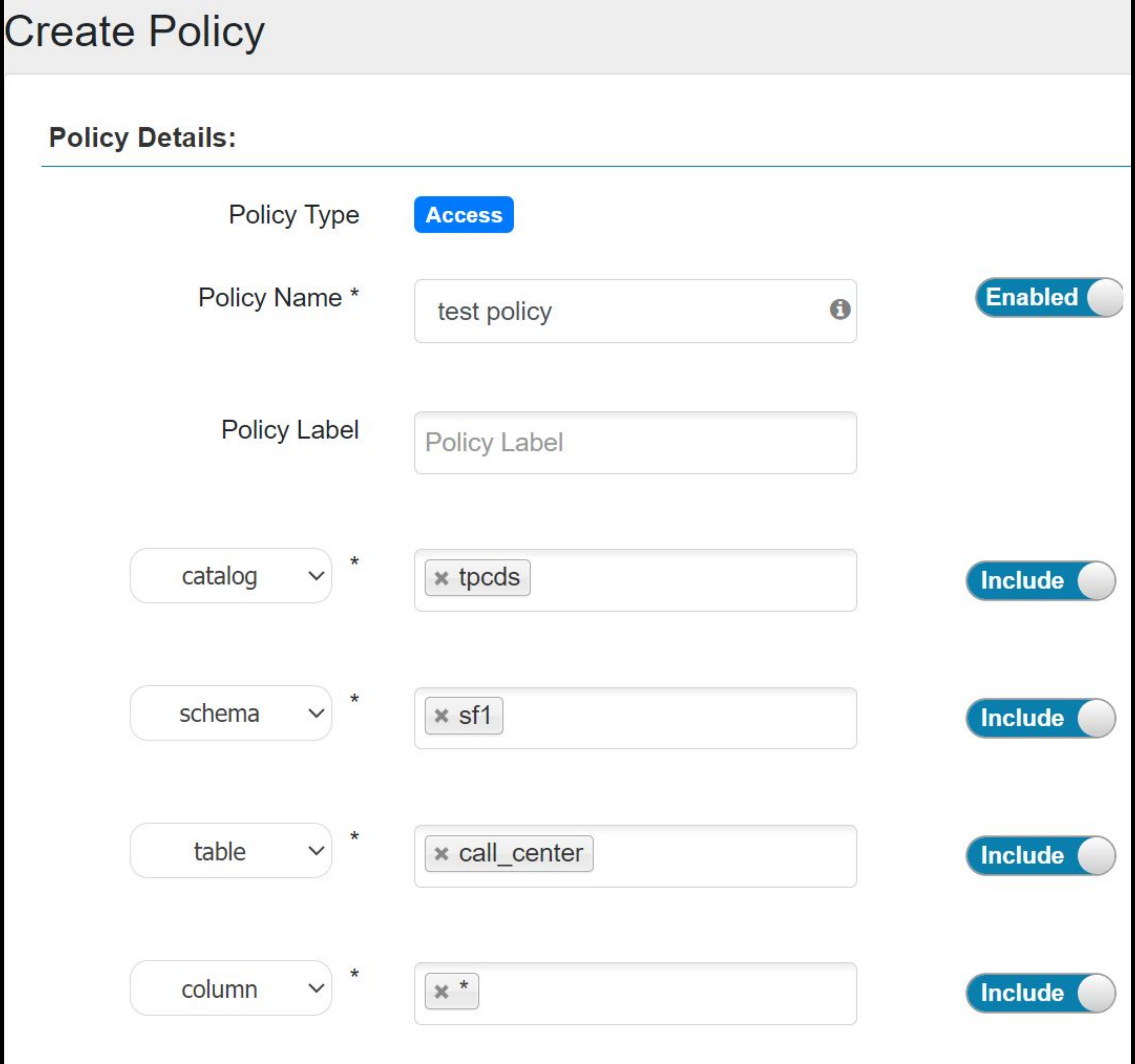
```
trino> show schemas from tpcds;
       Schema
----------------------
 information_schema
 sf1
 sf10
 sf100
 sf1000
 sf10000
 sf100000
 sf300
 sf3000
 sf30000
 tiny
(11 rows)

Query 20231212_144632_00010_au3wz, FINISHED, 2 nodes
http://127.0.0.1:8080/ui/query.html?20231212_144632_00010_au3wz
Splits: 20 total, 20 done (100.00%)
CPU Time: 0.0s total,   314 rows/s, 3.57KB/s, 31% active
Per Node: 0.1 parallelism,    17 rows/s,   201B/s
Parallelism: 0.1
Peak Memory: 1.34KB
0.32 [11 rows, 128B] [34 rows/s, 401B/s]
```



```
trino> show tables from tpcds.sf1;
         Table
------------------------
 call_center
 catalog_page
 catalog_returns
 catalog_sales
 customer
 customer_address
 customer_demographics
 date_dim
 dbgen_version
 household_demographics
 income_band
 inventory
 item
 promotion
 reason
 ship_mode
 store
 store_returns
 store_sales
 time_dim
 warehouse
 web_page
 web_returns
 web_sales
 web_site
(25 rows)
```



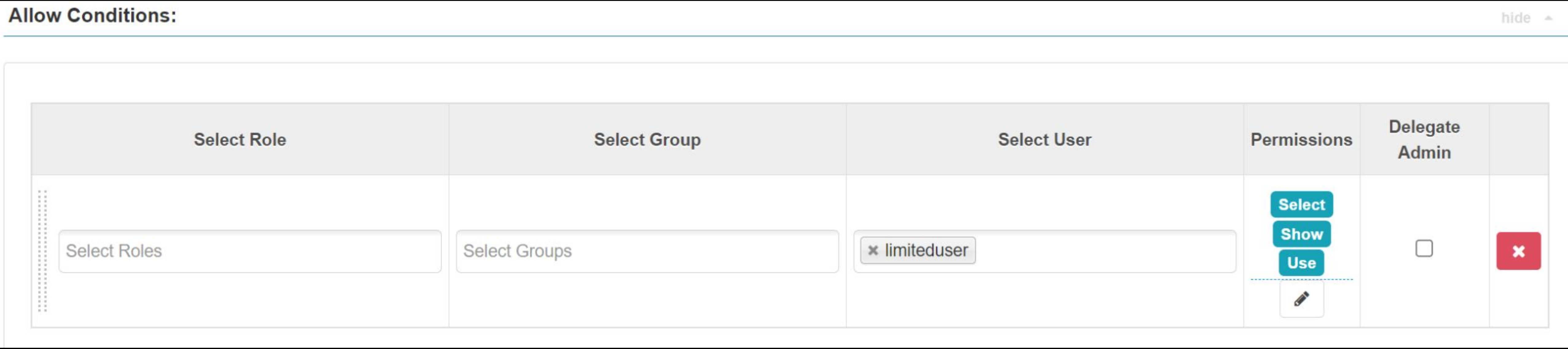
```
trino> select * from tpcds.sf1.call_center limit 10;
Query 20231212_144700_00012_au3wz failed: Access Denied: Cannot select from c
_open_date_sk, cc_mkt_desc, cc_street_number, cc_name, cc_call_center_sk, cc_
et_type, cc_gmt_offset] in table or view tpcds.sf1.call_center
io.trino.spi.security.AccessDeniedException: Access Denied: Cannot select fro
 cc_open_date_sk, cc_mkt_desc, cc_street_number, cc_name, cc_call_center_sk,
treet_type, cc_gmt_offset] in table or view tpcds.sf1.call_center
```



Create Policy

Policy Details:

Policy Type	Access	
Policy Name *	test policy	Enabled
Policy Label	Policy Label	
catalog *	× tpcds	Include
schema *	× sf1	Include
table *	× call_center	Include
column *	× *	Include



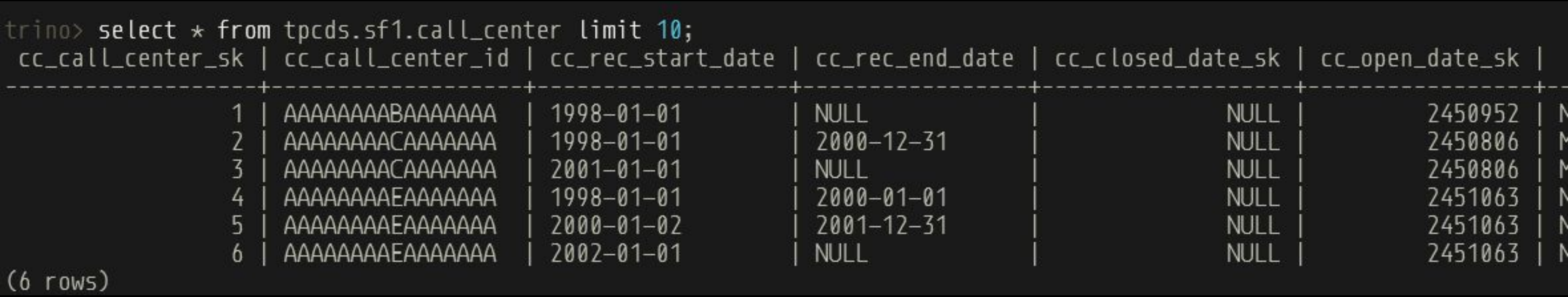
Allow Conditions:

hide

Select Role	Select Group	Select User	Permissions	Delegate Admin	
Select Roles	Select Groups	× limiteduser	Select Show Use	☐	✖

https://<ranger>/service/plugins/policies/exportJson

```
{
    "metaDataInfo": {
        "Host name": "1ab01992b468",
        "Exported by": "admin",
        "Export time": "Dec 12, 2023, 2:54:15 PM",
        "Ranger apache version": "2.3.1"
    },
    "policies": [
        {
            "service": "trinoservice",
            "name": "all - trinouser",
            "policyType": 0,
            "policyPriority": 0,
            "description": "Policy for all - trinouser",
            "isAuditEnabled": true,
            "resources": {
                "trinouser": {
                    "values": [
                        "*"
                    ],
                    "isExcludes": false,
                    "isRecursive": false
                }
            },
            "policyItems": [
```



```
trino> select * from tpcds.sf1.call_center limit 10;
 cc_call_center_sk | cc_call_center_id | cc_rec_start_date | cc_rec_end_date | cc_closed_date_sk | cc_open_date_sk |
-------------------+-------------------+-------------------+-----------------+-------------------+-----------------+--
                 1 | AAAAAAAABAAAAAAA  | 1998-01-01        | NULL            |              NULL |         2450952 | N
                 2 | AAAAAAAACAAAAAAA  | 1998-01-01        | 2000-12-31      |              NULL |         2450806 | M
                 3 | AAAAAAAACAAAAAAA  | 2001-01-01        | NULL            |              NULL |         2450806 | M
                 4 | AAAAAAAAEAAAAAAA  | 1998-01-01        | 2000-01-01      |              NULL |         2451063 | N
                 5 | AAAAAAAAEAAAAAAA  | 2000-01-02        | 2001-12-31      |              NULL |         2451063 | N
                 6 | AAAAAAAAEAAAAAAA  | 2002-01-01        | NULL            |              NULL |         2451063 | N
(6 rows)
```

Thank you!

https://www.bloomberg.com/careers

Contact me: parteagagonz@bloomberg.net

Bloomberg Engineering

TechAtBloomberg.com

© 2023 Bloomberg Finance L.P. All rights reserved.